

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` , `frozenset`

- Boolean Type: `bool`

- Binary Types: `bytes` , `bytearray` , `memoryview`

- None Type: `NoneType`

Getting the Data Type

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

```
In [1]: # Print the data type of the variable x:
```

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

```
<class 'int'>
```

Setting the Data Type

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

Example Data Type Try it

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 = frozenset({"apple", "banana", "cherry"}) frozenset

x = True bool

x = b"Hello" bytes

x = bytearray(5) bytearray

x = memoryview(bytes(5)) memoryview

x = None NoneType

Setting the Specific Data Type

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

```
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
```

In []: