

BTEC Level 3 Computing

Unit 1 - Principles of Computer Science

Built In Functions (Python 3.x)

Built In Functions Definition



A built-in function is a pre-made function that is already available in a programming language.



You can use it without needing to create it yourself.



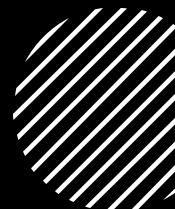
These functions perform common tasks, like printing text, doing math, or manipulating data.



Reinventing the wheel is most likely not necessary.

Built-in functions

Built In Functions Categories



The three main types are:



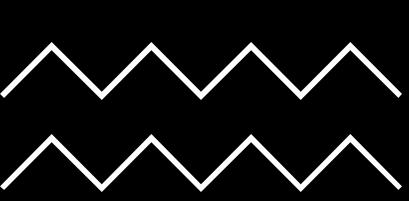
Arithmetic functions.



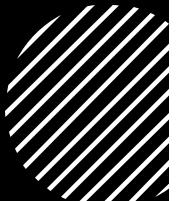
String handling functions.



General functions.



Arithmetic Operations

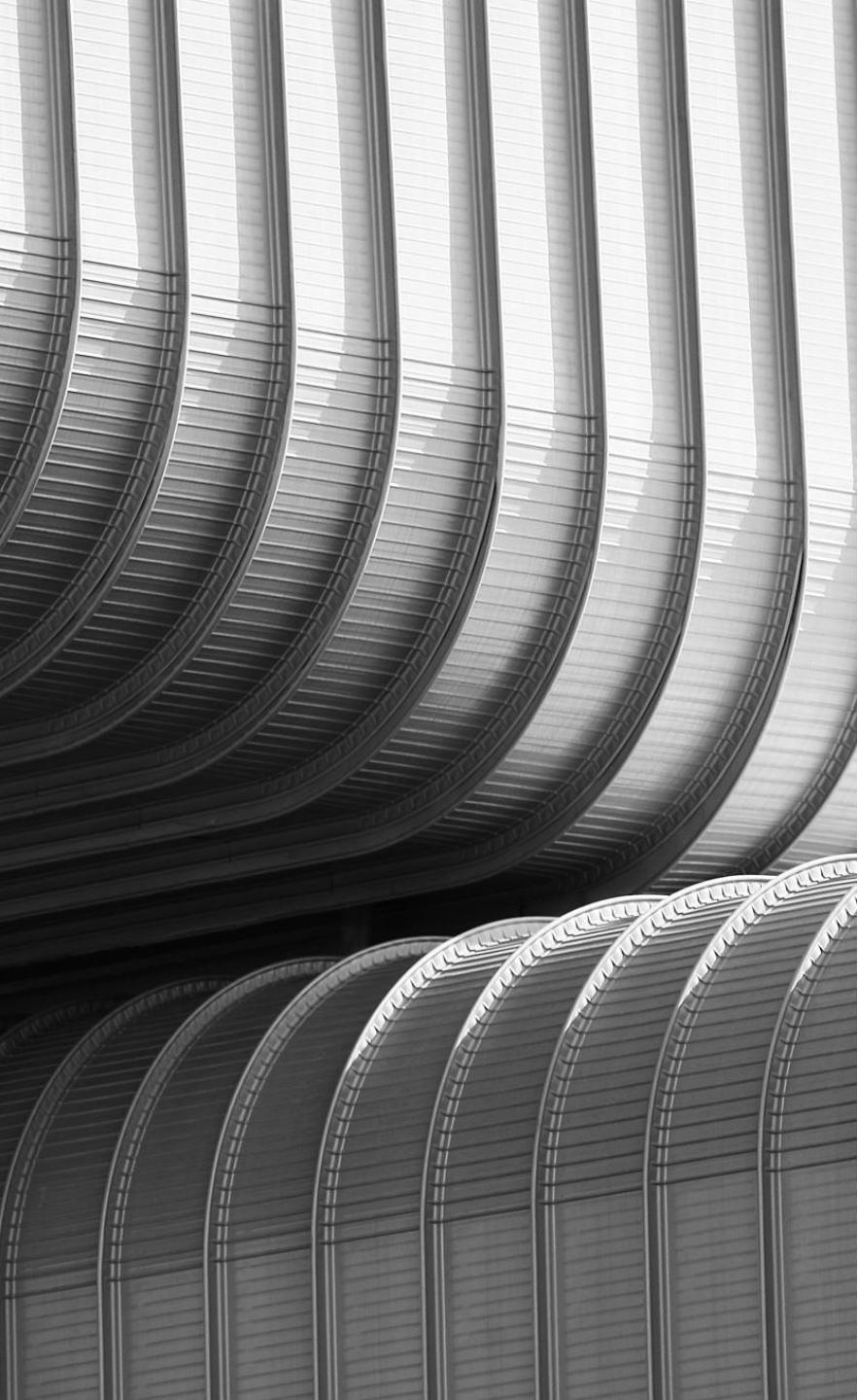


Random.

Range.

Round.

Truncation.



String Handling

Concatenation.

Length.

Position.

String conversion:

- integer/float to string
- string to integer/float.

General Functions



Input. Open. Print. Range.

Arithmetic Function - "RANDOM"

- The purpose of the random function in programming languages is to generate a number or value that appears to be chosen by chance.
- It helps create unpredictable results, which is useful for various purposes like:
 - Games.
 - Simulations.
 - Security.
 - Testing.





Arithmetic Function - "RANDOM" Example

- Code provided as a file.
- ```
import random

Print a random integer between 1 and 10 (inclusive)
print(random.randint(1, 10))

Print a random floating-point number between 0 and
1
print(random.random())

Choose a random element from a list
my_list = ["apple", "banana", "cherry"]
print(random.choice(my_list))
```



# Arithmetic Function - "RANGE"

---

- The range function in programming is a shortcut for creating a sequence of numbers.
- Instead of typing out a long list of numbers, you can use range to generate them automatically.
- For example, range(5) would generate the sequence 0, 1, 2, 3, 4. This is often used in loops to repeat a block of code a specific number of times.

# Arithmetic Function - "RANGE"

## Example

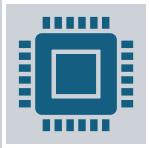
- # Arithmetic Function - "RANGE" Example

```
start_number = 10
end_number = 1251
```

```
for temp_value in range (start_number, end_number):
 print(temp_value)
```



# Arithmetic Function - "ROUND"



The round function in programming languages is used to make a number simpler by adjusting it to the nearest whole number or to a specific number of decimal places.



For example, if you have 3.14159 and use `round(3.14159, 2)`, it will give you 3.14.



It is useful when you want to reduce the number of digits after the decimal point, making the number easier to work with or display.

# Arithmetic Function - "ROUND" Example

- # Arithmetic Function - "ROUND" Example

```
my_height = 1.25673
```

```
print (round(my_height)) # rounds to a whole number
```

```
print (round(my_height, 2)) # this checks the third
```

# Arithmetic Function - "TRUNCATION"



Truncation in programming means cutting off the decimal part of a number, leaving only the whole number part.



It does NOT round the number; it simply removes everything after the decimal point.



For example, if you truncate 5.99, it becomes 5, and if you truncate "-3.75", it becomes "-3".



Truncation is useful when you want just the integer part of a number without rounding.

# Arithmetic Function - "TRUNCATION"

## Example

- # Arithmetic Function - "TRUNCATION" Example

```
import math

number = 3.14159

print(math.trunc(number)) # using math function

print(int(number)) # forces truncation by type casting
```



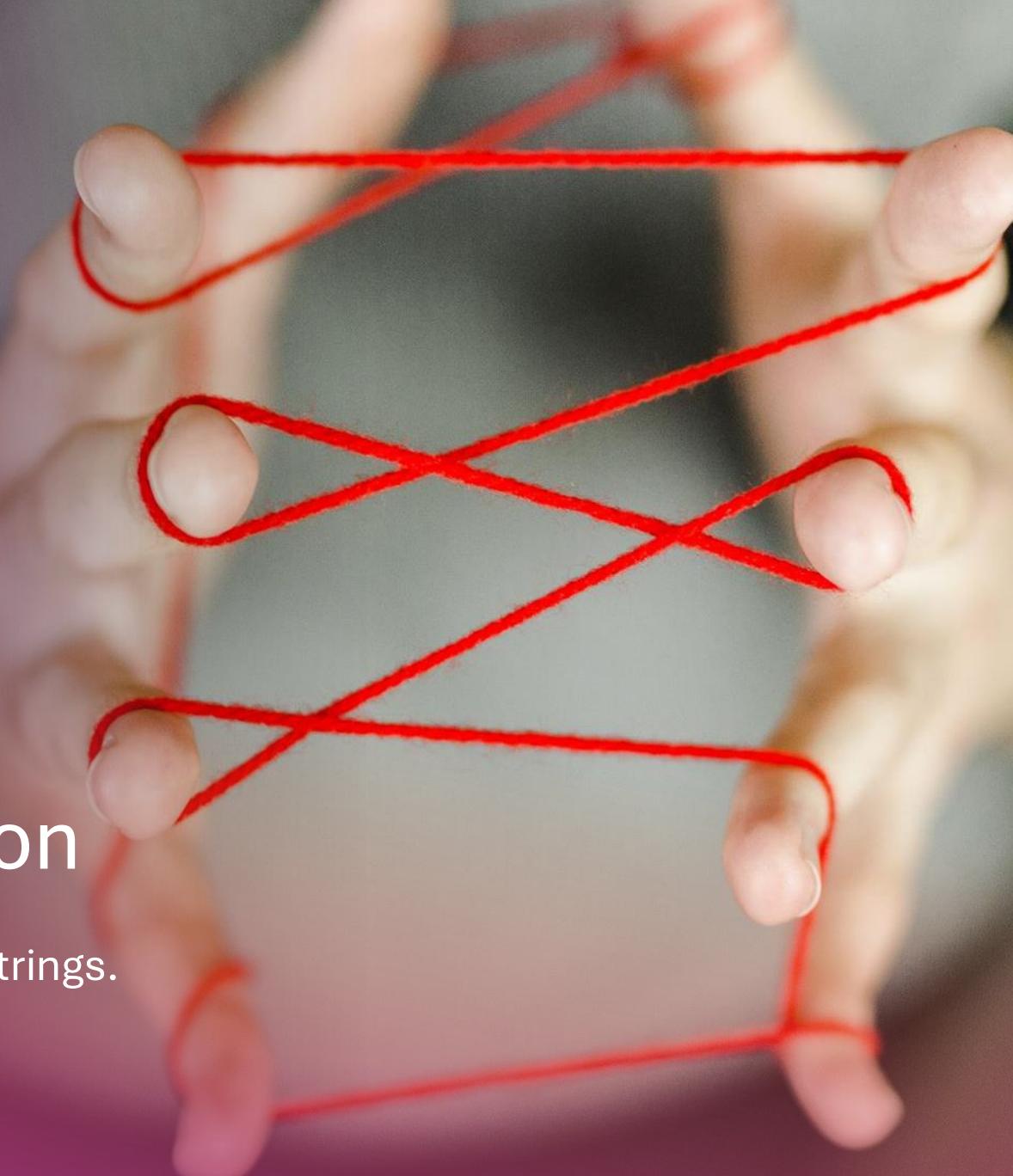


# String Handling Functions

- These can only be used for strings.
- You can force some other data types to be strings, then use them.
- Concatenation.
- Length.
- Position.
- String conversion: integer/float to string or string to int/float.

# Concatenation

This allows you to join strings.



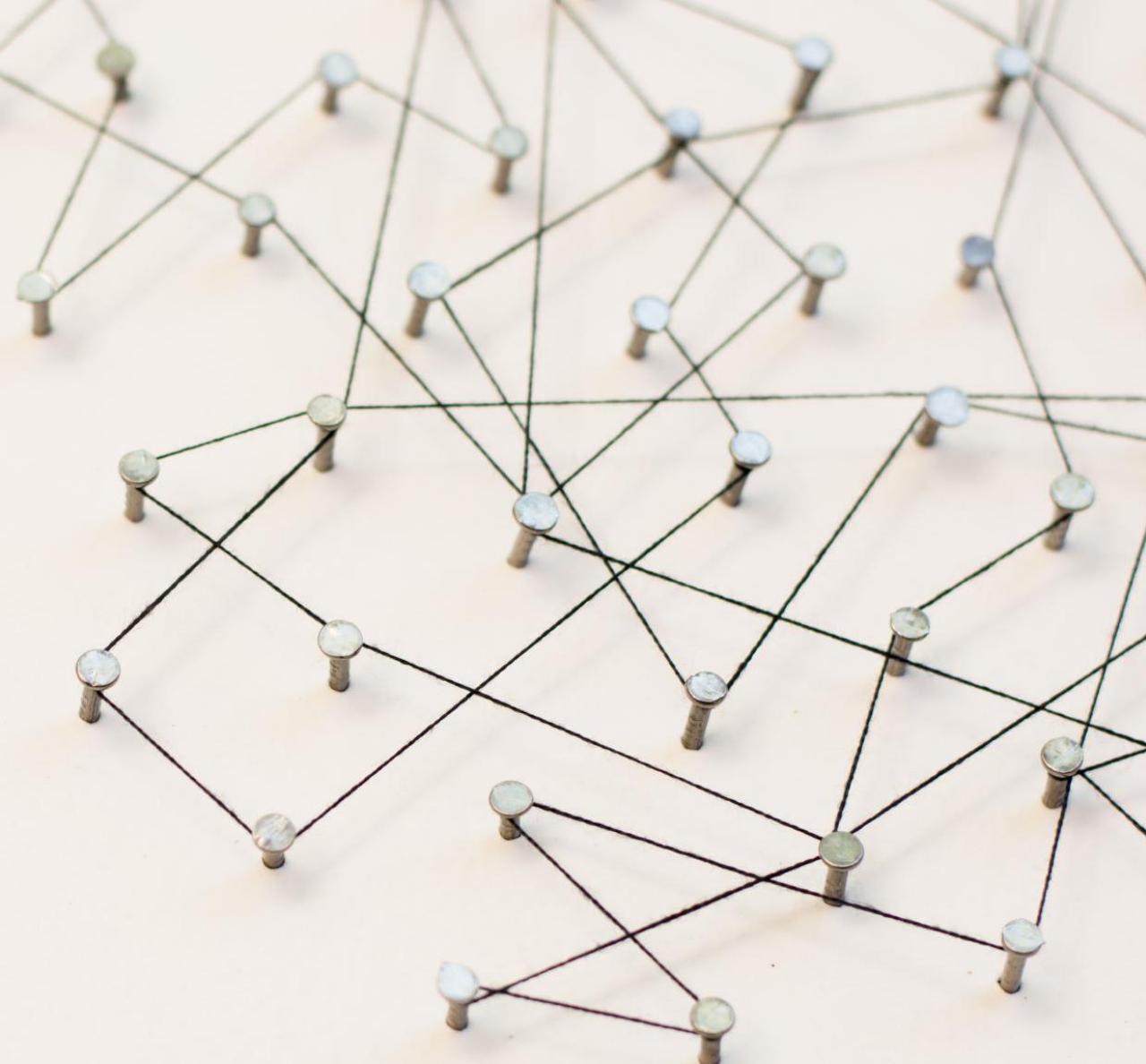
# String Concatenation

- # String Concatenation Python 3 Example

```
Using the + operator:
greeting = "Hello"
name = "Alice"
message = greeting + ", " + name + "!"
print(message) # Output: Hello, Alice!
```

# Length

- This is used to check how many characters there are in a string.
- Space and punctuation are counted.



# String Length Example

- # String Length Example

```
name = "RonsTechHub! "
print(len(name))
```

# Position



This tell you the location of a character in a string.



In Python count always starts from 0, it will always show one less than the actual count (zero index).

# String Position Example

- # String Length Example

```
name = "RonsTechHub! "

location = 0
for letters in name:
 print("Location:", location, "Letter:",letters)
 location=location+1
```



# String Conversion

- Sometimes you might need to convert a string to an integer or float.
- Or, you might need to convert an integer or float to a string.

# Integer To String

- # Integer To String

```
age = 51
```

```
age_2 = str(age)
```

```
print("Age Type:",type(age))
print("Age2 Type:",type(age_2))
```

# Float To String

- # Integer To String

```
height = 1.00157
```

```
height_2 = str(height)
```

```
print("Age Type:",type(height))
print("Age2 Type:",type(height_2))
```

# String To Integer

- # String To Integer

```
age = "120"
```

```
age_2 = int(age)
```

```
print("Age Type:", type(age))
print("Age2 Type:", type(age_2))
```

# String To Float

- # String To Integer

```
height = "2.0453"
```

```
height_2 = float(height)
```

```
print("Age Type:", type(height))
print("Age2 Type:", type(height_2))
```

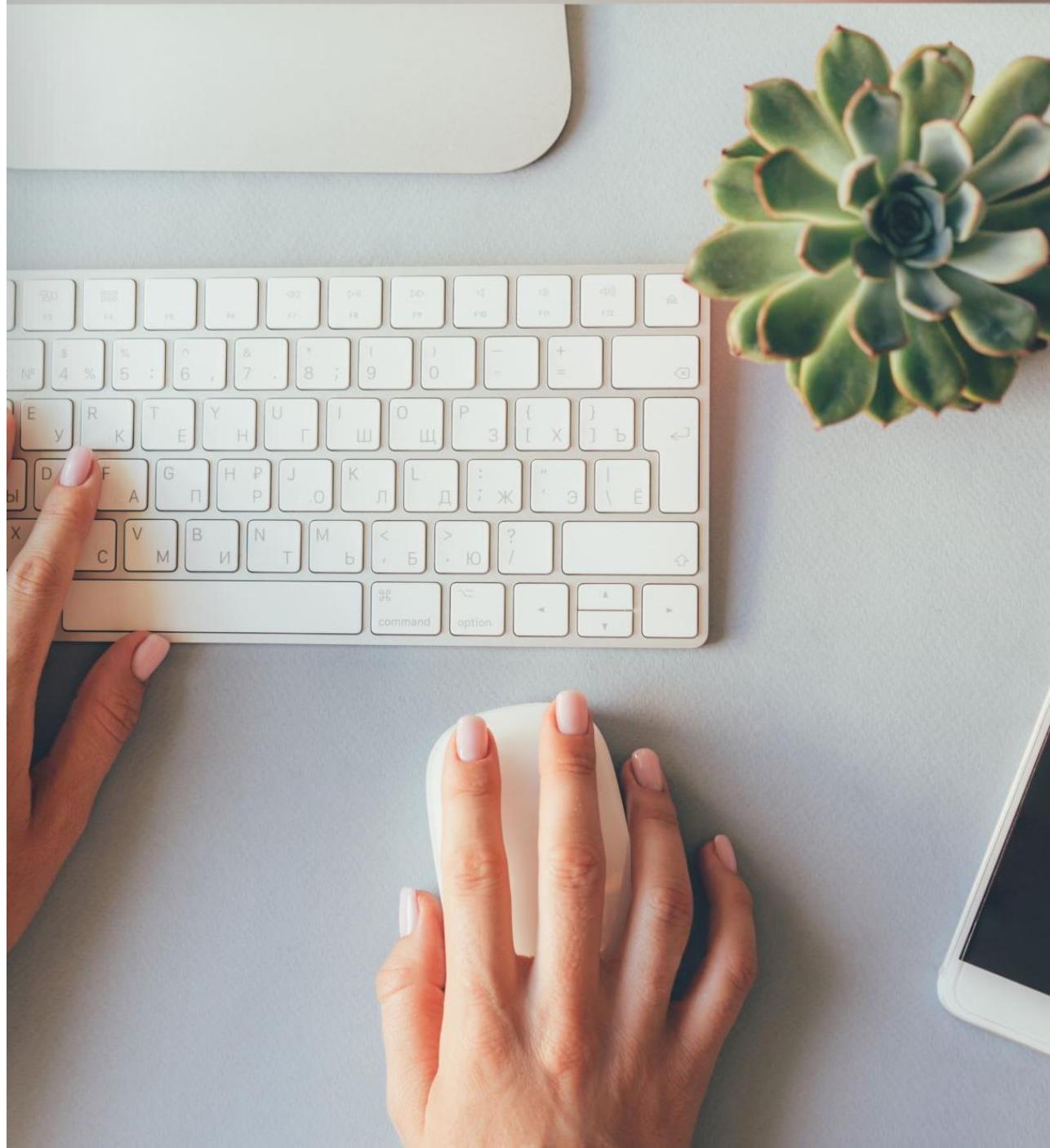


# General Functions

- These are some of the most popular functions used. This is for Python 3.
- Input.
- Open.
- Print (already shown).
- Range (already shown).

# Input

- This is one method which can be used to get input from a user using the keyboard.



# Input Function

- `name = input("What is your name?")`  
`print(name)`

- `age = input("How old are you?")`  
`print(age)`



# Open Function

- Used to open files, typically these will be "txt" and "csv" files in Python.
- Open "**can**" also create the file if it does not exist.

# Open Function Example

- `try:`  
    `with open("my_text_file.txt", "r") as file:`  
        `content = file.read() # Read the whole file`  
        `print(content)`  
    `except FileNotFoundError:`  
        `print("File not found.")`



# Next Time

Validating Data