ExecuTrain

Impulsamos tu talento tecnológico

MICROSOFT	RED HAT	VIRTUALIZACIÓN
CIBERSEGURIDAD	DESARROLLO	OFFICE
BIG DATA	BLOCK CHAIN	BASES DE DATOS
GESTIÓN DE SERVICIOS IT	CLOUD COMPUTING	METODOLOGÍAS EN PROYECTOS
SISTEMAS OPERATIVOS	Y MÁS...	

www.executrain.com.mx

¿Por qué ExecuTrain?

ExecuTrain es un proveedor de entrenamiento corporativo a nivel internacional y líder mundial en la capacitación empresarial. Contamos con más de 30 años de Experiencia y con más de 75 mil personas capacitadas a nivel Nacional.

Te guiamos en la definición de tus requerimientos de capacitación, en las diferentes etapas:

- ✓ Detección de necesidades, evaluación de conocimientos, plan de capacitación y seguimiento posterior para elegir el plan de capacitación como tú lo necesitas.
- ✓ El **más amplio catálogo de cursos**, desde un nivel básico hasta los niveles de conocimientos más especializados.
- ✓ En ExecuTrain el material y la **metodología están diseñados por expertos en aprendizaje humano**. Lo que te garantiza un mejor conocimiento en menor tiempo.
- ✓ Tú puedes confiar y estar seguro del aprendizaje porque nuestro **staff de instructores es de primer nivel**, algunos de los cuales son consultores en reconocidas empresas.
- ✓ No pierdas tu tiempo, los cursos están diseñados para un aprendizaje práctico.

Nuestro compromiso es que tú aprendas, si no quedas satisfecho con los resultados del programa, podrás volver a tomar los cursos hasta tu entera satisfacción o la devolución de tu dinero.

Modalidad de Servicio

Cursos en Fecha Calendario

Súmate a nuestros grupos en fechas públicas.

Cursos Privados

On site, en nuestras instalaciones o en línea con instructor en vivo.

Autoestudio con soporte de instructor

Cursos en modalidad autoestudio, con acceso 24/7 a la plataforma de estudio, con soporte de instructor y foros de ayuda

ExecuTrain
Impulsamos tu talento tecnológico

Duration: 20 Hours

Python / Introduction to Python

In this Python training course, students learn to program in Python. The course is aimed at students new to the language who may or may not have experience with other programming languages.

This Python course is taught using Python 3; however, differences between python 2 and python 3 are noted.

Target Audience

This course is designed to be inclusive, catering to a wide range of learners, from absolute beginners to those with some programming knowledge, ensuring a comprehensive understanding of Python 3 while acknowledging the legacy of Python 2 where relevant.

Prerequisites

- ✓ Some programming experience.

At course completion

- ✓ Learn how Python works and what it's good for.
- ✓ Understand Python's place in the world of programming languages
- ✓ Learn to work with and manipulate strings in Python.
- ✓ Learn to perform math operations with Python.
- ✓ Learn to work with Python sequences: lists, arrays, dictionaries, and sets.
- ✓ Learn to collect user input and output results.
- ✓ Learn flow control processing in Python.
- ✓ Learn to write to and read from files using Python.
- ✓ Learn to write functions in Python.
- ✓ Learn to handle exceptions in Python.
- ✓ Learn to work with dates and times in Python

Modules

- Getting Familiar with the Terminal
  - Windows PowerShell
  - Mac Terminal
  - Visual Studio Shortcut
- Running Python
  - Python Interactive Shell
- Running a Python File
  - Right-click and Run
- Hello, World
- Literals
- Python Comments
  - Multi-line Comments
- Data Types
- Exploring Types
- Variables
  - Variable Names
    - Keywords
  - Variable Assignment
  - Simultaneous Assignment
- A Simple Python Script
- Constants
- Deleting Variables
- Writing a Python Module
  - The Main () Function
- Print () Function
  - Named Arguments
- Collecting User Input
- Hello, You!
- Reading from and Writing to Files
  - Reading from a File
    - With Blocks
- Working with Files
- Defining Functions
- Variable Scope
- Global Variables
- Function Parameters
  - Using Parameter Names in Function Calls
- A Function with Parameters
- Default Values
- Parameters with Default Values
- Returning Values
- Importing Modules
  - Module Search Path
  - Pyc Files
- Methods vs. Functions
- Arithmetic Operators
  - Modulus and Floor Division
- Floor and Modulus
- Assignment Operators
- Precedence of Operations
- Built-in Math Functions
  - Int(x)
  - Float(x)
  - Abs(x)
  - Min() and max()
  - Pow(base,Exp[,mod])
  - Round(number[, ndigits])
  - Sum(iter[, start])
- The math Module
  - Common Methods of the math Module
    - Math.celi(x)
    - Math.floor(x)
    - Math.trunc(x)
    - Math.fabs(x)
    - Math.factorial(x)
    - Math.pow(x,y)
    - Math.sqrt(x)
  - Additional meth Methods
- The random Module
  - Common Methods of the random Module
    - Random.random()
    - Random.randint(a,b)
    - Random.randrange(b)
    - Random.randrange(a,b,step)
    - Random.uniform(a,b)
    - Random.choice(seq) and random.shuffle (seq)
  - Seeding
- How May Pizzas do we Need?
- Dice Rolling
- Quotation Marks and Specials Characters
  - Escaping Characters
    - Special Characters
    - Raw Strings
    - Bad

        - Good
    - Triple Quotes
- String Indexing
- Indexing Strings
    - Challenge
- Slicing Strings
    - Challenge
- Concatenation and Repetition
    - Concatenation
    - Repetition
- Repetition
- Combining Concatenation and Repetition
- Python Strings are Immutable
- Common String Methods
    - String Methods that Return a Copy of the String
        - Metods that Change Case
        - Str.replace(old,new[, count])
        - Methods that Strip Characters
    - String Methods that Return a Boolean
        - Str.isspace()
        - Str.isdigit(), str.isdecimal(), and str.isnumeric()
        - Str.startswith() and str.endswith()
    - String Methods that Return Number
        - String Methods that Return a Position (Index) of a Substring str.count(sub[, end]])
- String Formatting
    - The format() Method
    - Format Specification
        - Type
        - Precision
        - Separating the Thousands
        - Width
        - Sign
        - Alignment
        - Fill
        - Percentage Type
    - Long Lines of Code
- Playing with Formatting
- Formatted String Literals (f-strings)
- Built.in String Functions
    - Str(object)
    - Len(string)
    - Min() and Max()
- Outputting Tab-delimited Text
- Definitions
- Sequences
- Lists
    - List Methods
    - Copying a List
    - Deleting List Elements
- Sequences and Random
    - Random.choice(seq)
    - Random.suffle(seq)
- Remove and Return Random Element
- Tuples
    - When to Use a Tuple
        - Empty and Single-element Tuples
- Ranges
- Converting Sequences to Lists
- Indexing
- Simple Rock, Paper, Scissors Game
- Slicing
- Slicing Sequences
- Min(), Max(), and Sum()
    - Min(iter) and max(iter)
    - Sum(iter[, start])
- Converting Sequences to Strings with str.join(seq)
- Splitting Strings into Lists
    - The splitlines() Method
- Unpacking Sequences
- Dictionaries
    - Common Dictionary Methods
        - Mydict.get(key[, default])
        - Mydict.pop(key[, default])
        - Mydict.popitem()
        - Mydict.copy()
        - Mydict.clear()
        - Update()
        - Setdefault()
    - Dictionary View Objects
    - Deleting Dictionary Keys
- The len() Function
- Creating a Dictionary from User Input

    - Challenge
- Sets
- *args and **kwargs
    - Using *args
    - Using **kwargs
- Creating, Activiting, Deactivating, and Deleting a Virtual Environment
- Packages with pip
    - Packages and Virtual Environments
- Working with a Virtual Environment
    - Instructions

**Esto temas deberán ser repasado en autoestudio a través del manual del curso*

- Conditional Statements
    - Elif and else
    - In
    - Not in
- Compound Conditions
- The is and is not Operators
- All() and Any()
- Ternary Operator
- In Between
- Loops in Python
    - While Loops
    - For Loops
- All True and Any True
- Break and continue
- Looping through Lines in a File
- Word Guessing Game
    - Challenge
    - The else Clause of Loops
    - For …else
- The enumerate() Function
- Generators
    - When to Use generators
        - Infinite Sequences
        - The Fibonacci Sequence
        - A Sequence of Unknown Length
    - The next() Function
- List Comprehensions
    - Creating Sublists with List Comprehensions
- Exception Basics
    - Multiple except Clauses
- Wildcard except Clauses
- Getting Information on Exceptions
- Raising Exceptions
- The else Clause
- The finally Clause
- Using Exceptions for Flow Control
- Running Sum
- Raising Your Own Exceptions
- Understanding Time
    - The Epoch
    - Python and Time
- The time Module
    - Clocks
        - Absolute Time
        - Relative Time
- Time Structures
    - Time.gmtime([secs])
    - Epoch as struct_time
    - Current Time as struct_time
    - Yesterday as struct_time
    - Tomorrow as struct_time
    - Time.localtime([secs])
    - Epoch as Local struct_time
    - Tomorrow as Local struct_time
- Times as Strings
    - Time.asctime([t])
    - String Representation of Epoch
    - String Representation of Current Local Time
    - Time.ctime([secs])
- Time and Formatted Strings
    - Time.strftime(format[, t])
    - Time.strptime(string[, format])
    - Formatting directives
- Pausing Execution with time.sleep()
- The datetime Module
    - Datetime.date Objects
        - Datetime.date Attributes
        - Datetime.date Methods
    - Datetime.time Objetcs
        - Datetime.time Attributes
        - Datetime.time Methods
- Datetime.datetime Objects
    - Datetime.datetime Attributes
    - Datetime.datetime Methods
- What Color Pants Should I Wear=
- Datetime.timedelta Objetcs
    - Datetime.timedelta Attributes
    - Datetime.timedelta.total_seconds() Method

    - The Time Delta Between Dates
- Report on Departure times
- Opening Files
    - Methods of file Objects
        - Reading Files
        - Closing Files
- Finding Text in a File
    - Challenge
- Writing to Files
    - Challenge
- List Creator
    - Challenge
- The os Module
    - Os.getcwd() and os.chdir(path)
    - Os.listdir(path)
    - Os.mkdir(dirname) and os.makedirs(path)
    - Os.rmdir(path) and os.removedirs(path)
    - Os.remove(path) and os.unlink(path)
    - Os.rename(source, destination) and os.renames(oldpath, newpath)
- Walking Directory
    - Os.walk(top, topdown=True, onerror=None, followlinks=False)
    - Os.walk () Parameters
- The os.path Module
    - Os.path.abspath(path)
    - Os.path.basename(path)
    - Os.path.dirname(path)
    - Os.path.exists(path)
    - Os.path.gatetime(path), os.path.getmtime(path), and os.path.getctime(path)
    - Os.path.getsize(path)
    - Os.path.isabs(path)
    - Os.path.relpath(path, start)
    - Os.path.isfile(path) and os.path.isdir(path)
    - Os.path.join(path, *paths)
    - Os.path.split(path)
    - Os.path.spliterxt(path)
- A Better Way to Open Files
- Comparing Lists
- PEP8
    - Maximum Line Length
    - Indentation
    - Continuation Lines
    - Blank Lines
    - UTF-8 Encoding
    - Imports
    - Quotes
    - Whitespace in Expressions and Statements
    - Good
    - Bad
    - Comments
    - Additional Content in PEP8
- Pylint
    - Linter.py

**Esto temas deberán ser repasado en autoestudio a través del manual del curso*