



# مبانی برنامه نویسی کامپیوتر

**مدرس: سعیده آل بویه**

کارشناس مهندسی عمران - نقشه برداری از دانشگاه تهران  
کارشناس ارشد مهندسی نقشه برداری - فتوگرامتری از دانشگاه تهران

نیمسال اول ۱۴۰۳-۱۴۰۲

دانشگاه صنعتی اراک

## مباحث مقدماتی در پایتون

- مقدمه (آشنایی با زبان برنامه نویسی پایتون، نصب پایتون)
- آشنایی با متغیرها، expression ها و statement ها در پایتون
- کار با Boolean ها در پایتون
- دستورات شرطی در پایتون (if...else)
- کار با توابع در پایتون
- حلقه تکرار while
- حلقه for

# مباحث مقدماتی در پایتون

- ساختارهای داده در پایتون
- کار با `string`ها و `list`ها در پایتون
- دیکشنری ها در پایتون
- کتابخانه ها در پایتون
- کتابخانه های استاندارد پایتون
- کار با فایل در پایتون
- فایل های `CSV` در پایتون

# مباحث پیشرفته در پایتون

- برنامه نویسی شیء گرا
- کار با دیتابیس ها در پایتون
- کاربرد پایتون در زمینه های مختلف (پروژه)

# پایتون مقدماتی

- Value vs. variable ?
- Variables
- Statements
- Expressions
- Boolean
- Control Structures: If...else

# Variables

# متغیرها

• تفاوت value با variable در چیست؟

• **متغیر در پایتون چیست؟**

اولین و پایه‌ای‌ترین مفهوم در یادگیری زبان برنامه‌نویسی پایتون آشنایی با متغیرها یا variable هاست.

متغیرها ساختارهایی هستند که برای ذخیره مقادیر در برنامه‌نویسی مورد استفاده قرار می‌گیرند.

**متغیر در پایتون** برای اشاره به مکان حافظه استفاده می‌شود.

## انواع داده

## Data Type

- رشته ها یا `string`ها : دنباله ای از حروف، مثل نام ها و عبارت ها و ...
  - اعداد صحیح یا `integer`ها : ساده ترین مقادیر عددی مورد استفاده در محاسبات
  - اعداد اعشاری یا `float`: برای محاسبات با دقت بالاتر
  - متغیرهایی که مقادیر آنها به صورت درست یا نادرست `True / False` است  
Boolean :
- ✓ Check type of variables with `type()`
- `type(۳۴) = int`

# Naming Rules

# نام‌گذاری متغیرها در پایتون

- نام متغیر باید با یک حرف یا کاراکتر زیرخط (اندرلاین) شروع شود
- نام متغیر نمی‌تواند با عدد شروع شود
- یک نام متغیر فقط می‌تواند شامل حروف، کاراکترهای عددی و آندرلاین باشد
- نام متغیرها به حروف کوچک و بزرگ حساس هستند
- از کلمات رزرو شده در پایتون (کلمات کلیدی) نمی‌توان برای نام‌گذاری متغیر استفاده کرد
- نام شناسه نباید حاوی هیچ فضای سفید یا کاراکتر خاصی باشد: @ \* %

bob    Bob    \_bob    \_2\_bob\_    bob\_2    BoB



# Naming conventions

# قراردادهای نام گذاری

قراردادهای پیشنهادی انجمن پایتون برای نام گذاری ها به شرح زیر است:

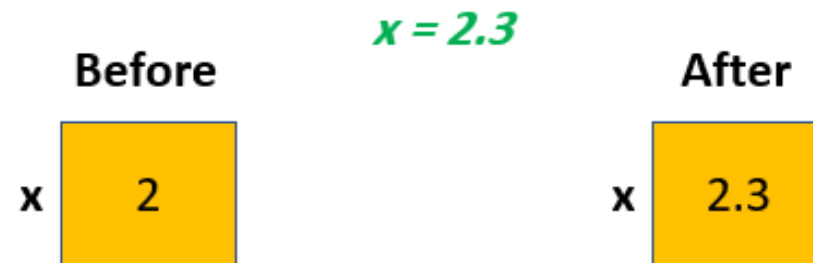
- `joined_lower` for functions, methods and, attributes
- `joined_lower` or `ALL_CAPS` for constants
- `StudlyCaps` for classes
- `camelCase` only to conform to pre-existing conventions
- Attributes: `interface`, `_internal`, `__private`

# Simple Assignment Statements

- *x* is a variable and 2 is its value
- *x* can be assigned different values; hence, it is called a variable

```
>>> x = 2
>>> print(x)
2
>>> x = 2.3
>>> print(x)
2.3
```

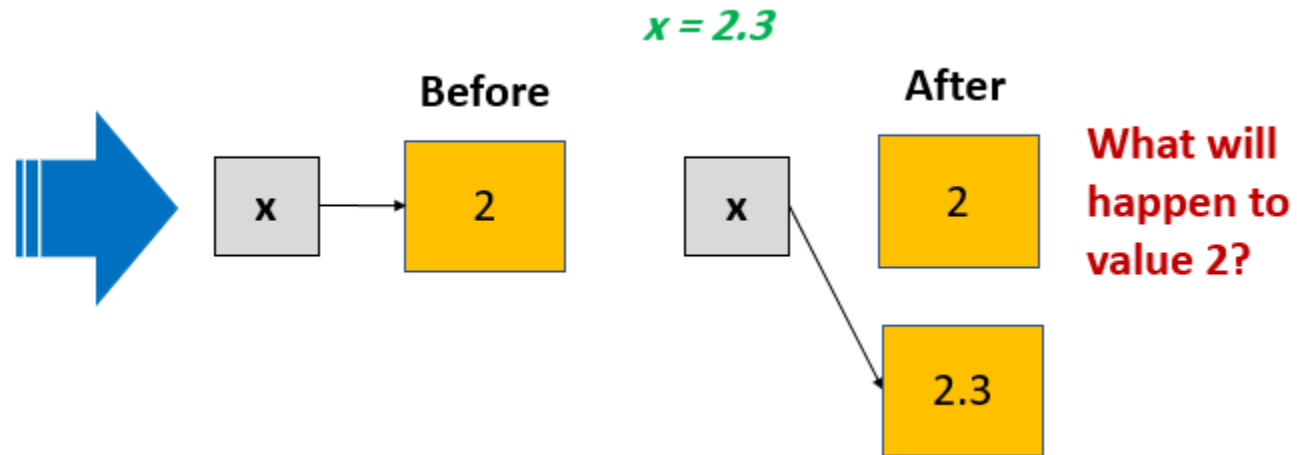
```
>>> x = 2
>>> print(x)
2
>>> x = 2.3
>>> print(x)
2.3
```



# Simple Assignment Statements

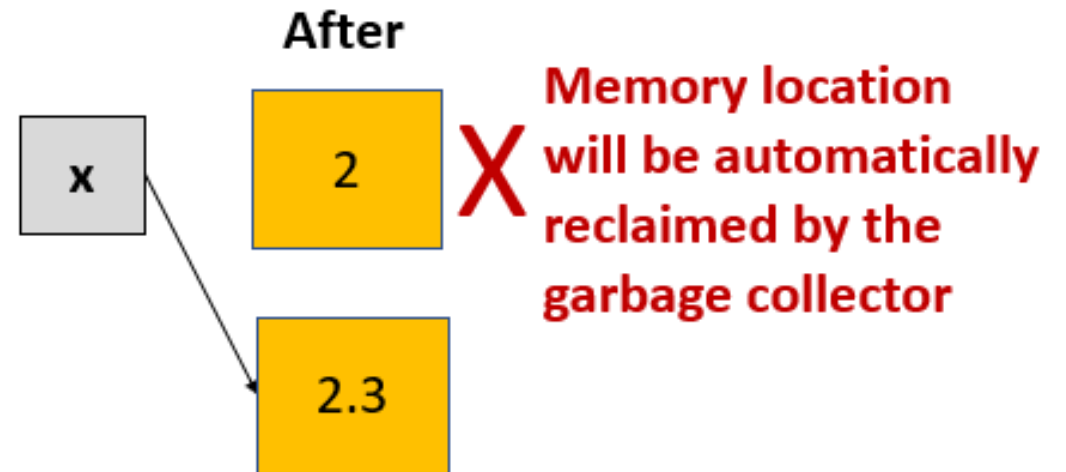
- Python assignment statements are actually slightly different from the “variable as a box” model
  - In Python, values may end up anywhere in memory, and variables are used to refer to them

```
>>> x = 2
>>> print(x)
2
>>> x = 2.3
>>> print(x)
2.3
```



# Simple Assignment Statements

- Interestingly, as a Python programmer you do not have to worry about computer memory getting filled up with old values when new values are assigned to variables
- Python will automatically clear old values out of memory in a process known as *garbage collection*



# Simultaneous Assignment

- Python allows us also to assign multiple values to multiple variables all at the same time

```
>>> x, y = 2, 3
>>> x
2
>>> y
3
>>>
```

# Exercise

**What are the reserved keywords in python?**

- **Deadline: 02. 08. 14**
- **Email: [s.alebooye@ut.ac.ir](mailto:s.alebooye@ut.ac.ir)**

# Statements, Operators, Expressions

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Float Division
**	Exponentiation
abs()	Absolute Value
//	Integer Division
%	Remainder

Python Built-In Numeric Operations

# Statements, Operators, Expressions

- You can produce new data (numeric or text) values in your program using *expressions*

- This is an expression that uses the *addition operator*
- This is another expression that uses the *multiplication operator*

```
>>> x = 2 + 3
>>> print(x)
5
>>> print(5 * 7)
35
>>> print("5" + "7")
57
```



# Statements, Operators, Expressions

- You can produce new data (numeric or text) values in your program using *expressions*

*Another  
example...*

```
>>> x = 6
>>> y = 2
>>> print(x - y)
4
>>> print(x/y)
3.0
>>> print(x//y)
3
```

*Yet another  
example...*

```
>>> print(x*y)
12
>>> print(x**y)
36
>>> print(x%y)
0
>>> print(abs(-x))
6
```

# Statements, Operators, Expressions

- در **پایتون** به هر بخش قابل اجرا یک `statement` می‌گویند. که این `statement`ها زیربنای برنامه‌های نوشته شده در این زبان است.
- عملگرها ساختارهای مهم دیگری هستند که برای انجام عملیات ریاضی و منطقی در پایتون از آنها استفاده می‌شود: که شامل عملگرهای محاسباتی مثل جمع، ضرب، تقسیم، تفریق، توان، مد یا باقیمانده تقسیم و عملگرهای منطقی شامل `and`، `Or`، `not` و همچنین عملگرهای مقایسه‌ای شامل تساوی (`==`)، بزرگتر و کوچکتر (`<`) می‌باشند.
- به هر ترکیبی از متغیرها و عملگرها و `Value`ها یک `Expression` می‌گویند.
- یکی از موارد مهمی که در هنگام تعریف `expression`ها و استفاده از عملگرها باید مد نظر قرار گیرد اولویت عملگرهاست. در برنامه نویسی برای اجرای عملگرها ترتیب خاصی در نظر گرفته اند که در ترتیب اولویت به ترتیب با پرانتز، توان، ضرب و تقسیم و جمع و تفریق می‌باشد.

# Datatype Conversion

# تبدیل انواع داده به هم

```
>>> x = 10
>>> float(x)
10.0
>>> str(x)
'10'
>>>
```

integer → float  
integer → string

```
>>> y = "20"
>>> float(y)
20.0
>>> int(y)
20
>>>
```

string → float  
string → integer

```
>>> z = 30.0
>>> int(z)
30
>>> str(z)
'30.0'
>>>
```

float → integer  
float → string

# Exercise

به ترتیب عبارات زیر را در ترمینال وارد می کنیم، به ترتیب چه مقادیری در ترمینال چاپ می شود؟

```
2 ** 3
33 // 17
11 % 3
```

چه مقداری در خروجی چاپ می شود؟

```
print( 'a'*1 + 'b'*2 + 'c'*3 )
```

به ترتیب عبارات زیر را در ترمینال وارد می کنیم. آخرین مقداری که در ترمینال چاپ می شود کدام است؟

```
age = input("How old are you? ")
type( age )
```

# Exercise

به ترتیب عبارات زیر را در ترمینال وارد می کنیم. آخرین مقداری که در ترمینال چاپ می شود کدام است؟

```
x = 5
x = x + 1
x = 5 * 2
x * 3
print( x )
```

به ترتیب عبارات زیر را در ترمینال وارد می کنیم. آخرین مقداری که در ترمینال چاپ می شود کدام است؟

```
hazine_zamin_metri = 10
masahate_zamin = 20
hazineye_zamin = hazine_zamin_metri * masahate_zamin
print( hazineye_zamin )
```

```
hazine_zamin_metri = 10
masahate_zamin = 20
hazineye_zamin = hazine_zamin_metri * masahate_zamin
masahate_zamin = 30
print( hazineye_zamin )
```

# Boolean

- Boolean ها یکی از انواع داده ها هستند که دو مقدار `true` و `false` را می پذیرند.
- این متغیرها نقش اساسی در زبان های برنامه نویسی به خصوص **پایتون** دارند.
- به کمک متغیرهای `Boolean` می توان عبارتهای شرطی و کنترلی را تعریف کرده و امکان تصمیم گیری را برای برنامه ها فراهم نمود.
- `boolean` ها معمولا در پاسخ به عملگرهای منطقی مثل `And` و `or` و `not` همچنین عملگرهای مقایسه ای مثل بزرگتر (`<`)، کوچکتر (`>`)، مساوی (`==`) و یا نامساوی (`!=`) استفاده می شوند.

# Exercise

کدام گزینه بدون خطا (Error) در پایتون اجرا می شود؟

```
a = 3  
print( a = 3 )
```

```
a = 7  
print( a == 3 )
```

کدام گزینه سن شخص را از ورودی می خواند و در صورتی که بزرگتر از ۲۰ سال بود در خروجی True چاپ می کند و در صورتی که کوچکتر یا مساوی ۲۰ بود در خروجی False چاپ می کند؟

```
age = int( input() )  
print( age > 20 )
```

```
age = input( int() )  
print( age >= 20 )
```

```
age = str( input() )  
print( age < 20 )
```

# Exercise

**What are the differences between Python 2 and Python 3?**

- **Deadline: 02. 08. 21**
- **Email: [s.alebooye@ut.ac.ir](mailto:s.alebooye@ut.ac.ir)**