Midterm knowledge sheet by Percy Teng <3

Arithmetic Operation:

* +, -, \*: addition, subtraction, multiplication just like math. Return a number (int/float)
  + 5 + 5 = 10;
  + 5.5 + 4.5 = 10.0
* /: division only returns a FLOAT
  + 5/2 = 2.5
  + 5/1 = 5.0
  + 5/0 >>> ZeroDivisionError
* //: floor division returns a int from flooring the result.
  + 9//3 = 3;
  + 10//3 = 3;
  + 11//3 = 3; (11/3 = 3.66667 but we floor it down to 3)
  + -11 // 3 = -4
  + 5//0 >>>ZeroDivisionError
* \*\*: exponents, returns a number
  + 5\*\*5 = 25
  + 2.5\*\*2.5 == 9.882117688026186
* %: modulus, return a number
  + 12 % 5 = 5 \* 2 + 2 -> 2
  + 12 % 3 = 0
* Operation order: \*\* (exponents) -> /,//,\*,% (multiplication, division, modulus) -> +, - (addition, subtraction)

Variables:

* Every value has its own memory address; variables are meant to store those address.
* Use function id() to find out the address that variables store if needed
* var\_int = 100;
* var\_str = ‘percy’
* var\_str\_int = ‘1’
  + e.g: 1 + 1 = 2 -> + between numbers are addition
  + e.g: ‘1’ + ‘1’ = ‘11’ -> + between strings are concatenation
  + >>>s = 'Jacqueline'
  + >>> s[1::2]
  + 'aqeie'
  + >>> s[::-1]
  + 'enileuqcaJ'
  + >>> s[-1::-2]
  + 'eieqa'
* var\_float = 3.1415
* var\_bool = True
  + or: one boolean value on each side, returns True if one or both side are True
    - 1 < 3 or 3 < 2 => True
    - 1 < 3 or 3 > 2 => True
    - 1 > 3 or 3 < 2 => False
* List in python: container/wrapper for a group of elements (different types of elements)
  + Create a list:

>>>list\_example = [1,2,3,4]

>>>list\_example

[1,2,3,4]

* + Access elements of a list:

>>>list\_example[0]

1

>>>list\_example[1:3]

[2,3]

>>> list\_example[-1]

4

Note: list[a:b] means get the elements from index a to index b-1.

* + Update elements of a list:

>>>list\_example[0] = 2

>>>list\_example

[2,2,3,4]

* + Get how many elements(length) of a list: use function len()

>>>len(list\_example)

4

* + concatenation:

>>> a = [1,2]

>>> b = [3,4]

>>> a + b

[1,2,3,4]

>>> a + 1

TypeError: can only concatenate list (not "int") to list

* + Deletion:
    - pop() function takes out the last element and return it
    - remove() takes an object as argument and remove it from the list

>>>list\_example = [1,2,3,4]

>>>list\_example.pop()

4

>>>list\_example

[1,2,3]

>>>list\_example.remove(2)

>>>list\_example

[1, 3]

* Note: You can access a character of a string the same way accessing elements in a list but you can’t update the value of a string since strings are immutable.

>>> str = ‘percy’

>>>str[0]

‘p’

>>>str[0] = ‘q’

TypeError: 'str' object does not support item assignment

* IMPORTANT:

>>>list\_one = [1,2,3,4]

>>>list\_two = list\_one

>>> list\_two[0] = [3]

>>>list\_two

[3,2,3,4]

>>>list\_one

[3,2,3,4]

* list\_one stores the memory address of [1,2,3,4], same as list\_two, so when we change the value of that list, both lists got updated while they are still pointing to the same memory address.

>>>int\_one = 5

>>>int\_two = int\_one

>>>int\_two = 6

>>>int\_two

6

>>>int\_one

5

* int\_one and int\_two once store the same address, but we change the memory address that int\_two stores, so they are not pointing to the same address anymore.
* If you are CONFUSED, just remember these two EXAMPLES!

Built in Functions

* print(...) : display/show on the screen

>>>print(‘hello, world’)

hello, world

* max()/min(): Returns the greatest/smallest value from the arguments
  + They take at least 1 arguments, and if there is only one argument, it has to be an iterable object such as list

>>>max(1,2,3,4,5,6)

6

>>>min(1)

TypeError: 'int' object is not iterable

>>>min([1,2,3,4])

1

* round(..)
  + round( 3.6) = 4
  + round (5.5) = 6
  + round(4.5) = ?
  + round(4.5) = 4
  + Only round up when integer part is an odd number
  + round(533.333333, 2) = 533.33
  + round(533.3333, -2 ) = ? round(555.33333, -2) = ?
  + round(533.3333, -2 ) = 500; round(555.33333, -2) = 600
* sum([list of number])

>>>sum([1,2,3,4])

10

>>>sum(1,2)

TypeError: 'int' object is not iterable

* find(): return Index if found and -1 otherwise.

str1 = "this is string example....wow!!!";

str2 = "exam";

print str1.find(str2)

print str1.find(str2, 10)

print str1.find(str2, 40)

15

15

-1