Welcome: 12 Hours Django Boot-Camp 2024

Boot-Camp Host

Rashedul Alam Shakil
Founder of Study Mart
Founder of aiQuest Intelligence
Automation & Database Programmer,
Siemens, Germany

Instructor

Abu Noman Basar
Software Engineer (Python),
eAppair Limited
Django Instructor at aiQuest Intelligence

Welcome: 12 Hours Django Boot-Camp 2024

Assistant Tutor

MD Mohaiminul Islam Imran
Django Developer

Assistant Tutor

Md Nuruzzaman
Django Developer

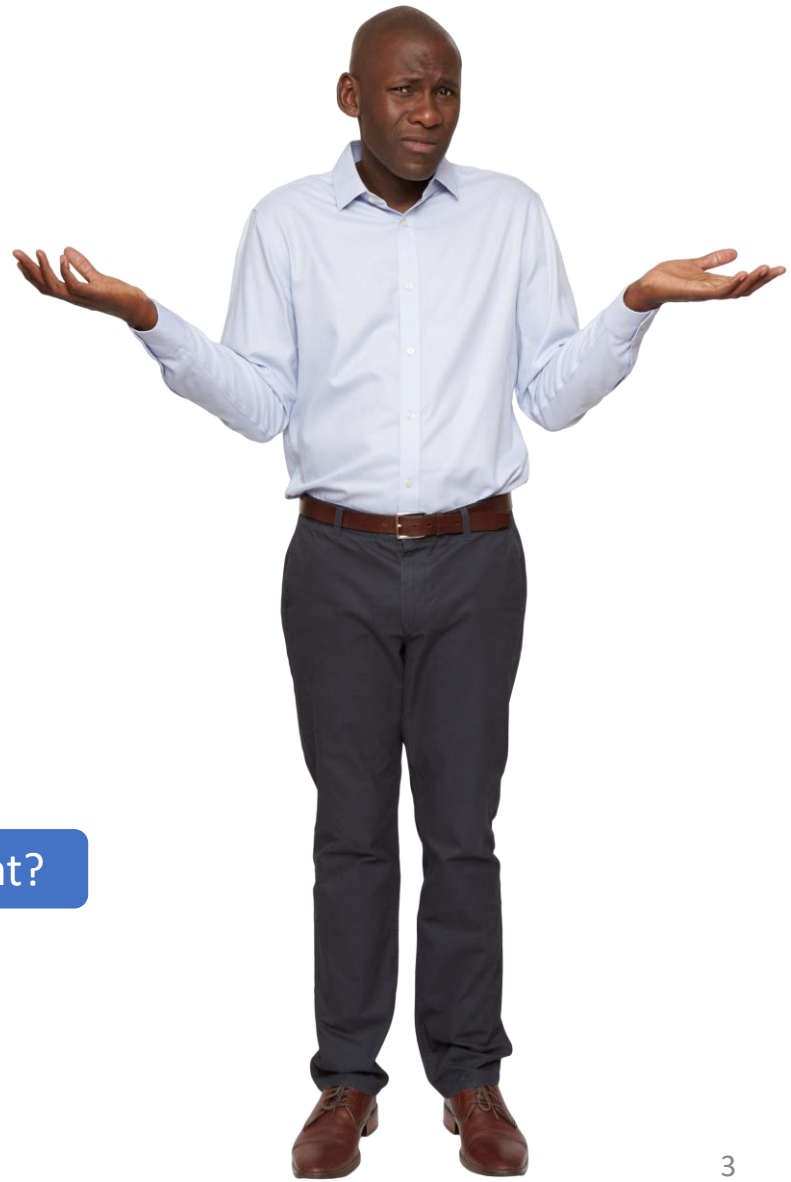
Assistant Tutor

Md Harun-Or-Rashid
Django Developer

12 Hours Django Boot-Camp 2024

Questions on Mind

- What is Python?
- What is Python Django?
- What is Frameworks?
- What is Web Development?
- What is Software Development?
- What is Full Stack Development?



Web Design

Web design is the process of **creating websites** that are both visually appealing and user-friendly. It involves several different disciplines, including:

- **User interface (UI) design:** This is the look and feel of the website, including the layout, colors, fonts, and images.
- **User experience (UX) design:** This is how easy it is for users to navigate the website and find the information they're looking for.

Web Development

WEB DEVELOPMENT

Web Development

Building websites and web apps is like writing instructions to make a house on the internet, with code as the building blocks.

Types of Web Development

Fig: Frontend

Fig: Backend

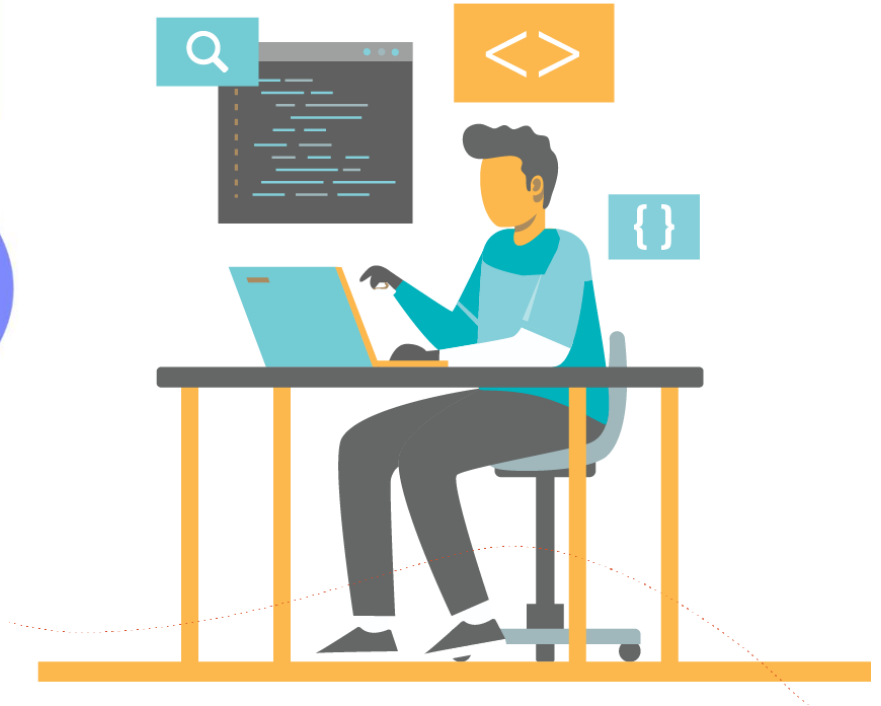


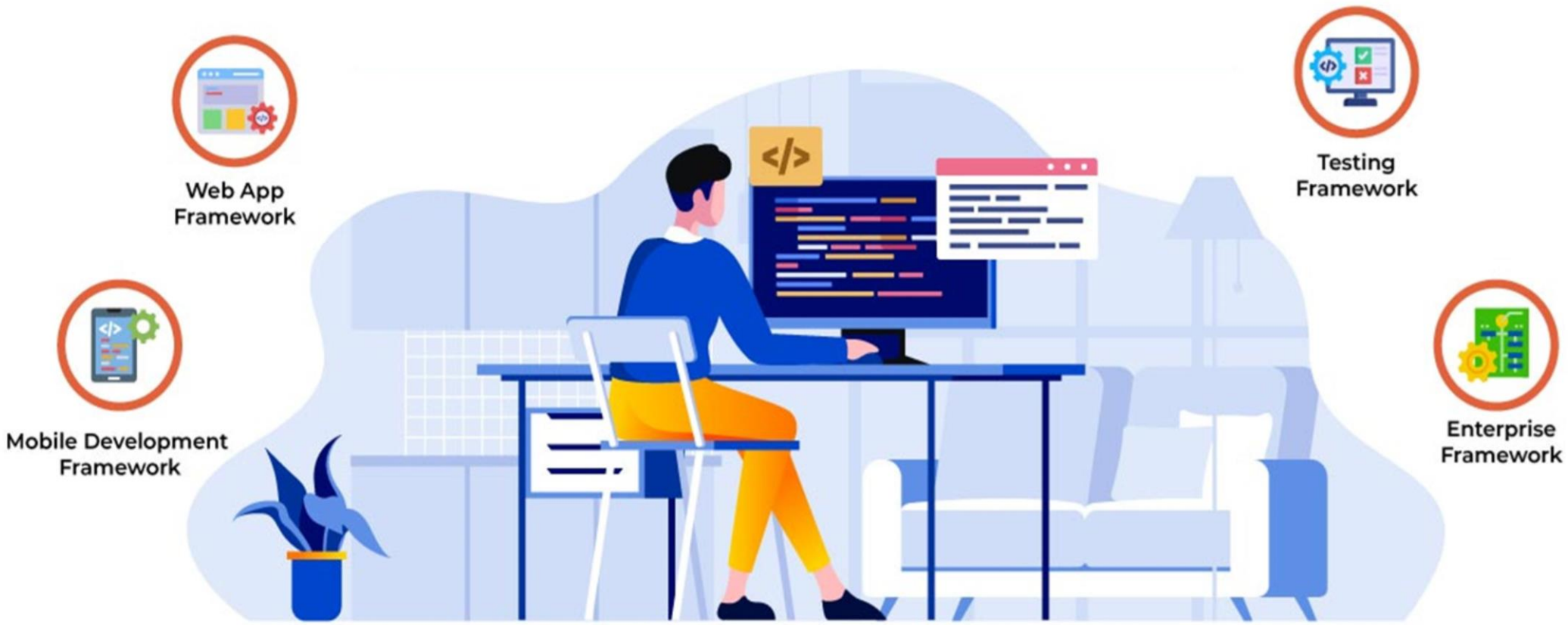
Fig: Full Stack

What is Frameworks

Fig: Complete Cake

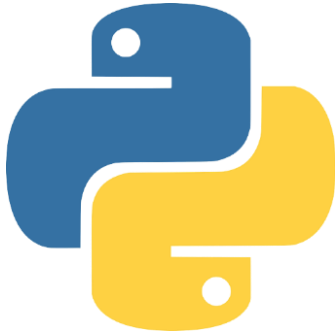
Fig: Frame to Make A Cake

Type of Frameworks



Web Development Framework

Django is a popular **free and open-source** web framework written in **Python**. It emphasizes speed, security, and reusability of code.



- **Fast development:** Get things done quicker with built-in features.
- **Secure:** Built-in protections keep your web application safe.
- **Scalable:** Grows with your project, handling small sites or large user bases.
- **Easy to learn:** Good structure makes it beginner-friendly.
- **Large community:** Plenty of resources and help available.

Let’s Talk About Python

Required Software

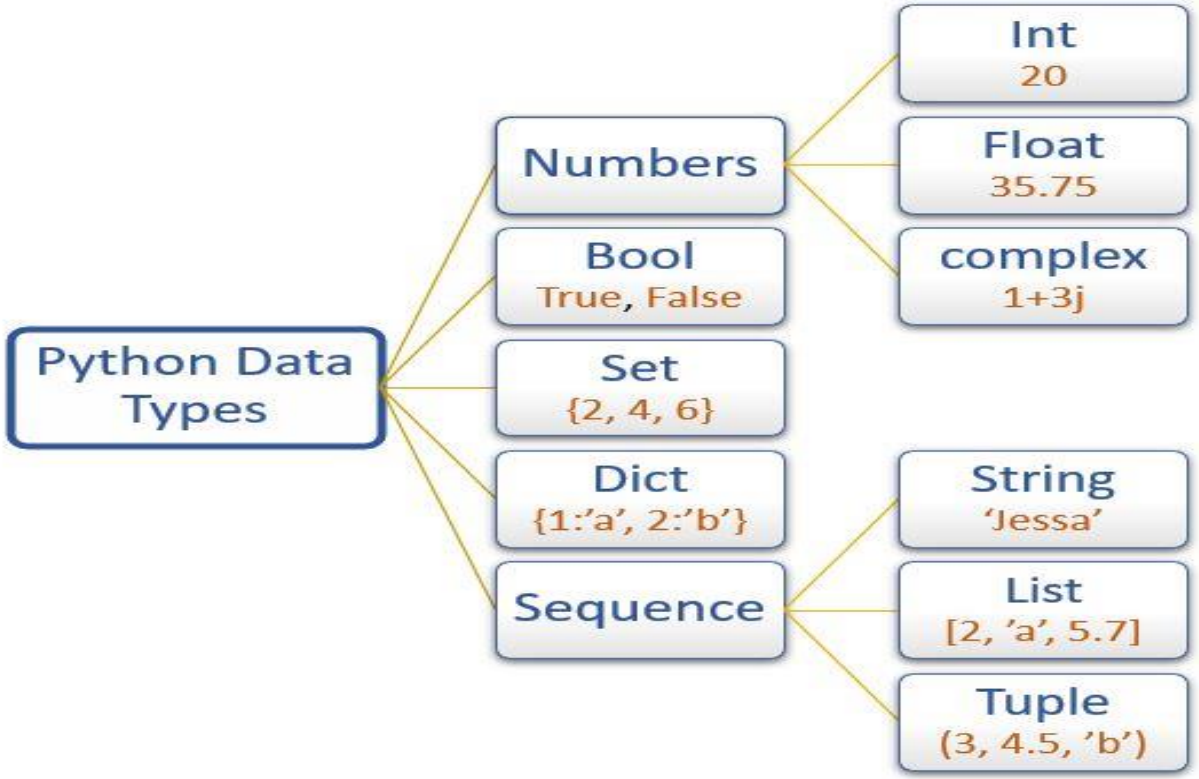
- Python 3 (Latest Version)
- VS Code
- PyCharm

Python I/O

- We use the print() function to output data to the standard output device (screen).
  - print('Hello World!')

- The input() method reads a line from input, converts it into a string, and returns it.
  - input('Enter anything ')

Python Data Types



Python Variables

- Variables are like a container for **storing data**.
- Compared to other programming languages, Python has **no command** for declaring a variable.
- A variable is created the moment you first assign a value to it.

Example:

Var = ‘data science’

Var2 = ‘study mart’

List of Keywords in Python: https://www.programiz.com/python-programming/keyword-list

Python Variables

A variable can have a short name (like x and y) or a more descriptive name.

- Keywords **can't** be used as a variable.
- A variable name **must** start with a letter or the underscore (_) character.
- A variable name **cannot** start with a number.
- A variable name **can only** contain alpha-numeric characters and underscores (A-Z, 0-9, and _).
- Variable names are **case-sensitive** (x, X, _x are three different variables).

Valid Example:

```
Var = 10
Var2 = 100
_var = 20
Var_2 = 10
V1a2r3 = 30
My_name = 'Study Mart'
```

Invalid Example:

```
9Var = 'data science'
Var-2 = 'study mart'
&var = 20
My name = 'Study Mart'
```

Python Variables

Multiple

- Multiple Variables:
  - x, y, z = “Data", “Science", “Smart“ -> Valid
  - x, y, z = “Data", “Science“ -> Invalid
- Comments:
  - Single Line
  - Multiple Line

Python Variables

Local Vs. Global

- Multi Word Variable Name
  - camelCaseVar
  - PascalCaseVar
  - snake_case_var

- Global Variable: Variables that are created outside of a function are known as global variables. Global variables can be used by everyone, both inside of functions and outside.

- Local Variable: Variables that are created inside of a function are known as local variables. local variables can be used inside of the function.

All about Python Strings

X = ‘Data Science’

Y = ’10’

Z = Something

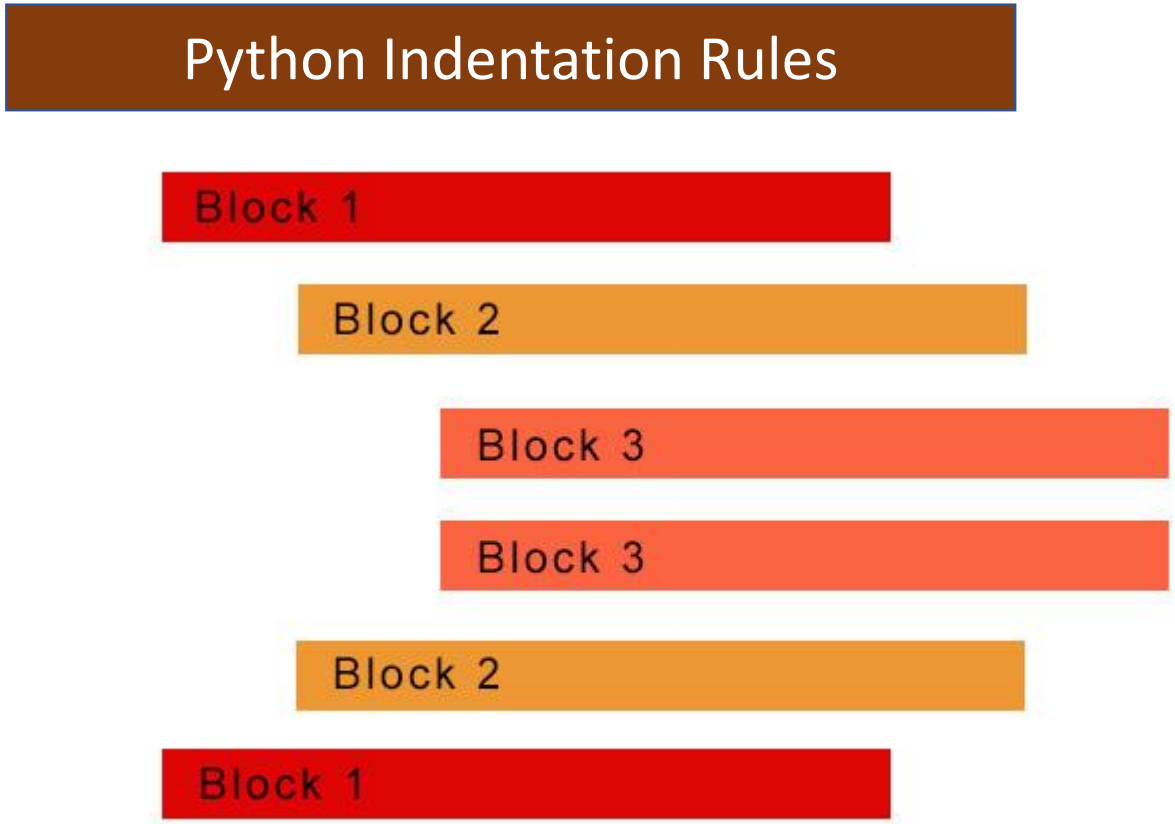
- String Formatting
- String Concatenation
- String methods

Python supports the usual logical conditions from mathematics:

- Equals: a == b
- Not Equals: a != b
- Greater than a > b
- Greater than or equal to a >= b
- Less than a < b
- Less than or equal to a <= b

Conditional Statements

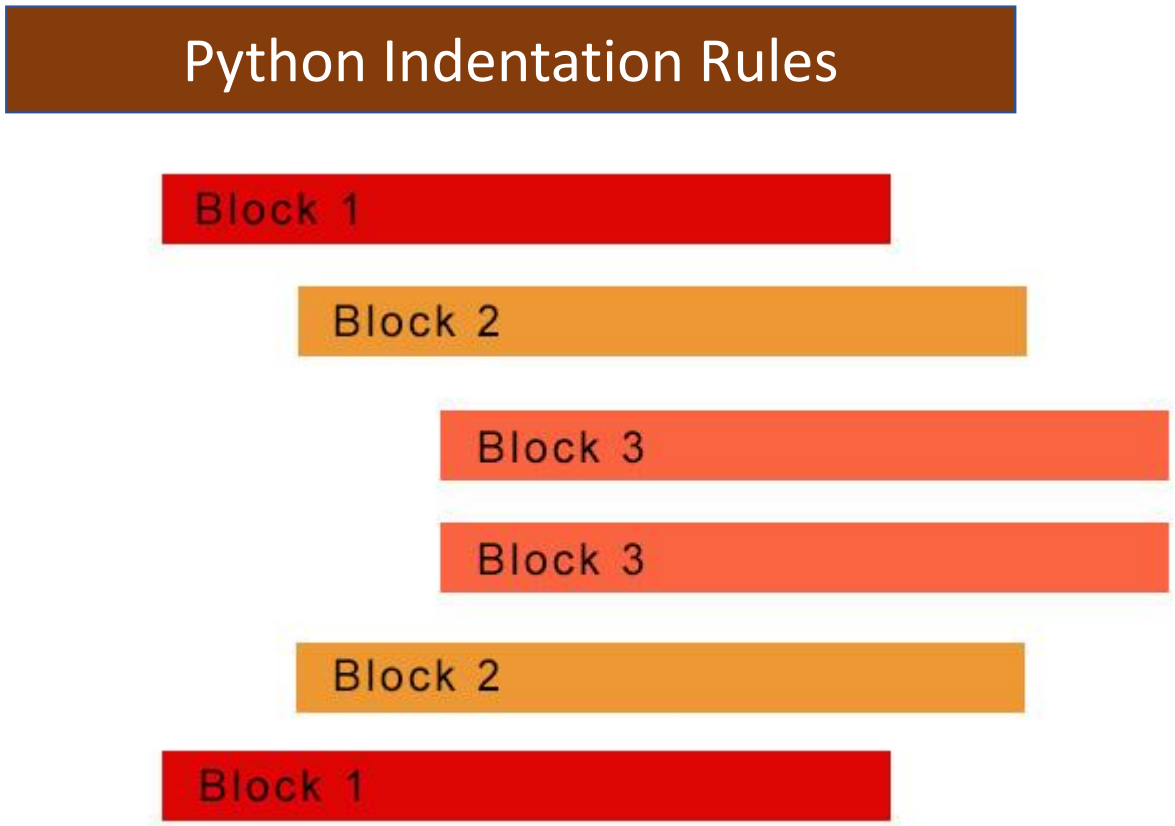
If, else




```
x = 50
y = 100

if y > x:
    print("y is greater than x")

elif x == y:
    print(" x and y are equal")

else:
     print(" x is y greater than y ")
```

Conditional Statements

If, else




```
scores = [85, 92, 78, 60, 45]

for score in scores:
    if score >= 90:
        grade = "A"
    else:
        if score >= 80:
            grade = "B"
        else:
            if score >= 70:
                grade = "C"
            else:
                if score >= 60:
                    grade = "D"
                else:
                    grade = "F“

    print(f"Score: {score}, Grade: {grade}")
```

Example:

```
subjects = [“ai", “data science", “statistics“, ”math”]
for x in  subjects :
            print(x)
```

Output:

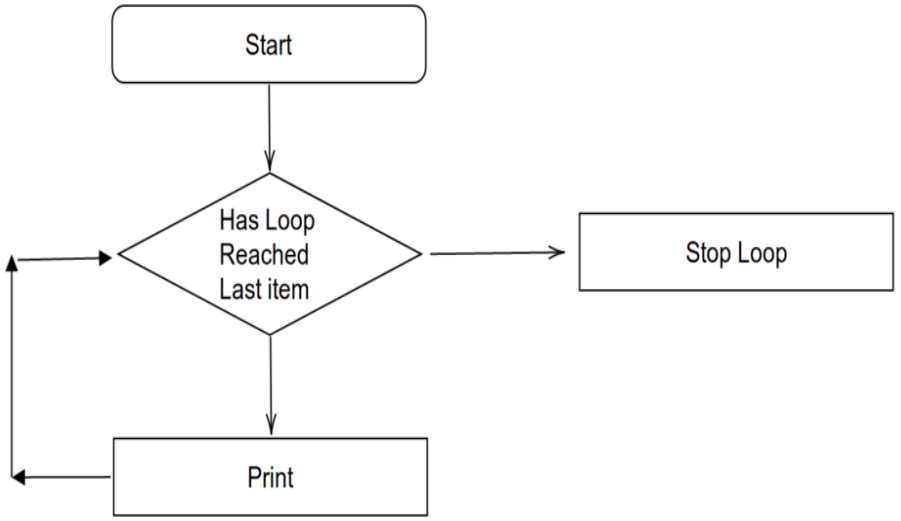
“ai“

“data science“

“statistics“

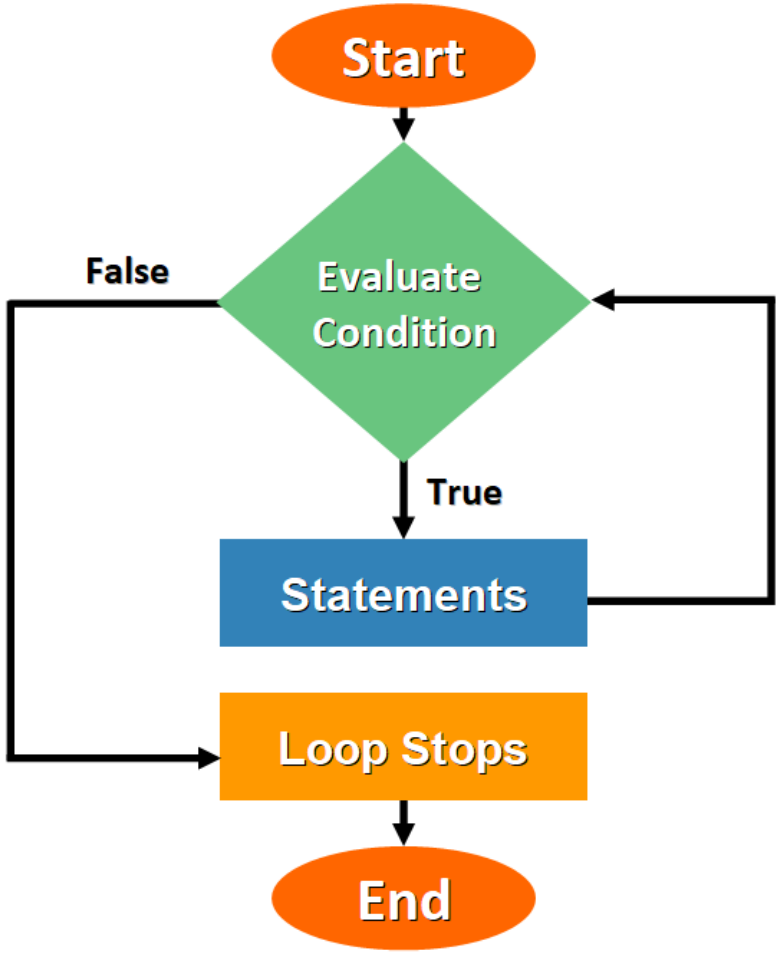
”math”

For Loop



Example:

```
counter = 1

while counter <= 5:
    print(counter)
    counter += 1
```



Break Statement

Break in Python terminates a loop completely when an external condition is given or not given. Python break is used within the code and is usually placed after an “if” statement.

A break statement can only be used inside a loop. This is because the purpose of a break statement is to stop a loop. You can use a break statement inside an if statement, but only if that if statement is inside a loop. The syntax of the break statement is: The syntax of the break statement is:

```
for val in sequence:
    # code
    if condition:
        break

    # code
```

```
while condition:
    # code
    if condition:
        break

    # code
```

Continue Statement

The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration. For example-

```
for val in sequence:
    # code
    if condition:
        continue
```

```
while condition:
    # code
    if condition:
        continue
```

Data Structures: List

- Ordered
- Changeable
- Allow Duplicates

L1 = ['data', 'science']
L2 = [1, 40, 300, 'Python', True, False]

Data Structures: Dictionary

- Ordered
- Changeable
- Does not Allow Duplicates

D1 = { "brand": “Apple", "model": “13 Pro Max", "year": 2022}

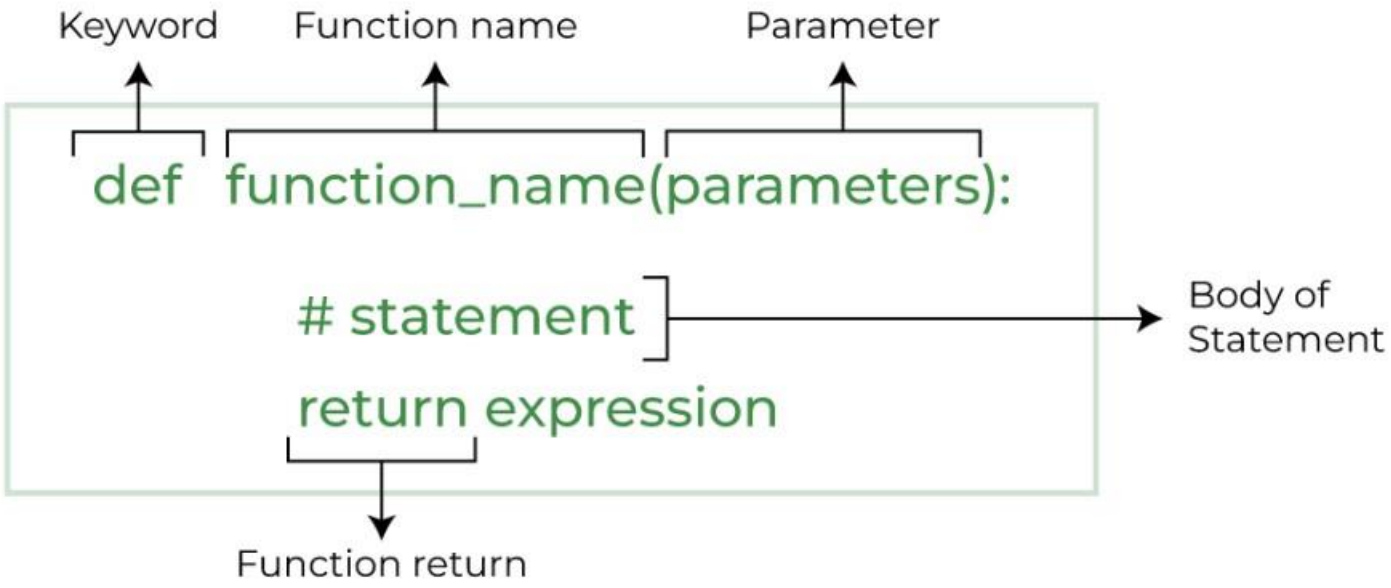
Here, (brand, model, year = id or key) & (Apple, 13 pro max, 2022 = Data)

Python Functions

Python Functions is a block of statements that return the specific task. It runs when it is called.

- In Python a function is defined using the **def** keyword.

- **Syntax:**



❑ Creating a Function:

```
def first_function():
    print('Welcome to python function lecture.')
```

❑ Calling a Function:

```
def first_function():
    print('Welcome to python function lecture.')

first_function()
```

Object-oriented Programming (OOP)

Python OOP relies on **four** main pillars:

- **Encapsulation:** Bundles data and functions together for protection and organization.
- **Inheritance:** Reuses code by creating new classes based on existing ones.
- **Polymorphism:** Allows objects to respond differently to the same call, creating flexibility.
- **Abstraction:** Hides complexity and focuses on essential details for easier use.

OOP: Inheritance

Inheritance allows us to define a class that inherits all the methods and properties from another class.

- **Parent class** is the class being inherited from, also called base class.
- **Child class** is the class that inherits from another class, also called derived class.