

Advanced Programming-Python

Presented by
Siddhanta Borah

Contents of Todays (Day-2) Discussion

- Modes of programming- In Python IDE
- Commenting in python
- Python Identifiers
- Keyword in python programming
- Variable declaration and Initialization
- Standard Data Types-Python programming
- `type()` Function

MODES OF PROGRAMMING

1. Interactive Mode:

Invoking the interpreter without passing a script file as a parameter brings up the following prompt. Type the following text at the Python prompt and press the Enter –

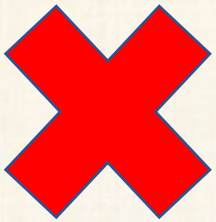
```
>>> print("hello")
```

MODES OF PROGRAMMING

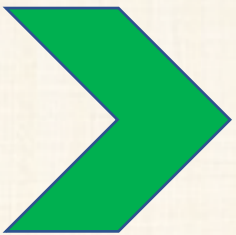
2. Script Mode

1. Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.
2. Python files have extension .py. Type the following source code in a Sample.py file. `print ("Hello")`

COMMENTING IN PYTHON



```
>>> 8+9 addition
SyntaxError: invalid syntax      # Output
>>>
```



```
>>> 8+9 #addition
17      # Output
>>>
```

PYTHON IDENTIFIER

A Python identifier is the name given to a variable, function, class, module or other object. An identifier can begin with an alphabet (A – Z or a – z), or an underscore (_) and can include any number of letters, digits, or underscores. Spaces are not allowed.

Examples of Identifiers	
Valid	Invalid
MyName	My Name (Space is not allowed)
My_Name	3dfg (cannot start with a digit)
Your_Name	Your#Name (Only alphabetic character, Underscore (_) and numeric are allowed)

RESERVED KEYWORDS IN PYTHON

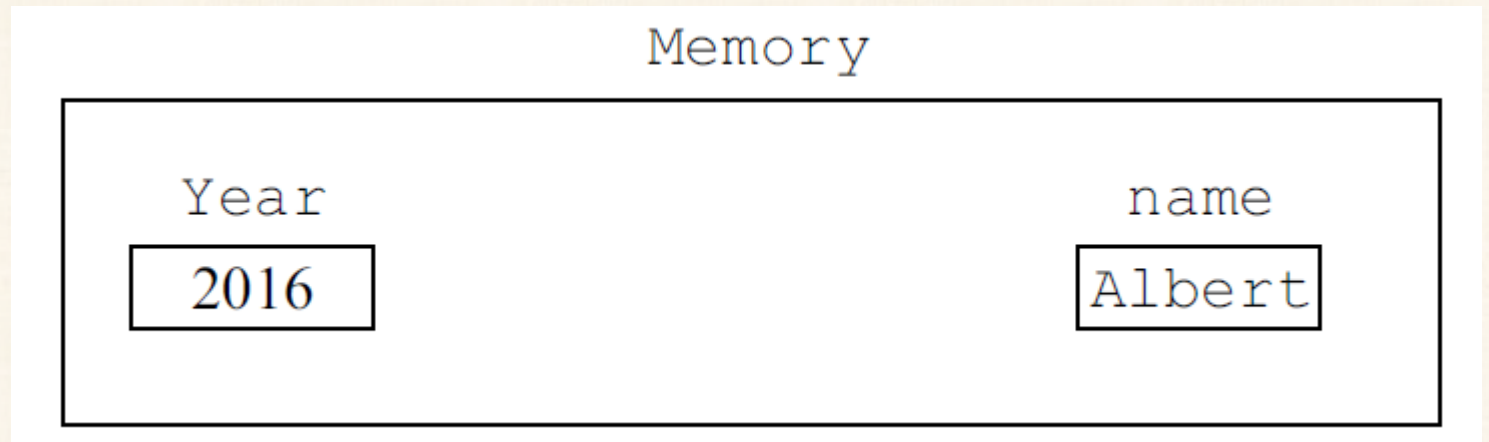
A list of reserved keywords in Python:

and	del	from	None	True
as	elif	global	nonlocal	try
assert	else	if	not	while
break	except	import	or	with
class	False	in	pass	yield
continue	finally	is	raise	
def	for	lambda	return	

VARIABLE DECLARATION AND INITIALIZATION

- In Python, variables do not need to be declared explicitly in order to reserve memory spaces as in other programming languages like C, Java, etc. When we initialize the variable in Python, Python Interpreter automatically does the declaration process.
- The general format of initializing a variable is as follows:
Variable = expression

```
>>> year=2016  
>>> name='Albert'
```



N.B: Whenever two values are successively assigned to a variable, the interpreter will forget the previous value assigned to it and store the latest value in the variable memory space.

STANDARD DATA TYPES-PYTHON PROGRAMMING

1. Numeric:

```
>>> num1=2          # integer number
>>> num2=2.5        # real number (float)
>>> num1
2                   # Output
>>> num2
2.5                # Output
>>>
```

```
In [9]: 5/2
Out[9]: 2.5

In [10]: 5*2
Out[10]: 10
```

```
In [14]: 2**3
Out[14]: 8
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

2. String:

- Single quotes or double quotes are used to represent strings
- There are many operations that can be performed on a string. There are several operators such as slice operator ([]) and [:]), concatenation operator (+), repetition operator (*), etc. Slicing is used to take out a subset of the string, concatenation is used to combine two or more than two strings and repetition is used to repeat the same string several times.

```
>>> sample_string = "Hello"    # store string value
>>> sample_string              # display string value
'Hello'                        # Output
>>> sample_string + "World"    # use of + operator
'HelloWorld'                   # Output
>>> sample_string * 3          # use of * operator
'HelloHelloHello'              # Output
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

String:

Example

```
>>>sample_string="Hello"  
>>>sample_string[1]          # display 1st index element.  
'e'                          # Output  
>>>sample_string[0:2]        # display 0 to 1st index elements  
'He'                         # Output
```

Example

```
>>> sample_string = "HelloWorld"
```

```
In [26]: sample_string [1:8]  
Out[26]: 'ellowor'
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

3. List:

- A list can contain the same or different type of items
- To declare a list in Python, we need to separate the items using commas and enclose them within square brackets ([])
- Similar to the string data type, the list also has plus (+), asterisk (*) and slicing [:] operators for concatenation, repetition and sub-list, respectively.

```
>>>first=[1,"two",3.0,"four"]      # 1st list
>>>second=["five", 6]              # 2nd list
>>>first                            # display 1st list
[1, 'two', 3.0, 'four']            # Output
>>>first+second                     # concatenate 1st and 2nd list
[1, 'two', 3.0, 'four', 'five', 6]# Output
>>>second * 3                       # repeat 2nd list
['five', 6, 'five', 6, 'five', 6] # Output
>>>first[0:2]                       # display sublist
[1, 'two']                         # Output
>>>
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

4. Tuple:

- Similar to a list, a tuple is also used to store sequence of items.
- Like a list, a tuple consists of items separated by commas. However, tuples are enclosed within parentheses rather than within square brackets.
- The main difference between tuple and a list is that Lists are mutable whereas Tuples are immutable. Tuples are read only lists. Once the items are stored, the tuple cannot be modified. The details is shown with the help of one example in next slide.

STANDARD DATA TYPES-PYTHON PROGRAMMING

Tuple:

```
In [31]: list1
Out[31]: [1, 2, 'one']
```

```
In [35]: list1[0]="one"
```

```
In [36]: list1
Out[36]: ['one', 2, 'one']
```

```
In [37]: list3=(7,8,9,'one')
```

```
In [38]: list3
Out[38]: (7, 8, 9, 'one')
```

```
In [39]: list3[1]=4
```

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

```
File "C:\Users\borah\AppData\Local\Temp\ipykernel_14516\153996877.py", line 1, in <module>
    list3[1]=4
```

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

STANDARD DATA TYPES-PYTHON PROGRAMMING

5. Dictionary:

- A Python dictionary is an unordered collection of key-value pairs.
- When we have the large amount of data, the dictionary data type is used. Keys and values can be of any type in a dictionary.
- Items in dictionary are enclosed in the curly-braces{ } and separated by the comma (,). A colon (:) is used to separate key from value. A key inside the square bracket [] is used for accessing the dictionary items.

```
>>> dict1 = {1:"first line", "second":2}    # declare dictionary
>>> dict1[3] = "third line"                # add new item
>>> dict1                                  # display dictionary
{1: 'first line', 'second': 2, 3: 'third line'}    #Output
>>> dict1.keys()                           # display dictionary keys
[1, 'second', 3]                               # Output
>>> dict1.values()                         # display dictionary values
['first line', 2, 'third line']                # Output
```


STANDARD DATA TYPES-PYTHON PROGRAMMING

6. Boolean:

- In a programming language, mostly data is stored in the form of alphanumeric but sometimes we need to store the data in the form of 'Yes' or 'No'. In terms of programming language, Yes is similar to True and No is similar to False.
- This True and False data is known as Boolean Data and the data types which stores this Boolean data are known as Boolean Data Types.

Example

```
>>> a = True
>>> type(a)
<type 'bool'>

>>> x = False
>>> type(x)
<type 'bool'>
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

7. Sets:

- In python programming language, unordered collection of data known as Set.
- Union, Intersection, Difference and Symmetric Difference are some operations which are performed on sets.
 - ❑ **Union:** Union operation performed on two sets returns all the elements from both the sets. It is performed by using `&` operator.
 - ❑ **Intersection:** Intersection operation performed on two sets returns all the element which are common or in both the sets. It is performed by using `|` operator.
 - ❑ **Difference:** Difference operation performed on two sets set1 and set2 returns the elements which are present on set1 but not in set2. It is performed by using `-` operator.
 - ❑ **Symmetric Difference:** Symmetric Difference operation performed on two sets returns the element which are present in either set1 or set2 but not in both. It is performed by using `^` operator.

STANDARD DATA TYPES-PYTHON PROGRAMMING

Sets

Example

```
# Defining sets
>>> set1 = set([1, 2, 4, 1, 2, 8, 5, 4])
>>> set2 = set([1, 9, 3, 2, 5])

>>> print set1                                #Printing set
set([8, 1, 2, 4, 5])                           #Output
>>> print set2
set([1, 2, 3, 5, 9])                           #Output

>>> intersection = set1 & set2                 #intersection of set1 and set2
>>> print intersection
set([1, 2, 5])                                 #Output

>>> union = set1 | set2                        # Union of set1 and set2
```

STANDARD DATA TYPES-PYTHON PROGRAMMING

Sets

```
>>> print union
set([1, 2, 3, 4, 5, 8, 9])           #Output

>>> difference = set1 - set2         # Difference of set1 and set2
>>> print difference
set([8, 4])                          #Output

>>> symm_diff = set1 ^ set2          # Symmetric difference of set1 and
                                     set2
>>> print symm_diff
set([3, 4, 8, 9])                   #Output
```

TYPE() FUNCTION

- `type()` function in Python programming language is a built-in function which returns the datatype of any arbitrary object.
- The object is passed as an argument to the `type()` function. `Type()` function can take anything as an argument and returns its datatype, such as integers, strings, dictionaries, lists, classes, modules, tuples, functions, etc.

```
>>> x = 10
>>> type(x)
<type 'int'>
```

#Output

```
>>> type('hello')
<type 'str'>
```

#Output

TYPE() FUNCTION (CONTD..)

```
>>> tup = (1,2,3)
```

```
>>> type(tup)
```

```
<type 'tuple'>
```

#Output

```
>>> li = [1,2,3]
```

```
>>> type(li)
```

```
<type 'list'>
```

#Output

Thank You