

علوی

مجموعه مدارس متوسطه اول علوی

# سه گانه فناوری

(سخت افزار، نرم افزار و ذهن افزار)

پایه هشتم



## فهرست

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| بخش اول: مبانی الکترونیک.....                                                  | ۴   |
| درس ۱: برق و مدار الکتریکی.....                                                | ۵   |
| درس ۲: مدار الکتریکی و کاربرد آن.....                                          | ۹   |
| درس ۳: موتور الکتریکی چگونه کار می کند؟.....                                   | ۱۳  |
| بخش دوم: نرم افزار.....                                                        | ۱۶  |
| درس ۴: آشنایی با فتوشاپ و محیط کار.....                                        | ۱۷  |
| درس ۵: آشنایی با ابزارهای انتخاب.....                                          | ۲۲  |
| درس ۶: آشنایی با ابزارهای ترسیم، متن و طراحی در فتوشاپ.....                    | ۲۷  |
| درس ۷: آشنایی با لایه ها در فتوشاپ.....                                        | ۳۲  |
| درس ۸: مدیریت لایه ها با ماسک، شفافیت و گروه بندی.....                         | ۳۷  |
| درس ۹: تنظیم نور، رنگ و روتوش تصاویر در فتوشاپ.....                            | ۴۲  |
| درس ۱۰: بازگشت به دنیای پایتون و آمادگی برای بینایی ماشین.....                 | ۴۸  |
| درس ۱۱: مهندسی کد؛ توابع و ساختاردهی پروژه.....                                | ۵۴  |
| درس ۱۲: آماده سازی کارگاه؛ نصب و راه اندازی کتابخانه OpenCV.....               | ۵۸  |
| درس ۱۳: تصویر دیجیتال در پشت صحنه؛ پیکسل ها و مدل های رنگی.....                | ۶۳  |
| درس ۱۴: اولین تعامل با تصویر؛ خواندن، نمایش و ذخیره تصویر با OpenCV.....       | ۶۸  |
| درس ۱۵: آماده سازی تصویر برای تشخیص چهره؛ تغییر اندازه و تبدیل به خاکستری..... | ۷۳  |
| درس ۱۶: پنجره ای رو به دنیای واقعی؛ دوربین و فریم زنده با OpenCV.....          | ۷۹  |
| درس ۱۷: مغز متفکر پروژه؛ آشنایی با الگوریتم Haar Cascade و تشخیص چهره.....     | ۸۳  |
| درس ۱۸: رسم کادر و تشخیص چهره در ویدئوی زنده.....                              | ۹۱  |
| درس ۱۹: جستجوی هوشمند؛ تشخیص اجزای صورت.....                                   | ۹۵  |
| درس ۲۰: آشنایی با چاپ سه بعدی.....                                             | ۹۸  |
| درس ۲۱: آشنایی با اجزای پرینتر سه بعدی و فرآیند چاپ.....                       | ۱۰۳ |
| درس ۲۲: طراحی سه بعدی با نرم افزار Tinkercad.....                              | ۱۰۷ |
| بخش سوم: ذهن افزار.....                                                        | ۱۱۴ |
| درس ۲۳: تفکر محاسباتی و حل مسئله.....                                          | ۱۱۵ |
| درس ۲۴: آشنایی با هوش مصنوعی.....                                              | ۱۲۲ |
| درس ۲۵: پرامپت نویسی برای هوش مصنوعی.....                                      | ۱۲۷ |

## سخنی با دبیر ارجمند

فناوری رایانه امروزه یکی از مهم‌ترین مهارت‌های زندگی در دنیای جدید است. هدف این کتاب، آشنایی دانش‌آموزان پایه هفتم با مفاهیم پایه رایانه و ایجاد مهارت‌های عملی برای ورود آگاهانه و خلاقانه به دنیای دیجیتال است.

این کتاب در سه بخش اصلی طراحی شده است:

**بخش اول سخت/فزار:** در این بخش دانش‌آموزان با اجزای اصلی رایانه، قطعات داخلی و جانبی، نحوه عملکرد آن‌ها، روش‌های عیب‌یابی و اصول اسمبل کردن یک سیستم کامپیوتری آشنا می‌شوند.

**بخش دوم نرم‌افزار:** در این بخش، دانش‌آموزان نصب و راه‌اندازی سیستم‌عامل، تنظیمات ویندوز، نرم‌افزارهای کاربردی مجموعه آفیس مانند ورد، پاورپوینت و اکسل را یاد می‌گیرند. همچنین با نقش هوش مصنوعی در استفاده بهتر از نرم‌افزارها، آشنایی مقدماتی با پایتون و کاربرد فناوری‌هایی مانند پرینتر سه‌بعدی آشنا خواهند شد.

**بخش سوم ذهن/فزار:** در این بخش، دانش‌آموزان با سواد و امنیت دیجیتال، شیوه استفاده صحیح و مسئولانه از فناوری و مهارت‌های لازم برای ورود به دنیای دیجیتال آشنا می‌شوند.

در پایان هر درس، فعالیت‌ها و پروژه‌هایی برای تقویت یادگیری و افزایش مهارت عملی دانش‌آموزان ارائه شده است. این پروژه‌ها بخشی از ارزیابی درس بوده و دانش‌آموزان باید آن‌ها را جمع‌آوری و ارائه دهند. همچنین ارزشیابی درس شامل آزمون کتبی و عملی خواهد بود تا هم دانش نظری و هم توانایی انجام کارهای واقعی بررسی شود.

امید است این جزوه بتواند مسیر یادگیری، خلاقیت و استفاده درست از فناوری را برای دانش‌آموزان عزیز هموار کند.

## سخنی با دانش‌آموزان گرامی

سلام!

در این جزوه، رایانه را تنها به‌عنوان یک وسیله برای کارهای روزمره نمی‌شناسیم، بلکه با مبانی الکترونیک، نرم‌افزارهای کاربردی، برنامه‌نویسی، هوش مصنوعی و فناوری‌های نوین آشنا می‌شویم. در کنار یادگیری مفاهیم، با انجام فعالیت‌ها و پروژه‌های عملی، مهارت حل مسئله، خلاقیت و استفاده صحیح از فناوری را تقویت می‌کنیم. این مسیر به ما کمک می‌کند تا با دنیای سخت‌افزار، نرم‌افزار و ذهن‌افزار آگاهانه‌تر و توانمندتر وارد شویم.

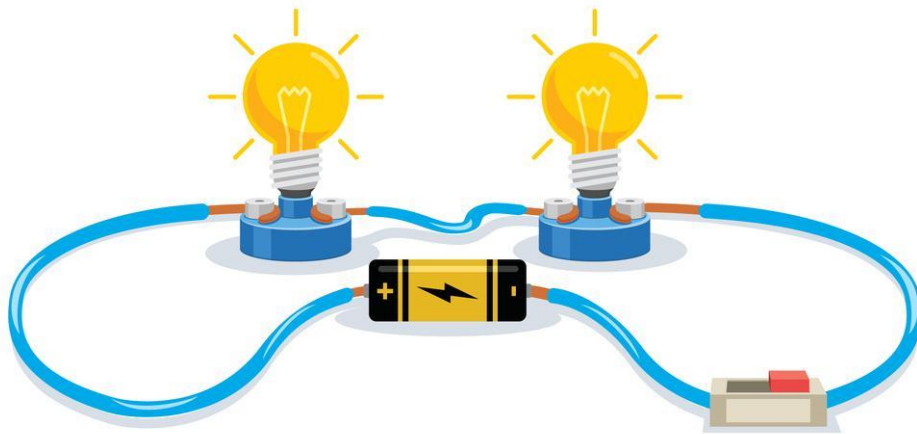
در این جزوه برای ساده‌تر شدن مطالب، از برخی علامت‌های اختصاری استفاده شده است:

|                           |                                                                                     |                                   |                                                                                       |
|---------------------------|-------------------------------------------------------------------------------------|-----------------------------------|---------------------------------------------------------------------------------------|
| خلاصه درس                 |  | مقدمه و شروع هر درس               |  |
| تمرین در منزل             |  | ورود به مطالب اصلی و توضیح مفاهیم |  |
| پروژه نهایی درس           |  | دستیار هوشمند و هوش مصنوعی        |  |
| چالش برای دانش‌آموزان قوی |  | آیا می‌دانید و اطلاعات بیشتر      |  |
|                           |                                                                                     | فعالیت کلاسی                      |  |

**بخش اول: مبانی الکترونیک**

**Fundamentals of Electronics**

## درس ۱: برق و مدار الکتریکی



در پایان این درس خواهید توانست:

- مفهوم انرژی الکتریکی را بیان کنید.
- اجزای اصلی یک مدار الکتریکی را بشناسید.
- نقش ولتاژ، جریان و مقاومت را در مدار توضیح دهید.
- یک مدار ساده برای روشن کردن لامپ بسازید.



تصور کنید در یک شب تاریک وارد اتاق می شوید. با فشردن کلید، اتاق روشن می شود. یا زمانی که دکمه کنترل تلویزیون را فشار می دهید، تلویزیون روشن می شود و با لمس صفحه تلفن همراه، برنامه های مختلف اجرا می شوند.

آیا تاکنون فکر کرده اید چه چیزی باعث می شود همه این وسایل به سرعت کار کنند؟ پاسخ این پرسش برق است. برق یکی از مهم ترین شکل های انرژی است که بسیاری از وسایل اطراف ما با کمک آن کار می کنند. در این درس با مفاهیم اولیه برق و مدار الکتریکی آشنا می شوید و در پایان، یک مدار ساده برای روشن کردن لامپ می سازید.

به وسایل اطراف خود در خانه یا مدرسه توجه کنید.

۱. اگر برق وجود نداشت، انجام کدام یک از کارهای روزانه شما دشوار می شد؟

۲. به نظر شما برق چگونه باعث روشن شدن یک لامپ می شود؟

درباره پاسخ های خود با هم کلاسی ها گفت و گو کنید.

## آشنایی با مصارف الکتریسته



بیشتر وسایل و تجهیزات پیرامون ما از انرژی الکتریکی تغذیه می کنند و این انرژی به سایر انرژی های مورد نیاز زندگی قابل تبدیل است. انرژی الکتریکی معمولاً با روش های مختلف تولید می شود و در بخش صنعتی، شامل کارخانه ها و کارگاه ها و در بخش مصرف کننده های، خانگی، اداری اماکن عمومی و تجاری به مصرف می رسد.

## فعالیت کلاس

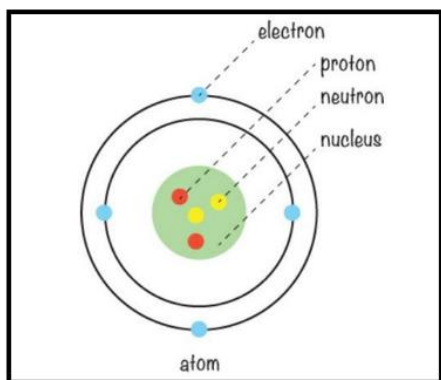
چگونه یک لامپ را روشن کنیم؟

وقتی که در یک اتاق کلید چراغ را فشار می دهید، لامپ بلافاصله روشن می شود. بیایید ببینیم برق چگونه باعث می شود این لامپ نور کند. برای شروع، یک آزمایش ساده انجام می دهیم.

برای انجام این پروژه به قطعات زیر نیاز دارید:

- باتری ۹ ولتی استاندارد: منبع انرژی مورد نیاز برای مدار شما.
- یک لامپ کوچک رشته ای (نور رشته ای) که برای ۹ تا ۱۲ ولت طراحی شده باشد.

## برق چگونه لامپ را روشن می کند؟



اما چطور برق باعث گرم شدن سیم درون لامپ می شود و چرا نور بلافاصله روشن می شود؟ چهار مفهوم اصلی وجود دارند که با هم ترکیب می شوند

تا این اتفاق بیفتد:

- الکترون ها
- جریان
- ولتاژ
- مقاومت

همه مواد اطراف ما از ذرات بسیار کوچکی به نام اتم ساخته شده‌اند. هر اتم از پروتون، نوترون و الکترون تشکیل می‌شود. پروتون‌ها بار مثبت و الکترون‌ها بار منفی دارند و همین بارهای الکتریکی نقش مهمی در پدیده‌های الکتریکی ایفا می‌کنند. در برخی مواد که به آن‌ها رسانا گفته می‌شود، الکترون‌ها می‌توانند با دریافت انرژی (مانند انرژی باتری) از اتمی به اتم دیگر حرکت کنند. این حرکت الکترون‌ها پایه ایجاد برق است.

وقتی باتری به یک مدار وصل می‌شود، ولتاژ ایجادشده باعث حرکت الکترون‌ها در سیم و به وجود آمدن جریان الکتریکی می‌شود. ولتاژ با واحد ولت (V) و جریان با واحد آمپر (A) اندازه‌گیری می‌شود.

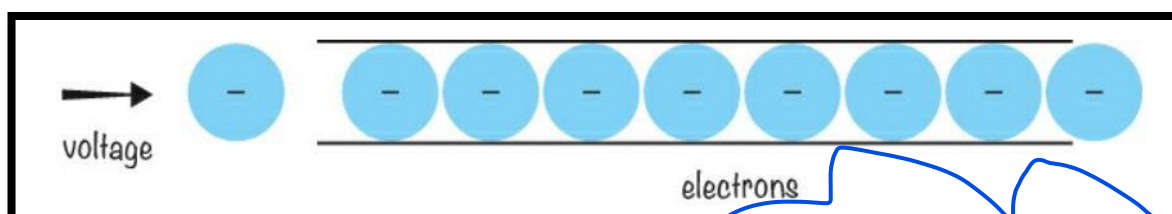
در برابر حرکت الکترون‌ها، عاملی به نام مقاومت وجود دارد که مقدار جریان را کاهش می‌دهد. مقاومت با واحد اهم ( $\Omega$ ) سنجیده می‌شود و نقش مهمی در کنترل جریان الکتریکی دارد.

عبور جهت‌دار الکترون‌ها از یک رسانا یا هادی الکتریسته شدت جریان الکتریکی نام دارد که با حرف I نشان داده می‌شود و واحد آن آمپر (A) است. یک منبع می‌تواند آب را در مسیری مشخص به جریان اندازد؛ در مدار الکتریکی به جای آب، الکترون در مسیر سیم جریان دارد هر چه میزان الکترون عبوری از سیم بیشتر باشد جریان بیشتر است.

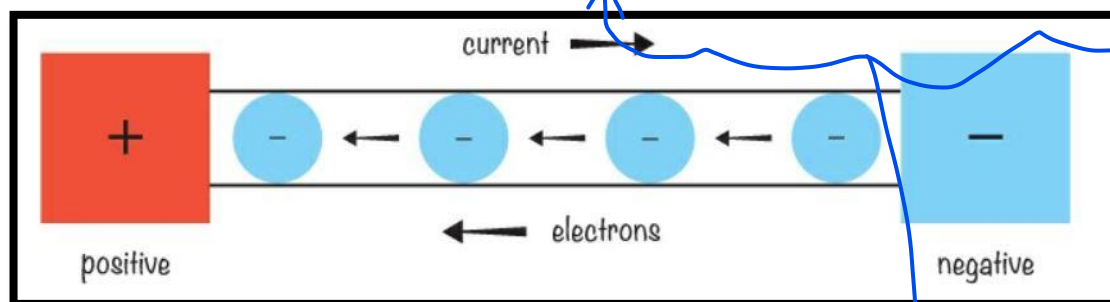
نکته: چون پتانسیل مثبت بیشتر و قویتر از پتانسیل منفی تعریف میشود در مدارهای الکتریکی طبق قرارداد جهت جریان الکتریکی از قطب مثبت به قطب منفی در نظر گرفته میشود

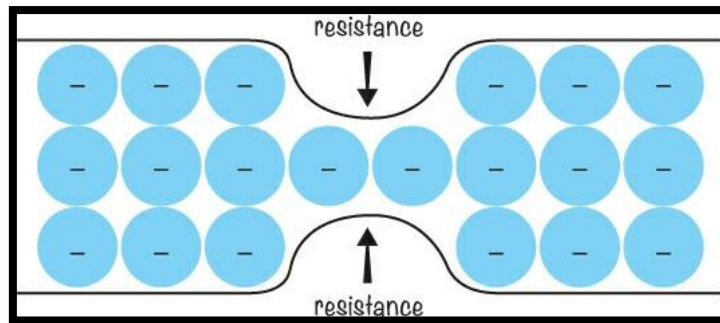
برخی از مواد رسانا هستند، این به چه معنی است؟ اگر به این مواد انرژی بدهید (مثل انرژی ذخیره شده در یک باتری)، الکترون‌ها شروع به حرکت از یک اتم به اتم دیگر می‌کنند! رشته فلزی (فیلامنت) درون لامپ شما از یک فلز رسانا ساخته شده است. این یعنی پر از الکترون‌هایی است که فقط منتظرند تا یک نیروی کوچک آن‌ها را به حرکت دریاورد!

ولتاژ الکترون‌ها را هل می‌دهد.



مقاومت، جریان را کاهش می‌دهد.

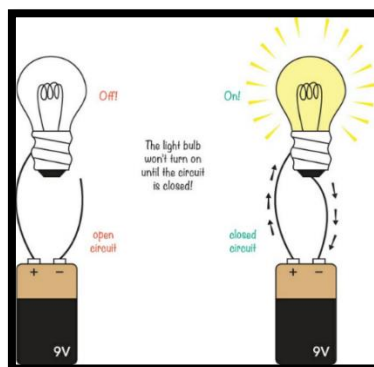




درون لامپ، رشته‌ای فلزی به نام فیلامنت قرار دارد که دو سر آن به قسمت‌های فلزی لامپ وصل است. وقتی باتری به لامپ متصل می‌شود، یک مدار الکتریکی تشکیل می‌گردد.

در مدار، الکترون‌ها در بخش خارجی مدار از قطب منفی باتری به سمت قطب مثبت حرکت می‌کنند؛ اما جهت قراردادی جریان الکتریکی از قطب مثبت به سمت قطب منفی در نظر گرفته می‌شود. ولتاژ یا اختلاف پتانسیل، نیروی لازم برای به حرکت درآمدن منظم الکترون‌ها در مدار را فراهم می‌کند.

فیلامنت دارای مقاومت است و عبور جریان از آن باعث گرم شدن رشته و در نهایت تولید نور می‌شود. اگر مدار در هر نقطه‌ای قطع شود، جریان برقرار نخواهد شد و لامپ خاموش می‌ماند. بنابراین، برای کار کردن وسایل برقی، وجود مدار بسته ضروری است.



### خلاصه درس



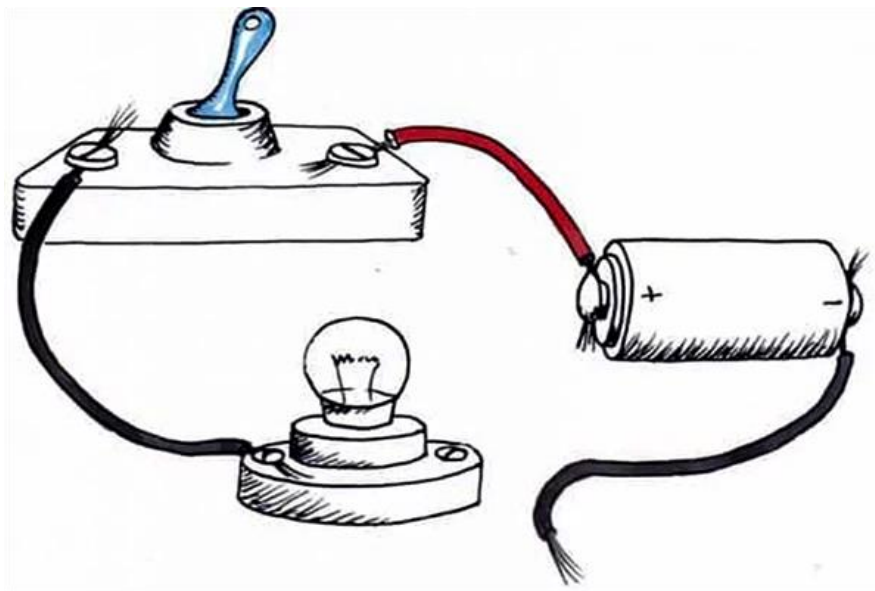
- برق یکی از مهم‌ترین شکل‌های انرژی است.
- حرکت الکترون‌ها در مواد رسانا باعث ایجاد جریان الکتریکی می‌شود.
- ولتاژ باعث حرکت الکترون‌ها و مقاومت باعث کاهش و کنترل جریان الکتریکی می‌شود.
- برای روشن شدن لامپ باید یک مدار بسته تشکیل شود؛ جریان با عبور از فیلامنت، آن را گرم کرده و باعث تولید نور می‌شود.

### پروژه



با استفاده از یک باتری، لامپ و سیم، یک مدار ساده بسازید.

## درس ۲: مدار الکتریکی و کاربرد آن



در پایان این درس خواهید توانست:

- عملکرد مدار الکتریکی را توضیح دهید.
- یک مدار ساده برای ساخت هشدار باز شدن در طراحی و اجرا کنید.

در این درس با نحوه عملکرد مدار الکتریکی آشنا می شوید و یاد می گیرید چگونه از این مفهوم برای ساخت یک هشدار ساده استفاده کنید. به تصویر یک سیستم لوله کشی آب و یک مدار الکتریکی نگاه کنید. به نظر شما:

۱. باتری در مدار شبیه کدام بخش لوله کشی است؟

۲. اگر مسیر لوله یا سیم قطع شود، چه اتفاقی می افتد؟

۳. کلید چه نقشی در مدار دارد؟

درباره پاسخ های خود با هم کلاسی ها گفت و گو کنید.

### مدار الکتریکی چگونه کار می کند؟



برای درک بهتر مدار الکتریکی، می توان آن را با یک سیستم لوله کشی آب مقایسه کرد:

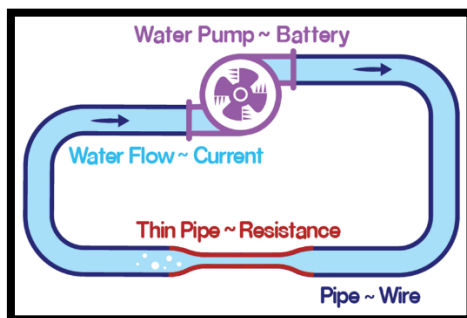
باتری مانند پمپ عمل می کند و با ایجاد فشار (ولتاژ) باعث حرکت جریان می شود. مقاومت مثل قسمت باریک لوله است که عبور جریان را سخت تر کرده و مقدار آن را کاهش می دهد. جریان الکتریکی نشان می دهد چه مقدار بار الکتریکی در هر ثانیه از یک نقطه در مدار عبور می کند. همان طور که آب فقط در یک مسیر حلقه ای و بسته جریان دارد، جریان برق هم فقط در مدار بسته برقرار می شود.

همچنین با کلید (سوئیچ) می توان مدار را کنترل کرد:

وقتی کلید بسته باشد، دو سیم به هم وصل می شوند، مدار کامل می شود و چراغ روشن است.

وقتی کلید باز باشد، اتصال قطع می شود، مدار ناقص می ماند و چراغ خاموش می شود.

کلیدها در وسایل مختلف مانند چراغ ها، رایانه، زنگ در و حتی در یخچال وجود دارند و نقش اصلی آن ها روشن و خاموش کردن یا کنترل جریان برق است.



- پمپ = باتری

پمپ باعث فشار و جریان آب در لوله می شود (مثل باتری که ولتاژ ایجاد می کند و جریان الکتریکی را در مدار به حرکت درمی آورد).

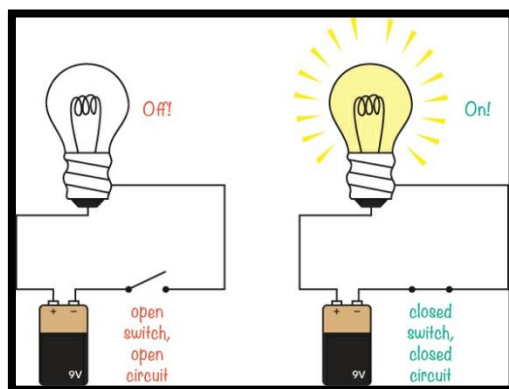
- باریکی در لوله = مقاومت<sup>۱</sup>

این قسمت جریان آب را کم می کند (مثل مقاومت در مدار که باعث کاهش جریان الکتریکی می شود).

<sup>۱</sup> Resistance

• جریان آب در لوله = جریان الکتریکی<sup>۲</sup>

هرچقدر آب بیشتری از داخل لوله عبور کند، جریان بیشتری داریم (مثل عبور تعداد بیشتری الکترون در سیم). پس مثل این که در یک سیستم لوله کشی، آب فقط وقتی جریان دارد که حلقه کامل باشد و پمپ روشن باشد، در مدار الکتریکی هم جریان فقط در یک مسیر بسته و با وجود منبع ولتاژ (باتری) برقرار می شود.



### فعالیت کلاس

هشدار دزدی (آلارم)

در این فصل یاد گرفتیم که برق برای انجام هر کار جالبی، به یک مدار بسته نیاز دارد. همچنین دیدیم که یک کلید چگونه مدار را باز و بسته می کند. حالا وقت آن است که یک مدار با کلید بسازیم! اما در این پروژه، کلید ما یک در خانه خواهد بود! ما قرار است در را تبدیل کنیم به یک کلید عظیم که وقتی کسی سعی کند وارد اتاق کند، یک هشدار صوتی به صدا در آید.

چگونه آلارم دزدی ما کار می کند؟ یک سیم بیرون افتاده را از بالای در آویزان می کنیم. یک تکه فویل آلومینیومی روی بالای در می چسبانیم. هر کدام از این قطعات را به یک طرف مدار وصل می کنیم.

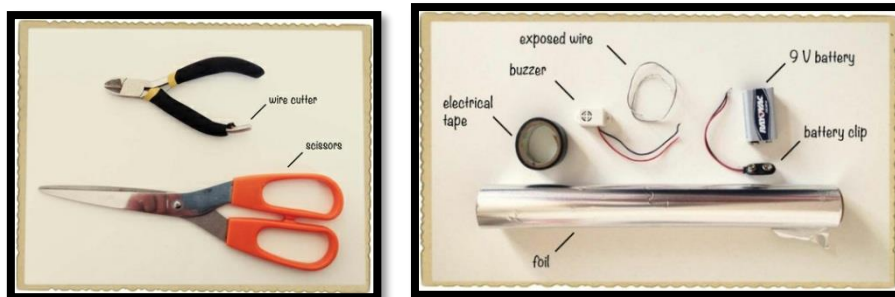
وقتی در بسته است → سیم و فویل با هم تماس ندارند → مدار باز است و هیچ اتفاقی نمی افتد.

ولی وقتی در باز می شود → سیم آویزان به فویل آلومینیومی می رسد و با آن تماس پیدا می کند → مدار بسته می شود و شروع به زنگ زدن می کند!

وسایل مورد نیاز: باتری ۹ ولتی - بازر - لنت برق - ورقه فویل - سر باتری - سیم بدون پوشش

ابزار مورد نیاز: سیم چین - قیچی

<sup>۲</sup> Current



### خلاصه درس



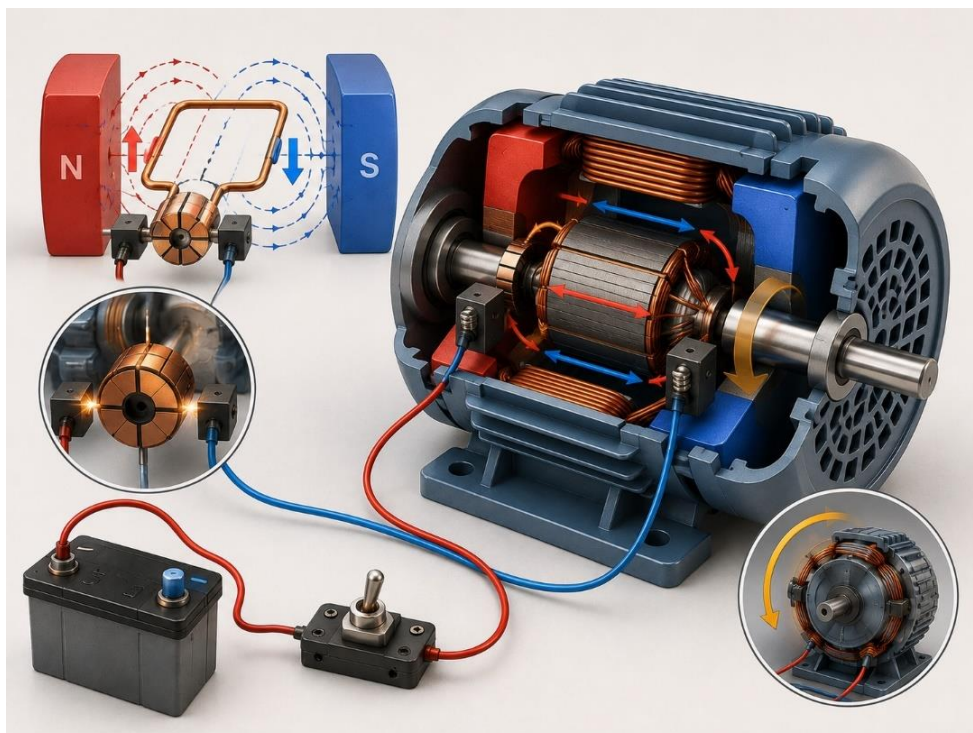
- مقاومت مقدار جریان الکتریکی را کاهش می دهد.
- کلید مدار را باز یا بسته می کند.
- جریان برق فقط در مدار بسته برقرار می شود.

### پروژه



با استفاده از باتری، بازر، سیم و یک کلید، مداری بسازید که با فشردن کلید، بازر به صدا درآید.

### درس ۳: موتور الکتریکی چگونه کار می کند؟



در پایان این درس خواهید توانست:

- مفهوم موتور الکتریکی را بیان کنید.
- نقش آهنربا و سیم پیچ را در موتور توضیح دهید.
- دلیل چرخش موتور الکتریکی را بیان کنید.
- یک موتور الکتریکی ساده بسازید.

پنکه، مخلوط کن، اسباب بازی های برقی، ماشین لباس شویی و بسیاری از وسایل دیگر چگونه حرکت می کنند؟ درون همه این وسایل، قطعه ای به نام موتور الکتریکی وجود دارد که انرژی الکتریکی را به انرژی حرکتی تبدیل می کند. در این درس با نحوه کار موتور آشنا می شوید و یک نمونه ساده از آن را خواهید ساخت. به وسایل اطراف خود نگاه کنید.

۱. سه وسیله ای را نام ببرید که در آن ها موتور الکتریکی وجود دارد.

۲. به نظر شما اگر موتور از این وسایل خارج شود، چه اتفاقی می افتد؟

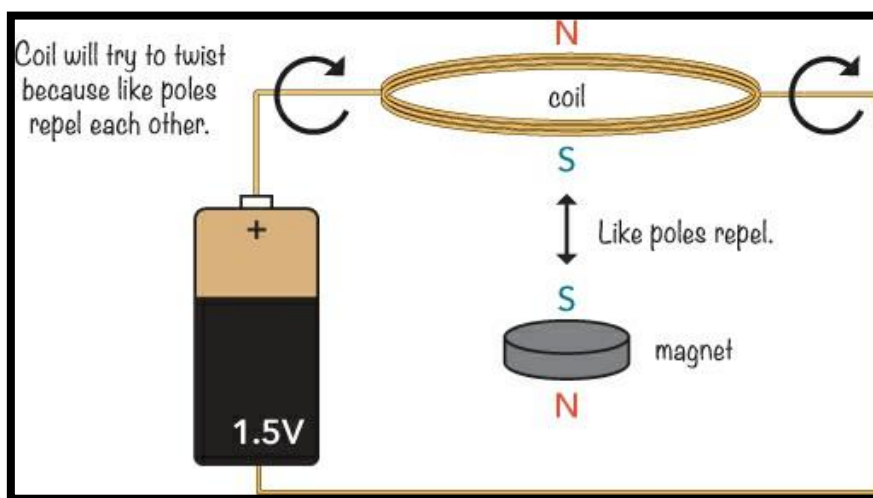
۳. موتور چگونه می تواند با استفاده از برق، حرکت ایجاد کند؟

دوباره پاسخ های خود گفت و گو کنید.

### آشنایی با موتور



سیمی که جریان الکتریکی از آن عبور می کند، یک میدان مغناطیسی ایجاد می کند. وقتی سیم پیچ به برق متصل باشد، دارای یک میدان مغناطیسی با قطب های جنوب و شمال خواهد بود، دقیقاً مانند هر آهنربای دیگری. قطب های مشابه یکدیگر را دفع می کنند و قطب های مخالف یکدیگر را جذب می کنند. بنابراین، اگر سیم پیچی که به یک آهنربا تبدیل شده است را روی یک آهنربای معمولی قرار دهید و قطب های یکسان آن ها نزدیک به هم باشند، سیم پیچ سعی خواهد کرد که خودش را بچرخاند.



اگر سیم پیچ را روی نوعی پایه قرار دهید به طوری که بتواند آزادانه روی آهنربا بچرخد، سیم پیچ بدون اینکه یک دور کامل بزند، به جلو و عقب تکان خواهد خورد. دلیل این امر آن است که وقتی سیم پیچ نیم دور بچرخد، قطب های مخالف رو به هم قرار می گیرند و یکدیگر را جذب می کنند و این باعث می شود سیم پیچ در جهت عکس حرکت کند. چگونه می توانید سیم پیچ را مدام در یک جهت بچرخانید؟

کافی است راهی پیدا کنید که باتری را درست در وسط چرخش قطع کنید و دوباره آن را وقتی سیم پیچ به نقطه شروع برگشت، وصل کنید. در این صورت این اتفاق رخ می دهد: سیم پیچ زمانی که برق دارد شروع به حرکت می کند و سیم پیچ را نیم دور به جلو هل می دهد. چون در وسط چرخش باتری قطع می شود، اما حرکت اولیه (اینرسی) سیم پیچ را به حرکت خود ادامه می دهد. وقتی سیم پیچ دوباره به محل اولیه خود بازگشت، باتری دوباره وصل می شود و سیم پیچ را دوباره به

جلو هل می دهد و این چرخه به همین ترتیب ادامه پیدا می کند. موتورهای الکتریکی دقیقاً بر اساس همین اصل اولیه جذب و دفع قطب های مغناطیسی کار می کنند.

## فعالیت کلاسی

ساخت یک موتور

در این پروژه، شما یک موتور الکتریکی ساده از ابتدا بسازید!

چه چیزهایی نیاز دارید؟

- یک سیم پیچ
- یک آهنربای قوی (مثل یک آهنربای دیسکی یا نواری)



## نحوه کار موتور

در این موتور ساده، سیم پیچ نقش بخش چرخان را دارد که به آن روتور گفته می شود. موتور به گونه ای ساخته می شود که جریان الکتریکی فقط در نیمی از هر چرخش از سیم پیچ عبور کند. در این حالت، نیروی مغناطیسی آهنربا باعث چرخیدن سیم پیچ می شود.

در نیمه دیگر چرخش، جریان قطع می شود، اما اینرسی (حرکت ادامه دار جسم) باعث می شود روتور به چرخش خود ادامه دهد. با تکرار این فرایند، روتور به طور پیوسته می چرخد. این پروژه، اصل کار موتور الکتریکی را به صورت ساده و عملی نشان می دهد.

وسایل مورد نیاز: لنت الکتریکی - لیوان یک بار مصرف - باتری - سیم - آهنربا و سیم چین

## خلاصه درس

- موتور قطعه ای است که انرژی الکتریکی را به انرژی حرکتی تبدیل می کند و در وسایلی مانند پنکه، مخلوط کن و ماشین لباس شویی وجود دارد.
- عبور جریان الکتریکی از سیم باعث ایجاد میدان مغناطیسی می شود و سیم پیچ مانند یک آهنربا دارای قطب شمال و جنوب می شود.

## پروژه

موتور ساخته شده را آزمایش کنید و بررسی کنید با تغییر تعداد دورهای سیم پیچ یا موقعیت آهنربا، سرعت چرخش موتور چه تغییری می کند. نتیجه مشاهده های خود را در یک جدول ثبت و با هم کلاسی های خود مقایسه کنید.

## بخش دوم: نرم افزار

**SoftWare**

## درس ۴: آشنایی با فتوشاپ و محیط کار



در پایان این درس خواهید توانست:

- با کاربردهای اصلی نرم افزار فتوشاپ آشنا شوید.
- محیط کار فتوشاپ را بشناسید.
- وظیفه هر بخش از محیط نرم افزار را توضیح دهید.
- یک فایل تصویری را در فتوشاپ باز کنید.
- با پنل ها و ابزارهای اصلی فتوشاپ کار کنید.



امروزه تصاویر نقش مهمی در آموزش، تبلیغات، شبکه‌های اجتماعی و طراحی دارند. بسیاری از تصاویر قبل از انتشار، ویرایش یا اصلاح می‌شوند. یکی از معروف‌ترین نرم‌افزارهای ویرایش تصویر، Adobe Photoshop است. فتوشاپ ابزاری قدرتمند برای طراحی و ویرایش تصاویر است و در زمینه‌هایی مانند طراحی پوستر، ساخت لوگو، رتوش عکس، طراحی صفحات وب و تولید محتوای دیجیتال کاربرد دارد. این نرم‌افزار هم توسط طراحان حرفه‌ای و هم توسط علاقه‌مندان به طراحی استفاده می‌شود. به پرسش‌های زیر پاسخ دهید.

۱. تاکنون با چه برنامه‌هایی عکس‌های خود را ویرایش کرده‌اید؟
۲. به نظر شما فتوشاپ در چه مشاغلی کاربرد دارد؟
۳. اگر بخواهید یک پوستر یا تصویر تبلیغاتی طراحی کنید، به چه ابزارهایی نیاز خواهید داشت؟

### فتوشاپ چیست؟



فتوشاپ یک نرم‌افزار حرفه‌ای برای ویرایش و طراحی تصاویر است که توسط شرکت Adobe توسعه یافته است. با استفاده از فتوشاپ می‌توان:

- تصاویر را ویرایش کرد.
- نور و رنگ عکس را اصلاح نمود.
- متن و اشکال گرافیکی ایجاد کرد.
- پوستر، بزر و کارت طراحی نمود.
- عکس‌های قدیمی را ترمیم کرد.
- تصاویر مناسب برای وب و شبکه‌های اجتماعی ساخت.

### نسخه‌های فتوشاپ

فتوشاپ در نسخه‌های مختلفی منتشر شده است.

نسخه‌های قدیمی با نام CS شناخته می‌شدند، اما نسخه‌های جدید با نام Creative Cloud (CC) عرضه می‌شوند. نسخه‌های CC امکانات بیشتری دارند، مرتب به‌روزرسانی می‌شوند و از فناوری‌های جدید مانند ابزارهای مبتنی بر هوش مصنوعی پشتیبانی می‌کنند.

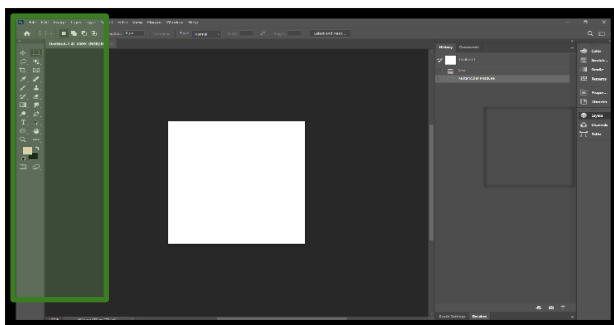
### آشنایی با محیط فتوشاپ

پس از اجرای فتوشاپ، بخش‌های مختلفی را مشاهده می‌کنید که هر کدام وظیفه مشخصی دارند.

#### ۱. نوار ابزار (Tools Panel)

در سمت چپ پنجره قرار دارد و شامل ابزارهای اصلی مانند:

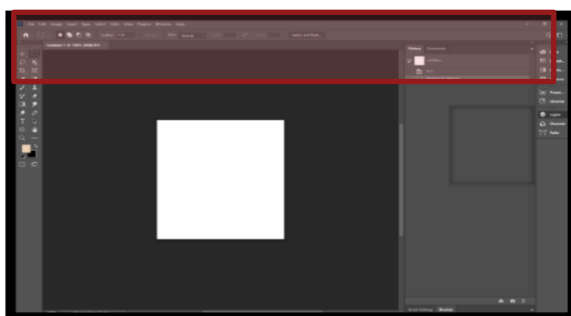
- انتخاب (Selection)
- قلم مو (Brush)



- پاک کن (Eraser)
  - متن (Text)
  - برش تصویر (Crop)
- است.

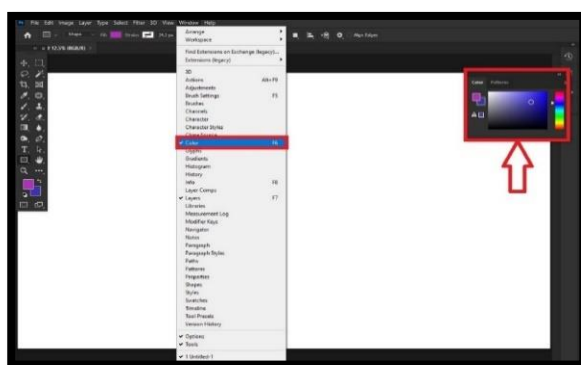
## ۲. نوار تنظیمات (Options Bar)

در بالای صفحه قرار دارد و تنظیمات مربوط به ابزار انتخاب شده را نمایش می دهد. برای مثال، اگر ابزار قلم مو را انتخاب کنید، اندازه و نوع قلم مو در این قسمت قابل تغییر است.



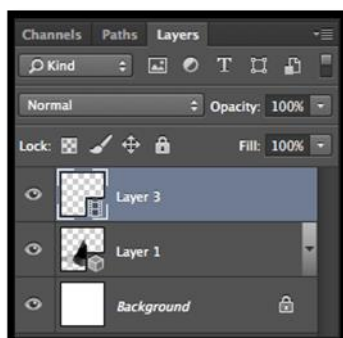
## ۳. پنل لایه ها (Layers)

یکی از مهم ترین بخش های فتوشاپ است. هر تصویر، متن یا شکل در یک لایه جداگانه قرار می گیرد و می توان آن را بدون تغییر سایر بخش ها ویرایش کرد.



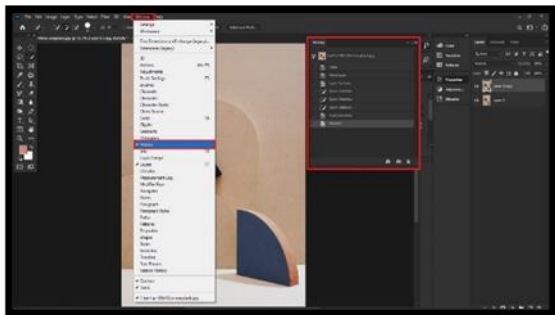
## ۴. پنل رنگ (Color Panel)

برای انتخاب رنگ قلم مو، متن یا اشکال استفاده می شود.



## ۵. پنل تاریخچه (History)

این پنل تمام مراحل انجام شده را ثبت می کند. اگر در هنگام طراحی اشتباهی رخ دهد، می توان به مراحل قبل بازگشت و تغییرات را اصلاح کرد. البته با بستن فایل، تاریخچه پاک می شود.



### نکات مهم

- برای بازگشت به مراحل قبل، از پنل **History** یا میانبر **Ctrl + Z** استفاده کنید.
- هر عنصر را در یک لایه جداگانه ایجاد کنید تا ویرایش آن آسان تر باشد.
- ابزار مناسب را انتخاب کنید و سپس تنظیمات آن را از نوار تنظیمات تغییر دهید.

### آیا می دانید؟



- فتوشاپ یکی از پرکاربردترین نرم افزارهای گرافیکی جهان است.
- بسیاری از پوسترها، جلد کتاب ها، تبلیغات و تصاویر شبکه های اجتماعی با فتوشاپ طراحی می شوند.
- امروزه نسخه های جدید فتوشاپ از ابزارهای مبتنی بر **هوش مصنوعی** برای حذف اشیاء، تولید تصویر و انتخاب هوشمند استفاده می کنند.

### فعالیت کلاسی



۱. نوار ابزار، پنل لایه ها، پنل رنگ و پنل تاریخچه را پیدا کنید.
۲. ابزار **Brush** را انتخاب کرده و یک خط رسم کنید.
۳. رنگ قلم را تغییر دهید و دوباره رسم کنید.
۴. نتیجه کار را با هم کلاسی خود مقایسه کنید.

## خلاصه درس

- فتوشاپ نرم افزاری برای طراحی و ویرایش تصاویر است.
- نسخه های جدید CC امکانات بیشتری نسبت به نسخه های قدیمی دارند.
- محیط فتوشاپ شامل نوار ابزار، پنل لایه ها، پنل رنگ، پنل تاریخچه و پنل ویژگی هاست.
- هر بخش از محیط فتوشاپ وظیفه مشخصی دارد و آشنایی با آن ها باعث افزایش سرعت و دقت در طراحی می شود.

## تمرین

۱. سه کاربرد مهم نرم افزار فتوشاپ را بنویسید.

.....

۲. وظیفه پنل Layers چیست؟

.....

۳. پنل History چه کاربردی دارد؟

.....

۴. تصویری از محیط فتوشاپ تهیه کرده و نام پنج بخش اصلی آن را مشخص کنید.

.....

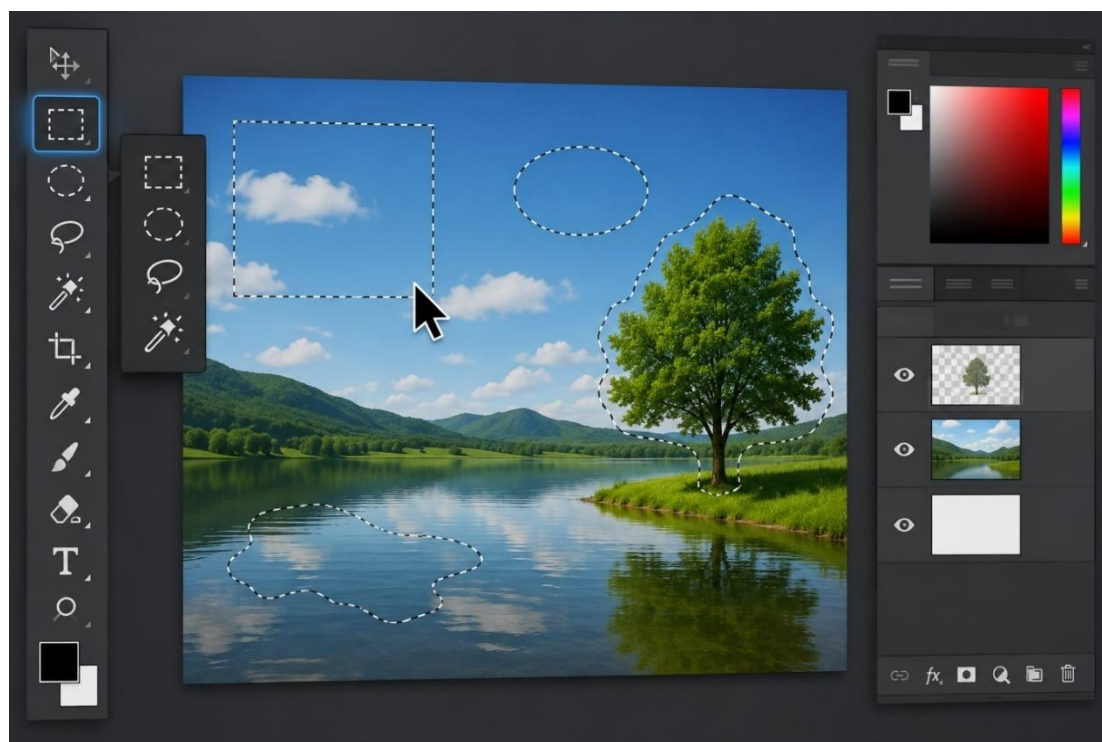
## پروژه

یک سند ایجاد کرده و با استفاده از ابزار Brush حداقل سه خط یا شکل ساده رسم کرده و نتیجه را در کلاس ارائه دهید.

## چالش

تحقیق کنید ابزارهای مبتنی بر هوش مصنوعی در نسخه های جدید فتوشاپ چه کارهایی انجام می دهند و دو نمونه از آن ها را معرفی کنید.

## درس ۵: آشنایی با ابزارهای انتخاب



در پایان این درس خواهید توانست:

- مفهوم (Selection) را در فتوشاپ توضیح دهید.
- ابزارهای مختلف انتخاب را بشناسید.
- ابزار مناسب را برای انتخاب بخش‌های مختلف تصویر انتخاب کنید.
- قسمتی از یک تصویر را انتخاب، ویرایش یا جابه‌جا کنید.



در بسیاری از ویرایش‌های تصویری لازم است تنها بخشی از تصویر را تغییر دهیم، نه کل آن. برای مثال ممکن است بخواهیم رنگ یک گل را تغییر دهیم، پس زمینه یک عکس را حذف کنیم یا شی را به تصویر دیگری منتقل کنیم. برای انجام این کارها ابتدا باید قسمت موردنظر را انتخاب (Select) کنیم.

فتوشاپ ابزارهای متنوعی برای انتخاب بخش‌های مختلف تصویر در اختیار کاربران قرار می‌دهد. هر ابزار برای نوع خاصی از انتخاب مناسب است و شناخت آن‌ها باعث می‌شود ویرایش تصاویر سریع‌تر و دقیق‌تر انجام شود. به پرسش‌های زیر پاسخ دهید.

۱. اگر بخواهید فقط یک گل را از داخل یک عکس جدا کنید، چه کاری انجام می‌دهید؟

۲. آیا تاکنون از ابزار انتخاب در نرم‌افزارهای ویرایش تصویر استفاده کرده‌اید؟

۳. به نظر شما چرا فتوشاپ چند نوع ابزار انتخاب مختلف دارد؟

## ابزار انتخاب چیست؟

ابزارهای انتخاب به ما کمک می‌کنند فقط بخش مشخصی از تصویر را انتخاب کنیم. پس از انتخاب، می‌توان آن قسمت را جابه‌جا، کپی، حذف، رنگ‌آمیزی یا ویرایش کرد، بدون اینکه سایر قسمت‌های تصویر تغییر کنند.

### ۱. ابزار Marquee Tool

کلید میانبر M: این ابزار برای ایجاد انتخاب‌های منظم به شکل مستطیل، مربع، دایره یا بیضی استفاده می‌شود.

انواع این ابزار:

• انتخاب مستطیلی یا مربعی: Rectangular Marquee Tool

Tool

• انتخاب دایره یا بیضی: Elliptical Marquee Tool

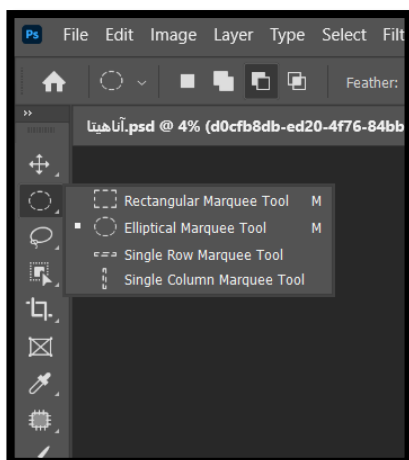
• انتخاب یک ردیف افقی به ضخامت یک پیکسل: Single Row

Row

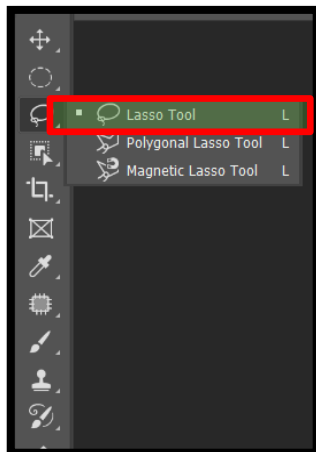
• انتخاب یک ستون عمودی به ضخامت یک پیکسل: Single Column

Single Column

کاربرد: زمانی که بخواهید بخشی از تصویر با شکل‌های ساده و منظم را انتخاب کنید، این ابزار بهترین گزینه است.



## ۲. ابزار Lasso Tool

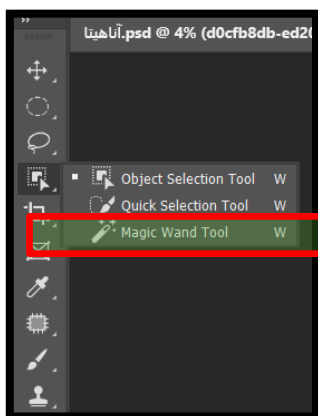


کلید میانبر L: این ابزار برای انتخاب نواحی با شکل های نامنظم استفاده می شود و مانند مدادی است که دور قسمت مورد نظر رسم می کند.

انواع این ابزار:

- انتخاب آزاد با حرکت دست: Lasso Tool
- انتخاب با رسم خطوط مستقیم و کلیک های متوالی: Polygonal Lasso Tool
- لبه های تصویر را تشخیص می دهد و به صورت خود کار به اطراف آن می چسبد: Magnetic Lasso Tool
- کاربرد: برای انتخاب اشیایی که شکل منظمی ندارند، مانند برگ، حیوان یا انسان، بسیار مناسب است.

## ۳. ابزار Magic Wand Tool

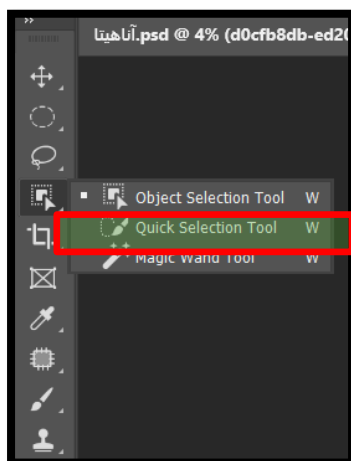


کلید میانبر W: این ابزار با یک کلیک، نواحی دارای رنگ های مشابه را انتخاب می کند.

یکی از مهم ترین تنظیمات این ابزار Tolerance است که میزان شباهت رنگ ها را مشخص می کند.

کاربرد: اگر پس زمینه تصویر یک رنگ باشد، مانند سفید یا آبی، می توان آن را به سرعت انتخاب کرد.

## ۴. ابزار Quick Selection Tool



کلید میانبر W: این ابزار مانند یک قلم مو عمل می کند. با کشیدن آن روی تصویر، فتوشاپ بخش های مشابه را به صورت هوشمند انتخاب می کند.

کاربرد: برای انتخاب سریع اشیایی مانند انسان، حیوان، لباس، خودرو یا سایر اجسام با مرز مشخص مناسب است.

## مقایسه ابزارهای انتخاب

| ابزار           | مناسب برای                                 |
|-----------------|--------------------------------------------|
| Marquee         | انتخاب های مستطیلی، مربعی، دایره ای و بیضی |
| Lasso           | انتخاب آزاد و اشکال نامنظم                 |
| Polygonal Lasso | انتخاب با خطوط مستقیم                      |
| Magnetic Lasso  | انتخاب خودکار لبه های تصویر                |
| Magic Wand      | انتخاب نواحی هم رنگ                        |
| Quick Selection | انتخاب سریع اشیای مشخص                     |

## نکات مهم

- پس از انتخاب بخشی از تصویر می توانید آن را جابه جا، کپی، حذف یا ویرایش کنید.
- برای دقیق تر کردن محدوده انتخاب از دستور Select and Mask استفاده کنید.
- با نگه داشتن کلید Shift می توان قسمت جدیدی را به محدوده انتخاب اضافه کرد.
- با نگه داشتن کلید Alt یا Option در سیستم عامل macOS می توان بخشی را از محدوده انتخاب حذف کرد.
- اگر نوار ابزار (Tools) نمایش داده نشد، می توان آن را از منوی Window → Tools دوباره فعال کرد.

## آیا می دانید؟

ابزار **Quick Selection** با استفاده از الگوریتم های هوشمند، مرز اشیای را تشخیص می دهد.

## فعالیت کلاسی

انتخاب و جابه جایی یک شی

۱. تصویر چن شی یا میوه را در فتوشاپ باز کنید.
۲. یکی از ابزارهای انتخاب را متناسب با شکل شی انتخاب کنید.
۳. یکی از میوه ها یا اشیای را انتخاب کنید.
۴. با استفاده از Move Tool آن را به محل دیگری منتقل کنید.
۵. نتیجه را ذخیره کرده و با هم کلاسی های خود مقایسه کنید.

## خلاصه درس

- ابزارهای انتخاب برای مشخص کردن بخشی از تصویر به کار می‌روند.
- ابزار Marquee برای انتخاب شکل‌های منظم مناسب است.
- ابزار Lasso برای انتخاب آزاد و اشکال نامنظم استفاده می‌شود.
- ابزار Magic Wand نواحی هم‌رنگ را انتخاب می‌کند.
- ابزار Quick Selection اشیا را به صورت هوشمند انتخاب می‌کند.
- با استفاده از کلیدهای Shift و Alt می‌توان محدوده انتخاب را مدیریت کرد.

## تمرین

۱. ابزارهای انتخاب در فتوشاپ چه کاربردی دارند؟

.....

۲. تفاوت ابزار Marquee Tool و Lasso Tool را بنویسید.

.....

۳. ابزار Magic Wand در چه شرایطی بهترین عملکرد را دارد؟

.....

۴. وظیفه ابزار Quick Selection Tool چیست؟

.....

۵. کاربرد کلیدهای Shift و Alt هنگام انتخاب را توضیح دهید.

.....

## پروژه

### طراحی یک تصویر ترکیبی

یک تصویر شامل چند شیء مختلف انتخاب کنید. سپس با استفاده از ابزارهای انتخاب، یکی از اشیا را جدا کرده و روی یک پس‌زمینه جدید قرار دهید. در پایان، تصویر نهایی را ذخیره و به کلاس ارائه کنید.

## چالش

یک تصویر با پس‌زمینه پیچیده انتخاب کنید و آن را با سه ابزار Lasso، Magic Wand و Quick Selection انتخاب کنید. سپس نتیجه هر ابزار را از نظر سرعت، دقت و کیفیت انتخاب با یکدیگر مقایسه کنید.

## درس ۶: آشنایی با ابزارهای ترسیم، متن و طراحی در فتوشاپ



در پایان این درس خواهید توانست:

- با ابزار Brush برای نقاشی و رنگ آمیزی آشنا شوید.
- از ابزار Eraser برای حذف بخش های تصویر استفاده کنید.
- متن را در تصویر ایجاد و ویرایش کنید.
- با ابزار Pen برای رسم مسیرها و شکل های دقیق آشنا شوید.
- از ابزارهای مناسب برای طراحی یک تصویر ساده استفاده کنید.



در طراحی و ویرایش تصاویر، تنها انتخاب بخش های مختلف تصویر کافی نیست. گاهی لازم است روی تصویر نقاشی کنیم، قسمت هایی را پاک کنیم، متن بنویسیم یا شکل های دقیق رسم کنیم. فتوشاپ ابزارهای متنوعی برای انجام این کارها در اختیار کاربران قرار می دهد.

در این درس با چهار ابزار مهم و پر کاربرد فتوشاپ یعنی Brush، Eraser، Text و Pen آشنا می شوید و یاد می گیرید چگونه از آنها در طراحی و ویرایش تصاویر استفاده کنید. به پرسش های زیر پاسخ دهید.

۱. اگر بخواهید روی یک عکس نقاشی کنید، از چه ابزاری استفاده می کنید؟

۲. برای نوشتن عنوان روی یک پوستر به چه ابزاری نیاز دارید؟

۳. اگر بخواهید دور یک تصویر را با دقت زیاد برش دهید، کدام ابزار مناسب تر است؟



## ۱. ابزار Brush یا قلم مو

کلید میانبر B : ابزار براش مانند یک قلم مو عمل می کند و برای نقاشی، رنگ آمیزی و ایجاد جلوه های مختلف روی تصویر به کار می رود.

مهم ترین تنظیمات براش:

- تعیین اندازه قلم مو: Size

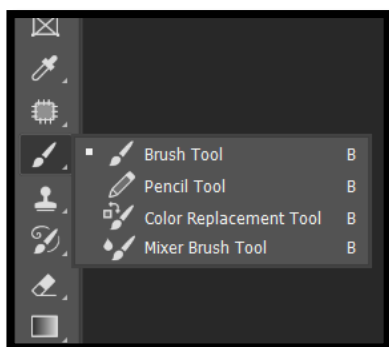
- تعیین میزان نرمی یا تیزی لبه های قلم: Hardness

- تعیین میزان شفافیت رنگ: Opacity

- کنترل سرعت و مقدار پخش شدن رنگ: Flow

با تغییر این تنظیمات می توان خطوط ظریف، ضخیم، نرم یا پررنگ ایجاد کرد.

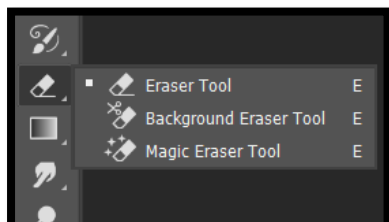
کاربردها: نقاشی دیجیتال، رنگ آمیزی تصاویر، ایجاد سایه و نور و طراحی جلوه های ویژه.



## ۲. ابزار Eraser یا پاک کن

کلید میانبر E : این ابزار برای پاک کردن قسمت های دلخواه تصویر یا لایه استفاده می شود و مانند Brush قابل تنظیم است.

انواع پاک کن:



- پاک کردن معمولی تصویر: Eraser Tool

- حذف پس زمینه های ساده و هم رنگ: Background Eraser Tool

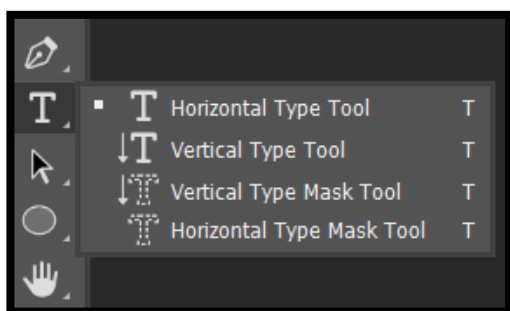
- حذف سریع قسمت های دارای رنگ مشابه با یک کلیک: Magic Eraser Tool

نکته: برای حذف کامل و شفاف شدن تصویر، بهتر است ابتدا لایه Background را به یک لایه معمولی تبدیل کنید.

### ۳. ابزار Text یا نوشتن متن

کلید میانبر T: با این ابزار می توان عنوان، توضیح، نام، شعار یا هر متن دیگری را روی تصویر نوشت. دو روش نوشتن متن:

- با یک کلیک ایجاد می شود و برای متن های کوتاه مناسب است: Point Text
- با رسم یک کادر ایجاد می شود و برای متن های چندخطی کاربرد دارد: Paragraph Text

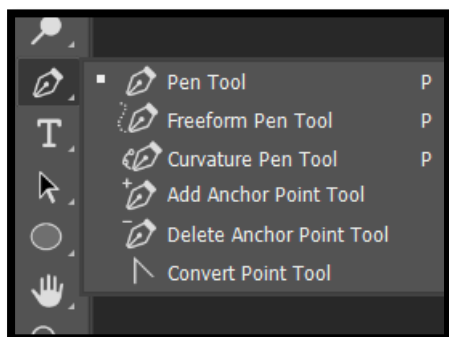


تنظیمات مهم متن:

- انتخاب نوع قلم (Font)
- اندازه نوشته (Size)
- رنگ متن (Color)
- حالت ضخیم یا مورب (Bold / Italic)
- چپ چین، راست چین یا وسط چین کردن متن
- هر متن در یک لایه جداگانه ذخیره می شود و می توان آن را بعداً ویرایش کرد.

### ۴. ابزار Pen یا قلم طراحی

کلید میانبر P: ابزار Pen برای رسم مسیرها، منحنی ها و شکل های دقیق استفاده می شود و یکی از دقیق ترین ابزارهای فتوشاپ است.



کاربردهای ابزار Pen:

- رسم شکل های دقیق
- دوربری حرفه ای تصاویر
- ایجاد مسیرهای صاف یا منحنی
- ساخت شکل های سفارشی
- پس از رسم مسیر، می توان:
- داخل آن را رنگ آمیزی کرد (Fill)
- برای آن خط دور رسم کرد (Stroke)

### تفاوت Stroke و Fill

| گزینه  | کاربرد             |
|--------|--------------------|
| Fill   | رنگ آمیزی داخل شکل |
| Stroke | رسم خط دور شکل     |

## نکات مهم

- اندازه، نرمی و شفافیت Brush را متناسب با نوع طراحی تنظیم کنید.
- برای نوشتن عنوان یا متن کوتاه از Point Text و برای متن های طولانی از Paragraph Text استفاده کنید.
- ابزار Pen برای انتخاب ها و طراحی های بسیار دقیق مناسب است.
- مسیرهای رسم شده با ابزار Pen را می توان رنگ آمیزی یا دور آن ها خط رسم کرد.

## فعالیت کلاسی



طراحی یک کارت تبریک:

- یک فایل جدید ایجاد کنید.
- با ابزار Brush یک پس زمینه ساده طراحی کنید.
- در صورت نیاز، بخش های اضافی را با Eraser پاک کنید.
- با ابزار Text یک پیام تبریک روی تصویر بنویسید.
- با ابزار Pen یک شکل ساده مانند قلب، ستاره یا گل رسم کنید.
- تصویر نهایی را ذخیره و به کلاس ارائه کنید.

## خلاصه درس



- ابزار Brush برای نقاشی و رنگ آمیزی استفاده می شود.
- ابزار Eraser برای حذف بخش های دلخواه تصویر کاربرد دارد.
- ابزار Text امکان نوشتن و ویرایش متن را فراهم می کند.
- ابزار Pen برای رسم مسیرها و شکل های دقیق استفاده می شود.
- با استفاده از Fill و Stroke می توان شکل های رسم شده را رنگ آمیزی یا دور آن ها خط رسم کرد.

## تمرین



۱. تفاوت ابزار Eraser و Magic Eraser را توضیح دهید.

.....

۲. ابزار Pen در چه مواردی استفاده می شود؟

.....

۳. تفاوت Fill و Stroke را بیان کنید.

.....

## پروژه

طراحی یک پوستر ساده

یک پوستر با موضوع طراحی کنید. در طراحی خود از موارد زیر استفاده کنید:

ابزار Brush برای ایجاد پس زمینه

ابزار Text برای نوشتن عنوان

ابزار Pen برای رسم یک شکل ساده

ابزار Eraser برای اصلاح بخش های اضافی

پس از تکمیل، فایل را ذخیره کرده و به کلاس ارائه دهید.

## چالش

چالش ۱: با استفاده از ابزار Pen یک لوگوی ساده طراحی کنید و با گزینه های Fill و Stroke آن را رنگ آمیزی نمایید.

چالش ۲: یک کارت تبریک دیجیتال طراحی کنید که در آن از هر چهار ابزار Brush، Eraser، Text و Pen استفاده شده باشد. سپس توضیح دهید هر ابزار چه نقشی در طراحی شما داشته است.

## درس ۷: آشنایی با لایه‌ها در فتوشاپ



در پایان این درس خواهید توانست:

- مفهوم لایه را توضیح دهید.
- اهمیت استفاده از لایه‌ها در طراحی را بیان کنید.
- انواع اصلی لایه‌ها را بشناسید.
- ترتیب لایه‌ها را تغییر دهید.
- از گزینه‌های شفافیت، قفل و مخفی کردن لایه‌ها استفاده کنید.

هنگام طراحی یک تصویر، معمولاً اجزای مختلفی مانند پس زمینه، عکس، متن و شکل های گرافیکی در کنار هم قرار می گیرند. اگر همه این اجزا روی یک تصویر واحد قرار داشته باشند، ویرایش آن ها دشوار خواهد بود. فتوشاپ برای حل این مشکل از لایه ها (Layers) استفاده می کند. هر بخش از تصویر روی یک لایه جداگانه قرار می گیرد و می توان آن را بدون تأثیر بر سایر بخش ها ویرایش کرد. به همین دلیل، لایه ها یکی از مهم ترین امکانات فتوشاپ هستند.

به پرسش های زیر پاسخ دهید.

اگر بخواهید فقط نوشته یک پوستر را تغییر دهید، آیا بهتر است همه اجزای پوستر در یک لایه باشند یا در لایه های جداگانه؟

به نظر شما چرا طراحان حرفه ای از لایه ها استفاده می کنند؟

اگر بخواهید تصویری را بدون تغییر متن آن ویرایش کنید، چه راهکاری پیشنهاد می دهید؟

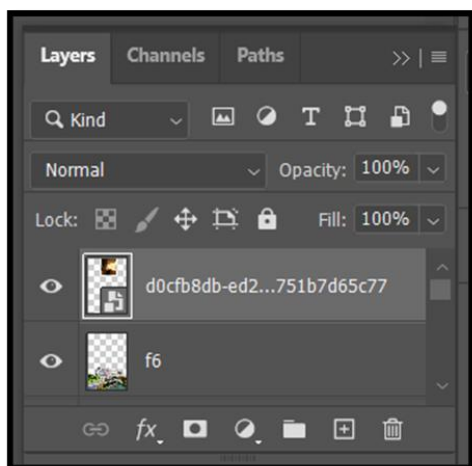
### لایه چیست؟

لایه را می توان به ورق های شفاف تشبیه کرد که روی هم قرار گرفته اند. روی هر ورق بخشی از طراحی قرار دارد و در کنار هم تصویر نهایی را تشکیل می دهند. برای مثال، در یک پوستر ممکن است: پس زمینه در یک لایه باشد. تصویر اصلی در لایه ای دیگر قرار گیرد. عنوان پوستر در یک لایه جداگانه نوشته شود. در این حالت، هر بخش را می توان بدون تغییر سایر قسمت ها ویرایش یا جابه جا کرد.

### چرا استفاده از لایه ها مهم است؟

استفاده از لایه ها مزایای زیادی دارد، از جمله:

- ویرایش مستقل هر بخش از تصویر
- جابه جایی آسان تصاویر، متن ها و شکل ها
- افزودن افکت ها بدون تغییر سایر قسمت ها
- تنظیم رنگ یا نور هر بخش به صورت جداگانه
- طراحی منظم و قابل ویرایش



### انواع مهم لایه ها

۱. لایه تصویری (Pixel Layer): این لایه شامل عکس ها یا نقاشی هایی است که از پیکسل تشکیل شده اند و بیشتر برای تصاویر دیجیتال استفاده می شود.

۲. لایه متنی (Text Layer): هر متنی که با ابزار Text ایجاد می شود، در یک لایه جداگانه قرار می گیرد و تا زمانی که به تصویر تبدیل نشده باشد، قابل ویرایش است.

۳. لایه شکلی (Shape Layer): این لایه برای رسم اشکالی مانند مستطیل، دایره، خط یا چندضلعی استفاده می شود. شکل های این لایه کیفیت خود را هنگام بزرگ یا کوچک شدن حفظ می کنند.

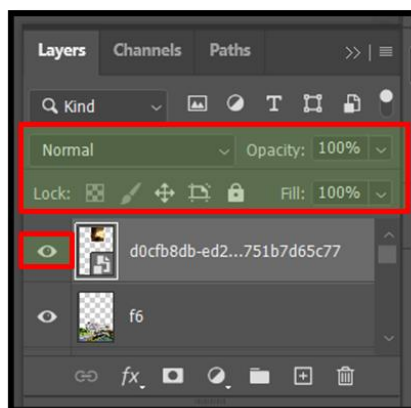
۴. لایه تنظیمات (Adjustment Layer): از این لایه برای تغییر نور، روشنایی، کنتراست و رنگ تصویر استفاده می شود، بدون آنکه تصویر اصلی تغییر دائمی کند. به همین دلیل، این روش ویرایش غیرمخرب نام دارد.

### تنظیمات مهم لایه ها

شفافیت (Opacity): با استفاده از گزینه Opacity می توان میزان دیده شدن یک لایه را تنظیم کرد.

۱۰۰٪ یعنی لایه کاملاً دیده می شود.

۰٪ یعنی لایه کاملاً نامرئی است.



مخفی کردن لایه: در کنار هر لایه، نماد چشم قرار دارد. با کلیک روی آن می توان لایه را موقتاً مخفی یا دوباره نمایش داد.

ترتیب لایه ها: لایه ای که در بالاترین قسمت پنل Layers قرار دارد، روی سایر لایه ها نمایش داده می شود. برای تغییر ترتیب لایه ها:

- لایه را با ماوس به بالا یا پایین بکشید.
- یا از میانبرهای صفحه کلید استفاده کنید.

قفل کردن لایه: برای جلوگیری از تغییرات ناخواسته، می توان لایه را قفل کرد.

برخی از انواع قفل عبارت اند از:

Lock All: قفل کامل لایه

Lock Position: جلوگیری از جابه جایی لایه

Lock Image Pixels: جلوگیری از تغییر محتوای تصویر

Lock Transparent Pixels: جلوگیری از تغییر بخش های شفاف لایه

### آیا می‌دانید؟



- بسیاری از فایل‌های طراحی حرفه‌ای ممکن است ده‌ها یا حتی صدها لایه داشته باشند.
- استفاده صحیح از لایه‌ها باعث می‌شود بتوانید هر زمان طراحی خود را بدون نیاز به شروع دوباره ویرایش کنید.
- لایه‌های تنظیمات یکی از بهترین روش‌ها برای اصلاح رنگ و نور تصاویر هستند، زیرا تصویر اصلی را به صورت دائمی تغییر نمی‌دهند.

### فعالیت کلاسی



- یک فایل جدید ایجاد کنید.
- یک تصویر را به عنوان پس‌زمینه وارد کنید.
- یک عنوان با ابزار Text اضافه کنید.
- یک شکل ساده رسم کنید.
- ترتیب لایه‌ها را تغییر دهید و نتیجه را مشاهده کنید.
- یکی از لایه‌ها را مخفی و سپس دوباره نمایش دهید.
- شفافیت یکی از لایه‌ها را تغییر دهید و تأثیر آن را بررسی کنید.

### خلاصه درس



- لایه‌ها امکان ویرایش مستقل بخش‌های مختلف تصویر را فراهم می‌کنند.
- مهم‌ترین انواع لایه‌ها شامل لایه تصویری، متنی، شکلی و تنظیمات هستند.
- با استفاده از Opacity می‌توان میزان شفافیت لایه را تنظیم کرد.
- ترتیب قرارگیری لایه‌ها بر نمایش اجزای تصویر تأثیر می‌گذارد.
- قفل کردن و مخفی کردن لایه‌ها به مدیریت بهتر پروژه کمک می‌کند.

### تمرین



۱. لایه را به زبان خود تعریف کنید.

.....

۲. دو مزیت استفاده از لایه‌ها را بنویسید.

.....

۳. چهار نوع مهم لایه در فتوشاپ را نام ببرید.

.....

۴. گزینه Opacity چه کاربردی دارد؟



### پروژه

طراحی یک پوستر آموزشی با لایه‌ها:

یک پوستر با موضوع دلخواه از کتاب‌های درسی خود طراحی کنید.

در طراحی:

- یک لایه برای پس زمینه
  - یک لایه برای تصویر
  - یک لایه برای عنوان
  - یک لایه برای شکل‌های گرافیکی
  - و در صورت نیاز یک لایه تنظیمات برای اصلاح رنگ یا نور
- ایجاد کنید و در پایان، فایل را ذخیره و به کلاس ارائه دهید.



### چالش

یک تصویر را در چند لایه مختلف طراحی کنید و با تغییر ترتیب لایه‌ها بررسی کنید که نتیجه نهایی چگونه تغییر می‌کند.

## درس ۸: مدیریت لایه‌ها با ماسک، شفافیت و گروه‌بندی



در پایان این درس خواهید توانست:

- مفهوم ماسک لایه (Layer Mask) را توضیح دهید.
- از ماسک برای مخفی کردن بخشی از تصویر بدون حذف دائمی آن استفاده کنید.
- شفافیت (Opacity) و حالت‌های ترکیب (Blending Modes) را تنظیم کنید.
- لایه‌ها را گروه‌بندی و مدیریت کنید.
- پروژه‌های خود را منظم‌تر و قابل ویرایش طراحی کنید.

در طراحی تصاویر، گاهی لازم است بخشی از یک تصویر را موقتاً مخفی کنیم، میزان شفافیت یک لایه را تغییر دهیم یا چند لایه مرتبط را در یک گروه قرار دهیم. فتوشاپ ابزارهایی در اختیار ما قرار می‌دهد که این کارها را بدون حذف دائمی اطلاعات انجام می‌دهند. این ویژگی‌ها باعث می‌شوند ویرایش تصاویر ساده‌تر، دقیق‌تر و حرفه‌ای‌تر باشد. به پرسش‌های زیر پاسخ دهید.

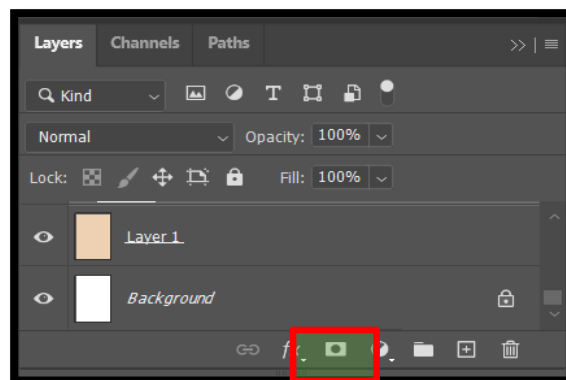
۱. اگر بخواهید بخشی از یک تصویر را موقتاً پنهان کنید، آیا بهتر است آن را پاک کنید یا فقط مخفی کنید؟
۲. چرا گاهی لازم است یک تصویر نیمه‌شفاف باشد؟
۳. اگر یک پروژه شامل ده‌ها لایه باشد، چگونه می‌توان آن را منظم نگه داشت؟

## ماسک لایه (Layer Mask)



ماسک لایه ابزاری است که به ما اجازه می‌دهد بخشی از یک لایه را بدون حذف دائمی مخفی یا دوباره نمایان کنیم. برای ایجاد ماسک:

۱. لایه موردنظر را انتخاب کنید.
۲. در پایین پنل Layers روی دکمه Add Layer Mask کلیک کنید.
۳. یک تصویر کوچک (Thumbnail) ماسک در کنار لایه ظاهر می‌شود.



## رنگ‌ها در ماسک چه معنایی دارند؟

| رنگ            | نتیجه                                            |
|----------------|--------------------------------------------------|
| سفید (White)   | بخش موردنظر دیده می‌شود.                         |
| مشکی (Black)   | بخش موردنظر مخفی می‌شود.                         |
| خاکستری (Gray) | بخش موردنظر به صورت نیمه‌شفاف نمایش داده می‌شود. |

با استفاده از ابزار Brush می‌توان روی ماسک نقاشی کرد و بخش‌های مختلف تصویر را پنهان یا آشکار نمود. اگر اشتباهی رخ دهد، کافی است رنگ قلم را از مشکی به سفید تغییر دهید تا قسمت مخفی شده دوباره نمایش داده شود. این روش، یک ویرایش غیرمخرب است؛ زیرا تصویر اصلی حذف نمی‌شود.

## کاربردهای ماسک

از ماسک می توان در کارهای مختلف استفاده کرد، مانند:

- حذف حرفه ای پس زمینه تصویر
- ترکیب دو یا چند تصویر
- محو کردن تدریجی بخشی از تصویر
- نمایش بخشی از یک افکت یا تصویر

## شفافیت (Opacity)

گزینه Opacity میزان شفافیت یک لایه را مشخص می کند.

- ۱۰۰٪: لایه کاملاً دیده می شود.
- ۰٪: لایه کاملاً نامرئی است.

برای مثال، اگر بخواهید نوشته ای کم رنگ تر دیده شود، کافی است مقدار Opacity آن را کاهش دهید.

## حالت های ترکیب (Blending Modes)

حالت های ترکیب مشخص می کنند که رنگ ها و نور یک لایه چگونه با لایه های زیرین ترکیب شوند.

برخی از حالت های پر کاربرد عبارت اند از:

| حالت     | کاربرد                                     |
|----------|--------------------------------------------|
| Normal   | نمایش عادی لایه                            |
| Multiply | تیره تر کردن تصویر (مناسب برای ایجاد سایه) |
| Screen   | روشن تر کردن تصویر (مناسب برای ایجاد نور)  |
| Overlay  | افزایش کنتراست و ترکیب نور و سایه          |
| Color    | تغییر رنگ بدون تغییر جزئیات تصویر          |

با ترکیب Opacity و Blending Modes می توان جلوه های متنوع و خلاقانه ایجاد کرد.

## گروه بندی لایه ها (Layer Groups)

در پروژه های بزرگ، تعداد لایه ها ممکن است زیاد شود. برای مرتب نگه داشتن پروژه می توان چند لایه مرتبط را در یک گروه (Group) قرار داد.

برای ایجاد گروه:

- چند لایه را انتخاب کرده و کلیدهای Ctrl + G را فشار دهید.
- یا از گزینه Create New Group در پنل Layers استفاده کنید.

## مزایای گروه بندی

- مرتب شدن پروژه
- جابه جایی هم زمان چند لایه
- مخفی یا نمایش دادن همه لایه های گروه
- اعمال تنظیمات یا افکت روی کل گروه
- مدیریت آسان پروژه های بزرگ

## نکات مهم

- از ماسک لایه به جای پاک کردن دائمی بخش های تصویر استفاده کنید.
- برای ایجاد لبه های نرم، از قلم موی نرم (Soft Brush) هنگام کار با ماسک استفاده کنید.
- کاهش مقدار Opacity باعث می شود لایه شفاف تر دیده شود.
- حالت Multiply برای ایجاد سایه و حالت Screen برای ایجاد نور مناسب است.
- لایه های مرتبط را در یک گروه قرار دهید تا پروژه منظم تر و مدیریت آن آسان تر شود.

آیا می دانید؟ 

- طراحان حرفه ای معمولاً به جای حذف قسمت های تصویر، از Layer Mask استفاده می کنند تا در صورت نیاز بتوانند تغییرات را به راحتی اصلاح کنند.
- بسیاری از جلوه های نوری و تصویری در پوسترها و تبلیغات با استفاده از Blending Modes ایجاد می شوند.

فعالیت کلاسی 

طراحی یک پوستر با لایه های گروه بندی شده:

۱. یک سند جدید ایجاد کنید.
۲. یک تصویر را به عنوان پس زمینه وارد کنید.
۳. یک عنوان و یک شکل گرافیکی به تصویر اضافه کنید..
۴. لایه های متن را در یک گروه و لایه های تصاویر را در گروهی دیگر قرار دهید.
۵. شفافیت یکی از لایه ها را تغییر دهید.
۶. با استفاده از ماسک، بخشی از تصویر را به صورت تدریجی مخفی کنید.
۷. نتیجه را ذخیره کرده و با هم کلاسی های خود مقایسه کنید.

**نکته:** یکی از اصول طراحی حرفه‌ای این است که هیچ گاه اطلاعات اصلی تصویر را حذف نکنید. استفاده از ماسک، تنظیم شفافیت و گروه‌بندی لایه‌ها به شما کمک می‌کند تا هر زمان که لازم باشد، بتوانید طرح خود را بدون از دست دادن اطلاعات ویرایش کنید.

### خلاصه درس



- Layer Mask بخشی از تصویر را بدون حذف دائمی مخفی می‌کند.
- رنگ سفید، مشکی و خاکستری در ماسک، میزان نمایش لایه را تعیین می‌کنند.
- Opacity میزان شفافیت لایه را تنظیم می‌کند.
- Blending Modes نحوه ترکیب لایه‌ها را مشخص می‌کنند.
- گروه‌بندی لایه‌ها باعث نظم بیشتر و مدیریت آسان‌تر پروژه می‌شود.

### تمرین



۱. ماسک لایه چیست و چه تفاوتی با پاک کردن تصویر دارد؟

.....

۲. در ماسک، رنگ‌های سفید، مشکی و خاکستری چه کاربردی دارند؟

.....

۳. دو حالت پر کاربرد **Blending Modes** را نام برده و کاربرد هر کدام را بنویسید.

.....

### پروژه



طراحی یک پوستر تبلیغاتی

یک پوستر با موضوع دلخواه طراحی کنید و در آن:

- از Layer Mask برای ترکیب دو تصویر استفاده کنید.
  - شفافیت یکی از لایه‌ها را تغییر دهید.
  - حداقل یک حالت ترکیب (Blending Mode) را به کار ببرید.
  - لایه‌های مرتبط را در گروه‌های جداگانه سازمان‌دهی کنید.
- در پایان، فایل پروژه را ذخیره کرده و مراحل انجام کار را برای کلاس توضیح دهید.

### چالش



یک تصویر را با سه حالت مختلف Multiply، Screen و Overlay آزمایش کنید و تفاوت نتیجه هر حالت را در یک جدول مقایسه نمایید.

## درس ۹: تنظیم نور، رنگ و روتوش تصاویر در فتوشاپ



در پایان این درس خواهید توانست:

- نور و رنگ تصاویر را به صورت ساده تنظیم کنید.
- با ابزارهای Dodge، Burn و Sponge آشنا شوید.
- کاربرد فیلترهای هوشمند در ویرایش تصاویر را توضیح دهید.
- از ابزار Camera Raw Filter برای بهبود کیفیت عکس استفاده کنید.
- اهمیت ویرایش طبیعی و حفظ کیفیت تصویر را درک کنید.

گاهی یک تصویر از نظر نور، رنگ یا کیفیت مناسب نیست. ممکن است عکس بیش از حد تیره، روشن یا کم رنگ باشد. فتوشاپ ابزارها و فیلترهای متنوعی در اختیار کاربران قرار می دهد تا بدون نیاز به گرفتن دوباره عکس، کیفیت آن را بهبود دهند.

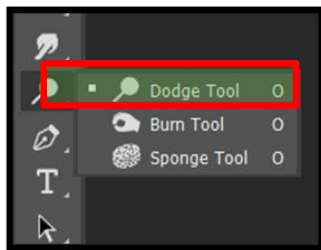
در این درس با چند ابزار مهم برای روشن کردن، تیره کردن، افزایش رنگ و اصلاح تصاویر آشنا می شوید. همچنین با برخی فیلترهای هوشمند فتوشاپ که برای حذف اشیای اضافی یا بهبود کیفیت عکس استفاده می شوند، آشنا خواهید شد. به پرسش های زیر پاسخ دهید.

۱. آیا تاکنون عکسی گرفته اید که خیلی تیره یا خیلی روشن باشد؟
۲. اگر بخواهید رنگ گل یا لباس در یک عکس زنده تر دیده شود، چه کاری انجام می دهید؟
۳. به نظر شما حذف یک شیء اضافی از عکس چگونه امکان پذیر است؟



### ۱. ابزار Dodge

Dodge Tool برای روشن تر کردن بخش های دلخواه تصویر استفاده می شود.



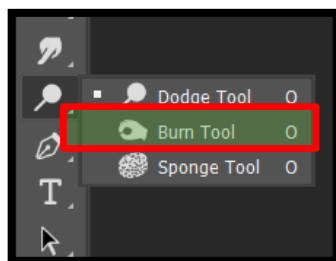
کاربردهای این ابزار:

- روشن تر کردن چهره
- افزایش نور در بخش های خاص تصویر
- برجسته کردن جزئیات عکس

برای طبیعی تر شدن نتیجه، بهتر است مقدار Exposure را کم انتخاب کنید و تغییرات را به آرامی انجام دهید.

### ۲. ابزار Burn

Burn Tool برعکس Dodge عمل می کند و برای تیره تر کردن قسمت های مختلف تصویر به کار می رود.



کاربردهای این ابزار:

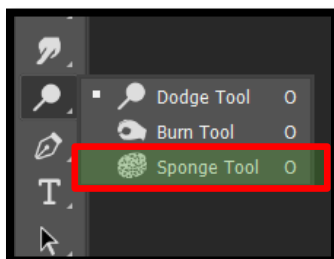
- ایجاد سایه
- افزایش عمق تصویر
- برجسته تر نشان دادن فرم چهره یا اشیا
- تیره کردن گوشه های تصویر برای جلب توجه به مرکز عکس

### ۳. ابزار Sponge

Sponge Tool میزان اشباع رنگ (Saturation) را تغییر می دهد.

با این ابزار می توان:

- رنگ ها را زنده تر و شاداب تر کرد (Saturate)
- شدت رنگ ها را کاهش داد (Desaturate)
- بخشی از تصویر را کم رنگ یا حتی نزدیک به سیاه و سفید نمایش داد.



## ۴. فیلترهای هوشمند

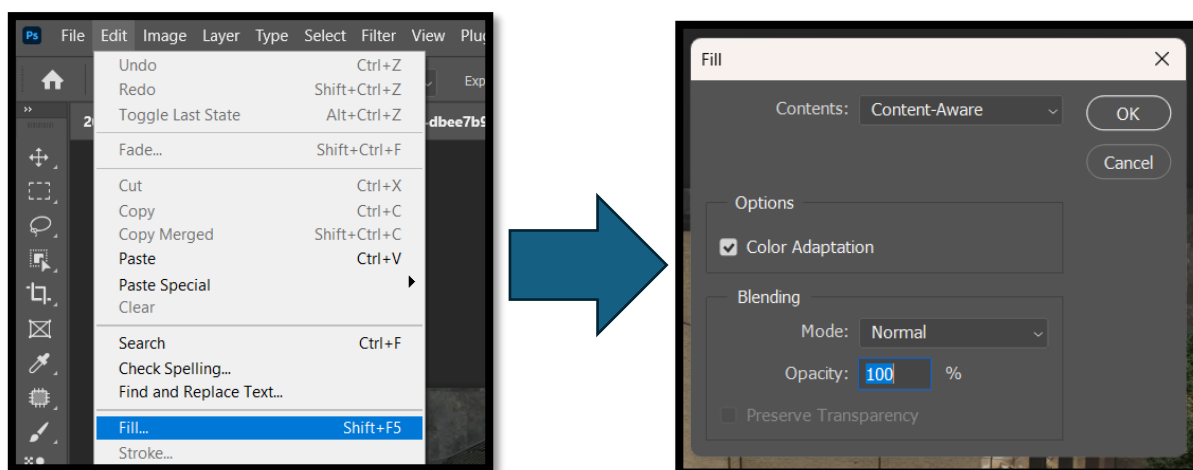
فتوشاپ ابزارهای هوشمندی دارد که بسیاری از ویرایش‌ها را به صورت خودکار انجام می‌دهند.

## ➤ Content-Aware Fill

اگر بخشی از تصویر را انتخاب کنید، فتوشاپ می‌تواند آن قسمت را با استفاده از بافت و رنگ اطراف به صورت هوشمند پر کند.

کاربردها:

- حذف افراد یا اشیای اضافی
- پاک کردن لکه‌ها
- حذف سیم، تابلو یا اجسام مزاحم از عکس



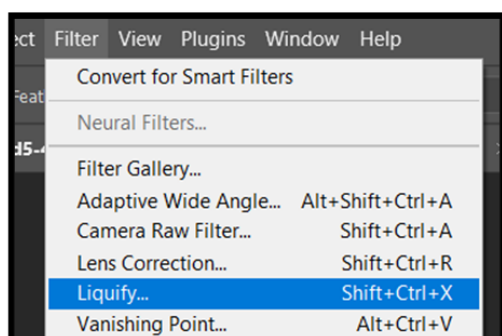
## ➤ Liquify

فیلتر Liquify برای تغییر شکل بخش‌های مختلف تصویر استفاده می‌شود.

از این فیلتر می‌توان برای:

- تغییر فرم صورت
- اصلاح خطوط لباس
- ایجاد جلوه‌های هنری
- تغییر اندازه برخی اجزای تصویر

استفاده کرد.

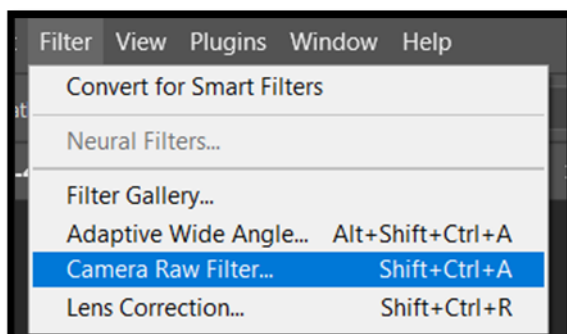


**نکته:** این ابزار باید با دقت و به اندازه استفاده شود تا تصویر طبیعی باقی بماند.

## ➤ Camera Raw Filter

یکی از پرکاربردترین ابزارهای فتوشاپ برای بهبود کیفیت تصاویر است.

با استفاده از آن می توان:



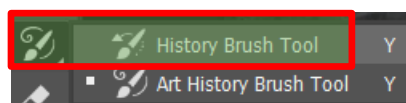
- روشنایی تصویر را تنظیم کرد.
- کنتراست را افزایش یا کاهش داد.
- رنگ ها را اصلاح کرد.
- وضوح تصویر را بیشتر کرد.
- نویز عکس را کاهش داد.

## ابزارهای کمکی:

## ➤ History Brush

این ابزار بخشی از تصویر را به یکی از حالت های قبلی بازمی گرداند.

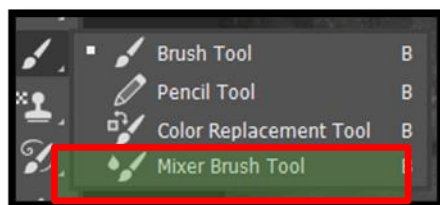
اگر بخشی از تصویر را اشتباه ویرایش کرده باشید، می توانید فقط همان قسمت را اصلاح کنید، بدون اینکه سایر تغییرات از بین برود.



## ➤ Mixer Brush

این ابزار رنگ ها را مانند قلم موهای واقعی با یکدیگر ترکیب می کند.

کاربردهای آن:



- نقاشی دیجیتال
- ایجاد افکت آبرنگ یا رنگ روغن
- ترکیب نرم رنگ ها
- طراحی هنری

## نکات مهم:

- برای طبیعی ماندن تصویر، ابزارهای Dodge و Burn را با شدت کم استفاده کنید.
- قبل از ویرایش، از لایه اصلی یک نسخه کپی بگیرید تا تصویر اولیه حفظ شود.
- برای حذف اشیای اضافی، Content-Aware Fill یکی از سریع ترین ابزارهاست.
- تنظیم نور و رنگ با Camera Raw Filter معمولاً نتیجه ای طبیعی تر از تغییر مستقیم تصویر ایجاد می کند.
- ویرایش خوب، ویرایشی است که طبیعی به نظر برسد و باعث از بین رفتن جزئیات تصویر نشود.



## فعالیت کلاسی



بهبود کیفیت یک تصویر:

یک تصویر را در فتوشاپ باز کنید.

از لایه اصلی یک نسخه کپی بگیرید.

با Camera Raw Filter روشنایی و رنگ تصویر را اصلاح کنید.

با ابزار Sponge رنگ یکی از بخش‌های تصویر را افزایش دهید.

با ابزار Burn گوشه‌های تصویر را کمی تیره کنید.

نتیجه را ذخیره کرده و با تصویر اولیه مقایسه کنید.

## دستیار هوشمند



### پنجره‌ای به دنیای فتوشاپ هوشمند (AI-Powered Photoshop)

تا اینجا با ابزارهای کلاسیک انتخاب و لایه‌بندی در فتوشاپ آشنا شدید. اما آیا می‌دانستید امروز هوش مصنوعی به «دستیارِ طراح» تبدیل شده است؟ ابزارهای جدید فتوشاپ (مانند Generative Fill) به شما اجازه می‌دهند با نوشتن چند کلمه، بخش‌های جدیدی به تصویر اضافه کنید یا اشیاء ناخواسته را در یک چشم‌به‌هم‌زدن حذف کنید.

**نکته طلایی:** هوش مصنوعی جایگزین «خلاقیت» شما نیست، بلکه سرعت رسیدن به ایده‌ها را چند برابر می‌کند. یادتان باشد که یک طراح حرفه‌ای، همیشه بر نتیجه نهایی نظارت می‌کند تا نتیجه‌ای با کیفیت و اخلاقی خلق شود.

## فعالیت کلاسی



«تصویر سازی برای آینده»

**هدف:** ادغام خلاقیت فردی با هوش مصنوعی برای ویرایش تصاویر.

۱. یک عکس ساده از منظره‌ای در شهر یا طبیعت انتخاب کنید.
۲. تصور کنید قرار است این مکان را به یک «شهر هوشمند در آینده» تبدیل کنید.
۳. با استفاده از ابزار هوش مصنوعی (یا ابزارهای ترسیم و ترکیب لایه‌ها)، سه المان جدید به تصویر اضافه کنید:
  - یک وسیله حمل و نقل مدرن (مثل ماشین پرنده یا ربات).
  - یک منبع انرژی سبز (مثل پنل خورشیدی یا توربین بادی کوچک).
  - یک فضای سبز یا باغچه عمودی روی دیوار ساختمان‌ها.

پرسش چالش‌برانگیز:

اگر از هوش مصنوعی بخواهیم تصویر یک شخص واقعی را تغییر دهد، چه مسائل اخلاقی ممکن است پیش بیاید؟ (در کلاس با هم گروهی‌هایتان بحث کنید که آیا دستکاری تصاویر افراد بدون اجازه آن‌ها کار درستی است؟)

خروجی شما:

دو تصویر تحویل دهید: ۱. تصویر اصلی (قبل از ویرایش) ۲. تصویر ویرایش شده (بعد از ترکیب با ایده‌های خلاقانه شما).

## خلاصه درس

- ابزار Dodge برای روشن تر کردن بخش های تصویر استفاده می شود.
- ابزار Burn قسمت های دلخواه را تیره تر می کند.
- ابزار Sponge شدت رنگ ها را افزایش یا کاهش می دهد.
- Content-Aware Fill برای حذف هوشمند اشیای اضافی کاربرد دارد.
- Camera Raw Filter یکی از بهترین ابزارها برای تنظیم نور، رنگ و کیفیت تصاویر است.
- استفاده متعادل از ابزارهای ویرایش، نتیجه ای طبیعی تر ایجاد می کند.

## تمرین

۱. تفاوت ابزار Burn و Dodge را توضیح دهید.

.....

۲. ابزار Sponge چه تغییری در تصویر ایجاد می کند؟

.....

۳. کاربرد Content-Aware Fill چیست؟

.....

۴. چرا بهتر است قبل از ویرایش تصویر، از لایه اصلی نسخه کپی تهیه کنیم؟

.....

## پروژه

بازسازی یک عکس

یک عکس قدیمی یا تصویری با نور نامناسب انتخاب کنید و مراحل زیر را انجام دهید:

- تنظیم نور و رنگ با Camera Raw Filter
- حذف یک شیء اضافی با Content-Aware Fill
- اصلاح نور با Dodge و Burn
- افزایش رنگ بخشی از تصویر با Sponge
- در پایان، تصویر اولیه و تصویر ویرایش شده را با هم مقایسه کرده و تغییرات انجام شده را برای کلاس توضیح دهید.

## چالش

تحقیق کنید ابزارهای مبتنی بر هوش مصنوعی در نسخه های جدید فتوشاپ چگونه به حذف اشیاء، بهبود کیفیت تصاویر و روتوش عکس ها کمک می کنند.

## درس ۱۰: بازگشت به دنیای پایتون و آمادگی برای بینایی ماشین



در پایان این درس خواهید توانست:

- ارتباط مفاهیم پایه برنامه نویسی با پروژه های هوش مصنوعی و تشخیص چهره را بیان کنید.
- از متغیرها برای ذخیره اطلاعات مورد نیاز برنامه استفاده کنید.
- اطلاعات مورد نیاز را با استفاده از تابع `input()` از کاربر دریافت کنید.
- با استفاده از دستورهای شرطی، پاسخ کاربر را بررسی کرده و تصمیم مناسب بگیرید.
- از حلقه ها برای تکرار دستورها و شبیه سازی جست و جوی پیوسته استفاده کنید.
- نتیجه بررسی و تصمیم برنامه را با استفاده از دستور `print()` نمایش دهید.
- مراحل اصلی عملکرد یک سیستم ساده تشخیص چهره را توضیح دهید.



وقتی نام «هوش مصنوعی» یا «بینایی ماشین» را می شنویم، ممکن است تصور کنیم با موضوعی بسیار پیچیده روبه رو هستیم. اما بیشتر پروژه های پیشرفته، از مفاهیم ساده و پایه آغاز می شوند.

در این دوره، قصد داریم یک سیستم ساده تشخیص چهره طراحی کنیم. شاید این پروژه در نگاه اول پیچیده به نظر برسد، اما بسیاری از بخش های آن بر پایه مفاهیمی ساخته می شوند که پیش تر در برنامه نویسی یاد گرفته ایم؛ مفاهیمی مانند دریافت ورودی، استفاده از شرط ها، تکرار دستورها و نمایش خروجی.

برای مثال، اگر رایانه بخواهد با استفاده از دوربین یک چهره را شناسایی کند، باید تصویرها را به صورت پیوسته بررسی کند. این تکرار، با استفاده از حلقه ها انجام می شود.

همچنین، برنامه باید تشخیص دهد که آیا در تصویر چهره ای وجود دارد یا خیر. برای انجام این کار، از دستورهای شرطی استفاده می شود. بنابراین، پیش از ورود به دنیای پردازش تصویر و تشخیص چهره، ابتدا مفاهیم اصلی برنامه نویسی را مرور می کنیم.

## برنامه نویسی چیست؟

✓ برنامه نویسی یعنی نوشتن دستور برای رایانه

رایانه مانند یک دانش آموز بسیار دقیق عمل می کند. هر کاری را که به او دستور دهیم انجام می دهد؛ اما دستورها باید روشن، درست و مرحله به مرحله نوشته شوند. برای مثال، می توانیم به رایانه دستور دهیم:

- یک پیام روی صفحه نمایش دهد.
- از کاربر اطلاعات دریافت کند.
- پاسخ کاربر را بررسی کند.
- اگر پاسخ درست بود، پیام مناسب نمایش دهد.
- اگر پاسخ نادرست بود، سؤال را دوباره تکرار کند.

رایانه می تواند همه این کارها را انجام دهد؛ اما برای اجرای دستورها باید از یک زبان برنامه نویسی استفاده کنیم. یکی از زبان های مناسب برای شروع برنامه نویسی، پایتون (Python) است. پایتون زبانی ساده، خوانا و قدرتمند است و در بسیاری از پروژه های واقعی هوش مصنوعی و بینایی ماشین نیز کاربرد دارد.

## مرور چند مفهوم پایه

| مثال                                                                                                 | توضیحات                                  | دستور                                                                                              |
|------------------------------------------------------------------------------------------------------|------------------------------------------|----------------------------------------------------------------------------------------------------|
| <pre>print("سلام") print("به کلاس هوش مصنوعی خوش آمدید")</pre>                                       | نمایش متن یا عدد روی صفحه                | <b>print</b>  ۱ |
| <pre>name = "علی" age = 14 print(name) print(age)</pre>                                              | ذخیره اطلاعات در حافظه برای استفاده مجدد | <b>متغیر</b>  ۲ |
| <pre>name = input("نام خود را وارد کنید:") print("سلام", name)</pre>                                 | دریافت اطلاعات از کاربر                  | <b>input</b>  ۳ |
| <pre>score = 18 if score &gt;= 10:     print("قبول شدی") else:     print("نیاز به تلاش بیشتر")</pre> | تصمیم گیری بر اساس شرط                   | <b>if</b>  ۴    |
| <pre>count = 1 while count &lt;= 3:     print("در حال بررسی...")     count = count + 1</pre>         | اجرای تکراری دستورات تا برقرار بودن شرط  | <b>while</b>  ۵ |

اکنون باید به این پرسش پاسخ دهیم که این مطالب چه ربطی به تشخیص چهره دارند. در یک پروژه تشخیص چهره:

- دوربین باید پشت سر هم تصویر بگیرد. این یعنی استفاده از حلقه.
  - برنامه باید بررسی کند که آیا در تصویر چهره‌ای وجود دارد یا نه. این یعنی استفاده از شرط.
  - اطلاعاتی مثل وضعیت دوربین، تعداد چهره‌ها یا نام فایل تصویر باید در حافظه نگهداری شوند. این یعنی استفاده از متغیر.
  - نتیجه باید روی صفحه نمایش داده شود. این یعنی استفاده از **print** یا در مراحل بعدی نمایش تصویر.
  - یعنی حتی یک پروژه حرفه‌ای هوش مصنوعی، از همین ابزارهای ساده آغاز می‌شود.
- فرض کنید یک نگهبان در ورودی مدرسه ایستاده است. او باید چند کار را انجام دهد:
- هر بار که یک نفر وارد شد، او را بررسی کند.
  - اگر کارت شناسایی داشت، اجازه ورود بدهد.
  - اگر نداشت، از او بخواهد دوباره کارت را نشان دهد.
- این کار از سه بخش ساخته شده است:
- تکرار بررسی افراد: حلقه
  - بررسی داشتن یا نداشتن کارت: شرط
  - نمایش پیام مناسب: خروجی
- رایانه نیز در پروژه تشخیص چهره همین کار را انجام می‌دهد. فقط به جای افراد، فریم‌های دوربین را بررسی می‌کند.

در این جلسه می خواهیم یک برنامه ساده بنویسیم که نقش یک سیستم جست و جو را بازی کند. این برنامه از کاربر می پرسد آیا چهره پیدا شده است یا نه. اگر کاربر پاسخ «بله» بدهد، برنامه پیام مناسب چاپ می کند. اگر پاسخ دیگری بدهد، اعلام می کند که هنوز در حال جست و جو است. این مثال هنوز از دوربین واقعی استفاده نمی کند، اما دقیقاً منطق فکری لازم برای پروژه اصلی را در اختیار دانش آموز قرار می دهد.

هدف این است که دانش آموز بفهمد چگونه یک برنامه می تواند:

- ورودی بگیرد،
- آن را بررسی کند،
- براساس آن تصمیم بگیرد.

```
print("سیستم جستجوی چهره فعال شد")
answer = input("آیا چهره پیدا شد؟ (بله/خیر) ")
if answer == "بله":
    print("چهره شناسایی شد")
else:
    print("هنوز چهره ای پیدا نشده است")
```

خط اول: این خط فقط یک پیام آغازین روی صفحه نشان می دهد تا کاربر بداند برنامه شروع شده است. خط دوم: در این خط، برنامه از کاربر سؤال می پرسد و پاسخ او را در متغیری به نام answer ذخیره می کند. خط سوم تا ششم: در این بخش:

- اگر مقدار answer برابر "بله" باشد، برنامه پیام «چهره شناسایی شد» را چاپ می کند.
- اگر هر چیز دیگری وارد شود، پیام دوم نمایش داده می شود.

### استفاده از حلقه برای جست و جوی مداوم

در یک سیستم واقعی، دوربین فقط یک تصویر را بررسی نمی کند؛ بلکه تصویرها را به صورت پیوسته دریافت و بررسی می کند.

برای درک مفهوم تکرار، برنامه ای می نویسیم که چند بار پیام «در حال جست و جو...» را نمایش دهد. این برنامه شبیه شرایطی است که سیستم هنوز چهره ای شناسایی نکرده و همچنان در حال بررسی تصویرها است.

```
count = 1
while count <= 5:
    print("در حال جست و جو...")
    count = count + 1
print("بررسی اولیه پایان یافت")
```

## بررسی عملکرد برنامه

خط اول:

یک متغیر به نام count ایجاد می کنیم تا تعداد دفعات تکرار را شمارش کند.

خط دوم:

تا زمانی که مقدار count کمتر یا مساوی ۵ باشد، دستورهای داخل حلقه اجرا می شوند.

خط بعد:

پیام «در حال جست و جو...» در هر بار اجرای حلقه نمایش داده می شود.

مرحله بعد:

پس از هر بار اجرای حلقه، مقدار count یک واحد افزایش پیدا می کند.

پایان برنامه:

وقتی شرط حلقه برقرار نباشد، برنامه از حلقه خارج شده و پیام پایان جست و جو را نمایش می دهد.

## فعالیت کلاسی

شبیه سازی یک سیستم جست و جو

برنامه ای بنویسید که:

۱. یک پیام شروع نمایش دهد.
۲. سه بار از کاربر پرسد: «آیا چهره پیدا شد؟»
۳. اگر کاربر پاسخ «بله» داد، پیام «چهره پیدا شد» را نمایش دهد.
۴. اگر پاسخ «خیر» بود، پیام «جست و جو ادامه دارد» را چاپ کند.

## خلاصه درس

- برنامه نویسی یعنی نوشتن دستورهای دقیق و مرحله به مرحله برای رایانه.
- پایتون زبانی ساده، خوانا و پر کاربرد در پروژه های هوش مصنوعی است.
- متغیرها برای ذخیره اطلاعات در برنامه استفاده می شوند.
- تابع input() اطلاعات را از کاربر دریافت می کند.
- دستورهای شرطی به برنامه کمک می کنند تا بر اساس شرایط مختلف تصمیم بگیرد.
- حلقه ها برای تکرار دستورها و انجام کارهای پیوسته به کار می روند.
- تابع print() نتیجه اجرای برنامه را روی صفحه نمایش می دهد.
- یک سیستم تشخیص چهره باید تصویرها را به صورت پیوسته بررسی کند.
- در پروژه های تشخیص چهره، حلقه برای بررسی مداوم تصویرها و شرط برای تشخیص وجود چهره استفاده می شود.



## پروژه

۱. برنامه‌ای بنویسید که نام کاربر را بگیرد و پیام زیر را چاپ کند:  
سلام [نام کاربر]، به آزمایشگاه بینایی ماشین خوش آمدی.
۲. برنامه‌ای بنویسید که یک عدد از کاربر بگیرد. اگر عدد بزرگ‌تر از ۰ بود، چاپ کند:  
سیستم فعال است  
و اگر صفر یا منفی بود، چاپ کند:  
سیستم غیرفعال است
۳. برنامه‌ای بنویسید که ۵ بار عبارت در حال بررسی تصویر را روی صفحه چاپ کند.
۴. یک برنامه ترکیبی بنویسید که:
  - از کاربر بپرسد «آیا دوربین روشن است؟»
  - اگر پاسخ بله بود، پیام دوربین آماده دریافت تصویر است را چاپ کند.
  - اگر پاسخ خیر بود، پیام لطفاً دوربین را بررسی کنید را نمایش دهد.

## درس ۱۱: مهندسی کد؛ توابع و ساختاردهی پروژه



در پایان این درس خواهید توانست:

- مفهوم تابع و کاربرد آن در برنامه نویسی را توضیح دهید.
- با استفاده از کلمه کلیدی `def` یک تابع تعریف کنید.
- تفاوت میان تعریف و فراخوانی تابع را بیان کنید.
- با تقسیم برنامه به بخش‌های کوچک‌تر، کدهای منظم‌تری بنویسید.
- از توابع برای سازمان‌دهی بخش‌های مختلف یک پروژه استفاده کنید.



در برنامه‌های بزرگ، تعداد دستورها افزایش پیدا می‌کند و مدیریت آن‌ها دشوارتر می‌شود. برای نوشتن برنامه‌های منظم، می‌توان دستورهای مرتبط را در بخش‌های جداگانه قرار داد. توابع به ما کمک می‌کنند تا هر بخش از برنامه را برای انجام یک وظیفه مشخص طراحی کنیم. در این درس با مفهوم تابع، نحوه تعریف و فراخوانی آن و نقش توابع در سازمان‌دهی پروژه آشنا می‌شویم.

۱. اگر بخواهید یک دستور را چند بار در برنامه اجرا کنید، آیا لازم است آن را هر بار دوباره بنویسید؟
۲. چگونه می‌توان دستورهای مربوط به یک کار مشخص را در یک بخش جداگانه قرار داد؟

## مهندسی کد و ساختاردهی پروژه



در دنیای واقعی، هنگامی که می‌خواهیم کاری پیچیده انجام دهیم، معمولاً آن را به چند مرحله کوچک‌تر تقسیم می‌کنیم. برای مثال، هنگام ساخت یک مدل هواپیما یا تهیه یک غذای مخصوص، انجام کار را مرحله‌به‌مرحله پیش می‌بریم. در برنامه‌نویسی نیز برای ساخت یک سیستم تشخیص چهره، بهتر است همه دستورها را به صورت پشت سر هم و در یک بخش ننویسیم. اگر برنامه شامل صدها خط کد باشد، پیدا کردن خطاها و ایجاد تغییر در آن دشوار خواهد شد. برای سازمان‌دهی بهتر برنامه، می‌توانیم دستورهای مرتبط را در بخش‌های جداگانه قرار دهیم. در اینجا مفهوم تابع (Function) به ما کمک می‌کند.

تابع مانند یک «جعبه ابزار» است. دستورهای مربوط به یک کار مشخص را داخل این جعبه قرار می‌دهیم و هر زمان به آن‌ها نیاز داشته باشیم، با فراخوانی نام تابع، همه دستورهای موجود در آن اجرا می‌شوند.

## تابع چیست؟ — تشبیه قطعات لگو

تصور کنید با قطعات لگو یک «چرخ» ساخته‌اید. اگر بخواهید برای یک ماشین چهار چرخ داشته باشید، آیا لازم است چهار بار مراحل ساخت چرخ را از ابتدا انجام دهید؟

خیر. می‌توانید یک الگوی مشخص برای ساخت چرخ داشته باشید و هر زمان به آن نیاز دارید، از آن استفاده کنید. تابع نیز به همین شکل عمل می‌کند:

- ابتدا تابع را تعریف می‌کنیم.
  - سپس هر زمان به اجرای دستورهای آن نیاز داشته باشیم، تابع را فراخوانی می‌کنیم.
- در پایتون، برای تعریف تابع از کلمه کلیدی `def` استفاده می‌کنیم. واژه `def` کوتاه‌شده کلمه `define` و به معنای «تعریف کردن» است.

## ساختار تابع در پایتون

```
def say_hello():
    print("سلام! به بخش تشخیص چهره خوش آمدید")
```

هر تابع معمولاً از سه بخش اصلی تشکیل می‌شود:

- کلمه کلیدی `def`: مشخص می‌کند که قصد تعریف یک تابع را داریم.
- نام تابع: نامی مناسب و گویا است که وظیفه تابع را نشان می‌دهد.
- بدنه تابع: شامل دستورهایی است که وظیفه مشخص شده را انجام می‌دهند.

دستورهای داخل تابع باید با تورفتگی (Indentation) نوشته شوند. تورفتگی به پایتون نشان می دهد که کدام دستورها به تابع مربوط هستند.

نکته: تعریف کردن یک تابع به معنای اجرای آن نیست. دستورهای داخل تابع فقط زمانی اجرا می شوند که تابع را فراخوانی کنیم.

## مدیریت بخش های مختلف پروژه

فرض کنید در پروژه تشخیص چهره، دو کار اصلی داریم:

- روشن کردن دوربین
- پردازش تصویر

می توانیم برای هر یک از این کارها یک تابع جداگانه تعریف کنیم. این کار باعث می شود کد برنامه مرتب تر، خوانا تر و مدیریت آن آسان تر شود.

برای مثال، می توانیم تابعی به نام `start_camera()` برای روشن کردن دوربین ایجاد کنیم.

```
# تعریف تابع اول
def start_camera():
    print("دوربین روشن شد")

# تعریف تابع دوم
def process_image():
    print("در حال پردازش تصویر برای یافتن چهره")

# فراخوانی توابع
start_camera()
process_image()
```

## توضیح کد

`def start_camera()`: این خط به پایتون می گوید ما می خواهیم تابعی به نام `start_camera` بسازیم.

`print("...")`: دوربین روشن شد این دستوری است که داخل تابع قرار دارد و فقط در صورت فراخوانی اجرا می شود.

`start_camera()`: در این خط ما تابع را صدا می زنیم (اجرا می کنیم). تا وقتی این خط نوشته نشود، دستورات داخل تابع هیچ گاه اجرا نخواهند شد.

## فعالیت کلاسی

ساخت سیستم «هشدار هوشمند»

سه تابع زیر را تعریف کنند و سپس آن ها را به ترتیب فراخوانی کنند:

۱. `check_sensor()`: چاپ کند "سنسور در حال اسکن محیط ..."
۲. `detect_face()`: چاپ کند "چهره ای در کادر دیده شد!"
۳. `alarm_on()`: چاپ کند "هشدار! چهره شناسایی شد."

```
def check_sensor():
    print("...سنسور در حال اسکن محیط")

def detect_face():
    print("!چهره ای در کادر دیده شد")

def alarm_on():
    print("...هشدار! چهره شناسایی شد")

# اجرای فرآیند
check_sensor()
detect_face()
alarm_on()
```

### خلاصه درس

- تابع مجموعه ای از دستورها برای انجام یک کار مشخص است.
- توابع باعث می شوند کدها منظم تر و خوانا تر شوند.
- برای تعریف تابع در پایتون از کلمه کلیدی `def` استفاده می کنیم.
- نام تابع باید ساده، مناسب و بیان کننده وظیفه آن باشد.
- دستورهای داخل تابع با تورفتگی نوشته می شوند.
- تعریف تابع به معنای اجرای آن نیست.
- برای اجرای دستورهای تابع باید آن را فراخوانی کنیم.
- توابع به تقسیم پروژه های بزرگ به بخش های کوچک تر کمک می کنند.
- استفاده از توابع، پیدا کردن خطاها و ایجاد تغییر در برنامه را آسان تر می کند.
- در پروژه تشخیص چهره می توان برای بخش هایی مانند روشن کردن دوربین و پردازش تصویر، توابع جداگانه ایجاد کرد.

### پروژه

۱. تابعی بنویسید به نام `system_info` که نام پروژه (تشخیص چهره) و نسخه آن (۱.۰) را چاپ کند.
۲. تابعی بنویسید به نام `clear_screen` که پیام "حافظه پاکسازی شد" را نمایش دهد.
۳. یک برنامه بنویسید که از کاربر بپرسد "آیا می خواهید سیستم را فعال کنید؟". اگر کاربر گفت بله، تابع `start_camera` (از مثال بالا) را فراخوانی کنید.
۴. آیا می توانید یک تابع بسازید که دو عدد را بگیرد و حاصل جمع آن ها را (به عنوان مختصات X و Y) چاپ کند؟

## درس ۱۲: آماده سازی کارگاه؛ نصب و راه اندازی کتابخانه OpenCV



در پایان این درس خواهید توانست:

- مفهوم کتابخانه را در برنامه نویسی توضیح دهید.
- نقش کتابخانه ها را در ساخت برنامه های پیشرفته بیان کنید.
- با کاربرد کتابخانه OpenCV در پردازش تصویر آشنا شوید.
- وظیفه ابزار pip را توضیح دهید.
- برخی از خطاهای رایج هنگام نصب کتابخانه ها را شناسایی کنید.

تا این مرحله، با ابزارها و دستورهای پایه پایتون آشنا شدیم. اما برای انجام کارهای تخصصی، مانند پردازش تصویر و تشخیص چهره، به ابزارهای بیشتری نیاز داریم. کتابخانه‌ها مجموعه‌ای از کدهای آماده هستند که به ما کمک می‌کنند برنامه‌های پیشرفته‌تری بسازیم. در این درس با مفهوم کتابخانه، ابزار pip و روش نصب کتابخانه OpenCV آشنا می‌شویم.

۱. آیا برای انجام همه کارها در برنامه‌نویسی باید کدها را از ابتدا بنویسیم؟
۲. به نظر شما برنامه‌نویسان چگونه از کدهای آماده دیگران استفاده می‌کنند؟

## آشنایی با کتابخانه‌ها و نصب OpenCV



تا این جلسه، با ابزارها و دستورهای پایه پایتون کار کردیم. پایتون به تنهایی یک زبان برنامه‌نویسی قدرتمند است؛ اما برای انجام کارهای تخصصی، مانند پردازش تصویر و تشخیص چهره، به ابزارهای پیشرفته‌تری نیاز داریم. تصور کنید پایتون مانند «موتور یک خودرو» است. موتور برای حرکت و انجام وظایف مختلف به بخش‌هایی مانند چرخ، فرمان، دنده و بدنه نیاز دارد. در برنامه‌نویسی نیز می‌توان ابزارها و قابلیت‌های آماده را به پایتون اضافه کرد. این ابزارها که پیش‌تر توسط برنامه‌نویسان و متخصصان ساخته شده‌اند، کتابخانه نام دارند. در این درس، یکی از معروف‌ترین کتابخانه‌های پردازش تصویر، یعنی OpenCV، را روی رایانه نصب و برای استفاده در پروژه آماده می‌کنیم.

### کتابخانه چیست؟

کتابخانه در برنامه‌نویسی، مجموعه‌ای از کدها و ابزارهای آماده است که برای انجام کارهای مشخص طراحی شده‌اند. می‌توان کتابخانه‌های برنامه‌نویسی را به کتابخانه‌های دنیای واقعی تشبیه کرد. اگر بخواهید اطلاعاتی درباره تاریخ به دست آورید، لازم نیست خودتان به گذشته سفر کنید یا یک کتاب تاریخی را از ابتدا بنویسید. می‌توانید به کتابخانه مراجعه کنید و از کتاب‌هایی استفاده کنید که پیش‌تر توسط تاریخ‌دانان نوشته شده‌اند.

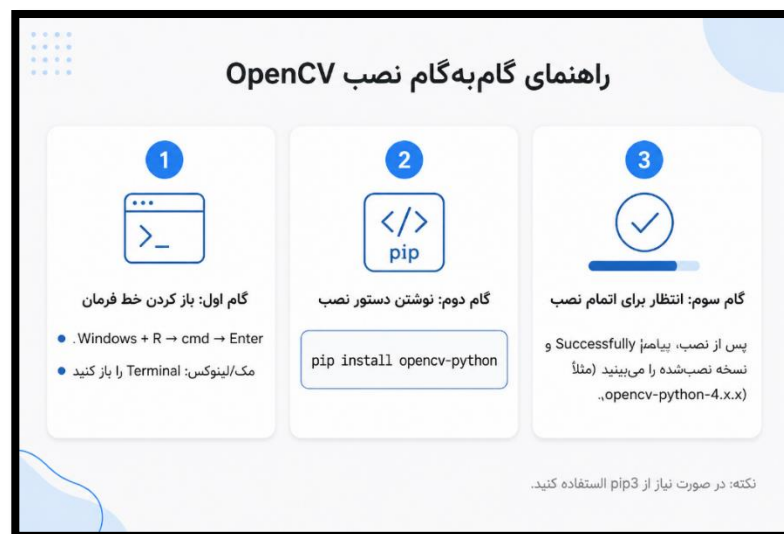
در برنامه‌نویسی نیز لازم نیست همه کدها را از ابتدا بنویسیم. برای مثال، در پروژه تشخیص چهره نیازی نیست خودمان کدهای پیچیده مربوط به بررسی پیکسل‌های تصویر را طراحی کنیم. می‌توانیم از ابزارها و کدهای آماده‌ای استفاده کنیم که متخصصان در قالب کتابخانه‌ها در اختیار برنامه‌نویسان قرار داده‌اند.

یکی از این کتابخانه‌ها OpenCV است. نام OpenCV کوتاه‌شده عبارت Open Source Computer Vision و به معنای «بنیادی رایانه‌ای متن‌باز» است.

کتابخانه OpenCV ابزارهای متنوعی برای کار با تصویر و ویدئو در اختیار ما قرار می‌دهد. از این کتابخانه می‌توان در پروژه‌هایی مانند پردازش تصویر، تشخیص چهره، شناسایی اشیاء و تحلیل ویدئو استفاده کرد.

## معرفی ابزار pip

برای نصب کتابخانه‌ها در پایتون، از ابزاری به نام pip استفاده می‌کنیم. pip وظیفه دارد کتابخانه‌های موردنیاز را از اینترنت دریافت و به صورت خودکار نصب کند. به همین دلیل می‌توان آن را «مدیر بسته‌های پایتون» نامید. برای استفاده از pip، معمولاً دستورهای نصب را در محیط خط فرمان سیستم عامل وارد می‌کنیم: در سیستم عامل ویندوز از Command Prompt یا CMD استفاده می‌شود. در سیستم عامل‌های مک و لینوکس از Terminal استفاده می‌شود. پس از باز کردن محیط خط فرمان، می‌توانیم دستور نصب کتابخانه را وارد و اجرا کنیم.



## خطاهای رایج هنگام نصب و روش حل آن‌ها

طول فرآیند نصب، ممکن است با مشکلاتی مواجه شوید. در جدول زیر خطاهای متداول و راهکار آن‌ها آمده است:

| خطا / مشکل                      | علت احتمالی                                        | راه حل                                                                                                                                         |
|---------------------------------|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>pip is not recognized...</p> | مسیر پایتون در متغیرهای سیستم عامل تعریف نشده است. | <p>هنگام نصب مجدد پایتون، گزینه Add Python to PATH را تیک بزنید. یا دستور زیر را اجرا کنید:</p> <pre>python -m pip install opencv-python</pre> |
| <p>WARNING: Retrying...</p>     | عدم اتصال به اینترنت یا ضعیف بودن شبکه.            | <p>اتصال اینترنت خود را بررسی کنید. در اینترنت مدارس، پروکسی یا محدودیت‌های شبکه را نیز بررسی نمایید.</p>                                      |
| <p>Permission Denied</p>        | عدم دسترسی مدیر سیستم (Administrator).             | <p>خط فرمان را Run as Administrator اجرا کنید، یا در مک/لینوکس بنویسید:</p> <pre>sudo pip install opencv-python</pre>                          |

در برنامه ساده زیر، ما ابتدا کتابخانه `cv2` (که نام رسمی `OpenCV` در پایتون است) را به برنامه معرفی می کنیم. سپس از رایانه می خواهیم نسخه دقیق این کتابخانه را روی صفحه چاپ کند. اگر کدی بدون خطا اجرا شود و شماره نسخه را نشان دهد، یعنی کتابخانه ما کاملاً آماده است!

```
# وارد کردن کتابخانه پردازش تصویر به برنامه
import cv2

# چاپ پیام تایید و نسخه کتابخانه
print("با موفقیت نصب شد OpenCV تبریک! کتابخانه")
print("نسخه نصب شده روی سیستم شما", cv2.__version__)
```

## فعالیت کلاسی

عیب یاب هوشمند نصب کتابخانه با استفاده از ساختار مدیریت خطای `try/except` (که در سال گذشته یاد گرفته اند) برنامه ای بنویسند که تلاش کند کتابخانه `cv2` را وارد برنامه کند.

- اگر کتابخانه با موفقیت وارد شد، پیام «کتابخانه آماده کار است» چاپ شود.
- اگر سیستم با خطای عدم وجود کتابخانه (`ModuleNotFoundError`) مواجه شد، به جای بسته شدن برنامه با خطای قرمز، پیام راهنمای «لطفاً ابتدا با دستور `pip install opencv-python` کتابخانه را نصب کنید» نمایش داده شود.

## خلاصه درس

- کتابخانه مجموعه ای از کدها و ابزارهای آماده است.
- کتابخانه ها به ما کمک می کنند برنامه های پیشرفته تری بسازیم.
- استفاده از کتابخانه ها باعث کاهش زمان و حجم کدنویسی می شود.
- در پایتون، کتابخانه `OpenCV` با نام `cv2` استفاده می شود.
- ابزار `pip` برای دریافت و نصب کتابخانه های پایتون به کار می رود.
- دستورهای نصب معمولاً در محیط خط فرمان اجرا می شوند.
- در ویندوز از `CMD` و در مک و لینوکس از `Terminal` استفاده می شود.
- برای نصب `OpenCV` می توان از دستور `pip install opencv-python` استفاده کرد.
- نمایش شماره نسخه `OpenCV` نشان می دهد که کتابخانه به درستی نصب شده و آماده استفاده است.



پروژه

۱. دستور دقیق نصب کتابخانه OpenCV در خط فرمان را یادداشت کنید.
۲. برنامه‌ای بنویسید که اگر نسخه OpenCV نصب شده روی سیستم بزرگ‌تر یا مساوی نسخه ۴.۰ بود، پیام سیستم شما به روز است و در غیر این صورت پیام پیشنهاد می‌شود کتابخانه را به‌روزرسانی کنید را چاپ کند.
۳. تحقیق کنید که مخفف کلمه OpenCV چیست و چه کاربردهایی در دنیای واقعی دارد؟ (یک بند توضیح دهید).
۴. دستور ارتقای یک کتابخانه به آخرین نسخه با استفاده از pip را در اینترنت جست‌وجو کرده و بنویسید.

## درس ۱۳: تصویر دیجیتال در پشت صحنه؛ پیکسل ها و مدل های رنگی



در پایان این درس خواهید توانست:

- مفهوم تصویر دیجیتال و پیکسل را توضیح دهید.
- بیان کنید که رایانه چگونه تصاویر را به داده های عددی تبدیل می کند.
- مفهوم شدت نور و بازه عددی ۰ تا ۲۵۵ را در تصاویر توضیح دهید.
- تفاوت مدل های رنگی RGB و BGR را بیان کنید.
- دلیل تبدیل تصاویر رنگی به تصویر خاکستری در پردازش تصویر را توضیح دهید.



انسان‌ها با نگاه کردن به یک تصویر می‌توانند چهره، رنگ‌ها و اشیای مختلف را تشخیص دهند؛ اما رایانه مانند انسان قدرت دیدن و درک تصویر را ندارد. رایانه فقط با عددها و داده‌ها کار می‌کند. برای اینکه بتوانیم پروژه‌هایی مانند تشخیص چهره بسازیم، ابتدا باید بدانیم یک تصویر چگونه به زبان رایانه تبدیل می‌شود. در این درس با ساختار تصاویر دیجیتال و نحوه ذخیره‌سازی آن‌ها در حافظه رایانه آشنا می‌شویم.

۱. وقتی یک عکس را روی صفحه گوشی مشاهده می‌کنیم، رایانه آن تصویر را چگونه ذخیره کرده است؟
۲. آیا رایانه واقعاً رنگ‌ها و چهره‌ها را مانند انسان می‌بیند؟
۳. کوچک‌ترین بخش تشکیل‌دهنده یک تصویر دیجیتال چیست؟
۴. به نظر شما چرا در پروژه‌های تشخیص چهره، گاهی تصاویر رنگی به سیاه و سفید تبدیل می‌شوند؟



### تصویر دیجیتال در پشت صحنه؛ پیکسل‌ها و مدل‌های رنگی

زمانی که ما به یک عکس روی صفحه نمایش رایانه یا گوشی خود نگاه می‌کنیم، چهره یک دوست، یک منظره زیبا یا یک متن را می‌بینیم. اما رایانه هیچ درک بصری شبیه به ما انسان‌ها ندارد. رایانه چشم ندارد؛ او فقط ریاضی و عددها را می‌فهمد! برای اینکه بتوانیم پروژه‌ای بنویسیم که چهره انسان را در تصویر تشخیص دهد، ابتدا باید بدانیم رایانه چگونه یک تصویر را به زبان خودش (یعنی به صورت مجموعه‌ای از اعداد) ترجمه می‌کند. در این جلسه، پشت صحنه تصاویر دیجیتال را بررسی می‌کنیم و می‌بینیم که یک عکس ساده، چطور به یک جدول بزرگ از اعداد تبدیل می‌شود.

### تصویر دیجیتال چیست؟ (تشبیه موزاییک‌ها)

یک دیوار موزاییکی را در نظر بگیرید که از هزاران مربع کوچک رنگی در کنار هم تشکیل شده است. وقتی از دور به این دیوار نگاه می‌کنید، یک نقاشی زیبا می‌بینید؛ اما وقتی خیلی به آن نزدیک می‌شوید، فقط مربع‌های تک‌رنگ کوچک را مشاهده می‌کنید. تصاویر دیجیتال نیز دقیقاً همین‌گونه هستند. هر تصویر دیجیتال از میلیون‌ها نقطه بسیار کوچک به نام پیکسل (Pixel) تشکیل شده است. پیکسل کوچک‌ترین جزء سازنده یک تصویر دیجیتال است. وقتی یک تصویر را در رایانه باز می‌کنیم، پایتون آن را به صورت یک ماتریس (جدول دو بعدی) از پیکسل‌ها می‌بیند. هر خانه از این جدول، نشان‌دهنده یک پیکسل است که مقدار عددی مشخصی دارد.

### شدت نور؛ بازه عددی ۰ تا ۲۵۵

در یک تصویر سیاه و سفید ساده، هر پیکسل فقط میزان تاریکی یا روشنایی آن نقطه را نشان می‌دهد. رایانه برای نشان دادن این میزان روشنایی (شدت نور)، از یک بازه عددی بین ۰ تا ۲۵۵ استفاده می‌کند:

- عدد ۰ نماینده رنگ کاملاً سیاه (نبود هیچ نوری) است.
- عدد ۲۵۵ نماینده رنگ کاملاً سفید (بیشترین میزان نور) است.
- اعداد بین این دو (مثلاً ۱۲۷)، طیف‌های مختلف رنگ خاکستری هستند.

## مدل های رنگی: تفاوت RGB و BGR

تصاویر رنگی کمی پیچیده تر هستند. یک پیکسل رنگی نمی تواند فقط با یک عدد ساخته شود. در دنیای دیجیتال، تمام رنگ ها از ترکیب سه رنگ اصلی ساخته می شوند: قرمز (Red)، سبز (Green) و آبی (Blue). به این ترکیب، مدل رنگی RGB می گویند.

در مدل RGB، برای هر پیکسل ۳ عدد داریم:

- مقدار کانال قرمز (۰ تا ۲۵۵)
- مقدار کانال سبز (۰ تا ۲۵۵)
- مقدار کانال آبی (۰ تا ۲۵۵)

اکثر نرم افزارها و زبان های برنامه نویسی تصاویر را با ترتیب RGB (ابتدا قرمز، بعد سبز، بعد آبی) می خوانند. اما کتابخانه OpenCV به دلایل تاریخی، تصاویر را با ترتیب برعکس یعنی BGR (ابتدا آبی، بعد سبز، بعد قرمز) در حافظه ذخیره می کند! این یعنی اگر مراقب نباشیم و تصویر OpenCV را در کتابخانه های دیگر نمایش دهیم، ممکن است رنگ پوست صورت آبی یا رنگ آسمان قرمز دیده شود.

تصویر خاکستری (Grayscale) چیست و چرا مهم است؟

تصویر رنگی اطلاعات زیادی دارد (برای هر پیکسل ۳ عدد). پردازش این حجم از اطلاعات برای رایانه زمان بر است. در پروژه تشخیص چهره، رنگ پوست یا رنگ لباس اهمیتی ندارد؛ چیزی که برای الگوریتم مهم است، شکل هندسی اجزای صورت (مانند جایگاه چشم ها، بینی و دهان) و سایه های روی آن ها است. بنابراین، برنامه نویسان بینایی ماشین معمولاً در اولین گام، تصویر رنگی را به تصویر خاکستری (Grayscale) تبدیل می کنند. با این کار:

۱. حجم داده ها به یک سوم کاهش می یابد (فقط یک کانال به جای سه کانال).
۲. سرعت پردازش و تشخیص چهره به شدت افزایش می یابد.

## شبیه سازی ماتریس یک تصویر

از آنجا که هنوز وارد کدنویسی باز کردن فایل تصویر نشده ایم، می خواهیم با ابزار ساده پایتون شبیه سازی کنیم که یک تصویر کوچک سیاه و سفید (مثلاً با ابعاد ۳ در ۳ پیکسل) چگونه در حافظه رایانه ذخیره می شود. در این کد، ما یک لیست دو بعدی (جدول) می سازیم که نماینده یک تصویر بسیار کوچک سیاه و سفید است. پیکسل وسط سفید، پیکسل های گوشه سیاه و بقیه خاکستری هستند. برنامه این جدول عددی را چاپ می کند تا ببینیم رایانه چگونه تصویر را ذخیره می کند.

```
# شبیه سازی یک تصویر ۳ در ۳ پیکسل
# ۰ = سفید = ۲۵۵، خاکستری = ۱۲۸، سیاه = ۰
image_matrix = [
    [0, 128, 0],
    [128, 255, 128],
    [0, 128, 0]
]

print("ماتریس تصویر در حافظه رایانه:")
for row in image_matrix:
    print(row)
```

## توضیح کد

- image\_matrix [...] : یک لیست دو بعدی پایتون می سازیم که نقش جدول پیکسل ها را دارد.
- for row in image\_matrix: : با استفاده از یک حلقه، ردیف های این جدول را یکی یکی می خوانیم.
- print(row) : هر ردیف عددی را روی صفحه چاپ می کنیم. این دقیقاً همان چیزی است که پایتون هنگام خواندن تصاویر واقعی می بیند.

## فعالیت کلاسی

پیدا کردن شدت نور رنگ ها

با توجه به توضیحات درس، مقادیر سه کانال رنگی (آبی، سبز، قرمز) را برای رنگ های زیر حدس بزنند و روی کاغذ بنویسند. سپس در پایتون یک دیکشنری ساده بسازند که این رنگ ها و مقادیر عددی BGR آن ها را ذخیره کرده و نمایش دهد:

- رنگ سفید خالص
- رنگ سیاه خالص
- رنگ آبی خالص در سیستم BGR
- رنگ قرمز خالص در سیستم BGR

## خلاصه درس

- رایانه تصاویر را مانند انسان مشاهده نمی کند و آن ها را به عدد تبدیل می کند.
- هر تصویر دیجیتال از بخش های کوچکی به نام پیکسل ساخته شده است.
- پیکسل کوچک ترین جزء تشکیل دهنده تصویر دیجیتال است.
- تصاویر در رایانه به شکل ماتریس هایی از عددها ذخیره می شوند.
- شدت نور در تصاویر سیاه و سفید با عددهای ۰ تا ۲۵۵ نمایش داده می شود.
- مدل RGB از سه رنگ قرمز، سبز و آبی تشکیل شده است.
- کتابخانه OpenCV تصاویر را با ترتیب BGR ذخیره می کند.
- تصاویر خاکستری اطلاعات کمتری دارند و سریع تر پردازش می شوند.

- در پروژه‌های تشخیص چهره، تبدیل تصویر به خاکستری باعث افزایش سرعت پردازش می‌شود.
- رایانه برای پردازش تصویر ابتدا باید تصویر را به داده‌های عددی تبدیل کند.



۱. پیکسل چیست؟ آن را با یک مثال روزمره توضیح دهید.

۲. چرا در پروژه‌های پردازش تصویر و تشخیص چهره، تصاویر را به حالت خاکستری (Grayscale) تبدیل می‌کنیم؟

۳. در سیستم رنگی BGR، مقادیر عددی زیر چه رنگی را تولید می‌کنند؟

$(0, 255, 0)$   
 $(255, 255, 255)$   
 $(0, 0, 0)$

۴. یک تصویر رنگی با ابعاد ۱۰۰ در ۱۰۰ پیکسل، در مجموع از چند عدد تشکیل شده است؟ (راهنمایی: هر پیکسل رنگی دارای ۳ عدد برای کانال‌های BGR است).

## درس ۱۴: اولین تعامل با تصویر؛ خواندن، نمایش و ذخیره تصویر با OpenCV



در پایان این درس خواهید توانست:

- تصاویر دیجیتال را با استفاده از کتابخانه OpenCV در محیط پایتون بارگذاری و مدیریت کنید.
- با استفاده از دستورات مناسب، یک تصویر را در پنجره‌ای جداگانه نمایش دهید.
- نقش توابع `imread`، `imshow`، `waitKey` و `destroyAllWindows` را در کار با تصاویر درک کنید.
- مشکلات رایج هنگام باز کردن تصاویر مانند اشتباه بودن مسیر فایل را شناسایی و برطرف کنید.
- یک تصویر پردازش شده یا کپی تصویر اصلی را با نام جدید ذخیره کنید و چرخه کامل کار با تصویر را اجرا کنید.

تا این مرحله، با مفهوم تصویر دیجیتال و نحوه نمایش آن در حافظه رایانه آشنا شدیم. یاد گرفتیم که رایانه برخلاف انسان، تصویر را به صورت مجموعه‌ای از اعداد و داده‌ها مشاهده می‌کند. اکنون زمان آن رسیده است که به صورت عملی با تصاویر واقعی کار کنیم. در پروژه تشخیص چهره، اولین قدم این است که بتوانیم تصویر را وارد برنامه کنیم، آن را مشاهده کنیم و در صورت نیاز تغییرات انجام‌شده را ذخیره کنیم. در این جلسه با استفاده از کتابخانه OpenCV یاد می‌گیریم چگونه یک تصویر واقعی مانند فایل‌های JPG یا PNG را در پایتون باز کنیم، نمایش دهیم و یک نسخه جدید از آن ایجاد کنیم. این مهارت‌ها پایه اصلی بسیاری از پروژه‌های بینایی ماشین هستند.

۱. وقتی یک عکس را در گوشی یا رایانه باز می‌کنیم، چه اتفاقی در پشت صحنه رخ می‌دهد؟
۲. اگر برنامه‌ای بخواهد یک تصویر را بررسی کند، ابتدا باید چه کاری انجام دهد؟
۳. چرا یک برنامه نمی‌تواند فقط با دستور نمایش تصویر، آن را برای مدت طولانی روی صفحه نگه دارد؟
۴. اگر آدرس یک فایل تصویر اشتباه باشد، به نظر شما برنامه چه رفتاری خواهد داشت؟

### خواندن یک تصویر در پایتون

این سه عملیات (خواندن، نمایش و ذخیره) پایه‌ای‌ترین زنجیره عملیاتی در هر سیستم بینایی ماشین هستند.

| تابع                                                                                                        | کاربرد                                             | نمونه کد                                                                       |
|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------------------|
| <b>cv2.imread()</b><br>  | خواندن تصویر از مسیر فایل و ورود پیکسل‌ها به حافظه | <pre>image = cv2.imread("path/to/image.jpg")</pre> <p>در مسیر نادرست: None</p> |
| <b>cv2.imshow()</b><br>  | نمایش تصویر موجود در حافظه در یک پنجره             | <pre>cv2.imshow("Window Name", image)</pre>                                    |
| <b>cv2.imwrite()</b><br> | ذخیره تصویر موجود در حافظه به صورت فایل روی دیسک   | <pre>cv2.imwrite("new_image.jpg", image)</pre>                                 |

اگر کدی بنویسید که فقط شامل cv2.imshow باشد، پنجره تصویر برای یک هزارم ثانیه باز شده و بلافاصله برنامه بسته می شود! برای حل این مشکل به دو تابع زیر نیاز داریم:

### ۱. تابع cv2.waitKey()

این تابع برنامه را متوقف نگه می دارد تا کاربر یک کلید روی کیبورد فشار دهد.

- اگر داخل پرانتز عدد ۰ قرار دهیم (cv2.waitKey(۰))، برنامه تا زمانی که کاربر دکمه ای را فشار ندهد، منتظر می ماند.

- اگر عددی بزرگ تر از صفر (مثلاً ۱۰۰۰) قرار دهیم، برنامه به همان میزان میلی ثانیه منتظر مانده و سپس به طور خودکار عبور می کند.

### ۲. تابع cv2.destroyAllWindows()

پس از اینکه کارمان با نمایش تصاویر تمام شد، برای اینکه حافظه موقت رایانه اشغال نماند و پنجره های اضافی باز نمانند، از این تابع برای بستن تمامی پنجره های ایجاد شده توسط OpenCV استفاده می کنیم.

## چرخه کامل مدیریت تصویر

ما یک فایل تصویر فرضی به نام face.jpg را می خوانیم، آن را نمایش می دهیم، منتظر می مانیم تا کاربر کلیدی فشار دهد و در نهایت آن را با نام دیگری ذخیره می کنیم.

پیش نیاز اجرا: برای اجرای درست این کد، باید یک فایل تصویر با نام face.jpg در همان پوشه ای که فایل پایتون شما قرار دارد، وجود داشته باشد.

## توضیح کد

```
import cv2

# ۱. تصویر از روی دیسک خوان
image = cv2.imread("face.jpg")

# بررسی اینکه آیا تصویر با موفقیت خوانده شده است یا خیر
if image is None:
    print("خطا: فایل تصویر یافت نشد یا خراب است")
else:
    print("تصویر با موفقیت خوانده شد")

# ۲. نمایش تصویر در یک پنجره به نام 'My Window'
cv2.imshow("My Window", image)

# منتظر ماندن برای فشردن یک کلید توسط کاربر
print("...برای بستن پنجره و ذخیره تصویر، یکی از کلیدهای کیبورد را فشار دهید")
cv2.waitKey(0)

# ۳. ذخیره تصویر در قالب یک فایل جدید
cv2.imwrite("face_copy.jpg", image)
print("ذخیره شد face_copy.jpg تصویر کپی با نام")

# بستن تمام پنجره های باز شده
cv2.destroyAllWindows()
```

- `image = cv2.imread("face.jpg")`: تصویر `face.jpg` را می خواند و در متغیر `image` ذخیره می کند.
- `if image is None`: بررسی می کند که آیا متغیر حاوی داده است یا به دلیل اشتباه بودن آدرس، خالی (None) مانده است.
- `cv2.imshow("My Window", image)`: پنجره ای با عنوان "My Window" باز کرده و تصویر ذخیره شده در متغیر `image` را نشان می دهد.
- `cv2.waitKey(0)`: برنامه را متوقف نگه می دارد تا کاربر کلیدی را روی صفحه کلید بفشارد. بدون این خط، پنجره بلافاصله بسته می شود.
- `cv2.imwrite("face_copy.jpg", image)`: یک کپی از فایل تصویر را با نام جدید ذخیره می کند.
- `cv2.destroyAllWindows()`: تمام پنجره های گرافیکی باز شده توسط برنامه را می بندد تا منابع سیستم آزاد شوند.

### فعالیت کلاسی

عنوان فعالیت: نمایشگر مشروط تصویر  
هدف فعالیت: تمرین کار با مسیر فایل ها، بررسی وجود فایل و نمایش تصویر بر اساس ورودی کاربر.  
بخواهید برنامه ای بنویسند که:

- ابتدا نام یک فایل تصویر را با دستور `input()` از کاربر بگیرد.
- تلاش کند آن تصویر را بخواند.
- اگر فایل وجود داشت، آن را نمایش دهد و ۵ ثانیه (۵۰۰۰ میلی ثانیه) بعد پنجره را خودکار ببندد.
- اگر فایل وجود نداشت، پیغام مناسبی روی صفحه چاپ کند.

### خلاصه درس

- تصاویر را با استفاده از تابع `cv2.imread()` وارد برنامه پایتون کنیم و آن ها را به صورت داده های قابل پردازش در اختیار رایانه قرار دهیم.
- درستی خواندن تصویر را بررسی کنیم تا در صورت اشتباه بودن مسیر یا نام فایل، برنامه با خطا مواجه نشود.
- تصاویر را با استفاده از تابع `cv2.imshow()` نمایش دهیم و با مفهوم ایجاد پنجره نمایش در OpenCV آشنا شویم.
- چرخه اصلی کار با تصاویر در بینایی ماشین، یعنی خواندن تصویر، نمایش، پردازش و ذخیره سازی آن را شناختیم.

## تمرین

۱. وظیفه هر یک از توابع زیر را به اختصار بنویسید:

• `cv2.imread()`

.....

• `cv2.imshow()`

.....

• `cv2.imwrite()`

.....

۲. اگر از دستور `cv2.waitKey(0)` استفاده نکنیم، چه اتفاقی برای پنجره نمایش تصویر می افتد؟

.....

۳. تفاوت `cv2.waitKey(0)` با `cv2.waitKey(2000)` چیست؟

.....

## پروژه

برنامه ای بنویسید که یک تصویر به نام `input.jpg` را بخواند و آن را با نام `output.png` ذخیره کند تغییر فرمت از

JPG به PNG

## درس ۱۵: آماده سازی تصویر برای تشخیص چهره؛ تغییر اندازه و تبدیل به خاکستری



در پایان این درس خواهید توانست:

- مفهوم پیش پردازش تصویر و اهمیت آن در پروژه های بینایی ماشین را توضیح دهید.
- اندازه تصاویر را با استفاده از ابزارهای OpenCV تغییر دهید و حجم داده های تصویری را کاهش دهید.
- نسبت تصویر (Aspect Ratio) را در هنگام تغییر اندازه حفظ کنید تا تصاویر دچار کشیدگی یا تغییر شکل نشوند.
- تصاویر رنگی را با استفاده از تابع `cv2.cvtColor()` به تصاویر خاکستری تبدیل کنید.
- تأثیر تغییر اندازه و تبدیل تصویر به خاکستری را در افزایش سرعت تشخیص چهره درک



در جلسه قبل یاد گرفتیم که چگونه یک تصویر را از پوشه رایانه بخوانیم، آن را نمایش دهیم و نسخه‌ای از آن را ذخیره کنیم. اما معمولاً در پروژه‌های واقعی بینایی ماشین، تصویر را بلافاصله برای تشخیص چهره استفاده نمی‌کنیم.

تصاویر ممکن است بسیار بزرگ باشند، رنگ‌های فراوان داشته باشند یا اطلاعاتی در خود داشته باشند که برای تشخیص چهره ضروری نیستند. برای مثال، اگر دوربین یک تصویر با کیفیت بالا و اندازه بسیار بزرگ تولید کند، رایانه باید میلیون‌ها پیکسل را بررسی کند. این کار می‌تواند سرعت برنامه را پایین بیاورد.

۱. چرا پردازش یک تصویر با اندازه بسیار بزرگ برای رایانه زمان بیشتری نیاز دارد؟
  ۲. اگر یک تصویر را فقط از سمت عرض بزرگ کنیم، چه تغییری در ظاهر چهره ایجاد می‌شود؟
  ۳. چرا در پروژه تشخیص چهره، رنگ لباس یا پس‌زمینه اهمیت زیادی ندارد؟
  ۴. به نظر شما چرا تبدیل تصویر رنگی به خاکستری می‌تواند باعث افزایش سرعت برنامه شود؟
- در این جلسه یاد می‌گیریم چگونه تصویر را برای پردازش بهتر آماده کنیم. دو مرحله مهمی که در پروژه تشخیص چهره استفاده می‌کنیم عبارت‌اند از:

۱. تغییر اندازه تصویر
  ۲. تبدیل تصویر رنگی به تصویر خاکستری
- این دو مرحله در پروژه نهایی ما بسیار مهم هستند؛ زیرا در جلسات آینده، پیش از جست‌وجوی چهره در هر فریم دوربین، ابتدا فریم را برای پردازش بهتر و سریع‌تر آماده می‌کنیم.



## پیش‌پردازش تصویر چیست؟

- پیش‌پردازش تصویر یعنی آماده‌سازی تصویر قبل از انجام عملیات اصلی مانند تشخیص چهره یا تشخیص اشیا. در پروژه تشخیص چهره، پیش از بررسی تصویر توسط الگوریتم، معمولاً تغییراتی روی آن انجام می‌دهیم تا:
- حجم داده‌ها کاهش پیدا کند.
  - سرعت پردازش افزایش یابد.
  - اطلاعات غیرضروری حذف شوند.
  - الگوریتم نتیجه دقیق‌تری تولید کند.

## تغییر اندازه تصویر چیست؟

هر تصویر دارای دو اندازه مهم است:

• عرض یا Width

• ارتفاع یا Height

برای مثال، یک تصویر ممکن است ۱۹۲۰ پیکسل عرض و ۱۰۸۰ پیکسل ارتفاع داشته باشد. این اندازه برای یک تصویر با کیفیت مناسب است، اما پردازش آن برای برنامه ممکن است زمان بیشتری نیاز داشته باشد.

با تغییر اندازه تصویر، می توانیم آن را کوچک تر یا بزرگ تر کنیم. در OpenCV برای این کار از تابع زیر استفاده می کنیم:

```
cv2.resize(image, (width, height))
```

در این دستور:

- image تصویر اصلی است.
- width عرض جدید تصویر است.
- height ارتفاع جدید تصویر است.

تصور کنید یک نقشه بسیار بزرگ از شهر در دست دارید. اگر بخواهید سریع مسیر کلی خانه تا مدرسه را پیدا کنید، لازم نیست همه جزئیات کوچک نقشه را ببینید. می توانید نقشه را کوچک تر کنید تا سریع تر آن را بررسی کنید. تغییر اندازه تصویر نیز همین کار را انجام می دهد: حجم داده ها را کم می کند تا برنامه سریع تر تصمیم بگیرد.

### حفظ نسبت تصویر

وقتی تصویر را تغییر اندازه می دهیم، باید مراقب باشیم که چهره افراد کشیده، پهن یا غیرطبیعی نشود. برای مثال، اگر یک عکس چهره را فقط از جهت عرض بزرگ کنیم اما ارتفاع آن را تغییر ندهیم، صورت فرد بیش از حد پهن دیده می شود. چنین تصویری ممکن است الگوریتم تشخیص چهره را دچار اشتباه کند. به نسبت بین عرض و ارتفاع تصویر، نسبت تصویر یا Aspect Ratio گفته می شود. برای حفظ این نسبت، ابتدا ضریب کوچک یا بزرگ شدن تصویر را حساب می کنیم و سپس ارتفاع جدید را با همان ضریب به دست می آوریم.

### تصویر خاکستری چیست؟

در جلسه چهارم یاد گرفتیم که یک تصویر رنگی در OpenCV معمولاً از سه کانال رنگی تشکیل می شود:

- آبی: B
- سبز: G
- قرمز: R

به این ترتیب رنگی، BGR می گوئیم.

پس در یک تصویر رنگی، هر پیکسل معمولاً سه عدد دارد. اما در تصویر خاکستری، هر پیکسل فقط یک عدد دارد که میزان روشنایی آن نقطه را نشان می دهد:

- ۰ یعنی سیاه
- ۲۵۵ یعنی سفید
- اعداد بین آن ها یعنی طیف های گوناگون خاکستری

برای تبدیل تصویر رنگی به خاکستری، از تابع cv2.cvtColor() استفاده می کنیم:

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

### چرا در تشخیص چهره از Grayscale استفاده می کنیم؟

در تشخیص چهره، الگوریتم بیشتر به شکل اجزای صورت، تفاوت روشنایی چشم‌ها، سایه‌های بینی، خطوط دهان و مرزهای چهره توجه می کند؛ نه به رنگ لباس یا رنگ پس زمینه. بنابراین، تبدیل تصویر به خاکستری چند مزیت مهم دارد:

۱. داده کمتری پردازش می شود.
  - تصویر رنگی سه کانال دارد، اما تصویر خاکستری یک کانال دارد.
  ۲. سرعت برنامه بیشتر می شود.
  - چون تعداد داده‌های بررسی شده کمتر است.
  ۳. الگوریتم روی شکل و نور تمرکز می کند.
  - در نتیجه، برای الگوریتم‌های کلاسیک تشخیص چهره مانند Haar Cascade مناسب تر است.
- مثال: کوچک کردن و خاکستری کردن تصویر
- در این مثال، فایل face.jpg را می خوانیم. سپس:
۱. اندازه آن را کوچک می کنیم.
  ۲. نسخه کوچک شده را به خاکستری تبدیل می کنیم.
  ۳. تصویر اصلی، تصویر کوچک شده و تصویر خاکستری را نمایش می دهیم.
  ۴. نسخه‌های جدید را ذخیره می کنیم.

```
import cv2
# خواندن تصویر
image = cv2.imread("face.jpg")
# بررسی وجود تصویر
if image is None:
    print("پیدا نشد face.jpg خطا: فایل")
else:
    # به دست آوردن ارتفاع و عرض تصویر اصلی
    height, width = image.shape[:2]
    # تعیین عرض جدید
    new_width = 600
    # محاسبه ضریب تغییر اندازه
    scale = new_width / width
    # محاسبه ارتفاع جدید با حفظ نسبت تصویر
    new_height = int(height * scale)
    # تغییر اندازه تصویر
    resized_image = cv2.resize(image, (new_width, new_height))
    # تبدیل تصویر کوچک شده به خاکستری
    gray_image = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
    # نمایش تصویرها
    cv2.imshow("Original Image", image)
    cv2.imshow("Resized Image", resized_image)
    cv2.imshow("Gray Image", gray_image)
    # ذخیره نسخه‌های جدید
    cv2.imwrite("face_resized.jpg", resized_image)
    cv2.imwrite("face_gray.jpg", gray_image)
    print("تصاویر جدید با موفقیت ذخیره شدند")
    # انتظار برای یک کلیک
    cv2.waitKey(0)
    # بستن پنجره‌ها
    cv2.destroyAllWindows()
```

برای اجرای این برنامه، باید فایل تصویری با نام face.jpg در همان پوشه‌ای باشد که فایل پایتون شما در آن قرار دارد. در این مثال، عرض جدید تصویر را ۶۰۰ پیکسل در نظر گرفته‌ایم. برنامه خودش ارتفاع مناسب را محاسبه می کند تا چهره داخل عکس کشیده یا فشرده نشود.

## فعالیت کلاسی

عنوان فعالیت: ساخت تصویر استاندارد برای تشخیص چهره

۱. یک تصویر از چهره یا یک تصویر معمولی در پوشه پروژه قرار دهید.
۲. نام فایل تصویر را از کاربر دریافت کنید.
۳. تصویر را با OpenCV بخوانید.
۴. اگر تصویر پیدا نشد، پیام خطا نمایش دهید.
۵. اگر تصویر پیدا شد، آن را به عرض ۵۰۰ پیکسل تغییر اندازه دهید.
۶. نسبت طول و عرض تصویر را حفظ کنید.
۷. تصویر کوچک شده را به حالت خاکستری تبدیل کنید.
۸. تصویر خاکستری را با نام prepared\_face.jpg ذخیره کنید.
۹. تصویر کوچک شده و خاکستری را نمایش دهید.

## خلاصه درس

- پیش پردازش تصویر برای آماده سازی تصاویر قبل از اجرای الگوریتم های هوش مصنوعی استفاده می شود.
- تغییر اندازه تصویر باعث کاهش حجم داده ها و افزایش سرعت پردازش می شود.
- هنگام تغییر اندازه تصویر باید نسبت تصویر (Aspect Ratio) حفظ شود تا شکل اشیا و چهره ها تغییر نکند.
- تصاویر رنگی در OpenCV با سه کانال BGR ذخیره می شوند.
- تصاویر خاکستری تنها یک کانال دارند و اطلاعات کمتری نسبت به تصاویر رنگی ذخیره می کنند.
- تبدیل تصویر رنگی به خاکستری با استفاده از cv2.cvtColor() باعث افزایش سرعت پردازش در پروژه های تشخیص چهره می شود.
- آماده سازی درست تصویر، یکی از مراحل مهم برای ساخت سیستم های بینایی ماشین دقیق و سریع است.

## تمرین

۱. پیش پردازش تصویر چیست؟

.....

۲. توضیح دهید چرا قبل از تشخیص چهره، تصویر را آماده می کنیم.

.....

۳. چرا بهتر است هنگام تغییر اندازه تصویر، نسبت عرض و ارتفاع را حفظ کنیم؟

.....

۴. تفاوت یک تصویر رنگی BGR با تصویر خاکستری چیست؟

.....

۵. اگر عرض یک تصویر ۱۲۰۰ پیکسل و ارتفاع آن ۸۰۰ پیکسل باشد و عرض جدید را ۶۰۰ در نظر بگیریم:

○ ضریب تغییر اندازه چند است؟.....

○ ارتفاع جدید چند پیکسل می‌شود؟.....



پروژه

برنامه‌ای بنویسید که فایل input.jpg را بخواند، آن را به حالت خاکستری تبدیل کند و با نام gray\_output.jpg ذخیره کند.



چالش

برنامه‌ای بنویسید که از کاربر عرض جدید تصویر را دریافت کند. اگر کاربر عددی کوچک‌تر یا مساوی صفر وارد کرد، پیام خطا نشان دهد و تصویر را تغییر اندازه ندهد.

## درس ۱۶: پنجره‌ای رو به دنیای واقعی؛ دوربین و فریم زنده با OpenCV



در پایان این درس خواهید توانست:

- مفهوم ویدئوی دیجیتال و ارتباط آن با مجموعه‌ای از تصاویر ثابت یا فریم‌ها را توضیح دهید.
- با استفاده از OpenCV به دوربین یا وب‌کم رایانه متصل شوید.
- فریم‌های زنده را به صورت پیوسته از دوربین دریافت و در یک پنجره نمایش دهید.
- از حلقه `while` برای دریافت و نمایش مداوم تصاویر دوربین استفاده کنید.
- با استفاده از کلید `q` برنامه را به صورت ایمن متوقف کرده و دوربین را آزاد کنید.

تا اینجا ما یاد گرفته ایم که چطور تصاویر ثابت را از حافظه سیستم بخوانیم، اندازه آن‌ها را تغییر دهیم و رنگشان را به خاکستری تبدیل کنیم. اما یک سیستم تشخیص چهره واقعی نمی‌تواند به تصاویر ثابت محدود باشد؛ این سیستم باید بتواند چهره افراد را به صورت زنده و مستقیم از طریق دوربین (وب‌کم) شناسایی کند. شاید در ابتدا فکر کنید پردازش ویدئوی زنده کار بسیار پیچیده و متفاوتی است. اما در این جلسه یاد می‌گیریم که ویدئو در واقع مجموعه‌ای از عکس‌های پشت سرهم است که با سرعت بالا نمایش داده می‌شوند. ما با یادگیری نحوه ارتباط با دوربین، اولین قدم بزرگ را برای تبدیل برنامه‌مان به یک سیستم بینایی ماشین بی‌وقفه (Real-time) برمی‌داریم.

۱. آیا می‌دانید یک ویدئو از چه بخش‌هایی تشکیل شده است؟

۲. چرا وقتی چند تصویر ثابت با سرعت بالا نمایش داده می‌شوند، آن‌ها را به شکل یک حرکت پیوسته می‌بینیم؟

### ویدئو چیست؟

آیا تا به حال با نقاشی کردن روی گوشه صفحات یک دفترچه و ورق زدن سریع آن، یک انیمیشن یا نقاشی متحرک ساخته‌اید؟ در این کار، هر صفحه یک نقاشی ثابت است، اما وقتی صفحات را با سرعت بالا ورق می‌زنید، چشمان شما آن را به صورت یک حرکت پیوسته و زنده می‌بیند. در دنیای دیجیتال نیز ویدئو دقیقاً به همین صورت کار می‌کند. به هر کدام از این تصاویر ثابت، یک فریم (Frame) گفته می‌شود. وقتی دوربین فیلم‌برداری می‌کند، در واقع دارد در هر ثانیه تعداد زیادی عکس (مثلاً ۳۰ عکس در ثانیه) می‌گیرد. پس برای اینکه یک ویدئوی زنده داشته باشیم، باید:

۱. به دوربین متصل شویم.

۲. در یک حلقه تکرار بی‌نهایت، عکس‌های جدید را یکی پس از دیگری با سرعت بالا دریافت کنیم.

۳. هر عکس (فریم) را بلافاصله روی صفحه نشان دهیم.



## ابزارهای کلیدی OpenCV برای کار با دوربین

برای تعامل با دوربین، کتابخانه OpenCV کلاس و متدهای زیر را در اختیار ما قرار می‌دهد:



| نکته مهم                                                                                                                          | نمونه کد                             | کاربرد                        | ابزار                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>عدد 0 دوربین پیش‌فرض است.</li> <li>برای دوربین‌های دیگر از 1 یا 2 استفاده کنید.</li> </ul> | <pre>cap = cv2.VideoCapture(0)</pre> | اتصال برنامه به دوربین سیستم  | <b>cv2.VideoCapture(0)</b>  <b>1</b> |
| <ul style="list-style-type: none"> <li>دوربین آماده → <b>True</b></li> <li>اتصال برقرار نشده → <b>False</b></li> </ul>            | <pre>cap.isOpened()</pre>            | بررسی آماده بودن دوربین       | <b>isOpened()</b>  <b>2</b>          |
| <ul style="list-style-type: none"> <li>موفقیت خواندن: <b>ret</b></li> <li>تصویر دریافتی: <b>frame</b></li> </ul>                  | <pre>ret, frame = cap.read()</pre>   | دریافت یک فریم جدید از دوربین | <b>read()</b>  <b>3</b>              |
| <ul style="list-style-type: none"> <li>پس از پایان کار همیشه اجرا شود.</li> <li>منابع سیستم آزاد می‌شوند.</li> </ul>              | <pre>cap.release()</pre>             | قطع ارتباط با دوربین          | <b>release()</b>  <b>4</b>           |

**نکته:**

ترتیب صحیح کار با دوربین در OpenCV به صورت زیر است:

 **VideoCapture(0)** →  **isOpened()** →  **read()** →  **release()**

## خروج امن با کلیدهای کیبورد

در جلسات قبلی برای توقف برنامه از `cv2.waitKey(1)` استفاده می کردیم که منتظر می ماند تا کاربر هر کلیدی را فشار دهد. اما در یک برنامه ویدئویی زنده، ما نمی توانیم برنامه را متوقف کنیم؛ برنامه باید مداوم فریم های جدید بگیرد. بنابراین، زمان انتظار را به مقدار بسیار کمی (مثلاً ۱ میلی ثانیه) تغییر می دهیم: `cv2.waitKey(1)`.

همچنین برای اینکه کاربر بتواند هر زمان که خواست برنامه را به طور منظم ببندد، شرطی می گذاریم که اگر کلید خاصی (مثل حرف 'q' روی کیبورد به معنی Quit) فشرده شد، حلقه دریافت فریم شکسته شده و برنامه به پایان برسد:

```
if cv2.waitKey(1) & 0xFF == ord('q') :
    break
```

مثال: نمایش ویدئوی زنده دوربین

در این برنامه، دوربین پیش فرض رایانه را روشن می کنیم، فریم های دریافتی را در یک پنجره به صورت زنده نمایش می دهیم و با فشردن کلید 'q' برنامه را به طور ایمن خاتمه می دهیم. برای اجرای این برنامه، مطمئن شوید که وب کم یا دوربین سیستم شما متصل است و برنامه دیگری (مثل تلگرام یا اسکایپ) در حال استفاده از دوربین نیست.

```
import cv2

# (0) اتصال به دوربین پیش فرض سیستم ۱.
cap = cv2.VideoCapture(0)

# بررسی اینکه آیا دوربین به درستی باز شده است
if not cap.isOpened():
    print("خطا: امکان دسترسی به دوربین وجود ندارد.")
else:
    print("فشار دهید 'q' دوربین فعال شد. برای خروج کلید")

    # حلقه بی نهایت برای دریافت مداوم فریم ها
    while True:
        # خواندن یک فریم از دوربین ۲.
        ret, frame = cap.read()

        # اگر در خواندن فریم مشکلی پیش آمد، حلقه را متوقف کن
        if not ret:
            print("خطا: دریافت فریم از دوربین متوقف شد.")
            break

        # نمایش فریم دریافتی در پنجره ۳.
        cv2.imshow("Live Video Feed", frame)

        # برای خروج (با تاخیر ۱ میلی ثانیه) 'q' بررسی فشردن کلید ۴.
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # آزاد کردن منابع و بستن پنجره ها ۵.
    cap.release()
    cv2.destroyAllWindows()
    print("دوربین غیرفعال شد و برنامه با موفقیت پایان یافت.")
```

## فعالیت کلاسی

عنوان فعالیت: آینه تصویر خاکستری (مای آینه)

هدف فعالیت: تلفیق مباحث پردازش تصویر (جلسه ۶) با جریان ویدئوی زنده دوربین (جلسه ۷).

برنامه‌ای بنویسند که دوربین را روشن کند و هم‌زمان دو پنجره به کاربر نشان دهد:

۱. پنجره اول: تصویر زنده و رنگی دوربین.

۲. پنجره دوم: تصویر زنده اما تبدیل شده به حالت خاکستری (Grayscale) و با اندازه کوچک‌تر (نصف اندازه اصلی).

کاربر باید بتواند با فشردن کلید "q" از برنامه خارج شود.

## خلاصه درس

- ویدئو از مجموعه‌ای از تصاویر ثابت به نام فریم تشکیل شده است.
- نمایش سریع و پیوسته فریم‌ها باعث می‌شود تصاویر به صورت متحرک دیده شوند.
- دوربین در هر ثانیه تعداد زیادی فریم ثبت و در اختیار برنامه قرار می‌دهد.
- برای دریافت تصاویر زنده از دوربین، از ابزار `cv2.VideoCapture()` استفاده می‌کنیم.
- برای دریافت و نمایش پیوسته فریم‌ها، از یک حلقه `while` استفاده می‌کنیم.
- تابع `(1) cv2.waitKey()` به برنامه اجازه می‌دهد بدون توقف، فریم‌های جدید را دریافت کند.
- با فشردن کلید `q` می‌توانیم حلقه دریافت تصویر را متوقف کنیم.
- پس از پایان برنامه، باید اتصال دوربین را آزاد و پنجره‌های `OpenCV` را ببندیم.

## تمرین

۱. ویدئو در دنیای دیجیتال چگونه ساخته و نمایش داده می‌شود؟

۲. آرگومان ورودی تابع `cv2.VideoCapture()` چه چیزی را مشخص می‌کند و مقدار ۰ به چه معناست؟

۳. در دستور `ret, frame = cap.read()`، متغیرهای `ret` و `frame` چه مقادیری را ذخیره می‌کنند؟

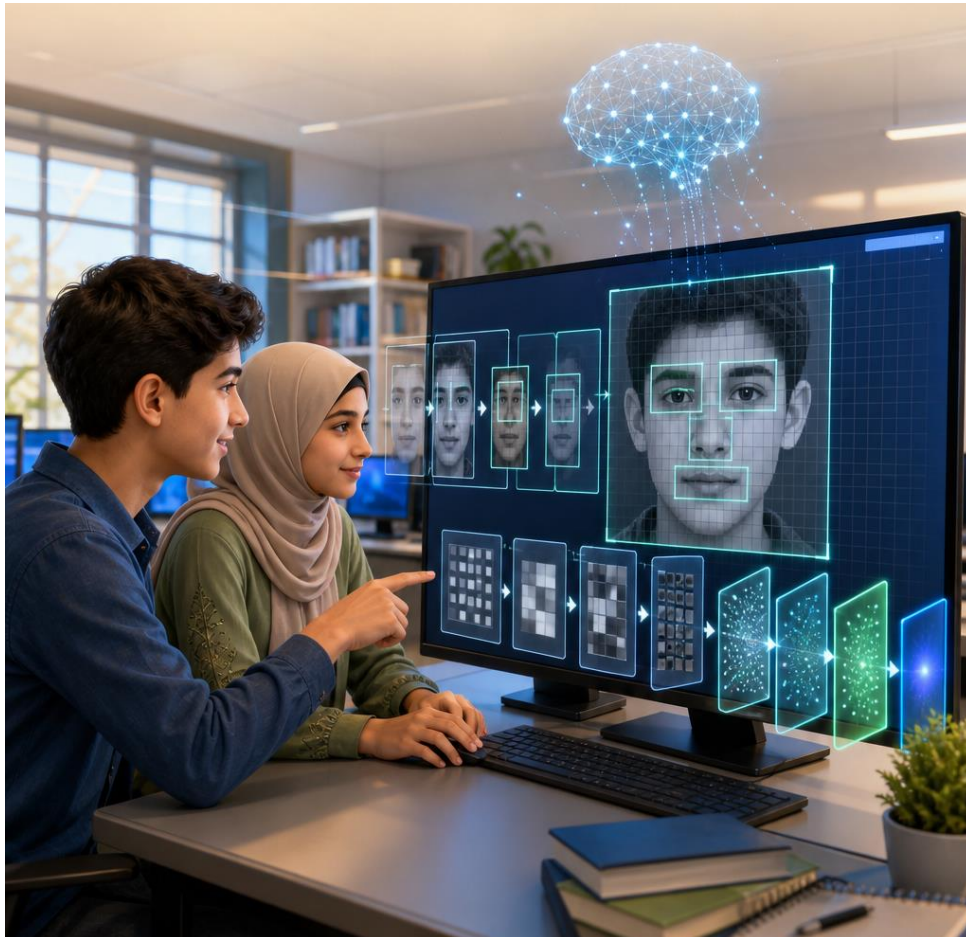
۴. چرا پس از پایان کار با دوربین، حتماً باید متد `cap.release()` را فراخوانی کنیم؟ اگر این کار را نکنیم چه مشکلی پیش می‌آید؟

## پروژه

برنامه‌ای بنویسید که تصویر دوربین را به صورت زنده دریافت کند و در صورتی که کاربر کلید `s` (مخفف Save) را فشار داد، تک‌فریم آن لحظه را با نام `snapshot.jpg` ذخیره کند و با فشردن کلید `q` خارج شود.

## درس ۱۷: مغز متفکر پروژه؛ آشنایی با الگوریتم Haar Cascade و

### تشخیص چهره



در پایان این درس خواهید توانست:

- مفهوم بینایی ماشین را توضیح دهید و چند نمونه از کاربردهای آن را نام ببرید.
- تفاوت میان تشخیص چهره و شناسایی چهره را بیان کنید.
- توضیح دهید که رایانه چگونه با بررسی داده‌های تصویری، بخش‌های شبیه به چهره را پیدا می‌کند.
- با الگوریتم Haar Cascade و نقش آن در تشخیص چهره آشنا شوید.
- تصویر رنگی را برای پردازش بهتر به تصویر خاکستری (Grayscale) تبدیل کنید.
- یک مدل آماده تشخیص چهره را در OpenCV بارگذاری کنید.
- با استفاده از تابع `detectMultiScale()`، محل چهره‌ها را در یک تصویر پیدا کنید.
- مفهوم مختصات  $x, y, w, h$  را در تعیین موقعیت و اندازه چهره درک کنید.



در جلسه های گذشته، یاد گرفتیم که چگونه یک تصویر را بخوانیم، نمایش دهیم، ذخیره کنیم، اندازه آن را تغییر دهیم و آن را به تصویر خاکستری تبدیل کنیم. همچنین در جلسه قبل، توانستیم با استفاده از OpenCV به دوربین رایانه متصل شویم و فریم های زنده را دریافت کنیم. اکنون یک پرسش مهم مطرح می شود:

رایانه از کجا می فهمد که در یک تصویر، چهره وجود دارد؟

انسان ها معمولاً با یک نگاه می توانند صورت افراد را تشخیص دهند. ما چشم ها، بینی، دهان، ابروها و شکل کلی صورت را می بینیم و خیلی سریع متوجه می شویم که با یک چهره روبه رو هستیم.

اما رایانه مانند انسان نمی بیند. از نگاه رایانه، تصویر فقط مجموعه بزرگی از عدد است. هر عدد، رنگ یا میزان روشنایی یک پیکسل را نشان می دهد. بنابراین، برای اینکه رایانه بتواند چهره را پیدا کند، باید روشی داشته باشد که این عددها را بررسی و الگوهای شبیه چهره را پیدا کند. در این جلسه، با یکی از الگوریتم های معروف OpenCV به نام Haar Cascade آشنا می شویم. این الگوریتم به برنامه کمک می کند تا در یک تصویر، محل تقریبی یک یا چند چهره را تشخیص دهد.



### بینایی ماشین چیست؟

بینایی ماشین یا بخشی از علم رایانه است که به برنامه ها کمک می کند تصویر و ویدئو را بررسی کنند و از آن ها اطلاعات به دست آورند.

برای مثال، یک سیستم بینایی ماشین می تواند:

- چهره افراد را در یک عکس پیدا کند.
- تعداد افراد موجود در تصویر را تخمین بزند.
- پلاک خودرو را تشخیص دهد.
- بارکد یا QR Code را بخواند.
- حرکت یک جسم را دنبال کند.
- در کارخانه، محصولات سالم و خراب را از هم جدا کند.
- در تصاویر پزشکی، بخش های مهم را بررسی کند.

در پروژه ما، هدف این است که برنامه بتواند پاسخ این سؤال را بدهد:

**آیا در این تصویر یا فریم دوربین، چهره ای وجود دارد؟ اگر بله، در کدام قسمت تصویر قرار گرفته است؟**

### تفاوت تشخیص چهره و شناسایی چهره

پیش از ادامه کار، لازم است دو مفهوم مهم را از هم جدا کنیم.

۱. تشخیص چهره

در تشخیص چهره، برنامه فقط محل چهره را در تصویر پیدا می کند.

برای مثال، برنامه ممکن است اعلام کند:

- یک چهره در تصویر پیدا شد.

- سه چهره در تصویر دیده می شود.
  - یک چهره از مختصات مشخصی شروع می شود.
- اما برنامه هنوز نمی داند این چهره متعلق به چه کسی است.

## ۲. شناسایی چهره

در شناسایی چهره، برنامه علاوه بر پیدا کردن چهره، تلاش می کند هویت فرد را نیز مشخص کند. برای مثال، ممکن است بگوید:

- این چهره متعلق به سارا است.
  - این فرد در سامانه ثبت شده است.
  - این چهره برای برنامه ناشناس است.
- نکته: در پروژه این دوره، هدف ما تشخیص چهره است، نه شناسایی هویت افراد. یعنی برنامه فقط چهره ها را پیدا می کند و در جلسه بعد دور آن ها کادر رسم خواهد کرد

## Haar Cascade چیست؟

Haar Cascade یک الگوریتم کلاسیک و شناخته شده در بینایی ماشین است که برای پیدا کردن الگوهای مشخص در تصویر استفاده می شود.

این الگوریتم می تواند برای تشخیص موارد گوناگونی به کار رود؛ مانند:

- چهره
- چشم
- لبخند
- بدن انسان
- برخی اجزای دیگر تصویر

در پروژه ما، از مدل آماده Haar Cascade برای پیدا کردن چهره های روبه رو استفاده می کنیم. Haar Cascade هنگام بررسی تصویر، به تفاوت روشنایی بخش های مختلف توجه می کند. برای مثال، در بسیاری از چهره ها:

- چشم ها نسبت به بخش هایی از گونه یا پیشانی تیره تر دیده می شوند.
- ناحیه بین دو چشم شکل خاصی دارد.
- بینی و اطراف آن الگوی روشنایی مشخصی ایجاد می کنند.
- فرم کلی صورت، ویژگی های قابل بررسی دارد.

الگوریتم با بررسی چنین الگوهای تلاش می کند تشخیص دهد که کدام قسمت تصویر به چهره شبیه است. تصور کنید در یک تصویر شلوغ به دنبال دوست خود می گردید. شما ابتدا همه جزئیات تصویر را با دقت یکسان بررسی نمی کنید. معمولاً به صورت مرحله ای عمل می کنید:

۱. ابتدا به دنبال شکل کلی سر و صورت انسان می گردید.

۲. سپس به چشم‌ها، بینی و دهان توجه می‌کنید.

۳. بعد بررسی می‌کنید که آیا این چهره می‌تواند متعلق به دوست شما باشد یا نه.

الگوریتم Haar Cascade نیز به صورت مرحله‌ای عمل می‌کند. ابتدا قسمت‌هایی از تصویر را که قطعاً چهره نیستند، سریع کنار می‌گذارد. سپس فقط قسمت‌های مشکوک را با دقت بیشتری بررسی می‌کند. به همین دلیل، واژه Cascade به معنی بررسی مرحله‌به‌مرحله یا زنجیره‌ای است.

### چرا تصویر خاکستری برای تشخیص چهره مناسب‌تر است؟

در جلسه ششم یاد گرفتیم که تصویر رنگی در OpenCV معمولاً سه کانال رنگ دارد:

- آبی یا B
- سبز یا G
- قرمز یا R

اما تصویر خاکستری فقط میزان روشنایی هر پیکسل را نگه می‌دارد.

الگوریتم Haar Cascade بیشتر به شکل کلی صورت و تفاوت میان بخش‌های روشن و تیره توجه می‌کند. بنابراین، لازم نیست همه اطلاعات رنگی تصویر را بررسی کند.

تبدیل تصویر به Grayscale چند فایده مهم دارد:

۱. حجم داده‌های تصویر کمتر می‌شود.
۲. پردازش تصویر سریع‌تر انجام می‌شود.
۳. الگوریتم روی شکل و روشنایی اجزای صورت تمرکز می‌کند.
۴. اجرای برنامه برای تشخیص چهره مناسب‌تر می‌شود.

برای تبدیل تصویر رنگی به خاکستری، از دستور زیر استفاده می‌کنیم:

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

### مدل تشخیص چهره چیست؟

Haar Cascade برای تشخیص چهره به یک فایل راهنما نیاز دارد. این فایل نتیجه آموزش الگوریتم با تعداد زیادی تصویر چهره و غیر چهره است. به این فایل، مدل یا طبقه‌بندی‌کننده (Classifier) گفته می‌شود. مدلی که در این جلسه استفاده می‌کنیم، نام زیر را دارد:

```
haarcascade_frontalface_default.xml
```

این نام چند بخش دارد:

- haar: مربوط به ویژگی های Haar
- cascade: بررسی مرحله ای
- frontalface: چهره از روبه رو
- default: مدل پیش فرض

پس این مدل برای تشخیص چهره هایی مناسب است که تقریباً رو به دوربین قرار دارند.

### فایل XML چه نقشی دارد؟

فایل XML شامل اطلاعاتی است که به الگوریتم می گوید چه الگوهایی می توانند مربوط به چهره باشند. ما لازم نیست فایل XML را خودمان طراحی کنیم. OpenCV چند مدل آماده در اختیار ما قرار می دهد. برای دسترسی به پوشه مدل های آماده OpenCV از عبارت زیر استفاده می کنیم و در مرحله بعد سپس نام فایل مدل را به این مسیر اضافه

```
cv2.data.harcascades
cv2.data.harcascades + "haarcascade_frontalface_default.xml"
```

می کنیم:

### تابع detectMultiScale() چیست؟

پس از بارگذاری مدل و تبدیل تصویر به خاکستری، نوبت به تشخیص چهره می رسد. برای این کار، از تابع زیر استفاده

```
faces = face_cascade.detectMultiScale(gray)
```

می کنیم:

این تابع تصویر خاکستری را بررسی می کند و بخش هایی را که احتمال دارد چهره باشند، پیدا می کند. کلمه MultiScale یعنی الگوریتم تصویر را در اندازه های گوناگون بررسی می کند. این موضوع مهم است؛ زیرا چهره ها ممکن است در تصویر اندازه های متفاوتی داشته باشند:

- یک فرد نزدیک دوربین است و چهره بزرگی دارد.
  - فرد دیگری دورتر است و چهره کوچک تری دارد.
  - در یک تصویر گروهی، چند چهره با اندازه های متفاوت دیده می شود.
- خروجی detectMultiScale() می تواند شامل هیچ چهره، یک چهره یا چند چهره باشد.

### مختصات چهره ها چگونه ذخیره می شود؟

برای هر چهره پیدا شده، الگوریتم چهار عدد تولید می کند: x, y, w, h معنای این مقادارها:

- X: فاصله گوشه بالای سمت چپ چهره از سمت چپ تصویر
- Y: فاصله گوشه بالای سمت چپ چهره از بالای تصویر
- W: عرض چهره
- h: ارتفاع چهره

```
x, y, w, h = 120, 80, 150, 150
```

یعنی:

- کادر چهره از نقطه (۸۰, ۱۲۰) شروع می شود.
- عرض آن ۱۵۰ پیکسل است.
- ارتفاع آن نیز ۱۵۰ پیکسل است.

### مثال: پیدا کردن چهره ها در یک تصویر

در این مثال، فایل face.jpg را می خوانیم، مدل Haar Cascade را بارگذاری می کنیم، تصویر را به خاکستری تبدیل می کنیم و در نهایت تعداد چهره ها و مختصات آن ها را در ترمینال چاپ می کنیم. برای اجرای این برنامه، به موارد زیر نیاز دارید:

۱. کتابخانه OpenCV باید نصب شده

باشد.

۲. یک تصویر دارای چهره با نام

face.jpg باید در پوشه پروژه قرار

داشته باشد.

۳. بهتر است چهره در تصویر واضح و

تقریباً رو به دوربین باشد.

```
import cv2
# بارگذاری مدل آماده تشخیص چهره
face_cascade = cv2.CascadeClassifier(
    cv2.data.haarcascades + "haarcascade_frontalface_default.xml"
)
# بررسی بارگذاری صحیح مدل
if face_cascade.empty():
    print("خطا: مدل تشخیص چهره بارگذاری نشد.")
else:
    # خواندن تصویر
    image = cv2.imread("face.jpg")
    # بررسی وجود فایل تصویر
    if image is None:
        print("پیدا نشد face.jpg: خطا: فایل.")
    else:
        # تبدیل تصویر رنگی به خاکستری
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        # تشخیص چهره ها
        faces = face_cascade.detectMultiScale(
            gray,
            scaleFactor=1.1,
            minNeighbors=5
        )
        # نمایش تعداد چهره های پیدا شده
        print("تعداد چهره های پیدا شده:", len(faces))
        # نمایش مختصات هر چهره
        for x, y, w, h in faces:
            print("مختصات چهره:")
            print("x =", x)
            print("y =", y)
            print("عرض =", w)
            print("ارتفاع =", h)
            print("-----")
        # نمایش تصویر اصلی و تصویر خاکستری
        cv2.imshow("Original Image", image)
        cv2.imshow("Gray Image", gray)
        # انتظار برای فشردن یک کلید
        cv2.waitKey(0)
        # بستن پنجره ها
        cv2.destroyAllWindows()
```

در این جلسه هنوز کادر روی چهره رسم نمی کنیم؛ فقط مختصات چهره را به دست می آوریم

### فعالیت کلاسی

عنوان فعالیت: شمارنده چهره در تصویر گروهی

شرح فعالیت:

۱. یک تصویر دارای یک یا چند چهره انتخاب کنید.

۲. فایل تصویر را با نام `class_photo.jpg` در پوشه پروژه قرار دهید.
۳. مدل تشخیص چهره را بارگذاری کنید.
۴. تصویر را بخوانید و به حالت خاکستری تبدیل کنید.
۵. با تابع `detectMultiScale()` چهره‌ها را پیدا کنید.
۶. تعداد چهره‌ها را در ترمینال چاپ کنید.
۷. برای هر چهره، مختصات شروع، عرض و ارتفاع را نمایش دهید.
۸. اگر هیچ چهره‌ای تشخیص داده نشد، پیام مناسب چاپ کنید.

### خلاصه درس

- بینایی ماشین به رایانه کمک می‌کند تصاویر و ویدئوها را بررسی کرده و از آن‌ها اطلاعات به دست آورد.
- رایانه تصویر را مانند انسان نمی‌بیند و آن را به مجموعه‌ای از داده‌ها و عددها تبدیل می‌کند.
- تشخیص چهره به معنی پیدا کردن محل چهره در تصویر است.
- شناسایی چهره علاوه بر پیدا کردن چهره، تلاش می‌کند هویت فرد را نیز مشخص کند.
- در این پروژه، هدف ما تشخیص چهره است، نه شناسایی هویت افراد.
- Haar Cascade یک الگوریتم شناخته شده برای پیدا کردن الگوهای مانند چهره، چشم و لبخند است.
- الگوریتم Haar Cascade به تفاوت میان بخش‌های روشن و تیره تصویر توجه می‌کند.
- تصویر خاکستری داده کمتری دارد و معمولاً سریع‌تر پردازش می‌شود.
- فایل XML اطلاعات مورد نیاز مدل تشخیص چهره را در اختیار الگوریتم قرار می‌دهد.
- تابع `cv2.CascadeClassifier()` برای بارگذاری مدل تشخیص چهره استفاده می‌شود.
- تابع `detectMultiScale()` بخش‌های شبیه به چهره را در تصویر پیدا می‌کند.
- مختصات هر چهره با چهار مقدار `x`، `y`، `w` و `h` مشخص می‌شود.
- این مختصات در جلسه بعد برای رسم کادر دور چهره به کار می‌روند.

### تمرین

۱. بین دو مفهوم زیر چه تفاوتی وجود دارد؟
  - تشخیص چهره (Face Detection)
  - شناسایی چهره (Face Recognition)

۲. چرا الگوریتم Haar Cascade معمولاً روی تصویر خاکستری اجرا می‌شود؟ حداقل دو دلیل بنویسید.

۳. فرض کنید مختصات یک چهره به شکل زیر باشد:

$$x, y, w, h = 50, 80, 120, 120$$

✓ به پرسش‌های زیر پاسخ دهید:

۱. گوشه بالای سمت چپ کادر چهره در چه مختصاتی قرار دارد؟

.....

۲. عرض کادر چند پیکسل است؟

.....

۳. ارتفاع کادر چند پیکسل است؟

.....

۴. گوشه پایین سمت راست کادر تقریباً در چه مختصاتی قرار می‌گیرد؟

.....

## درس ۱۸: رسم کادر و تشخیص چهره در ویدئوی زنده



در پایان این درس خواهید توانست:

- جریان ویدئویی زنده را با استفاده از دوربین رایانه دریافت کنید.
- مفهوم فریم و نقش آن را در تشکیل ویدئو توضیح دهید.
- با استفاده از حلقه `while True`، فریم‌های دوربین را به صورت پیوسته دریافت و پردازش کنید.
- هر فریم رنگی را به تصویر خاکستری تبدیل کنید.
- با استفاده از الگوریتم Haar Cascade، چهره‌ها را در فریم‌های زنده تشخیص دهید.
- مختصات چهره‌های شناسایی شده را برای رسم کادر به کار ببرید.

تا اینجا یاد گرفتیم چگونه با استفاده از الگوریتم Haar Cascade یک چهره را در یک عکس ثابت شناسایی کنیم. اما در دنیای واقعی، دوربین‌ها همیشه در حال فیلم برداری هستند.

آیا می‌توانیم همین کار را به صورت زنده انجام دهیم؟

در این جلسه یاد می‌گیریم چگونه جریان ویدئویی دوربین را دریافت کنیم، در هر لحظه چهره‌ها را تشخیص دهیم و بلافاصله دور آن‌ها کادر بکشیم.

- ویدئویی که در دوربین مشاهده می‌کنیم، از چه چیزی تشکیل شده است؟
- اگر دوربین در هر ثانیه چندین تصویر دریافت کند، چگونه یک تصویر متحرک ایجاد می‌شود؟
- چرا برای تشخیص چهره، تصویر را به حالت خاکستری تبدیل می‌کنیم؟
- چگونه می‌توانیم محل تشخیص داده‌شده چهره را روی تصویر نمایش دهیم؟

## رسم کادر و تشخیص چهره در ویدئوی زنده



برای تشخیص چهره در تصویر زنده، باید سه مفهوم کلیدی را درک کنیم:

- ❖ ویدئو به مثابه فریم‌ها: رایانه ویدئو را به صورت مجموعه‌ای از عکس‌های ثابت (فریم) می‌بیند که پشت سر هم نمایش داده می‌شوند. برنامه ما باید در هر ثانیه ده‌ها بار فریم‌ها را دریافت و پردازش کند.
- ❖ حلقه بی‌نهایت (while True): ما به یک حلقه نیاز داریم که تا وقتی دوربین روشن است، مدام عکس بگیرد، پردازش کند و نتیجه را روی صفحه نمایش دهد.
- ❖ پردازش خاکستری VS نمایش رنگی: الگوریتم تشخیص چهره روی تصویر خاکستری بسیار سریع‌تر و دقیق‌تر عمل می‌کند. اما چون ما دوست داریم کادرهای رنگی زیبا ببینیم، عملیات "تشخیص" را روی تصویر خاکستری انجام می‌دهیم و سپس کادر را روی "تصویر رنگی" اصلی رسم می‌کنیم.

```
import cv2

# بارگذاری فایل مدل تشخیص چهره ۱۰
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

# اتصال به دوربین ۲۰
cap = cv2.VideoCapture(0)

while True:
    # دریافت فریم از دوربین
    ret, frame = cap.read()
    if not ret:
        break

    # تبدیل به خاکستری برای تشخیص بهتر ۳۰
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # تشخیص چهره‌ها در فریم خاکستری ۴۰
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

    # رسم کادر برای هر چهره پیدا شده ۵۰
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)

    # نمایش خروجی ۶۰
    cv2.imshow('Real-time Face Detection', frame)

    # خروج یا زدن کلید 'q' ۷۰
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# آزاد کردن دوربین و بستن پنجره‌ها
cap.release()
cv2.destroyAllWindows()
```

تصور کنید دوربین یک «نقاش سریع» است.

۱. نقاش (دوربین) یک عکس از صحنه می گیرد.

۲. ما عکس را به خاکستری تبدیل می کنیم تا «مغز متفکر» (الگوریتم) سریع تر چهره را در آن پیدا کند.

۳. الگوریتم مختصات مثلاً  $y=100x$ ،  $50$  را به ما می دهد.

۴. ما حالا روی عکس اصلی (رنگی) در آن مختصات یک مستطیل سبز می کشیم.

این چرخه آنقدر سریع تکرار می شود (مثلاً ۳۰ بار در ثانیه) که چشم شما فکر می کند کادر همیشه دور چهره است.

### فعالیت کلاسی

آزمایش حرکت: برنامه را اجرا کنید و مقابل دوربین حرکت کنید. چقدر طول می کشد تا کادر شما را پیدا کند؟ (تأخیر را بسنجید).

تعداد افراد: از یک دوست بخواهید کنار شما قرار بگیرد. آیا برنامه همزمان دور هر دو نفر کادر می کشد؟

خطای دید: برنامه را اجرا کنید و با یک ماسک یا عینک آفتابی امتحان کنید. آیا هنوز هم شناسایی می شوید؟ چرا؟

### خلاصه درس

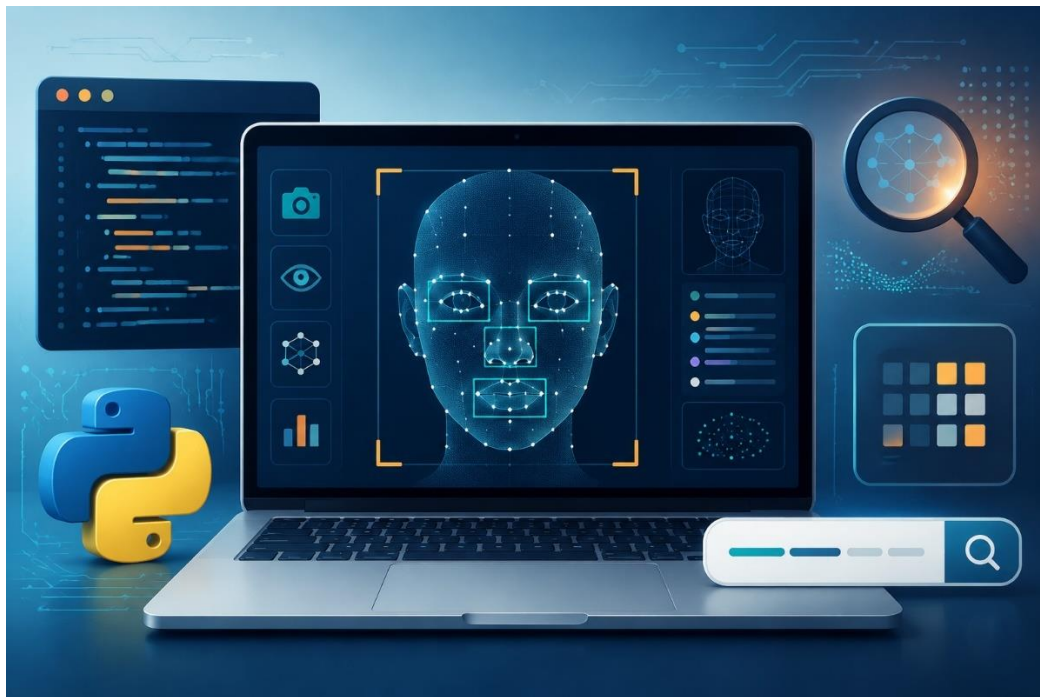
- ویدئو از مجموعه ای از تصاویر ثابت به نام فریم تشکیل شده است.
- دوربین در هر ثانیه چندین فریم دریافت می کند.
- برای پردازش پیوسته فریم ها، از حلقه `while True` استفاده می کنیم.
- هر فریم ابتدا از دوربین دریافت می شود.
- تصویر رنگی برای نمایش نتیجه نگهداری می شود.
- تصویر خاکستری برای اجرای الگوریتم تشخیص چهره استفاده می شود.
- الگوریتم Haar Cascade چهره ها را در تصویر خاکستری پیدا می کند.
- مختصات هر چهره با مقدارهای  $x$ ،  $y$ ،  $w$  و  $h$  مشخص می شود.
- با استفاده از تابع `cv.rectangle()` می توان دور چهره ها کادر رسم کرد.
- رنگ کادر در `OpenCV` با ترتیب `BGR` مشخص می شود.
- حلقه `for` برای رسم کادر دور همه چهره های شناسایی شده استفاده می شود.
- با فشردن کلید `q` می توان برنامه را متوقف کرد.
- پس از پایان برنامه، باید دوربین را با `camera.release()` آزاد کنیم.
- با استفاده از `cv.destroyAllWindows()` پنجره های `OpenCV` بسته می شوند.



### پروژه

۱. تغییر رنگ: رنگ کادر را از سبز به قرمز (۰, ۰, ۲۵۵) تغییر دهید.
۲. ضخامت خط: ضخامت کادر را که در کد عدد ۳ است، به ۱۰ تغییر دهید تا کادر ضخیم تر شود.
۳. نمایش متن: با استفاده از دستور `putText` cv، کاری کنید که بالای کادر هر نفر عبارت "Face" نمایش داده شود.
۴. تغییر پارامتر: مقدار `minNeighbors` را به ۲ تغییر دهید. چه اتفاقی می افتد؟ آیا کادرهای اشتباهی بیشتری ظاهر می شوند؟ (این پارامتر دقت مدل را کنترل می کند).

## درس ۱۹: جستجوی هوشمند؛ تشخیص اجزای صورت



در پایان این درس خواهید توانست:

- مفهوم ناحیه مورد علاقه یا ROI (Region of Interest) را توضیح دهید.
- ابتدا چهره را در تصویر پیدا کرده و سپس جستجوی چشم‌ها را به محدوده چهره محدود کنید.
- از مختصات چهره برای جدا کردن بخشی از تصویر استفاده کنید.
- با استفاده از مدل‌های آماده OpenCV، چشم‌ها را در محدوده چهره تشخیص دهید.
- تفاوت جست‌وجو در کل تصویر و جست‌وجو در یک ناحیه مشخص را بیان کنید.
- با محدود کردن محدوده جست‌وجو، سرعت و دقت برنامه را افزایش دهید.



تا اینجا آموختیم چگونه کل یک «چهره» را شناسایی کنیم. اما سیستم‌های هوشمند واقعی (مانند سیستم تشخیص خواب‌آلودگی راننده) نیاز دارند جزئیات دقیق‌تری ببینند. در این جلسه یاد می‌گیریم که نباید همیشه «همه چیز را در کل تصویر» جستجو کرد.

۱. اگر بخواهید یک وسیله کوچک را در اتاق پیدا کنید، آیا همه بخش‌های خانه را جست‌وجو می‌کنید؟

۲. به نظر شما، برای پیدا کردن چشم‌ها بهتر است کل تصویر بررسی شود یا فقط محدوده چهره؟



## ناحیه مورد علاقه یا ROI چیست؟

برای تشخیص چشم‌ها، ما از مفهوم «ناحیه مورد علاقه» استفاده می‌کنیم:

- مشکل جستجوی سراسری: اگر از رایانه بخواهیم در کل تصویر به دنبال چشم بگردد، احتمال اینکه اشیاء دیگر (مثل طرح روی لباس یا سایه دیوار) را به اشتباه «چشم» تشخیص دهد، بسیار زیاد است.

- راه حل (ROI): ما ابتدا چهره را پیدا می‌کنیم. سپس یک «قیچی مجازی» برمی‌داریم و فقط همان مستطیل چهره را از تصویر اصلی جدا می‌کنیم. حالا به رایانه می‌گوییم: «فقط داخل این مستطیل کوچک (ROI) به دنبال چشم بگرد». این کار باعث می‌شود رایانه فقط روی ویژگی‌های صورت تمرکز کند.

ما یاد می‌گیریم چگونه با محدود کردن محدوده جستجو، دقت و سرعت برنامه خود را به شدت افزایش دهیم.

تصور کنید می‌خواهید در یک کتابخانه بزرگ (تصویر اصلی)، یک لغت خاص را پیدا کنید.

- روش غیر بهینه: تمام کتاب‌های کل کتابخانه را ورق بزنید (بسیار کند و پرهزینه).

- روش بهینه (ROI): ابتدا بخش «کتاب‌های علوم» (چهره) را پیدا می‌کنید، سپس فقط همان یک کتاب (ROI) را از قفسه برمی‌دارید و در آن به دنبال لغت می‌گردید. این روش بسیار سریع و دقیق است.

```
import cv2

# بارگذاری مدل‌های چهره و چشم
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret: break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # تشخیص چهره
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:
        # رسم کادر آبی دور صورت
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

        # تعریف ROI (برش محدوده چهره)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]

        # تشخیص چشم فقط داخل محدوده چهره
        eyes = eye_cascade.detectMultiScale(roi_color)

        for (ex, ey, ew, eh) in eyes:
            # ROI نسبت به رسم کادر قرمز دور چشم
            cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 0, 255), 2)

    cv2.imshow('ROI Eye Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

## فعالیت کلاسی

۱. **تست چشمک:** برنامه را اجرا کنید و مقابل دوربین چشمک بزنید. آیا برنامه همچنان چشم باز شما را تشخیص می دهد؟
۲. **تست هم تیمی:** از یک دوست بخواهید کنار شما قرار بگیرد. آیا برنامه می تواند چشم های هر دو نفر را به طور جداگانه شناسایی کند؟
۳. **تغییر فاصله:** به آرامی از دوربین دور شوید. در چه فاصله ای برنامه دیگر نمی تواند چشم ها را پیدا کند؟ (این تست محدودیت های مدل را نشان می دهد).

## تمرین

۱. **پیام هشدار:** با استفاده از دستور `if len(eyes) == 0`، کاری کنید که اگر در محدوده چهره چشمی پیدا نشد، در ترمینال پیام «چشم ها شناسایی نشدند!» چاپ شود.
۲. **رنگ بندی:** رنگ کادر صورت را به زرد (۰, ۲۵۵, ۲۵۵) و رنگ کادر چشم ها را به بنفش (۰, ۲۵۵, ۲۵۵) تغییر دهید.
۳. **تغییر ضخامت:** در دستور `cv2.rectangle` مربوط به چشم ها، ضخامت خط را به عدد ۱ کاهش دهید تا کادرها ظریف تر باشند.

## خلاصه درس

- اچیه مورد علاقه یا ROI بخشی از تصویر است که اطلاعات مهم مورد نیاز برنامه در آن قرار دارد.
- برای تشخیص چشم، ابتدا باید چهره در تصویر شناسایی شود.
- پس از تشخیص چهره، فقط محدوده مربوط به چهره برای جست و جوی چشم ها بررسی می شود.
- استفاده از ROI باعث کاهش حجم پردازش و افزایش سرعت برنامه می شود.
- محدود کردن محدوده جست و جو، احتمال تشخیص اشتباه را کاهش می دهد.
- مختصات چهره با چهار مقدار `x`، `y`، `w` و `h` مشخص می شود.
- مختصات چشم ها نسبت به محدوده چهره محاسبه می شوند.
- برای رسم کادر چشم روی تصویر اصلی، باید مختصات چشم و مختصات چهره با یکدیگر ترکیب شوند.
- مدل های Haar Cascade می توانند برای تشخیص اجزای مختلف صورت، مانند چشم ها، استفاده شوند.
- تشخیص چشم می تواند پایه ای برای ساخت سیستم هایی مانند تشخیص خواب آلودگی راننده باشد.

## پروژه

### سامانه هوشمند تشخیص چهره و چشم

برنامه ای با استفاده از کتابخانه OpenCV طراحی کنید که بتواند به دوربین رایانه متصل شود و تصویر زنده را دریافت کند. برنامه باید چهره افراد را با استفاده از الگوریتم Haar Cascade تشخیص دهد و دور هر چهره یک کادر رسم کند. سپس با استفاده از مفهوم ناحیه مورد علاقه (ROI)، فقط محدوده چهره را بررسی کرده و چشم ها را تشخیص دهد. دور چشم های شناسایی شده نیز کادر رسم شود. برنامه باید با فشردن کلید `q` به صورت ایمن بسته شود.

## درس ۲۰: آشنایی با چاپ سه بعدی



در پایان این درس خواهید توانست:

- چاپ سه بعدی را به زبان ساده توضیح دهید.
- تفاوت پرینتر معمولی و پرینتر سه بعدی را بیان کنید.
- چند کاربرد چاپ سه بعدی را در زندگی روزمره نام ببرید.
- مراحل کلی ساخت یک قطعه با پرینتر سه بعدی را توضیح دهید.
- اهمیت طراحی سه بعدی در ساخت قطعات را درک کنید.

امروزه فناوری‌های نوین، روش ساخت بسیاری از وسایل را تغییر داده‌اند. یکی از این فناوری‌ها چاپ سه‌بعدی است. با استفاده از پرینتر سه‌بعدی می‌توان بسیاری از وسایل مورد نیاز را تنها با طراحی یک مدل در رایانه تولید کرد. چاپ سه‌بعدی در صنایع مختلف مانند پزشکی، معماری، خودروسازی، آموزش و حتی ساخت اسباب‌بازی کاربرد فراوانی دارد. آشنایی با این فناوری، نخستین گام برای ورود به دنیای طراحی و ساخت دیجیتال است.

۱. آیا تاکنون وسیله‌ای را دیده‌اید که با پرینتر سه‌بعدی ساخته شده باشد؟
۲. به نظر شما پرینتر سه‌بعدی چگونه یک جسم واقعی را تولید می‌کند؟
۳. اگر بتوانید هر وسیله‌ای را طراحی و چاپ کنید، دوست دارید چه چیزی بسازید؟
۴. پرینتر معمولی چه تفاوتی با پرینتر سه‌بعدی دارد؟

### چاپ سه‌بعدی چیست؟

چاپ سه‌بعدی روشی برای ساخت اجسام واقعی از روی یک مدل دیجیتالی است. در این روش، پرینتر جسم را لایه به لایه می‌سازد تا در پایان، قطعه کامل ایجاد شود. در پرینترهای معمولی، تنها جوهر روی کاغذ قرار می‌گیرد؛ اما پرینتر سه‌بعدی از مواد پلاستیکی مخصوص استفاده می‌کند و یک جسم واقعی تولید می‌کند. به همین دلیل به این فناوری ساخت افزایشی (Additive Manufacturing) نیز گفته می‌شود؛ زیرا قطعه با اضافه شدن لایه‌های متوالی ساخته می‌شود.

### تاریخچه چاپ سه‌بعدی

در ابتدا، چاپ سه‌بعدی با عنوان نمونه‌سازی سریع<sup>۳</sup> شناخته می‌شد اصطلاحی که امروزه هنوز گاهی استفاده می‌شود، البته به ندرت. پیش از اینکه پرینترهای سه‌بعدی مقرون به صرفه در دسترس عموم قرار گیرند، این فناوری تنها برای ساخت نمونه‌های اولیه به کار می‌رفت.

یکی از کاربردهای معمول نمونه‌سازی سریع، طراحی و تولید یک دورکنترل تلویزیون بود. هزینه‌های مقدماتی برای تولید انبوه مانند ساخت قالب‌ها، فرآیند تولید خود، آزمایش‌ها و غیره می‌تواند به ده‌ها هزار دلار برسد. بنابراین، تولیدکننده باید کاملاً مطمئن باشد که دورکنترل به‌خوبی در دست کاربر قرار می‌گیرد و تمام دکمه‌ها به راحتی در دسترس هستند. در اینجا نمونه‌سازی وارد عمل می‌شود. اگرچه قیمت پرینترهای سه‌بعدی در آن زمان بسیار بالا بود، اما هزینه تولید یک نمونه واحد با روش‌های قدیمی حدود هزار دلار بود که باز هم صرفه‌جویی قابل توجهی محسوب می‌شد. با این حال، به دلیل هزینه بالای دستگاه‌ها، هیچ امیدی برای ورود آن‌ها به دست کاربران عادی وجود نداشت خوشبختانه این وضعیت تغییر کرد.

<sup>۳</sup> Rapid Prototyping



## آیا می دانید؟

چاپ سه بعدی به شکلی که امروزه آن را می شناسیم، در سال ۱۹۸۴ کشف شد، زمانی که چارلز دبلیو. هال بنیان گذار شرکت ۳D Systems، برای اختراع خود **استریولیتوگرافی** درخواست ثبت اختراع داد. هال اولین کسی بود که داده های دیجیتال سه بعدی را به صورت فیزیکی چاپ کرد. این فناوری که معمولاً به اختصار SLA نامیده می شود، تا به امروز نیز مورد استفاده قرار می گیرد. در بخش مربوط به روش های مختلف چاپ سه بعدی می توانید اطلاعات بیشتری درباره آن کسب کنید.

## پرینتر سه بعدی چگونه کار می کند؟

برای ساخت یک قطعه با پرینتر سه بعدی، معمولاً مراحل زیر انجام می شود:

۱. ایده پردازی

ابتدا مشخص می کنیم چه وسیله ای قرار است ساخته شود.

۲. طراحی سه بعدی

مدل وسیله در نرم افزارهای طراحی مانند تینکرکد (Tinkercad) طراحی می شود.

۳. آماده سازی فایل

مدل طراحی شده به نرم افزار مخصوص چاپ منتقل می شود تا برای چاپ آماده شود.

۴. چاپ قطعه

پرینتر با ذوب کردن فیلامنت و قرار دادن آن به صورت لایه لایه، قطعه را می سازد.



## کاربردهای چاپ سه بعدی

چاپ سه بعدی در بسیاری از زمینه ها استفاده می شود، از جمله:

- ساخت نمونه اولیه محصولات
- تولید قطعات صنعتی
- ساخت وسایل پزشکی
- طراحی ماکت های معماری
- تولید قطعات آموزشی
- ساخت اسباب بازی و وسایل تزئینی
- ساخت قطعات یدکی

## مزایای چاپ سه بعدی

چاپ سه بعدی مزایای فراوانی دارد:

- ساخت سریع نمونه ها
- کاهش هزینه تولید نمونه اولیه
- امکان ساخت قطعات سفارشی
- کاهش دورریز مواد
- تبدیل ایده به محصول واقعی در مدت کوتاه



## فعالیت کلاسی



به تصویر یک پرینتر سه بعدی نگاه کنید و حدس بزنید هر قسمت چه وظیفه ای دارد. (در درس بعد با اجزای آن آشنا خواهید شد.)

## آیا می دانید؟



- پرینترهای سه بعدی می توانند هزاران قطعه مختلف را تنها با تغییر فایل طراحی تولید کنند.
- بسیاری از مدارس، دانشگاه ها و کارخانه ها از پرینتر سه بعدی برای آموزش و ساخت نمونه های اولیه استفاده می کنند.
- امروزه حتی برخی قطعات پزشکی و دندان پزشکی نیز با فناوری چاپ سه بعدی ساخته می شوند.

## خلاصه درس



- چاپ سه بعدی روشی برای ساخت اجسام واقعی از روی یک مدل دیجیتالی است.
- پرینتر سه بعدی قطعه را به صورت لایه لایه تولید می کند.
- طراحی سه بعدی نخستین مرحله ساخت یک قطعه است.
- چاپ سه بعدی در صنایع، پزشکی، آموزش و بسیاری از مشاغل کاربرد دارد.
- این فناوری امکان ساخت سریع و دقیق بسیاری از قطعات را فراهم می کند.

## تمرین



۱. چاپ سه بعدی را به زبان خود تعریف کنید.

.....

۲. دو تفاوت بین پرینتر معمولی و پرینتر سه بعدی را بنویسید.

.....

۳. سه کاربرد چاپ سه بعدی را نام ببرید.

.....

۴. مراحل کلی ساخت یک قطعه با پرینتر سه بعدی را به ترتیب بنویسید.

.....

۵. به نظر شما مهم ترین مزیت چاپ سه بعدی چیست؟ دلیل خود را بنویسید.

.....

## درس ۲۱: آشنایی با اجزای پرینتر سه بعدی و فرآیند چاپ



در پایان این درس خواهید توانست:

- با اجزای اصلی پرینتر سه بعدی آشنا شوید.
- با مراحل آماده سازی و فرایند چاپ سه بعدی آشنا شوید.

در درس قبل با مفهوم چاپ سه بعدی و کاربردهای آن آشنا شدید. اما پرینتر سه بعدی چگونه یک طرح دیجیتالی را به یک جسم واقعی تبدیل می کند؟

برای پاسخ به این پرسش، ابتدا باید با اجزای اصلی پرینتر و وظیفه هر بخش آشنا شویم. شناخت این قسمت ها به ما کمک می کند هنگام طراحی و چاپ، عملکرد دستگاه را بهتر درک کنیم و از آن به درستی استفاده کنیم.

- به نظر شما کدام قسمت پرینتر، قطعه را می سازد؟
- ماده اولیه پرینتر سه بعدی چیست؟
- اگر هنگام چاپ، مواد اولیه تمام شود، چه اتفاقی می افتد؟
- چرا پرینتر باید قطعه را لایه به لایه بسازد؟

### اجزای اصلی پرینتر سه بعدی

پرینتر سه بعدی از بخش های مختلفی تشکیل شده است که هر کدام وظیفه مشخصی دارند.

#### ❖ صفحه چاپ (Print Bed)

صفحه چاپ محلی است که قطعه روی آن ساخته می شود. این صفحه باید کاملاً تراز و تمیز باشد تا قطعه هنگام چاپ از جای خود جدا نشود.

#### ❖ نازل (Nozzle)

نازل قطعه ای فلزی است که فیلامنت ذوب شده از آن خارج می شود. قطر نازل معمولاً بسیار کوچک است تا بتواند لایه های نازک و دقیق ایجاد کند.

#### ❖ اکسترودر (Extruder)

اکسترودر فیلامنت را به سمت نازل هدایت می کند و مقدار خروج مواد را کنترل می کند.

#### ❖ فیلامنت (Filament)

فیلامنت رشته ای از جنس پلاستیک مخصوص است که ماده اولیه چاپ را تشکیل می دهد. این رشته وارد اکسترودر شده و پس از ذوب شدن از نازل خارج می شود.

#### ❖ محوره های حرکتی X، Y و Z

پرینتر برای ساخت قطعه در سه جهت حرکت می کند:

- محور X حرکت به چپ و راست
- محور Y حرکت به جلو و عقب
- محور Z حرکت به بالا و پایین

حرکت هماهنگ این سه محور باعث می شود قطعه با دقت ساخته شود.

## انواع رایج فیلامنت

### PLA

- رایج ترین فیلامنت آموزشی
- چاپ آسان
- مناسب برای مدارس و مبتدیان
- دوستدار محیط زیست نسبت به بسیاری از پلاستیک ها

### PETG

- مقاومت بیشتر
- استحکام مناسب
- مناسب قطعات کاربردی

### ABS

- مقاومت بالا در برابر گرما
- چاپ آن دشوارتر است و بیشتر در صنعت استفاده می شود.

## مراحل آماده سازی چاپ

قبل از شروع چاپ، باید مدل سه بعدی آماده شود.

مراحل کلی عبارت اند از:

۱. طراحی مدل در نرم افزار طراحی سه بعدی
۲. ذخیره فایل با فرمت STL
۳. وارد کردن فایل به نرم افزار آماده سازی چاپ مانند تینکرکد
۴. تنظیم اندازه و محل قرار گیری قطعه
۵. تبدیل فایل به دستورهای قابل فهم برای پرینتر
۶. آغاز چاپ

## نکات ایمنی

هنگام کار با پرینتر سه بعدی باید نکات زیر رعایت شود:

- هنگام چاپ به نازل دست نزنید؛ زیرا بسیار داغ است.
- هنگام حرکت دستگاه، دست خود را وارد قسمت های متحرک نکنید.
- از کشیدن یا جدا کردن قطعه قبل از پایان چاپ خودداری کنید.

- دستگاه باید روی سطح صاف و محکم قرار گیرد.
- پس از پایان کار، محیط اطراف پرینتر را مرتب نگه دارید.

### فعالیت کلاسی

در گروه‌های دو نفره درباره وظیفه هر یک از اجزای پرینتر گفت و گو کنید و نتیجه را برای کلاس توضیح دهید.

### خلاصه درس

- پرینتر سه بعدی از چند بخش اصلی تشکیل شده است.
- نازل، فیلامنت را ذوب کرده و به صورت لایه لایه روی صفحه چاپ قرار می دهد.
- محوره‌های X، Y و Z حرکت دقیق پرینتر را کنترل می کنند.
- فیلامنت ماده اولیه چاپ سه بعدی است.
- قبل از چاپ، مدل سه بعدی باید برای پرینتر آماده سازی شود.
- رعایت نکات ایمنی هنگام کار با پرینتر ضروری است.

### تمرین

۱. وظیفه نازل در پرینتر سه بعدی چیست؟

.....

۲. فیلامنت چیست و چه کاربردی دارد؟

.....

۳. سه مرحله از آماده سازی فایل برای چاپ را نام ببرید.

.....

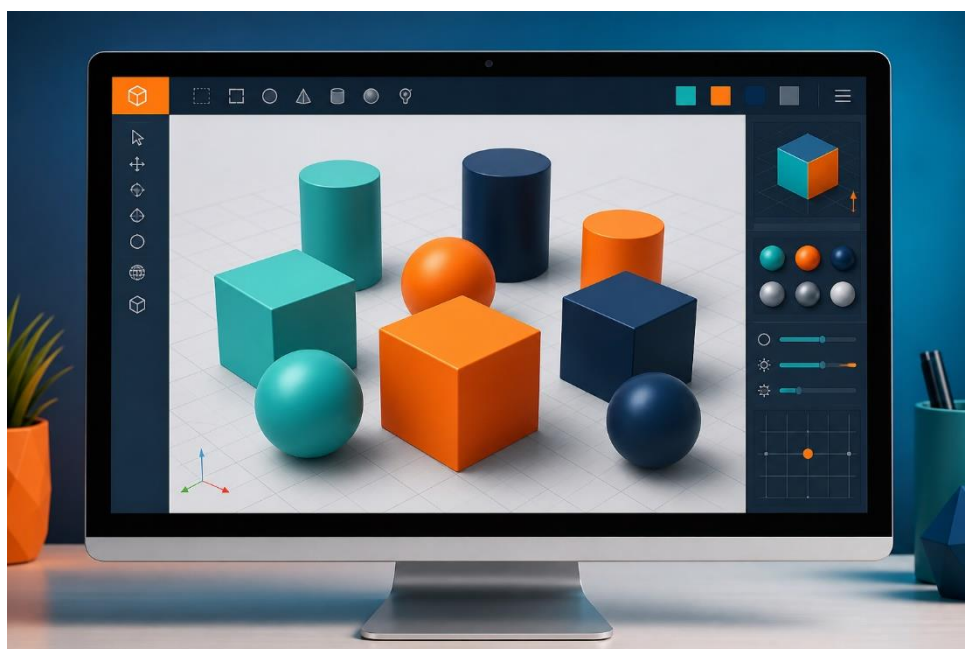
### پروژه

یک پرینتر سه بعدی را از طریق اینترنت یا آزمایشگاه مدرسه مشاهده کنید و گزارشی کوتاه تهیه کنید.

در گزارش خود به پرسش‌های زیر پاسخ دهید:

۱. نام پرینتر چیست؟
۲. از چه نوع فیلامنتی استفاده می کند؟
۳. مهم ترین اجزای آن کدام اند؟
۴. این پرینتر بیشتر برای چه کاربردی مناسب است؟

## درس ۲۲: طراحی سه بعدی با نرم افزار Tinkercad



در پایان این درس خواهید توانست:

- محیط نرم افزار Tinkercad را بشناسید.
- اشکال سه بعدی ساده را در صفحه طراحی قرار دهید.
- اندازه، محل و جهت اشکال را تغییر دهید.
- با ابزارهای Group و Hole مدل های جدید ایجاد کنید.
- یک مدل ساده را طراحی کرده و با فرمت STL ذخیره کنید.



پس از آشنایی با پرینتر سه بعدی، اکنون زمان آن رسیده است که خودمان یک مدل سه بعدی طراحی کنیم. برای این کار از نرم افزار Tinkercad استفاده می کنیم. این نرم افزار محیطی ساده و کاربردی دارد و به کاربران کمک می کند بدون نیاز به برنامه نویسی، مدل های سه بعدی مختلف را طراحی کنند. بسیاری از دانش آموزان، طراحان و علاقه مندان به چاپ سه بعدی، نخستین تجربه طراحی خود را با این نرم افزار آغاز می کنند.

- اگر بخواهید یک جاکلیدی برای خود طراحی کنید، از چه شکل هایی استفاده می کنید؟
- آیا می توان با کنار هم قرار دادن چند شکل ساده، یک وسیله جدید ساخت؟
- برای اینکه پرینتر بتواند یک مدل را چاپ کند، ابتدا چه کاری باید انجام دهیم؟
- اگر بخواهید روی یک قطعه نام خود را بنویسید، چگونه این کار را انجام می دهید؟

### نرم افزارهای طراحی سه بعدی



امروزه طیف گسترده ای از برنامه های کاربردی برای طراحی و مدل سازی سه بعدی در دسترس کاربران قرار دارد که هر یک با توجه به سطح نیاز و مهارت، قابلیت های متفاوتی را ارائه می دهند:

- ابزارهای ساده و مبتدی: برخی برنامه ها مانند «تینک کد» (Tinkercad)، بسیار ساده و زودآموز هستند و اغلب به صورت تحت وب اجرا می شوند؛ این ابزارها برای کسانی که تازه قصد ورود به دنیای چاپ سه بعدی را دارند، گزینه ای بسیار مناسب محسوب می شوند.
- مدل سازی پارامتریک: برای کاربرانی که به طراحی بر مبنای محاسبات و پارامترهای هندسی علاقه مند هستند، نرم افزاری نظیر «اوپن اسکد» (OpenSCAD) امکانات ویژه ای فراهم می کند.
- ابزارهای حرفه ای: برای پروژه های پیچیده مهندسی و صنعتی، استفاده از برنامه های قدرتمندی مانند «اتودسک فیوژن ۳۶۰» (Autodesk Fusion ۳۶۰) پیشنهاد می شود که ابزارهای کامل و دقیقی را در اختیار طراحان قرار می دهد.

### نرم افزار تینک رکد چیست؟

Tinkercad یک نرم افزار طراحی سه بعدی آنلاین است که توسط شرکت Autodesk توسعه یافته است. در این نرم افزار، مدل ها با استفاده از اشکال هندسی مانند مکعب، استوانه، کره و منشور ساخته می شوند. با تغییر اندازه، جابه جایی و ترکیب این اشکال می توان مدل های گوناگونی طراحی کرد.

## آشنایی با محیط نرم افزار

پس از ورود به نرم افزار، بخش های زیر را مشاهده می کنید.

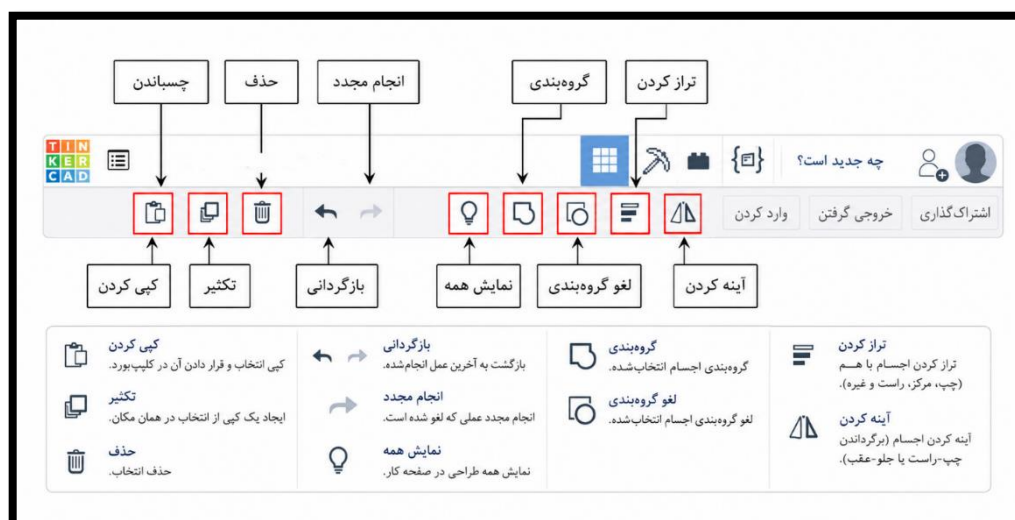
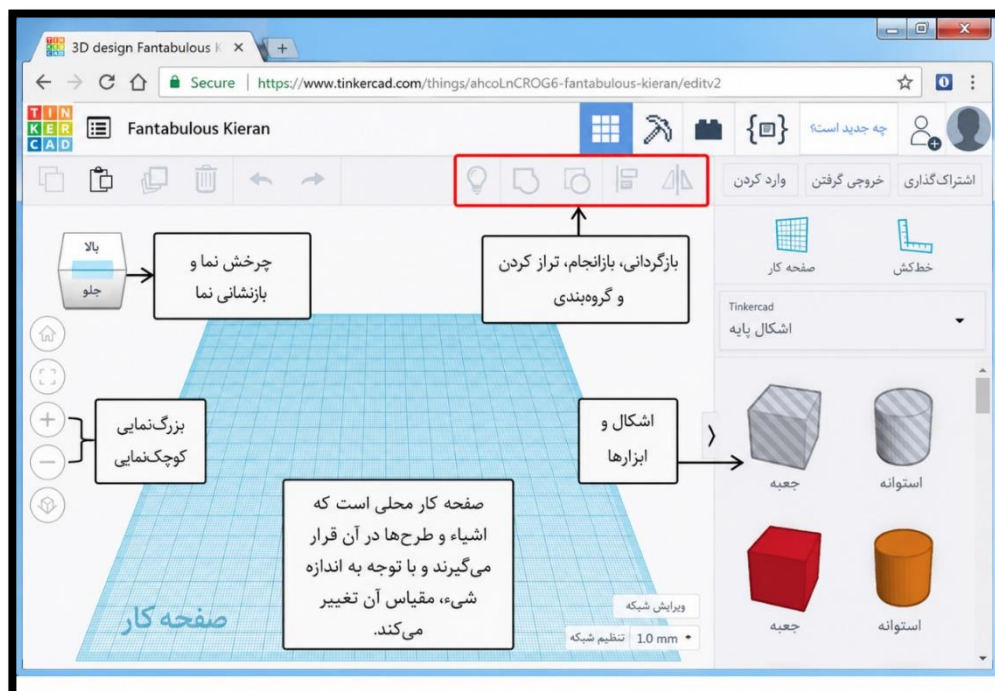
صفحه طراحی (Workplane): محلی است که مدل روی آن ساخته می شود.

اشکال سه بعدی (Basic Shapes): در این قسمت اشکال مختلف مانند مکعب، استوانه، کره و سایر حجم ها قرار دارند و می توان آن ها را وارد صفحه طراحی کرد.

ابزار جابه جایی: برای تغییر محل قرار گرفتن مدل روی صفحه استفاده می شود.

ابزار تغییر اندازه: با کشیدن دسته های اطراف شکل، طول، عرض و ارتفاع آن تغییر می کند.

ابزار چرخش: این ابزار برای تغییر جهت قرار گرفتن مدل استفاده می شود.



## ساخت مدل با اشکال ساده

در Tinkercad تقریباً همه مدل‌ها از ترکیب چند شکل ساده ساخته می‌شوند. برای مثال یک جاکلیدی می‌تواند از:

- یک مکعب
- یک استوانه
- یک سوراخ دایره‌ای

تشکیل شود. ابتدا اشکال را روی صفحه قرار می‌دهیم، سپس اندازه آن‌ها را تغییر داده و در محل مناسب قرار می‌دهیم.

## ابزار Hole

گاهی لازم است بخشی از مدل حذف شود.

در Tinkercad این کار با ابزار Hole انجام می‌شود.

ابتدا شکل موردنظر را به حالت Hole تبدیل می‌کنیم، سپس آن را روی مدل قرار داده و هر دو شکل را انتخاب می‌کنیم. پس از استفاده از دستور Group، قسمت موردنظر از مدل حذف می‌شود.

## ابزار Goup

ابزار Group چند شکل را به یک مدل واحد تبدیل می‌کند.

این ابزار باعث می‌شود تمام اجزای مدل به صورت یک قطعه واحد جابه‌جا یا ویرایش شوند.

## ذخیره فایل برای چاپ

پس از پایان طراحی، باید مدل ذخیره شود.

برای چاپ سه‌بعدی معمولاً از فرمت STL استفاده می‌شود.

## فعالیت کلاسی

یک مکعب و استوانه را وارد صفحه طراحی کنید و اندازه آن را به دلخواه تغییر دهید.

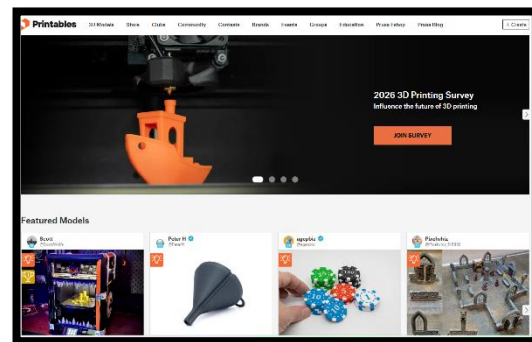
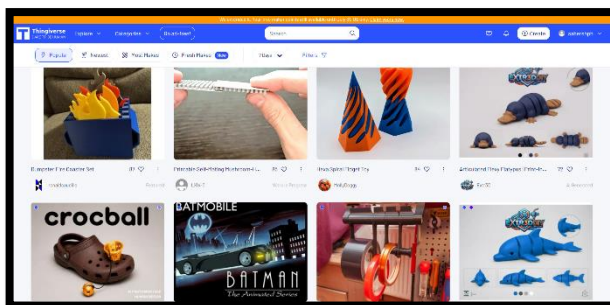
همه اشکال را انتخاب کرده و با استفاده از دستور Group آن‌ها را به یک مدل تبدیل کنید.

مدل نهایی را با فرمت STL ذخیره کنید.

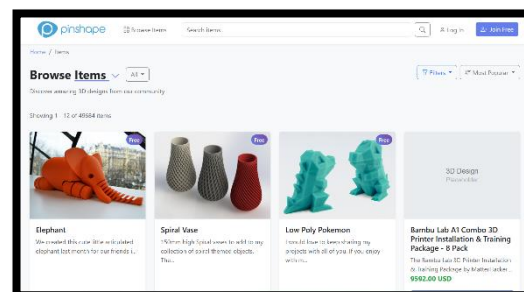
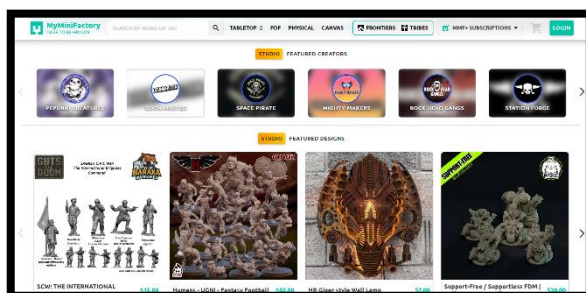
## کتابخانه‌ها و پایگاه‌های آنلاین مدل‌های سه‌بعدی

سریع‌ترین و ساده‌ترین روش برای آغاز کار با پرینترهای سه‌بعدی، استفاده از مدل‌های آماده و رایگانی است که در شبکه جهانی اینترنت در دسترس قرار دارند. این مدل‌ها معمولاً با پسوندهای رایج (STL) یا (OBJ) ارائه می‌شوند که استانداردهای جهانی برای تبادل داده‌های هندسی سه‌بعدی محسوب می‌شوند. وب‌سایت‌های متعددی وجود دارند که مجموعه‌ای گسترده از این مدل‌ها را برای بارگیری فراهم کرده‌اند. برترین و کاربردی‌ترین این پایگاه‌ها در فهرست زیر معرفی شده‌اند:

- پروساپرینترز
- آدرس وب سایت [www.prusaprinters.org](http://www.prusaprinters.org)
- نوع مدل‌ها: رایگان
- تینگی ورس
- آدرس وب سایت [www.thingiverse.com](http://www.thingiverse.com)
- نوع مدل‌ها: رایگان



- پروساپرینترز
- آدرس وب سایت [www.myminifactory.com](http://www.myminifactory.com)
- نوع مدل‌ها: رایگان و غیر رایگان
- پروساپرینترز
- آدرس وب سایت [www.pinshape.com](http://www.pinshape.com)
- نوع مدل‌ها: رایگان و غیر رایگان



## دستیار هوشمند

استفاده از اینترنت و هوش مصنوعی برای طراحی بهتر گاهی اوقات ایده‌ای برای طراحی داریم، اما نمی‌دانیم شکل آن چگونه است یا از چه قسمت‌هایی تشکیل شده است. در این شرایط می‌توانیم از موتورهای جست‌وجو و ابزارهای هوش مصنوعی کمک بگیریم.

**جست‌وجوی اینترنتی**

پیش از شروع طراحی، می‌توانید تصاویر وسیله مورد نظر را در اینترنت جست‌وجو کنید. مشاهده تصاویر از زوایای مختلف به شما کمک می‌کند شکل و ابعاد قطعه را بهتر بشناسید.

برای مثال اگر قصد طراحی یک جاکلیدی، گلدان یا پایه تلفن همراه را دارید، ابتدا نمونه‌های مختلف آن را مشاهده کنید و سپس طراحی خود را آغاز کنید.

### کمک گرفتن از هوش مصنوعی

ابزارهای هوش مصنوعی می‌توانند در مراحل مختلف طراحی به شما کمک کنند. برای مثال می‌توانید از آن‌ها بخواهید:

- ایده‌هایی برای طراحی یک وسیله ارائه دهند.
- مراحل طراحی یک مدل در Tinkercad را توضیح دهند.
- اندازه تقریبی یک قطعه را پیشنهاد کنند.
- روش ساخت یک مدل را مرحله به مرحله بیان کنند.

### نمونه پرامیت

می‌خواهم در نرم‌افزار Tinkercad یک جامدادی ساده طراحی کنم. مراحل طراحی را به زبان ساده و مناسب دانش آموز پایه هشتم توضیح بده.

یا

برای طراحی یک پایه تلفن همراه در Tinkercad از چه اشکال هندسی استفاده کنم؟

یا

یک جاکلیدی با طرح ستاره پیشنهاد بده و مراحل ساخت آن را توضیح بده.

### از تصویر تا مدل سه بعدی

بعضی از ابزارهای هوش مصنوعی می‌توانند از روی یک تصویر، مدل سه بعدی تولید کنند یا فایل‌هایی برای ویرایش در نرم‌افزارهای طراحی ایجاد نمایند. البته در بسیاری از موارد لازم است مدل تولیدشده را بررسی و اصلاح کنید تا برای چاپ مناسب باشد.

بنابراین بهتر است همیشه پیش از چاپ:

- اندازه مدل را بررسی کنید.
- ضخامت قطعات را کنترل کنید.
- از قابل چاپ بودن مدل مطمئن شوید.

**نکته:** هوش مصنوعی یک **دستیار طراحی** است، نه جایگزین خلاقیت شما. بهترین نتیجه زمانی به دست می‌آید که ابتدا خودتان ایده پردازی کنید و سپس از هوش مصنوعی برای کامل تر کردن طرح یا رفع اشکال‌های آن کمک بگیرید.

## خلاصه درس



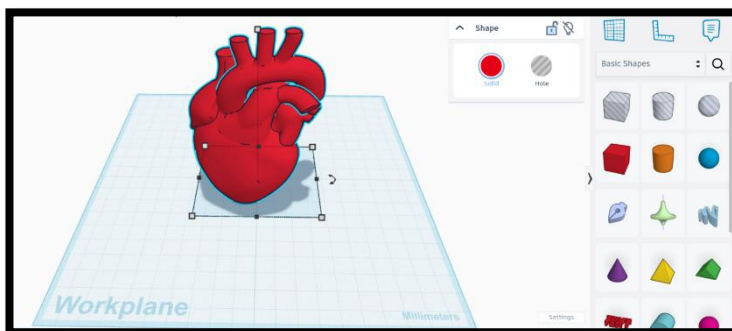
- Tinkercad یک نرم افزار ساده برای طراحی سه بعدی است.
- مدل های سه بعدی از ترکیب اشکال هندسی ساخته می شوند.
- با ابزارهای Move، Rotate و Scale می توان مدل را ویرایش کرد.
- ابزار Hole برای حذف بخشی از مدل و ابزار Group برای ترکیب اشکال به کار می رود.
- پس از پایان طراحی، مدل با فرمت STL ذخیره می شود.

## پروژه



یکی از مدل های زیر را در نرم افزار تینکر کد طراحی کنید:

- پایه تلفن همراه
  - پس از پایان طراحی:
  - ۱. فایل را با فرمت STL ذخیره کنید.
  - ۲. از محیط طراحی یک تصویر تهیه کنید.
  - ۳. توضیح دهید در طراحی خود از چه اشکالی استفاده کرده اید.
- نمون فایل:



## بخش سوم: ذهن افزار

### Mindware

## درس ۲۳: تفکر محاسباتی و حل مسئله



در پایان این درس خواهید توانست:

- مفهوم الگوریتم را توضیح دهید.
- مراحل حل مسئله را نام ببرید.
- اهمیت ترتیب انجام کارها را بیان کنید.
- الگوها را تشخیص دهید.
- یک مسئله را به بخش‌های کوچک‌تر تقسیم کنید.

تا حالا شده برای رسیدن به مدرسه دنبال بهترین و کوتاه ترین مسیر بگردید؟ یا بخواهید اتاقتان را سریع تر مرتب کنید و فکر کنید از کجا شروع کنید؟ شاید هم هنگام انجام یک پروژه درسی، مجبور شده باشید کار را به چند مرحله تقسیم کنید.

۱. به نظر شما برای حل این مسئله ها چه کارهایی انجام می دهیم؟
  ۲. آیا همه مسئله ها را می توان یک باره حل کرد؟
  ۳. اگر یک کار بزرگ را به چند بخش کوچک تقسیم کنیم، حل آن آسان تر می شود؟
  ۴. چطور می توانیم برای انجام یک کار، یک ترتیب درست و مرحله به مرحله طراحی کنیم؟
- امروز می خواهیم با روشی آشنا شویم که به ما کمک می کند مسئله ها را دقیق تر بررسی کنیم، آن ها را به بخش های کوچک تر تقسیم کنیم و برای حلشان راه حل های منظم و مرحله به مرحله بسازیم. به این شیوه فکر کردن، **تفکر محاسباتی** می گویند.

به فرایندی که فرد سعی می کند با کمک آن برای مسائل زندگی روزمره خویش و دیگران راه حل های مؤثری پیدا کند، «حل مسئله» گویند.

برای حل مسئله باید درباره راه حل های مختلف تصمیم گیری کرد. شما در طول زندگی همواره در حال تصمیم گیری هستید. مثلاً برای اینکه پروژه های درس کار و فناوری را بهتر انجام دهید، چه اقدامی می کنید؟ چگونه محصول بهتری آماده می کنید؟ در واقع با تصمیم گیری درباره چگونگی حل مسئله، برای حل آن برنامه ریزی می کنید. اگر خود را ملزم کنید که برای همه مراحل زندگی برنامه ریزی کنید، یاد می گیرید که قبل از اقدام به هر کاری درباره آن سنجیده و منطقی فکر کنید و آینده نگر باشید.

برنامه ریزی قبل از انجام دادن هر کار باعث می شود که آن را با اعتماد به نفس و تسلط بیشتری شروع کنید و اگر در حین اجرا با مشکلی مواجه شدید، جای سردرگم شدن می کوشید برای آن راه حل مناسبی پیدا کنید. گاهی اوقات می توان به سادگی راه حل مسئله را تشخیص داد، مانند محاسبه مجموع دو عدد یا محاسبه مساحت یک دایره، اما اگر مسئله پیچیده باشد باید آن را به چند مسئله کوچک تر تقسیم کرد. به این ترتیب با حل زیرمسئله ها و کنار هم قرار دادن آن ها، امکان حل مسئله اصلی فراهم می شود.

برای حل درست مسئله، باید فکر کردن منظم و مرحله به مرحله را یاد بگیریم. به این نوع فکر کردن، تفکر محاسباتی می گویند.



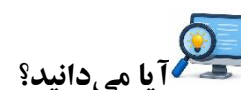
تفکر محاسباتی یعنی:

## الگوریتم چیست؟

الگوریتم روشی گام به گام برای حل مسئله است. به هر دستورالعملی که مراحل انجام دادن کاری را با زبانی دقیق و با جزئیات کافی بیان می کند، به طوری که ترتیب مراحل و شرط خاتمه عملیات در آن کاملاً مشخص باشد، «الگوریتم» گویند. مثال ساده، اگر بخواهیم یک چای درست کنیم، می توانیم این مراحل را بنویسیم:

۱. آب را در کتری بریز.
۲. کتری را روی شعله بگذار.
۳. صبر کن تا آب جوش بیاید.
۴. چای را در قوری بریز.
۵. آب جوش را اضافه کن.
۶. چند دقیقه صبر کن.
۷. چای را در فنجان بریز.

این یک الگوریتم است، چون مراحل آن مشخص و مرتب اند.



## آیا می دانید؟

واژه «الگوریتم» از نام ریاضی دان، ستاره شناس و جغرافی دان مشهور ایرانی، ابوجعفر محمد بن موسی الخوارزمی گرفته شده است. خوارزمی در سال ۱۶۰ خورشیدی در خراسان به دنیا آمد و بیشتر عمر خود را در بغداد زندگی کرد و در ۲۲۶ خورشیدی از دنیا رفت. وی تعدادی کتاب علمی را از زبان های یونانی و سانسکریت به عربی ترجمه کرد و چندین کتاب در زمینه ریاضیات، نجوم، جغرافیا و تاریخ نوشت. این کتاب ها سال ها پس از مرگ وی به زبان لاتین ترجمه شدند و در دانشگاه های اروپایی مورد مطالعه قرار گرفتند. واژه «جبر» از عنوان یکی از کتاب های او گرفته شده است.

## ویژگی های یک الگوریتم خوب:

- روشن و قابل فهم باشد.
- ترتیب در آن مشخص باشد.
- همه مراحل لازم را داشته باشد.
- به نتیجه برسد و از ابهام دور باشد.

## مراحل حل مسئله

حل مسئله معمولاً به صورت گام به گام انجام می شود:



## ترتیب انجام کارها

در بسیاری از کارها، ترتیب انجام مراحل بسیار مهم است.

اگر مراحل را جابه جا انجام دهیم، ممکن است کار به نتیجه نرسد.

مثال:

برای بستن بند کفش، ابتدا باید بند را از سوراخ ها رد کنیم، بعد آن را گره بزنیم.

اگر گره را قبل از رد کردن بندها انجام دهیم، کار درست پیش نمی رود.

نکته مهم:

- در الگوریتم ها، ترتیب دستورها مثل پله های یک نردبان است؛
- اگر یکی از پله ها حذف یا جابه جا شود، رسیدن به هدف دشوار یا ناممکن می شود.

## تشخیص الگو

الگو یعنی شباهت یا نظم تکرار شونده در بین چیزها، اعداد، شکل ها یا رویدادها.

چرا تشخیص الگو مهم است؟

وقتی الگو را پیدا می کنیم:

- مسئله را سریع تر می فهمیم.
- راه حل را راحت تر پیدا می کنیم.
- می توانیم آینده یا ادامه کار را بهتر پیش بینی کنیم.

مثال ها:

- در دنباله ۲، ۴، ۶، ۸، ۱۰ ... الگو این است که هر بار ۲ تا اضافه می شود.
  - در ساعت، عقربه ها با نظم مشخصی حرکت می کنند.
  - در یک برنامه، بعضی دستور ها ممکن است بارها تکرار شوند.
- در تفکر محاسباتی تشخیص الگو کمک می کند بفهمیم یک مسئله شبیه کدام مسئله های قبلی است و چگونه می توان آن را حل کرد.

## تقسیم مسئله به بخش های کوچکتر

یکی از مهم ترین مهارت های تفکر محاسباتی، شکستن مسئله بزرگ به مسئله های کوچک تر است.

چرا این کار مهم است؟

- مسئله ساده تر می شود.
- حل آن راحت تر می شود.
- احتمال خطا کمتر می شود.
- می توان هر بخش را جدا گانه بررسی کرد.

مثال:

اگر بخواهیم یک برنامه رایانه ای بنویسیم، بهتر است ابتدا:

- ورودی ها را مشخص کنیم.
- مراحل کار را بنویسیم.
- خروجی را تعیین کنیم.
- سپس کد را مرحله به مرحله بنویسیم.

این چهار مهارت به هم وابسته اند:

۱. تشخیص الگو: کمک می کند مسئله را بهتر بفهمیم.

۲. تقسیم مسئله به بخش های کوچک تر: حل آن را آسان می کند.
۳. الگوریتم: مسیر حل را مشخص می کند.
۴. ترتیب انجام کارها: باعث می شود الگوریتم درست اجرا شود.

پس تفکر محاسباتی فقط برنامه نویسی نیست؛ بلکه یک روش درست فکر کردن برای حل مسئله است.

آیا می دانید؟



تشخیص الگو یکی از پایه های مهم حل مسئله در ریاضی و کامپیوتر است.

فعالیت کلاسی



۱- یکی از کارهای روزانه خود را انتخاب کنید، مانند آماده شدن برای رفتن به مدرسه سپس مراحل آن را به ترتیب بنویسید.

۲- مسئله «آماده کردن یک ارائه کلاسی» را به بخش های کوچک تر تقسیم کنید.

۳- یک مسئله بزرگ را انتخاب کنید و آن را به چند بخش کوچک تر تقسیم کنید.

خلاصه درس



- الگوریتم مجموعه ای از مراحل مرتب برای حل مسئله است.
- حل مسئله با تعریف نیاز، بررسی، جمع آوری اطلاعات، انتخاب راه حل و اجرای آن انجام می شود.
- ترتیب انجام کارها در رسیدن به نتیجه بسیار مهم است.
- تشخیص الگو به فهم بهتر مسائل کمک می کند.
- تقسیم مسئله به بخش های کوچک تر باعث می شود حل آن آسان تر شود.
- این مهارت ها پایه های مهم تفکر محاسباتی هستند.

تمرین



۱. الگوریتم چیست؟

.....

۲. چرا ترتیب مراحل در حل مسئله مهم است؟

.....

۳. تشخیص الگو چه کمکی به ما می کند؟

.....

۴. چرا باید مسئله را به بخش های کوچک تر تقسیم کنیم؟

.....

۵. تفکر محاسباتی یعنی چه؟

.....



پروژه

فرض کنید هر روز صبح باید در مدت ۳۰ دقیقه برای رفتن به مدرسه آماده شوید؛ اما گاهی بعضی کارها را فراموش می کنید یا به دلیل انجام نادرست کارها، دیر به مدرسه می رسید. با استفاده از چهار مهارت تفکر محاسباتی، یک روش منظم برای آماده شدن و رفتن به مدرسه طراحی کنید.

اگر ترتیب دو مرحله را تغییر دهید، چه اتفاقی می افتد؟ آیا الگوریتم شما همچنان درست اجرا می شود؟

یک فلوجارت، جدول مرحله ای یا پوستر آموزشی طراحی کنید که در آن مسیر آماده شدن برای مدرسه، از زمان بیدار شدن تا خروج از خانه، به صورت منظم نشان داده شده باشد.

## درس ۲۴: آشنایی با هوش مصنوعی



در پایان این درس خواهید توانست:

- تفاوت مغز انسان و هوش مصنوعی را بیان کنید.
- نقش آموزش در عملکرد هوش مصنوعی را درک کنید.
- کاربردهای شبکه‌های عصبی را در زندگی روزمره معرفی کنید.

انسان با استفاده از مغز خود یاد می گیرد، تصمیم می گیرد و مسائل را حل می کند. دانشمندان تلاش کرده اند سامانه هایی طراحی کنند که بتوانند تا حدی مانند مغز انسان یاد بگیرند و تصمیم بگیرند. نتیجه این تلاش، شبکه های عصبی مصنوعی است که یکی از مهم ترین فناوری های هوش مصنوعی به شمار می رود.

امروزه بسیاری از سامانه های هوشمند مانند تشخیص چهره، ترجمه زبان، دستیارهای صوتی و خودروهای هوشمند از شبکه های عصبی مصنوعی استفاده می کنند.

به پرسش های زیر پاسخ دهید.

انسان چگونه چهره دوستان خود را تشخیص می دهد؟

آیا رایانه می تواند مانند انسان یاد بگیرد؟ چرا؟

به نظر شما تلفن همراه چگونه چهره صاحب خود را تشخیص می دهد؟

### هوش مصنوعی چیست؟



هوش مصنوعی مجموعه ای از روش ها و برنامه های رایانه ای است که تلاش می کند شیوه یادگیری و تصمیم گیری مغز انسان را شبیه سازی کند. این سامانه ها می توانند با دریافت اطلاعات، آموزش ببینند و به مرور زمان عملکرد خود را بهبود دهند.

### مقایسه هوش طبیعی و مصنوعی

| ویژگی         | هوش انسان                      | هوش مصنوعی (شبکه عصبی)                |
|---------------|--------------------------------|---------------------------------------|
| واحد پردازش   | نورون زیستی                    | نورون مصنوعی (ریاضی)                  |
| سرعت پردازش   | آهسته تر ولی بسیار انعطاف پذیر | سریع ولی محدود به داده ها             |
| قدرت یادگیری  | بسیار گسترده و خلاقانه         | محدود به داده ها و الگوریتم           |
| انرژی مصرفی   | کم و بهینه                     | نسبتاً زیاد (نیاز به برق و سخت افزار) |
| احساس و آگاهی | دارد                           | ندارد                                 |

### شبکه عصبی مصنوعی چیست؟

شبکه عصبی مصنوعی مدلی است که از نحوه عملکرد سلول های عصبی مغز انسان الهام گرفته است.

این شبکه از واحدهای کوچکی به نام **نورون مصنوعی** تشکیل شده است. هر نورون اطلاعات را دریافت می کند، آن ها را پردازش می کند و نتیجه را به نورون های دیگر منتقل می کند. همکاری تعداد زیادی از این نورون ها باعث می شود رایانه بتواند الگوها را تشخیص دهد و مسائل مختلف را حل کند.

### شبکه عصبی چگونه یاد می گیرد؟

شبکه عصبی برای اینکه بتواند کار خود را درست انجام دهد، باید آموزش ببیند.

در مرحله آموزش:

- نمونه های زیادی از اطلاعات به شبکه داده می شود.

- شبکه پاسخ خود را ارائه می دهد.
  - اگر پاسخ درست نباشد، تنظیمات داخلی آن اصلاح می شود.
  - این فرایند بارها تکرار می شود تا دقت شبکه افزایش یابد.
- به همین دلیل، هرچه داده های آموزشی بیشتر و با کیفیت تر باشند، عملکرد شبکه نیز بهتر خواهد بود.

### نقش وزن ها در شبکه عصبی

در شبکه های عصبی، ارتباط بین نورون ها دارای وزن (Weight) است. وزن ها میزان اهمیت هر ورودی را مشخص می کنند. هنگام آموزش، این وزن ها تغییر می کنند تا شبکه بتواند پاسخ های دقیق تری ارائه دهد. به همین دلیل، یادگیری شبکه در واقع با تنظیم وزن ها انجام می شود.

### کاربردهای شبکه های عصبی

شبکه های عصبی مصنوعی در بسیاری از فناوری های امروزی استفاده می شوند، از جمله:

- تشخیص چهره در تلفن های همراه
- ترجمه خودکار زبان ها
- تشخیص گفتار و دستیارهای صوتی
- پیش بینی وضعیت آب و هوا
- تشخیص بیماری ها در پزشکی
- خودروهای هوشمند
- پیشنهاد فیلم، موسیقی یا محصولات در فروشگاه های اینترنتی

### نکات مهم

- شبکه های عصبی از عملکرد مغز انسان الهام گرفته اند، اما دقیقاً مانند مغز انسان نیستند.
- شبکه عصبی بدون آموزش نمی تواند عملکرد مناسبی داشته باشد.
- کیفیت داده های آموزشی بر دقت نتایج هوش مصنوعی تأثیر زیادی دارد.
- وزن ها در شبکه عصبی هنگام آموزش تغییر می کنند و باعث بهبود عملکرد آن می شوند.

### آیا می دانید؟



مغز انسان حدود ۸۶ میلیارد سلول عصبی (نورون) دارد، اما شبکه های عصبی مصنوعی تنها مدل ساده ای از عملکرد این نورون ها هستند.

➤ بسیاری از سامانه های هوش مصنوعی که امروزه از آن ها استفاده می کنیم، مانند تشخیص چهره یا ترجمه زبان، بر پایه شبکه های عصبی مصنوعی ساخته شده اند.

➤ هرچه شبکه عصبی داده های بیشتری ببیند، معمولاً دقت آن نیز بیشتر می شود؛ البته کیفیت داده ها نیز اهمیت زیادی دارد.

## فعالیت کلاسی

شبیه سازی مغز مصنوعی در کلاس

فعالیت گروهی: ساخت شبکه عصبی با طناب و کارت

وسایل لازم:

- کارت هایی برای نورون ها
- طناب یا نخ برای اتصال نورون ها
- کاغذ رنگی برای ورودی، پنهان و خروجی

مراحل اجرا:

دانش آموزان به سه گروه تقسیم می شوند: ورودی، پنهان، خروجی. کارت ها به شکل نورون طراحی و نام گذاری می شوند. طناب ها به عنوان اتصال بین نورون ها استفاده می شوند. یک داده فرضی (مثلاً تشخیص حیوان) وارد شبکه می شود و مسیر پردازش نمایش داده می شود. در پایان، دانش آموزان نقش نورون ها را بازی می کنند و نتیجه نهایی را اعلام می کنند.

## خلاصه درس

- هوش مصنوعی تلاش می کند شیوه یادگیری مغز انسان را شبیه سازی کند.
- شبکه عصبی مصنوعی از نورون های مصنوعی تشکیل شده است.
- شبکه های عصبی با استفاده از داده های آموزشی یاد می گیرند.
- وزن ها در هنگام آموزش تغییر می کنند و باعث بهبود عملکرد شبکه می شوند.
- شبکه های عصبی در بسیاری از فناوری های هوشمند امروزی کاربرد دارند.

## تمرین

۱. هوش مصنوعی را به زبان خود تعریف کنید.

.....

۲. شبکه عصبی مصنوعی از چه الگویی الهام گرفته است؟

.....

۳. چرا شبکه های عصبی به آموزش نیاز دارند؟

.....

۴. نقش وزن ها در شبکه عصبی چیست؟

.....

۵. سه کاربرد شبکه های عصبی مصنوعی را نام ببرید.

.....



پروژه

**بررسی کاربردهای شبکه‌های عصبی**

به صورت گروهی یکی از کاربردهای شبکه‌های عصبی (مانند تشخیص چهره، ترجمه زبان، پزشکی یا خودروهای هوشمند) را انتخاب کنید. درباره نحوه استفاده از شبکه عصبی در آن کاربرد تحقیق کرده و نتیجه را در قالب یک پوستر یا ارائه کوتاه به کلاس معرفی کنید.

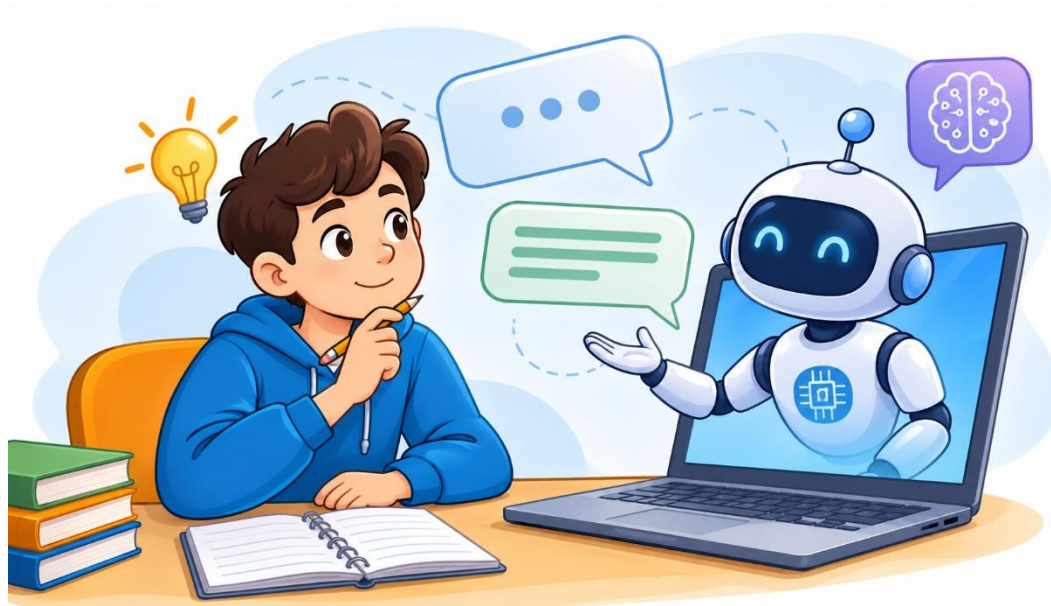


چالش

چالش ۱: تحقیق کنید «یادگیری عمیق (Deep Learning) چیست و چه تفاوتی با شبکه‌های عصبی ساده دارد.

چالش ۲: یکی از برنامه‌های هوش مصنوعی که روزانه از آن استفاده می‌کنید (مانند مترجم، دستیار صوتی یا موتور جست‌وجو) را انتخاب کنید و توضیح دهید شبکه عصبی چگونه می‌تواند در عملکرد آن نقش داشته باشد.

## درس ۲۵: پرامپت نویسی برای هوش مصنوعی



در پایان این درس خواهید توانست:

- مفهوم پرامپت را به زبان ساده توضیح دهید.
- یک درخواست روشن و مؤثر برای هوش مصنوعی بنویسید.
- یک پرامپت را با توجه به نقش، هدف، زمینه، محدودیت و قالب تنظیم کنید.
- پاسخ هوش مصنوعی را بررسی، اصلاح و راستی آزمایی کنید.
- از هوش مصنوعی به صورت مسئولانه، اخلاقی و مفید در کارهای درسی و زندگی روزمره استفاده کنید.



امروزه ابزارهای هوش مصنوعی می توانند در کارهایی مانند توضیح درس، خلاصه نویسی، ایده پردازی، تولید متن ساده و پاسخ به پرسش ها کمک کنند. اما این ابزارها فقط وقتی خوب عمل می کنند که ما سؤال یا دستور خود را درست و دقیق بنویسیم. به متن یا دستوری که به هوش مصنوعی می دهیم، پرامپت می گویند.

اگر پرامپت مبهم باشد، پاسخ هم ممکن است ناقص یا نامناسب باشد.

اگر پرامپت روشن، دقیق و مناسب باشد، پاسخ بهتر می شود.

بنابراین، پرامپت نویسی یعنی یاد گرفتنِ چطور سؤال کنیم تا پاسخ بهتر بگیریم.

۱. آیا تا به حال از یک برنامه یا سایت خواسته اید چیزی برایتان بنویسد یا توضیح دهد؟

۲. اگر یک سؤال خیلی کوتاه و مبهم پرسیم، چه اتفاقی می افتد؟

۳. چرا بعضی پاسخ ها دقیق و بعضی پاسخ ها ناقص هستند؟

۴. به نظر شما برای گرفتن پاسخ بهتر از یک هوش مصنوعی، چه چیزهایی لازم است؟

۵. آیا هوش مصنوعی همیشه درست می گوید؟

### پرامپت چیست؟



پرامپت (Prompt) یعنی درخواست یا دستوری که ما برای یک ابزار هوش مصنوعی می نویسیم تا کاری را انجام دهد. پرامپت می تواند یک سؤال، یک درخواست برای نوشتن، خلاصه کردن، توضیح دادن یا یک راهنمای مرحله به مرحله باشد.

مثال های ساده از پرامپت:

- این متن را در ۳ جمله خلاصه کن.

- این مفهوم را برای دانش آموز پایه هشتم ساده توضیح بده.

- برای این موضوع ۵ ایده پیشنهاد بده.

### چرا پرامپت نویسی مهم است؟

کیفیت پاسخ هوش مصنوعی تا حد زیادی به کیفیت پرامپت بستگی دارد. اگر پرامپت:

- مبهم باشد: پاسخ هم کلی و نامرتبط می شود.

- دقیق و کامل باشد: پاسخ کاربردی تر و نزدیک تر به نیاز ما خواهد بود.

پرامپت نویسی یک مهارت است؛ یعنی با تمرین یاد می گیریم چطور بهتر سؤال کنیم تا زمان کمتر تلف شود و نتیجه بهتر باشد.

### ویژگی های یک پرامپت خوب

یک پرامپت خوب معمولاً این ویژگی ها را دارد:

۱. هدف روشن: دقیقاً مشخص می کند خروجی چه چیزی باشد.

۲. اطلاعات زمینه ای کافی: موضوع، سطح مخاطب و شرایط را توضیح می دهد.

۳. محدودیت مشخص: می گوید چه چیزهایی رعایت شود (مثل کوتاهی متن، ساده نویسی، پرهیز از موارد خاص).

۴. قالب خروجی: شکل پاسخ را تعیین می کند (فهرست، جدول، مراحل شماره دار، ۵ جمله و ...).

۵. قابلیت بررسی: طوری نوشته می شود که بتوان خروجی را کنترل و اصلاح کرد.

### الگوی ساده و استاندارد برای نوشتن پرامپت

۱. نقش: هوش مصنوعی چه نقشی داشته باشد؟ (مثلاً: معلم، ویراستار، ایده پرداز)

۲. هدف: چه کاری انجام دهد؟ (مثلاً: توضیح، خلاصه نویسی، ساخت سوال)

۳. زمینه: چه اطلاعاتی لازم است بداند؟ (موضوع، پایه تحصیلی، مخاطب)

۴. محدودیت: چه چیزهایی را رعایت کند؟ (کوتاه، ساده، بدون اطلاعات شخصی، بدون حدس بی پایه)

۵. قالب: پاسخ را چگونه ارائه کند؟ (جدول، فهرست، مرحله ای، ۵ جمله)

۶. ارزیابی: چگونه کیفیت پاسخ بررسی شود؟ (ذکر منابع، بیان نکات مشکوک، پیشنهاد روش بررسی)

نکته: همه پرامپت ها لازم نیست هر ۶ بخش را داشته باشند؛ اما هرچه کار مهم تر و دقیق تر باشد، استفاده از این بخش ها کمک بیشتری می کند.

### نمونه سازی: از پرامپت ضعیف تا پرامپت قوی

➤ پرامپت ضعیف (مبهم):

«درباره آلودگی هوا بنویس»

این درخواست مشخص نمی کند متن برای چه کسی است، چه اندازه ای دارد و باید درباره کدام بخش از آلودگی هوا توضیح دهد.

➤ پرامپت بهتر (دقیق تر):

«یک متن کوتاه و ساده درباره علت های آلودگی هوا و راه های کاهش آن برای دانش آموزان پایه هشتم بنویس»

در این درخواست، موضوع، مخاطب و هدف مشخص شده است.

پرامپت کامل تر (ساختار یافته):

«نقش: یک معلم علوم پایه هشتم باش.

موضوع: آلودگی هوا.

مخاطب: دانش آموزان پایه هشتم.

هدف: توضیح علت های آلودگی هوا و معرفی راه های کاهش آن.

خروجی: یک متن آموزشی در حدود ۱۵۰ کلمه بنویس. ابتدا آلودگی هوا را در دو جمله تعریف کن، سپس سه علت و

سه راهکار برای کاهش آن ارائه بده. از زبان ساده و مثال های روزمره استفاده کن و در پایان، یک پرسش برای گفت و گو

در کلاس مطرح کن»

## اصلاح پرامپت (چرخه‌ی آزمون و بهبود)

گاهی پاسخ اول دقیق نیست. در این حالت، به جای تکرار همان سؤال، باید پرامپت را اصلاح کنیم. راه‌های ساده برای بهتر کردن پرامپت:

- شفاف‌تر کردن هدف: «خلاصه کن» یا «مقایسه کن» یا «مرحله به مرحله توضیح بده»
- افزودن سطح مخاطب: «برای پایه هشتم» یا «به زبان خیلی ساده»
- محدود کردن پاسخ: «در ۶ خط»، «در ۵ نکته»، «بدون مثال علمی پیچیده»
- درخواست قالب مشخص: «جدول دو ستونه»، «فهرست شماره‌دار»
- درخواست بازبینی: «دو ایراد احتمالی پاسخ را خودت پیدا کن و اصلاح کن»

## راستی آزمایی (Fact-check) و خطاهای رایج

هوش مصنوعی ممکن است:

- اطلاعات قدیمی بدهد،
  - برخی جزئیات را اشتباه بگوید،
  - یا با اطمینان، جمله‌ای بسازد که منبع دقیق ندارد (این حالت را «توهم» یا *Hallucination* می‌نامند).
- پس هنگام استفاده در تکالیف و پروژه‌ها باید:

۱. پاسخ را با کتاب درسی یا منابع معتبر آموزشی مقایسه کنیم.
۲. اگر پاسخ شامل عدد، تاریخ، نام کتاب/قانون/شخص بود، آن را دقیق‌تر بررسی کنیم.
۳. برای موضوعات حساس (پزشکی، حقوقی، امنیتی) فقط به خروجی هوش مصنوعی اکتفا نکنیم.

## نکات ایمنی و اخلاقی در نوشتن پرامپت

در پرامپت‌ها نباید اطلاعات خصوصی نوشته شود؛ مانند:

- شماره تلفن، آدرس، رمز عبور، کد ملی، اطلاعات بانکی
- عکس یا اطلاعات خصوصی دیگران بدون اجازه

همچنین در استفاده آموزشی باید:

- از هوش مصنوعی برای یادگیری کمک بگیریم، نه برای کپی‌کاری.
- اگر از آن کمک گرفته‌ایم، صادقانه در پروژه یا تکلیف (طبق قوانین مدرسه) اعلام کنیم.
- خروجی را ویرایش و شخصی‌سازی کنیم تا فهم خودمان را نشان دهد.

## فعالیت کلاسی

با انتخاب یک موضوع دلخواه، دو یا سه پرامپت با میزان جزئیات متفاوت بنویسید. سپس پرامپت‌ها را در کلاس با یکدیگر مقایسه کنید و به پرسش‌های زیر پاسخ دهید:

۱. کدام پرامپت، درخواست را روشن تر و دقیق تر بیان می کند؟
۲. چه ویژگی هایی باعث می شود یک پرامپت بهتر و کامل تر باشد؟
۳. کدام پرامپت پاسخ مناسب تر و دقیق تری تولید می کند؟ چرا؟
۴. اگر بخواهیم پاسخ کوتاه تر باشد، چگونه باید پرامپت را تغییر دهیم؟
۵. آیا مشخص کردن موضوع، هدف، مخاطب و شکل پاسخ می تواند کیفیت پاسخ را بهتر کند؟ توضیح دهید.

### خلاصه درس



- پرامپت یعنی دستور یا درخواستی که به هوش مصنوعی می دهیم.
- پرامپت خوب باید روشن، دقیق، کوتاه و هدفمند باشد.
- برای نوشتن پرامپت می توان از الگوی نقش، هدف، زمینه، محدودیت، قالب و ارزیابی استفاده کرد.
- باید پاسخ هوش مصنوعی را بررسی و راستی آزمایی کنیم.
- استفاده از هوش مصنوعی باید مسئولانه، اخلاقی و بدون افشای اطلاعات شخصی باشد.

### تمرین



۱. یک پرامپت برای توضیح «سواد دیجیتال» بنویسید.

.....

۲. یک پرامپت برای خلاصه کردن یک درس از کتاب فارسی یا علوم بنویسید.

.....

### پروژه



برای موارد زیر پرامپت بنویسید:

- توضیح درس
- خلاصه نویسی
- برنامه ریزی مطالعه
- حل یک مسئله ساده
- پیدا کردن ایده برای پروژه