

Types

Introduction

- ❖ The data in Python is in the form of objects, either objects of new types we define or objects of built-in types that Python provides.
- ❖ An object in Python, as in other OOP languages, is just a piece of memory with values and associated operations.
- ❖ As we shall see, there are no type declarations in Python. The syntax of the executed expression determines the types of the objects we create and use.

The `type` Function

- ❖ Using the `type` function we can get the type of values we have in our code.

```
a = [3, 5, 21, 23, 5]  
print(type(a))
```

Samples for Built-in Types

| Type | Examples |
|-------|---------------------------------------|
| float | 12.4 |
| str | 'abc', "abc", "ab'c" |
| list | [12, [2,3,4], 'a'] |
| dict | {'one':'The One', 'two': 'Two Files'} |
| tuple | (1, 'abc', 23, "A") |
| set | {'a', 'b', 'c'} |

Types Categories

- ❖ The available types are grouped into categories. Each category and its characteristics.

The Numbers Category

- ❖ This category includes the following types:

`int, float, long, decimal and complex.`

- ❖ Each type of this category is expected to support addition, multiplication etc.

The Sequences Category

- ❖ This category includes the `str`, `list`, `bytearray`, `bytes`, and `tuple`.
- ❖ Each and every type that belong to this category supports indexing, slicing and concatenation.

The Set Category

❖ This category includes the types `set` and `frozenset`.

When instantiating these types we get an object that represents a collection of other objects that cannot repeat themselves.

The Set Category

```
a = {3, 5, 21, 23, 5, "fafa", 5, 3, 23, 23, "fafa"}  
print(a)
```



The Mappings Category

- ❖ This category includes `dict`. Having an object of the `dict` type we can use it to hold key-value pairs.

The Mappings Category

```
a = { 123123:"haim michael", 42534:"moshe solomon",  
      454234:"david magen"}  
  
print(a.get(542534))
```



Types Conversion

- ❖ We can force a type conversion by calling one of the available built-in functions.

```
int(4.2+1.2)
```

```
float(40)
```

Variables

- ❖ Variables are created when they are first assigned with a value.

```
num = 12 #There is no need to define a variable in advance
```

Variables

- ❖ Variables we use in expressions must be assigned with a value before we use them.