INFIQUE AI

Training | Development | Research

Python Programming Syllabus

Python - Fundamentals

- Installation
- Python – Syntax
- Python – Variables and Datatypes
- Python – Numbers Strings
- Sequences
- List
- Tuples
- Ranges
- Dictionary
- String
- Sets
- Operators
- If. Else. Statements
- For Loop
- While Loop
- Break
- Continue
- Pass
- Date & Time
- Functions
- Package
- modules
- Reading a File
- Writing into File
- Python Exceptions
- Regular Exp Mathematics

List & Tuples

List & Operation

- Append
- Clear
- Copy
- Count
- Extend
- Index
- Insert
- Pop
- Remove
- Reverse

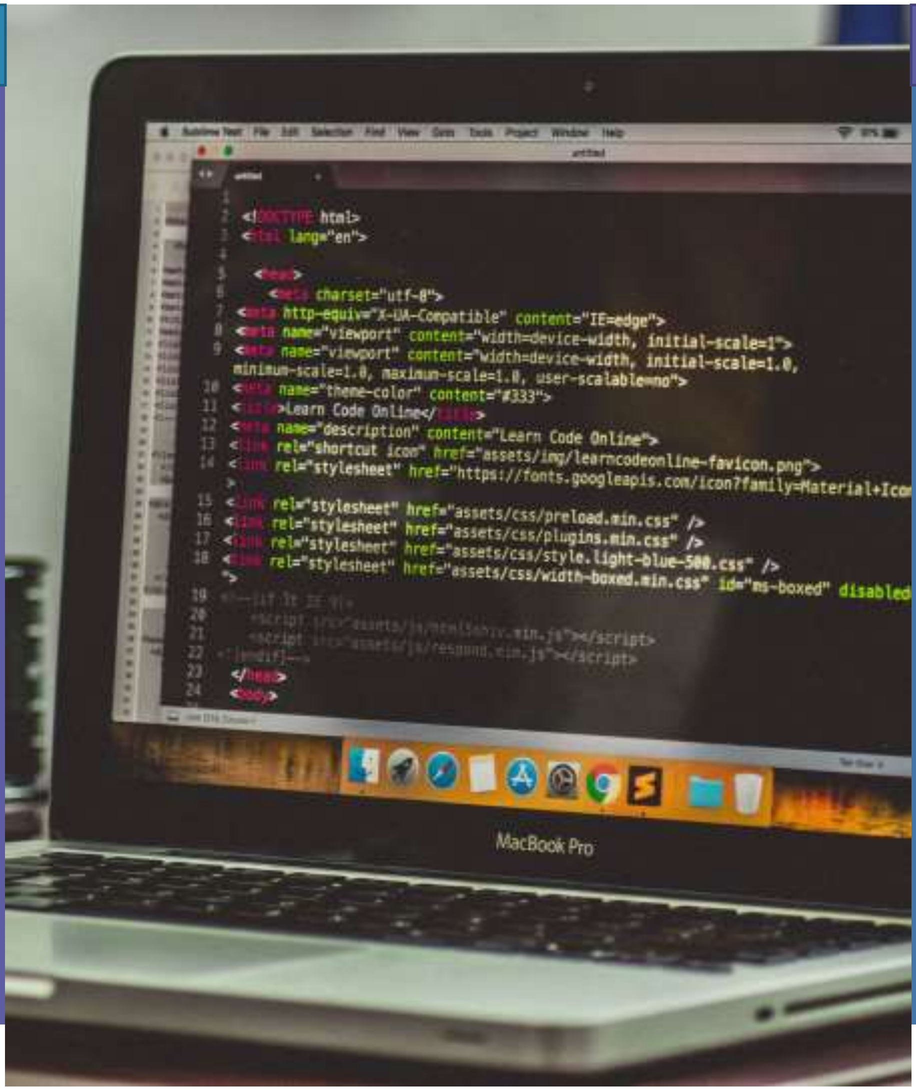
Tuples

- Count
- Index

Dictionary & String

Dictionary

- Clear
- Copy
- From-keys
- Get
- Items
- Keys
- Pop
- Pop-item
- Set-default
- Update



```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">
<meta name="theme-color" content="#333">
<title>Learn Code Online</title>
<meta name="description" content="Learn Code Online">
<link rel="shortcut icon" href="assets/img/learncodeonline-favicon.png">
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icon
>
<link rel="stylesheet" href="assets/css/preload.min.css" />
<link rel="stylesheet" href="assets/css/plugins.min.css" />
<link rel="stylesheet" href="assets/css/style.light-blue-500.css" />
<link rel="stylesheet" href="assets/css/width-boxed.min.css" id="ms-boxed" disabled
">
<!--[if lt IE 9]>
    <script src="assets/js/html5shiv.min.js"></script>
    <script src="assets/js/respond.min.js"></script>
<![endif]-->
</head>
<body>
```

String

- Find
- R-index
- R-partition
- R-split
- R-strip
- Starts-with
- Strip
- Title
- Z-fill

Conditional & If Statement

Conditional statements are **used through the various programming languages to instruct the computer on the decision to make when given some conditions**. These decisions are made if and only if the pre-stated conditions are either true or false , depending on the functions the programmer has in mind

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

•if statement.
•if-else statement.
•if- elif-else ladder.

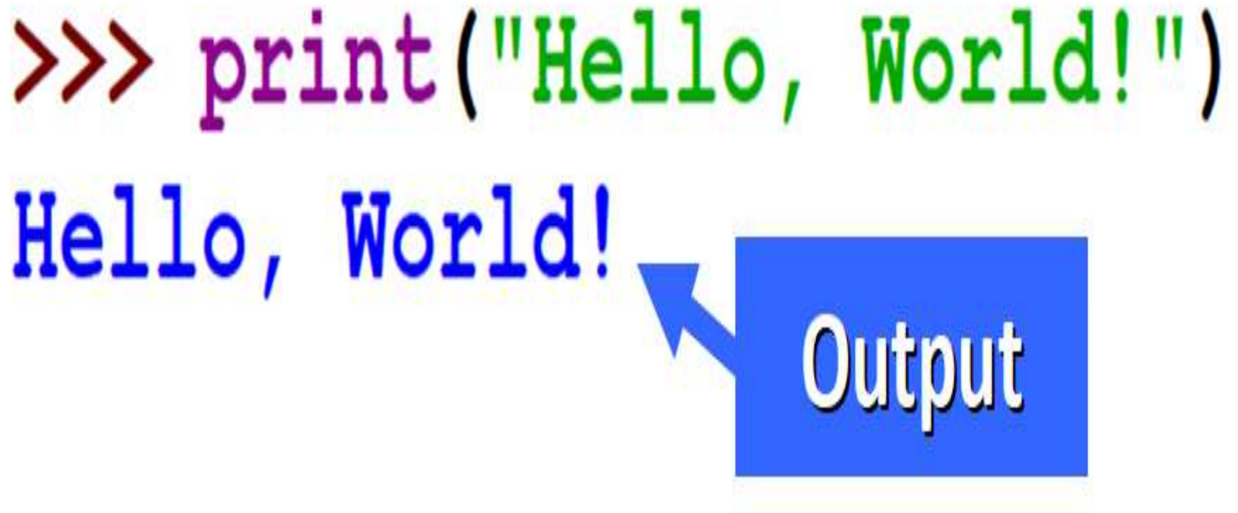
Functions

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.

- Creating a Function
- Calling a Function
- Arguments
- Parameters or Arguments
- Number of Arguments
- *args & **kwargs
- Default Parameter Value
- Python Lambda
- Syntax
- Why Use Lambda Functions



```
>>> print("Hello, World!")
Hello, World!
```

Python First Program

Loop

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

For Loop

- The break Statement
- The continue Statement
- The range() Function
- The pass Statement
- Nested Loops
- Else in For Loop

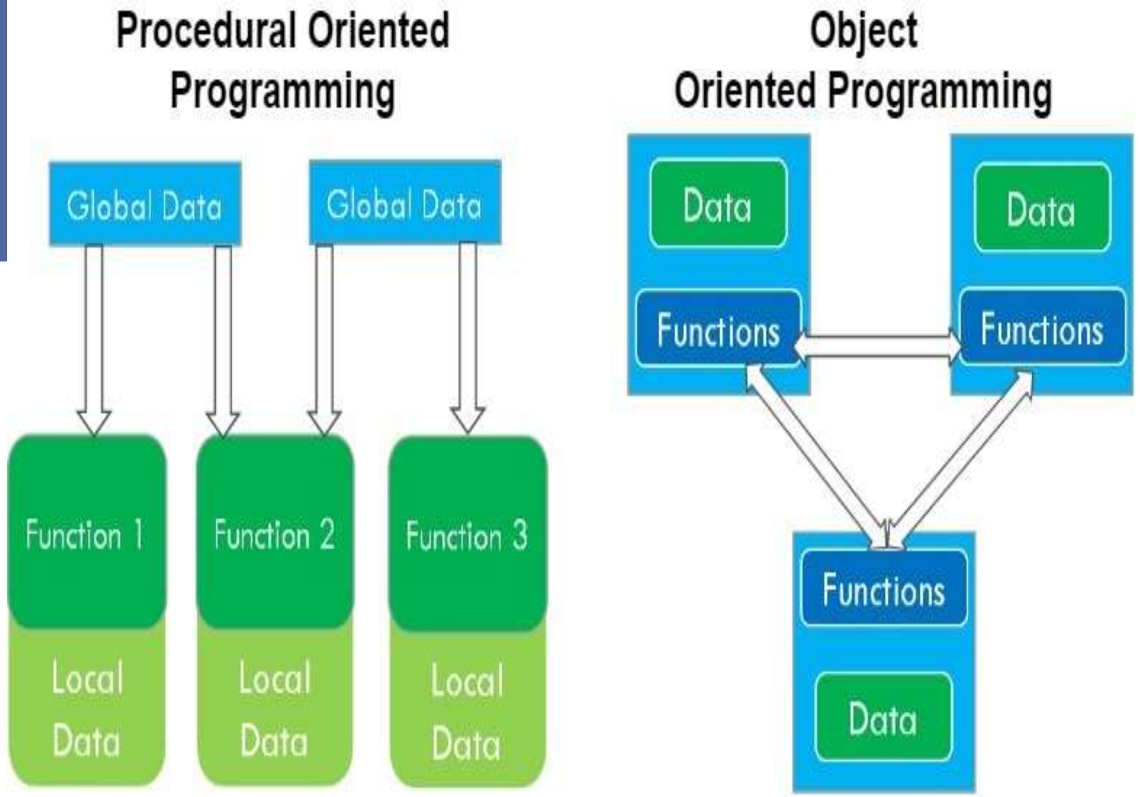
While Loop

- The break Statement
- The continue Statement
- The else Statement

Oops

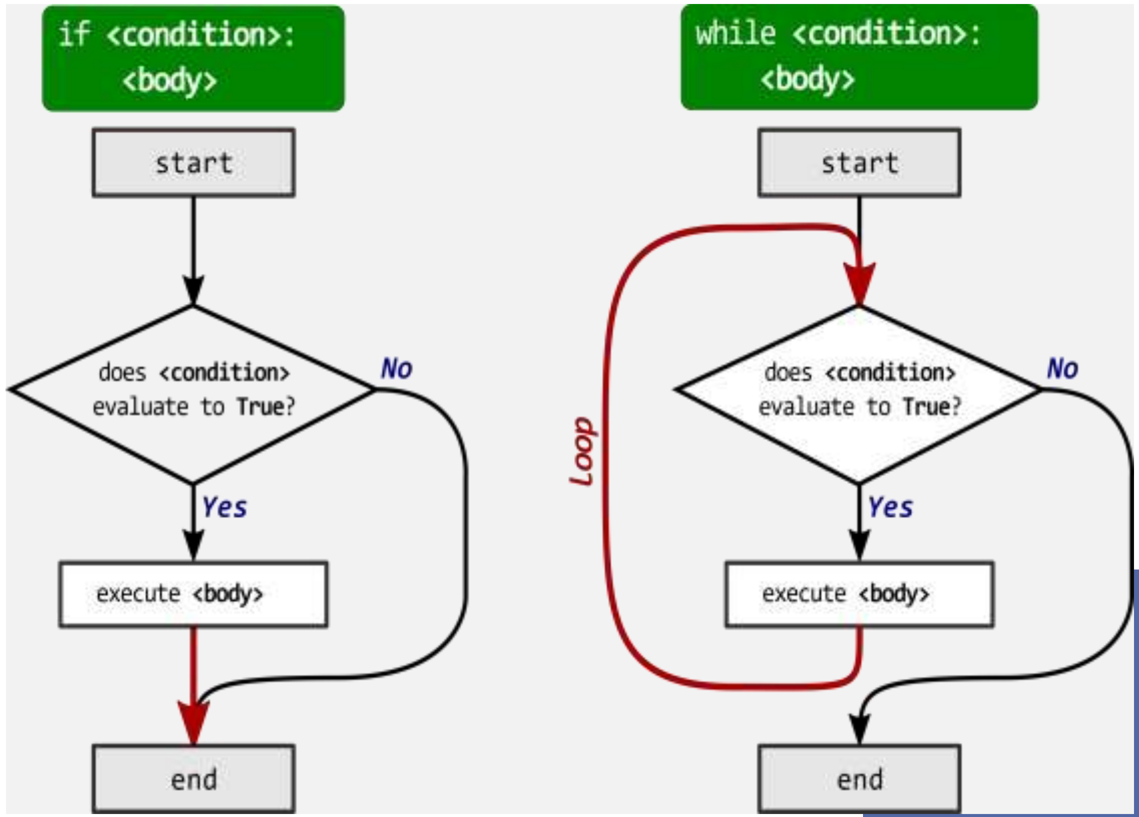
Object Oriented Programming is a way of computer programming using the idea of “objects” to represents data and methods. It is also, an approach used for creating neat and reusable code instead of a redundant one. the program is divided into self-contained objects or several mini-programs.

- Class.
- Object.
- Method.
- Inheritance.
- Polymorphism.

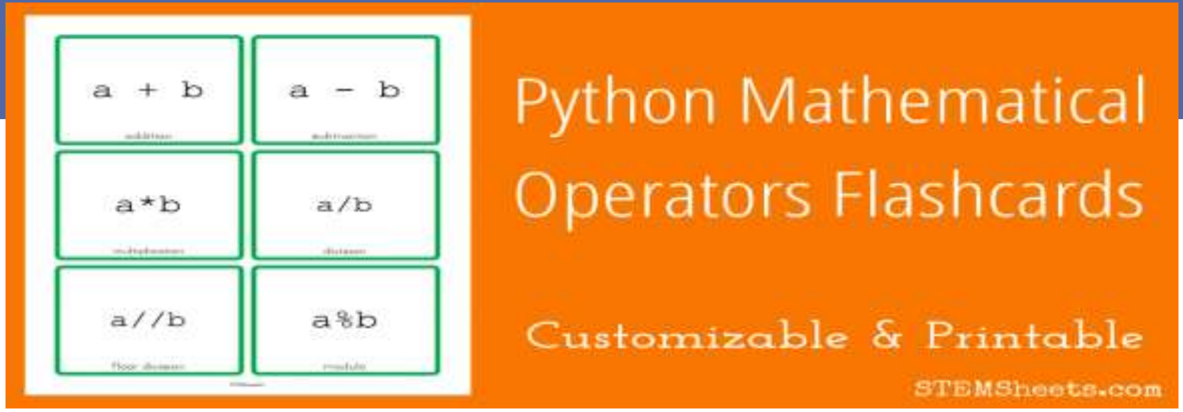


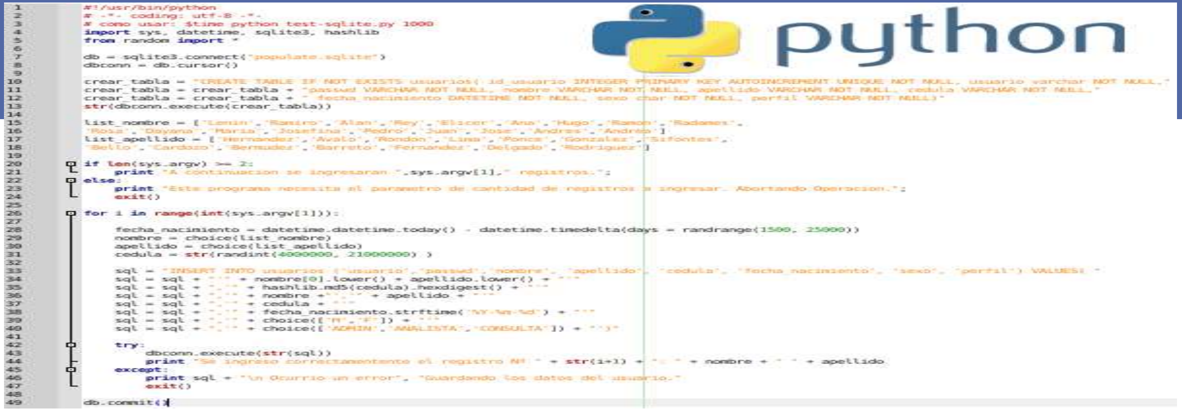
OOPS in Python

Loop in Python



Modules & Package





```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# como usar: $time python test-sqlite.py 1000
import sys, datetime, sqlite3, hashlib
from random import *

db = sqlite3.connect("populate.sqlite")
dbconn = db.cursor()

crear_tabla = "CREATE TABLE IF NOT EXISTS usuarios( id_usuario INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE NOT NULL, usuario varchar NOT NULL,"
crear_tabla = crear_tabla + "passwd VARCHAR NOT NULL, nombre VARCHAR NOT NULL, apellido VARCHAR NOT NULL, cedula VARCHAR NOT NULL,"
crear_tabla = crear_tabla + " fecha_nacimiento DATETIME NOT NULL, sexo char NOT NULL, perfil VARCHAR NOT NULL)"
str(dbconn.execute(crear_tabla))

list_nombre = ['Lenin','Ramiro','Alan','Rey','Elicer','Ana','Hugo','Ramon','Radames',
'Rosa','Dayana','Maria','Josefina','Pedro','Juan','Jose','Andres','Andrea']
list_apellido = ['Hernandez','Avalo','Rondon','Lima','Ponce','Gonzalez','Sifontes',
'Bello','Cardozo','Bermudez','Barreto','Fernandez','Delgado','Rodriguez']

if len(sys.argv) >= 2:
    print "A continuacion se ingresaran ",sys.argv[1]," registros.";
else:
    print "Este programa necesita el parametro de cantidad de registros a ingresar. Abortando Operacion.";
    exit()

for i in range(int(sys.argv[1])):

    fecha_nacimiento = datetime.datetime.today() - datetime.timedelta(days = randrange(1500, 25000))
    nombre = choice(list_nombre)
    apellido = choice(list_apellido)
    cedula = str(randint(4000000, 21000000) )

    sql = "INSERT INTO usuarios ('usuario','passwd','nombre', 'apellido', 'cedula', 'fecha_nacimiento', 'sexo', 'perfil') VALUES( "
    sql = sql + "'" + nombre[0].lower() + apellido.lower() + "'"
    sql = sql + "','" + hashlib.md5(cedula).hexdigest() + "'"
    sql = sql + "','" + nombre +"','" + apellido + "'"
    sql = sql + "','" + cedula + "'"
    sql = sql + "','" + fecha_nacimiento.strftime('%Y-%m-%d') + "'"
    sql = sql + "','" + choice(['M','F']) + "'"
    sql = sql + "','" + choice(['ADMIN','ANALISTA','CONSULTA']) + "')"

    try:
        dbconn.execute(str(sql))
        print "Se ingreso correctamentento el registro Nº " + str(i+1) + ". " + nombre + " " + apellido
    except:
        print sql + "\n Ocurrio un error", "Guardando los datos del usuario."
        exit()

db.commit()
```

Modules

- What is a Module
- How to Create a Module
- What Use a Module

Package

- How to Importing module from a package
- List Packages
- Using a Package
- Download a Package

Python Iterators

- Iterator vs Iterable
- Looping Through an Iterator
- Create an Iterator
- Stop Iteration

Generators & Decorators

Generators

- Generator-Function
- Generator-Object

Etc.

Decorators

- Syntax for Decorator
- Chaining Decorators
- Inside the Decorator

Etc.

Exception Handling

- What is Exception Handling
- Many Exceptions
- Try
- Else
- Finally
- Raise an exception

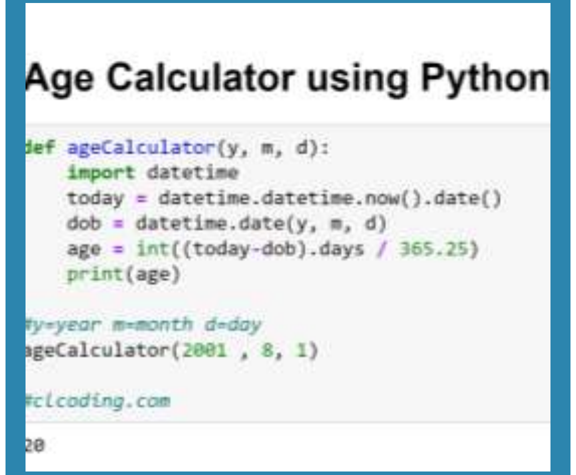
File Handling

- What is File Handling
- Syntax
- Python Read Files
- Write to an Existing File
- Create a New File
- Delete a File
- Delete Folder

PYQT5

- PyQt5 – Home
- PyQt5 – Introduction
- PyQt5 - What’s New
- PyQt5 - Hello World
- PyQt5 - Major Classes
- PyQt5 - Using Qt Designer
- PyQt5 - Signals & Slots
- PyQt5 - Layout Management
- PyQt5 - Basic Widgets
- PyQt5 – Q Dialog Class
- PyQt5 - Q message Box

Project



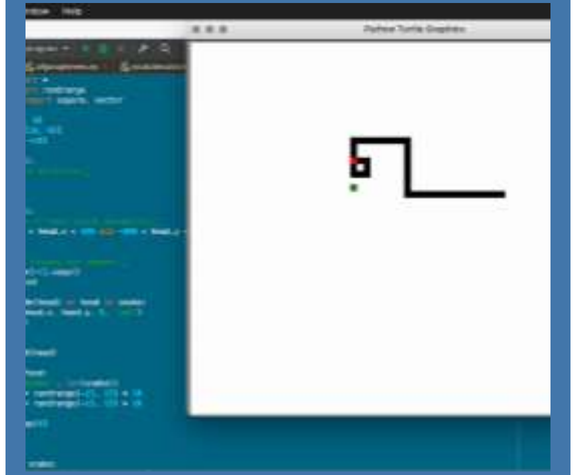
Age Calculator using Python

```
def ageCalculator(y, m, d):
    import datetime
    today = datetime.datetime.now().date()
    dob = datetime.date(y, m, d)
    age = int((today-dob).days / 365.25)
    print(age)

#y=year m=month d=day
ageCalculator(2001 , 8, 1)

#clcoding.com
```

```
20
```

Calculator Project

Tic-Tac-Toe Game Project



Snack Game Project

Puzzle Game Project PYQT5