

# Python Data Essentials: Programming Fundamentals

## Using Conditional Statements



**Mateo Prigl**  
Software Developer

# Prerequisites

**Set up Python on  
your OS**

**Run Python scripts  
from files and  
notebooks**

**Basic data types and  
structures**



# Overview



## Who should be taking this course?

- You want to learn the basics of Python programming
- You want to get into data analysis with Python (Pandas, NumPy...)

All code and slides are available to download from the "Exercise Files" tab on the course's page

Ask questions in the "Discussion" tab; don't ask questions in the feedback form



# Welcome to the course!



# Python Data Types

**Integer**

**Float**

**String**

**List**

**Dictionary**



# Boolean Data Type

True

False



**What if we want to  
conditionally control the  
flow of the program?**



# Controlling the Program's Flow with an if Statement

The code block that belongs to the if statement is indented

script.py

```
bool_var = True

if bool_var:
    print("It is true!")
    print("It is really true!")

print("Moving on!")
```



# Controlling the Program's Flow with an if Statement

The code block that belongs to the if statement is indented

script.py

```
bool_var = False

if bool_var:
    print("It is true!")
    print("It is really true!")

print("Moving on!")
```



# Explicit Type Conversion

script.py

```
string_num = '5.5'  
print(string_num)  
  
float_num = float(string_num)  
print(float_num)  
  
int_num = int(float_num)  
print(int_num)  
  
string_num = str(int_num)  
print(string_num)
```

```
> python3 script.py  
5.5  
5.5  
5  
5
```



# Boolean Type Conversion

script.py

```
print(bool(None))      # None keyword (no value, null)  
print(bool(0))        # Numbers equal to zero  
print(bool([]))       # Empty list  
print(bool({}))        # Empty dictionary  
print(bool(()))       # Empty tuple  
print(bool(3))         # Numbers which are not zero  
print(bool([1,3]))     # Non-empty list
```

```
> python3 script.py  
False  
False  
False  
False  
False  
True  
True
```



# Comparison Operators



```
n1 = 3  
n2 = 5  
n3 = 3
```

```
n1 < n2      # 3 < 5  True  
n1 > n2      # 3 > 5  False
```

```
n1 <= n3     # 3 <= 3  True  
n3 >= n2     # 3 >= 5  False
```

```
n1 == n3     # 3 == 3  True  
n1 != n2     # 3 != 5  True
```

◀ Less than  
◀ Greater than

◀ Less than or equal to  
◀ Greater than or equal to

◀ Equal  
◀ Not equal



# Using the else Statement

The else statement has to be defined after the code block of the if statement

script.py

```
number = 4
True
if number == 4:
    print("Number is 4")
else:
    print("Number is not 4")

print("Done with checking the number")
```

```
> python3 script.py
Number is 4
Done with checking the
number
```



# Using the else Statement

The else statement has to be defined after the code block of the if statement

script.py

```
number = 2
    False
if number == 4:
    print("Number is 4")
else:
    print("Number is not 4")

print("Done with checking the number")
```

```
> python3 script.py
Number is not 4
Done with checking the
number
```



# Using the elif Statement

The elif statement has to be defined after the code block of the if statement

script.py

```
number = 3

if number == 4: False
    print("Number is 4")
elif number == 3: True
    print("Number is 3")
else:
    print("Number is not 3 or 4")

print("Done with checking the number")
```

```
> python3 script.py
Number is 3
Done with checking the
number
```



# Logical Operators



```
num = 30
```

```
True          True  
num > 20 and num < 50 True  
# 30 > 20 and 30 < 50
```

```
False        True  
num == 20 and num != 50 False  
# 30 == 20 and 30 != 50
```

```
True        True  
num > 20 or num < 50 True  
# 30 > 20 and 30 < 50
```

```
False        True  
num == 20 or num != 50 True  
# 30 == 20 and 30 != 50
```

◀ The **and** operator checks if all of the conditions are **true**

◀ If any one of the conditions is **false**, the whole expression is **false**

◀ The **or** operator checks if any one of the conditions is **true**

◀ If just one of the conditions is **true**, the whole expression is **true**



```
num = 30
```

```
not num == 30      False  
# not 30 == 30
```

```
not num != 30      True  
# not 30 != 30
```

```
not False      True
```

```
not True      False
```

```
not num      False  
# not True
```

◀ The **not** operator applies to only one condition and it reverses its boolean value

◀ If the condition results in **false**, the **not** operator will negate it to **true**

◀ If the value is **true**, the **not** operator will negate it to **false**



**Up Next:**

# **Implementing Loops**

---

