Python Essentials | Jan 2021 Batch 1 | Day 2 Overview

DAY 02 AGENDA | Total Duration 1:32:08

- Variable
- Data type & Keywords
- Types of Operators
- Conditional Statements
- Loops

And, to understand in detail please go through the below TIMESTAMPS

https://m.youtube.com/watch?v=Gswkz9CIEr8&feature=youtu.be

Day 2 | Jan 2021 Batch 1 | LetsUpgrade - Python Essentials

TIMESTAMP for EACH TOPIC:

2:48 - Difference between command prompt and IDLE

5:41 - Variable

9:17 - Data type

16:00 - Keyword

18:40 - Input & output function

26:24 - Assignment 1

31:27 - Operators

32:40 - Arithmetic Operators

38:33 - Relational Operators

39:38 - Assignment Operators

41:27 - Logical Operators

43:07 - Identity Operator

44:48 - Membership Operator

50:17 - Conditional Statements

58:36 - Nested Conditional Statements

1:11:57 - Loops

1:24:13 - Assignment 2

Variable:

Variable is a memory allocation that stores the data according to the data type. A variable is created the moment you first assign a value to it.

Example:

```
X=13
```

- Here **"X"** is a variable and "13" is data. The data type for 13 is an integer (int).
- Variables do not need to be declared with any particular type and can change type after they have been set.

Data Type:

Data types are the classification of data items. Python has the following data types built-in by default.

They are,

- **Text Type: str**
- **Numeric Types: int, float, complex**
- **Sequence Types: list, tuple, range**
- **Mapping Type: dict** (Dictionary)
- **Boolean Type: bool**
- **Set Types: set, frozen set**

In python data types have two categories.

- **Mutable (changeable) -** *list, dictionary and set are mutable*
- **Immutable (unchangeable)** *-Numeric (int , float, complex) ,string and tuple are immutable*

Keywords:

- Keywords are **reserved words** in Python. It cannot be used as variable names, function names, or any other identifiers.
- There are 33 keywords in the python 3.7 version. If we were using more than 3.7 or 3.8 there are 35 keywords.

Operators:

Operators are used to performing operations on variables and values. The values the operator works on are called **operands**.

Example: `10 + 13`

Here, 10 and 13 are operands and '+' is the operator that performs an operation between those numbers.

Types of operators:

Arithmetic Operators:

Arithmetic operators are used with numeric values to perform common mathematical operations.

- '+' Used to return the sum of two numbers
- '-' Used to return the difference between two numbers
- '*' Used to return the product of two numbers
- '\' Used to divide two numbers and return the quotient of it
- '%' Also used to divide two numbers and but it returns the remainder of it
- '//'Floor division - division that results into the whole number adjusted to the left in the number line
- '**' Exponent - left operand raised to the power of right

Relational operators:

Relational operators is a relationship between two variables. It is also known as Comparison operators.

- '>' Greater than - True if the left operand is greater than the right
- '<' Less than - True if the left operand is less than the right
- '==' Equal to - True if both operands the equal
- '!=' Not equal to - True if operands are not equal
- '>=' Greater than or equal to - True if the left operand is greater than or equal to the right
- '<=' Less than or equal to - True if left operand is less than or equal to the right

Assignment operators:

Assignment operators are used to assigning values to variables.

- '=' Eg: x=5
- '+=' Eg: x+=5(x=x+5)
- '-=' Eg: x-=5(x=x-5)
- '%=' Eg: x%=5(x=x%5)
- '*=' Eg: x*=5(x=x*5)
- '**=' Eg: x**=5(x=x**5)
- '/=' Eg: x/=5(x=x/5)
- '//=' Eg:x//=5(x=x//5)

Conditional statements:

Conditional statements in python perform different computations or actions depending on whether a specific Boolean constraint evaluates to true or false.

There are 3 types of conditional statements.

- if- Conditional statement
- elif - Conditional statement
- else- Conditional statement

if- Conditional statement:

If the first condition is true then we use if-conditional statement. If we have only one condition we can check if the condition satisfies then the output excueted else no output is excueted

Example:

```
if 3<4:
        print(“Hello”)
```

Output: `Hello`

elif - Conditional statement:

If we have two or more conditions we can check if the condition 1 satisfies then the output 1 excueted or condition 2 satisifies output 2 is excueted.If we need to check two or more conditions we can use this statement.

Example:

```
 if 3>4:
&amp;nbsp; print(“Hello”)
 elif 4>3:
print(“Hi”)
```

Output: `Hi`

else - Conditional statement:

If we have two or more conditions we can check them if the condition satisfies then the corresponding output statement excueted.No condition is satisified then else statement is excueted.The else keyword catches anything which isn't caught by the preceding conditions.

Example:

```
if 3>4:
        print(“Hello”)
elif 4>3:
        print(“Hi”)
else:
        print(“Bye”)
```

Output: `Hi`

Loops:

Python has two primitive loop commands.

- while loop
- for loop

while loop:

while loop is loop which can execute a set of statements as long as the condition is true.

Syntax:

```
while condition:
        #statement
        #increment / decrement
```

Example:

```
a= int(input(“Enter a number:”))
while a<10:
 &amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; print(“Hello”,a)
 &amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; a=a+1
else:
 &amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; print(“bye”)
```

Output:

```
Enter a number: 1
Hello 1
Hello 2
Hello 3
Hello 4
Hello 5
Hello 6
Hello 7
Hello 8
Hello 9
bye
```

🚀**Assignment and Materials for the day have been uploaded in the drive.**

💾**Attendance Form: https://forms.gle/nwMDKzpMEbVf2jHz9**

📨 **Assignment Submission Form: https://forms.gle/NC2UA5pHGXveETbu7**

📱 **Assignment and Materials Details: https://drive.google.com/drive/folders/1Rlh3eEarHNPB_j1J3CFRngh1S8E_8an8?usp=sharing**

Batch 1 | All Details | Python Essentials JAN 2021