



اگر به خاطر داشته باشید در بخش‌های ابتدایی آموزش پایتون به شما گفتیم هر برنامه کاربردی با هدف ساختن، خواندن، به‌روزرسانی و حذف داده‌ها ساخته می‌شود. اما پیش از آن که بتوانید تغییری روی داده‌ها اعمال کنید، ابتدا باید بتوانید آن‌ها را ذخیره کرده و مهم‌تر از آن بتوانید داده‌ها را در برنامه خود فراخوانی کرده و به شکل موقت در حافظه نگه‌داری کنید. هر زبان برنامه‌نویسی داده‌ها را بر مبنای قوانین خاص خود دریافت کرده و به ما اجازه می‌دهد آن‌ها را ویرایش کنید.

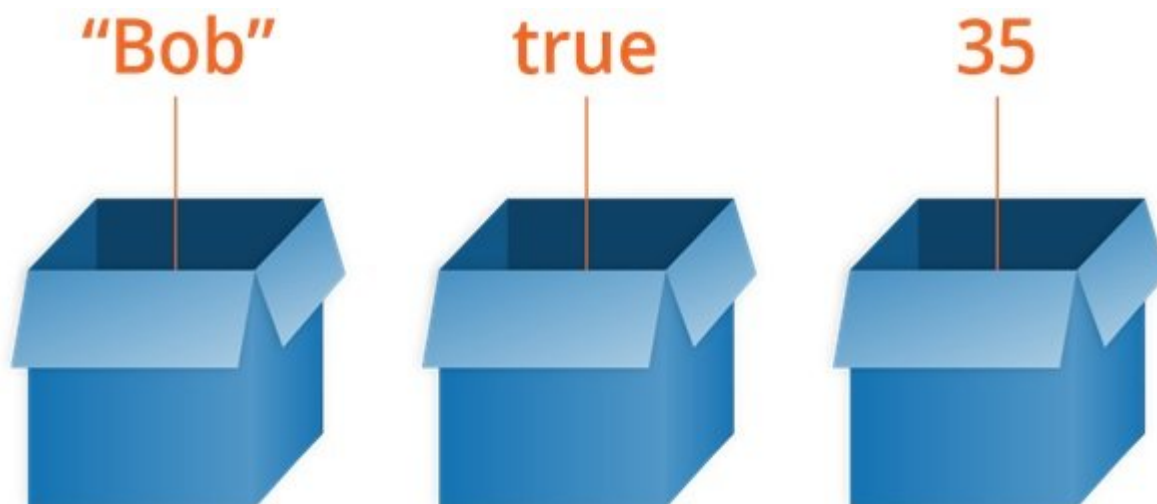
برای مطالعه بخش پنجم آموزش رایگان پایتون [اینجا](#) کلیک کنید

ذخیره‌سازی اطلاعات

یک برنامه کاربردی برای انجام درست وظایف خود باید به شکلی به اطلاعات دسترسی داشته باشد. برنامه‌های کاربردی اطلاعات را درون حافظه اصلی ذخیره‌سازی می‌کنند. اما حافظه اصلی به شکل موقت می‌تواند اطلاعات را نگه‌داری کند. زمانی که شما کامپیوتر خود را خاموش می‌کنید، همه اطلاعات درون حافظه اصلی پاک خواهند شد، در نتیجه پیش از خاموش کردن کامپیوتر باید اطلاعات را روی یکی از رسانه‌های ذخیره‌ساز همچون هارددیسک، فلش درایو یا حافظه‌های حالت جامد ذخیره‌سازی کنید. نکته مهمی که باید به آن دقت کنید به نوع اطلاعات باز می‌گردند. اطلاعات همگی شکل یکسانی ندارند. برخی از اطلاعات عددی هستند، در حالی که برخی دیگر متنی هستند. در نتیجه پیش از ویرایش داده‌ها مجبور هستید درباره انواع مختلفی از داده‌ها اطلاعاتی کسب کنید. اما پیش از این کار باید با مبحث مهمی به نام متغیرها آشنا شوید. متغیرها یکی از ارکان اصلی و مهم دنیای برنامه‌نویسی هستند.

متغیرها، جعبه‌هایی برای ذخیره‌سازی داده‌ها

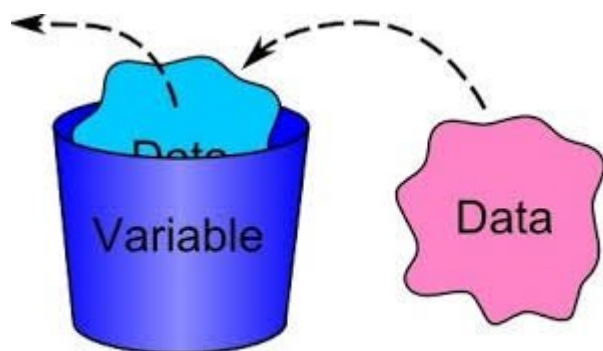
هنگام کار با برنامه‌ها، اطلاعات درون متغیرها ذخیره‌سازی می‌شوند. اما متغیر چیست؟ متغیر شبیه به یک جعبه است. در دنیای واقعی شما برای جابه‌جایی اشیاء از جعبه‌هایی استفاده می‌کنید. به‌طور مثال در زمان خرید یک جفت کفش، فروشنده کفش‌ها را درون جعبه مخصوص آن‌ها قرار داده و به شما تحویل می‌دهد. در این سناریو جعبه نقش یک متغیر در دنیای برنامه‌نویسی و کفشی که درون آن قرار می‌گیرد نقش اطلاعاتی را دارد که درون یک متغیر قرار می‌گیرند.



در دنیای برنامه‌نویسی برای کار کردن با اطلاعات مجبور هستید از متغیرها استفاده کنید. متغیرها طرف‌هایی هستند که اطلاعات درون آن‌ها قرار می‌گیرند. ما از متغیرها برای مشاهده یا ویرایش اطلاعات استفاده می‌کنیم. اجازه دهید این مسئله را با ذکر مثالی روشن کنیم. فرض کنید، لیوانی در اختیار دارید که مقداری آب در آن قرار گرفته است. اما آب درون لیوان کثیف شده یا قابل آشامیدن نیست. در این حالت شما آب لیوان را خالی کرده و آب تمیزی درون لیوان می‌ریزید. در حقیقت شما در حال تغییر محتوای درون لیوان هستید. در دنیای برنامه‌نویسی نیز در اغلب موارد چنین وضعی حاکم است. زمانی که اطلاعات جدیدی به دست آورید، مجبور هستید اطلاعات درون یک متغیر را تغییر داده و ویرایش کنید. برای این منظور باید مقادیر جدید جایگزین مقادیر فعلی شوند. در دنیای برنامه‌نویسی هر متغیر تنها بخشی از اطلاعات را ذخیره‌سازی می‌کند.

از جعبه‌های درستی برای ذخیره‌سازی داده‌ها استفاده کنید

مردم در بیشتر موارد تمایل دارند که اشیاء اشتباهی را درون جعبه‌ها قرار دهند. به‌طور مثال، برخی از مردم یک جفت کفش را درون یک کیسه لباس یا یک قلم را درون یک جعبه کفش قرار می‌دهند! یک چنین اشتباهی ممکن است عجیب به نظر برسد، اما در دنیای برنامه‌نویسی افرادی که تازه به این وادی قدم گذاشته‌اند مشابه چنین اشتباهی را مرتکب می‌شوند.



خوشبختانه، پایتون در این زمینه عملکرد شفافی و کمک فراوانی به نویسندگان می‌کند. این شفافیت به شما اجازه می‌دهد اعداد ذخیره شده در یک متغیر و متن ذخیره شده در متغیر دیگری را به شکل متمایز از یکدیگر دریافت کرده یا مشاهده کنید. توجه داشته باشید که هر متغیر برای ذخیره‌سازی نوع خاصی از اطلاعات استفاده می‌شود. در مورد جزئیات مربوط به متغیرها نگران نباشید، در مقاله‌های آتی اطلاعات بیشتری در مورد آن‌ها به دست خواهید آورد، اما اکنون این نکته را در نظر داشته باشید که هر نوع داده‌ای در متغیر خاصی ذخیره‌سازی می‌شود. پایتون از متغیرهای خاص برای ذخیره‌سازی اطلاعات استفاده می‌کند تا کار برنامه‌نویسان ساده شده و اطمینان حاصل کند که اطلاعات به شیوه ایمن ذخیره‌سازی می‌شود.

کامپیوترها تنها صفرها و یکها را می‌شناسند

کامپیوترها هیچ‌گونه آگاهی ندارند که چه اطلاعاتی در اختیار آنها قرار می‌دهید، آنها تنها درباره صفرها و یک‌هایی که ولتاژ را تغییر می‌دهند، اطلاع دارند. اگر از صفرها و یکها یک سطح بالاتر بروید، آن‌گاه مشاهده می‌کنید که کامپیوترها می‌توانند با اعداد کار کنند. با این حال، اعداد، کاراکترها، تاریخ‌ها، زمان و هر نوع اطلاعاتی که تصور شما را می‌کنید همگی حول محور صفرها و یکها برای یک کامپیوتر قابل درک هستند. به‌طور مثال، زمانی که کاراکتر A را روی صفحه‌کلید فشار می‌دهید، انتظار دارید که روی صفحه‌نمایش کاراکتر A را مشاهده کنید، اما در پشت صحنه کاراکتری که تایپ کرده‌اید در سیستم دودویی به شکل 01000001 و در سیستم دهگان به شکل 65 ذخیره می‌شود. دقت کنید که یک کاراکتر A یا یک مقدار تاریخ همچون 1/28/2019 هیچ‌گونه معنایی برای کامپیوتر ندارد.

تعریف نوع‌های داده‌ای پایتون

هر زبان برنامه‌نویسی از متغیرها برای ذخیره‌سازی انواع مختلفی از اطلاعات استفاده می‌کند. **پایتون** نیز از این قاعده مستثنا نیست. اما متغیر تنها یک اصطلاح است. برای آن‌که متغیرها معنا و مفهوم خاصی داشته باشند باید ماهیت آن‌ها مشخص شود. به‌طور مثال، در دنیای واقعی شما نمی‌توانید از واژه جعبه استفاده کنید و انتظار داشته باشید مردم معنای واژه شما را بدانند. اما زمانی که می‌گویید جعبه کفش‌ها، جعبه ابزار و... در حقیقت درباره نوع جعبه‌ها در حال صحبت کردن هستید. در دنیای برنامه‌نویسی نیز برای مشخص کردن نوع متغیرها از اصطلاحی استفاده می‌کنیم که نوع داده ای (data type) نامیده می‌شود. آگاهی در مورد نوع داده‌ای یک متغیر حائز اهمیت است، زیرا به شما می‌گوید که چه نوع اطلاعاتی درون متغیر قرار داشته یا چه نوع اطلاعاتی را درون یک متغیر می‌توانید قرار دهید. همچنین، زمانی که می‌خواهید اطلاعاتی در یک متغیر ذخیره کنید، شما باید درباره نوع داده‌ای آن متغیر اطلاع دقیقی داشته باشید. **پایتون** به شما اجازه نمی‌دهد یک متن را در متغیری قرار دهید که برای ذخیره‌سازی اعداد تعریف شده است. (مثال قلم در جعبه کفش را به یاد آورید!) اگر چنین کاری را انجام دهید نه تنها اطلاعات شما مخدوش می‌شوند، بلکه مشکلاتی نیز در برنامه شما به وجود خواهد آمد. در دنیای **پایتون** شما با نوع‌های داده‌ای عددی، رشته‌ای و بولین سروکار دارید. نوع‌های داده‌ای هر یک تعاریف خاصی دارند که در شماره‌های آتی با تعریف هر یک از آن‌ها آشنا خواهیم شد.

قرار دادن اطلاعات درون متغیرها

برای آن‌که بتوانید درون هر متغیری مقداری را قرار دهید، باید از کاراکتری استفاده کنید که مقدار شما را به یک متغیر اختصاص دهد. در برنامه‌نویسی برای اختصاص یک مقدار به یک متغیر از عملگر تخصیص‌دهنده = استفاده می‌کنیم. نحوه به‌کارگیری این عملگر چگونه است؟

1. محیط IDLE را باز کنید.

2. فرض کنید در نظر دارید عدد 5 را درون متغیری به نام myVar قرار دهید.

3. در پنجره اصلی **پایتون** برای اختصاص مقدار فوق به این متغیر از ترکیب نحوی زیر استفاده کرده و کلید اینتر را فشار دهید.

```
myVar=5
```

دقت کنید در زمان انتصاب مقداری به یک متغیر، **پایتون** هیچ‌گونه اطلاعاتی در اختیاران قرار نمی‌دهد، اما متغیر شما درون بخشی از حافظه قرار گرفته و مقدار 5 درون آن قرار گرفته است. از این پس شما می‌توانید با تایپ نام این متغیر از آن استفاده کنید.

4. اکنون myVar را تایپ کرده و کلید اینتر را فشار دهید. مشاهده می‌کنید که **پایتون** مقدار 5 را به شما نشان می‌دهد.

```
>>> myVar=5
```

```
>>> myVar
```

```
5
```

```
>>> |
```

نوع‌های عددی

انسان‌ها در ارتباط با اعداد یک دیدگاه کلی دارند. ما 1 و 1.0 را به عنوان اعداد یکسانی تصور می‌کنیم و به راحتی از این دو مقدار به جای یکدیگر استفاده می‌کنیم، در حالی که مقدار دوم دارای یک بخش اعشار است. برای **پایتون** مقادیر 1 و 1.0 دو مقدار متفاوت از هم هستند و فرآیند پردازش آن‌ها متفاوت از دیگری خواهد بود. در دنیای برنامه‌نویسی **پایتون** شما با انواع مختلفی از اعداد همچون مقادیر صحیح، اعشاری (نقطه شناور) و اعداد مختلط سروکار خواهید داشت.

در شماره آینده آموزش **پایتون** اطلاعات بیشتری در ارتباط با مقادیر عددی به دست خواهیم آورد.

برای مطالعه تمام قسمت‌های منتشر شده آموزش‌های رایگان پایتون اینجا کلیک کنید

لطفا نظرات خود در مورد این آموزش و ادامه آن را در بخش دیدگاه در انتهای صفحه اعلام نمایید و نظارت سایر کاربران را نیز ببینید.

معرفی کانال آموزش برنامه‌نویسی و پایتون:

برنامه‌نویسی | پایتون: [@Python_0to100](https://t.me/Python_0to100)

(کانال سری آموزش‌های رایگان پایتون)

آموزش پایتون: [@learnpy](https://t.me/learnpy)

(آموزش پایتون با فلش کارت)

تاریخ انتشار:

09 بهمن 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/programming/14514/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-python-%E2%80%93%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D9%85%D9%81%D9%87%D9%88%D9%85->

%D9%85%D8%AA%D8%BA%DB%8C%D8%B1%D9%87%D8%A7