



عنوان مقاله: آشنایی با مفاهیم مهم پایتون : متغیرها، لیست ها و شرط ها

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۲۱ دی ۱۴۰۲

منبع: <https://nikamooz.com/important-concepts-of-python/>

مفاهیم پایتون به گونه‌ای هستند که شما می‌توانید با کمی تمرین و آموزش، آن‌ها را بیاموزید. در مقالات پیشین، به [معرفی زبان برنامه نویسی پایتون](#) و [نحوه نصب و آغاز کار با پایتون](#) و [کاربردهای Python در هوش مصنوعی](#) پرداختیم. در این مطلب، قصد داریم سه مورد از کلیدی‌ترین مفاهیم پایتون، شامل متغیرها (Variables)، لیست‌ها و عبارات شرطی (Condition Statements) را مورد بررسی قرار دهیم.

مفاهیم پراهمیت در پایتون کدامند؟

در زبان برنامه نویسی پایتون مفاهیم پراهمیتی همچون کنترل جریان، [توابع لامبدا](#) (Lambda)، دیکشنری، برنامه‌نویسی هم‌روند / غیرهم‌روند، کتابخانه‌های یادگیری ماشین (Machine Learning)، توسعه وب و سایر موارد وجود دارند. یادگیری این مفاهیم پایتون، در رشد و توسعه مهارت برنامه‌نویسی شما نقش کلیدی ایفا می‌کنند. برای آشنایی بیشتر با Python، پیشنهاد می‌شود [مقاله آینده شغلی زبان برنامه نویسی پایتون](#) را نیز مطالعه کنید.

متغیرها در پایتون

متغیر (Variable)، یک نام‌گذاری نمادین است که به یک مقدار (Value) خاص داده می‌شود. زبان برنامه نویسی پایتون از [زبان‌های دینامیک تایپ](#) (Dynamic Typed) به شمار می‌رود؛ این یعنی، لازم نیست نوع (Type) یک متغیر به صورت صریح (Explicit) تعیین شود. زبان پایتون از نوع‌های داده مختلفی از جمله عدد صحیح (Integer)، عدد ممیزشمار (Float)، رشته (String)، لیست (List)، [تاپل](#) (Tuple) و دیکشنری (Dictionary) پشتیبانی می‌کند. برای انتساب مقدار به یک متغیر، باید از علامت مساوی (=) استفاده شود. به‌عنوان مثال، $x = 10$ به این معنا است که متغیر x دارای مقدار ۱۰ است.

نکات مهم متغیرها در پایتون

متغیرها، یکی از پراهمیت‌ترین مفاهیم پایتون محسوب می‌شوند. در ادامه، به نکاتی درخصوص متغیرها در پایتون اشاره می‌شود.

- **امکان مقداردهی مجدد:** شما می‌توانید مقدار یک متغیر را با انتساب یک مقدار جدید تغییر دهید.
- **مقداردهی همزمان چند متغیر:** امکان مقداردهی چند متغیر در یک خط وجود دارد. به‌عنوان مثال، اگر شما $a, b, c = 1, 3, 7$ را در یک خط دستور بنویسید، متغیرها هرکدام به‌صورت $a = 1$ ، $b = 3$ و $c = 7$ مقداردهی خواهند شد.
- **Convention مقادیر ثابت:** معمولاً برای تعریف متغیرهای ثابت، از حروف بزرگ انگلیسی استفاده می‌شود. البته این موضوع به ترجیحات شخص باز می‌گردد.
- **نمایش مقدار متغیر:** برای چاپ مقدار متغیر، لازم است از تابع Print استفاده کنید. فرض کنید بخواهید مقدار متغیر x که عدد ۸ است را در خروجی چاپ کنید. در چنین شرایطی، با نوشتن $print(x)$ ، عدد ۸ در خروجی نمایش داده خواهد شد.
- **طول عمر و Scope متغیر:** Scope یا دامنه متغیر نشان می‌دهد که چگونه می‌توان به آن متغیر دسترسی داشت. ضمن اینکه یک متغیر تا زمانی قابل استفاده است که در حافظه وجود داشته باشد. بنابراین، طول عمر متغیر در پایتون به این موضوع بستگی دارد.
- **تبدیل نوع داده:** می‌توان از توابع پایتون، شامل $int()$ ، $float()$ ، $str()$ و سایر موارد به منظور **تبدیل نوع** (Type Conversion) بهره برد.



قوانین نامگذاری متغیرها در پایتون

- نام متغیر باید با یک حرف یا کاراکتر Underscore آغاز شود.
- نام متغیر نباید با یک عدد شروع شود.
- نام متغیر می‌تواند فقط حروف انگلیسی (A-Z)، اعداد و کاراکتر _ را شامل شود.
- نام متغیرها در پایتون، **حساس به حروف بزرگ و کوچک** (Case-Sensitive) است. (این یعنی، OFF ، off و Off سه متغیر مختلف به حساب می‌آیند).
- نام متغیر نباید از **کلمات کلیدی رزرو شده پایتون** (Python Keywords) باشد.

لیست در پایتون

لیست در پایتون به‌عنوان یک ساختمان داده کاربردی به طور گسترده استفاده می‌شود. لیست در پایتون، مجموعه‌ای از آیتم‌های مرتب‌شده است. توجه شود که آیتم‌های لیست، می‌توانند **نوع های داده** (Data Types) مختلفی باشند. برای ایجاد لیست در پایتون، باید از نماد براکت [] استفاده شود. برای درک بهتر، به لیست زیر توجه کنید.

```
mylist = [5, 52, 36, "Peach", "banana", False]
```

نکات مهم لیست در پایتون

مشابه متغیرها، لیست نیز یکی از کاربردی‌ترین مفاهیم پایتون به شمار می‌رود. نکات کلیدی کار با لیست در Python در ادامه فهرست شده‌اند.

- **دسترسی به عناصر لیست:** برای دسترسی به عناصر لیست در پایتون، از ایندکس‌گذاری (Indexing) استفاده می‌شود. توجه شود که ایندکس از عدد صفر آغاز می‌شود؛ بنابراین، به طور پیش‌فرض ایندکس اولین عنصر، صفر است، مگر اینکه به طور صریح اشاره شود که ایندکس‌گذاری باید از یک آغاز شود. ایندکس منفی در پایتون به آخرین عنصر از لیست اطلاق می‌شود. به مثال زیر دقت کنید.

```
first = mylist[0]
last = mylist[-1]
```

در این مثال، مقدار اولین عنصر mylist و آخرین عنصر آن، به ترتیب در دو متغیر first و last قرار داده شده است.

- **بُرش لیست:** شما می‌توانید با به‌کارگیری **فرآیند Slicing**، بخشی از یک لیست در پایتون را استخراج کنید.

- **عملیات مربوط به لیست:** الحاق دو لیست (+ | Concatenation)، تکرار لیست (* | Reptition) و len() یا همان دریافت عناصر لیست، همگی از کلیدی‌ترین مفاهیم پایتون هستند و در لیست امکان استفاده از آن‌ها فراهم شده است. در مثال زیر، این موارد استفاده شده‌اند.

```
combinedlist = mylist + [4, 5]
repeatedlist = mylist * 2
length = len(mylist)
```

- **ویرایش لیست:** شما می‌توانید با کمک ایندکس‌گذاری، اموری همچون درج و حذف عناصر لیست را انجام دهید.
- **متدهای لیست:** Method های لیست، در اموری مانند مرتب‌سازی، کپی کردن، حذف و معکوس کردن لیست به کار می‌روند.
- **لیست تودرتو (Nested List):** لیست در پایتون می‌تواند شامل لیست‌هایی باشد. به مثال زیر توجه کنید.

```
nestedlist = [[1, 2, 3], [4, 5, 6]]
```



عبارات شرطی در پایتون

عبارات شرطی (Condition Statements) و آشنایی با نحوه کارکرد آن‌ها، حائز اهمیت فراوانی است. همان‌طور که از نام آن مشخص است، این عبارات به هدف بررسی شرط‌ها در برنامه‌نویسی به کار می‌روند و شما می‌توانید براساس شروط خاصی، جریان یک برنامه را کنترل کنید.

کاربرد عبارات شرطی در پایتون چیست؟

از میان انواع مفاهیم پایتون ، عبارات شرطی هر یک کاربردهایی دارند که در ادامه لیست شده‌اند:

- **تصمیم‌گیری:** عبارت شرطی در پایتون، به برنامه این قابلیت را می‌دهند که براساس شرایط خاص، کدهای خاصی را اجرا کند.
- **رسیدگی به سناریوهای پیچیده:** عبارات شرطی برای ساخت نرم‌افزارهای تطبیق‌پذیر و قدرتمندی مناسب است که قابلیت پاسخ‌دهی به شرایط مختلف را دارند.
- **کنترل جریان:** Condition Statement ها با نظارت بر جریان و روند برنامه، تعیین می‌کنند که کدام بلوک‌های کد باید اجرا شوند.
- **رسیدگی به خطا:** عبارات شرطی در رسیدگی به خطاها و Exception ها استفاده می‌شوند و به برنامه‌نویسان این امکان را می‌دهند که شرایط غیرمنتظره را مدیریت و پیش‌بینی کنند.
- **منطق الگوریتمی:** در طراحی الگوریتم، عبارات شرطی در شرح عملیات منطقی و اتخاذ تصمیمات، نقش کلیدی دارند.
- **فیلترینگ داده‌ها:** می‌توان با بهره‌مندی از عبارات شرطی در پایتون، پردازش و فیلترینگ داده‌ها را انجام داد. به‌عنوان مثال، امکان مرتب‌سازی و ویرایش داده‌ها براساس شرایط خاص وجود دارد.
- **رسیدگی به رخدادها:** در برنامه‌نویسی مبتنی بر رخداد (Event)، عبارات شرطی به منظور پاسخ‌گویی به Event ها یا Trigger ها مورد استفاده قرار می‌گیرند. این موضوع معمولاً در اپلیکیشن‌های GUI، توسعه وب و سایر سیستم‌های تعاملی کاربردی است.

عبارات شرطی یکی بنیادی‌ترین مفاهیم پایتون به شمار می‌روند و وجود آن‌ها، مزیت‌های فوق‌را به همراه دارد. در عمل، این مفاهیم پایتون در بخش‌های مختلف برنامه‌نویسی مورد استفاده قرار می‌گیرند. با این دیدگاه مقدماتی در بخش بعدی مطلب، به بررسی انواع عبارات شرطی پایتون می‌پردازیم.

انواع عبارات شرطی در پایتون

انواع Condition Statement ها در پایتون عبارتند از:

- **عبارت شرطی If:** به واسطه این دستور پایتون ، اگر یک شرط خاص صدق کند، بلوک کد مشخصی اجرا خواهد شد.
- **عبارت شرطی If Else:** در If Else اگر یک شرط صحیح باشد، کد مربوط به آن اجرا می‌شود و در صورتی که شرط برابر با False باشد، یک بلوک کد دیگر اجرا خواهد شد. مثال زیر، کاربرد این مفهوم پایتون را نشان می‌دهد.

```
Temperature = 27
if Temperature > 30:
    print("It's a hot day!")
else:
    print("It's not too hot today.")
```

در مثال فوق، متغیر Temperature با عدد ۲۷ مقداردهی شده است. حال به کمک یک عبارت شرطی If Else می‌توان بررسی کرد که آیا هوا گرم است یا خیر. اگر دمای هوا (متغیر Temperature) بیش‌تر از ۳۰ باشد، گرم‌بودن هوا از طریق تابع print در خروجی مشخص می‌شود. در غیر این صورت، یک متن در خروجی تحت عنوان گرم‌نبودن هوا چاپ خواهد شد.

- **عبارت شرطی If Elif Else:** اگر بخواهید چندین شرط را به ترتیب بررسی کنید، می‌توانید عبارت شرطی if...elif...else را به کار ببرید. در مثال زیر، مقدار متغیر Grade بررسی می‌شود و طبق آن، حرف انگلیسی مرتبط با شرایط در خروجی چاپ خواهد شد.

```
grade = 75
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
elif grade >= 60:
    print("D")
else:
    print("F")
```

چون Grade = 75 است، بنابراین، در خروجی حرف C چاپ خواهد شد. این یعنی، شرط Grade >= 70 صدق می‌کند و سایر شرط‌های بالایی آن، قابل قبول نیستند. این نوع از Condition Statement از کاربردی‌ترین مفاهیم پایتون به شمار می‌رود و برای مواقعی مناسب است که بخواهید با بررسی دنباله‌ای از شرط‌ها، بلوک کد مربوط به اولین شرط صحیح اجرا شود.

- **عبارت شرطی Nested If:** در زبان پایتون، این امکان وجود دارد که عبارات If را به صورت تودرتو (Nested) به کار برد و از آن در شرط‌های پیچیده‌تر بهره‌مند شد. به منظور درک بهتر، به کد زیر توجه کنید:

```
x = 10
y = 5
if x > 0:
    print("x is positive")
    if y > 0:
        print("y is also positive")
    else:
        print("y is not positive")
else:
    print("x is not positive")
```

معرفی مفاهیم اولیه پایتون

مفاهیم پایتون به دلیل نحو ساده و قابل درک این زبان برنامه‌نویسی، به راحتی قابل یادگیری هستند. در این مقاله، به بررسی متغیرها، لیست‌ها و عبارات شرطی و نحوه کارکرد آن‌ها در پایتون پرداخته شد. شما می‌توانید با یادگیری مفاهیم پایتون در مسیر شغلی توسعه‌دهنده پایتون قرار بگیرید و مهارت‌های خود را به صورت حرفه‌ای استفاده کنید.