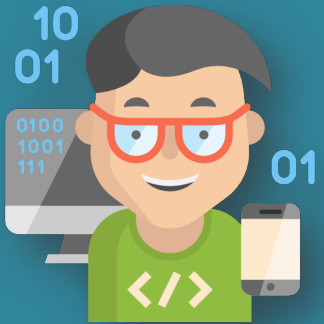


## Python Tutorial



# Agenda

1

Installing Python

2

Python Basics

3

Data Structures in Python

4

Python Functions

5

Object Oriented Programming

6

Numerical Computing with NumPy

7

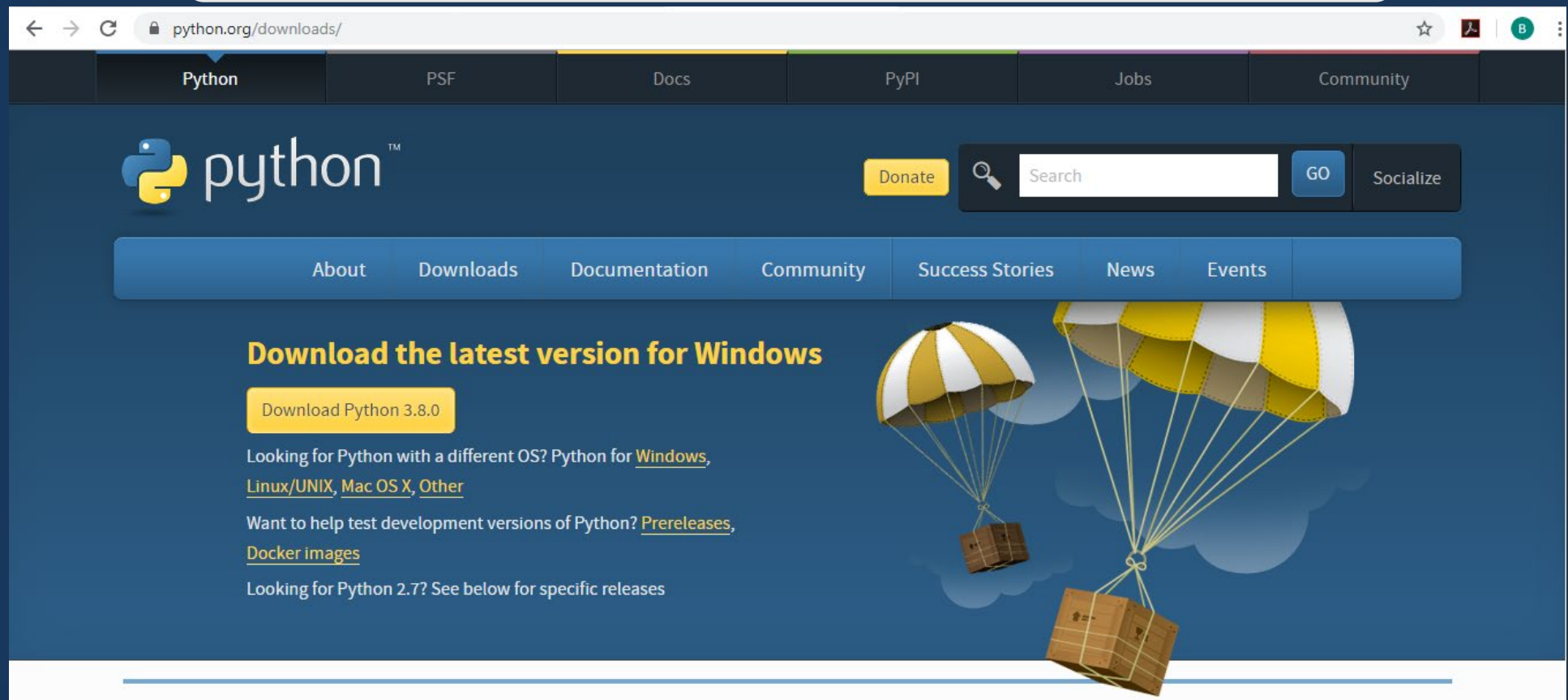
Data Manipulation with Pandas

8

Data Visualization with Matplotlib

# Installing Python

This is the site to install Python -> <https://www.python.org/downloads/>



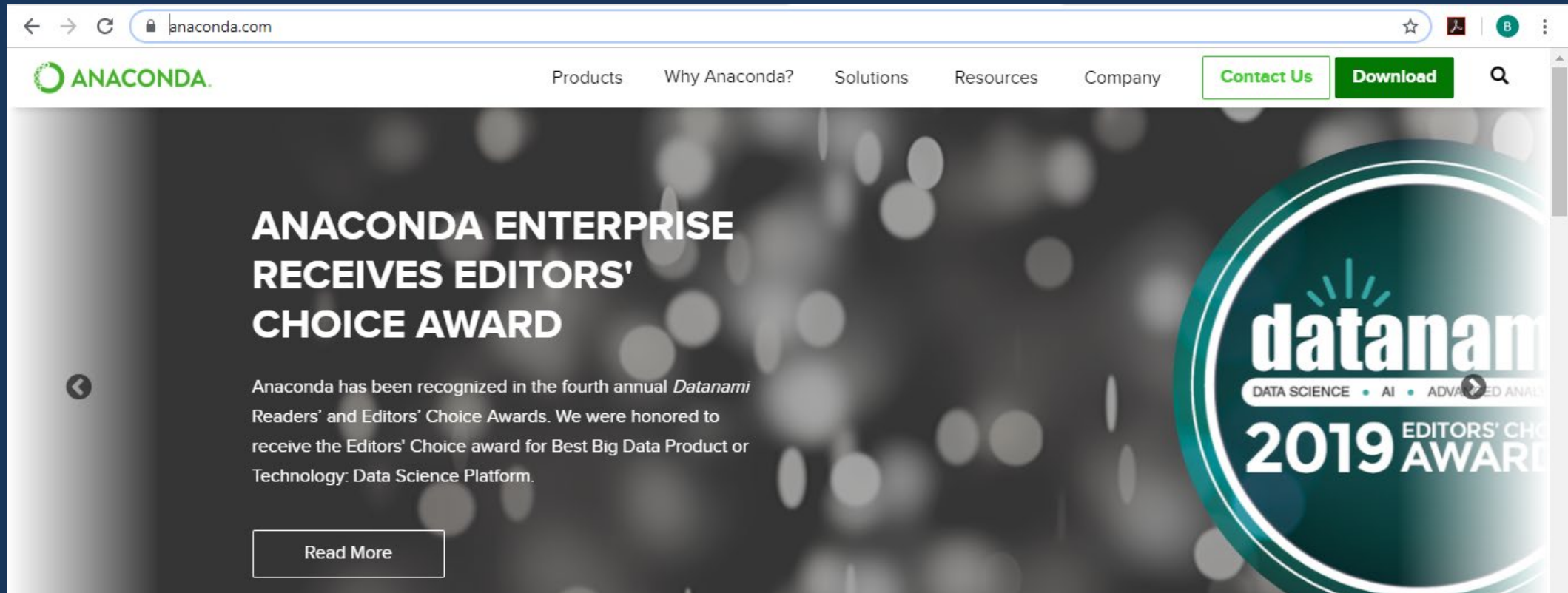
# Installing PyCharm

This is the site to install PyCharm -> <https://www.jetbrains.com/pycharm/>



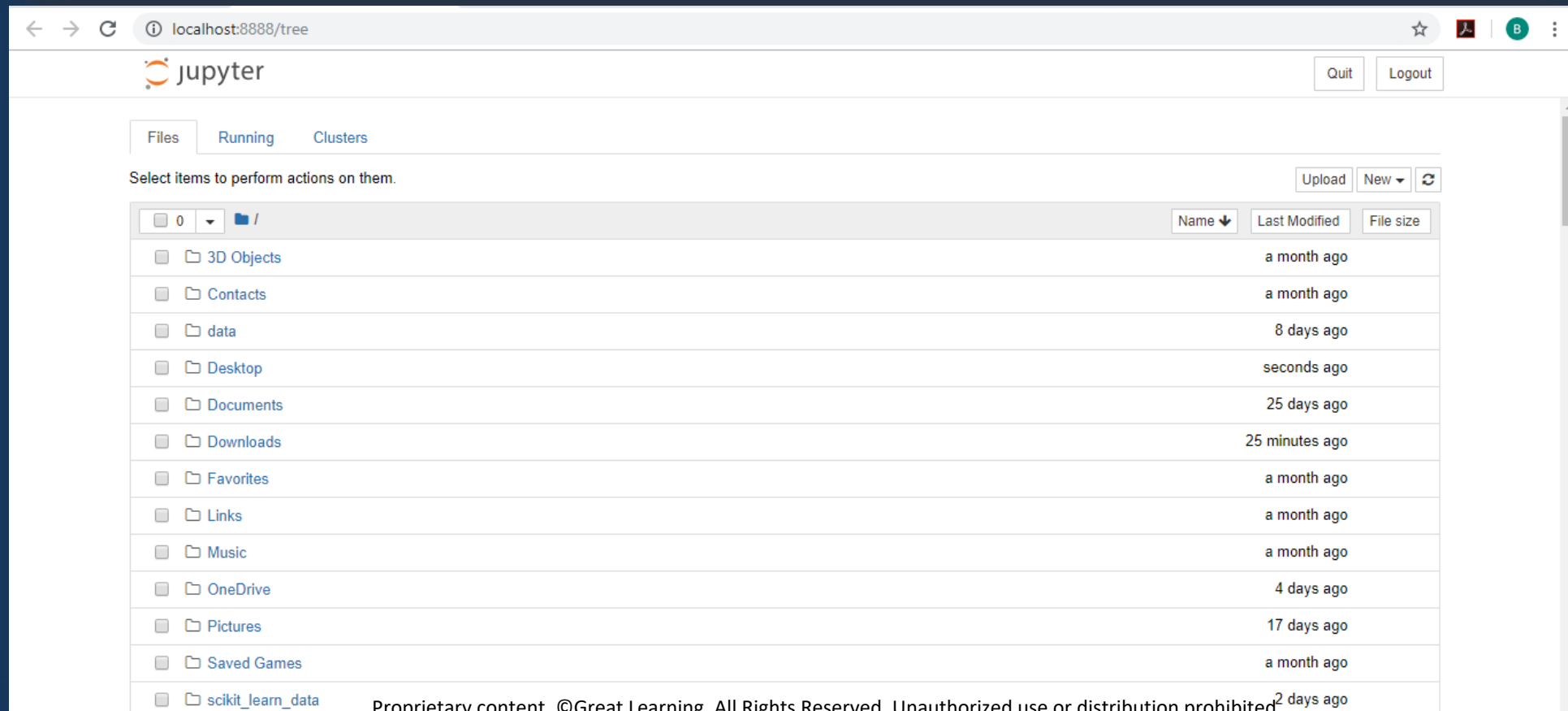
# Installing Anaconda

This is the site to install Anaconda -> <https://www.anaconda.com/>



# Intro to Jupyter Notebook

Jupyter Notebook is a browser-based interpreter that allows us to interactively work with Python



# Variables in Python



How do I  
store data?

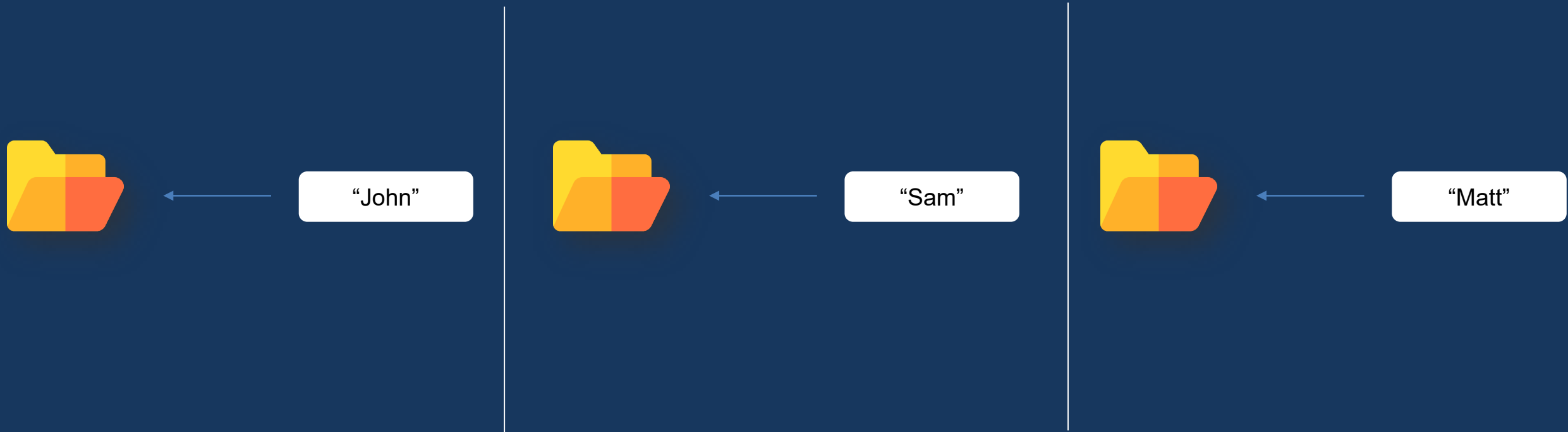
“John”

“Sam”

“Matt”

# Variables in Python

Data/Values can be stored in temporary storage spaces called variables





# DataTypes in Python

Every variable is associated with a data-type

10, 500

int

3.14, 15.97

float

TRUE, FALSE

Boolean

"Sam", "Matt"

String

# Operators in Python

Arithmetic Operators

Relational Operators



Logical Operators

# Python Strings

Strings are sequence of characters enclosed within single quotes(' '), double quotes(" ") or triple quotes(""" """)

'Hello World'

"This is Sparta"

""" I am going to  
France tomorrow"""

# Data-Structures in Python



Tuple

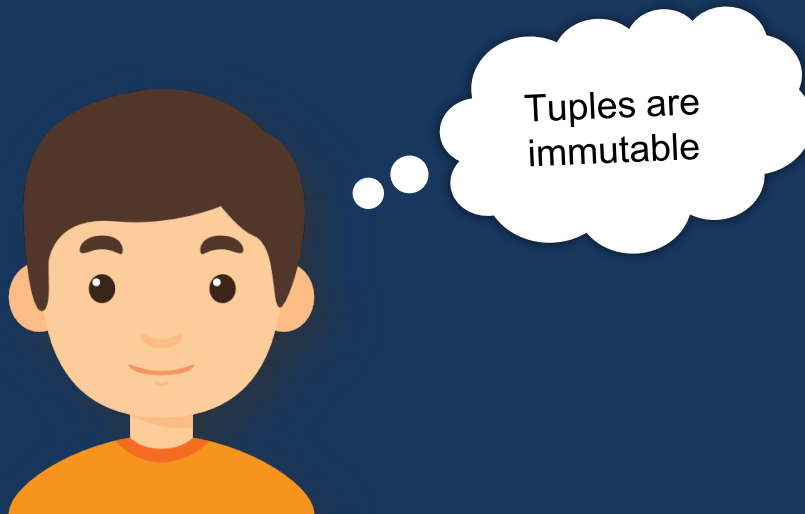
List

Dictionary

Set

# Tuple in Python

Tuple is an ordered collection of elements enclosed within ()



```
tup1=(1,'a',True)
```

# List in Python

List is an ordered collection of elements enclosed within []



```
l1=[1,'a',True]
```

# Dictionary in Python

Dictionary is an unordered collection of key-value pairs enclosed with {}



```
Fruit={"Apple":10,"Orange":20}
```

# Set in Python

Set is an unordered and unindexed collection of elements enclosed with {}



Duplicates  
are not  
allowed in  
Set

```
s1={1,"a",True}
```



# If Statement

If  
It's raining:  
Sit inside



else  
Go out and Play Football



# Looping Statements

Looping statements are used to repeat a task multiple times

while

for



Keep filling this  
bucket with a  
mug of water  
**while** it is not full



# For Loop

For Loop is used to iterate over a sequence(tuple, list, dictionary..)

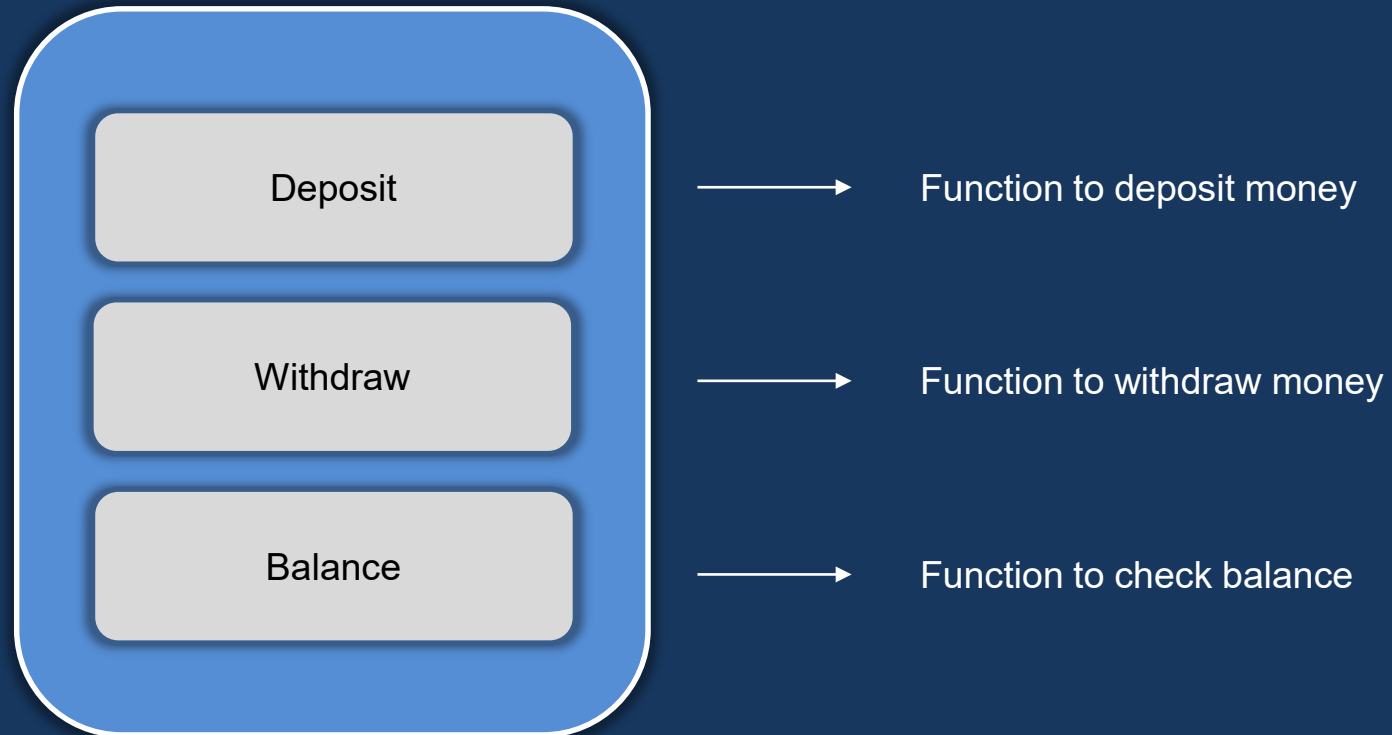


This is the  
syntax of for  
loop

```
for val in sequence:  
    Body of for
```

# Python Functions

Function is a block of code which performs a specific task



# Python Object Oriented Programming



# Classes

Class is a template/blue-print for real-world entities



Properties

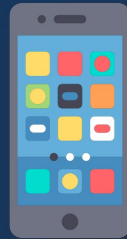
- Color
- Cost
- Battery Life

Behavior

- Make Calls
- Watch Videos
- Play Games

# Objects

Objects are specific instances of a class



Apple



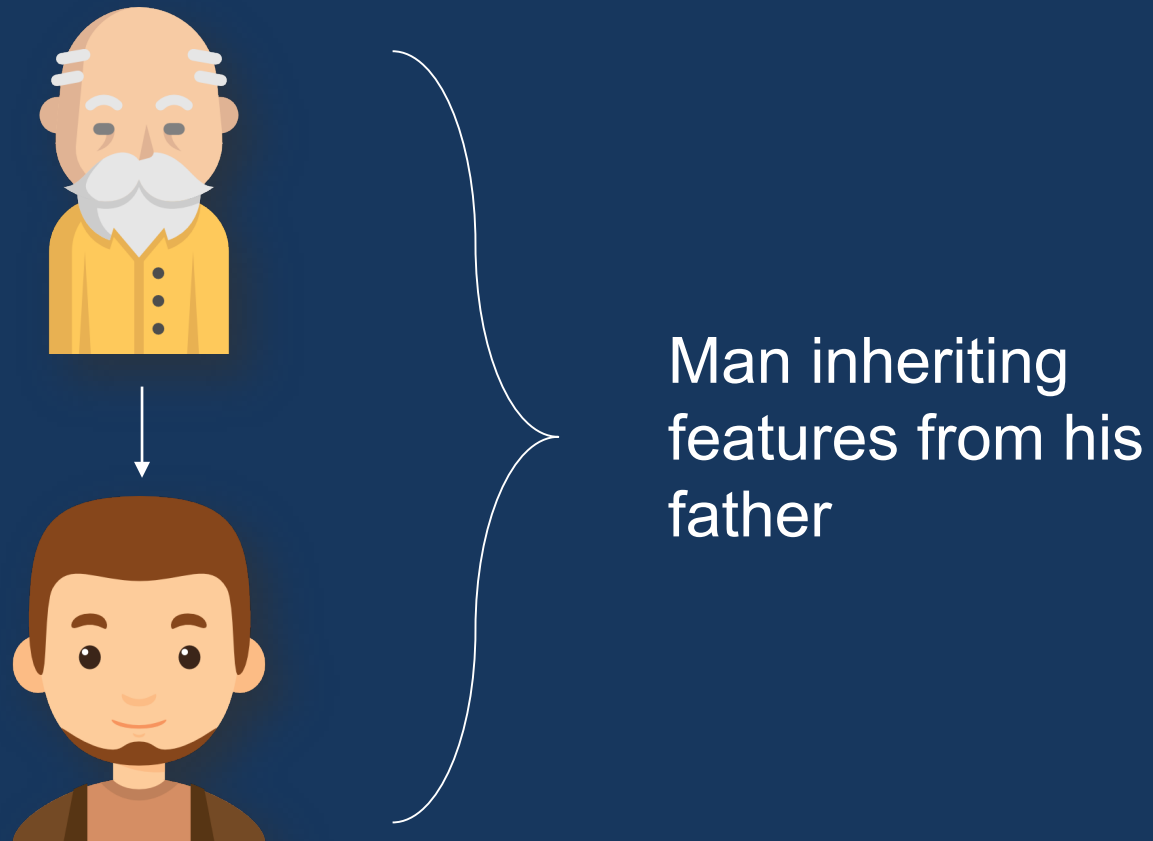
Motorola



Samsung

# Inheritance in Python

With inheritance one class can derive the properties of another class



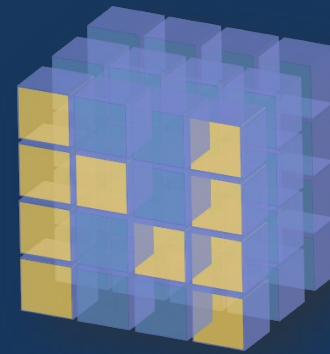


# Python NumPy

NumPy stands for Numerical python and is the core library for numeric and scientific computing



It consists of multi-dimensional array objects and a collection of routines for processing those arrays



# NumPy

# Creating NumPy Array

## Single-dimensional Array

```
In [3]: import numpy as np  
  
        n1=np.array([10,20,30,40])  
        n1  
  
Out[3]: array([10, 20, 30, 40])
```

## Multi-dimensional Array

```
In [6]: import numpy as np  
  
        n2=np.array([[10,20,30,40],[40,30,20,10]])  
        n2  
  
Out[6]: array([[10, 20, 30, 40],  
               [40, 30, 20, 10]])
```

# Initializing NumPy Array

Initializing NumPy array with zeros

```
In [30]: import numpy as np  
n1=np.zeros((1,2))  
n1
```

```
Out[30]: array([[0., 0.]])
```

```
In [31]: import numpy as np  
n1=np.zeros((5,5))  
n1
```

```
Out[31]: array([[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.]])
```

# Initializing NumPy Array

Initializing NumPy array with same number

```
In [38]: import numpy as np  
n1=np.full((2,2),10)  
n1
```

```
Out[38]: array([[10, 10],  
               [10, 10]])
```

# Initializing NumPy Array

## Initializing NumPy array within a range

```
In [34]: import numpy as np  
         n1=np.arange(10,20)  
         n1
```

```
Out[34]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
In [35]: import numpy as np  
         n1=np.arange(10,50,5)  
         n1
```

```
Out[35]: array([10, 15, 20, 25, 30, 35, 40, 45])
```

# Initializing NumPy Array

Initializing NumPy array with random numbers

```
In [46]: import numpy as np  
         n1=np.random.randint(1,100,5)  
         n1  
  
Out[46]: array([95, 88, 26, 22, 76])
```

# NumPy-Shape

Checking the shape of NumPy arrays

```
In [4]: import numpy as np  
        n1=np.array([[1,2,3],[4,5,6]])  
        n1.shape
```

```
Out[4]: (2, 3)
```

```
In [5]: n1.shape = (3,2)  
        n1.shape
```

```
Out[5]: (3, 2)
```

# NumPy Array Mathematics

## Addition of NumPy Arrays

```
In [13]: import numpy as np  
n1=np.array([10,20])  
n2=np.array([30,40])  
  
np.sum([n1,n2])
```

```
Out[13]: 100
```

```
In [14]: np.sum([n1,n2],axis=0)
```

```
Out[14]: array([40, 60])
```

```
In [15]: np.sum([n1,n2],axis=1)
```

```
Out[15]: array([30, 70])
```



# Joining NumPy Arrays

vstack()

```
In [32]: import numpy as np
n1=np.array([10,20,30])
n2=np.array([40,50,60])

np.vstack((n1,n2))

Out[32]: array([[10, 20, 30],
               [40, 50, 60]])
```

hstack()

```
In [33]: import numpy as np
n1=np.array([10,20,30])
n2=np.array([40,50,60])

np.hstack((n1,n2))

Out[33]: array([10, 20, 30, 40, 50, 60])
```

column\_stack()

```
In [34]: import numpy as np
n1=np.array([10,20,30])
n2=np.array([40,50,60])

np.column_stack((n1,n2))

Out[34]: array([[10, 40],
               [20, 50],
               [30, 60]])
```

# Python Pandas

Pandas stands for Panel Data and is the core library for data manipulation and data analysis



It consists of single and multi-dimensional data-structures for data-manipulation



# Pandas Data-Structures

Single-dimensional



Series Object

Multi-dimensional



Data-frame

# Pandas Series Object

Series Object is one-dimensional labeled array

```
In [2]: import pandas as pd  
s1=pd.Series([1,2,3,4,5])  
s1
```

```
Out[2]: 0    1  
        1    2  
        2    3  
        3    4  
        4    5  
        dtype: int64
```

```
In [4]: type(s1)
```

```
Out[4]: pandas.core.series.Series
```

# Changing Index

```
In [2]: import pandas as pd  
s1=pd.Series([1,2,3,4,5])  
s1
```

```
Out[2]: 0    1  
        1    2  
        2    3  
        3    4  
        4    5  
        dtype: int64
```



```
In [5]: import pandas as pd  
s1=pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])  
s1
```

```
Out[5]: a    1  
        b    2  
        c    3  
        d    4  
        e    5  
        dtype: int64
```

## Series Object from Dictionary



You can also create  
a series object from  
a dictionary!!

```
In [8]: import pandas as pd  
pd.Series({'a':10,'b':20,'c':30})
```

```
Out[8]: a    10  
       b    20  
       c    30  
       dtype: int64
```

# Pandas Dataframe

Dataframe is a 2-dimensional labelled data-structure



A data-frame  
comprises of rows  
and columns

Out[9]:

	Name	Marks
0	Bob	76
1	Sam	25
2	Anne	92

# Creating a Dataframe



This is how you can  
create a data.frame

```
In [9]: import pandas as pd
```

```
pd.DataFrame({"Name": ['Bob', 'Sam', 'Anne'], "Marks": [76, 25, 92]})
```

```
Out[9]:
```

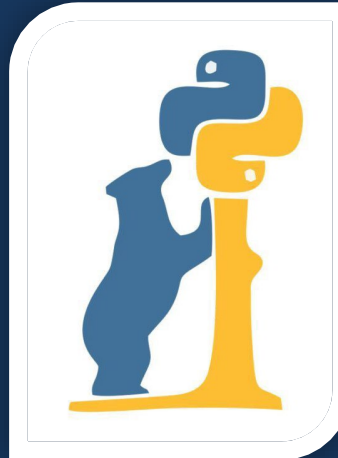
	Name	Marks
0	Bob	76
1	Sam	25
2	Anne	92



# Dataframe In-Built Functions

head()

shape()



describe()

tail()

# .iloc[]

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

`iris.iloc[0:3,0:2]`

	Sepal.Length	Sepal.Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2

.loc[]

```
iris.loc[0:3, ("Sepal.Length", "Petal.Length")]
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



	Sepal.Length	Petal.Length
0	5.1	1.4
1	4.9	1.4
2	4.7	1.3
3	4.6	1.5

# Python Matplotlib

Matplotlib is a python library used for data visualization



You can create bar-plots, scatter-plots, histograms and a lot more with matplotlib

**matplotlib**

# Thank You