- type() : get the type of a variable
- print() : output variables
- int()
- float()
- str()

} Type Casting

- input() : allow user input
- len() : .For a string : get the nb of letters
  .For a list : get the nb of elements

→ String Methods :

string.count(value) : nb of times the value occurs
string.find(value) : return the position of the value and -1 if the value is not found
string.replace(oldvalue, newvalue) : replace values
string.lower() / upper()
string.split(separator) : split the string at the specified separator
string.strip(characters) : remove space

→ List Methods :

list.append(element) : add element at the end of the list
list.insert(pos, elmnt) : add element at a specific position
list.remove(elmnt) : remove the specified element
list.pop(pos) : remove element at a specific position
list.clear() : remove all elements from the list
list.extend(iterable) : adds element from any iterable (list, set, tuple, ...) to the end of the current list.
list.sort() : sort the list ascending by default
list.sort(reverse = True) : sort the list descending

list.reverse() : reverse the order of the list
list.copy() : return a copy of the list
list.count(value) : return the nb of elements with specified value
list.index(elmnt) : return the position of the element

→ **Math built-in functions**
- .max() : highest value in an iterable
- .min() : lowest value in an iterable
- .abs() : absolute value
- .pow(x, y) : x^y

→ **Math Module**
!! need to import math
- .math.sqrt() : Square root of a number
- .math.pi : return the value of PI

→ **Numpy** (import numpy as np)
np.array([, , ,]) : create a numpy ndarray object

arr.ndim : attribute that return the dimension of an array

ndmin : argument used is the array() function to define the nb of dimension

- arr.dtype : property that check the data type of the array
- dtype : optional argument used in the array() function that allow us to create an array with a defined data type.
- arr.astype(type); function that create a copy of the array and allow us to specify the data type
- arr.copy() : function that make a copy of an array (new array)
- arr.view() : View of the original array (affected by changes)
- arr.base : attribute that check if an array owns its data
  if yes it return None (case of copy)
  if no it return the original array (case of view)
- arr.shape : return the shape of the array (nb of element in each dimension)
- arr.reshape() : function that change the shape of an array (its a view not copy)

!! arr.reshape(-1) : convert multidimensional array into 1D array

- np.nditer(arr) : function used in the for loop to iterate through each scalar of the array without the need to use n for loops

- where(condition) : Searching an array for a certain value and return index

np.searchSorted (arr, value) : Methode that return the index where the specified value would be inserted to maintain the sort order

np.sort(arr) : Sort the specified array

→ **Random** (From numpy import random)

- .randint(100) : random int from 0 to 100
- .rand() : random float from 0 to 1
- .randint(100, size=(5)) : 1D array with 5 random int from 0 to 100
- .randint(100, size=(3,5)) : 2D array with 3 rows, each row have 5 random int from 0 to 100
- .rand(5) : 1D array with 5 random float
- .rand(3,5) : 2D array with 3 rows, each one contain 5 random float

- random.choice([...]): generate a random value based on an array of values
- random.choice([...], size=(3,5)): generate 2-D array with 3 rows in each containing 5 values from the provided array
- we can specify the probability of occurance for each value in the provided array using the p argument (0 to 1)... with the sum of all probabilities should be 1
- random.shuffle(arr): change the arrangement of element in the array inplace
- random.permutation(arr): change the arrangement of element and return a new array

→ Seaborn (import seaborn as sns)
(import matplotlib.pyplot as plt)

- sns.displot(arr) } visualize random distributions
  plt.show() } with a histogram
- sns.displot(arr, hist=False) } visualize random dist
  plt.show() } without histogram

- random.normal(): method to get a Normal (Gaussian) distribution
  - => it take 3 paremeters :
    - loc : Mean
    - scale: Standard deviation
    - size: Shape of the returned array

- random.binomial() : method to get a binomial distribution
  - => it take 3 paramaters
    - n: number of trials
    - p: probability of occurence of each trial
    - size: The shape of the returned array

- random.poisson() : method to get a poisson distribution
  - -> it take 2 parameters:
    - lam: rate or Know nb of occurrences
    - size: Shape of the returned array

- random.uniform(): method to get a uniform distribution
  - -> it take 3 parameters:
    - a: lower bound (default 0)
    - b: upper bound (default 1)
    - size: Shape of the return array

→ Definitions:

1) Normal Distribution:

. Distribution that fits the probability distribution of many events
. It's continuous
. The curve of the Normal Distribution is also known as the Bell Curve because the bell-shaped curve

2) Binomial Distribution:

. Distribution that describe the outcome of binary scenarios
. It's discrete
. If there are enough data points it will be similar to Normal dist.

3) Poisson Distribution:

. Estimate how many times an event can happen in a specific time
. It's discrete

4) Uniform Distribution:

Used to describe probability where every event has equal chances of occuring

→ Benefits of using random numbers in Python

Using Random nb in Python provides essential tools for tasks such as simulation, modeling, random sampling, game development, cryptography machine learning, algorithm randomization, testing, and user experience enhancement.