2/9/19 Class 2: Python Programming Fundamentals

I. **Datatypes and Class**
II. **Variables and Literals**
III. **Operators and Functions**
(https://docs.python.org/3.6/library/ for more methods)
IV. **Control Flow: if/else/elif**

I. Datatypes and Class

Datatypes: a set of values and operations that can be used on them → abstract concept (not specific to any language)
Class: program-code-template written by programmers to create an object of the datatype (defines the properties and operations for each datatype) → programming language-specific

Datatype	Python class	Functions defined for our use
Integers	`int`	arithmetic, comparisons
Floating point numbers	`float`	arithmetic, comparisons
Booleans	`bool`	negation, and/or
Strings (sequence of characters)	`str`	concatenation, indexing

II. Variables and Literals

Code	Variable	Literal	Class of variable
`a=2`	`a`	`2`	`int`
`c = a>3`	`c`	`a>3 (False)`	`bool`
`d = a==2`	`d`	`a==2 (True)`	`bool`
`bob_friend = 'fred'`	`bob_friend`	`'fred'`	`str`

** Make sure you understand the difference between `=` and `==`

`=`	It serves to assign the literal on the right to the variable on the left - `x==y` means 'literal `y` has been assigned to `x`'
`==`	- Used as part of a Boolean literal - Allows for literal to be either `True` or `False` - `x==y` means 'is `x` equal to `y`?'

III. Operators and Functions (includes review of last class)

1. Integers, Floats (`int, float`)

1) Basic operators

Arithmetic operators:
+ (addition), - (subtraction), / (division), // (floor division), %(modulus), **(exponent), *(multiplication)

Comparison operators:
== (equal), != (unequal), >, <, <=, >=

2) Functions in `math` module

First, the keyword `import`: brings in functions that are not built-in from a module (e.g. math module)

Eg) Run the following code and understand what each line is printing

**notice how we first import `math`, then also specify the math module again when calling the constants and functions from this module: e.g. `math.pi`, not just `pi`

```
import math
print('some constants in Math module')
print('pi: ',math.pi)

a=3
print('some common functions')
print('square root of a: ', round(math.sqrt(a),2))
```

- `math.pi`: calling the constant `pi` within the `math` module
- `math.sqrt(a)`: calling the function `sqrt` within the `math` module on the variable `a`
- `round(math.sqrt(a),2)`: rounding the square root of `a` to 2 decimal points

3) Casting between `float` and `int`

```
In: int(7.99)
Out: 7

In: float(7)
Out: 7.0
```

2. Booleans

1) Boolean logic: `and, or, not`

2) Casting between `bool` and `int`

```
In: bool(1)
Out: True

In: bool(0)
Out: False

In: int(True)
Out: 1

In: int(False)
Out: 0
```

3. Strings

1) Concatenation

```
In: a= 'happy'+'joy'
In: a
Out: 'happyjoy'
```

2) Indexing and splicing

- To obtain the **character at the x(th) index** of the string `s`, you use `s[x]`
- To obtain the **length** of the string `s`, you use `len(s)`
- **Splicing**: To obtain all characters from 0^{th} index to 5^{th} index of the string `s`, you use `s[0:6]` (Remember that the 'end' index is exclusive, i.e. not included in your 'splice')

e.g. Make sure you understand the following:

```
In[1]: s = 'ihatethecold'

In[2]: [0]
Out[2]: 'i'

In[3]: s[1]
Out[3]: 'h'

In[4]: s[0:2]
Out[4]: 'ih'

In[5]: s[3:8]
Out[5]: 'tethe'

In[6]: len(s)
Out[6]: 12

In[7]: s[0:len(s)-1]
Out[7]: 'ihatethecol'
```

4. Input/Output functions

1) We already know **output** function: `print('…')`

2) Input:

variable_name = input("*string to print before user enters info*")

- waits for the user to input information and assigns that string to the variable
- To make the input info into a different datatype other than a string, we must use the casting functions seen above: `int()`, `float()`, or `bool()`

Eg)

```
name=input("What's your name dude?: ")
fav1=input("What's your favorite color?: ")
fav2=int(input("What's your favorite number?: "))
fav3=float(input("What's your favorite non-integer number?: "))
```

IV. Control Flow (if/else/elif)

```
if (Boolean expression1):
   things to execute if expression1 is true

elif (Boolean expression2):
   things to execute if 1 is false and 2 is true

else:
   things to execute if 1 and 2 are false
```

Eg) '3' will print out in the example below

```
a=20
if(a<10):
    b=12
elif(a>30):
    b=15
else:
    b=3
print b
```

Eg2) Run both case 1 and case 2 and make sure you understand why the output is different!

Case1>>

```
x=3
if(x==3):
    print 'happy'
print 'sad'
```

Case 2>>

```
if(x==3):
    print 'happy'
else:
    print 'sad'
```