

## برنامه نویسی پایتون ویژه نوجوانان

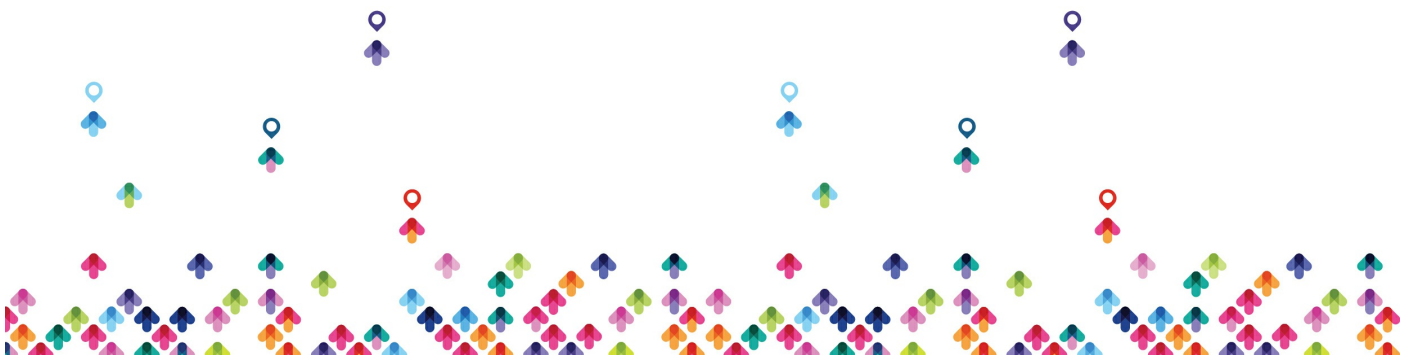
هدف اصلی از یادگیری پایتون مقدماتی برای نوجوانان، آشنایی با مفاهیم برنامه‌نویسی و توسعه نرم‌افزار است. این زبان برنامه‌نویسی، به دلیل سادگی و قدرت خود، برای شروع کار در حوزه برنامه‌نویسی گزینه مناسبی به شمار می‌آید.

### مروری بر دوره

هدف اصلی از یادگیری پایتون مقدماتی برای نوجوانان، آشنایی با مفاهیم برنامه‌نویسی و توسعه نرم‌افزار است. این زبان برنامه‌نویسی، به دلیل سادگی و قدرت خود، برای شروع کار در حوزه برنامه‌نویسی گزینه مناسبی به شمار می‌آید.

اهداف دیگری که برای یادگیری پایتون مقدماتی می‌توان عنوان کرد، عبارتند از:

- ۱- آشنایی با مفاهیم برنامه‌نویسی: یادگیری پایتون مقدماتی به نوجوانان کمک می‌کند تا با مفاهیم اساسی برنامه‌نویسی مانند متغیرها، عملگرها، شرطها و حلقه‌ها آشنا شوند.
- ۲- توسعه مهارت‌های محاسباتی: یادگیری پایتون مقدماتی نوجوانان را برای تقویت و توسعه مهارت‌های محاسباتی‌شان آماده می‌کند. آن‌ها می‌توانند با استفاده از پایتون، مسائل ریاضی و محاسباتی را حل کرده و الگوریتم‌های مختلفی را پیاده‌سازی کنند.
- ۳- ایجاد علاقه به علوم کامپیوتر: یادگیری پایتون مقدماتی می‌تواند علاقه نوجوانان به علوم کامپیوتر را برانگیزاند. آن‌ها می‌توانند با استفاده از پایتون، برنامه‌های ساده‌ای نوشته و خلاقیت‌شان را در حل مسائل برنامه‌نویسی به کار بگیرند.
- ۴- آماده‌سازی برای یادگیری زبان‌های برنامه‌نویسی پیشرفته: یادگیری پایتون مقدماتی به نوجوانان کمک می‌کند تا پایه‌های قوی



برای یادگیری زبان‌های برنامه‌نویسی پیشرفته مانند جاوا، سی شارپ و سی پلاس پلاس آماده شوند.

## آنچه در این دوره خواهید آموخت

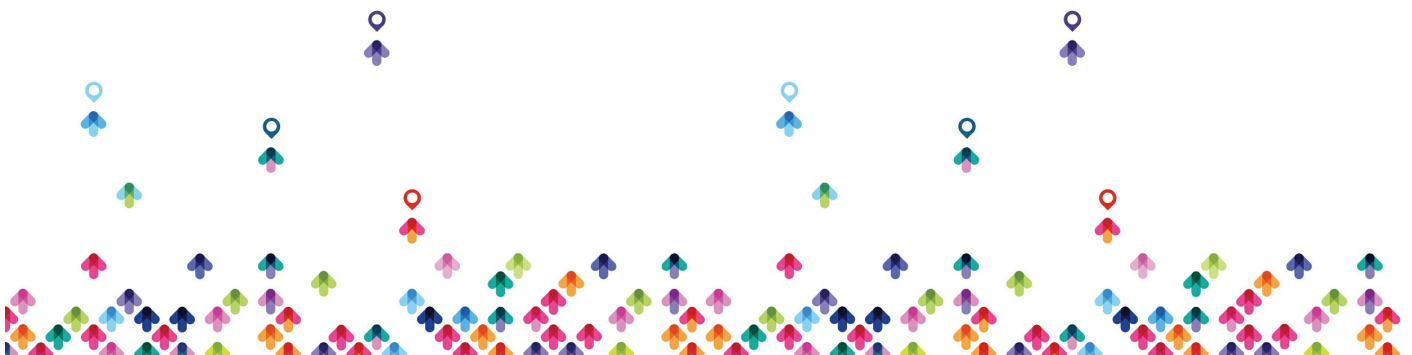
- هدف از انجام پروژه کوچک در دوره پایتون مقدماتی برای نوجوانان، تقویت مفاهیم آموخته شده و اعمال عملی آن‌ها در یک محیط واقعی است.

در این پروژه کوچک به موارد زیر می‌پردازیم:

- استفاده از متغیرها و عملگرها: ایجاد برنامه‌ای ساده که با دریافت اطلاعات از کاربر اطلاعات محاسبات ساده انجام دهد.
- ساختارهای کنترلی: توسعه یک برنامه با استفاده از شرطی‌ها و حلقه‌ها برای انجام وظایف مختلف.
- توابع: استفاده از توابع در یک پروژه برنامه‌نویسی، مثلاً تعریف توابع برای اجرای وظایف خاص.
- لیست‌ها و رشته‌ها: ایجاد یک برنامه که با لیست یا رشته کار می‌کند و انجام عملیات مختلف بر روی آن‌ها.
- ورودی و خروجی فایل: نوشتن یا خواندن اطلاعات از یک فایل در یک برنامه.
- پروژه به دانشجویان این امکان را می‌دهد تا مفاهیم اولیه را در یک محیط عملی اعمال کرده و با فرایند برنامه‌نویسی آشنا شوند. همچنین، این اجازه را می‌دهد تا مهارت‌های ارتباطی و حل مسائل‌شان را تقویت کنند. به عنوان مثال، ممکن است از دانشجویان خواسته شود تا یک برنامه ایجاد کنند که اطلاعات یک فرد را دریافت کرده و سپس محاسبات ساده‌ای بر روی آن اطلاعات انجام دهند، نتایج را نمایش داده یا در یک فایل ذخیره کند.
- این پروژه‌ها، به افزایش اعتماد به نفس و احساس موفقیت در مسیر یادگیری برنامه‌نویسی دانشجویان کمک فراوانی خواهد کرد.

## سرفصل‌ها

. پایتون چیست؟



پایتون یک زبان برنامه‌نویسی جذاب و کاربردی است که برای یادگیری آسان و توسعه برنامه‌های مختلف به کار می‌رود.

## ۲. چرا پایتون؟

پایتون به عنوان یک زبان ساده و کارآمد، برای شروع برنامه‌نویسی از سنین کم، مناسب‌ترین گزینه است. چرا که به راحتی می‌توان با آن انواع برنامه‌ها را نوشت و ایده‌های خلاقانه را پیاده‌سازی کرد.

## ۳. چرا ما پایتون را انتخاب کرده‌ایم؟

ما پایتون را به عنوان بهترین گزینه برای آموزش به نوجوانان و جوانانی که قصد ورود به دنیای برنامه‌نویسی را دارند، انتخاب کردیم؛ زیرا این زبان بسیار منعطف است و در انواع پروژه‌های کاری مورداستفاده قرار می‌گیرد.

## ۴. نصب مفسر پایتون

برای شروع، نحوه دانلود و نصب پایتون را از [سایت رسمی پایتون](#) می‌آموزید.

## ۵. استفاده از پایتون شل و IDLE به منظور نوشتن اولین برنامه

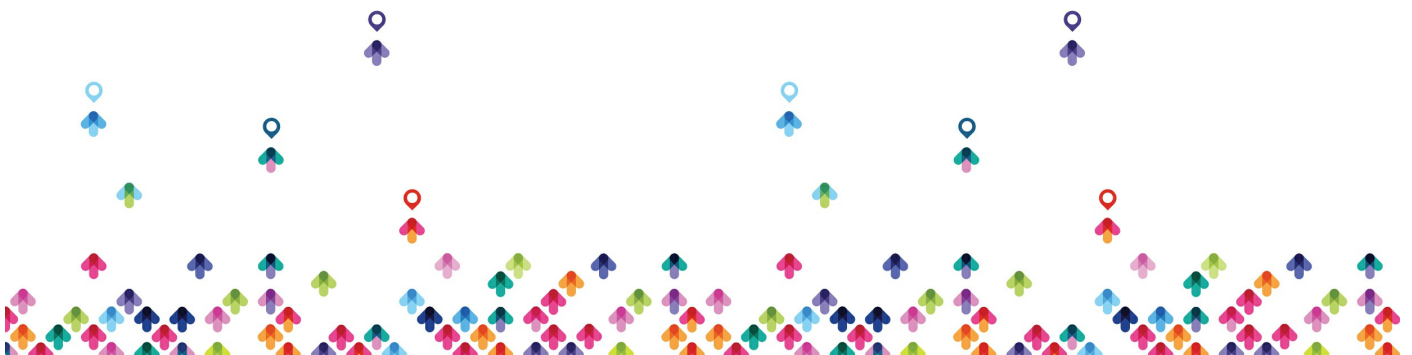
با استفاده از پایتون شل یا محیط IDLE، اولین برنامه کاربردی را می‌نویسیم.

## ۶. شناخت انواع متغیرها

احتمالاً می‌دانید که متغیرها ابزارهایی هستند که به شما امکان ذخیره‌سازی اطلاعات در برنامه‌ها را می‌دهند؛ در این مرحله با انواع متغیرها آشنا شده و نحوه کار با آن‌ها را می‌آموزید.

## ۷. نحوه نام‌گذاری یک متغیر در پایتون

با یادگیری قواعد نام‌گذاری متغیرها در پایتون می‌توانید کدهای خواناتری بنویسید.



## ۸. علامت واگذاری

با استفاده از علامت‌های واگذاری، می‌توانید کدهایتان را به بخش‌های مختلف تقسیم کرده و خوانایی آن‌ها را بالا ببرید.

## ۹. عملگرهای اصلی

با استفاده از عملگرهای اصلی مانند +، -، \* و /، عملیات ریاضی را در پایتون انجام خواهید داد.

## ۱۰. شناخت سایر عملگرهای واگذاری

با انواع عملگرهایی که در واگذاری متغیرها مورد استفاده قرار می‌گیرند آشنا خواهید شد.

## ۱۱. شناخت انواع داده‌ها

آشنایی با انواع مختلف داده‌ها مانند عدد صحیح، اعشاری، و رشته.

## ۱۲. آشنایی با توابع درون‌ساخت رشته‌ای

آموزش نحوه استفاده از توابع درون‌ساخت برای انجام عملیات مختلف بر روی رشته‌ها.

## ۱۳. فرمت‌دهی رشته‌ها با عملگر %

یادگیری روش‌های مختلف فرمت‌دهی رشته با استفاده از عملگر %.

## ۱۴. فرمت‌دهی رشته‌ها با تابع Input

CSS

استفاده از تابع Input برای گرفتن ورودی از کاربر و فرمت‌دهی رشته.



## ۱۵. تایپ کستینگ

تبدیل انواع مختلف داده به رشته با استفاده از تایپ کستینگ.

## ۱۶. نوع داده‌ای لیست (List)

آشنایی با لیست، یکی از ساختارهای داده مهم در پایتون.

## ۱۷. نوع داده‌ای تاپل (Tuple)

شناخت مفهوم تاپل و کاربردهای آن در برنامه‌نویسی.

## ۱۸. نوع داده‌ای دیکشنری (Dictionary)

آشنایی با دیکشنری و نحوه استفاده از آن در ذخیره اطلاعات.

## ۱۹. نحوه تعاملی کردن اولین برنامه

mathematica

یادگیری توابع `Print` و `Input` برای ساخت برنامه‌های تعاملی.

## ۲۰. تابع `Input`

CSS

کاربرد تابع `Input` برای گرفتن ورودی از کاربر هنگام اجرای برنامه.



## ۲۱. تابع **Print**

mathematica

استفاده از تابع **Print** برای نمایش اطلاعات به کاربر.

## ۲۲. کوتیشن سه تایی

آشنایی با مفهوم و کاربرد کوتیشن سه تایی برای نوشتن رشته‌های چندخطی.

## ۲۳. عبور از کاراکترها

استفاده از حلقه‌ها برای عبور از هر کاراکتر در یک رشته.

## ۲۴. ایجاد انتخاب و تصمیم

آشنایی با عبارات شرطی برای ایجاد تصمیمات در برنامه.

## ۲۵. آشنایی با عبارات شرطی **if**

استفاده از عبارات شرطی **if** برای انجام عملیات مختلف بر اساس شرایط.

## ۲۶. عبارت شرطی **inline if**

نحوه استفاده از عبارت شرطی در یک خط کد.

## ۲۷. حلقه **for**

نحوه استفاده از حلقه **for** برای تکرار عملیات‌ها بر روی یک مجموعه داده.



## ۲۸. حلقه **while**

arduino

استفاده از حلقه **while** برای تکرار عملیات‌ها تا زمانی که یک شرط مشخص شود.

## ۲۹. کلمه کلیدی **break**

kotlin

استفاده از کلمه کلیدی **break** برای خروج از حلقه در شرایط خاص.

## ۳۰. کلمه کلیدی **continue**

kotlin

استفاده از کلمه کلیدی **continue** برای انجام مرحله‌ای خاص در حلقه.

## ۳۱. عبارت کنترلی **try , except**

استفاده از عبارت کنترلی **try** و **except** برای مدیریت خطاها در برنامه.

## ۳۲. توابع و ماژول‌ها

شناخت مفهوم توابع و ماژول‌ها به عنوان اجزای مهم برنامه‌نویسی.

## ۳۳. توابع چه هستند؟



تعریف توابع و نحوه استفاده از آنها در برنامه‌نویسی.

## ۳۴. نحوه تعریف توابع

در این مرحله نحوه ایجاد توابع خودتان را برای انجام کارهای خاص در برنامه می‌آموزید.

## ۳۵. حوزه متغیرها

آشنایی با مفهوم حوزه متغیرها و نحوه دسترسی به آنها در توابع.

## ۳۶. وارد کردن ماژول‌ها

نحوه وارد کردن و استفاده از ماژول‌ها در برنامه‌ها.

## ۳۷. ایجاد ماژول شخصی

ساخت ماژول‌های شخصی برای استفاده در سایر برنامه‌ها.

## ۳۸. آموزش نحوه کار با فایل‌ها

مقدمه‌ای درباره نحوه کار با فایل‌ها در پایتون.

## ۳۹. بازکردن و خواندن فایل‌های متنی

استفاده از دستورات پایه برای باز کردن و خواندن اطلاعات از فایل‌های مت.

## ۴۰. آموزش نحوه وارد کردن ماژول‌ها





در اینجا یاد می‌گیریم چگونه از ماژول‌های آماده استفاده کنیم تا بتوانیم توسط آن‌ها، برنامه‌های مان را بهبود ببخشیم.

## ۴۱. ایجاد ماژول شخصی

با ایجاد ماژول‌های شخصی، برنامه‌هایمان را به قدرت و انعطاف بیشتری مجهز خواهیم کرد.

## ۴۲. نحوه کار کردن با فایل‌ها

آموزش نحوه مدیریت فایل‌ها در پایتون و انجام عملیات خواندن و نوشتن.

## ۴۳. بازکردن و خواندن فایل‌های متنی

آموزش خواندن محتوای فایل‌های متنی و استفاده از آن در برنامه‌ها.

## ۴۴. استفاده از حلقه برای خواندن فایل‌های متنی

آموزش نحوه کار با حلقه‌ها برای خواندن متوالی فایل‌ها و انجام عملیات.

## ۴۵. نوشتن در یک فایل متنی

نحوه نوشتن و ذخیره اطلاعات در فایل‌های متنی به صورت دائمی.

## ۴۶. بازکردن و خواندن فایل‌های متنی با **buffer size**

بهینه‌سازی عملیات خواندن فایل با استفاده از **buffer size**.

## ۴۷. بازکردن و خواندن و نوشتن فایل‌های باینری

آموزش مدیریت فایل‌های باینری و انجام عملیات خواندن و نوشتن در آنها.



## ۴۸. حذف و تغییر نام فایل‌ها

آموزش حذف و تغییر نام فایل‌ها با استفاده از پایتون.

## ۴۹. لامبدا (**Lambda**) در پایتون

مفهوم لامبدا و کاربردهای آن در پایتون.

## ۵۰. چگونگی استفاده لامبدا در پایتون

نحوه استفاده از لامبدا در تعریف توابع کوتاه و سریع.

## ۵۱. لامبدهای پایتون به همراه نگاشت، فیلتر و کاهش

شناخت کاربردهای لامبدا در انجام عملیات نگاشت، فیلتر و کاهش بر روی داده‌ها.

## ۵۲. فایده عملی لامبدا

توجیه استفاده از لامبدا در مواقع خاص برنامه‌نویسی.

## ۵۳. مسائلی که هنگام استفاده از لامبدا باید در نظر داشت

نکات مهم و احتیاط‌هایی که در زمان استفاده از لامبدا باید مدنظر قرار داد.

## ۵۴. شیء‌گرایی

مفهوم اصلی شیء‌گرایی و چگونگی پیاده‌سازی آن در برنامه‌نویسی.

## ۵۵. مفهوم کلاس در شیء‌گرایی



تعریف و کاربردهای کلاس در برنامه‌نویسی شیء‌گرا.

## ۵۶. مفهوم نمونه‌سازی در شیء‌گرایی

آشنایی با نحوه ایجاد نمونه از یک کلاس و استفاده از آن.

## ۵۷. آشنایی با چهار اصل برنامه‌نویسی شیء‌گرا

یادگیری اصول اصلی Encapsulation، Abstraction، Inheritance و Polymorphism در برنامه‌نویسی شیء‌گرا.

## ۵۸. تعریف کلاس در پایتون

گام‌های ایجاد یک کلاس در پایتون و تعیین ویژگی‌ها و متدها.

## ۵۹. ساخت شیء در پایتون

نحوه ایجاد شیء از یک کلاس و استفاده از آن در برنامه.

## ۶۰. متد سازنده در پایتون

تعریف و استفاده از متد سازنده (Constructor) در کلاس‌های پایتون.

## ۶۱. ارث‌بری در پایتون

کاربرد و نحوه استفاده از ارث‌بری در پایتون.

## ۶۲. ارث‌بری در پایتون

آشنایی با مفهوم ارث‌بری و چگونگی به اشتراک گذاشتن ویژگی‌ها و متدها بین کلاس‌ها در پایتون.



## ۶۳. وراثت چندگانه در پایتون

نحوه استفاده از وراثت چندگانه برای ارث‌بری از چند کلاس و به اشتراک گذاشتن ویژگی‌ها.

## ۶۴. وراثت چند سطحی در پایتون

گسترش وراثت به چند سطح و ساختارهای پیچیده‌تر در برنامه‌نویسی شیء‌گرا.

## ۶۵. کپسوله‌سازی در پایتون

مفهوم کپسوله‌سازی و استفاده از آن برای محافظت از ویژگی‌ها و متدها در کلاس‌ها.

## ۶۶. ساخت برنامه و ابزار گرافیکی با پایتون

معرفی به ساخت برنامه‌ها و ابزارهای گرافیکی با استفاده از پایتون.

## ۶۷. رابط کاربری گرافیکی (GUI)

آشنایی با مفاهیم و اصول رابط کاربری گرافیکی و نحوه ساخت آنها با استفاده از پایتون.

## ۶۸. کتابخانه تکینتر (Tkinter) چیست؟

معرفی به کتابخانه Tkinter برای ساخت رابط‌های گرافیکی در پایتون.

## ۶۹. ایجاد برنامه با تکینتر

گام به گام ساخت یک برنامه ساده با استفاده از Tkinter.

## ۷۰. ابزارک‌های تکینتر



آشنایی با ابزارک‌ها و استفاده از آن‌ها در رابط‌های گرافیکی.

## ۷۱. کلاس ابزارک‌های تکینتر

نحوه ایجاد کلاس‌های ابزارک در Tkinter و تنظیمات مربوط به آن‌ها.

## ۷۲. مدیریت هندسه ابزارک‌های تکینتر

آشنایی با مفهوم مدیریت و ارتباط ابزارک‌ها با یکدیگر در تکینتر.

## ۷۳. ساخت چند ابزار با تکینتر

ایجاد چند ابزار مختلف در یک صفحه گرافیکی با استفاده از Tkinter.

## ۷۴. پروژه و پیاده‌سازی

پیاده‌سازی یک پروژه برنامه‌نویسی با تمامی مفاهیم آموخته شده.

