

Fundamentos de Python

Capacitación de analítica avanzada



Agenda

1

Introducción y
objetivos de la sesión

4

Funcionalidades
claves

2

¿Qué es Python?,
características

5

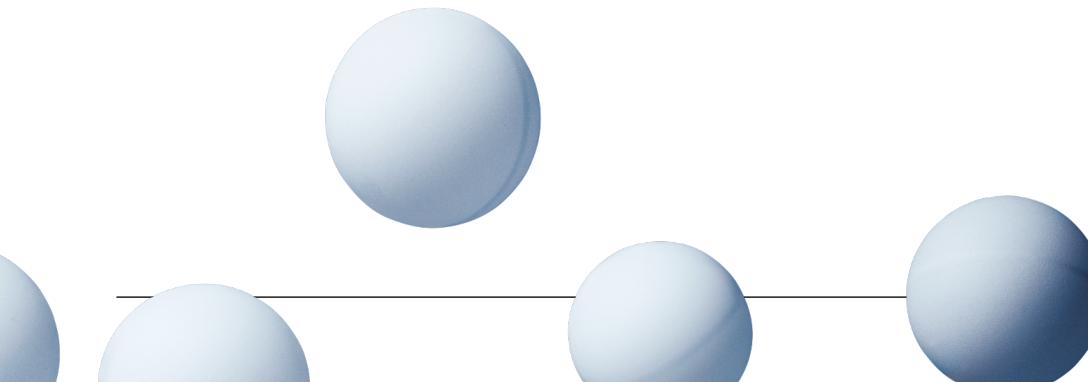
Casos de uso
comunes

3

Tipos de datos y
estructuras

6

Python: un solo
lenguaje para diversas
funciones



Recordemos cómo se integra esta sesión en el programa

● Intervenciones teóricas

● Intervenciones prácticas en modo demo/tutorial

#	Módulo	Área	Tipo	Sesión
2.1	Fundamentos para el análisis de datos	<i>Coding</i>	●	Sistema Unix, terminal y bash
2.2	Fundamentos para el análisis de datos	<i>Coding</i>	●	Configuración del ambiente de desarrollo: brew, apt, conda & pip
2.3	Fundamentos para el análisis de datos	<i>Coding</i>	●	Fundamentos de Python
2.4	Fundamentos para el análisis de datos	<i>Coding</i>	●	Fundamentos de Python
2.5	Fundamentos para el análisis de datos	<i>Data engineering</i>	●	Fundamentos de Pandas para el análisis de datos

Mini-proyecto

Introducción y objetivos de la sesión



Introducción

Como principal herramienta para el análisis de datos y base de los demás iniciativas de código abierto sobre la cual se fundamentan los modelos analíticos se encuentra Python. Entender el origen, estructura y funcionalidades básicas de Python es el primer paso para iniciar en el análisis avanzado de datos,

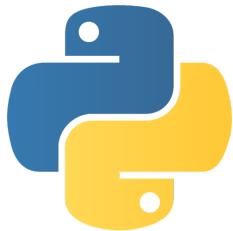


Objetivos

- Entender el origen y contexto sobre el cual funciona Python
- Conocer las características de Python que lo hacen favorable para el análisis de datos
- Conocer las estructuras básicas de Python
- Conocer las aplicaciones más comunes de Python

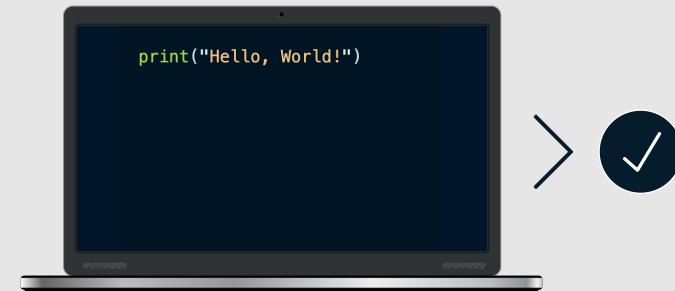
¿Qué es Python?

Python es un lenguaje de programación de alto nivel, de código abierto ampliamente usado en la actualidad debido a su versatilidad, simplicidad y popularidad en diversas aplicaciones.



Veamos algunas de las características que lo hacen famoso

Lenguaje interpretado



Python es un lenguaje interpretado, cada vez que se ejecuta una línea se ejecuta el código, dando un resultado

Lenguaje compilado



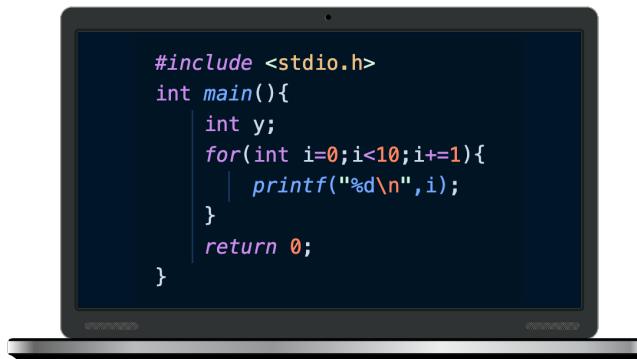
Características de python

Simple lectura

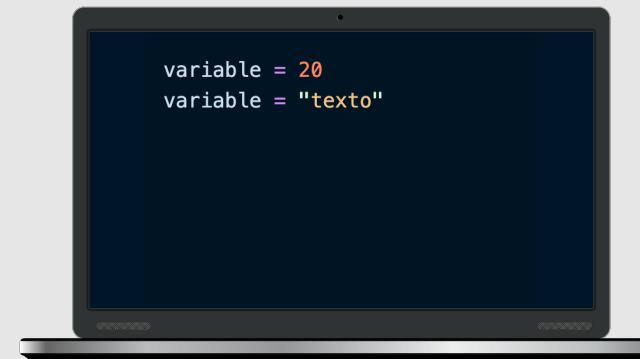


Python es minimalista en su sintaxis

Compleja lectura

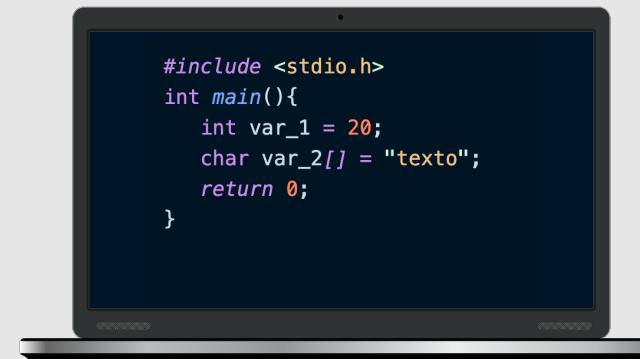


Definición dinámica de tipo de objetos



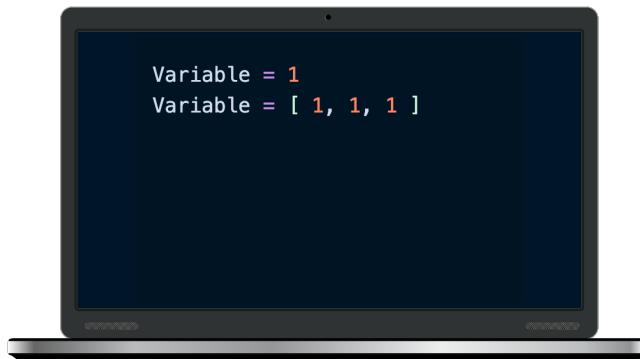
Python define el tipo de dato con la asignación de valor

Definición estática de tipo de objetos



Características de python

Manejo automático de memoria



Python ajusta la memoria conforme se requiera

Manejo explícito de la memoria



Programación orientada a objetos



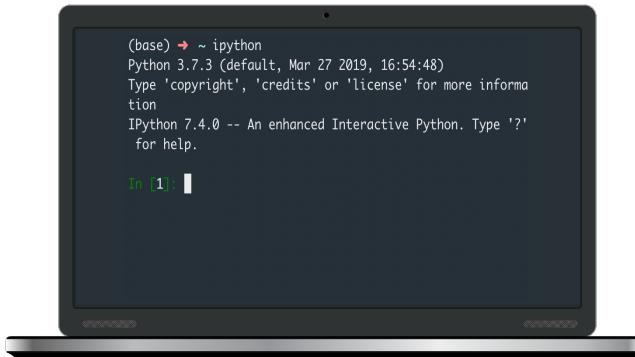
Python permite y se caracteriza por usar un paradigma de programación más usual

Programación procedural explícito de la memoria



Fundamentos de Python (1/2)

Estructuras, clases y tipos de variables



```
(base) ~ ~ ipython
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

¿Cómo programar en Python?

Python usa un interprete para ejecutar las acciones



```
In [1]: def mi_funcion():
...:     return 1
...:

In [2]:
```

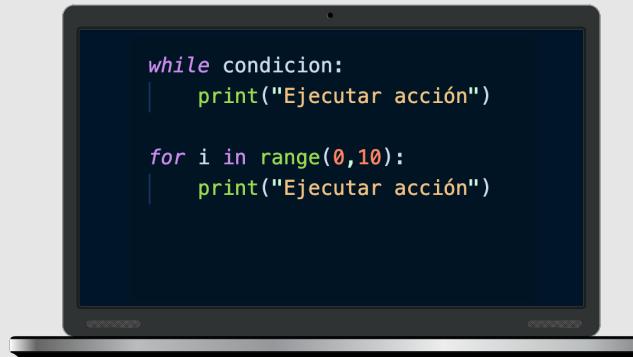
Indentación + ":" para definir la jerarquía entre bloques de códigos

Estructuras de control de flujo



```
if dinero > 1000 :
    return "Comprar"
elif valor < 0 :
    return "No comprar"
else:
    return "Prestamo"
```

Condicionales



```
while condicion:
    print("Ejecutar acción")

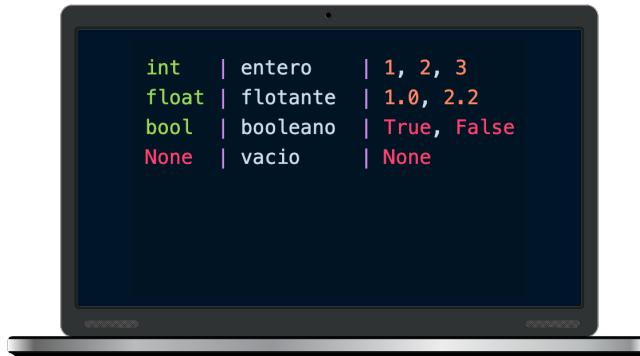
for i in range(0,10):
    print("Ejecutar acción")
```

Iteradores

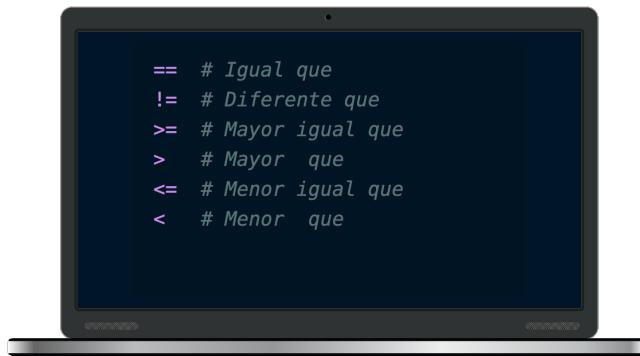
Fundamentos de Python (1/2)

Estructuras, clases y tipos de variables

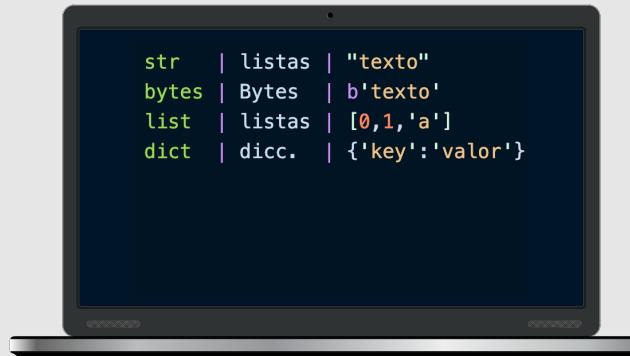
Tipos de datos y estructuras



Operadores racionales



Colecciones de datos



Conclusión

Python es un lenguaje minimalista, simple, intuitivo que concentra su poder en la creación de herramientas de forma simple y una curva de aprendizaje muy rápida

Python tiene diversas funcionalidades



**Procesamiento de
texto**



**Uso de módulos
matemáticos y
científicos**



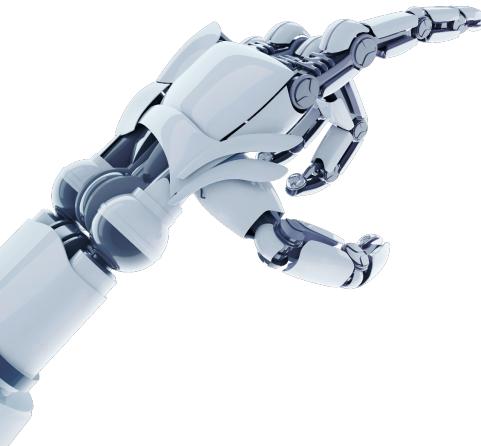
Manejo de archivos



**Interacción con el
sistema operativo**

Casos de uso más comunes

**Aplicaciones
más comunes :**



Desarrollo web



Servicio y mantenimiento de
infraestructura



Ciencia de datos y aprendizaje
automático



Análisis numérico

Un solo lenguaje para diversas funciones en ciencia de datos

Al ser un lenguaje de programación multipropósito, ha permitido que diversas áreas construyendo la misma solución, tengan un lenguaje común y por ende mayor sinergia en el desarrollo

Etapa	Almacenamiento	» Back-end	» AA- análisis	» Front-end
Funciones	Soportar sistemas y bases de datos y darle continuidad a estos a la operación de estos	Infraestructura que soporta las operaciones	Modelos predictivos y descriptivos que agregan valor al negocio	Código que soporta salida del contenido y el contacto con el usuario final

Creación de contenido y referencias

Autores



Iván Torroledo

Data science fellow

Ivan_Torroledo@mckinsey.com

Bogotá



Simon Tamayo

Data science specialist

Simon_Tamayo@mckinsey.com

Bogotá

Referencias

El contenido ha sido desarrollado gracias a la contribución de los autores y las siguientes fuentes externas:

- Documentación oficial Python. Python Software Foundation. Recuperado en Marzo 2020:
<https://docs.python.org/3.7/tutorial/index.html>