

PYTHON

OBJETIVOS

El objetivo del curso es introducir el lenguaje de programación Python, su sintaxis, estructuras básicas y tipos de datos, ofreciendo una visión de las posibilidades que ofrece.

Aunque no es estrictamente obligatorio, se aprovecha más la formación si se posee algún conocimiento de programación en otros lenguajes.

DURACIÓN

24 horas

REQUISITOS PREVIOS

Conocimientos de programación

CONTENIDOS

- Introducción
 - ¿Qué es Python?: Algunas características, propiedades del lenguaje, biblioteca estándar.
 - Primeros pasos: Ejemplo de código, ejecutando Python.
 - Recursos: Documentación, cómo pedir ayuda.
- Tipos de Datos
 - Distintos tipos, aritmética real, divisiones, conversiones.
 - Cadenas de texto: Construyendo, accediendo a los caracteres, métodos propios, Unicode, expresiones regulares.
 - Tuplas y listas: Construyendo, accediendo a los elementos, modificándolas, métodos propios,
 - Nombres y objetos, espacios de nombres, revinculación de nombres, objetos mutables e inmutables
 - Conjuntos: Construyendo, accediendo y modificando, operaciones entre conjuntos.
 - Diccionarios: Construyendo, accediendo por clave, recorriendo diccionarios, operaciones entre diccionarios.
- Controles de flujo

- if, if/else, if/elif/else, condiciones.
 - for, continue/break, for/else, recorriendo números, comprensiones de listas, enumerate, zip, map, itertools.
 - while, continue/break, while/else.
 - Excepciones: ¿Por qué? Capturando excepciones, generándolas, definiendo nuevas
- Encapsulando código
 - Funciones: Definiendo una función, llamando a una función, tipos de argumentos, espacios de nombres, recursividad, funciones Lambda, generadores.
 - Módulos: Usando módulos, creando un módulo, evitando la ejecución, paquetes.
 - Otros temas
 - Archivos: Modo texto y binario en combinaciones Python 2 y 3, Windows, Linux; Trabajando en binario. Trabajando con texto. Módulos y funciones relevantes.
 - Entornos virtuales: Conceptos: distintos import paths, aislación. Creación y uso de un virtualenv.
 - Llamadas al sistema: Cómo disparar procesos. Esperar el resultado. Ver salida estándar, ver errores.

