Introduction to Programming with Python

DUCSS - Dublin University Computer Science Society

Dylan Lewis, Owen Shepherd, David Phillips, Oisin Carroll, Ezi Ozoani, Brian Whelan, Arne Philipeit

Volunteers are here to help

Ask Questions

Experiment with the Code

Plan for Today

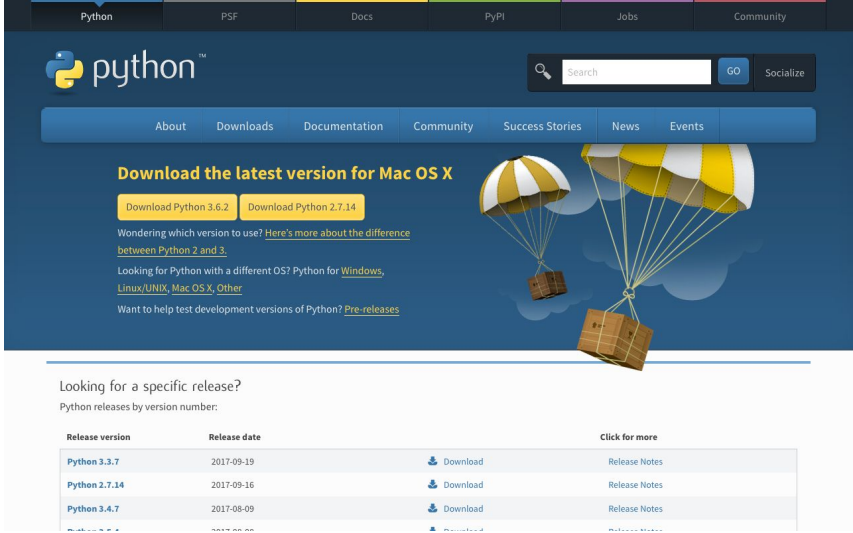
- Installing Python
- Python Tool

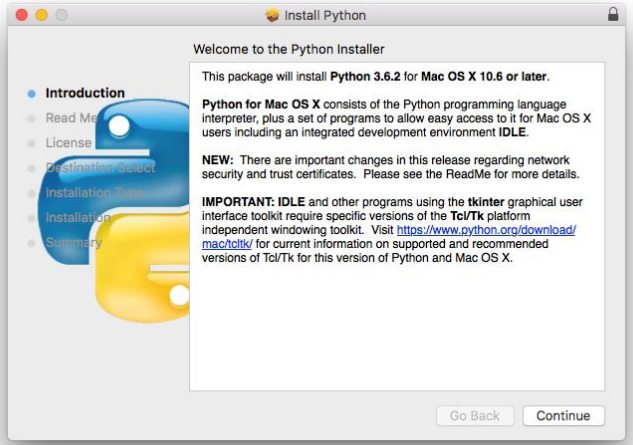
- “Hello World!”
- Variables
- Branching
- Loops
- Functions

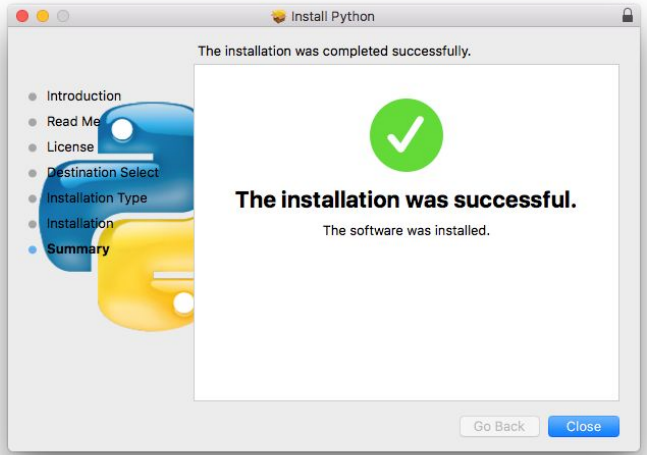
- Exercises

Installing Python

- Go to python.org/downloads
- Download Version 3.x.x
- Follow the Installation Instructions







Checking your Installation

- Open Terminal (Linux/macOS) or the Command Line (Windows)
- Run “python”

  Expected output:

```
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

- If version is not 3.x.x
  - Enter “exit()”
  - Run “python3”

Python Tool

- Entering manual commands:

```
python
>>> #your python commands here
>>> exit()
```

- Running Python scripts (.py files):

```
python path/to/your/python/script.py
```

Sample Code

github.com/arneph/python-workshops

“Hello World!”

```
python
>>> 'Hello World!'
'Hello World!'
```

“Hello World!”

```
>>> 'Hel' + 'lo!'
'Hello!'
```

“Hello World!”

```
>>> A = 'Hello'
>>> B = ' '
>>> C = 'World!'
>>> A + B + C
'Hello World!'
```

“Hello World!”

- Open Editor/Notepad/your favorite text editor
- Create a new file
- Enter the following:

```
X = 'Hello World!'

print(X)
```

- Save the file as “hello.py” on your desktop
- In Terminal/the Command Line:

```
python /Desktop/hello.py
Hello World!
```

hello_world.py

Variables

- Storing values:

```
firstName = 'Gina'
lastName = 'Hopper'
born = 1903
```

- Changing values:

```
firstName = 'Grace'
born = 1906
```

variables.py

Variables

- Values can be used in computation and for function calls

```
age = 2017 - born
name = firstName + ' ' + lastName
print(name + ' would now be')
print(age)
print('years old today')
```

variables.py

Variables

```
firstName = 'Gina'
lastName = 'Hopper'
born = 1903

firstName = 'Grace'
born = 1906

age = 2017 - born
name = firstName + ' ' + lastName
print(name + ' would now be')
print(age)
print('years old today')
```

variables.py

Variables

```
Grace Hopper would be
111
years old today
```

variables.py

Branching

- if-statements:

```
A = 9
B = 7
if A > B:
    print('A is larger')
    print('A wins!')

if B > A:
    print('B is larger')
    print('A loses! :-(')

print('Game over')
```

if_1.py

Branching

```
A is larger
A wins!
Game over
```

if_1.py

Branching

- if-statements (continued):

```
A = 42
B = 42
if A > B:
    print('A wins!')
elif A == B:
    print('Draw!')
else:
    print('B wins!')
```

if_2.py

Branching

```
Draw!
```

if_2.py

Branching

- Combining Conditions:

```
username = 'marc z'
password = 'dadada'
if username == 'marc z' and password == 'dadada':
    print('Hello Marc')
    print('Get a better password!')
else:
    print('You are not Marc!')
```

if_3.py

Branching

```
Hello Marc
Get a better password!
```

if_3.py

Branching

- Combining Conditions (continued):

```
age = 42
if age < 21 or age >= 30:
    print('Not a primary target')
    if age >= 18 and age < 21:
        print('Keep watching')
    else:
        print('Remove from watchlist')
```

if_4.py

Branching

```
Not a primary target
Remove from watchlist
```

Loops

- while-loops:

```
i = 1
while i <= 30:
    print(i ** 2)    #Print i^2
    i += 1           #Add 1 to i
```

loops_1.py

Loops

```
1
4
9
16
25
36
49
64
81
100
121
...
```

loops_1.py

Loops

- for-loops:

```
factorial = 1
for x in range(1, 12):
    factorial *= x
    print(factorial)
```

loops_2.py

Loops

```
1
2
6
24
120
720
5040
40320
362880
3628800
39916800
```

loops_2.py

Loops

- continue & break:

```
for x in range(2, 40):
    prime = True
    for y in range(2, x):
        if x % y == 0:
            prime = False
            break                   #Break out of inner for-loop
    if prime == False:
        continue                    #Continue with next item in
    print(x)                        #outer for-loop
```

loops_3.py

Loops

```
2
3
5
7
11
13
17
19
23
29
31
37
```

loops_3.py

Functions

- Using Functions:

```
print('Hello!')

A = pow(19, 2)
B = input()      #Get keyboard input from the user

for x in range(42, 888):
    #...
```

Functions

- Defining Functions:

```
def isOdd(x):
    return x % 2 == 1

print(isOdd(27))
print(isOdd(2048))
print(isOdd(22))
```

functions_1.py

Functions

```
True
False
False
```

functions_1.py

Functions

- Defining Functions (continued):

```
def fizzbuzz(x):
    a = (x % 3 == 0)
    b = (x % 5 == 0)
    if a and b:
        print('Fizz Buzz')
    elif a:
        print('Fizz')
    elif b:
        print('Buzz')
    else:
        print(x)
```

functions_2.py

Functions

- Defining Functions (continued):

```
#...

for x in range(1, 50):
    fizzbuzz(x)
```

functions_2.py

Functions

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
...
```

functions_2.py

Exercises

- Prompt the user for a name and password and check if they are correct
  Hint: use the `input()` function

- Rewrite the prime numbers example with a function
  Hint: define a function `isPrime(x)`

- Print all Fibonacci numbers up to 10946