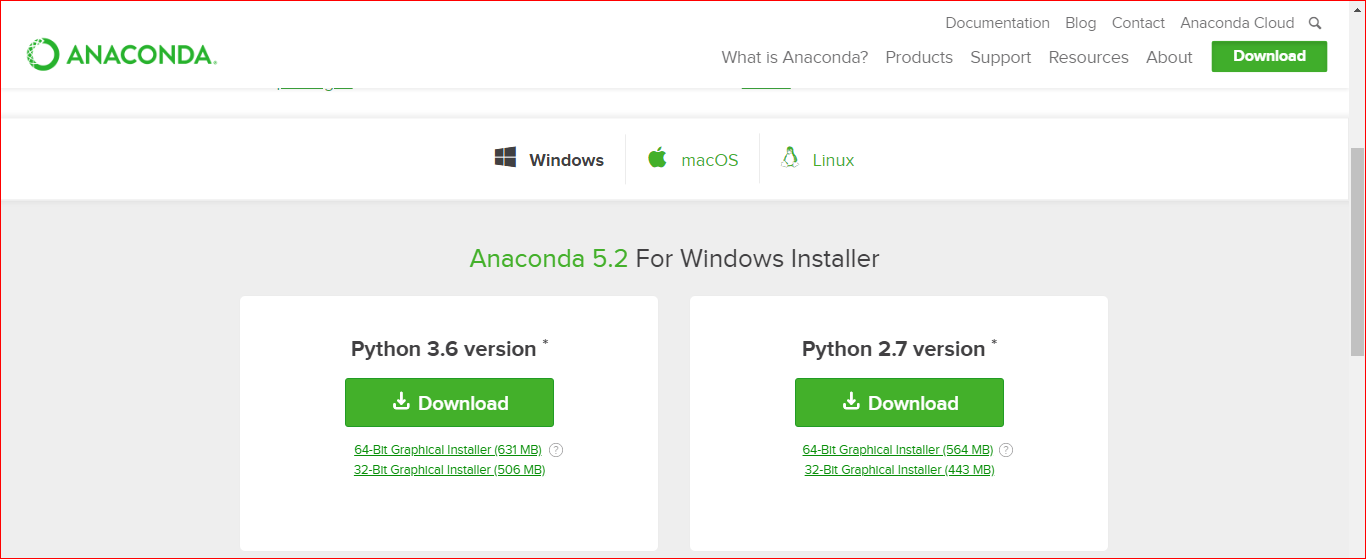
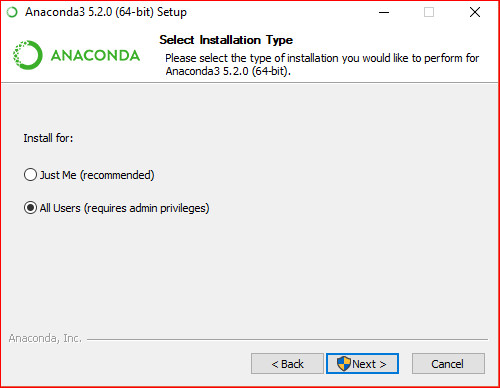
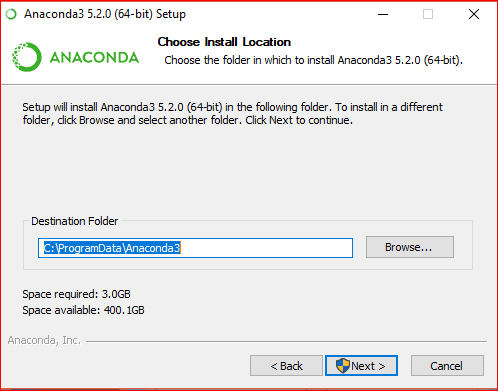
## Columbia University

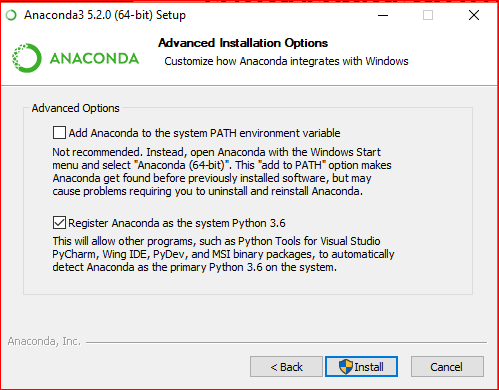
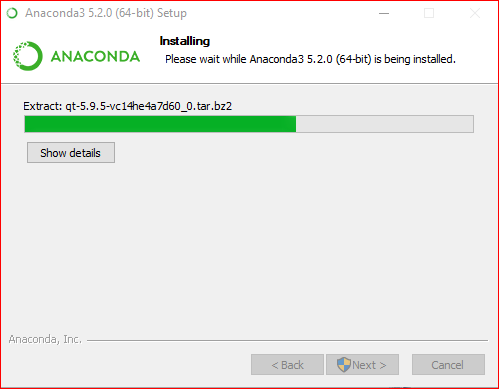
**How to install Anaconda on Windows**

<https://www.anaconda.com/what-is-anaconda/>



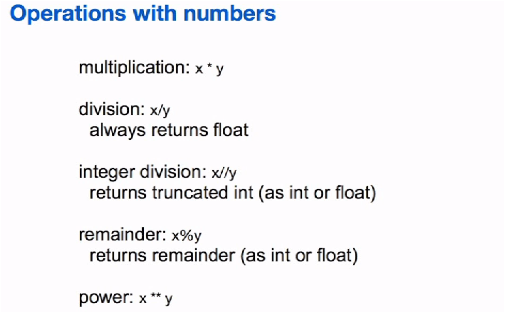
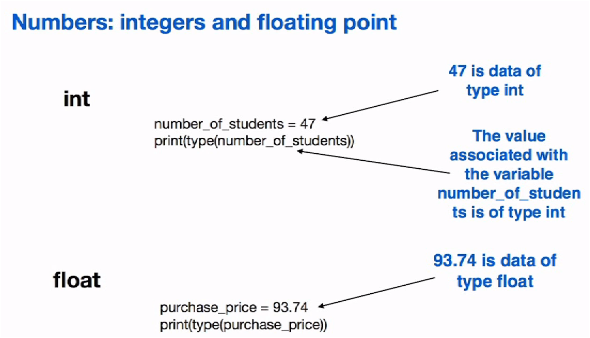
Durante el proceso de instalación, Habrá que seleccionar el PATH de la implementación de Python 3.5 en la carpeta del sistema.

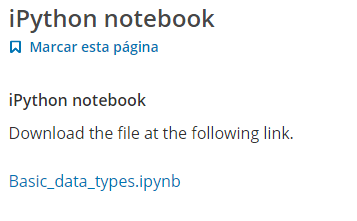
 

### Week1:

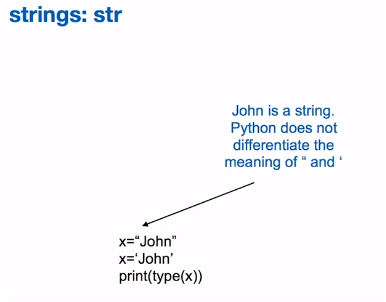
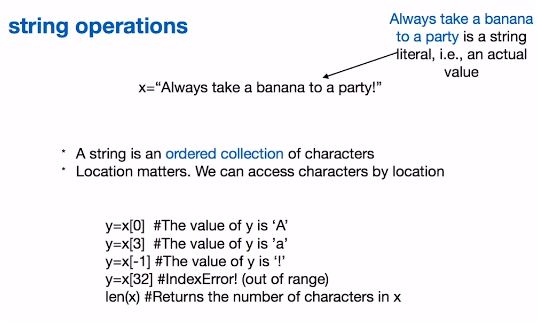
## 1.1 Python data types

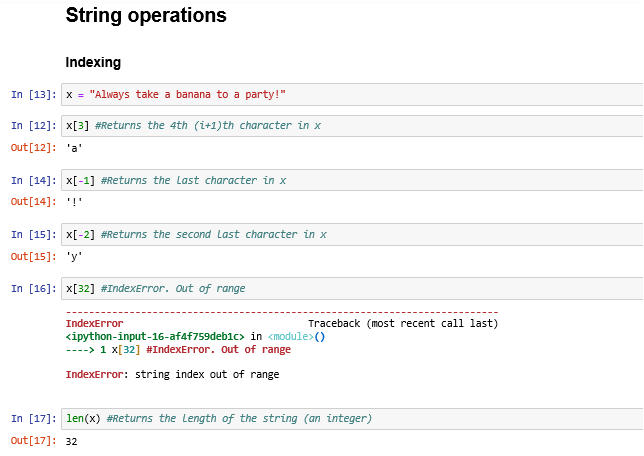


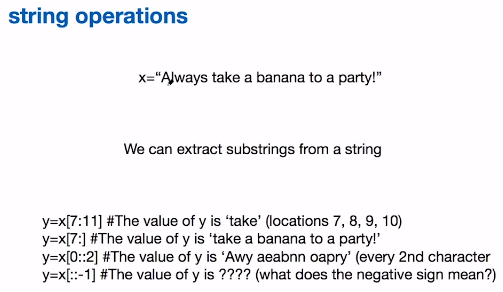


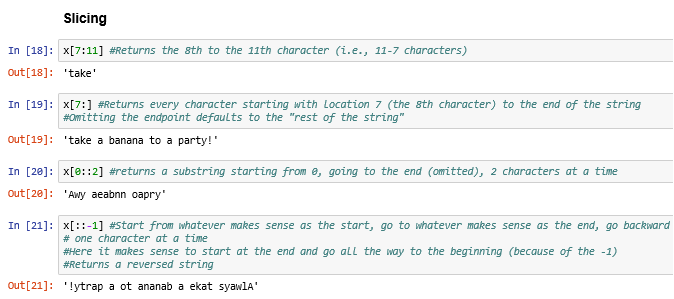


## 1.2 Strings in Python

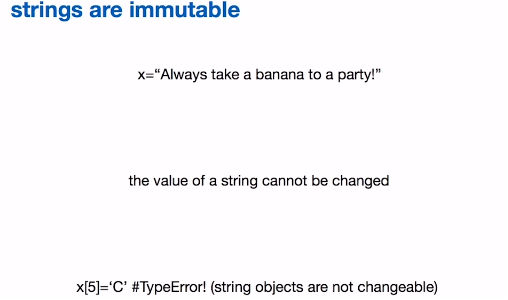
 



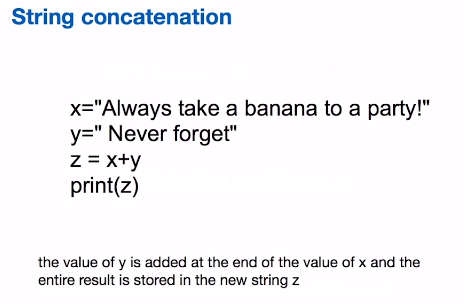


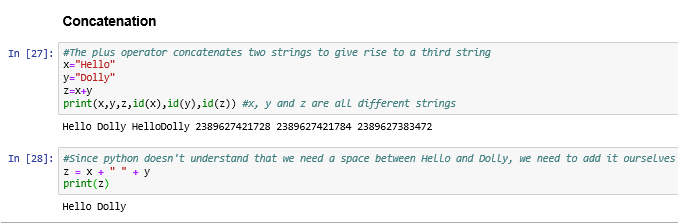




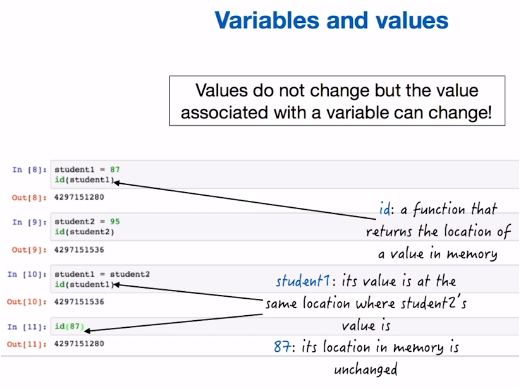




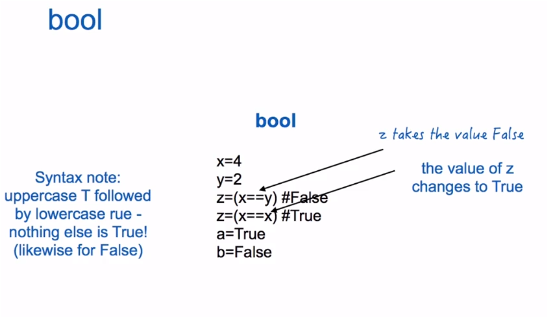


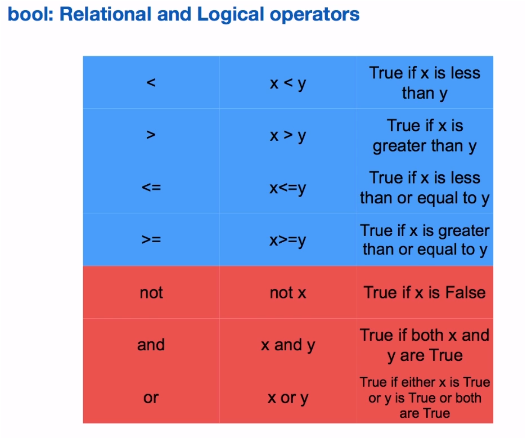


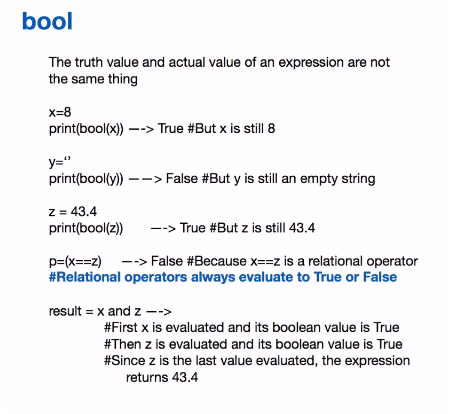
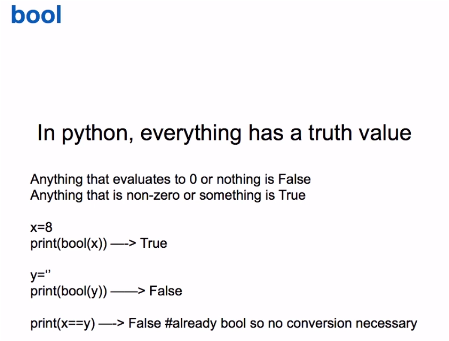
## 1.3 Variables and values



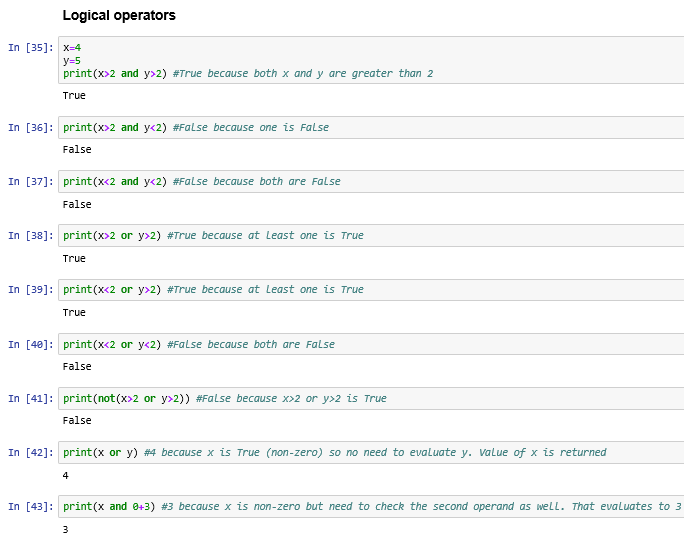
## 1.4 Boolean types in Python





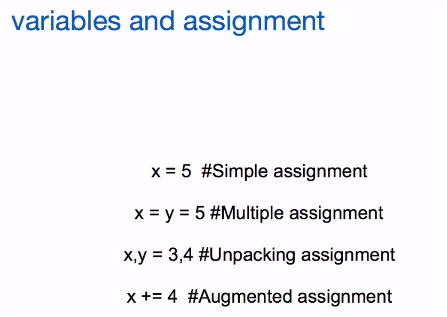
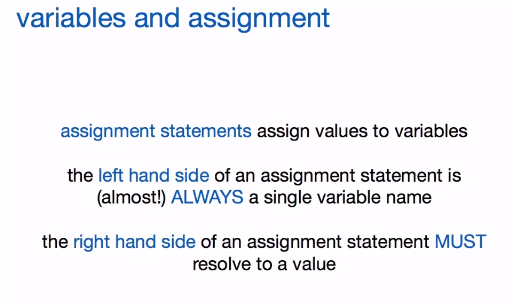


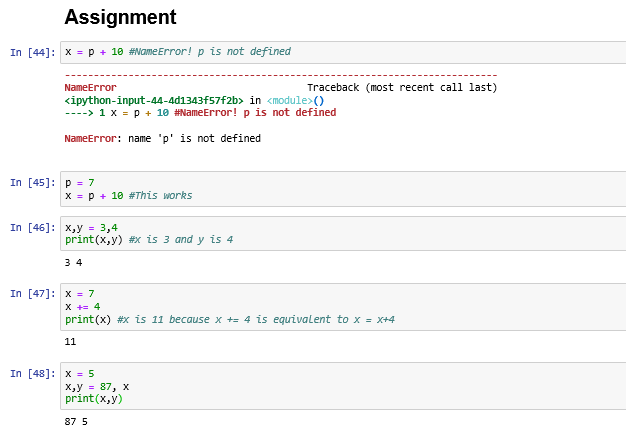




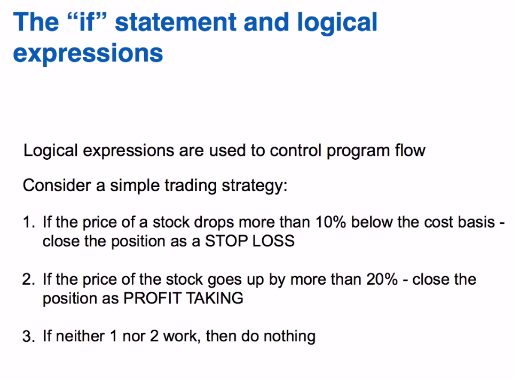
## 1.5 Assignment Operations

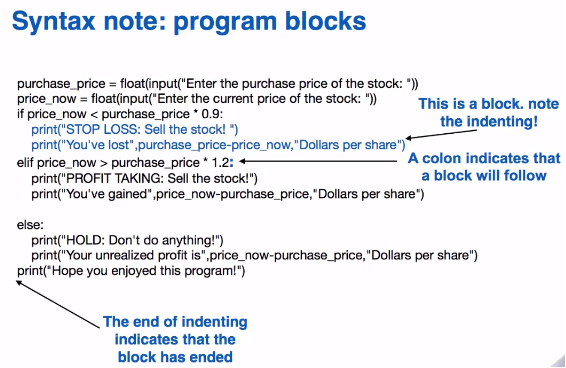


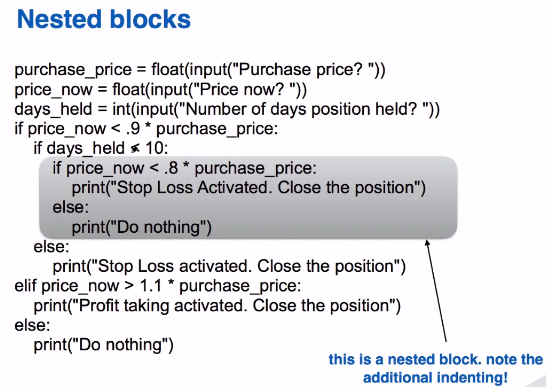


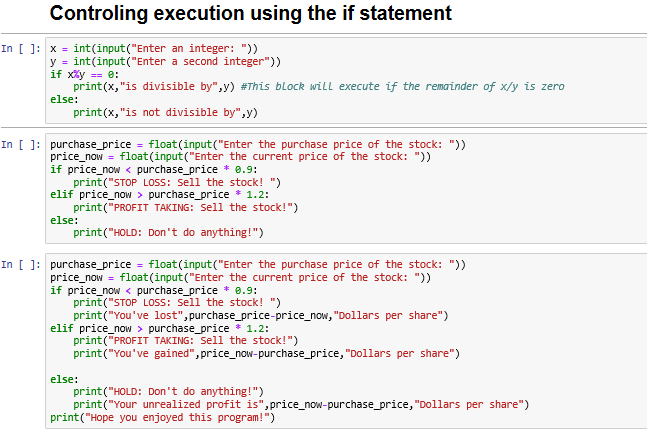


## 1.6 The if statement









**Ejemplo:**

x = int(input("Enter an integer: "))

y = int(input("Enter a second integer"))

if x%y == 0:

print(x,"is divisible by",y) #This block will execute if the remainder of x/y is zero

else:

print(x,"is not divisible by",y)

**Ejemplo:**

purchase\_price = float(input("Enter the purchase price of the stock: "))

price\_now = float(input("Enter the current price of the stock: "))

if price\_now < purchase\_price \* 0.9:

print("STOP LOSS: Sell the stock! ")

elif price\_now > purchase\_price \* 1.2:

print("PROFIT TAKING: Sell the stock!")

else:

print("HOLD: Don't do anything!")

**Ejemplo:**

purchase\_price = float(input("Enter the purchase price of the stock: "))

price\_now = float(input("Enter the current price of the stock: "))

if price\_now < purchase\_price \* 0.9:

print("STOP LOSS: Sell the stock! ")

print("You've lost",purchase\_price-price\_now,"Dollars per share")

elif price\_now > purchase\_price \* 1.2:

print("PROFIT TAKING: Sell the stock!")

print("You've gained",price\_now-purchase\_price,"Dollars per share")

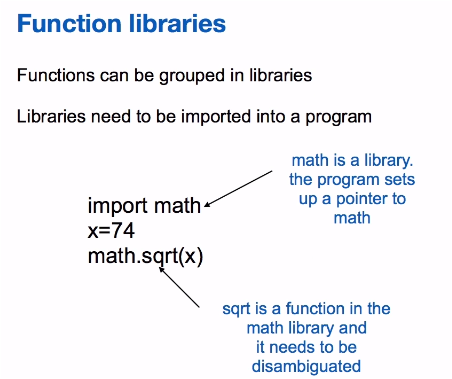
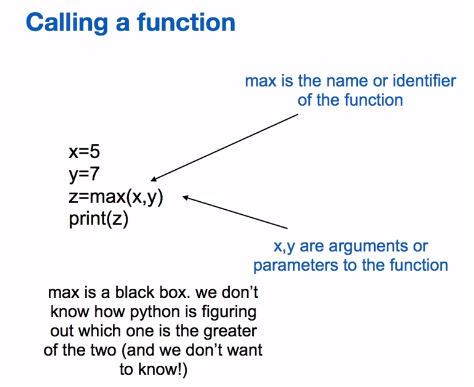
else:

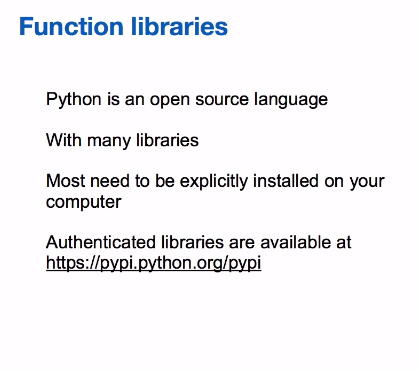
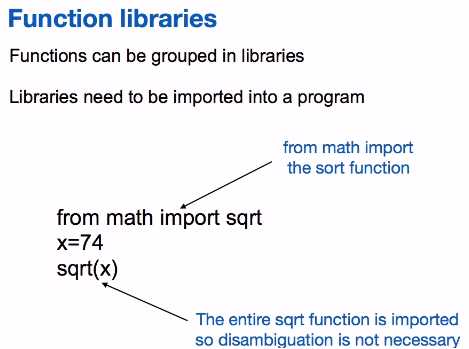
print("HOLD: Don't do anything!")

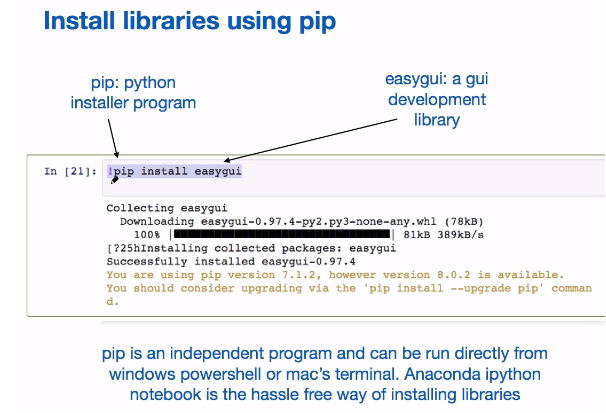
print("Your unrealized profit is",price\_now-purchase\_price,"Dollars per share")

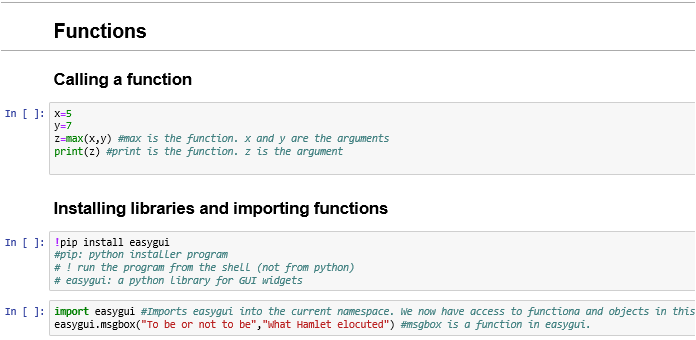
print("Hope you enjoyed this program!")

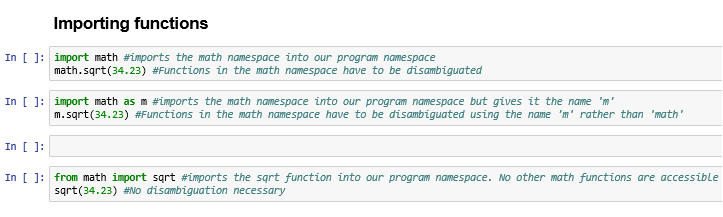
## 1.7 Functions (Part 1)

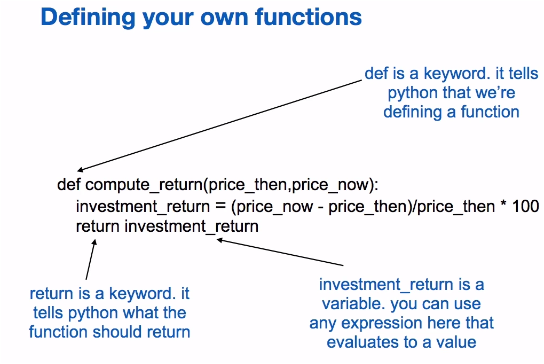












## 1.8 Functions (Part 2)

## 

## 

## 

## 

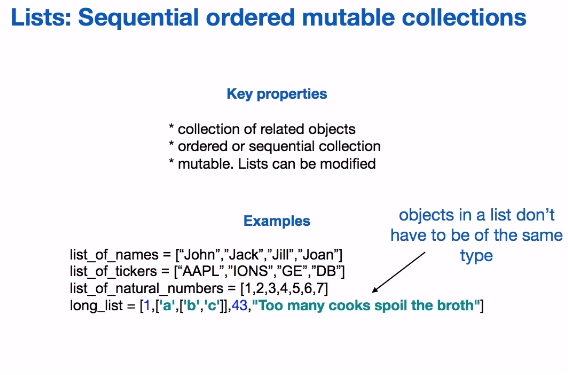
## 

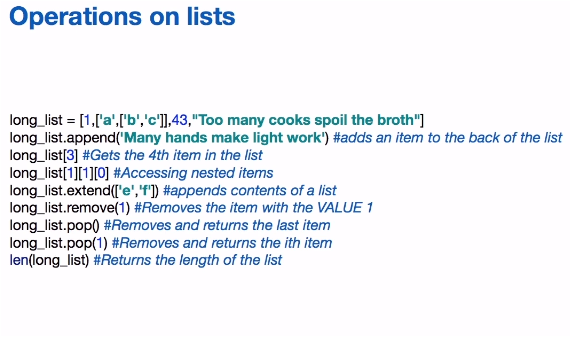
## 

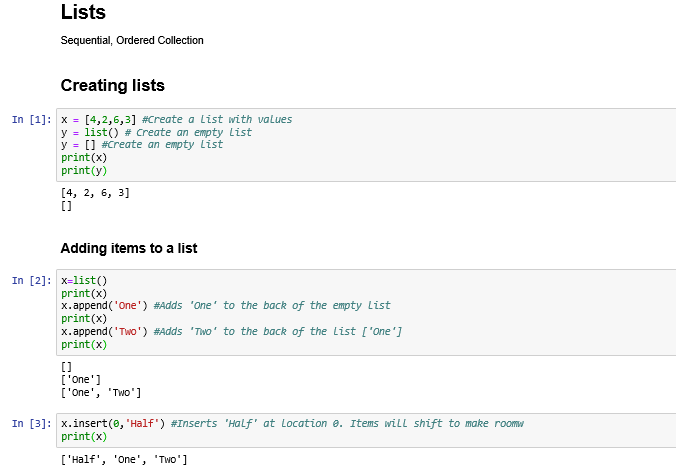
## 

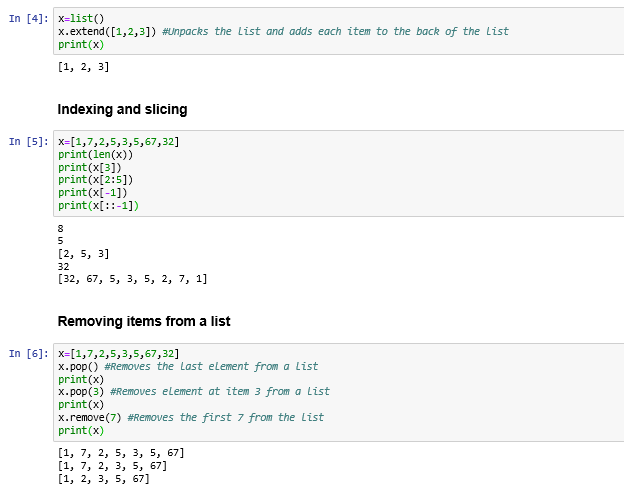
### Week2:

## 2.1 Lists

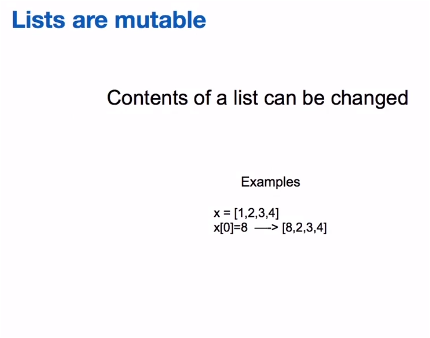
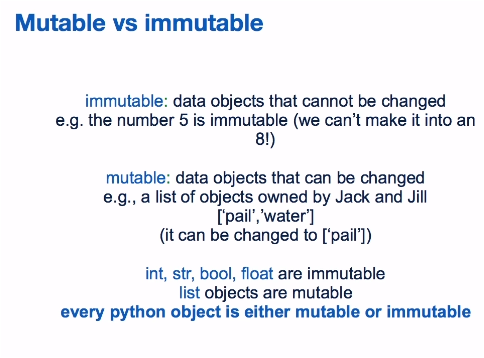


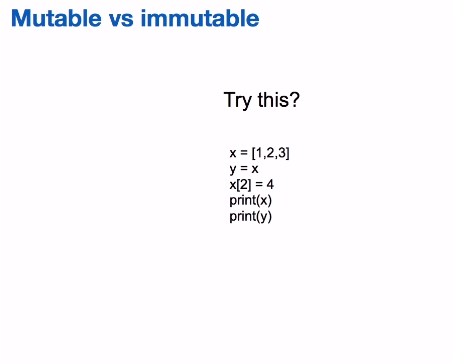
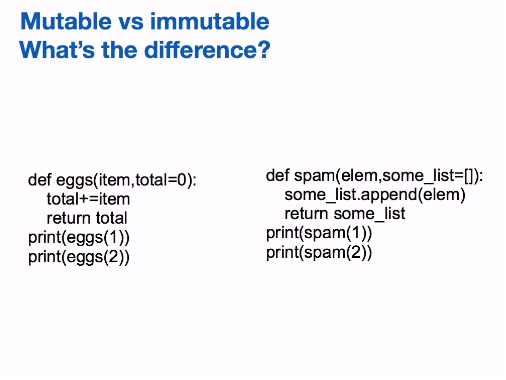




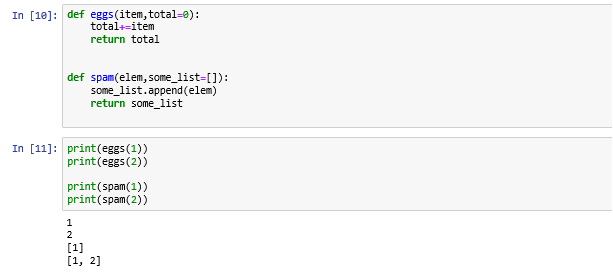


## 2.2 Mutability

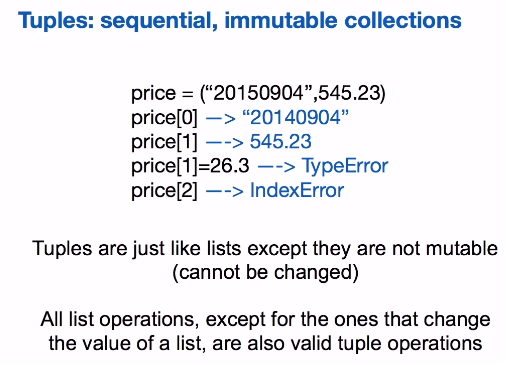
 

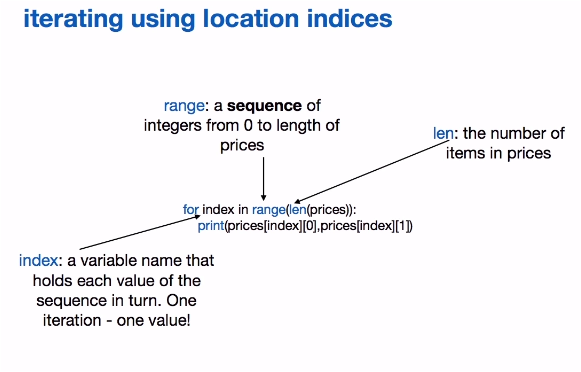
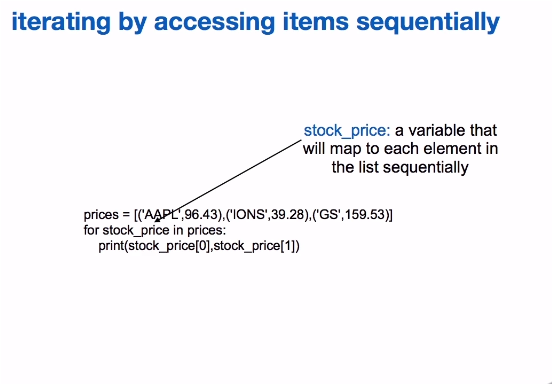
 

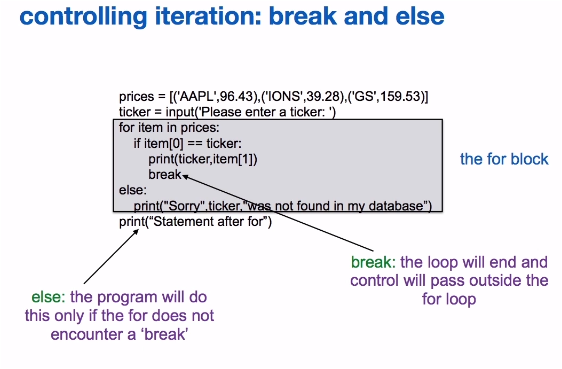




## 2.3 Iteration

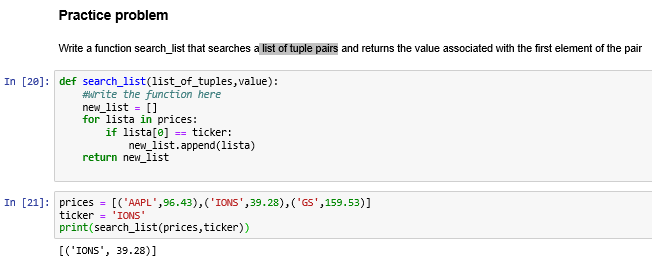


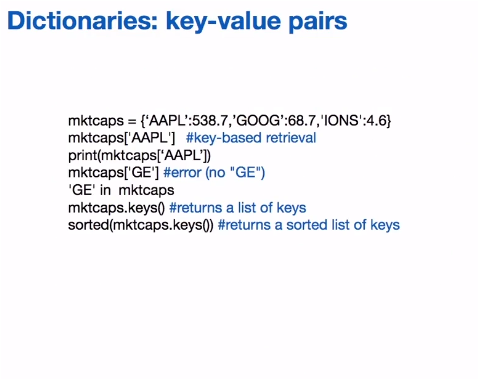
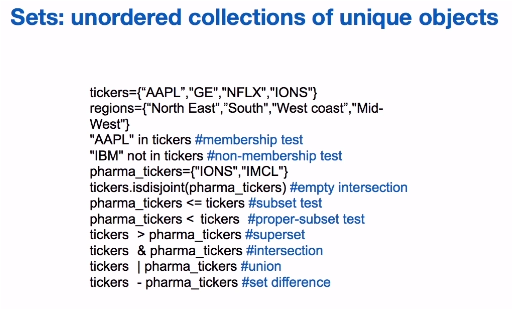


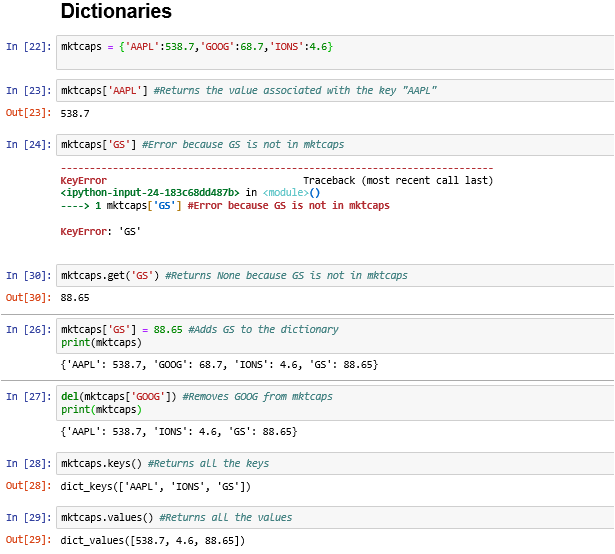


## 2.4 Example

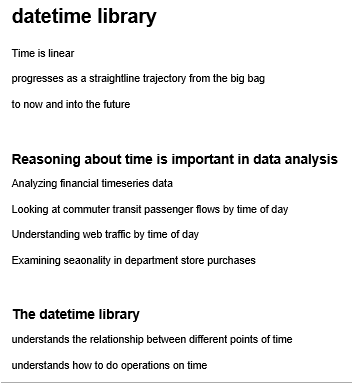


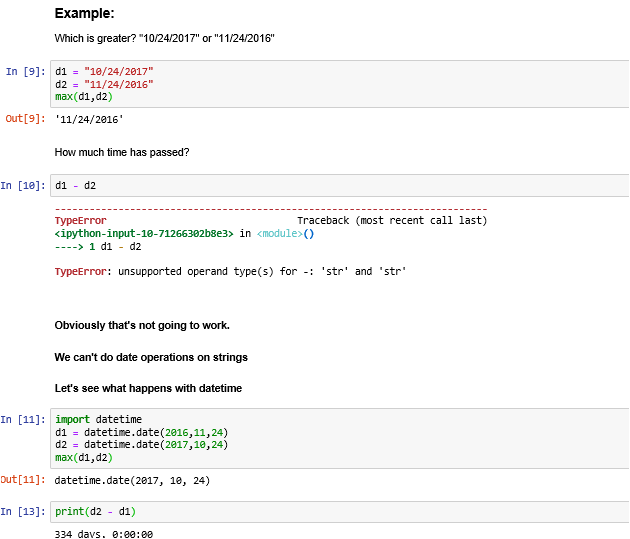
## 2.5 Dictionaries and sets





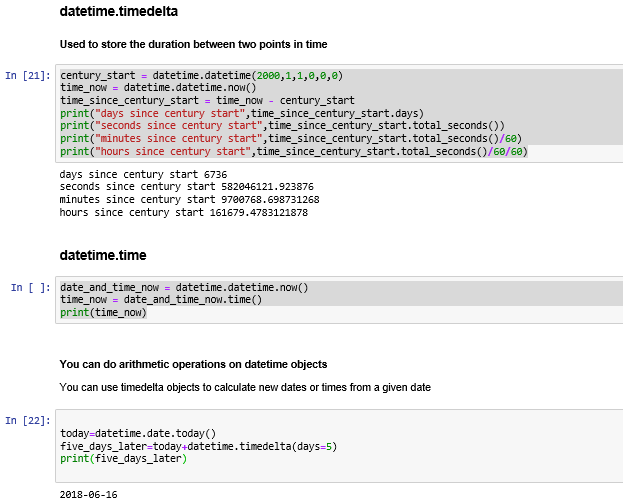
## 2.6 datetime library I



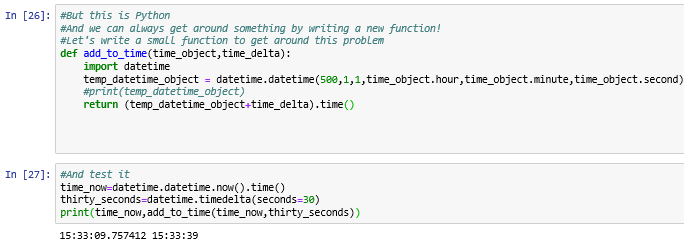


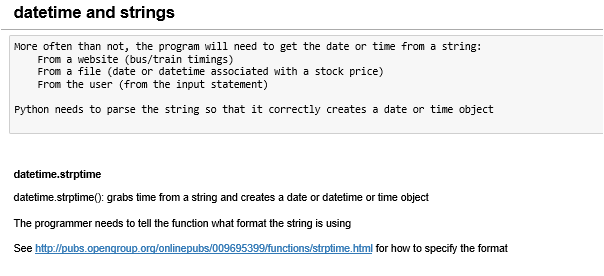


## 2.7 datetime library II











## 2.8 Bucketing time Part I

## 

## 

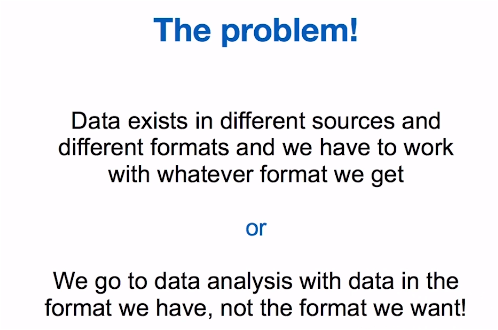
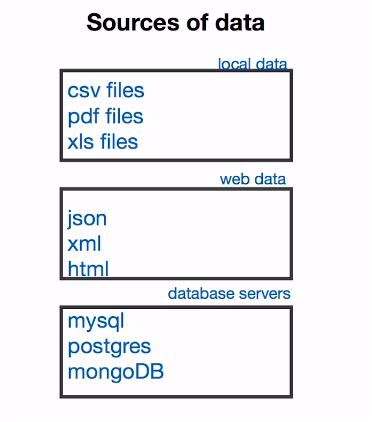
## 

## 

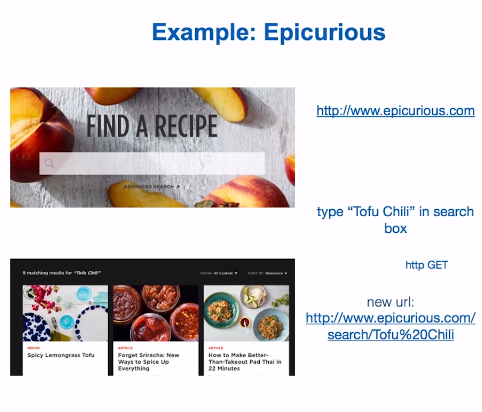
## 

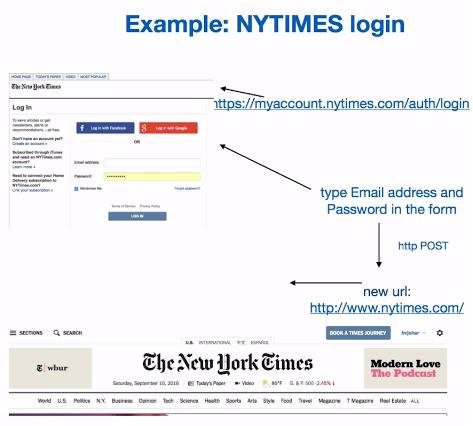
### Week3:

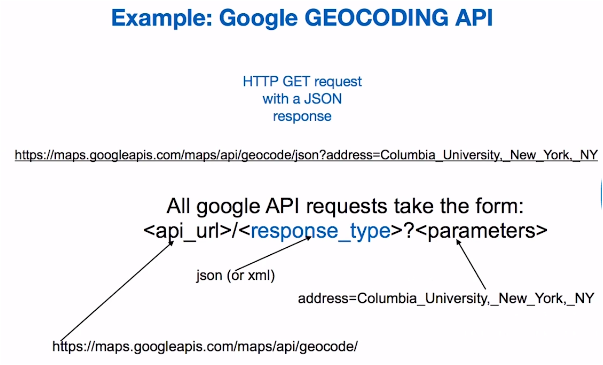
## 3.1 Getting data (Part 1)

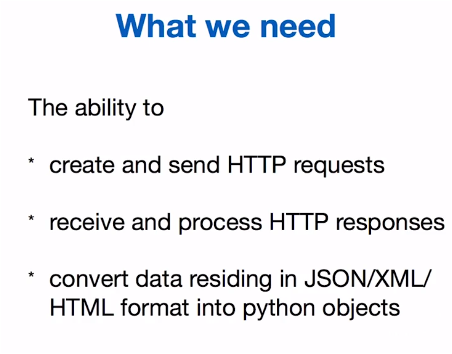
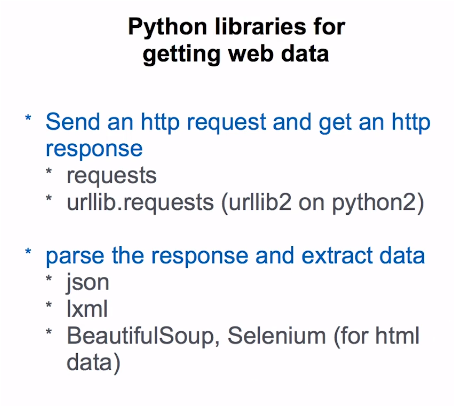
 



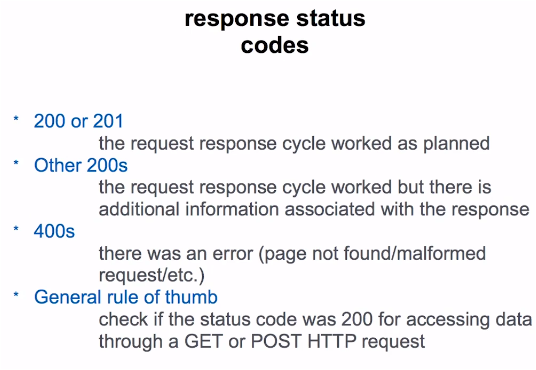


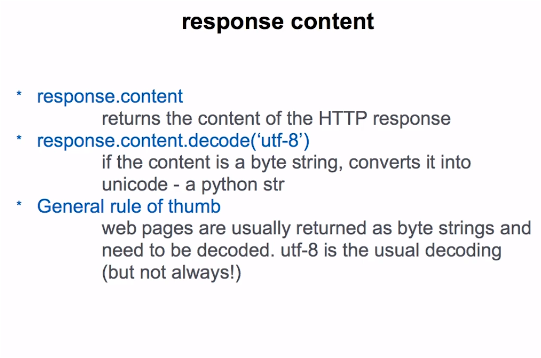


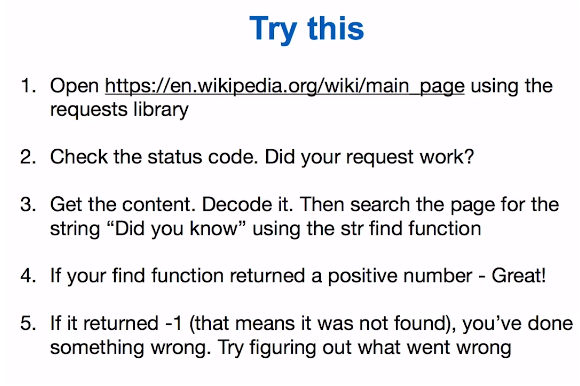
## 3.2 Getting data (Part 2)



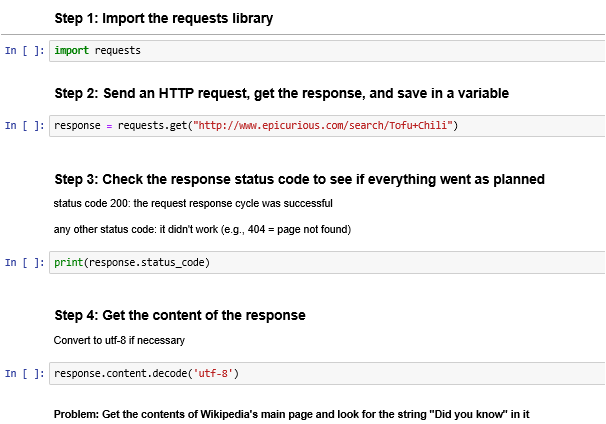
 





**IMPORTANTE: se debe instalar**

**C:\>pip install requests**



url = "https://en.wikipedia.org/wiki/main\_page"

#The rest of your code should go below this line

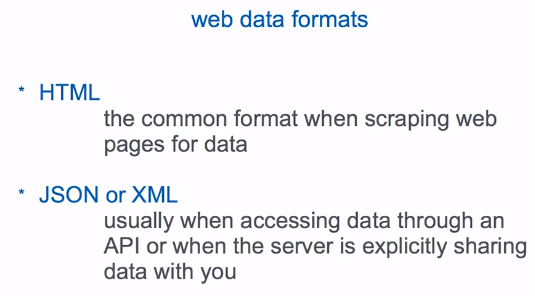
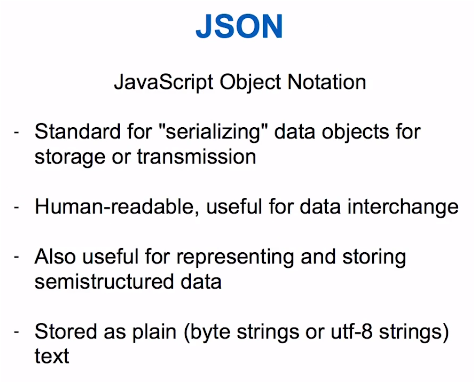
wiki\_page\_response = requests.get(url)

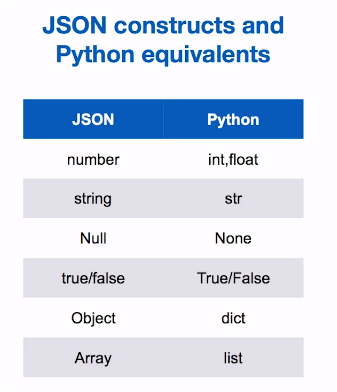
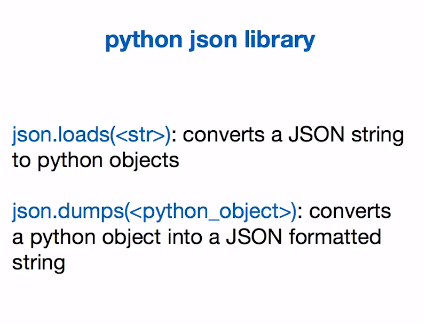
wiki\_text = wiki\_page\_response.content.decode('utf-8')

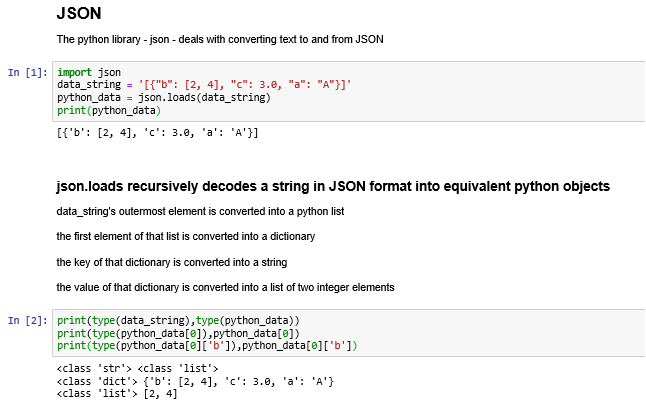
wiki\_text.find('Did you know')

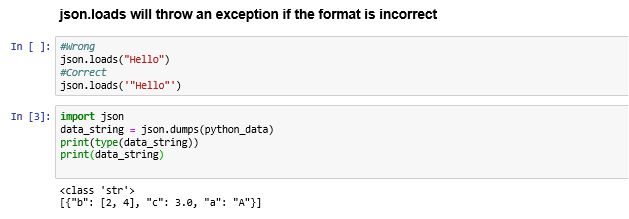


## 3.3 Web Data Formats

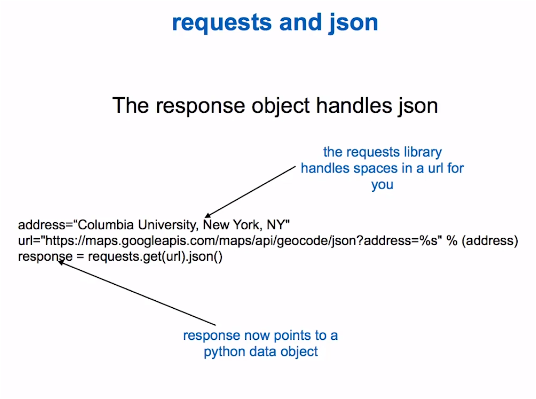
 

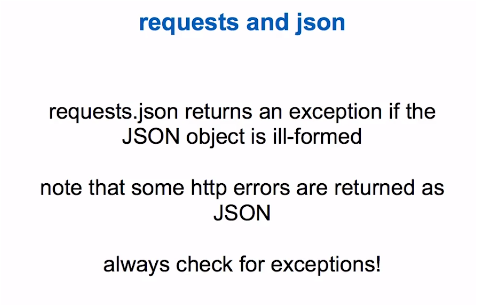
 

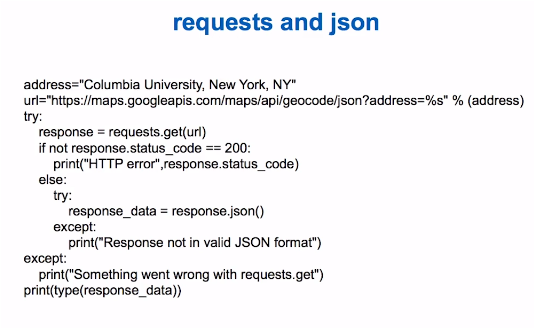




## 3.4 JSON, Google API (Part 1)









import requests

address="Columbia University, New York, NY"

url="https://maps.googleapis.com/maps/api/geocode/json?address=%s" % (address)

try:

response = requests.get(url)

if not response.status\_code == 200:

print("HTTP error",response.status\_code)

else:

try:

response\_data = response.json()

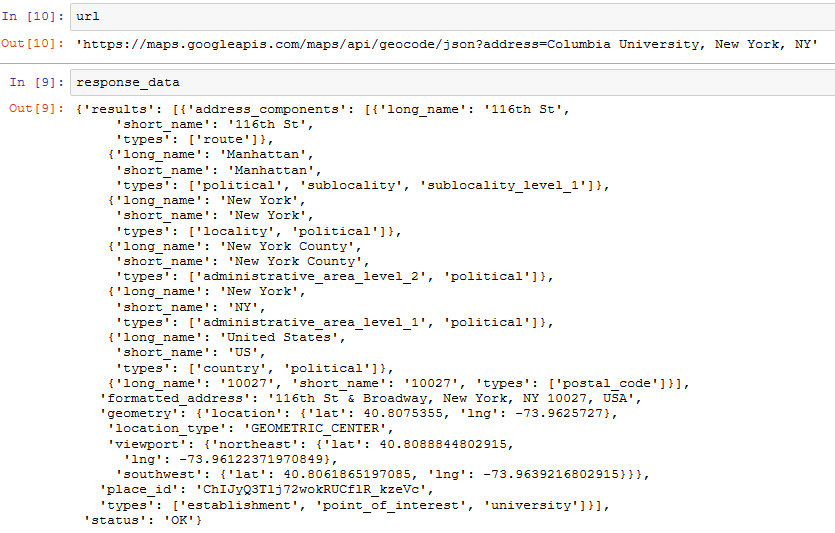
except:

print("Response not in valid JSON format")

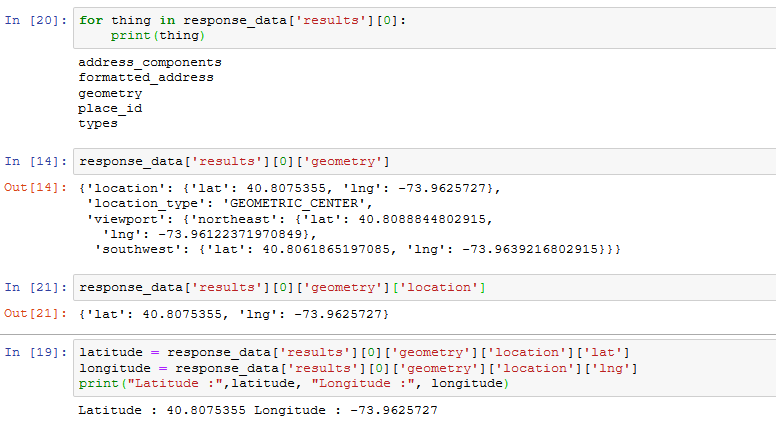
except:

print("Something went wrong with requests.get")

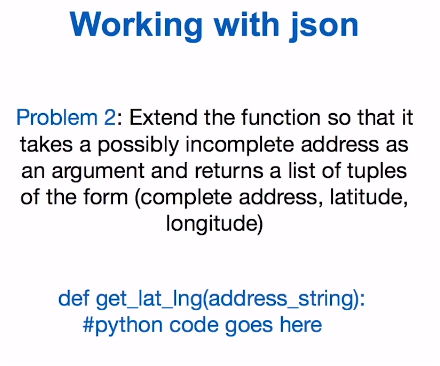
print(type(response\_data))







## 3.5 JSON, Google API (Part 2)



Resuelto:

**W3\_Practice\_1\_return\_latitude\_longitude\_tuple**

def get\_lat\_lng(address\_string):

#python code goes here

import json

import requests

url="https://maps.googleapis.com/maps/api/geocode/json?address=%s" % (address\_string)

try:

response = requests.get(url)

if not response.status\_code == 200:

print("HTTP error",response.status\_code)

else:

try:

response\_data = response.json()

except:

print("Response not in valid JSON format")

except:

print("Something went wrong with requests.get")

latitude = response\_data['results'][0]['geometry']['location']['lat']

longitude = response\_data['results'][0]['geometry']['location']['lng']

return latitude, longitude

**W3\_Practice\_2\_return\_complete\_address\_latitude\_longitude\_tuple**

def get\_lat\_lng(address\_string):

#python code goes here

import json

import requests

url="https://maps.googleapis.com/maps/api/geocode/json?address=%s" % (address\_string)

try:

response = requests.get(url)

if not response.status\_code == 200:

print("HTTP error",response.status\_code)

else:

try:

response\_data = response.json()

except:

print("Response not in valid JSON format")

except:

print("Something went wrong with requests.get")

latitude = response\_data['results'][0]['geometry']['location']['lat']

longitude = response\_data['results'][0]['geometry']['location']['lng']

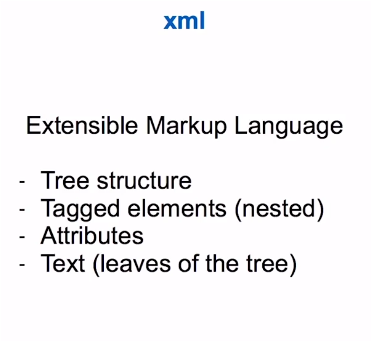
address = response\_data['results'][0]['formatted\_address']

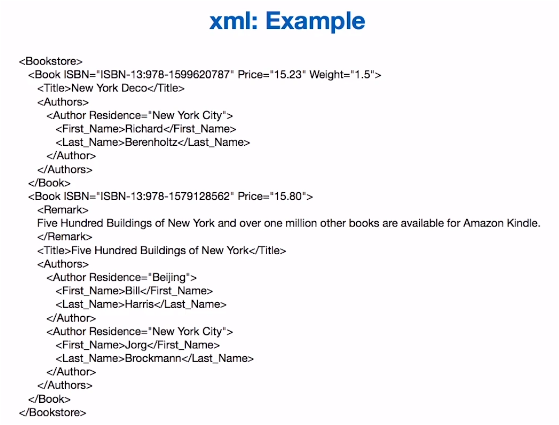
#print("Latitude :", latitude, "Longitude :",longitude)

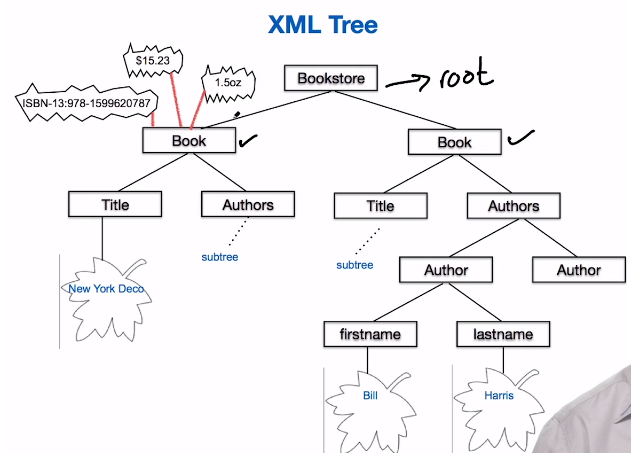
#print("Address :", address)

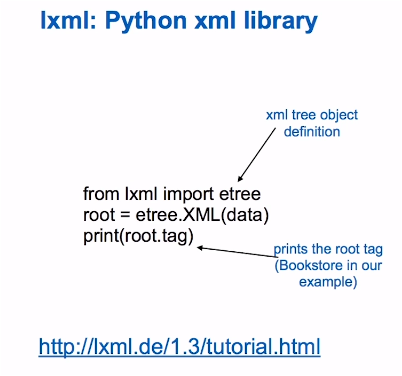
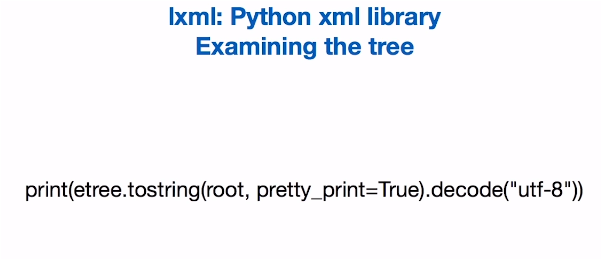
return address, latitude, longitude

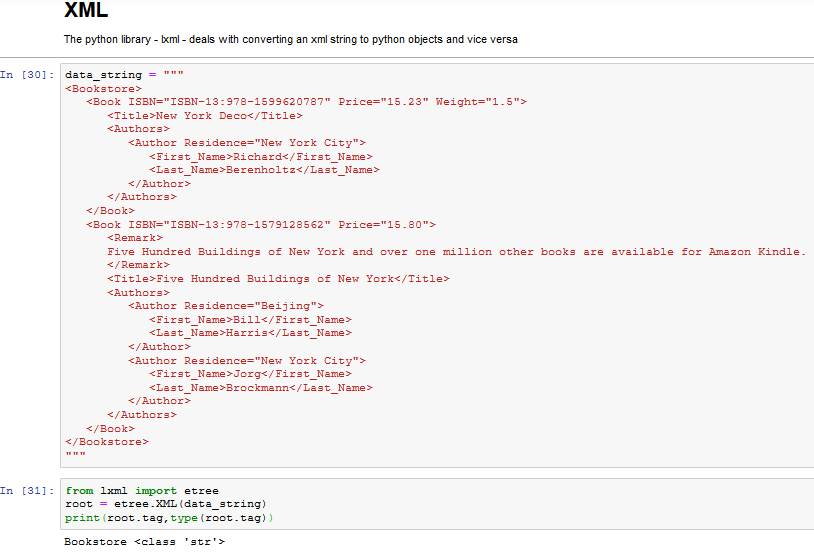
## 3.6 XML (Part 1)

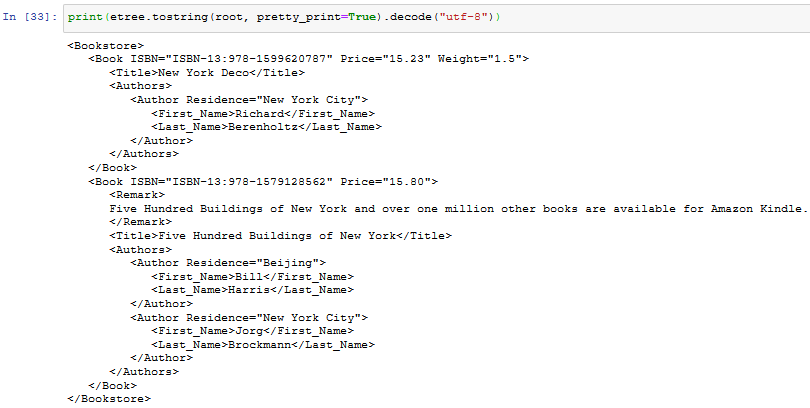










## 3.7 XML (Part 2)



