

Python Fundamentals

Part 1: Objects and Functions

Colby Witherup Wood
NU IT Research Computing Services



This Photo by Unknown Author is licensed under CC BY-ND



Python is a language

How do we talk to our computer in Python?

- **Interactive programming:** through a shell (interpreter, console) one line at a time
- **Batch programming:** running a whole script (a plain text file containing one to many lines of code)
- With the help of a GUI (graphical user interface). GUIs for coding are called **IDEs (Integrated Development Environments).**

How do we talk to our computer in Python?

- **Interactive programming:** Part 1
- **Batch programming:** Part 2
- **IDE (Jupyter Notebook):** Part 3

Let's get the tech worked out

Options to open a Python console:

1. Open Anaconda Navigator and open Jupyter Lab. Select Python 3 Console.
2. Mac: Open the Terminal and type `ipython` or `python`
3. PC: Open Anaconda Prompt and type `ipython` or `python`

Python is:

50% Syntax

- Which words to use
- Punctuation
- Order
- Indentation
- Shortcuts

50% Logic

- Which tools to use in which order
- Creative
- Specific to one problem

Try not to worry if you don't understand everything today

Objects and functions

The two main concepts of Python.

Object: a particular piece of data (like a noun)

Function: something you can do to/with an object (like a verb)

Objects and functions

You can use certain functions with certain classes of objects.

Some functions are shared between objects.

Objects and functions



Molly

- Feed Molly
- Pet Molly
- Draw Molly



My bicycle

- Pedal my bicycle
- Pet my bicycle (limited use)
- Draw my bicycle

Objects and functions



Molly (class Dog)

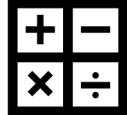
- Feed a dog
- Pet a dog
- Draw a dog



My bicycle (class Bicycles)

- Pedal a bicycle
- Pet a bicycle (limited use)
- Draw a bicycle

Data types



3 (class Integer)

- Divide ($3 / 3 = 1$)
- Convert 3 to 3.0
- Add ($3 + 3 = 6$)



"3" (class String)

- Replace ("3" becomes "1")
- Convert "3" to 3.0
- Add ("3" + "3" = "33")

Today's objects

- Integer (whole number) 3
- Float (number with digits following a decimal point) 3.0
- String (text or characters) "3" or "three"
- Boolean (True or False)
- List (group of objects in a strict order) [3, 3, 4, 5, 3]
- *Assigning objects to variables*

Today's functions

- Basic operators: + - * /
- Functions to convert between data types
- String functions
- List functions
- *Writing your own function*
- *Using specialty functions made by others*

Let's code!



Zen of Python

Explicit is
better than
implicit

Today's objects

- ✓ Integer (whole number) 3
- ✓ Float (number with digits following a decimal point) 3.0
- ✓ String (text or characters) "3" or "three"
- ✓ Boolean (True or False)
 - List (group of objects in a strict order) [3, 3, 4, 5, 3]
 - *Assigning objects to variables*

Today's functions

- ✓ Basic operators: + - * /
- ✓ Functions to convert between data types
 - String functions
 - List functions

- *Writing your own function*
- *Using specialty functions made by others*

Variable names

- No spaces
- Case matters
- Must start with a letter
- Should be meaningful

Style conventions:

- camelCase
- separate_with_underscores

Python functions

Parentheses immediately follow the function name.

1. Can stand on their own, and take an object as an **argument** (do something **with** the object)

```
print(my_name)           len(my_name)
```

2. Can follow an object, with or without arguments (do something **to** the object)

```
my_name.replace("C", "c")  my_name.lower()
```

In Python indexing, we start counting with 0



Indexing strings

```
"H   e   l   l   o       W   o   r   l   d   !"  
 0   1   2   3   4   5   6   7   8   9   10  11
```

"Hello World!"[6:12] is "World!"

To take a substring, you start with the position of the first character you want included, and end with the position 1 past the last character you want.

Exercise – Hidden message

Store a longer string as a variable

Write code to reveal a hidden, shorter message

You can use indexing and functions like
`replace()`, `+`, and `capitalize()`

Bonus points if you can make the two strings relate
in a clever way

Today's objects

- ✓ Integer (whole number) 3
- ✓ Float (number with digits following a decimal point) 3.0
- ✓ String (text or characters) "3" or "three"
- ✓ Boolean (True or False)
 - List (group of objects in a strict order) [3, 3, 4, 5, 3]
- ✓ *Assigning objects to variables*

Today's functions

- ✓ Basic operators: + - * /
- ✓ Functions to convert between data types
- ✓ String functions
- List functions
- *Writing your own function*
- *Using specialty functions made by others*

Try both. What is the difference?

```
frodo = ["hobbit"]
```

```
sam = ["hobbit"]
```

```
frodo
```

```
sam
```

```
frodo.append("ring")
```

```
frodo
```

```
sam
```

```
harry = ["wizard"]
```

```
ron = harry
```

```
harry
```

```
ron
```

```
harry.append("scar")
```

```
harry
```

```
ron
```



Try this.

```
frodo = ["hobbit"]
sam = frodo.copy
frodo
sam
frodo.append("ring")
frodo
sam
```

Exercise – Conversion function

Choose a unit to convert from this list:

inches to feet, Fahrenheit to Celsius, centimeters to inches,
seconds to days, teaspoons to cups

Using pen and paper, write the code for a new function that:

- takes a number as the argument
- converts the number from the first unit to the second
- returns a number

Define the function in the console and test it on several numbers

Python modules (aka packages)

Modules are groups of related functions and object classes

They are not automatically loaded when you enter a Python interpreter

You must **import** the module every time you start Python

If you do not have the module on your computer, you must first **install** it (only once)

Python modules (aka packages)

- Anaconda comes with many of the most common modules pre-installed
- Many others can be installed from the command line
- Some must be downloaded from the creator
- There are modules for every field and use