



Python in 10 minutes

Part 2:

Dr. Mark Williamson

Purpose:

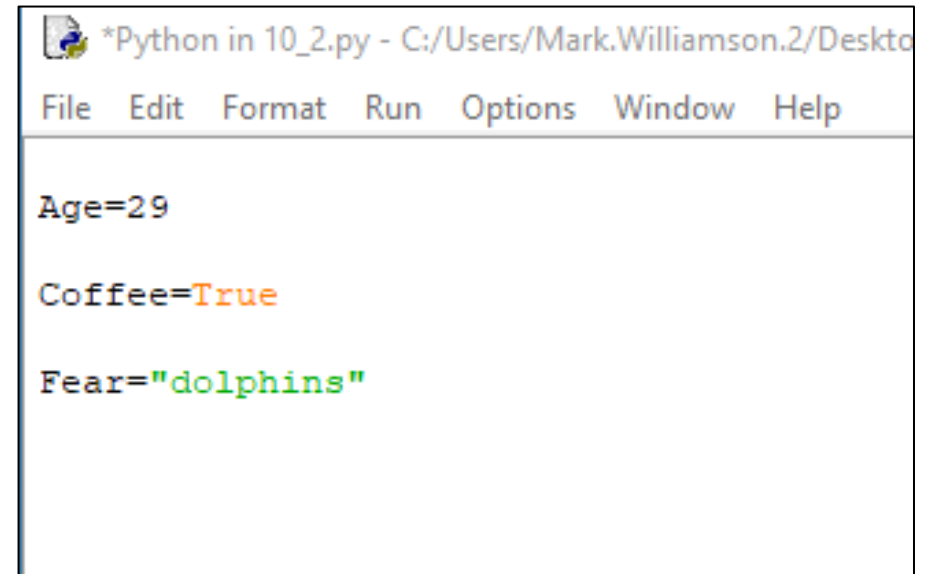
- Quick, bite-size guides to basic usage and tasks in Python
- I'm no expert, I've just used it for various tasks, and it has made my life easier and allowed me to do things I couldn't manually
- I'd like to share that working knowledge with you

Lesson 2: Obligatory Fundamentals

- Obligatory Fundamentals
 - Variables
 - Lists
 - Math
 - Functions and Methods
 - Dictionaries

Lesson 2: Variables

- Variables are named locations used to store all sorts of objects
 - Can store number, strings, lists, dictionaries, Booleans, etc.
- Create them by *variable_name = value*
- The name of the variable can be whatever you want
 - However, can't start name with number or symbol
 - Also can't be name of something already in Python, like a function
- **Open a Python Script:** create 3 variables
 - Age: your age in years
 - Coffee: whether you've had coffee today or not (True/False)
 - Fear: string of the animal you are most afraid of (enclose in quotes)



```
*Python in 10_2.py - C:/Users/Mark.Williamson.2/Desktop
File Edit Format Run Options Window Help

Age=29

Coffee=True

Fear="dolphins"
```

Lesson 2: Lists

- Lists are groups of objects stored in a variable
- Create them by `List = [object1, object2, ... objectN]`
- Can store the same objects or different ones
- Can also store other variables
- Can access specific parts of the list by index
 - `List[0]` is the first object, `List[1]` is the second, etc.
 - `List[-1]` is the last object
- **In Python Script:** create 3 lists
 - FavNum: your 5 favorite numbers
 - BreakList: strings of items you had for breakfast
 - Attributes: your Age, Coffee, and Fear variables
 - Access the Coffee variable in the shell

```
Python in 10_2.py - C:/Users/Mark.Williamson.2/Desktop/Williamson Data/Python/Python in ...
File Edit Format Run Options Window Help

Age=29
Coffee=True
Fear="dolphins"
FavNum=[1, 2, 4, 7, 42]
BreakList=["eggs", "ham", "hashbrowns", "toast", "jam", "coffee"]
Mark=[29, True, "dolphins"]
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Mark.Williamson.2/Desktop/Williamson Data/Python/Python in 1
0_2.py
>>> Mark[1]
True
>>>
```

Lesson 2: Math

- Python can do lots of basic math operations
- Can do operations on variables that are numbers
- Also can get modules added on that do much more sophisticated things
- **In Python Script:**
 - Add 5, 22, and 7 in the Shell
 - Multiple 7 and 550, then subtract 37 in the Shell
 - Create variables (Num1 and Num2), then put any number in each and divide them in the Shell

```
Python in 10_2.py - C:\Users\Mark.Williamson.2\Desktop\Williamson Data\Python\Python in ...
File Edit Format Run Options Window Help

Age=29
Coffee=True
Fear="dolphins"
FavNum=[1, 2, 4, 7, 42]
BreakList=["eggs", "ham", "hashbrowns", "toast", "jam", "coffee"]
Mark=[29, True, "dolphins"]
Num1 = 5
Num2 = 10
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help


Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Mark.Williamson.2\Desktop\Williamson Data\Python\Python in 1
0_2.py
>>> 5+22+7
34
>>> 7*550-37
3813
>>> Num1/Num2
0.5
>>>
```

Math Operators		
From Highest to Lowest precedence:		
Operators	Operation	Example
**	Exponent	2 ** 3 = 8
%	Modulus/Remaider	22 % 8 = 6
//	Integer division	22 // 8 = 2
/	Division	22 / 8 = 2.75
*	Multiplication	3 * 3 = 9
-	Subtraction	5 - 2 = 3
+	Addition	2 + 2 = 4

<https://www.pythoncheatsheet.org/>

Lesson 2: Functions and Methods

- Python has lots of built-in functions and methods
- You can also create your own
- Some functions/methods only work for certain data (numbers, lists, etc.)
- Typical format
 - Function: *functionName(arguments)*
 - Method: *variable.methodName()*
- Most important*
 - Functions: `print()`, `abs()`, `round()`, `min()`, `max()`, `sorted()`, `sum()`, `len()`, `type()`,
 - Methods: `.lower()`, `.upper()`, `.strip()`, `.replace()`, `.split()`, `.join()`, `.append()`
- **In Python Script:** using the Shell
 - Print your Breakfast list with the `print()` function
 - Use `.upper()` method to make your Fear variable uppercase
 - Sum your favorite number list
 - Add your favorite dessert to your name list by using `.append()`, then print it

A screenshot of a Python 3.8.1 Shell window. The window title is "Python 3.8.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following code and output:

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Mark.Williamson.2\Desktop\Williamson Data\Python\Python in 1
0_2.py
>>> print(BreakList)
['eggs', 'ham', 'hashbrowns', 'toast', 'jam', 'coffee']
>>> Fear.upper()
'DOLPHINS'
>>> sum(FavNum)
56
>>> BreakList.append("cheesecake")
>>> print(BreakList)
['eggs', 'ham', 'hashbrowns', 'toast', 'jam', 'coffee', 'cheesecake']
>>>
```

Lesson 2: Dictionaries

- A dictionary is a collection which is unordered, changeable and indexed
- They written with curly brackets, and they have keys and values
- Create them by: `dictName = { key1: value1, key2: value2,...}`
- The keys must be unique, but the values can be the same
 - Keys can be strings or numbers
 - Values can be strings, numbers, lists, or even dictionaries
- You can access items in a dictionary by referring to its key name in the format: `dictName[key1]`
- You can add items to a dictionary using this format: `dictName[key3] = value3`
- **In Python Script:**
 - Create a dictionary call myDictionary: Add the keys of “Name”, “Age”, “Job”, FavColor”, and “CoffeeCups” with your specific values
 - Access your favorite color (FavColor)
 - Add the item “Dinosaurs” with the value of True or False, depending on if you like dinosaurs or not, then print the dictionary

```
Python in 10_2.py - C:\Users\Mark.Williamson.2\Desktop\Williamson Data\Python\Python in ...
File Edit Format Run Options Window Help

Age=29
Coffee=True
Fear="dolphins"
FavNum=[1, 2, 4, 7, 42]
BreakList=["eggs", "ham", "hashbrowns", "toast", "jam", "coffee"]
Mark=[29, True, "dolphins"]

Num1 = 5
Num2 = 10

myDictionary ={
    "Name": "Mark",
    "Age": 29,
    "Job": "Statistician",
    "FavColor": "green",
    "CoffeeCups": 2
}
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Mark.Williamson.2\Desktop\Williamson Data\Python\Python in 10_2.py
>>> myDictionary["FavColor"]
'green'
>>> myDictionary["Dinosaurs"]=True
>>> print(myDictionary)
{'Name': 'Mark', 'Age': 29, 'Job': 'Statistician', 'FavColor': 'green', 'CoffeeCups': 2, 'Dinosaurs': True}
>>> |
```


Lesson 2: Resources

- I've just scratched the surface on all the sorts of fundamentals in Python
- Check out the cheat sheet list below for a lot more information
 - <https://sinxloud.com/python-cheat-sheet-beginner-advanced/>