Variables

Variables are also used to store the data types like string, integer, and float values.

Example

In [2]:
```
message1 = 'Hello Python World'
message2 = "Hello Python World"
message3 = """Hello Python World"""
message4 = 100

print(message1)
print(message2)
print(message3)
print(message4)
```

```
Hello Python World
Hello Python World
Hello Python World
100
```

A variable holds a value. You can change the value of variable at any point.

Naming Rules

Variables can only contain letters, numbers and underscores

Variable names can start with a letter or an underscore, but cannot start with an underscore.

Spaces are not allowed in variable names, so we use underscores instead of spaces. For example, use student_name instead of "student name".

You cannot use python keywords as variable names.

Variables names should be descriptive, without being too long. For example, mc_wheels instead of just "wheels" and number_of_wheels_on_a_motorcycle

Variable names are case sensitive (Ram, ram, RAM are three different variables)

In [ ]:
```
#Legal Variable Names
myvar = "hackers"
my_var = "hackers"
_my_var = "hackers"
myVar = "hackers"
MYVAR = "hackers"
myvar2 = "hackers"

#Illegal Variable Names
```

```
2myvar = "hackers"
my-var = "hackers"
my var = "hackers"
```

NameError

There is one common error when using variables, that you will most certainly encounter at some point. Take a look at this code, and see if you can figure out why it causes an error.

In [4]:
```
# my variable name is message and not messages
message =  "Thank you for sharing python with the world, Guido"
print(messages)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[4], line 3
      1 # my variable name is message and not messages
      2 message =  "Thank you for sharing python with the world, Guido"
----> 3 print(messages)

NameError: name 'messages' is not defined
```

Let's look through this error message. First, we see it is a NameError. Then we see the file that caused the error, and a green arrow shows us what line in that file caused the error. Then we get some more specific feedback, that "name 'mesage' is not defined".

You may have already spotted the source of the error. We spelled message two different ways. Python does not care whether we use the variable name "message" or "mesage". Python only cares that the spellings of our variable names match every time we use them.

We can fix NameErrors by making sure all of our variable names are spelled consistently.

In [5]:
```
#Assigning a value to a variable
message_for_kids = "preparing young minds for future with AI"
print(message_for_kids)
```

```
preparing young minds for future with AI
```

In [6]:
```
#Changing the value of a variable
message_for_kids = "preparing yound minds for future with AI"
print(message_for_kids)
message_for_kids= 100
print(message_for_kids)
```

```
preparing yound minds for future with AI
100
```

In [7]:
```
 #Assigning different values to different variables in single line
a,b,c = 5,2.6,"Hello"
print(a)
print(type(a))
print(b)
```

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

```
5
<class 'int'>
2.6
<class 'float'>
Hello
<class 'str'>
```

In [8]:
```
# Assigning same value to different variable
x=y=z="Same"
print(x)
print(y)
print(z)
```

```
Same
Same
Same
```

In [9]:
```
# deleting a variable
num = 12
print(num)
del num
print(num)
```

```
12
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 5
      3 print(num)
      4 del num
----> 5 print(num)

NameError: name 'num' is not defined
```

In [10]:
```
# variable type
# string type variable
# integer type variable
# float type variable
var = "hello world"
var1 = 45
var2 = 5.009
var = "a"
```

In [12]:
```
#boolean variable type
python = True
java = False
```

Exercises

Hello World - variable

Store your own version of the message "Hello World" in a variable, and print it.

In [13]:
```
hw = "Hello World"
print(hw)
```

```
Hello World
```

One Variable, Two Messages:

Store a new message in the same variable, and then print that new message.

In [14]:
```
var = "Hello World"

var = "Hello Python"
print(var)
```

```
Hello Python
```