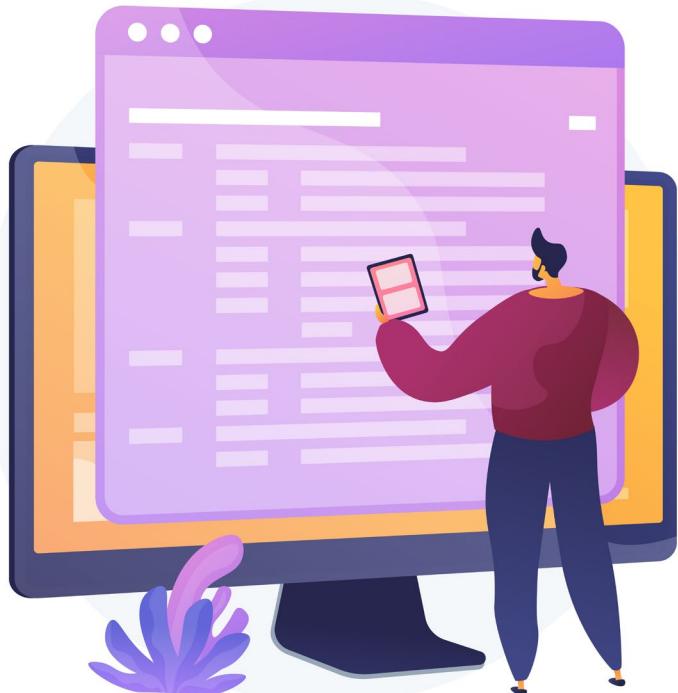


Django Tutorial



Agenda

Intro to Python

Web Frameworks

Python Fundamentals

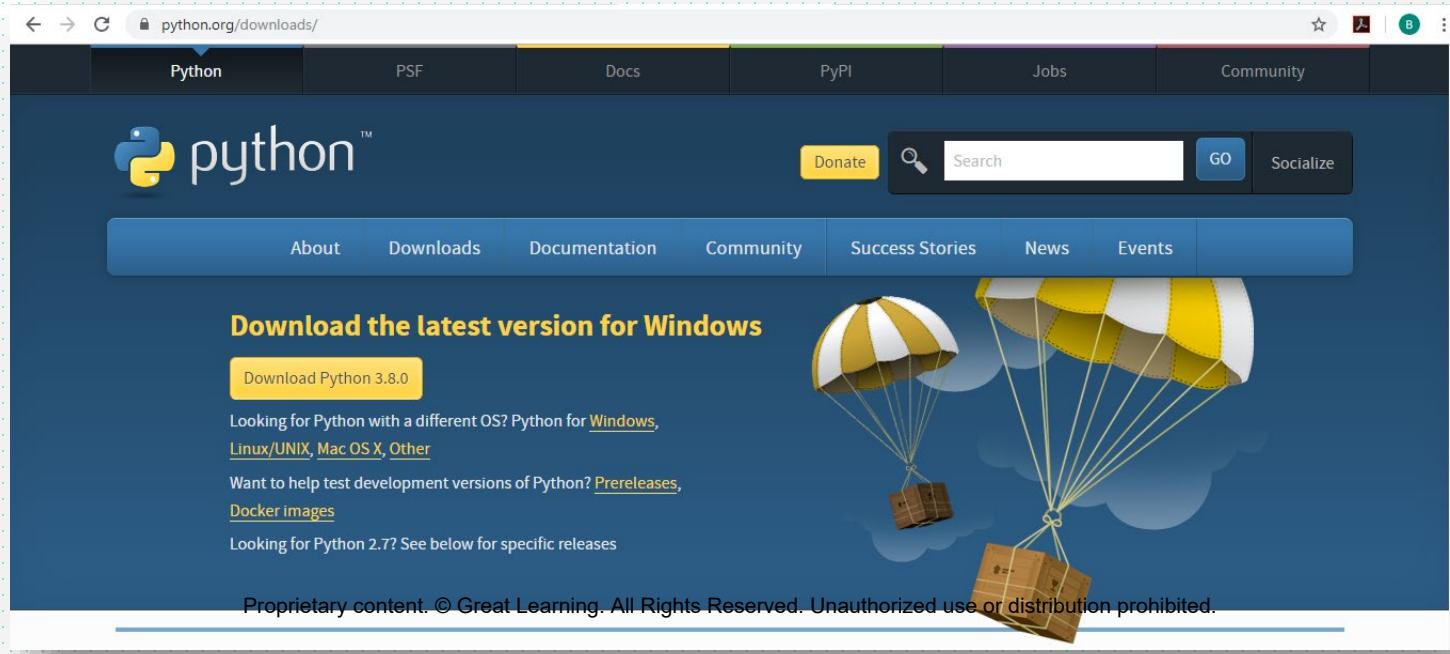
Intro to Django

Visualization with Python

Django App

Installing Python

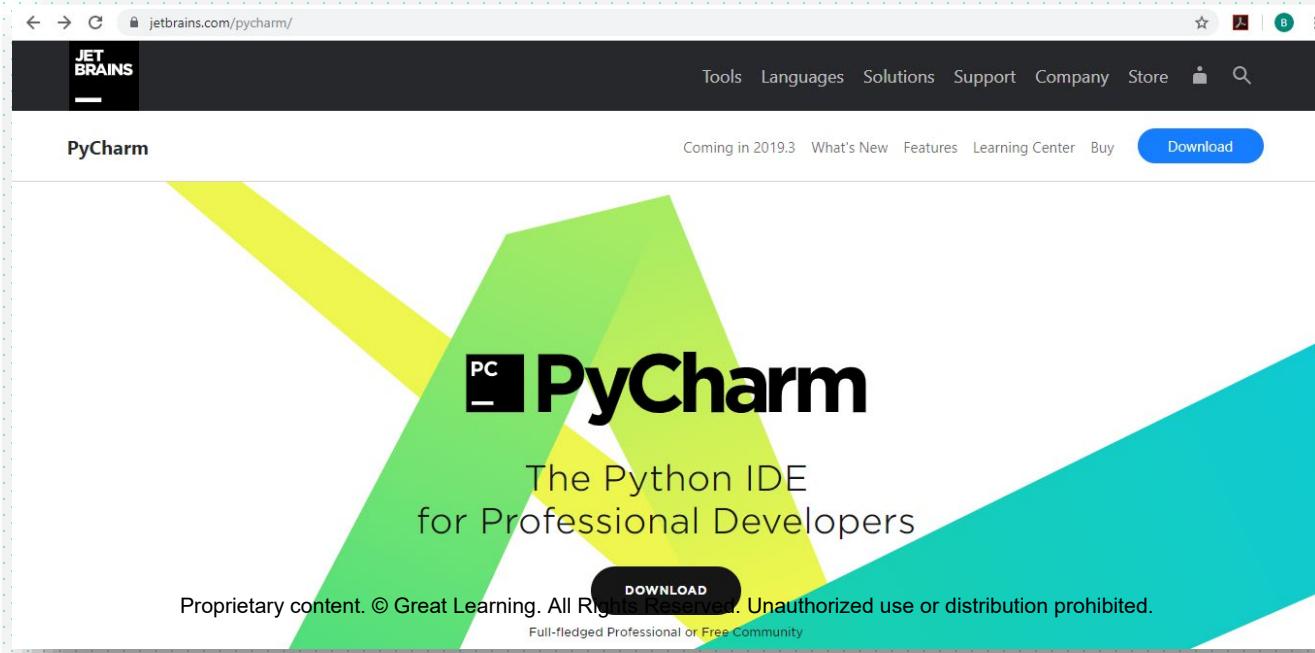
This is the site to install Python -> <https://www.python.org/downloads/>



The screenshot shows the Python Downloads page at [python.org/downloads/](https://www.python.org/downloads/). The page has a dark blue header with the Python logo and the word "python" in white. A navigation bar below the header includes links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a large yellow button labeled "Download Python 3.8.0". Below this button, text provides links for Python on Windows, Linux/UNIX, Mac OS X, and Other platforms. Further down, there are links for Prereleases and Docker images. A note for Python 2.7 is also present. To the right of the text, there is a graphic of two brown cardboard boxes descending from the sky, each attached to a yellow and white striped parachute. At the bottom of the page, a footer bar contains the text "Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited." and the Great Learning logo.

Installing PyCharm

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



Variables in Python



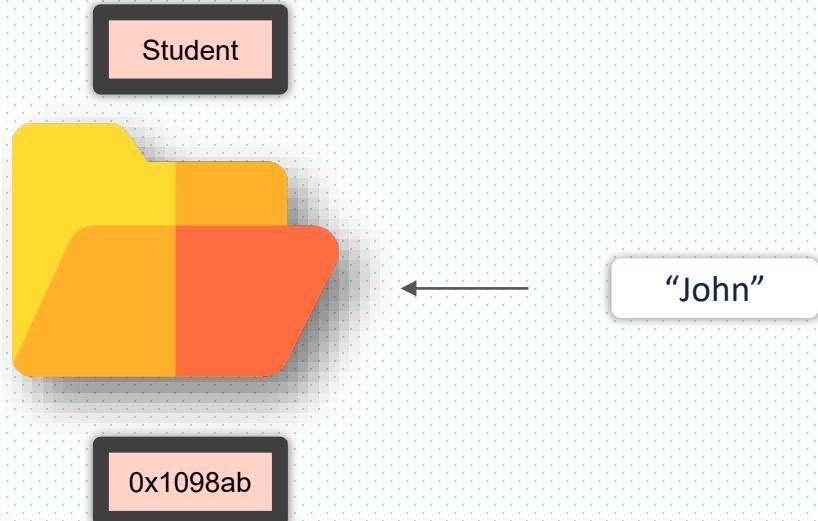
“John”

“Sam”

“Matt”

Variables in Python

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



Variables in Python

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



“John”



“Sam”



“Matt”

Data Types in Python

Every variable is associated with a data-type

10, 500

3.14, 15.97

TRUE, FALSE

“Sam”, “Matt”

int

float

Boolean

String

Operators in Python

Arithmetic Operators

Relational Operators

Logical Operators



Python Tokens

Smallest meaningful component in a program



Keywords

Identifiers

Literals

Operators

Python Keywords

Keywords are special reserved words

False	class	Finally	Is	Return
None	continue	For	Lambda	Try
True	def	From	Nonlocal	While
and	del	Global	Not	With
as	elif	If	Or	Yield

Python Identifiers

Identifiers are names used for variables, functions or objects

Rules

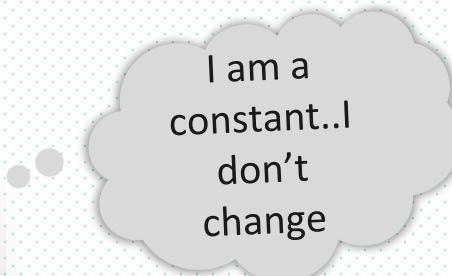
No special character expect _(underscore)

Identifiers are case sensitive

First letter cannot be a digit

Python Literals

Literals are constants in Python



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""

Extracting Individual Characters

```
In [23]: my_string="My name is John"
```

```
In [24]: my_string[0]
```

```
Out[24]: 'M'
```

```
In [5]: my_string="My name is John"
```

```
In [6]: my_string[-1]
```

```
Out[6]: 'n'
```

String Functions

Finding length of string

```
In [28]: len(my_string)  
Out[28]: 15
```

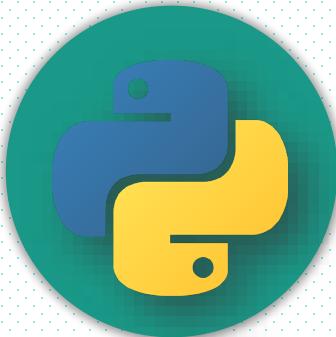
Converting String to lower case

```
In [30]: my_string.lower()  
Out[30]: 'my name is john'
```

Converting String to upper case

```
In [31]: my_string.upper()  
Out[31]: 'MY NAME IS JOHN'
```

Data structures in Python



Tuple

List

Dictionary

Set

Tuple in Python

Tuple is an ordered collection of elements enclosed within ()



Tuples are
immutable

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

Extracting Individual Elements

```
In [20]: tup1=(1,"a",True,2,"b",False)  
tup1[0]  
  
Out[20]: 1
```

```
In [21]: tup1=(1,"a",True,2,"b",False)  
tup1[-1]  
  
Out[21]: False
```

```
In [22]: tup1=(1,"a",True,2,"b",False)  
tup1[1:4]  
  
Out[22]: ('a', True, 2)
```

Modifying a Tuple

You cannot modify a tuple because it is immutable

```
In [49]: tup1[2]="hello"
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
<ipython-input-49-2fc16622751e> in <module>
```

```
----> 1 tup1[2]="hello"
```

```
TypeError: 'tuple' object does not support item assignment
```

Tuple Basic Operations

Finding length of Tuple

```
In [24]: tup1=(1,"a",True,2,"b",False)  
len(tup1)  
Out[24]: 6
```

Concatenating Tuples

```
In [25]: tup1 = (1,2,3)  
tup2 = (4,5,6)  
tup1+tup2  
Out[25]: (1, 2, 3, 4, 5, 6)
```

Tuple Functions

Minimum Value

```
In [32]: tup1=(1,2,3,4,5)  
        min(tup1)  
  
Out[32]: 1
```

Maximum Value

```
In [33]: tup1=(1,2,3,4,5)  
        max(tup1)  
  
Out[33]: 5
```

List in Python

List is an ordered collection of elements enclosed within []



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

Extracting Individual Elements

```
In [58]: l1=[1,"a",2,"b",3,"c"]  
l1[1]
```

```
Out[58]: 'a'
```

```
In [59]: l1=[1,"a",2,"b",3,"c"]  
l1[2:5]
```

```
Out[59]: [2, 'b', 3]
```

Modifying a List

Reversing elements of a list

```
In [40]: ll=[1,"a",2,"b",3,"c"]
ll.reverse()
ll

Out[40]: ['c', 3, 'b', 2, 'a', 1]
```

Sorting a list

```
In [43]: ll = ["mango","banana","guava","apple"]
ll.sort()
ll

Out[43]: ['apple', 'banana', 'guava', 'mango']
```

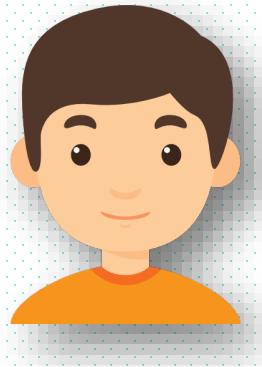
Inserting element at a specified index

```
In [41]: ll=[1,"a",2,"b",3,"c"]
ll.insert(1,"Sparta")
ll

Out[41]: [1, 'Sparta', 'a', 2, 'b', 3, 'c']
```

Dictionary in Python

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



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

Extracting Keys and Values

```
In [1]: fruit={"Apple":10,"Orange":20,"Banana":30,"Guava":40}  
fruit.keys()  
  
Out[1]: dict_keys(['Apple', 'Orange', 'Banana', 'Guava'])
```

Learning outcome

```
In [70]: fruit={"Apple":10,"Orange":20,"Banana":30,"Guava":40}  
fruit.values()  
  
Out[70]: dict_values([10, 20, 30, 40])
```

Modifying a Dictionary

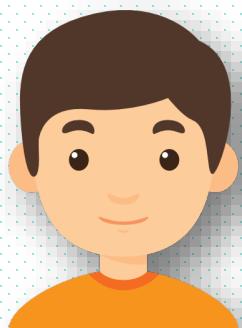
Adding a new element:

```
In [2]: fruit={"Apple":10,"Orange":20,"Banana":30,"Guava":40}  
fruit["Mango"]=50  
fruit  
Out[2]: {'Apple': 10, 'Orange': 20, 'Banana': 30, 'Guava': 40, 'Mango': 50}
```

```
In [3]: fruit={"Apple":10,"Orange":20,"Banana":30,"Guava":40,"Mango":50}  
fruit["Apple"]=100  
fruit  
Out[3]: {'Apple': 100, 'Orange': 20, 'Banana': 30, 'Guava': 40, 'Mango': 50}
```

Set in Python

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



Duplicates
are not
allowed in
a set

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

Set Operations

```
In [7]: s1={1,"a",True,2,"b",False}  
s1.add("Hello")  
s1
```

```
Out[7]: {1, 2, False, 'Hello', 'a', 'b'}
```

```
In [9]: s1={1,"a",True,2,"b",False}  
s1.remove("b")  
s1
```

```
Out[9]: {1, 2, False, 'a'}
```

```
In [8]: s1={1,"a",True,2,"b",False}  
s1.update([10,20,30])  
s1
```

```
Out[8]: {1, 10, 2, 20, 30, False, 'a', 'b'}
```

Set Functions

```
In [11]: s1 = {1,2,3}  
         s2 = {"a","b","c"}  
  
         s1.union(s2)  
  
Out[11]: {1, 2, 3, 'a', 'b', 'c'}
```

```
In [13]: s1 = {1,2,3,4,5,6}  
         s2 = {5,6,7,8,9}  
  
         s1.intersection(s2)  
  
Out[13]: {5, 6}
```

Thank You