

# Introduction to Python

## #

- to make comments in your script

## \*\*

- Exponentiation:  $4^{**}2 = 16$

## %

- Modulo:

## [start : end]

- inclusive : exclusive

## list() / [:]

- to make a copy of the list without changing the real one
- E.g:

```
# Create list areas
areas = ['hallway', 'kitchen', 'living room', '
         bedroom', 'bathroom', 'hallway', 'kitchen']

# Create areas_copy
areas_copy = list(areas)

# Change areas_copy
areas_copy[0] = 'entrance'
areas_copy[4] = 'kitchen'
areas_copy[6] = 'living room'
```

```
areas_copy[0] = 5.0

print(areas)
print(areas_copy)
[11.25, 18.0, 20.0, 10.75, 9.5]
[5.0, 18.0, 20.0, 10.75, 9.5]
```

---

# Functions in Python:

## 1) round()

- round(number)
  - round(1.68)
    - result: 2
- round(number, ndigits)
  - round(1.68, 1)
    - result: 1.7

## 2) help(x) / ?x

- ask for information about a function
- E.g. help(max) / ?max

## 3) .index()

- to get the index of the first element of a list that matches its input

## 4) .count()

- to get the number of times an element appears in a list
  - `append()`, that adds an element to the list it is called on,
  - `remove()`, that removes the first element of a list that matches the input, and
  - `reverse()`, that reverses the order of the elements in the list it is called on.
- 

# Packages in Python

## Used for Data Science:

1. Numpy
2. Matplotlib
3. Scikit-learn

datacamp

← Course Outline →

Daily XP 350

Packages 50 XP

### Import package

```
import numpy
array([1, 2, 3])
```

```
import numpy as np
np.array([1, 2, 3])
```

```
array([1, 2, 3])
```

```
NameError: name 'array' is not defined
```

```
from numpy import array
array([1, 2, 3])
```

```
array([1, 2, 3])
```

numpy.array([1, 2, 3])

array([1, 2, 3])

This from import version to use specific parts of a package can be

Show Transcript

This course is also available on the mobile app

Get It!

## import numpy as np is the best to use!!!

General imports, like `import math`, make **all** functionality from the `math` package available to you. However, if you decide to only use a specific part of a package, you can always make your import more selective:

```
from math import pi
```

## Numpy

- example of its use is on arrays

[←](#)
[Course Outline](#)
[→](#)

Daily XP 650

## Numpy

50 XP

### Numpy

```

import numpy as np
np_height = np.array(height)
np_height

array([ 1.73,  1.68,  1.71,  1.89,  1.79])

np_weight = np.array(weight)
np_weight

array([ 65.4,  59.2,  63.6,  88.4,  68.7])

bmi = np_weight / np_height ** 2
bmi

array([ 21.852,  20.975,  21.75 ,  24.747,  21.441])

```

The first person's BMI was calculated by dividing the first element in np\_weight by the square of the first

INTRODUCTION TO PYTHON

Show Transcript

Got It!

[←](#)
[Course Outline](#)
[→](#)

Daily XP 1250

## 2D Numpy Arrays

50 XP

### Subsetting

	0	1	2	3	4	
array([[	1.73,	1.68,	1.71,	1.89,	1.79],	0
[	65.4,	59.2,	63.6,	88.4,	68.7]])	1

```

np_2d[:,1:3]

array([[ 1.68,  1.71],
       [ 59.2 ,  63.6 ]])

np_2d[1,:]

array([ 65.4,  59.2,  63.6,  88.4,  68.7])

```

Finally, 2D numpy arrays enable you to do element-wise calculations, the same way you did it with 1D numpy arrays.

INTRODUCTION TO PYTHON

Show Transcript

Got It!

**np\_2d[row, column]**