

Programming and Problem Solving through Python Language O Level / A Level

Chapter -3: Introduction to Python Language

Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them. Variables can store data of different types, and different types can do different things.

Python has five standard data types –

- Numbers
- String
- List
- Tuple
- Dictionary

Python has the following data types built-in by default, in these categories:

Text Type:	<code>str</code>
Numeric Types:	<code>int</code> , <code>float</code> , <code>complex</code>
Sequence Types:	<code>list</code> , <code>tuple</code> , <code>range</code>
Mapping Type:	<code>dict</code>
Set Types:	<code>set</code> , <code>frozenset</code>
Boolean Type:	<code>bool</code>
Binary Types:	<code>bytes</code> , <code>bytearray</code> , <code>memoryview</code>

Example of Data Types

Example	Data Type
<code>x = "Hello World"</code>	<code>str</code>
<code>x = 20</code>	<code>int</code>
<code>x = 20.5</code>	<code>float</code>
<code>x = 1j</code>	<code>complex</code>

x = ["apple", "banana", "cherry"]	list
x = ("apple", "banana", "cherry")	tuple
x = range(6)	range
x = {"name" : "John", "age" : 36}	dict
x = {"apple", "banana", "cherry"}	set
x = frozenset({"apple", "banana", "cherry"})	frozenset
x = True	bool
x = b"Hello"	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryview

Setting the Specific Data Type

If you want to specify the data type, you can use the following constructor functions:

Example	Data Type
x = str("Hello World")	str
x = int(20)	int
x = float(20.5)	float
x = complex(1j)	complex
x = list(("apple", "banana", "cherry"))	list
x = tuple(("apple", "banana", "cherry"))	tuple
x = range(6)	range
x = dict(name="John", age=36)	dict
x = set(("apple", "banana", "cherry"))	set
x = frozenset(("apple", "banana", "cherry"))	frozenset
x = bool(5)	bool
x = bytes(5)	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryview

Python Numbers

Number data types store numeric values.

Python supports four different numerical types –

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values)
- complex (complex numbers)
 - Python allows you to use a lowercase l with long, but it is recommended that you use only an uppercase L to avoid confusion with the number 1. Python displays long integers with an uppercase L.
 - A complex number consists of an ordered pair of real floating-point numbers denoted by $x + yj$, where x and y are the real numbers and j is the imaginary unit.

Int

Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

Example **Integers:**

```
x = 1  
y = 35656222554887711  
z = -3255522
```

Float

Float, or "floating point number" is a number, positive or negative, containing one or more decimals.

Example **Floats:**

```
x = 1.10  
y = 1.0  
z = -35.59
```

Float can also be scientific numbers with an "e" to indicate the power of 10.

Example **Floats:**

```
x = 35e3  
y = 12E4  
z = -87.7e100
```

Complex

Complex numbers are written with a "j" as the imaginary part:

Example **Complex:**

```
x = 3+5j  
y = 5j  
z = -5j
```

Type Conversion

We can convert from one type to another with the `int()`, `float()`, and `complex()` methods:

Example

Convert from one type to another:

```
x = 1 # int
```

```
y = 2.8 # float
```

```
z = 1j # complex
```

```
#convert from int to float:
```

```
a = float(x)
```

```
#convert from float to int:
```

```
b = int(y)
```

```
#convert from int to complex:
```

```
c = complex(x)
```

```
print(a)
```

```
print(b)
```

```
print(c)
```

```
print(type(a))
```

```
print(type(b))
```

```
print(type(c))
```