

Python DeCal

Week 2

WEDNESDAY



Announcements

- 1st Hw is released today! Due next week before wednesday lecture
- Something about advanced topics poll, and final project ideas
- Office Hour Poll
 - <https://www.when2meet.com/?9754936-Zp46U>
- bCourses and Piazza

Command Line

- We need to understand this stuff before we can actually code in python because we need to know how to find our files in the first place
- The command lines are different in Mac and in Windows/Linux, but the general ideas are the same.

Directories (“Folders”) - Mac/Windows

Finding your current working directory: `pwd / cd`

Going into a certain directory: `cd /Your/Path`

List Files in a directory: `ls / dir`

Create a directory: `mkdir dir_name`

Move a file: `mv/move ori_path dest_path`

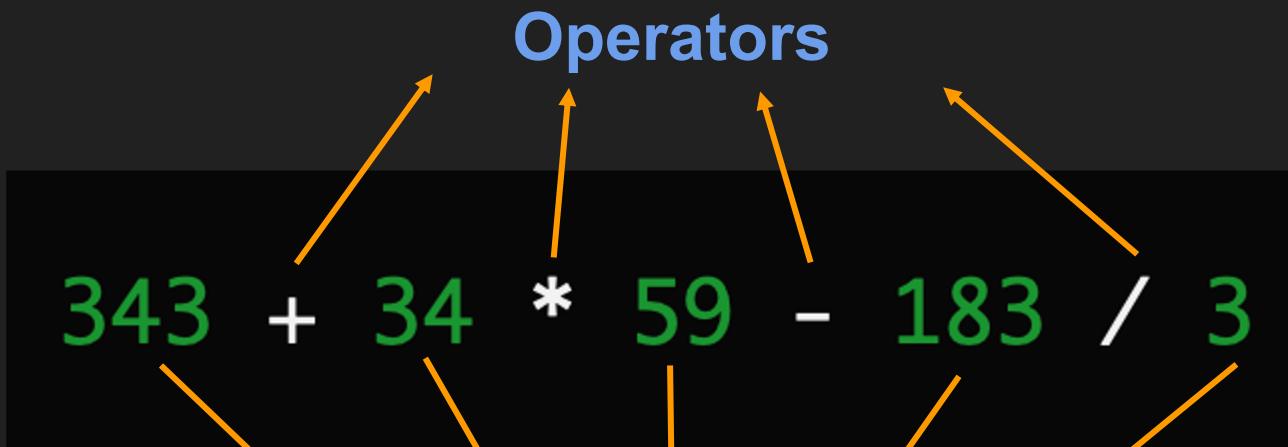
Rename a file: `mv/move ori_name new_name`

Remove a file: `rm/del file`

Better take a
screenshot now!

We are going to
do some
demonstrations.

Expressions



Values

Data Types-Numbers

- Integers (**int**) and Floats (**float**) - “NUMBERS”

- Integers: **-3 -2 -1 0 1 2 3**
- Floats: **3.3 13451.133434 98.7**

- How to check the type of an object?

type(your_object)

- Here is a question for you:

- What is the output of **type(1838849138304840103724482.)?**

| | | |
|-----------|----------------|------------------------------|
| + | Addition | $4 + 7 \longrightarrow 11$ |
| - | Subtraction | $12 - 5 \longrightarrow 7$ |
| * | Multiplication | $6 * 6 \longrightarrow 36$ |
| / | Division | $30 / 5 \longrightarrow 6$ |
| % | Modulus | $10 \% 4 \longrightarrow 2$ |
| // | Quotient | $18 // 5 \longrightarrow 3$ |
| ** | Exponent | $3 ** 5 \longrightarrow 243$ |

Data Types-Names

- **VARIABLES - “Giving Names”**

- $x = 1, y = 2 \longrightarrow x + y \longrightarrow 3$

- We want to name the variables nicely :D

- Cannot start the name with a number
 - Don't start with "O", "I", or "L"
 - Make your variable name IMMEDIATELY clear and recognisable
 - https://visualgit.readthedocs.io/en/latest/pages/naming_convention.html

Data Types-Characters

- Strings (**str**) - “WORDS”
 - ‘Hello World’
 - “Go Bears!”
 - Either type of quotation marks is fine so long as they are consistent
- You can add two strings

```
>>> "134" + "34"
```

What is the output of **7 * "1"**?

“13434”

- Indexing (Python index starts from 0, not 1!)

```
>>> "Hello World"[0]
```

“H”

```
>>> "Hello World"[5]
```

“ ”

Data Types-Lists

- List (**list**) `[4, 9, 7.5, 'astronomy', ["Berkeley", True]]`
 - Calling certain elements (INDEXING)
 - `[1,2,3,4,5,6][0]` → 1 `[1,2,3,4,5,6][-1]` → 6
 - `[1,2,3,4,5,6][2:5]` → `[3,4,5]` (from the 2nd element to the 5th, not included)
 - You can modify the elements in a list
 - `my_list = [1,2,3]` → `my_list[0] = 37`
`my_list` → `[37,2,3]`
 - Operation
 - `3 * [1,2]` → `[1,2,1,2,1,2]`
 - `[1,2] + [1,2,3]` → `[1,2,1,2,3]`

Data Types - Tuples & Dictionaries

- Tuple (**tuple**) **(1,2,3,4)**
 - Indexing and operations are the same
 - You CANNOT change the elements
 - `my_tuple = (1, 2, [3, 4])` --> `my_tuple[-1][0] = 100`
What is going to happen??????
- Dictionary (**dict**)
 - `d = {Stars: ['Sirius', 'Sun'], Planets: ['Venus', 'Saturn']}`
 - Accessing items:
`>>> d['Stars']`
`['Sirius', 'Sun']`

Data Types-Booleans

- Booleans (**bool**) - “True or False”
 - **True**
 - A value is true unless it is not
 - **False**
 - 0, [], (), {}, "", **None**,
- Here is a question for you:
 - What is the output of **bool(" ")**?
- Operations :
 - **3 > 2**
 - **0 <= -100**
 - **23 == 23**
 - **73 != 9**

Fill in the Attendance Form Please
<https://tinyurl.com/someone-didnt-sleep>

Also, please fill in the Office Hour W2M as well
<https://www.when2meet.com/?9754936-Zp46U>

FRIDAY



Recap

- Basic Command Line
- Data Types, Can you name them all?
- Functions

Before We Get Started

- Breakout room time!
- Challenge 1: What does `cd`, `mkdir`, `pwd`, `ls` all mean? How are they used?
- Challenge 2 : If I `pwd` and the terminal reads `Users/yourname/Desktop`

How would I get to a new directory

`Users/yourname/Desktop/PyDeCal`

If there is no pre existing folders on the desktop

Introduction to Functions

- Just like a mathematical function $f(x)$
 - Takes a value(s) → outputs some other value
- Another Example: I have some values and I want to convert it to a string and attach “.jpg” at the end.
(e.g. [23, 7] → “[23, 7].jpg”)
- Built-in functions
 - `print`
 - `abs`
 - `max`
 - `min`

```
def func(x):  
    y = x**100 + 23*x**2 + 56  
    return y
```

Advanced Functions

- How do we make some more complicated functions?
- What can we do with more complicated functions?

Advanced Functions

- How do we make some more complicated functions?
- What can we do with more complicated functions?

$$\frac{dP}{dr} = -\frac{GM(r)\rho(r)}{r^2}$$
$$\frac{dT}{dr} = -\frac{3L(r)\kappa\rho(r)}{16\pi acT^3 r^2}$$
$$\frac{dM}{dr} = 4\pi r^2 \rho(r)P = \frac{kT}{\bar{m}}\rho(r)$$

Advanced Functions

EW

- How do we make some more complicated functions?
- What can we do with more complicated functions?

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = - \frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + \rho g_x$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + \rho g_y$$

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = - \frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho g_z$$

Variable Scope

- Outside functions = Global variable
- Inside functions = Local variable

Global

```
G = 6.67e-11 #kg^-1 m^3 s^-2
```

```
pi = 3.14
```

```
def Luminosity(radius, temp):
```

```
    sigma_sb = 5.67e-8 #watt m^-2 K^-4
```

```
    L = 4*pi*radius**2 * sigma_sb * temp**4
```

```
return L
```

Local

DEMO

Advanced/Extra Material

- Lambda functions!
- The quick and dirty way of making a function

The diagram shows a code snippet for a lambda function:

```
quadratic = Lambda x: x**2|
```

Annotations with arrows point to specific parts of the code:

- Function Name: Points to the word "Lambda".
- Lambda Declaration: Points to the word "Lambda".
- Variable: Points to the variable "x".
- Return Statement: Points to the expression "x**2".