Machine Learning Diploma

Python2: Sequences & Control Statements

AMIT

Agenda

- ➔ Strings (cont)
- ➔ Operators (Arithmetic, Logical...)
- ➔ List, tuple, set and Dictionary
- ➔ Control statement If Else

AMIT

1. Strings (Cont)

Printing Strings

➔ Print statement can have different formats

```
In [18]: print("my name is ", name)
my name is  ahmed

In [20]: print("my name is {} and my age is {}".format(name, age))
my name is ahmed and my age is 20

In [21]: print("my name is {0} and my age is {1}".format(name, age))
my name is ahmed and my age is 20

In [24]: print("my name is {2} and my age is {0}".format(name, age, gender))
my name is male and my age is ahmed
```


AMIT

Escape Charaters

➔ Sometimes we want to print strings having special characters. So we add backslash before the special character as follows:

“My name is \“Ahmed\”.”

➔ Escape Characters:

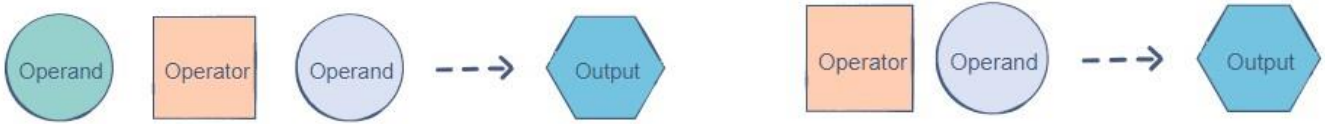
- \’: single quote
- \\: Backslash
- \n: New line
- \t: tab
- \b: Backspace

2. Operators

Operators

➔ Operators are used to perform arithmetic and logical operations on data. They enable us to manipulate and interpret data to produce useful outputs.

➔ Python operators follow the infix or prefix notations:



➔ The 5 main operator types in Python are:

Arithmetic - Comparison - Assignment - Logical - Bitwise

AMIT

PYTHON OPERATORS

Arithmetic		
Operator	**Purpose**	**Example**
+	Addition (Sum of two operands)	a + b
-	Subtraction (Difference between two operands)	a - b
*	Multiplication (Product of two operands)	a* b
/	Float Division (Quotient of two operands)	a / b
//	Floor Division (Quotient with fractional part)	a//b
%	Modulus (Integer remainder of two operands)	a % b
**	Exponent (Product of an operand n times by itself)	a ** n

AMIT

PYTHON OPERATORS

Comparison		
Operator	**Purpose**	**Example**
>	Greater than (If left > right hence return true)	a > b
<	Less than (if left < right hence return true)	a < b
==	Equal to (if left equals right return true)	a == b
!=	Not equal to (if left not equals right return true)	a != b
>=	Greater than or equal (if left GTE right return true)	a >= b
<=	Less than or equal (if left LE right return True)	a <= b

AMIT

PYTHON OPERATORS

Logical		
Operator	**Purpose**	**Example**
and	If a and b are both true hence return true	a and b
or	If either a or b is true hence return true	a or b
not	If a is true return false and vice versa	not a

AMIT

PYTHON OPERATORS

Bitwise		
Operator	**Purpose**	**Example**
&	If a and b are both one in bit level return 1	a & b
|	If a and b are either one or both in bit level return 1	a | b
~	Invert all the bits of the passed operand	~ a
^	If a and b are either 1but not both in bit level return 1	a ^ b
>>	Shift the bits of a to the right n times	a >> n
<<	Shift the bits of a to the left n times	a << n

AMIT

PYTHON OPERATORS

➔ LogicTables:

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NOT Truth Table

A	B
0	1
1	0

AMIT

PYTHON OPERATORS

Assignment		
Operator	**Purpose**	**Example**
arithmetic=	Any arithmetic operation followed by = which apply the arithmetic operation of the left operand and put the result in it.	a += 1equivalent to a = a + 1
bitwise=	Any bitwise operation followed by = which apply the bitwise operation of the left operand and put the result in it.	a &= 1equivalent to a = a & 1

AMIT

PYTHON OPERATORS

Identity		
Operator	**Purpose**	**Example**
is	If both operands refers to same object return True	a is b
Is not	If both operands refers to different objects return True	a is not b

Membership		
Operator	**Purpose**	**Example**
in	If a given value exists in a given sequence	“s” in [“a”, “n”, “s”]
not in	If a given value doesn’t exist in a given sequence	“s” in [“a”, “n”, “u”]

AMIT

Quiz:

➔ What is the operator used?

- OR
- XOR
- AND
- NOT

AMIT

Quiz(Solution):

➔ What is the operator used?

- OR
- XOR
- AND
- NOT

Quiz:

➔What is the output of printing ‘result’?

- False
- -21
- 5
- 5.25

```
x = 20
y = 5
result = (x + True) / (4 - y * False)
```


AMIT

Quiz(Solution):

➔What is the output of printing ‘result’?

- False
- -21
- 5
- 5.25

```
x = 20
y = 5
result = (x + True) / (4 - y * False)
```


AMIT

Task 1 :

➔ Gravitational force is the attractive force that exists between two masses. It can be calculated by using the following formula:

$$\frac{GMm}{r^2}$$

Write a code that accepts these constants as input and produces this output:

Sample Input

$G = 6.67 * 10^{-11}$

$M_{Earth} = 6.0 * 10^{24}$

$m_{Moon} = 7.34 * 10^{22}$

$r = 3.84 * 10^{8}$

Sample Output

$F_G = 1.99 * 10^{20}$

AMIT

3. Lists

Lists

A list is a collection of mutable items in a particular order , List constants are surrounded by square brackets and the elements in the list are separated by commas.
A list element can be any Python object - even another list

- ➔ Syntax: list_variable = *“d”, “a”, 4, 5+
- ➔ You can apply len(), type().
- ➔ It can contain different data types.
- ➔ You can use list() constructor instead of the square brackets.
- ➔ Same rules of indexing and slicing strings apply here.
- ➔ You can modify an item using basic assignment: list_variable*0+ =‘c’

Lists Important Methods

- ➔ Append(item): adds an element to the end of the list
- ➔ Insert(pos, item): adds an element at the position pos.
- ➔ Extend(another_list): concatenate another_list/any other sequence to the end
- ➔ Remove(item): removes the item from a list
- ➔ Pop(pos): removes the item at the position pos. if no pos, removes last.
- ➔ Clear(): delete items of the list but the list itself is still there.
- ➔ Copy(): you can't do list1 = list2, instead use list1 = list2.copy()

Search for a method to sort the items based on any metric.

AMIT

4. Tuples

Tuples:

- ➔ Syntax: Tuple_variable = (“a”, “b”, “c”)
- ➔ You can apply len(), type().
- ➔ It can contain different data types.
- ➔ You can use tuple() constructor instead of the brackets.
- ➔ Same rules of indexing and slicing strings apply here.
- ➔ You cannot modify an item using basic assignment: tuple_variable[0] = ‘c’

Tuples Notes & Methods

- ➔ Changing a tuple: convert into list then change then convert back into tuple.
- ➔ Join two tuples together: tuple1 + tuple2, knowing that tuple1,tuple2 > 1 element
- ➔ Multiply tuples: tuple1 * 2

- ➔ count(): Returns the number of times a specified value occurs in a tuple
- ➔ index(): Searches the tuple for a specified value and returns the position of where it was found

5. Sets

Sets:

- ➔ Syntax: Set_variable = ,“a”, “b”, 6-
- ➔ You can apply len(), type().
- ➔ It can contain different data types.
- ➔ You can use set() constructor instead of the curly brackets.
- ➔ You cannot access them the same way of indexing.
- ➔ You cannot modify an item using basic assignment: set_variable*0+ =‘c’

Sets Notes & Methods:

- ➔ Accessing Set elements use: IN operator.
- ➔ You can add items using add()
- ➔ Same as extend() in lists you can use update() to add two sets/any other sequence
- ➔ Remove(): to remove and item from the set
- ➔ Union() == Update() but union() returns a new set. Update() modifies.
- ➔ Intersection(): get the duplicated items from two sets and return a new set.
- ➔ intersection_update() same as intersection() but updates directly.

6. Dictionaries

Dictionaries:

- ➔ Syntax: Dict_variable = ,“name”: “Merna”, “age”: 20, 1: *1,2,3+-
- ➔ You can apply len(), type().
- ➔ It can contain different data types.
- ➔ You can use dict() constructor instead of the curly brackets.
- ➔ You can access them using Keys.
- ➔ You can modify an item using basic assignment: dict_variable*‘name’+= ‘Ahmed’

PYTHON DATA TYPES: DICTIONARY

Dictionaries can be deleted using the *del* function in python.

Duplicate keys are not allowed. Last key will be assigned while others are ignored. Some important built-in functions:

- ***.clear()*** to clear all elements of the dictionary.
- ***.copy()*** to copy all elements of the dictionary to another variable.
- ***.fromkeys()*** to create another dictionary with the same keys.
- ***.get(key)*** to get the values corresponding to the passed key.
- ***.has_key()*** to return True if this key is in the dictionary.
- ***.items()*** to return a list of dictionary (key, value) tuple pairs.
- ***.keys()*** to return list of dictionary dictionary keys.
- ***.values()*** to return list of dictionary dictionary values.
- ***.update(dict)*** to add key-value pairs to an existing dictionary.

AMIT

Comparison between the rest of data types:

List	Tuple	Set	Dictionary
Ordered	Ordered	Unordered	Ordered
Changeable	Unchangeable	Changeable	Changeable
Duplicates yes	Duplicates yes	No Duplicates	No Duplicates
Indexed	Indexed	Unindexed	Indexed with key

Comparison between the rest of data types:

➔ Ordered means that each element won't change its place until you modify it.

➔ Changeable means you can edit its element.

➔ No Duplicates means that it only contains unique values.

➔ Indexed means you can access each element by its index/position except dictionaries you access elements using keys.

Task 3:

➔ Write a program that accepts a sequence of comma-separated numbers from user and generates a list and a tuple with those numbers

7. Control Statements

If Else Conditional Statement:

➔ To control actions taken by the software.

Syntax:

```
If (condition):
    Statements
elif (condition):
    Statements
else:
    Statements
```

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

```
a = 2
b = 330
print("A") if a > b else print("B")
```

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```


AMIT

If Else Conditional Statement:

- ➔ Conditions are ANY comparison that yields to True or False.
- ➔ Multiple conditions can be combined using logical operators.
- ➔ Statements are executed in order Once the condition is True, The rest of following conditions are not checked.
- ➔ Nested If are allowed; if block inside another if block.

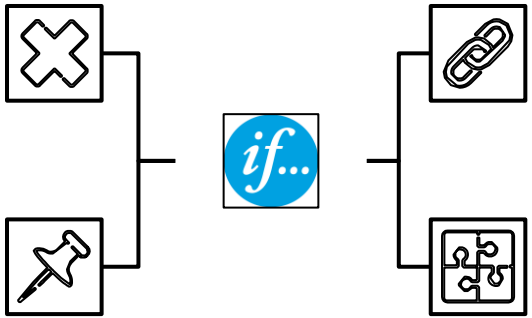
CONDITIONS

can be used to enter the if in case of false condition and the else in case of true condition.

not

in

can be used to check on the membership of an item in a data structure as lists, sets or tuples.



can be used to check on the fulfillment of all the conditions separated by the and operator.

and

or

can be used to check on the fulfillment of one of the conditions separated by the or operator.

AMIT

Any Questions?

AMIT

THANK YOU!