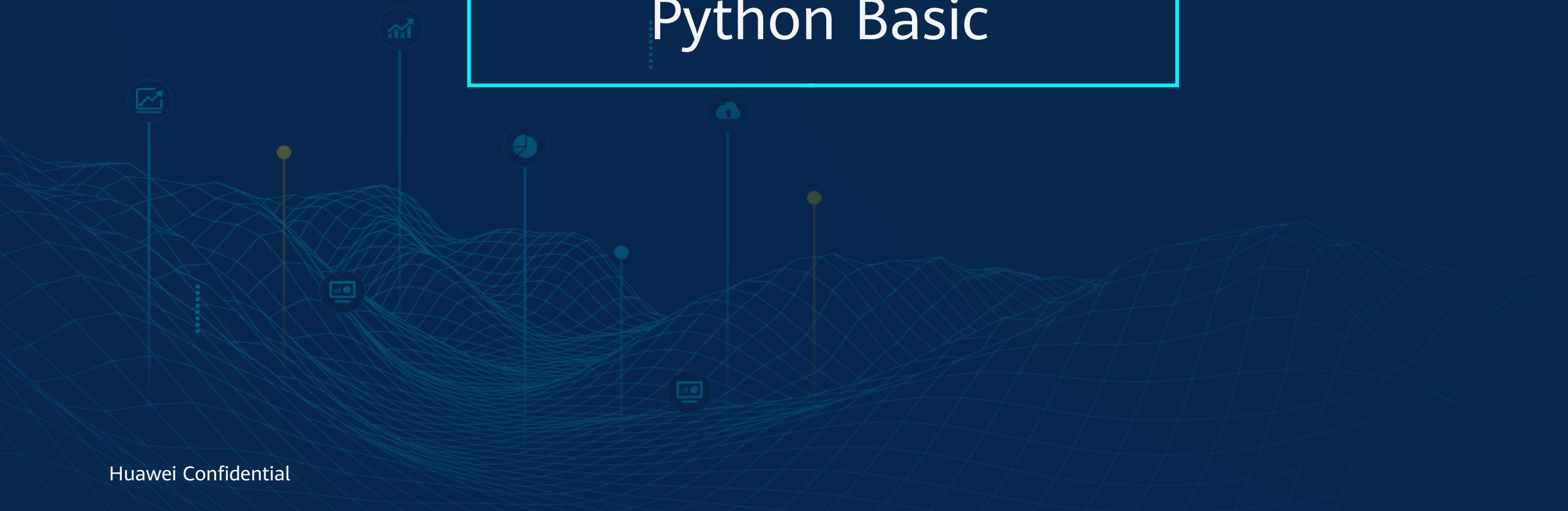
Python Programming Basic

Python Basic



Foreword

In this course, we are going to learn the basics of Python programming.

Objectives

Upon finishing the course, you will be able to :

- Know some Python basic syntax
- Know some Python data types
- Know python control flow statement
- Master the creation of python functions
- Know python object oriented programming

Contents

Python Basics

- Hello world
- Comments
- Libraries and modules
- Variables
- Naming rules and keywords
- Switching values
- Indentation
- Standard I/O

Contents

Python Data Types

- Numeric: int, float, complex
- Text: str
- Sequence: list, tuple, range
- Mapping: dict
- Set: set, frozenset
- Boolean: bool
- Binary: bytes, bytearray, memoryview

Python vs C

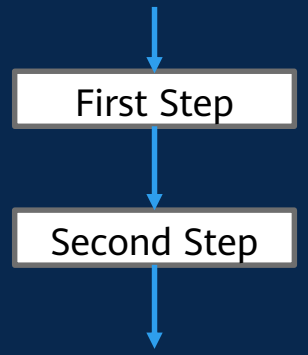
- Deep copy and Shallow copy
- Operators

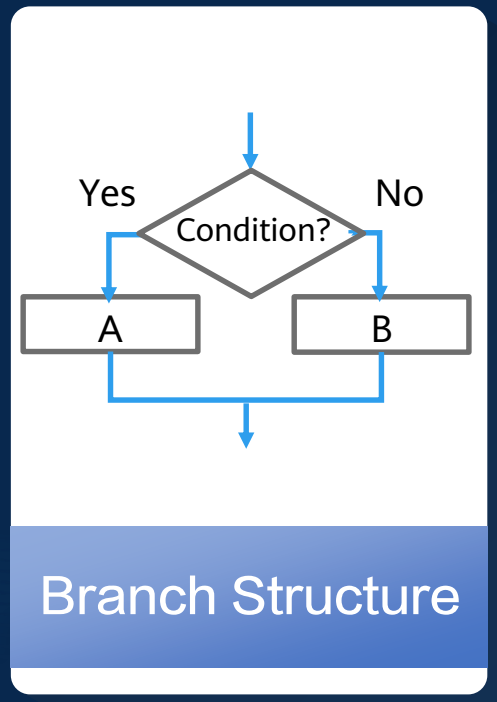
Contents

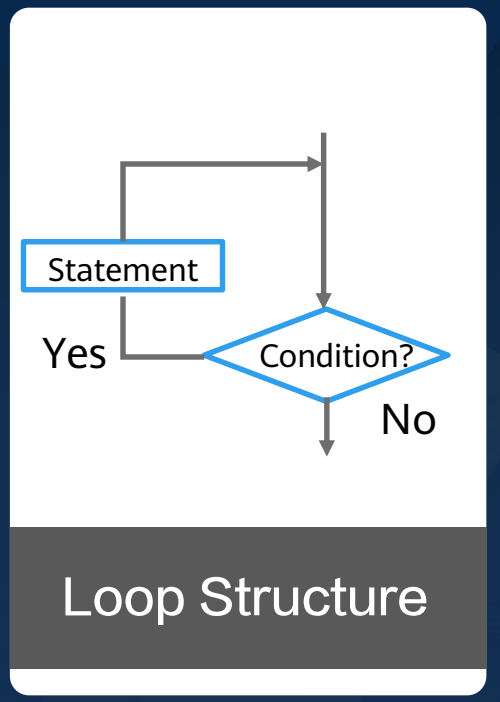
What is Control Flow?

The control flow of a program is the execution order of the program code.

There are three basic control flow structures in Python.



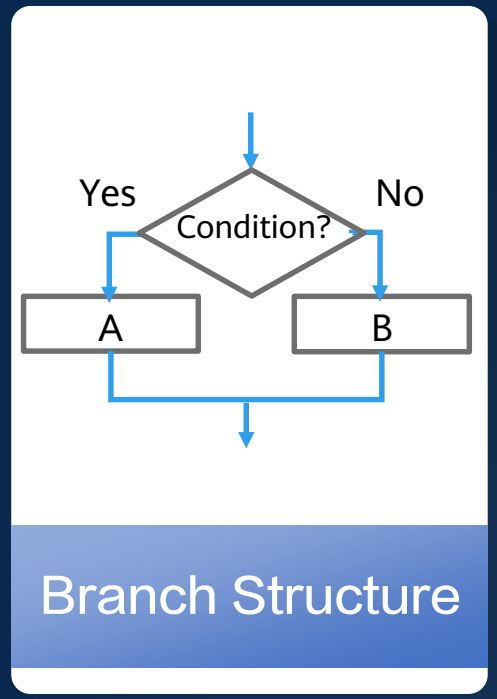




Conditional Statement

A conditional statement is features of a programming language, which perform different computations or actions depending on whether a programmer-specified Boolean condition evaluates to true or false.

In Python, we use the if-elif-else statement to express conditional statement.

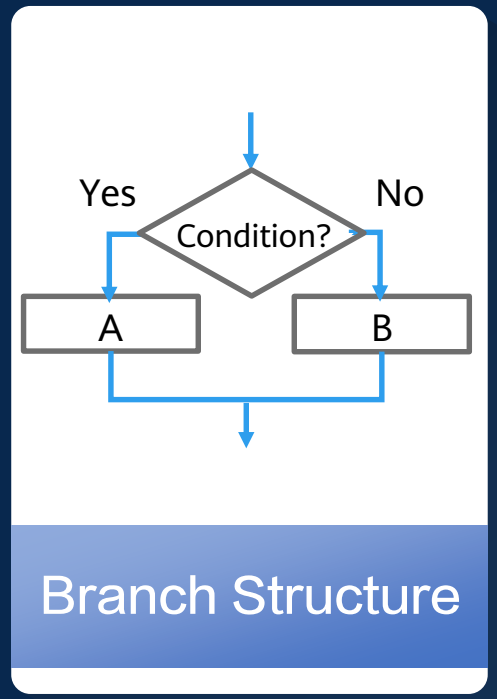


Loops

Python programming language provides following types of loops to handle looping requirements. Python provides three ways for executing the loops.

- While Loop
- For Loop
- Nested Loops

While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.



Contents

Functions in Python

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

```
In [16]: def myFunction(a, b):
             c = a*b
             c = c+1
             return c

In [17]: myFunction(10,10)

Out[17]: 101
```

Higher-order Function

A function is called Higher Order Function if it contains other functions as a parameter or returns a function as an output.

Apart from higher-order functions, all other functions are first-order functions.

```
In [32]: def loud(text):
             return text.upper()+"!"

         def quiet(text):
             return text.lower() +"..."

         def speak(func, text):
             words = func(text)
             print(words)

         speak(loud, "Python is Cool")
         speak(quiet, "Python is Cool")

         PYTHON IS COOL!
         python is cool...
```

Anonymous Function

An anonymous function is a function that is not bound to an identifier.

Anonymous functions are often arguments being passed to higher-order functions, or used for constructing the result of a higher-order function that needs to return a function.

```
oldList = [1, 2, 3, 4, 5]

newList = list(filter(lambda x: (x%2 == 0) , oldList))

print(newList)
```

```
[2, 4]
```

Contents

What is Object Oriented Programming ?

Object-oriented programming is a computer programming model that organizes software design around objects, rather than functions and logic.

What is Object Oriented Programming ?

Bellow is a comparison of object oriented programming and traditional function and logic programming.

```
In [10]: class Employee:
           def __init__(self, name, age):
             self.name = name
             self.age = age

         e1 = Employee("Kevin", 63)

         print(e1.name)
         print(e1.age)
```

```
Kevin
63
```

main.c

```c
#include <stdio.h>
char name[50][100];
int age[100];


void setName(int a, char *n) {
    int i = 0;
    while(*n) {
        name[a][i]=*n;
        i++;
        n++;
    }
}

void setAge(int i, int n) {
    age[i]=n;
}


int main() {
    setName(0, "Kevin");
    setAge(0, 63);
    printf("%s", name[0]);
    printf("\n");
    printf("%d", age[0]);
    return 0;
}
```

```
63
```

Class and Object

An object is a collection of data (variables) and methods (functions) that act on those data.

A class is a blueprint for an object.

An object is also called an instance of a class and the process of creating this object is called instantiation.

We will mainly introduce the following concepts of class and object:

- Inheritance
- Abstraction
- Polymorphism

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.

Abstraction

An abstract class can be considered as a blueprint for other classes.

A class which contains one or more abstract methods is called an abstract class.

We can create a set of methods that must be created within any child classes built from the abstract class.

Polymorphism

Objects can take on more than one form depending on the context.

The program will determine which meaning or usage is necessary for each execution of that object.

Summary

This chapter introduces some Python basic syntax, data types, control flow statements, functions and object oriented programming.

More Information

Online learning website

- https://e.huawei.com/en/talent/#/home

Huawei Knowledge Base

- https://support.huawei.com/enterprise/en/knowledge?lang=en

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI