**Python es el lenguaje para aprender**

**Si hubiera un solo idioma...**

Python a menudo se describe como poderoso, pero fácil de aprender. Es un lenguaje de programación de propósito general, pero viene con miles de bibliotecas para trabajar con datos. A menudo es el primer idioma que las personas aprenden, sin embargo, es el idioma que las personas usan décadas más tarde en su carrera. Python es a menudo la primera herramienta que las personas toman cuando necesitan crear un prototipo rápido, sin embargo, es el lenguaje que impulsa algunos de los proyectos más complejos. El poder y la flexibilidad con miles de bibliotecas hicieron de Python el lenguaje de elección para científicos de datos, ingenieros de datos, analistas de datos e investigadores de IA en empresas líderes y nuevas empresas por igual. Python es esencial para cualquier currículum y es una habilidad imprescindible para cualquiera que trabaje con datos.

En resumen, si solo pudieras aprender un lenguaje de programación, Python debería ser el lenguaje para aprender.

**Algunas cosas que debe saber sobre este curso**

Este curso es a su propio ritmo. No hay un instructor que te siga o que te apresure, aprendes en tu propio horario. Esto también significa que vas a necesitar ser disciplinado en tus estudios.

Puede iniciarlo en cualquier momento y puede tardar todo el tiempo que necesite para completarlo. Su lugar en el curso será recordado y sus laboratorios serán guardados. Sin embargo, le recomendamos encarecidamente que no tome descansos prolongados entre lecciones, ya que la mente tiende a olvidar lo que no practica.

Puede revisar el curso tantas veces como desee, incluso después de completarlo.

Este es un curso práctico donde obtendrá habilidades prácticas haciendo laboratorios prácticos. Proporcionamos un entorno de laboratorio virtual completo en la nube al que puede acceder desde cualquier lugar, siempre que tenga conexión a Internet.

Le recomendamos encarecidamente que utilice el navegador Chrome o Firefox

Puede ver videos y leer materiales del curso en un dispositivo móvil. Sin embargo, es probable que necesite usar una computadora para laboratorios prácticos, ya que el tamaño pequeño de la pantalla y la falta de teclado hacen que los dispositivos móviles sean una mala opción para los laboratorios prácticos.

**Objetivos de aprendizaje**

Los objetivos de este curso son comenzar con Python como lenguaje de programación y darle una idea de cómo comenzar a trabajar con datos en Python.

En este curso aprenderás sobre:

* 1. Qué es Python y por qué es útil
  2. La aplicación de Python
  3. Cómo definir **variables**
  4. **Conjuntos** y **sentencias condicionales** en Python
  5. El propósito de tener **funciones** en Python
  6. Cómo operar en archivos para **leer y escribir datos** en Python
  7. Cómo usar **pandas**, un paquete imprescindible para cualquiera que intente analizar datos en Python

**Temario**

**Módulo 1 - Conceptos básicos de Python**

* 1. Tu primer programa
  2. Tipos
  3. Expresiones y variables
  4. Operaciones de cadena

**Módulo 2 - Estructuras de datos de Python**

* 1. Listas y tuplas
  2. Establece
  3. Diccionarios

**Módulo 3 - Fundamentos de programación de Python**

* 1. Condiciones y ramificación
  2. Bucles
  3. Funciones
  4. Objetos y clases

**Módulo 4 - Trabajar con datos en Python**

* 1. Lectura de archivos con open
  2. Escribir archivos con open
  3. Carga de datos con pandas
  4. Trabajar con y guardar datos con pandas

**Módulo 5 - Trabajar con matrices Numpy y API simples**

* 1. Matrices 1D Numpy
  2. Matrices 2D Numpy
  3. API sencillas
  4. Configuración de API

**Esquema de calificación**

1. La nota mínima de aprobación para el **curso** es del 70% con los siguientes pesos:
   1. 50% - Todas las preguntas de revisión
   2. 50% - El examen final
2. Aunque las preguntas de revisión y el examen final tienen una calificación aprobatoria del 50% respectivamente, la única calificación que importa es la calificación general del **curso**.
3. Las preguntas de revisión no tienen límite de tiempo. Le recomendamos que revise el material del curso para encontrar las respuestas.  Recuerde que las preguntas de revisión valen el 50% de su calificación final.
4. El examen final tiene un límite de tiempo de 1 hora.
5. Los intentos son por **pregunta** tanto en las preguntas de revisión como en el examen final:
   1. Un intento - Para preguntas Verdadero/Falso
   2. Dos intentos: para cualquier pregunta que no sea Verdadero / Falso
6. No hay penalizaciones por intentos incorrectos.
7. Al hacer clic en el botón "**Verificación final**" cuando aparece, significa que su envío es **FINAL.**  NO podrá volver a enviar su respuesta para esa pregunta nunca más.
8. Verifique sus calificaciones en el curso en cualquier momento haciendo clic en la pestaña "Progreso".

**Registro de cambios**

**2018-10-02**

1. Cambiar el contenido en Notebook 3.1 - Mavis

**2017-12-01**

1. Los cuadernos cambiaron ligeramente J.S

**2017-12-01**

1. Orden de cuaderno cambiado J.S

**2017-11-12:**

1. Reformuló algunas oraciones para mayor claridad. Se han corregido algunos errores tipográficos.

**2017-09-15:**

1. Lanzamiento del curso recientemente renovado

**Objetivos de aprendizaje**

Después de completar esta sección, usted:

* 1. Comprender los tipos de datos disponibles en Python
  2. Familiarizarse con las expresiones y variables de Python
  3. Ser capaz de realizar operaciones básicas en cadenas

Lo más importante es que escribirá sus primeros bits de código Python utilizando Jupyter Notebook, el mismo entorno rico utilizado por miles de científicos de datos, analistas de datos e ingenieros de datos.

**Your First Program 1 (1:15)**

Probemos tu primer programa en Python. Una declaración o expresión es una instrucción.

El equipo se ejecutará o ejecutará. Quizás el programa más simple que puedes escribir es una declaración impresa. Cuando ejecute la instrucción print, Python

Simplemente muestre el valor entre paréntesis. El valor entre paréntesis se denomina argumento. Si está utilizando un cuaderno de Júpiter en este Por supuesto, verá un pequeño rectángulo con la declaración.

Esto se llama célula. Si selecciona esta celda con el ratón, entonces Haga clic en el botón Ejecutar celda. La instrucción se ejecutará. El resultado se mostrará debajo la célula. Seguiremos este paradigma en los vídeos.

Es costumbre comentar su código. Esto le dice a otras personas lo que hace su código. Tú Simplemente ponga un símbolo hash procediendo a su comentario. Cuando ejecute el código, Python ignorará el comentario. Un error sintáctico es cuando Python no Comprenda su código. Por ejemplo, si escribe print "frint", obtendrá un error Mensaje. Un error semántico es cuando su lógica es incorrecta.

Por ejemplo, si escribe Python 102 en lugar de Python 101, no recibe un mensaje de error, Pero su código es incorrecto.

**Types (2:57)**

Un tipo es cómo Python representa diferentes tipos de datos. En este video, discutiremos algunos tipos ampliamente utilizados en Python. Puedes tener diferentes tipos en Python.

Pueden ser enteros como 11, números reales como 21.213. Incluso pueden ser palabras.

Los enteros, los números reales y las palabras se pueden expresar como diferentes tipos de datos.

En el siguiente gráfico se resumen tres tipos de datos para los últimos ejemplos.

La primera columna indica la expresión. La segunda columna indica el tipo de datos.

Podemos ver el tipo de datos real en Python usando el comando type.

Podemos tener int, que significa un entero, y float que significa float, esencialmente un número real. La cadena de tipo es una secuencia de caracteres.

Aquí hay algunos enteros; Los enteros pueden ser negativos o positivos.

Cabe señalar que hay un rango finito de enteros, pero es bastante grande.

Los flotadores son números reales; Incluyen los enteros, pero también los números entre los enteros.

Considere los números entre 0 y 1. Podemos seleccionar números entre ellos; Estos números son carrozas. Del mismo modo, considere los números entre 0.5 y 0,6. Podemos seleccionar números entre ellos; Estos también son flotadores.

Podemos continuar el proceso, acercándonos a diferentes números. Por supuesto, hay un límite, Pero es bastante pequeño. Puede cambiar el tipo de expresión en Python; Esto se denomina fundición de tipos. Puede convertir un int en un float.

Por ejemplo, puede convertir o convertir el entero 2 en un flotante 2.

Nada cambia realmente. Si convierte un flotador a un número entero, debe

Ten cuidado. Por ejemplo, si convierte el float 1.1 a

1, perderá algo de información. Si una cadena contiene un valor entero, puede convertirlo a int. Si convertimos una cadena que contiene un número no entero valor, obtenemos un error. Echa un vistazo a más ejemplos en el laboratorio.

Puede convertir un int en una cadena o un float en una cadena.

Boolean es otro tipo importante en Python. Un booleano puede asumir dos valores.

El primer valor es verdadero, solo recuerda que usamos una T mayúscula.

Los valores booleanos también pueden ser falsos, con una F mayúscula.

Usando el comando type en un valor booleano, obtenemos el término bool, esto es la abreviatura de Booleano. Si convertimos un booleano verdadero a un entero o flotar, obtendremos un 1. Si convertimos un booleano falso a un entero o flotar, obtenemos un cero. Si lanzas un 1 a un booleano, obtienes un verdadero.

Del mismo modo, si lanzas un 0 a un booleano, obtienes un falso.

Consulte los laboratorios para obtener más ejemplos o consulte Python.org para otros tipos en Python.

**Expresiones y variables (3:50)**

En este video cubriremos expresiones y variables.

Las expresiones describen un tipo de operación que realizan los equipos.

Las expresiones son operaciones que realiza Python. Por ejemplo, operaciones aritméticas básicas como Añadiendo varios números. El resultado, en este caso, es 160.

Llamamos a los números operandos y los símbolos matemáticos, en este caso suma, se llaman Operadores. Podemos realizar operaciones como la resta usando el signo de resta. En este caso, el resultado es un número negativo.

Podemos realizar operaciones de multiplicación utilizando el asterisco. El resultado es 25.

En este caso los operandos vienen dados por – y \*.

También podemos realizar la división con la barra diagonal.

25 dividido por 5 es 5. 25 dividido por 6 es aproximadamente 4.167.

En Python 3, la versión que usaremos en este curso, ambos resultarán en un flotador.

Podemos usar la barra doble para la división de enteros, donde el resultado es redondeado.

Tenga en cuenta que, en algunos casos, los resultados no son los mismos que los de la división regular.

Python sigue las convenciones matemáticas al realizar expresiones matemáticas.

Las siguientes operaciones están en un orden diferente.

En ambos casos, Python realiza la multiplicación, luego la adición para obtener el resultado final.

Hay muchas más operaciones que puedes hacer con Python. Consulte los laboratorios para obtener más ejemplos.

También cubriremos operaciones más complejas a lo largo del curso.

Las expresiones entre paréntesis se realizan primero. Luego multiplicamos el resultado por 60, El resultado es 1920. Ahora veamos las variables.

Podemos usar variables para almacenar valores, en este caso, asignamos un valor de 1 a la variable my\_variable utilizando el operador de asignación, es decir, el signo igual.

Luego podemos usar el valor en otro lugar del código escribiendo el nombre exacto de la variable.

Usaremos dos puntos para denotar el valor de la variable.

Podemos asignar un nuevo valor a my\_variable usando el operador de asignación.

Asignamos un valor de 10. La variable ahora tiene un valor de 10.

El valor antiguo de la variable no es importante. Podemos almacenar los resultados de las expresiones, para Por ejemplo, agregamos varios valores y asignamos el resultado a x. x ahora almacena el resultado. También podemos realizar operaciones en x y guardar El resultado de una nueva variable y. y ahora tiene un valor de 2,666 También podemos realizar operaciones en x y asignar el valor x.

La variable x ahora tiene un valor de 2,666. Como antes, el valor antiguo de x no es importante.

También podemos usar el comando type en variables.

Es una buena práctica usar nombres de variables significativos, para que no tenga que realizar un seguimiento de qué está haciendo la variable. Digamos que nos gustaría convertir el número de minutos en los ejemplos resaltados al número de horas en el siguiente conjunto de datos de música.

Llamamos a la variable que contiene el número total de minutos, total\_min. Es común Para utilizar el carácter de subrayado para representar el comienzo de una nueva palabra, también puede utilizar mayúscula carta. Llamamos a la variable que contiene el total número de horas total\_hr. Podemos obtener el número total de horas por

dividiendo total\_min por 60. El resultado es de aproximadamente 2.367 horas.

Si modificamos el valor de la primera variable, el valor de la variable cambiará.

Los valores de los resultados finales cambian en consecuencia, pero no tenemos que modificar el resto de los código.