

Python Class Presentation

Topic - Basic Syntax, Variables, Keywords,
and Data Types in Python

Done by -
Akash Yadav
Adarsh Kumar
Ali Abbas Thakkar

Basic Syntax in Python

1. Conditional Statements

a) if statement

```
if(condition):  
    (statements)
```

b) if-else statement

```
if(condition):  
    (statements)  
else:  
    (statements)
```

c) if-else-if statement

```
if(condition 1):  
    (statements)
```

```
elif(condition 2):  
    (statements)
```

```
else:  
    (statements)
```

d) nested-if statement

```
if(condition 1):  
    if(condition 2):  
        (statements)
```

2. Iteration Statements / Looping

a) while loop

```
while(condition):  
    (statements)
```

b) for loop

```
for i in range (condition):  
    (statements)
```

c) nested loop

```
while(condition):  
    for i in range (condition):  
        (statements)
```

Variables

A Python variable is a reserved memory location to store values. In other words, a variable in a python program gives data to the computer for processing.

Every value in Python has a datatype. Different data types in Python are Numbers, List, Tuple, Strings, Dictionary, etc. Variables can be declared by any name or even alphabets like a, aa, abc, etc.

Declaring and Initialising Variables

Code

```
num = 22  
print(num)
```

Code

```
name='Python'  
print(name)
```

Code

```
marks=69.12  
print(marks)
```

Output

22

Output

Python

Output

69.12

Local Variables - Scope is limited to the function

Global Variables - Can be used across all functions

Deleting a Variable

Code:

```
num=46
```

```
·
```

```
·
```

```
·
```

```
del num
```

Keywords

Keywords in Python are reserved words that cannot be used as ordinary identifiers. They must be spelled exactly as they are written.

All the keywords of Python contain lower-case letters only.

Examples:

```
If( b<a and c<a):
```

```
    print("a is greatest")
```

```
elif(c<b):
```

```
    print("b is greatest")
```


Reserved Keywords

and	assert	in	del	else
raise	from	if	continue	not
pass	while	finally	yield	is
as	break	return	elif	except
def	global	import	for	or
print	lambda	with	class	try

Data Types

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

There are various data types in Python. Some of the important types are listed in the next slide

1. Python's Numbers:

Integers, floating point numbers and complex numbers falls under Python numbers category.

They are defined as int, float and complex class in Python.

Code

```
a = 5
```

```
print(a, "is of type", type(a))
```

```
a = 2.0
```

```
print(a, "is of type", type(a))
```

```
a = 1+2j
```

```
print(a, "is complex number?", isinstance(1+2j,complex))
```

2. Python List

List is an ordered sequence of items. It is one of the most used datatype in Python and is very flexible. All the items in a list do not need to be of the same type.

Lists are mutable, meaning, value of elements of a list can be altered.

Code:

```
thislist = ["apple", "banana", "cherry"]
```

```
thislist[1] = "blackcurrant"
```

```
print(thislist)
```

Output: ['apple', 'blackcurrant', 'cherry']

3. Python Tuple

Tuple is an ordered sequence of items same as list. The only difference is that tuples are immutable. Tuples once created cannot be modified.

It is defined within parentheses () where items are separated by commas.

Code:

```
thistuple = ("apple", "banana", "cherry")
```

```
thistuple[1] = "blackcurrant"
```

```
# The values will remain the same:
```

```
print(thistuple)
```

Output: ("apple", "banana", "cherry")

4. Dictionary

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

Code:

```
thisdict = {"brand": "Ford", "model": "Mustang",  
"year": 1964}  
  
print(thisdict)
```

Output:{'brand': 'Ford', 'model': 'Mustang', 'year':
1964}

5. String Literals

String literals in python are surrounded by either single quotation marks, or double quotation marks. 'hello' is the same as "hello".

Code:

```
a = "Hello, World!"  
print(a[1])  
OUTPUT: Hello, World!
```

Code:

```
a = "Hello, World!"  
print(a.lower())  
OUTPUT:hello, world!
```