# PYTHON TYPES & DATA STRUCTURES

## DICT

```
my_dict = {}                                          # empty dictionary
my_dict = { "key1": "value1", "key2": "value2" }      # create a dictionary
my_dict["arjan"] = "codes"                            # add a record
my_dict["arjan"] = "still codes"                      # change the value
my_dict["arjan"]           # return value if the key is present, raises a KeyError if not
my_dict.get("arjan")       # return value if the key is present, returns None if not
"arjan" in my_dict         # True if the key is in the dictionary, False otherwise
my_dict.keys()             # list containing the keys
my_dict.values()           # list containing the values
len(my_dict)               # the number of key-value pairs
del my_dict["arjan"]       # delete a key-value pair from the dict
my_dict.clear()            # delete all the key-values pairs
my_dict | { "k3": "v3" }   # merge 2 dictionaries into a new one (Python >=3.9)
```

## Numeric types

```
int           # integer numbers
              # (unlimited precision)
float         # floating point numbers
complex       # complex numbers (re+im)

fractions.Fraction (std lib)
decimal.Decimal (std lib)
```

## SET

```
my_set = set()                     # empty set
my_set = {1, 2, 3, 4}              # create a set
2 in my_set                        # True
my_set.difference({1, 2})          # {3, 4}
my_set.intersection({1, 2, 6})     # {1, 2}
my_set.isdisjoint({1, 2, 3, 6})    # False
my_set.issubset({1, 2, 3, 4, 5})   # True
my_set.union({6, 7})               # {1, 2, 3, 4, 5, 6, 7}
my_set | {6, 7}                    # {1, 2, 3, 4, 5, 6, 7} (Python >= 3.9)
```

## Common numeric operations

```
x + y         # sum                pow(x, y) # power
x - y         # difference         x ** y    # power
x * y         # product            x % y     # remainder
x / y         # quotient           -x        # negation
x // y        # floored quotient
abs(x)        # absolute value
```

## LIST

```
my_list = []                      # empty list
my_list = [1, 2, 3, 1, 2]         # create a list
my_list[0] = 2                    # change a value
my_list.append(3)                 # append the value 3
[1, 12, 3, -5, 12].count(12)      # count the number of occurrences (result: 2)
len(my_list)                      # number of items in the list
my_list.clear()                   # clear the list
4 in my_list                      # True if the item is in the list, False otherwise
del my_list[3]                    # remove item at index 3
my_list.index(5)                  # index of the first occurrence of 5
                                  # raises a ValueError if the item is not in the list
[1, 2, 3] + [4, 5, 6]             # concatenate two lists (result: [1, 2, 3, 4, 5, 6])
```

## Strings

```
x = "hi"                 # create a string
"hi" + "!"               # "hi!"
"a" * 4                  # "aaaa"
"a b c".split()          # ["a", "b", "c"]
"hi".upper()             # "HI"
"Hi".lower()             # "hi"
"hi".capitalize()        # "Hi"
" x ".strip()            # "x"
"hi".startswith("h")        # True
"hi".endswith("x")          # False
"hi".replace("i", "o")      # "ho"
```

## TUPLE

```
my_tuple = ()             # empty tuple
my_tuple = (1, 2, 3)      # create a tuple
a, b, c = my_tuple        # unpack the tuple
my_tuple.count(1)         # number of 1s in the tuple
my_tuple[1]               # 2
my_tuple.index(2)         # index of the value 2
len(my_tuple)             # length of the tuple
a, b = b, a               # using a tuple to swap two values
(1, 2) + (3, 4)           # (1, 2, 3, 4)
(1, 2) * 3                # (1, 2, 1, 2, 1, 2)
3 in (1, 2, 3)            # True
```

## Boolean operations

```
x or y        # if x is false, then y, else x
x and y       # if x is false, then x, else y
not x         # if x is false, then True, else False
```

## Comparisons

```
<             # strictly less than
<=            # less than or equal
>             # strictly greater than
>=            # greater than or equal
==            # equal
!=            # not equal
is            # object identity
is not        # negated object identity
```

version 1.2

www.arjancodes.com