Week 1: Intro to Python

DSUA111: Data Science for Everyone, NYU, Fall 2020

TA Jeff, jpj251@nyu.edu

https://github.com/jpowerj/dsua111-sections
(https://github.com/jpowerj/dsua111-sections)

We'll be coding in Python - Why?

- Python rocks
- Very easy to write and to read
- Huge library of outside code you can use
- Now the default language for machine learning
- But also applied to many other usecases, such as the web (for example Instagram's backend)

What is Jupyter?

- We'll be writing python code - but there are many different tools to write that code in
- For example, there are many different ways to write word documents, like Microsoft Word, Google Docs, text edit, etc.
- The words are the same, but different platforms have different tools
- The same is true for coding, there are many different platforms or IDEs (Integrated Development Enviroment) such as Pycharm, sublimetext, Spyder, or even just text edit
- We'll be using Jupyter Notebooks, which is probably the most popular IDE for datascience

So what is a programming language? What is Python?

- From wikipedia:
    - "A programming language is a formal language, which comprises a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms."

Programming Languages

- In theory, they tell the computer what to do.
- But in practice, computers dumb - they only understand a small number of VERY specific instructions
- So learning to program is basically two things:
    1. Learning those instructions
    2. Figuring out how to put them together (into a program) that does what you want it to do
- Let's start

First, let's create a variable

- We'll create a variable (AKA an object), called 'a', and assign it the value 5
- We'll type it in a cell (the square thing where we write text), then press SHIFT + ENTER (or press run above)

In [1]:
```
a = 5
```

In [2]:
```
a == 6
```

Out[2]:
```
False
```

In [3]:
```
a == 5
```

Out[3]:
```
True
```

In [4]:
```
print(a)
```

```
5
```

What is a?

- we assign (or map) 'a' to the number 5 with just one '='
- Ok, so we created 'a', what does this mean?
- Well, we created a mapping to 'a'. When I tell the computer, "hey, use 'a' to..." the computer will look at 'a', and see what it maps to
- So let's ask the computer what it maps to, we can usually do this with a print statement, which is just..

In [5]:
```
print(a)
```

```
5
```

In [6]:
```
a
```

Out[6]:
```
5
```

Jupyter's default output behavior

- In Jupyter I can also just run a line of Python code by itself in a cell to output the result

```
In [7]: c = 47

In [8]: d = 5

In [9]: c + d
Out[9]: 52

In [10]: 2 + 5
Out[10]: 7

In [11]: answer = c + d

In [12]: answer
Out[12]: 52
```

So how is this useful?

- First things first, computers can do arithmetic (it's the basis for all the other fun things it can do!)
- Addition and subtraction:

In [13]:
```
12 + 25
```

Out[13]:
```
37
```

In [14]:
```
120030498 + 12375
```

Out[14]:
```
120042873
```

In [15]:
```
123 - 5
```

Out[15]:
```
118
```

Multiplication and Division

In [16]:
```
123/65
```

Out[16]:
```
1.8923076923076922
```

In [17]:
```
1/123123321321321874981239847298
```

Out[17]:
```
8.121938145172706e-30
```

In [18]:
```
3/4
```

Out[18]:
```
0.75
```

In [19]:
```
12*4
```

Out[19]:
```
48
```

In [20]:
```
c/d
```

Out[20]:
```
9.4
```

Exponents!

- For example we can compute 3^4 as:

In [21]:
```
3**4
```

Out[21]:
```
81
```

In [22]:
```
new_answer = c**2
```

In [23]:
```
new_answer
```

Out[23]:
```
2209
```

And remember, we can do roots as exponents by making them fractions:

- So $\sqrt{16}$

  $= 16^{\frac{1}{2}}$

  $=$

In [24]:
```
16**(1/2)
```

Out[24]: 4.0

In [25]:
```
16**(1/4)
```

Out[25]: 2.0

Math with variables

- First off, notice how we use parenthesis, just like in normal math (Python knows PEMDAS)
- But we can get fancier than just typing in specific numbers like 16, 1/4, etc.
- Python can do arithmetic on objects that map to numbers too
- so let's write a simple program

In [26]:
```
a = 15

b = 30

answer = a + b

print(answer)
```

```
45
```

In [27]:
```
answer
```

Out[27]:
```
45
```

In [28]:
```
a
```

Out[28]:
```
15
```

From Numbers to Variable and back to Numbers

- Notice, I reassigned 'a' to a new value, and created b
- I then created a new object, called 'answer', that maps to their new value
- and finally I printed that out
- Ok, so how could this be useful?
- Now lets do it for the Pythagorean (pythogorean?) theorem (https://en.wikipedia.org/wiki/Pythagorean_theorem)

In [29]:
```
a = 6

b = 8

c = (a**2 + b**2)**(1/2)

print(c)
```

```
10.0
```

In [30]:
```
c
```

Out[30]:
```
10.0
```

When you make the computer angry

- Finally, computers are very, very sensitive
- They can't handle it when they don't understand the code
- This is called an error!
- It is NO BIG DEAL (srsly, it's not... more on that later)

Let's make it angry, for fun

In [31]:
```
c_hello
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-31-f0a6956ee073> in <module>
----> 1 c_hello

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

What's happening?

- All of my variables still exist, nothing is broken
- Before python runs the code, it reads it
- If there are instructions in there that are not Python (for example typos)
- Python 'throws' an error - it doesn't actually run it.
- All the variables you created before are still in memory - the computer hasn't forgotten them
- Python even tells you what is wrong
- So make mistakes, experiment, etc. Python doesn't care while you learn

Please try: https://www.learnpython.org/ (https://www.learnpython.org)