Variables in Python

"John"

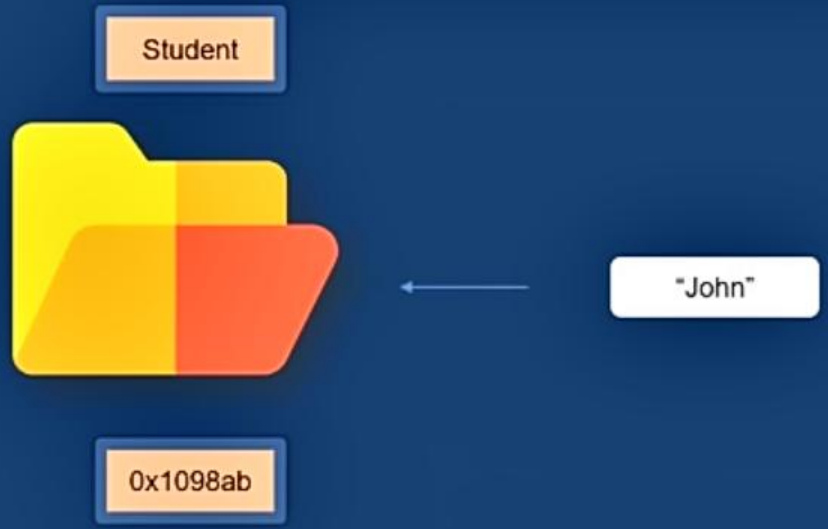
"Sam"

"Matt"

Variables in Python

greatlearning
Learning for Life

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



Variables in Python

greatlearning
Learning for Life

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

DataTypes in Python

greatlearning
Learning for Life

Every variable is associated with a data-type

10, 500	3.14, 15.97	TRUE, FALSE	"Sam", "Matt"
int	float	Boolean	

Operators in Python

greatlearning
Learning for Life

Arithmetic Operators

Relational Operators

Logical Operators

Python Tokens

greatlearning
Learning for Life

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

greatlearning
Learning for Life

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

greatlearning
Learning for Life

Literals are constants in Python

Python Strings

greatlearning
Learning for Life

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

greatlearning
Learning for Life

```
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

greatlearning
Learning for Life

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'
```

String Functions

greatlearning
Learning for Life

Replacing a substring

```
In [33]: my_string.replace('y','a')
Out[33]: 'Ma name is John'
```

Number of occurrences of substring

```
In [7]: new_string = "hello hello world"
In [8]: new_string.count("hello")
Out[8]: 2
```

String Functions

greatlearning
Learning for Life

Finding the index of substring

```
In [13]: s1 = 'This is sparta!!!'
         s1.find('sparta')

Out[13]: 8
```

Splitting a String

```
In [15]: fruit = 'I like apples, mangoes, bananas'

         fruit.split(',')

Out[15]: ['I like apples', ' mangoes', ' bananas']
```

Tuple in Python

greatlearning
Learning for Life

Tuple is an ordered collection of elements enclosed within ()


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

Extracting Individual Elements

greatlearning
Learning for Life

```
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

greatlearning
Learning for Life

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 Basic Operations

greatlearning
Learning for Life

Repeating Tuple Elements

```
In [29]: tup1 = ('sparta',300)
         tup1*3

Out[29]: ('sparta', 300, 'sparta', 300, 'sparta', 300)
```

Repeating and Concatenating

```
In [31]: tup1 = ('sparta',300)
         tup2 = (4,5,6)
         tup1*3 + tup2

Out[31]: ('sparta', 300, 'sparta', 300, 'sparta', 300, 4, 5, 6)
```

Tuple Functions

greatlearning
Learning for Life

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

greatlearning
Learning for Life

```
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

Changing the element at 0th index

```
In [35]: l1=[1,"a",2,"b",3,"c"]
         l1[0]=100
         l1

Out[35]: [100, 'a', 2, 'b', 3, 'c']
```

Popping the last element

```
In [37]: l1=[1,"a",2,"b",3,"c"]
         l1.pop()
         l1

Out[37]: [1, 'a', 2, 'b', 3]
```

Appending a new element

```
In [36]: l1=[1,"a",2,"b",3,"c"]
         l1.append("Sparta")
         l1

Out[36]: [1, 'a', 2, 'b', 3, 'c', 'Sparta']
```

Modifying a List

greatlearning
Learning for Life

Reversing elements of a list

```
In [40]: l1=[1,"a",2,"b",3,"c"]
         l1.reverse()
         l1

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

Sorting a list

```
In [43]: l1 = ["mango","banana","guava","apple"]
         l1.sort()
         l1

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

Inserting element at a specified index

```
In [41]: l1=[1,"a",2,"b",3,"c"]
         l1.insert(1,"Sparta")
         l1

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

List Basic Operations

greatlearning
Learning for Life

Concatenating Lists

```
In [44]: l1 = [1,2,3]
         l2 = ["a","b","c"]
         l1+l2

Out[44]: [1, 2, 3, 'a', 'b', 'c']
```

Repeating elements

```
In [45]: l1 = [1,"a",True]
         l1*3

Out[45]: [1, 'a', True, 1, 'a', True, 1, 'a', True]
```

Dictionary in Python

greatlearning
Learning for Life

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


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

Extracting Keys and Values

greatlearning
Learning for Life

Extracting Keys

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

Out[1]: dict_keys(['Apple', 'Orange', 'Banana', 'Guava'])
```

Extracting Values

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

Out[70]: dict_values([10, 20, 30, 40])
```

Modifying a Dictionary

greatlearning
Learning for Life

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}
```

Changing an existing element

```
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}
```

Dictionary Functions

greatlearning
Learning for Life

Update one dictionary's elements with another

```
In [4]: fruit1={"Apple":10,"Orange":20}
        fruit2={"Banana":30,"Guava":40}

        fruit1.update(fruit2)

        fruit1

Out[4]: {'Apple': 10, 'Orange': 20, 'Banana': 30, 'Guava': 40}
```

Popping an element

```
In [6]: fruit={"Apple":10,"Orange":20,"Banana":30,"Guava":40}
        fruit.pop("Orange")
        fruit

Out[6]: {'Apple': 10, 'Banana': 30, 'Guava': 40}
```

Set in Python

greatlearning
Learning for Life

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


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