For Aptech Centre Use Only

Data Types in Python

Session 02

Session Overview

In this session, you will be able to:

- Describe the hierarchy of a Python program
- Identify the built-in data types in Python
- List the functions and methods of different built-in data types

For Aptech Centre Use Only

Python Program Hierarchy and Data Types (1-2)

- Composed of **modules**, **statements**, **objects**, and **expressions**.

Program Hierarchy

A program contains one or more modules.

A module has statements.

Statements have expressions.

Expressions generate and process objects.

For Aptech Centre Use Only

Python Program Hierarchy and Data Types (2-2)

Useful objects in Python:

Type	Example
Strings	'John', ‘person's’
Numbers	567, 9.264, i+4j, 0b110
Dictionaries	{'food':'rice', 'drink':'cola'}, dict(hours=5, mins =10)
Files	open('session.txt')
Tuples	(14, 'June', 2017, 11, 35, 22), tuple('jar')
Sets	set(‘xyz')
Others (Core Types)	Boolean, None

For Aptech Centre Use Only

Numeric Types (1-3)

- Numeric objects that Python supports:

  - Integer and floating-point
  - Decimal (fixed-precision)
  - Complex number
  - Fraction (rational numbers)
  - Boolean (`true` and `false`)
  - Built-in functions and modules such as `round`, `math`, and `random`
  - Expressions, bitwise operations, unlimited integer precision, and hex, octal, and binary formats

For Aptech Centre Use Only

Numeric Types (2-3)

A developer can use IDLE interactively to create data types to understand Python objects. Python supports:

- Integers, which are whole numbers either positive or negative (... -1, 0, 1...).
- Floating-point numbers, which have their fractional part, such as 1.4 and 31.4e-10.

For Aptech Centre Use Only

Numeric Types (3-3)

- Complex numbers such as 2+6j and 4J.
- Octal, hex, and binary numbers such as 00166, 0x8ff, and 0b101010, respectively.
- Booleans called bool in Python such as true and false.

For Aptech Centre Use Only

Special Numeric Types

- In Python, Boolean is called bool which is subclass of int.
- True and False are integers, and we can do arithmetic operations on them.

```
>>> True + True
>>> False * 10
```

```
Output:
2
0
```

- Using Boolean expressions as numbers

```
>>> a = 3; b = 4
>>> (a < b) * 10 + (a == b) * 20
```

```
Output:
10
```

For Aptech Centre Use Only

Operations on Numbers (1-2)

- Operations include a type of numbers or other objects and operators for calculating values during runtime.
- Python allows writing an expression using the operator symbols and the standard mathematical notation.
- For example, to add two numbers a and b, the expression is `a + b`.

For Aptech Centre Use Only

Operations on Numbers (2-2)

- It tells Python to use the + operator to the values of `a` and `b`.
- The result is the sum of `a` and `b` and is another number object.

For Aptech Centre Use Only

Numeric Functions (1-2)

Trigonometric functions:

- **`math.acos(x)`** : Gives the x's arc cosine, in radians.
- **`math.asin(x)`** : Gives the x's arc sine, in radians.
- **`math.atan(x)`** : Gives the x's arc tangent, in radians.
- **`math.cos(x)`** : Gives the cosine of x radians.
- **`math.sin(x)`** : Gives the sine of x radians.
- **`math.tan(x)`** : Gives the tangent of radians.

For Aptech Centre Use Only

Numeric Functions (2-2)

Hyperbolic functions in Python:

- **`math.acosh(x)`** : Gives the inverse hyperbolic cosine of x.
- **`math.asinh(x)`** : Gives the inverse hyperbolic sine of x.
- **`math.atanh(x)`** : Gives the inverse hyperbolic tangent of x.
- **`math.cosh(x)`** : Gives the hyperbolic cosine of x.
- **`math.sinh(x)`** : Gives the hyperbolic sine of x.
- **`math.tanh(x)`** : Gives the hyperbolic tangent of x.

For Aptech Centre Use Only

Strings

String is a popular data type.

How to create a string in Python?

Code Snippet:

```
>>> name = "John"
>>> print (name)
```

Output:

```
John
```

For Aptech Centre Use Only

Escape Characters (1-2)

Escape characters used in Python:

Escape Character	Description
\a	Indicates an alert or a bell.
\b	Represents a backspace.
\cx or \C-x	Represents the action, control + x.
\e	Represents the action of the Esc key.
\f	Indicates form feed.
\M-\C-x	Indicates meta-control-x.

For Aptech Centre Use Only

Escape Characters (2-2)

Escape Character	Description
\n	Indicates the newline character to insert a new line.
\nnn	Represents an octal notation, with n in the range 0.7.
\r	Indicates carriage return.
\s	Inserts space.
\t	Inserts tabbed space.
\v	Represents vertical tab.
\x	Represents character x.
\xnn	Represents a hexadecimal notation, where n is in the range 0.9, a.f, or A.F

For Aptech Centre Use Only

String Operations (1-2)

Different operations on strings:

Consider a="Peter" and b="Python"

Operator	Description	Example
+	Concatenation: Adds values on any side of the operator.	>>> a+b 'PeterPython'
*	Repetition: Generates new strings by concatenating the multiple copies of the same string.	>>> a*2 'PeterPeter'
[]	Slice: Returns the character from the specified index.	>>> a[1] 'e'
[:]	Range Slice: Returns characters from the given range of indices.	>>> a[1:3] 'ete'

For Aptech Centre Use Only

String Operations (2-2)

Operator	Description	Example
in	Membership: Gives a boolean value, True, if the specified character is present in the given string.	>>> 'oh' in a False
not in	Membership: Gives a boolean value, True, if the specified character is absent in the given string.	>>> 'y' not in a True
r/R	Raw String: Overtakes the actual meaning of escape characters. The syntax is the same as a standard Python string, but has the raw string operator, r, preceding the first quote mark.	>>> print r'\n' \n >>>print R'\n' \n
%	Format: Formats a string.	>>>print "My name is %s" %('John') My name is John

For Aptech Centre Use Only

Triple Quotes

How to use triple quotes?

Code Snippet:

```
>>>para_str = """A long para with several lines
and escape characters such as TAB ( \t ). The
escape characters shall show up when displayed.
Even the new lines even if given within the
brackets [ \n ] shall also show up."""
>>> print (para_str)
```

```
Output:
```

A long para with several lines and escape characters such as TAB (	). The escape characters shall show up when displayed. Even the new lines even if given within the brackets [ ] shall also show up.

For Aptech Centre Use Only

Built-in String Methods (1-3)

Built-in methods of string:

Method	Description
string.capitalize()	Converts the first letter of a string in upper case.
string.center(width[, fillchar])	Gives a space-padded string and centers the original string as per the total width columns.
string.count(str, beg= 0,end=len(string))	Counts the number of times string str is present in a string or substring in between the start index beg and end index.
string.endswith(suffix, beg=0, end=len(string))	Returns a boolean value indicating whether a string or substring concludes with suffix.
string.find(str, beg=0 end=len(string))	Returns the index position of string str if it is present in string or substring. It returns -1 if not found.
string.index(string, beg=0, end=len(string))	Is identical to find(), but generates an exception if the string is not found.

For Aptech Centre Use Only

Built-in String Methods (2-3)

Method	Description
string.isalnum()	Returns true if a string has alphanumeric literals and one of them is a character.
string.islower()	Returns true if the string's characters are in lower case.
string.isupper()	Returns true if the string's characters are in upper case.
string.len()	Returns the total length of a string.
string.lstrip()	Removes the leading white spaces.
max(string)	Returns the greatest alphabetical character from string. For example, it returns y from string merry.
min(string)	Returns the lowest alphabetical character from the specified string.

For Aptech Centre Use Only

Built-in String Methods (3-3)

Method	Description
string.replace(old,new[,max])	Replaces all occurrences of old with new in a string. It replaces with max occurrences if max is specified.
string.rfind(string, beg=0,end=len(string))	Performs a backward search in a string and is identical to find().
string.rindex(string, beg=0, end=len(string))	Performs a backward search in a string and is identical to index().
string.split(str='', num=string.count(str))	Divides a string as per the delimiter string str and returns an array of substrings. If there is no delimiter specified, it uses space to split the string. If number is specified, the division results into that many substrings.
string.startswith(str, beg=0,end=len(string))	Returns true if a string or a substring starts with string str.
string.upper()	Converts all letters that are in lower case into upper case.

For Aptech Centre Use Only

Lists

How to create and access a list?

Code Snippet:

```
>>> colors = [orange, red, blue]
>>> print colors[0]
>>> print colors[2]
>>> print len(colors)
```

```
Output:
orange
red
3
```

For Aptech Centre Use Only

Built-in List Methods

Assume two lists namely,

`list = [1,2,3,4,5,6,7]` and

`list1 = [8,9,10]`.

List methods in Python:

`list.append(element)`	`list.insert(index, element)`	`list.extend(list1)`	`list.index(element)`
`list.remove(element)`	`list.sort()`	`list.reverse()`	`list.pop(index)`

For Aptech Centre Use Only

Built-in List Functions

Python supports following built-in functions on lists:

- `min(list1)`
- `max(list1)`
- `list(sequence)`
- `len(list1)`

Where list1 is list and sequence is tuple.

For Aptech Centre Use Only

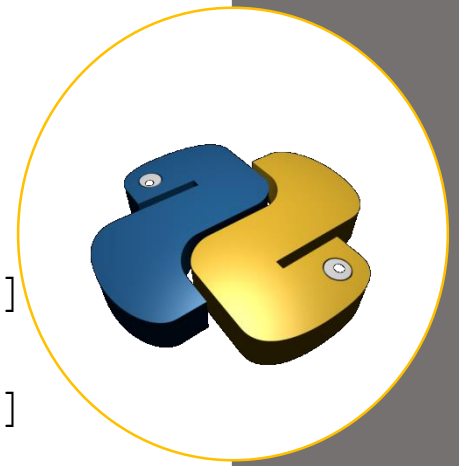
Slicing Lists

How to slice a list in different ways?

Code Snippet:

```
>>> values = [1,2,3,4,5,6]
>>> values[1:3] # Index 1 to index 3.
>>> values[2:-1] #Index 2 to index one from last.
>>> values[:2] #Start through index 2.
>>> values[2:] #Index 2 through end.
>>> values[::2] #Start through end and skip ahead
2 places each time.
```

```
Output:
[2, 3]
[3, 4, 5]
[1, 2]
[3,4,5,6]
[1,3,5]
```



For Aptech Centre Use Only

Tuples

- Tuple is a series of immutable objects. It is a sequence identical to a list.
- How to create a tuple?

Code Snippet:

```
>>> my_tuple = (1, 2, 3, 4, 5)
>>> single_elem_tuple = (1,)
```

For Aptech Centre Use Only

Dictionaries

- Python has an efficient value/key hash table structure termed as '`dict`'.
- How to create a dictionary?

Code Snippet:

```
>>> dict1 = {}
>>> dict1['rd'] = 'red'
>>> dict1['yel'] = 'yellow'
>>> dict1['bl'] = 'blue'
>>> print (dict1)
```

```
Output:
{'rd': 'red', 'yel':
'yellow', 'bl': 'blue'}
```

For Aptech Centre Use Only

Built-in Dictionary Functions and Methods (1-2)

Few built-in functions for dictionary objects:

```
len(dict)
```

```
str(dict)
```

```
%type(variable)
```

For Aptech Centre Use Only

Built-in Dictionary Functions and Methods (2-2)

Python provides following dictionary methods:

- `dict.clear()`
- `dict.copy()`
- `dict.fromkeys(sequence, value)`
- `dict.get(key_name)`
- `dict.has_key(key_name)`
- `dict.items()`
- `dict.update(dict1)`
- `dict.keys()`
- `dict.values()`

For Aptech Centre Use Only

Summary (1-4)

- Python programs are composed of modules, statements, expressions, and objects.
- Python programming involves implementing built-in or user-defined object data structures to represent different components.
- Built-in data types or objects in Python are strings, numbers, tuples, dictionaries, lists, and sets.

For Aptech Centre Use Only

Summary (2-4)

- Program units such as modules, classes, and functions are also objects.
- Python supports a variety of numeric literals such as float, binary, hexadecimal, octal, and complex numbers.
- Python supports a variety of numerical operations such as ternary selection, logical OR and AND, comparisons, exponentiation, and bitwise operations.

For Aptech Centre Use Only

Summary (3-4)

- Strings in Python are enclosed in single or double quotes.
- Some of the useful string operations are concatenate, slicing, membership, and raw string.
- List literals are within square brackets.
- Some of the useful list methods are `append()`, `insert()`, `sort()`, `reverse()`, and `extend()`.
- It is possible to slice a list using the format [`start`, `stop`, and `increment`], which is similar to what is applied to a tuple.

For Aptech Centre Use Only

Summary (4-4)

- A tuple is a series of immutable objects that are unchangeable.
- Unlike lists, tuples are in parentheses, hold diverse data types, and are quick to iterate.
- Python has an efficient key/value hash table structure called a dictionary whose contents are in key-value pairs within braces { }.

For Aptech Centre Use Only