

## Programa formativo: Introducción a la programación a través de Python (OCW 2023)

En la actualidad, es fundamental no solo saber utilizar los ordenadores (alfabetización digital), sino también comprender cómo funcionan y tener la capacidad de crear nuestros propios programas para resolver problemas específicos. La programación se ha convertido en una habilidad cada vez más valiosa y demandada en diversos ámbitos profesionales. Aprender a programar en Python, un lenguaje de programación de alto nivel y gran versatilidad brinda a los estudiantes la oportunidad de adquirir competencias fundamentales en el campo de la programación.

Este curso tiene como objetivo proporcionar a los estudiantes los conocimientos básicos necesarios para iniciarse en la programación con Python. A través de este curso, los estudiantes desarrollarán una comprensión sólida de los conceptos fundamentales de la programación y adquirirán habilidades prácticas para resolver problemas mediante el uso de Python. Estas habilidades les brindan a los estudiantes una ventaja competitiva en el mercado laboral, ya que muchas profesiones y sectores demandan conocimientos básicos de programación.

### CONTENIDOS

El curso contendrá diferentes contenidos para trabajar varios conceptos de programación, concretamente en forma de [Jupyter Notebooks](#) (cuadernos). Para ponerlos en práctica, se comenzará con una descripción y una breve guía sobre diferentes plataformas de computación en la nube ([CoCalc](#) y [Google Colab](#)), para posteriormente introducir estos conceptos.

En concreto, los contenidos del curso son los que siguen:

1. Introducción a entornos de programación:
  - a. CoCalc: es una plataforma en línea diseñada para satisfacer las necesidades de científicos, ingenieros y estudiantes que trabajan en campos como las matemáticas, la ciencia de datos y la informática. Este entorno virtual proporciona una amplia gama de herramientas y recursos para realizar cálculos científicos, desarrollar código en varios lenguajes de programación (como Python o R), y crear documentos interactivos. En concreto se trabajará:
    - i. Uso de terminal.
    - ii. Comandos.
    - iii. Intérprete de Python.
  - b. Google Colab: es una plataforma de Google que ofrece un entorno en línea gratuito y basado en la nube para la ejecución de código en Python. Es especialmente popular entre científicos de datos, desarrolladores y estudiantes, ya que proporciona acceso a potentes recursos de cómputo, incluyendo GPU y TPU, para la ejecución de proyectos de aprendizaje automático y análisis de datos. Además, permite la colaboración en tiempo real y la creación de documentos interactivos utilizando Jupyter Notebooks.

- i. Uso general de la plataforma: como conectar una máquina virtual, compartir cuadernos, gestión de ficheros, etc.
    - ii. Uso general de cuadernos: diferentes tipos de celdas (texto y código), exportar un cuaderno, etc.
    - iii. Markdown: lenguaje de marcado con el que es posible formatear texto de manera sencilla.
  2. Conociendo Python: se comienza a trabajar con la sintaxis de Python, así como con diferentes tipos de datos y cómo podemos ejecutar instrucciones utilizando variables y según el uso de diferentes operadores.
    - a. Sintaxis de Python: reglas que se deben seguir para escribir código en Python.
    - b. Tipos de datos: existen diferentes categorías de valores.
      - i. Enteros: números sin parte decimal.
      - ii. Flotantes: números con parte decimal.
      - iii. Cadena de texto: secuencias de caracteres.
      - iv. Booleano: representa un valor binario (verdadero o falso).
      - v. None: representa la ausencia de valor.
    - c. Variables: sirven para almacenar datos en memoria y acceder a ellos mediante un nombre asignado.
    - d. Operadores: símbolos que se utilizan para realizar operaciones.
      - i. Operadores aritméticos: sirven para realizar cálculos matemáticos.
      - ii. Operadores de asignación: permiten asignar valores a variables.
      - iii. Operadores de comparación: sirven para comparar valores.
      - iv. Operadores lógicos: permiten realizar operaciones lógicas en valores booleanos.
  3. Ampliando conocimientos de Python: se trabajan conceptos más complejos, como la ejecución de una u otra instrucción según se cumpla una condición o como utilizar estructuras de Python para repetir instrucciones o para crear fragmentos de código dentro de nuestro programa.
    - a. Condicionales: permiten tomar decisiones basadas en condiciones.
    - b. Listas: sirven para almacenar colecciones de elementos.
    - c. Rangos: representan secuencias numéricas o índices.
    - d. Bucles: permiten la automatización de repeticiones del código.
      - i. For: iteran sobre elementos.
      - ii. While: se ejecutan mientras se cumpla una condición.
    - e. Funciones: fragmentos de código que realizan tareas específicas.
  4. Librerías: es posible aumentar las funcionalidades de Python si importamos librerías o módulos ya programados para, por ejemplo, trabajar con matrices o generar gráficos.
    - a. math: proporciona funciones matemáticas para realizar operaciones avanzadas, como funciones trigonométricas, logaritmos, raíces cuadradas, etc. Es una librería estándar de Python, por lo que no requiere instalación por separado.
    - b. NumPy: permite utilizar arreglos multidimensionales (arrays) y realizar operaciones de manera eficiente con ellos.
    - c. Pandas: proporciona estructuras de datos y herramientas para ejecutar análisis de datos fácilmente.

- d. matplotlib: es una librería de visualización que permite crear gráficos y visualizaciones de datos.  
También se verá cómo utilizar el gestor de paquetes pip, para instalar paquetes que no estén entre los estándar.
- 5. Materiales de consulta: se ofrecen una serie de enlaces a tutoriales y plataformas con ejercicios con los que poner en práctica los contenidos del curso, así como trabajarlos aún más.