

Simple data types and operators

In [0]:

```
name = 'iPhone 5s'
```

In [0]:

```
type(name)
```

Out[0]:

str

In [0]:

```
print(name)
```

iPhone 5s

In [0]:

```
screen_size = 4.3  
type(screen_size)
```

Out[0]:

float

In [0]:

```
num_of_speakers = 1
```

In [0]:

```
type(num_of_speakers)
```

Out[0]:

int

In [0]:

```
print('I just bought an ' + name)
```

I just bought an iPhone 5s

In [0]:

```
a = 3  
b = 2  
print(a * b, a / b, a + b, a - b, a ** b, a // b)
```

6 1.5 5 1 9 1

In [0]:

```
is_android = False  
type(is_android)
```

Out[0]:

bool

In [0]:

```
print(is_android == True)
```

```
false
```

```
In [0]:
```

```
print (is_android == False and num_of_speakers > 1)
```

```
False
```

Code blocks - If, For, While

```
In [0]:
```

```
if is_android == True:
    print(name + ' supports Android')
    print('Continuing to print inside the if block')
print('Printing this outside the block')
```

```
Printing this outside the block
```

```
In [0]:
```

```
for i in range(5):
    print(i, i**2)
```

```
0 0
1 1
2 4
3 9
4 16
```

```
In [0]:
```

```
range[?]
```

```
In [0]:
```

```
i = 0
while i < 5:
    print(i, i**2)
    i += 1 # equivalent to i = i + 1
```

```
0 0
1 1
2 4
3 9
4 16
```

Functions

```
In [0]:
```

```
def print_squares(stop_val):
    for i in range(stop_val):
        print (i , i**2)
```

```
In [0]:
```

```
print_squares(5)
```

```
0 0
1 1
2 4
3 9
4 16
```

```
In [0]:
```

```
print_squares(3)
```

```
0 0
1 1
2 4
```

In [0]:

```
def sum_of_squares(stop_val):
    sum_ = 0
    for i in range(stop_val):
        sum_ += i ** 2
    return sum_
```

In [0]:

```
sum_of_squares(5)
```

Out[0]:

```
30
```

In [0]:

```
sum_5 = sum_of_squares(5)
```

In [0]:

```
def print_squares_with_startval(stop_val, start_val = 0):
    """Prints the squares of numbers starting from start_val (inclusive) and ending with stop_val (exclusive)"""
    for i in range(start_val, stop_val):
        print (i , i**2)
```

In [0]:

```
print_squares_with_startval(5, 2)
```

```
2 4
3 9
4 16
```

In [0]:

```
print_squares_with_startval[?]
```

Lists

In [0]:

```
my_phone = [name, is_android, screen_size]
```

In [0]:

```
type(my_phone)
```

Out[0]:

```
list
```

In [0]:

```
print(my_phone)
```

```
['iPhone 5s', False, 4.3]
```

In [0]:

```
my_phone[2]
```

Out[0]:

4.3

In [0]:

```
type(my_phone[0])
```

Out[0]:

str

In [0]:

```
my_phone[1:3]
```

Out[0]:

[False, 4.3]

In [0]:

```
my_dads_phone = my_phone  
print(my_dads_phone)
```

['iPhone 5s', False, 4.3]

In [0]:

```
my_phone[2] = 4
```

In [0]:

```
print(my_phone)
```

['iPhone 5s', False, 4]

In [0]:

```
print(my_dads_phone)
```

['iPhone 5s', False, 4]

In [0]:

```
my_dads_phone = list(my_phone)  
# my_dads_phone = my_phone[:]
```

In [0]:

```
my_phone[2] = 4.3
```

In [0]:

```
print(my_phone)  
print(my_dads_phone)
```

['iPhone 5s', False, 4.3]

['iPhone 5s', False, 4]

In [0]:

```
len(my_phone)
```

Out[0]:

3

In [0]:

```
len?
```

In [0]:

```
my_phone.append('3G')
```

In [0]:

```
print(my_phone)
```

```
['iPhone 5s', False, 4.3, '3G']
```

In [0]:

```
len(my_phone)
```

Out[0]:

```
4
```

In [0]:

```
iphone5s_some_prefs = ['single_sim', 35000]
```

In [0]:

```
my_phone = my_phone + iphone5s_some_prefs
```

In [0]:

```
print(my_phone)
```

```
['iPhone 5s', False, 4.3, '3G', 'single_sim', 35000]
```

In [0]:

```
for data in my_phone:  
    print(data)
```

```
iPhone 5s  
False  
4.3  
3G  
single_sim  
35000
```

In [0]:

```
range(5)
```

Out[0]:

```
range(0, 5)
```

In [0]:

```
print(list(range(5)))
```

```
[0, 1, 2, 3, 4]
```

In [0]:

```
a = list(range(5))  
print(a)
```

```
[0, 1, 2, 3, 4]
```

In [0]:

```
b = list(map(lambda x: x**2, a))
```

In [0]:

```
print(b)
```

```
[0, 1, 4, 9, 16]
```

In [0]:

```
c = list(filter(lambda x: x % 2 == 0, b))
print(c)
```

[0, 4, 16]

In [0]:

```
print(my_phone)
type(my_phone)
```

['iPhone 5s', False, 4.3, '3G', 'single_sim', 35000]

Out[0]:

list

Tuples, Sets, and Dicts

In [0]:

```
my_phone_tuple = ('iPhone 5s', False, 4.3)
print(my_phone_tuple)
type(my_phone_tuple)
```

('iPhone 5s', False, 4.3)

Out[0]:

tuple

In [0]:

```
my_phone_tuple[0:2]
```

Out[0]:

('iPhone 5s', False)

In [0]:

```
for data in my_phone_tuple:
    print(data)
```

iPhone 5s
False
4.3

In [0]:

```
my_phone_tuple[2] = 4
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-73-01379ee61114> in <module>()
----> 1 my_phone_tuple[2] = 4
```

TypeError: 'tuple' object does not support item assignment

In [0]:

```
import time
```

In [0]:

```
tic = time.time()
print(tic)
```

1548596668.6366014

In [0]:

```
time.time[?]
```

In [0]:

```
tic = time.time()
# do something that takes some time that is to be measured
toc = time.time()
print('Elapsed time in seconds', toc - tic)
```

Elapsed time in seconds 1.8596649169921875e-05

In [0]:

```
my_list = list(range(1000000))
tic = time.time()
my_list_squared = map(lambda i: i**2, my_list)
toc = time.time()
print('Elapsed time in seconds', toc - tic)
```

Elapsed time in seconds 0.013445615768432617

In [0]:

```
my_tuple = tuple(range(1000000))
tic = time.time()
my_tuple_squared = map(lambda i: i**2, my_tuple)
toc = time.time()
print('Elapsed time in seconds', toc - tic)
```

Elapsed time in seconds 0.012768983840942383

In [0]:

```
my_set = set(my_phone)
```

In [0]:

```
print(my_set)
```

{False, 4.3, 'single_sim', '3G', 35000, 'iPhone 5s'}

In [0]:

```
my_phone.append(4.3)
```

In [0]:

```
print(my_phone)
```

['iPhone 5s', False, 4.3, '3G', 'single_sim', 35000, 4.3]

In [0]:

```
my_set = set(my_phone)
```

In [0]:

```
print(my_set)
```

{False, 4.3, 'single_sim', '3G', 35000, 'iPhone 5s'}

In [0]:

```
my_list = list(range(1000000))
tic = time.time()
print(98731 in my_list)
print(131591 in my_list)
print(1111111111 in my_list)
```

```
toc = time.time()
print('Elapsed time', toc - tic)
```

```
True
True
False
Elapsed time 0.018808364868164062
```

In [0]:

```
my_set = set(range(1000000))
tic = time.time()
print(98731 in my_set)
print(131591 in my_set)
print(1111111111 in my_set)
toc = time.time()
print('Elapsed time', toc - tic)
```

```
True
True
False
Elapsed time 0.0013599395751953125
```

In [0]:

```
my_phone_dict = {}
```

In [0]:

```
type(my_phone_dict)
```

Out[0]:

```
dict
```

In [0]:

```
my_phone_dict['name'] = 'iPhone 5s'
my_phone_dict['isAndroid'] = False
my_phone_dict['screenSize'] = 4.3
```

In [0]:

```
print(my_phone_dict)
```

```
{'name': 'iPhone 5s', 'isAndroid': False, 'screenSize': 4.3}
```

In [0]:

```
my_phone_dict['name']
```

Out[0]:

```
'iPhone 5s'
```

In [0]:

```
print(my_phone_dict.keys())
```

```
dict_keys(['name', 'isAndroid', 'screenSize'])
```

In [0]:

```
print(my_phone_dict.values())
```

```
dict_values(['iPhone 5s', False, 4.3])
```

In [0]:

```
print(my_phone_dict.items())
```

```
dict_items([('name', 'iPhone 5s'), ('isAndroid', False), ('screenSize', 4.3)])
```


In [0]:

```
for key in my_phone_dict.keys():  
    print(key, ': ', my_phone_dict[key])
```

```
name : iPhone 5s  
isAndroid : False  
screenSize : 4.3
```

Python Packages

In [0]:

```
import math
```

In [0]:

```
a = math.sqrt(100)  
print(a)
```

```
10.0
```

In [0]:

```
a = math.pow(100, 0.5)  
print(a)
```

```
10.0
```

In [0]:

```
x = 100  
y = 1  
for i in range(1, x):  
    y *= i  
print('Factorial of', x, 'is', y)
```

```
Factorial of 100 is 933262154439441526816992388562667004907159682643816214685929638952175  
999932299156089414639761565182862536979208272237582511852109168640000000000000000000000
```

In [0]:

```
y = math.factorial(x)
```

In [0]:

```
import math as m
```

In [0]:

```
y = m.factorial(x)
```

In [0]:

```
from math import factorial
```

In [0]:

```
y = factorial(x)
```

In [0]:

```
vals = list(range(1, 100))  
tic = time.time()  
for x in vals:  
    y = 1  
    for i in range(1, x):  
        y *= i
```

```

toc = time.time()
print('Elapsed time in secs with own function', toc - tic)

tic = time.time()
for x in vals:
    y = math.factorial(x)
toc = time.time()
print('Elapsed time in secs with own function', toc - tic)

```

Elapsed time in secs with own function 0.0009882450103759766
Elapsed time in secs with own function 0.0001125335693359375

In [0]:

```

!echo 'def hello():' > my_first_module.py
!echo '    print("hello, i am living in a different file!!!")' >> my_first_module.py

```

In [0]:

```

!cat my_first_module.py

```

```

def hello():
    print("hello, i am living in a different file!!!")

```

In [0]:

```

import my_first_module

```

In [0]:

```

my_first_module.hello()

```

hello, i am living in a different file!!!

In [0]:

```

from my_first_module import hello

```

In [0]:

```

hello()

```

hello, i am living in a different file!!!

File handling

In [0]:

```

!wget https://www.dropbox.com/s/w94odi4aq1k441f/mobile_cleaned.csv

```

```

--2019-01-29 04:28:34--  https://www.dropbox.com/s/w94odi4aq1k441f/mobile_cleaned.csv
Resolving www.dropbox.com (www.dropbox.com)... 162.125.1.1, 2620:100:6016:1::a27d:101
Connecting to www.dropbox.com (www.dropbox.com)|162.125.1.1|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /s/raw/w94odi4aq1k441f/mobile_cleaned.csv [following]
--2019-01-29 04:28:34--  https://www.dropbox.com/s/raw/w94odi4aq1k441f/mobile_cleaned.csv
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com/cd/0/inline/AaRLKMBgAIR8slYjOWY514P50mlwKgb8cJPirxOdMrTbVk9GjcHfXlJ7_BngxlHfBKutYm5PN-ZYWw4GIumNdUkZMMlPYrn5NCKbvmNozcCNcvqVs5QoswZ35QckW9HMXQk/file# [following]
--2019-01-29 04:28:34--  https://uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com/cd/0/inline/AaRLKMBgAIR8slYjOWY514P50mlwKgb8cJPirxOdMrTbVk9GjcHfXlJ7_BngxlHfBKutYm5PN-ZYWw4GIumNdUkZMMlPYrn5NCKbvmNozcCNcvqVs5QoswZ35QckW9HMXQk/file
Resolving uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com (uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com)... 162.125.1.6, 2620:100:6016:6::a27d:106
Connecting to uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com (uc0dbee8e87e6f20a0ef0deb5bf3.dl.dropboxusercontent.com)|162.125.1.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14044 (14K) [text/plain]

```

Saving to: 'mobile_cleaned.csv'

mobile_cleaned.csv 100%[=====>] 13.71K --.-KB/s in 0s

2019-01-29 04:28:34 (148 MB/s) - 'mobile_cleaned.csv' saved [14044/14044]

In [0]:

```
ls
```

mobile_cleaned.csv my_first_module.py __pycache__ sample_data

In [0]:

```
file = open('mobile_cleaned.csv', 'r')
```

In [0]:

```
s = file.readline()
```

In [0]:

```
print(s)
```

```
sim_type,aperture,gpu_rank,weight,stand_by_time,processor_frequency,thickness,flash_type,
front_camera_resolution,auto_focus,screen_size,frames_per_second,FM,no_of_reviews_in_gsmarena_in_week,os,phone_height,screen_protection,sim_size,price,talk_time,video_resolution,display_resolution,removable_battery,display_type,primary_camera_resolution,battery_type,ram_memory,internal_memory,brand_rank,no_of_cores,micro_sd_slot,screen_pixel_density,water_proof_rate,phone_width,expandable_memory,version,usb_type,battery_capacity,processor_rank,is_liked
```

In [0]:

```
print(s.split(','))
```

```
['sim_type', 'aperture', 'gpu_rank', 'weight', 'stand_by_time', 'processor_frequency', 'thickness', 'flash_type', 'front_camera_resolution', 'auto_focus', 'screen_size', 'frames_per_second', 'FM', 'no_of_reviews_in_gsmarena_in_week', 'os', 'phone_height', 'screen_protection', 'sim_size', 'price', 'talk_time', 'video_resolution', 'display_resolution', 'removable_battery', 'display_type', 'primary_camera_resolution', 'battery_type', 'ram_memory', 'internal_memory', 'brand_rank', 'no_of_cores', 'micro_sd_slot', 'screen_pixel_density', 'water_proof_rate', 'phone_width', 'expandable_memory', 'version', 'usb_type', 'battery_capacity', 'processor_rank', 'is_liked\n']
```

In [0]:

```
file.close()
```

In [0]:

```
file.readline()
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-30-1ba352a1fd91> in <module>()
----> 1 file.readline()
```

ValueError: I/O operation on closed file.

In [0]:

```
with open('mobile_cleaned.csv', 'r') as file:
    print(file.readline())
```

```
sim_type,aperture,gpu_rank,weight,stand_by_time,processor_frequency,thickness,flash_type,
front_camera_resolution,auto_focus,screen_size,frames_per_second,FM,no_of_reviews_in_gsmarena_in_week,os,phone_height,screen_protection,sim_size,price,talk_time,video_resolution,display_resolution,removable_battery,display_type,primary_camera_resolution,battery_type,
```

```
ram_memory,internal_memory,brand_rank,no_of_cores,micro_sd_slot,screen_pixel_density,water_proof_rate,phone_width,expandable_memory,version,usb_type,battery_capacity,processor_rank,is_liked
```

In [0]:

```
with open('mobile_cleaned.csv', 'r') as file:
    print(file.read())
```

In [0]:

```
with open('mobile_cleaned.csv', 'r') as file:
    for line in file:
        for word in line.split(','):
            print(word)
        print('-----')
```

In [0]:

```
with open('my_first_file_output.txt', 'w') as file:
    file.write('hello world from python code')
```

In [0]:

```
!cat my_first_file_output.txt
```

hello world from python code

Python Classes

In [0]:

```
class MobilePhone:
    """This is a sample class to illustrate how Python classes work"""
    def __init__(self, name, is_android = False, screen_size = 4.3):
        self.name = name
        self.is_android = is_android
        self.screen_size = screen_size
        self.rating = -1

    def has_rating(self):
        return self.rating > -1
```

In [0]:

```
new_phone = MobilePhone('iPhone 5s')
```

In [0]:

```
type(new_phone)
```

Out[0]:

```
__main__.MobilePhone
```

In [0]:

```
print(new_phone.name, new_phone.is_android, new_phone.screen_size)
```

iPhone 5s False 4

In [0]:

```
new_phone.screen_size = 4
```

In [0]:

```
new_phone.has_rating()
```

```
Out[0]:
```

```
True
```

```
In [0]:
```

```
new_phone.rating = 3.9
```

```
In [0]:
```

```
MobilePhone.__doc__
```

```
Out[0]:
```

```
'This is a sample class to illustrate how Python classes work'
```

```
In [0]:
```

```
class iPhone(MobilePhone):
    def __init__(self, name):
        MobilePhone.__init__(self, name, False, 4)

    def __str__(self):
        return self.name + " " + str(self.is_android) + " " + str(self.screen_size)
```

```
In [0]:
```

```
new_iphone = iPhone('iPhone 5s')
```

```
In [0]:
```

```
new_iphone.is_android
```

```
Out[0]:
```

```
False
```

```
In [0]:
```

```
print(new_iphone)
```

```
iPhone 5s False 4
```

NumPy

```
In [0]:
```

```
import numpy as np
```

```
In [0]:
```

```
X = np.array([1, 2, 3])
print(X)
```

```
[1 2 3]
```

```
In [0]:
```

```
type(X)
```

```
Out[0]:
```

```
numpy.ndarray
```

```
In [0]:
```

```
l = [4, 5, 10]
X = np.asarray(l)
type(X)
```

```
Out[0]:  
numpy.ndarray
```

```
In [0]:
```

```
print(X)
```

```
[ 4  5 10]
```

```
In [0]:
```

```
X = np.asarray(1, float)  
print(X)
```

```
[ 4.  5. 10.]
```

```
In [0]:
```

```
X.shape
```

```
Out[0]:
```

```
(3,)
```

```
In [0]:
```

```
np.array[?]
```

```
In [0]:
```

```
X = np.array([[1, 2, 3], [4, 5, 6]])
```

```
In [0]:
```

```
print(X)
```

```
[[1 2 3]  
 [4 5 6]]
```

```
In [0]:
```

```
X.shape
```

```
Out[0]:
```

```
(2, 3)
```

```
In [0]:
```

```
X[1, 2]
```

```
Out[0]:
```

```
6
```

```
In [0]:
```

```
X[0, 0:2]
```

```
Out[0]:
```

```
array([1, 2])
```

```
In [0]:
```

```
X[0, :]
```

```
Out[0]:
```

```
array([1, 2, 3])
```

```
In [0]:
```

```
X = np.zeros((4, 5))
```

```
In [0]:
```

```
print(X)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

```
In [0]:
```

```
X = np.eye(4, 4)
```

```
In [0]:
```

```
print(X)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

```
In [0]:
```

```
X = np.random.random((4, 5))
```

```
In [0]:
```

```
print(X)
```

```
[[0.87341778 0.90078901 0.91818823 0.33747593 0.98491756]
 [0.33485602 0.25379737 0.90456387 0.02067163 0.90923984]
 [0.78282717 0.64131637 0.86989012 0.82875304 0.52809011]
 [0.20675582 0.33911448 0.51724198 0.13715521 0.12568884]]
```

```
In [0]:
```

```
Z = X.T
```

```
In [0]:
```

```
print(Z)
```

```
[[0.87341778 0.33485602 0.78282717 0.20675582]
 [0.90078901 0.25379737 0.64131637 0.33911448]
 [0.91818823 0.90456387 0.86989012 0.51724198]
 [0.33747593 0.02067163 0.82875304 0.13715521]
 [0.98491756 0.90923984 0.52809011 0.12568884]]
```

```
In [0]:
```

```
Y = X.reshape(20, 1)
```

```
In [0]:
```

```
print(Y)
```

```
[[0.87341778]
 [0.90078901]
 [0.91818823]
 [0.33747593]
 [0.98491756]
 [0.33485602]
 [0.25379737]
 [0.90456387]
 [0.02067163]
 [0.90923984]
 [0.78282717]
 [0.64131637]
```

```
[0.86989012]  
[0.82875304]  
[0.52809011]  
[0.20675582]  
[0.33911448]  
[0.51724198]  
[0.13715521]  
[0.12568884]]
```

In [0]:

```
A = np.arange(5)  
print(A)
```

```
[0 1 2 3 4]
```

In [0]:

```
B = np.arange(5)
```

In [0]:

```
C = A + B  
D = A - B  
E = A * B  
print(C, D, E)
```

```
[0 2 4 6 8] [0 0 0 0 0] [ 0  1  4  9 16]
```

In [0]:

```
print(A, A + 1)
```

```
[0 1 2 3 4] [1 2 3 4 5]
```

In [0]:

```
A = np.random.random((2, 3))  
print(A)  
print(A + 1)
```

```
[[0.55610055 0.99531044 0.92377496]  
 [0.34752321 0.77827385 0.86643146]]  
[[1.55610055 1.99531044 1.92377496]  
 [1.34752321 1.77827385 1.86643146]]
```

In [0]:

```
A = np.floor(np.random.random((2, 3))*10)  
print(A)
```

```
[[4.  1.  9.]  
 [4.  9.  8.]]
```

In [0]:

```
u = [1, 2, 3]  
v = [-1, 0, 1]  
  
p1 = np.inner(u, v)  
print(p1)
```

```
2
```

In [0]:

```
p2 = np.outer(u, v)  
print(p2)
```

```
[[-1  0  1]  
 [-2  0  2]  
 [-3  0  3]]
```


In [0]:

```
A = np.ones((2, 3))
B = np.ones((3, 2))
np.dot(A, B)
```

Out[0]:

```
array([[3., 3.],
       [3., 3.]])
```

In [0]:

```
np.dot(B.T, A.T)
```

Out[0]:

```
array([[3., 3.],
       [3., 3.]])
```

In [0]:

```
A = np.ones((2, 3))
print(A)
```

```
[[1. 1. 1.]
 [1. 1. 1.]]
```

In [0]:

```
A.sum()
```

Out[0]:

```
6.0
```

In [0]:

```
A.sum(axis = 0)
```

Out[0]:

```
array([2., 2., 2.])
```

In [0]:

```
A.sum(axis = 1)
```

Out[0]:

```
array([3., 3.])
```

In [0]:

```
A.max()
```

Out[0]:

```
1.0
```

In [0]:

Plotting

In [0]:

```
import matplotlib.pyplot as plt
import numpy as np
```

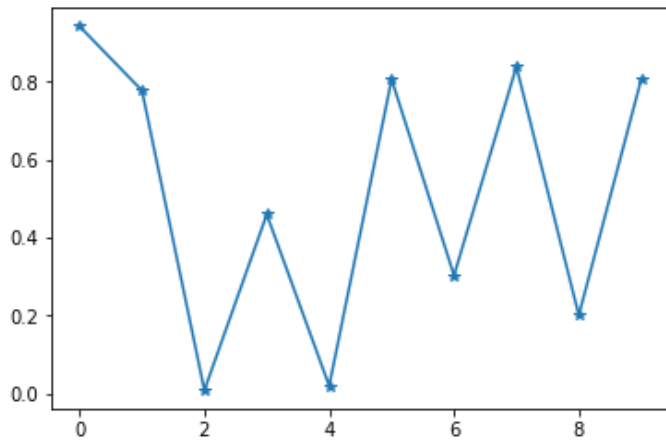
In [3]:

```
x = np.random.random((10, 1))  
print(x)
```

```
[[0.94252673]  
 [0.77750795]  
 [0.00880047]  
 [0.4599122 ]  
 [0.02001392]  
 [0.80663361]  
 [0.30345352]  
 [0.83899138]  
 [0.201209 ]  
 [0.80648529]]
```

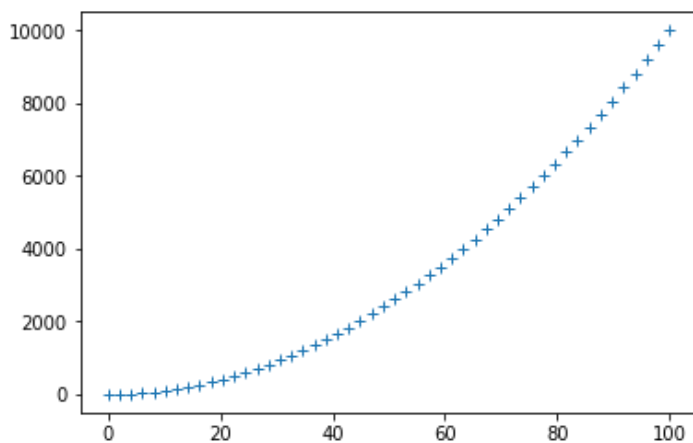
In [4]:

```
plt.plot(x, '*-')  
plt.show()
```



In [5]:

```
x = np.linspace(0, 100, 50)  
y = np.power(x, 2)  
plt.plot(x, y, '+')  
plt.show()
```



In [0]:

```
import seaborn as sns
```

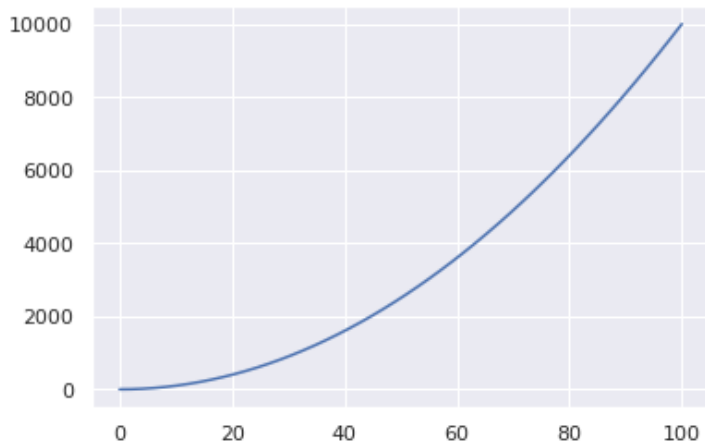
In [0]:

```
sns.set()
```

In [10]:

```
sns.lineplot(x, y)
```

```
plt.show()
```



In [11]:

```
!wget https://www.dropbox.com/s/w94odi4aqlk441f/mobile_cleaned.csv
```

```
--2019-01-30 03:39:21-- https://www.dropbox.com/s/w94odi4aqlk441f/mobile_cleaned.csv
Resolving www.dropbox.com (www.dropbox.com)... 162.125.1.1, 2620:100:601b:1::a27d:801
Connecting to www.dropbox.com (www.dropbox.com)|162.125.1.1|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /s/raw/w94odi4aqlk441f/mobile_cleaned.csv [following]
--2019-01-30 03:39:21-- https://www.dropbox.com/s/raw/w94odi4aqlk441f/mobile_cleaned.csv
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com/cd/0/inline/AaXtK-RoBPJW_KRPZLdiu7qKaot7E0VSGx4-g31vrQwTHwAMLpvgEDdF-ML9s6vxnLR9LdedLqYItxXf_OsgTlEg9zmH4yUZbBg2qeDI_D2ZCaNX8WRmv10YLeC9PobK-EU/file# [following]
--2019-01-30 03:39:21-- https://uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com/cd/0/inline/AaXtK-RoBPJW_KRPZLdiu7qKaot7E0VSGx4-g31vrQwTHwAMLpvgEDdF-ML9s6vxnLR9LdedLqYItxXf_OsgTlEg9zmH4yUZbBg2qeDI_D2ZCaNX8WRmv10YLeC9PobK-EU/file
Resolving uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com (uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com)... 162.125.1.6, 2620:100:6016:6::a27d:106
Connecting to uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com (uc9accbe78cc6be848cb8d6c33ce.dl.dropboxusercontent.com)|162.125.1.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14044 (14K) [text/plain]
Saving to: 'mobile_cleaned.csv'
```

```
mobile_cleaned.csv 100%[=====>] 13.71K --.-KB/s in 0s
```

```
2019-01-30 03:39:22 (299 MB/s) - 'mobile_cleaned.csv' saved [14044/14044]
```

In [12]:

```
!ls
```

```
mobile_cleaned.csv sample_data
```

In [13]:

```
import pandas as pd
data = pd.read_csv('mobile_cleaned.csv')
data.head()
```

Out[13]:

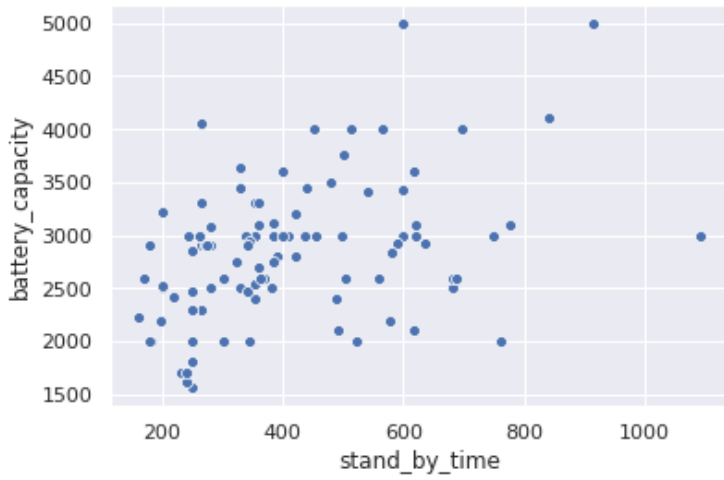
	sim_type	aperture	gpu_rank	weight	stand_by_time	processor_frequency	thickness	flash_type	front_camera_resolution
0	0	12	55	155.0	250	1.3	10.5	5	2.00
1	0	1	55	132.0	300	1.3	10.6	5	0.30
2	0	9	55	142.0	329	1.5	8.5	5	2.00
3	0	8	55	152.0	385	1.3	8.0	5	2.00
4	0	12	55	155.0	250	1.3	10.5	5	2.00

4	1	1	55	234.0	385	1.3	7.9	5	1.92
sim_type	aperture	gpu_rank	weight	stand_by_time	processor_frequency	thickness	flash_type	front_camera_resolution	

5 rows x 40 columns

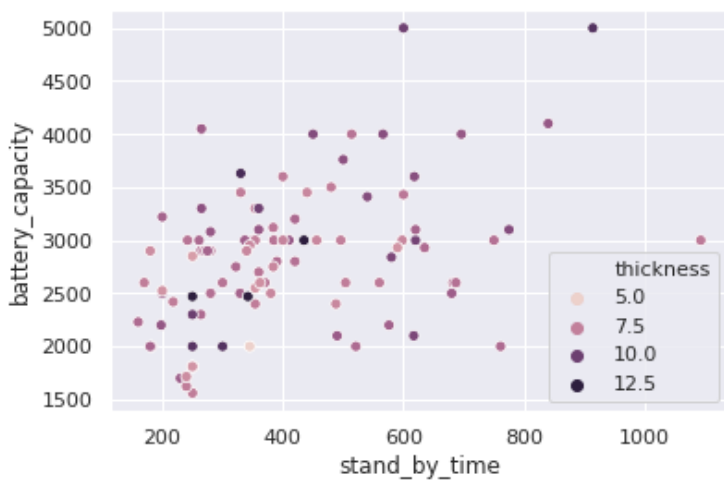
In [14]:

```
ax = sns.scatterplot(x="stand_by_time", y="battery_capacity", data=data)
```



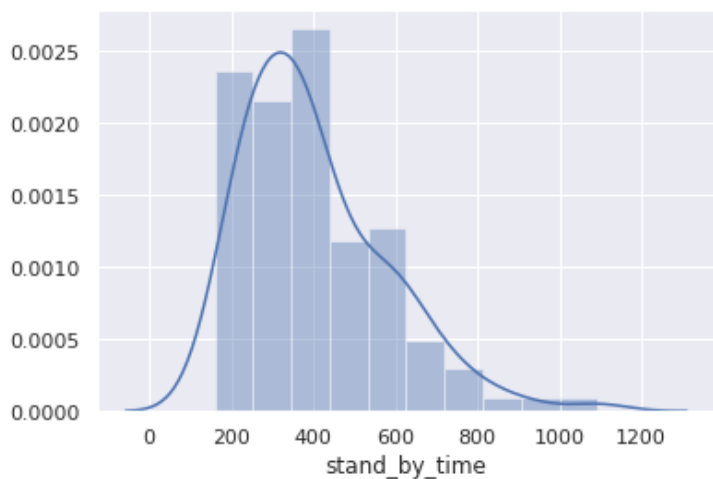
In [15]:

```
ax = sns.scatterplot(x = "stand_by_time", y = "battery_capacity", hue="thickness", data=data)
```



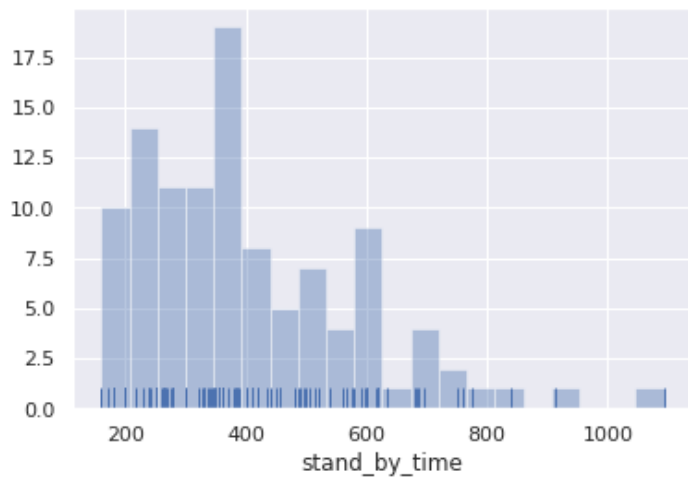
In [16]:

```
ax = sns.distplot(data["stand_by_time"])
```



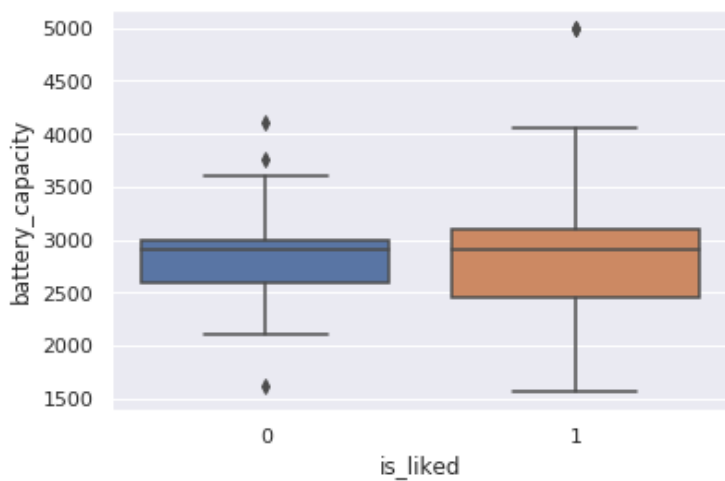
In [18]:

```
ax = sns.distplot(data["stand_by_time"], kde=False, rug=True, bins = 20)
```



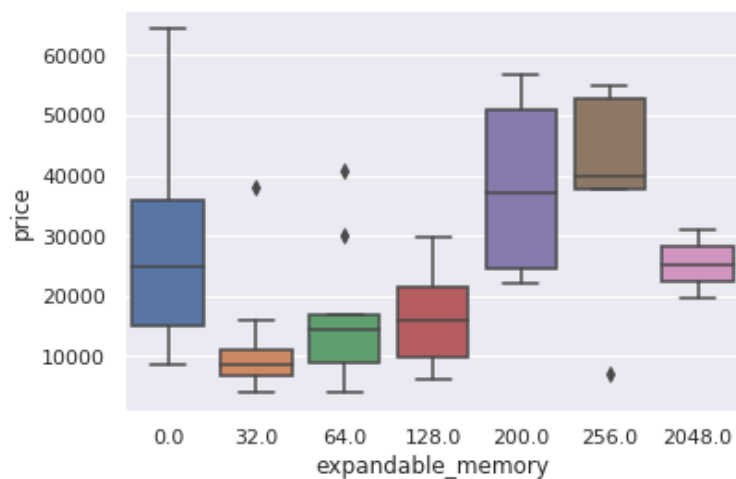
In [19]:

```
ax = sns.boxplot(x="is_liked", y="battery_capacity", data=data)
```



In [20]:

```
ax = sns.boxplot(x = "expandable_memory", y = "price", data=data)
```



In [21]:

```
uniform_data = np.random.rand(10, 12)
print(uniform_data)
```

```
[[0.62491604 0.88334463 0.76721209 0.04706149 0.64960519 0.09148659
 0.76849056 0.90272269 0.07146538 0.77627369 0.4124186 0.67956342]
 [0.45548836 0.4876875 0.22350859 0.30771614 0.60472426 0.19006129
 0.72897388 0.65494919 0.49207741 0.04270331 0.5502374 0.80623169]
 [0.69817448 0.22082512 0.30029009 0.54180497 0.83827649 0.3274147
 0.81655606 0.15802621 0.46820606 0.51228008 0.22052747 0.28225611]]
```

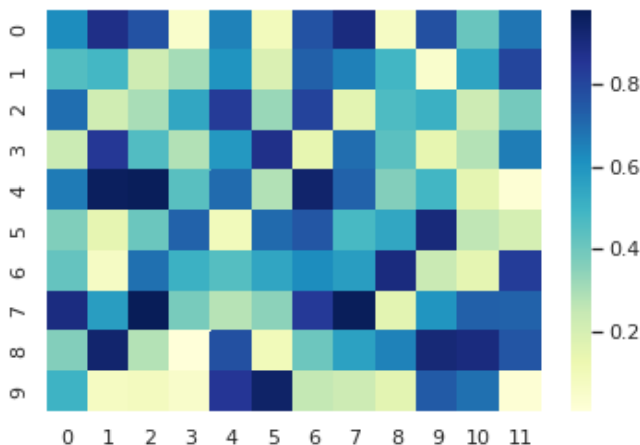
```

0.81655696 0.15802631 0.46830696 0.51228008 0.23033747 0.3893336 ]
[0.23272425 0.85067672 0.45886029 0.28411672 0.58921849 0.87861663
 0.14584435 0.69899915 0.44236351 0.14481195 0.28154491 0.66121262]
[0.66498596 0.9669128 0.97356934 0.44623722 0.70384071 0.28491051
 0.93937698 0.7257628 0.36255622 0.48903574 0.15466639 0.01934421]
[0.36955269 0.15193055 0.40823228 0.72879157 0.09543699 0.70760605
 0.75626141 0.4772465 0.54338043 0.91019615 0.26187476 0.20292168]
[0.42258257 0.06850217 0.69350717 0.51070935 0.45189624 0.54763336
 0.6232661 0.57526455 0.90104464 0.23775837 0.15510767 0.8349885 ]
[0.89823254 0.57349707 0.98107809 0.38344971 0.27614357 0.34831192
 0.84277478 0.97355667 0.15605751 0.60059912 0.72908447 0.72746959]
[0.36600451 0.93675926 0.27874127 0.00315576 0.77596193 0.10013316
 0.40733324 0.56404447 0.65192765 0.91870158 0.90328383 0.76220001]
[0.5033979 0.07450816 0.08645823 0.04982399 0.85342485 0.95025602
 0.25416557 0.23071508 0.15783928 0.7475679 0.69168648 0.01890993]]

```

In [23]:

```
ax = sns.heatmap(uniform_data, cmap="YlGnBu")
```



In [24]:

```
!wget https://s3.amazonaws.com/thinkific-import/153034/3f9yaVLKTyUGmlBPcSOA_logo_long.png
```

```

--2019-01-30 03:48:08-- https://s3.amazonaws.com/thinkific-import/153034/3f9yaVLKTyUGmlBPcSOA_logo_long.png
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.10.197
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.216.10.197|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27958 (27K) [image/png]
Saving to: '3f9yaVLKTyUGmlBPcSOA_logo_long.png'

```

```
3f9yaVLKTyUGmlBPcSO 100%[=====>] 27.30K --.-KB/s in 0.07s
```

```
2019-01-30 03:48:09 (404 KB/s) - '3f9yaVLKTyUGmlBPcSOA_logo_long.png' saved [27958/27958]
```

In [25]:

```
!ls
```

```
3f9yaVLKTyUGmlBPcSOA_logo_long.png mobile_cleaned.csv sample_data
```

In [0]:

```
!mv 3f9yaVLKTyUGmlBPcSOA_logo_long.png onefourthlabs.png
```

In [27]:

```
!ls
```

```
mobile_cleaned.csv onefourthlabs.png sample_data
```

In [0]:

```
import matplotlib.image as mpimg
```

```
In [0]:
```

```
img = mpimg.imread('onefourthlabs.png')
```

```
In [30]:
```

```
print(img)
```

```
[[[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]

 [[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]

 [[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]

 ...

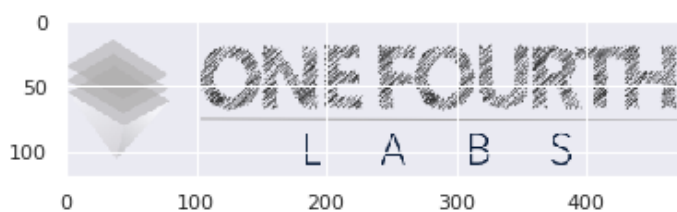
 [[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]

 [[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]

 [[0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  ...
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]
  [0. 0. 0. 0.]]]
```

```
In [31]:
```

```
imgplot = plt.imshow(img)
```



In [0]:

Pandas

In [0]: