

Department of Social Statistics
Faculty of Humanities and Social Sciences
University of Sri Jayewardenepura
ITE/MIT 1213 Fundamentals of Programming

Tutorial – 01

Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

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

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set
Boolean Type:	bool

Getting the Data Type

You can get the data type of any object by using the **type()** function:

Example

Print the data type of the variable x:

```
x = 5  
print(type(x))
```

Setting the Data Type

In Python, the data type is set when you assign a value to a variable:

Example	Data Type
x = "Hello World"	str
x = 20	int
x = 20.5	float
x = 1j	complex
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 = True	bool

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

<code>x = float(20.5)</code>	float
<code>x = complex(1j)</code>	complex
<code>x = list(("apple", "banana", "cherry"))</code>	list
<code>x = tuple(("apple", "banana", "cherry"))</code>	tuple
<code>x = range(6)</code>	range
<code>x = dict(name="John", age=36)</code>	dict
<code>x = set(("apple", "banana", "cherry"))</code>	set
<code>x = bool(5)</code>	bool

There are four collection data types in the Python programming language:

- [List](#) is a collection which is ordered and changeable. Allows duplicate members.
- [Tuple](#) is a collection which is ordered and unchangeable. Allows duplicate members.
- [Set](#) is a collection which is unordered, unchangeable (but you can remove items and add new items), and unindexed. No duplicate members.
- [Dictionary](#) is a collection which is ordered and changeable. No duplicate members.

When choosing a collection type, it is useful to understand the properties of that type. Choosing the right type for a particular data set could mean retention of meaning, and, it could mean an increase in efficiency or security.