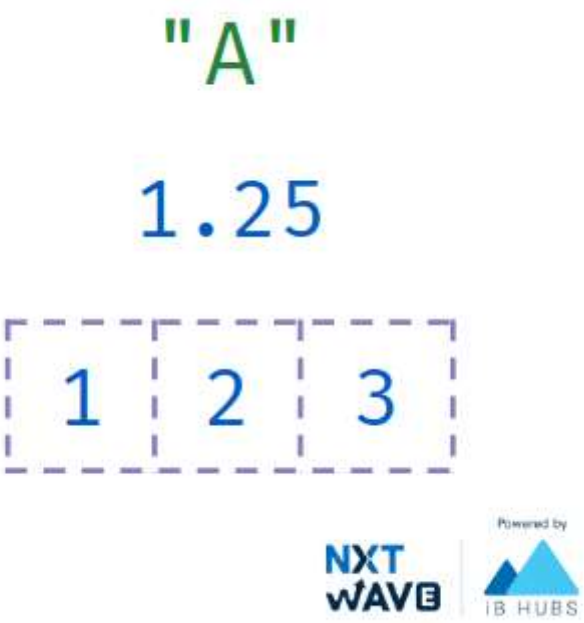
Miscellaneous Topics

1. What is an Object in Python?

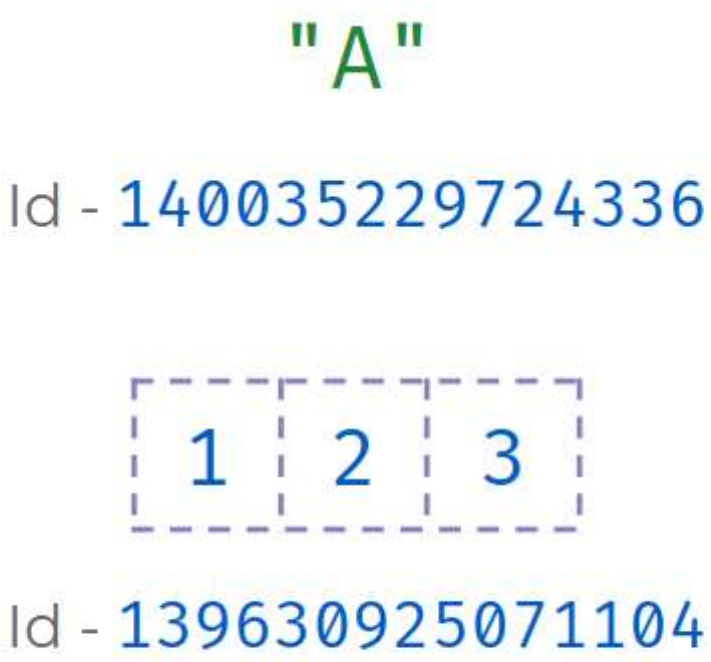
In general, anything that can be assigned to a variable in Python is referred to as an **object**.

Strings, Integers, Floats, Lists, Functions, Modules etc... are all objects.



Identity of an Object

Whenever an object is created in Python, it will be given a **unique identifier (id)**. This unique id can be different each time you run the program.



Every object that you use in a Python program will be stored in Computer Memory

The unique id will be related to the location where the object is stored in the Computer Memory.

2. What are Modules in Python?

In the Python context, any file containing a Python code is called a **Module**.

Some examples of modules are collections, random, datetime, math, etc...

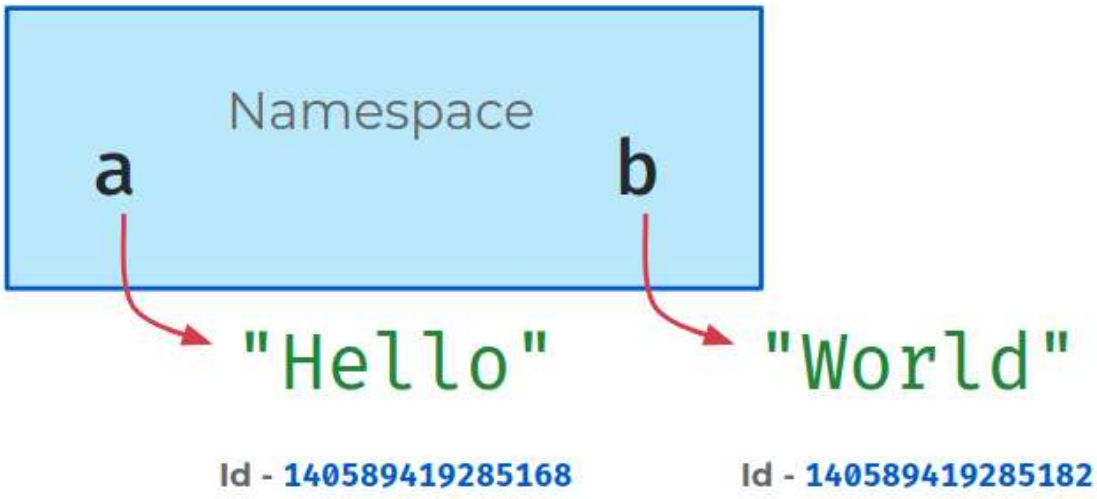
3. What are Packages in Python?

In the Python context, any file containing a Python code is called a Module. A Package is a collection of modules.

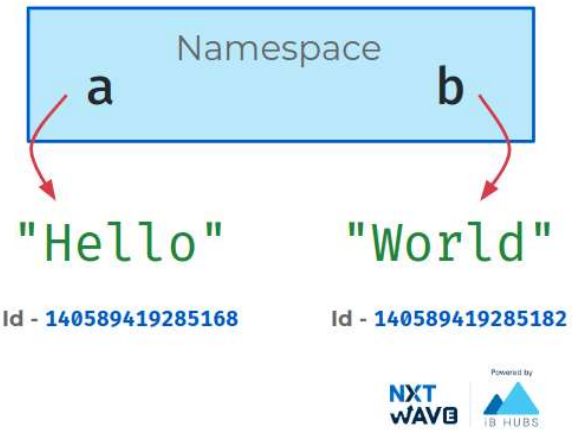
4. What are Namespaces?

A **namespace** is a collection of currently defined names along with information about the object that the name references.

It ensures that names are **unique** and won't lead to any conflict.



Namespaces allow us to have the same name referring different things in **different namespaces**.



Code

PYTHON

```python
def greet_1():
    a = "Hello"
    print(a)
    print(id(a))

def greet_2():
    a = "Hey"
    print(a)
    print(id(a))

print("Namespace - 1")
greet_1()
print("Namespace - 2")
greet_2()
```

Collapse

Output

```
Namespace - 1
Hello
140639382368176
Namespace - 2
Hey
140639382570608
```

5. What is Scope?

The scope of a variable is the region in which that variable can be accessed.

6. What are Global and Local variables?

Global Variables:

If a variable is declared outside of all functions and conditional statements then that variable is known as **Global variable**.

These can be accessed anywhere in the program. If the value of the global variable is modified inside a function or conditional statement then the changes are reflected in the rest of the program.

Local Variables:

If a variable is declared inside of a function or conditional statement then that variable is known as **Local variable**.

These can be accessed inside a function or conditional statement in which they are declared. If the value of the local variable is modified in one function, then the changes are not reflected in another function.

7. What are Exceptions in Python?

Even when a statement or expression is syntactically correct, it may cause an error when an attempt is made to execute it. Errors detected during execution are called **Exceptions**.

8. How to Handle Exceptions in Python?

Python provides a way to **catch** the exceptions that were raised so that they can be properly handled.

- Exceptions can be handled with **try-except** block.
- Whenever an exception occurs at some line in the try block, the execution stops at that line and jumps to except block.

PYTHON

```python
try:
    # Write the code that might cause exceptions
except:
    # The code to be run when there is an exception.
```

Code

PYTHON

```python
try:
    print(x)
except:
    print("x is not defined")
```

Output

```
x is not defined
```

Handling Specific Exceptions

We can specifically mention the **name of the exception** to catch all exceptions of that specific type.

Syntax

PYTHON

```python
try:
    # Write code that might cause exceptions
except Exception:
    # The code to be run when there is an exception
```

Example

Code

PYTHON

```python
try:
    print(x)
except NameError:
    print("x is not defined")
except:
    print("Unhandled Exception")
```

Output

```
x is not defined
```

Handling Multiple Exceptions

We can write **multiple exception blocks** to handle different types of exceptions differently.

Syntax

PYTHON

```python
try:
    # Write code that might cause exceptions
except Exception1:
    # The code to be run when there is an exception1
except Exception2:
    # The code to be run when there is an exception
```

Example_1

Code

PYTHON

```python
try:
    a = int(input())
    b = int(input())
    c = a/b
    print(c)
except ZeroDivisionError:
    print("Denominator can't be 0")
except ValueError:
    print("Input should be an integer")
except:
    print("Something went wrong")
```

Collapse

Input

```
Rahul
Teja
```

Output

```
Input should be an integer
```

Example_2

Code

```python
try:
    a = int(input())
    b = int(input())
    c = a/b
    print(c)
except ZeroDivisionError:
```

```
    print("Denominator can't be 0")
except ValueError:
    print("Input should be an integer")
except:
    print("Something went wrong")
```

Collapse

Input

```
1
0
```

Output

```
Denominator can't be 0
```

MARK AS COMPLETED

Submit Feedback

Notes Discussions