Numbers

In [1]:
```
2+3
```
Out[1]: 5

In [2]:
```
2*2
```
Out[2]: 4

In [3]:
```
2/4
```
Out[3]: 0.5

In [8]:
```
# Modulo or the "mod" operator -> Returns the remainder
2 % 4
```
Out[8]: 2

In [9]:
```
50 % 5
```
Out[9]: 0

In [10]:
```
2 ** 2
```
Out[10]: 4

In [11]:
```
2 ** 3
```
Out[11]: 8

In [12]:
```
(2 + 3) * (3 + 1)
```
Out[12]: 20

Variable Assignment -> Data are assigned to variables, so that they can be easily referenced easily later on in the code.

Python uses DYNAMIC TYPING. This means that we can reassign variables to different data types.

In [26]:
```
my_dogs = 2
```

In [27]:
```
my_dogs
```
Out[27]: 2

In [28]:
```
my_dog = 2
my_dog = ["cindy","kiril"]
```

In [29]:
```
a = 5
```

In [30]:
```
b = 10
```

In [31]:
```
a + b
```

Out[31]: 15

In [32]:
```
a = a + b
```

In [33]:
```
# Now the value of a is permanently affected
a
```

Out[33]: 15

In [34]:
```
a
```

Out[34]: 15

In [35]:
```
type(a)
```

Out[35]: int

In [36]:
```
c = 10.2
type(c)
```

Out[36]: float

In [37]:
```
my_income = 100
tax_rate = 0.5
my_tax = my_income * tax_rate
print(my_tax)
```

```
50.0
```

Strings

Because strings are ordered sequences, we can use indexing and slicing on them.

Slicing Syntax -> [start:end:step]

start will be the numerical index of the value

stop will be the last value excluded

step will be count of skips between the values

In [40]:
```
## This will print only the lastly printed string
"hello world1"
"hello world2"
```

Out[40]: 'hello world2'

In [41]:
```
print("Hello world")
```

```
Hello world
```

In [42]:
```
len("Hello")
```

Out[42]: 5

String Indexing & Slicing

In [43]:
```
myString = "Hello World"
```

In [44]:
```
myString[0]
```

Out[44]:
```
'H'
```

In [45]:
```
myString[0:3]
```

Out[45]:
```
'Hel'
```

In [46]:
```
myString[-1]
```

Out[46]:
```
'd'
```

In [47]:
```
myString[-3]
```

Out[47]:
```
'r'
```

In [48]:
```
myString[::-1]
```

Out[48]:
```
'dlroW olleH'
```

String properties and methods

Strings are immutable

In [49]:
```
name = "Sam"
```

In [50]:
```
name[0] = "p"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-50-2a92c6ab6b20> in <module>
----> 1 name[0] = "p"

TypeError: 'str' object does not support item assignment
```

In [54]:
```
x = 'Hello World'
x + " it's beautiful outside"
```

Out[54]:
```
"Hello World it's beautiful outside"
```

In [56]:
```
letter = "z"
letter * 10
```

Out[56]:
```
'zzzzzzzzzz'
```

In [62]:
```
# Methods
x = "Hello World"
print(x.upper())
print(x.lower())
print(x.split())
```

```
HELLO WORLD
hello world
['Hello', 'World']
```

In [65]:
```
y = "It's gonna be a long journey"
print(y.split("n"))
```

```
["It's go", '', 'a be a lo', 'g jour', 'ey']
```

Print formatting with strings

This is known as string interpolation

In [66]:
```
print("Hello, my name is {name} and I'm {age} years old".format(name = "Sai",ag
e = 21))
```

```
Hello, my name is Sai and I'm 21 years old
```

In [69]:
```
name = "Sai"
age = 21
print(f"Hello, my name is {name} and I'm {age} years old")
```

```
Hello, my name is Sai and I'm 21 years old
```

In [75]:
```
result = (325/340) * 100
```

In [80]:
```
# {value:width.precision f}
print(f"You've scored {result:1.2f} percentile")
```

```
You've scored 95.59 percentile
```

In [ ]:
```

```