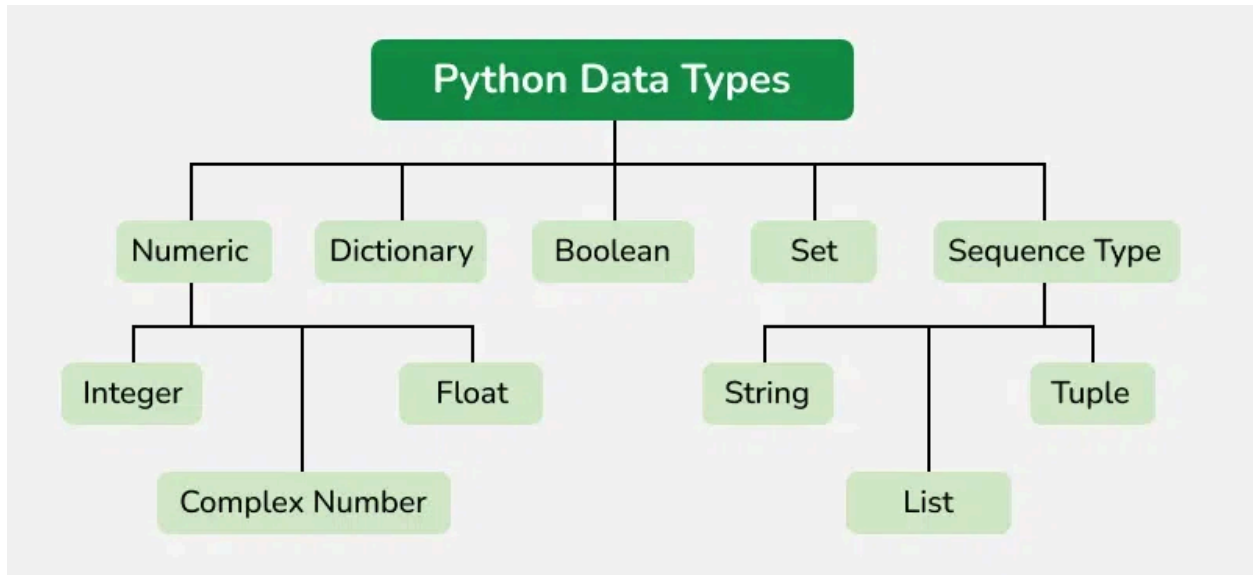


Variables

متغیرها برای ذخیره داده‌ها استفاده می‌شوند و پایتون به طور خودکار نوع آنها را تعیین می‌کند.

نکته! توجه داشته باشید که در تصویر زیر، انواع مختلفی از داده‌ها نمایش داده شده‌اند. در حال حاضر، تنها با تعداد محدودی از آنها کار می‌کنیم، اما در طول ترم به تدریج سایر انواع داده را نیز بررسی خواهیم کرد.



```
1 | # Storing different types of data
2 | age = 25          # Integer
3 | height = 5.8     # Float
4 | is_student = True # Boolean
5 | name = "John"    # String
6 |
7 | # Printing the variables
8 | print(age, height, is_student, name)
```

متغیرها با تایپ مشخص (Explicit Typing)

در پایتون نیازی به مشخص کردن نوع متغیر نیست، اما می‌توانیم با استفاده از *type hinting* نوع داده‌ها را شبیه C نشان دهیم.

مفهوم **Type Hinting** در پایتون روشی برای مشخص کردن نوع داده‌هایی است که یک متغیر، ورودی یا خروجی تابع باید داشته باشد. این قابلیت در نسخه‌های جدیدتر پایتون (از نسخه 3.5 به بعد) معرفی شد تا به افزایش خوانایی، درک بهتر کد، و کاهش خطاهای احتمالی کمک کند.

هرچند که پایتون یک زبان **داینامیک تایپ** (Dynamic Typing) است و به‌طور پیش‌فرض نوع داده‌ها را در زمان اجرا تعیین می‌کند، با استفاده از **Type Hinting** می‌توان به‌صورت اختیاری نوع داده‌ها را تعریف کرد.

برای مثال می‌توان نوع متغیرها را با استفاده از : مشخص کرد:

```
1 # Explicitly specifying types (like in C)
2 a: int = 10 # Integer
3 b: float = 3.14 # Float
4 c: bool = True # Boolean
5 d: str = 'a' # String
6
7 print(a, b, c, d)
```

گرفتن تایپ متغیرها

برای بدست آوردن نوع داده یک متغیر در پایتون، می‌توانید از تابع داخلی `type()` به صورت زیر استفاده کنید.

مثال : پرینت کردن تایپ متغیرها

```
1 x = "Zara"
2 y = 10
3 z = 10.10
4
5 print(type(x)) # <class 'str'>
6 print(type(y)) # <class 'int'>
7 print(type(z)) # <class 'float'>
```

متغیرهای پایتون - تخصیص چند گانه

پایتون این امکان را به شما می‌دهد که چندین متغیر را در یک دستور واحد مقداردهی اولیه کنید.

مثال : تخصیص چند گانه

```
1 | a, b, c = 5, 10, "hi"
2 | print(a) # output: 5
3 | print(b) # output: 10
4 | print(c) # output: "hi"
```

روش ساده برای Swap دو متغیر در پایتون:

در پایتون، جابجا کردن دو متغیر (swap) بسیار ساده است و نیازی به استفاده از متغیر موقت یا روش‌های پیچیده ندارید.

پایتون اجازه می‌دهد که به راحتی دو متغیر را بدون استفاده از متغیر کمکی جابجا کنید. کافی است از سینتکس خاص پایتون استفاده کنید:

```
1 | # Before swapping
2 | a = 5
3 | b = 10
4 |
5 | # Swapping the values
6 | a, b = b, a # This swaps the values of a and b
7 |
8 | # After swapping
9 | print("a =", a)
10 | print("b =", b)
```

رشته‌ها (Strings) در پایتون:

در پایتون، **رشته (String)** یک نوع داده است که برای ذخیره کردن متن استفاده می‌شود. رشته‌ها در پایتون یک توالی از کاراکترها هستند که می‌توانند شامل حروف، اعداد، فاصله‌ها، یا هر نوع کاراکتر دیگری باشند.

تعریف رشته در پایتون:

رشته‌ها معمولاً داخل "علامت نقل قول (quotation marks)" قرار می‌گیرند:

- علامت نقل قول تک (') یا علامت نقل قول دوتایی (") برای تعریف رشته‌ها استفاده می‌شود.

```
1 | print("Hello")
2 | print('Hello')
```

- شما می‌توانید یک رشته چندخطی را به یک متغیر تخصیص دهید با استفاده از سه علامت نقل قول:

```
1 | a = """Lorem ipsum dolor sit amet,
2 |   consectetur adipiscing elit,
3 |   sed do eiusmod tempor incididunt ut
4 |   labore et dolore magna aliqua."""
5 | print(a)
```

- برای گرفتن یک کاراکتر از یک رشته به شکل زیر عمل می‌کنیم (گرفتن کاراکتر در پوزیشن دوم)

```
1 | a = "salam"
2 | ch = a[1]
3 | print(ch) # output: a
4 | print(type(ch)) # <class 'str'>
5 | print(type(a)) # <class 'str'>
```

- در پایتون، نوع داده‌ی مستقلی به نام "کاراکتر" وجود ندارد؛ بلکه هر کاراکتر، در واقع یک رشته‌ای با طول ۱ محسوب می‌شود.

- اگر نیاز به کاراکترهای یونیکد داشته باشید، پایتون از UTF-8 به صورت پیش‌فرض پشتیبانی می‌کند.

استاندارد UTF-8 یک روش کدگذاری کاراکترها است که برای نمایش متن در کامپیوترها و اینترنت استفاده می‌شود. این استاندارد می‌تواند همه‌ی کاراکترهای موجود در زبان‌های مختلف دنیا (مانند فارسی، انگلیسی، عربی، چینی و غیره) را پشتیبانی کند.

```
1 | `ch = 'پ'
2 | print(ord(ch)) # output: 1662 (unicode of 'پ' is 1662)
```

نوع داده collection

در پایتون، اصطلاح "**collection**" به هر نوع داده‌ای اطلاق می‌شود که امکان نگهداری مجموعه‌ای از اشیاء یا مقادیر را فراهم می‌کند و می‌توان عملیات مختلفی روی آن‌ها انجام داد. این مفهوم شامل انواع مختلف داده‌هایی است که برای ذخیره‌سازی و مدیریت مجموعه‌ای از مقادیر به کار می‌روند.

انواع collection در پایتون:

۱. لیست‌ها (Lists)

۲. تاپل‌ها (Tuples)

۳. مجموعه‌ها (Sets)

۴. دیکشنری‌ها (Dictionaries)

در ادامه‌ی درس با این مفاهیم آشنا خواهیم شد؛ اما در حال حاضر تنها با لیست کار می‌کنیم. \ لیست (List) در پایتون یک ساختار داده‌ی مرتب و قابل تغییر است که می‌تواند مجموعه‌ای از عناصر با انواع داده‌ی مختلف را در خود نگه دارد.

```
1 | fruits = ["Apple", "Cherry", "Pineapple", 2]
2 | print(fruits) # output: ["Apple", "Cherry", "Pineapple", 2]
```

PYnative.com

List in Python

$$L = [20, 'Jessa', 35.75, [30, 60, 90]]$$

↑
L[0]

↑
L[1]

↑
L[2]

↑
L[3]

- ✓ **Ordered:** Maintain the order of the data insertion.
- ✓ **Changeable:** List is mutable and we can modify items.
- ✓ **Heterogeneous:** List can contain data of different types
- ✓ **Contains duplicate:** Allows duplicates data

نکته! مفاهیم collection و lists هفته‌ی بعدی بیشتر بررسی خواهند شد.

اطلاعات بیشتر در

https://www.w3schools.com/python/python_variables.asp