

«Talento Tech»

Data Analytics con Python

Clase 01





Clase N° 1 | Conceptos básicos

Temario:

- Presentación del recorrido propuesto e introducción del proyecto final.
- Conoceremos el lenguaje de programación Python y sus aplicaciones.
- Descubriremos el rol de un Data Analyst y su importancia.
- Aprenderemos a declarar variables en Python. Los tipos de datos, operadores y a introducir datos por teclado.



Hoja de Ruta



PROYECTO FINAL



https://www.canva.com/design/DAGNw1foMC8/bnQALgV0UfFpyyzMrvZ0Lw/edit?utm_content=DAGNw1foMC8&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

¿Qué es Python?



Python es un lenguaje de programación de alto nivel en donde uno de sus objetivos es la legibilidad del código.

Los lenguajes de alto nivel son aquellos que intentan utilizar el lenguaje natural (el lenguaje que utilizamos día a día los humanos para hablar y escribir) para realizar programas, y que necesitan como soporte un programa que los traduzca al lenguaje máquina.

Es altamente demandado por las empresas y startups, siendo los desarrolladores que programan con Python los que acceden a trabajos mejor remunerados y abarcan mayor variedad de temas.

Ámbitos de aplicación de Python.

Es uno de los lenguajes de programación más demandados por las empresas del ecosistema tech, teniendo mucha proyección.

Es uno de los lenguajes más versátiles que existen, puede ser usado en muchos campos diferentes. Es decir, permite programar videojuegos, aplicaciones móviles, criptomonedas, entre otras cosas.

Resulta una tecnología clave para la Ciencia de Datos, Machine Learning y Big Data, teniendo un rol protagónico en el futuro de las empresas y startups que se están construyendo.

Además de los usos asociados al mundo Data, Python también aplica al Desarrollo Web y es utilizado para el desarrollo de plataformas y aplicaciones como Netflix o Instagram.

Por su sintaxis sencilla, es un lenguaje que resulta fácil de aprender. Una ventaja adicional que ofrece Python es que a partir de su comprensión, podrás entender más fácilmente otros lenguajes de programación por lo que es ideal para dar tus primeros pasos aunque nunca hayas visto nada de código.

¿Qué es Data Analytics?



Data Analytics o Análisis de datos es un proceso que consiste en examinar y analizar datos para sacar conclusiones. Estas conclusiones nos van a permitir tomar mejores decisiones sobre el problema que estemos estudiando.

Por ejemplo, imagina que querés comprar un nuevo celular. Podrías ir directamente y comprar el primero que veas, pero en general si vamos a comprar algo queremos saber qué nos va a durar en el tiempo y evaluar su costo y beneficios comparado a lo que podemos gastar. En ese caso lo que haremos es establecer primero cuál es nuestro presupuesto. Un Data Analyst o analista de datos hace algo similar a lo que vos harías.

- Primero definimos cuál es el problema: ¿Cuál es el mejor celular que puedo adquirir con mi presupuesto?

- Luego debemos recopilar información.
- - Vamos a obtener información sobre los precios de los diferentes productos, las características (cámara, memoria ram, memoria de almacenamiento, duración de la batería, tamaño de la pantalla, conectividad).
 - Nos pueden ser útiles las opiniones de los distintos usuarios sobre los productos.
 - Buscaremos información sobre precios, descuentos, ofertas, medios de pago
- Ya con los datos disponibles los limpiaremos y prepararemos. Puede ser que tengamos información duplicada y que tengamos que organizarla de manera que podamos comparar fácilmente. Tal vez lo que más nos interesa es la duración de la baterías, podremos ordenar los datos de nuestro rango de presupuesto en función a la duración de la baterías.
- Después tendremos que analizar nuestros datos. Compararemos precios, beneficios, características, opiniones y analizaremos las ofertas y medios de pago.
- En el siguiente paso tendremos que interpretar los datos. Veremos cuales son los mejores que se adaptan a nuestras necesidades y presupuesto. Podemos evaluar cuál tiene mejor memoria, cuál tiene mejores calificaciones y opiniones positivas, cuál tiene mejor cámara, cuáles son las ofertas.
- Por último tomaremos una decisión. Nos basamos en todo el análisis previo para elegir cuál nos conviene comprar

Todo el proceso anterior es similar al que hace un data analysts. El data analyst lo hace con mayor cantidad de datos y dispuestos de distintas maneras. Así que sin darnos cuenta, al querer tomar una decisión, recopilamos datos, los limpiamos, los ordenamos, analizamos e interpretamos, finalmente en base a todo nuestro análisis tomamos la mejor decisión.

¿Qué hace un Data Analyst? ¿Qué debe saber un Data Analyst?

El Data Analyst o Analista de Datos se especializa en recopilar, procesar y analizar los datos para extraer información que ayude a tomar decisiones.

Dentro de sus funciones se ocupa de:

1. Recolectar datos.

- Puede obtener datos de diversas fuentes: bases de datos, archivos, sistemas empresariales, encuestas.
- Debe asegurarse la precisión de los datos y su integridad (que sean exactos, tengan consistencia y sean completos).
- 2. Limpiar y preparar los datos.
 - Eliminar datos que puedan estar duplicados, corregir errores, evaluar datos faltantes.
 - Transformar y estructurar los datos para el análisis. Podemos tener un dato que este descripto de distintas maneras según la fuentes. Deben asegurarse de que todos los datos tengan el formato adecuado.
- 3. Analizar los datos.
 - Utilizar herramientas de análisis de datos y estadística para identificar tendencias, patrones y relaciones en los datos.
- 4. Visualizar los datos.
 - Crear tablas, gráficos y paneles para comunicar sus hallazgos de manera efectiva.
- 5. Interpretar los resultados.
 - Extraer conclusiones de los datos analizados.
 - Presentar informes.

Un Data Analyst no solo debe contar con habilidades técnicas en herramientas de análisis de datos como Python, SQL (Structured Query Language), Excel, R u otras, sino que además debe conocer diferentes herramientas de visualización de datos como son las distintas librerías, Tableau, Power Bi, etc.

Otro aspecto fundamental para ser un buen Data Analytics, se vincula con poseer herramientas de comunicación que le permitan comunicar y presentar sus resultados a audiencias que no poseen el mismo nivel de conocimientos técnicos.

Nos proponemos que a lo largo del curso aprendas todas las herramientas que necesitas para convertirte en un gran analista de datos.

¡Comencemos!

Introducción a Google Colab.



Google Colab es un servicio que nos provee **Google** que nos permitirá programar y ejecutar código en Python.

No necesitas configurar nada. Para utilizarlo solamente necesitas tener una cuenta de google y disponer de conexión a internet. Además puedes compartir tus archivos con otras personas y crear el código de forma colaborativa.

Para utilizarlo debes estar logueado en tu cuenta de google e ingresar al siguiente enlace:

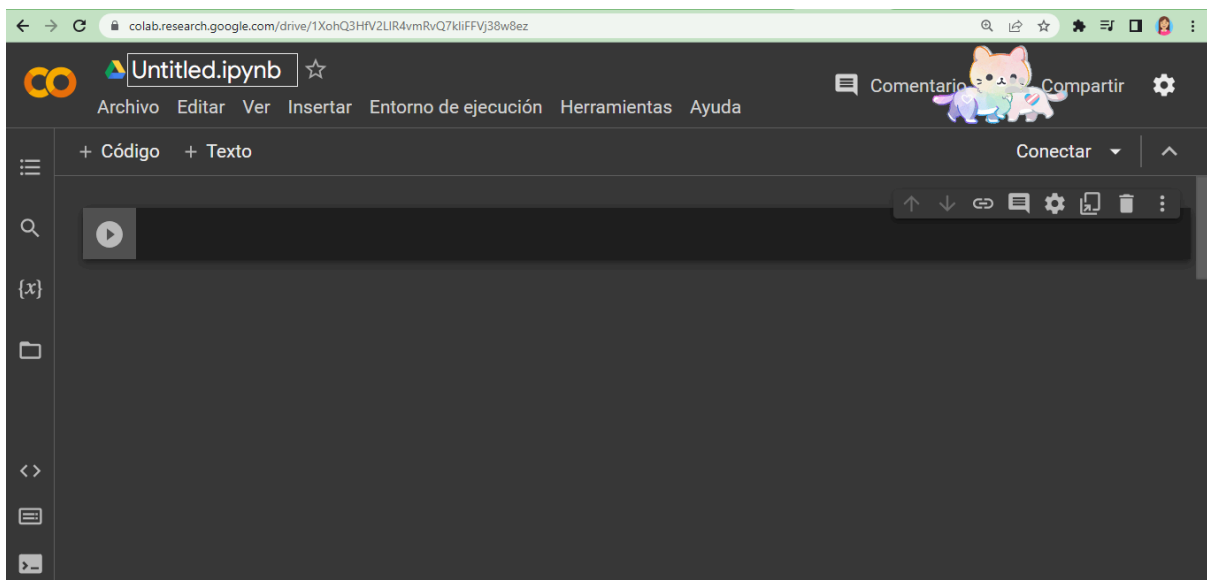
<https://colab.research.google.com/>

Cuando ingreses verás algo similar a la siguiente imagen:

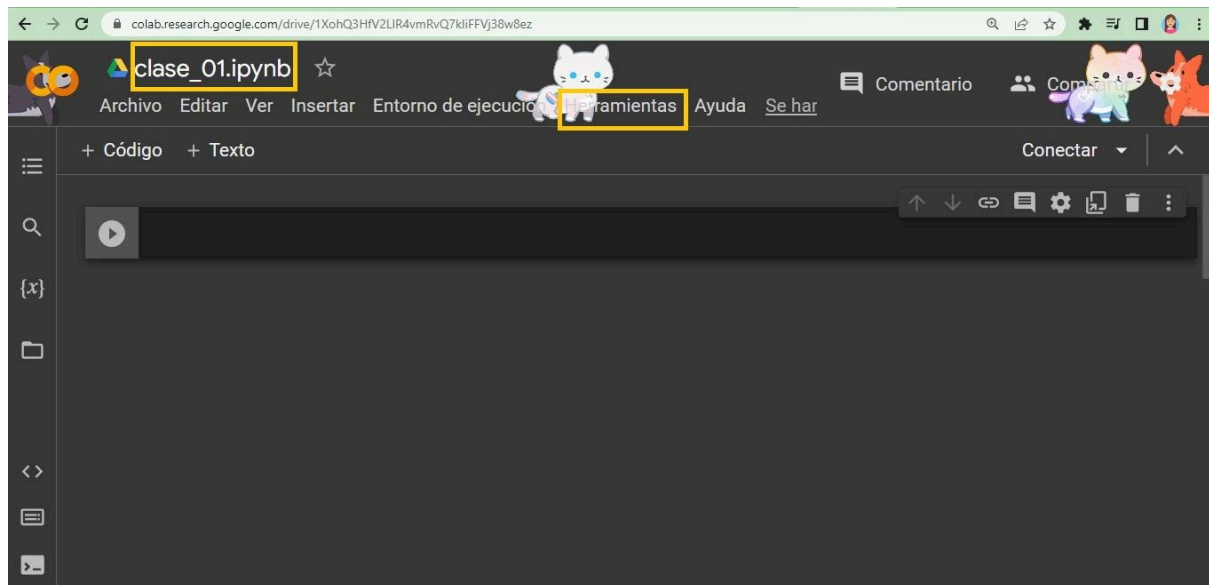


Una vez en el sitio haremos click en “Nuevo Cuaderno”.

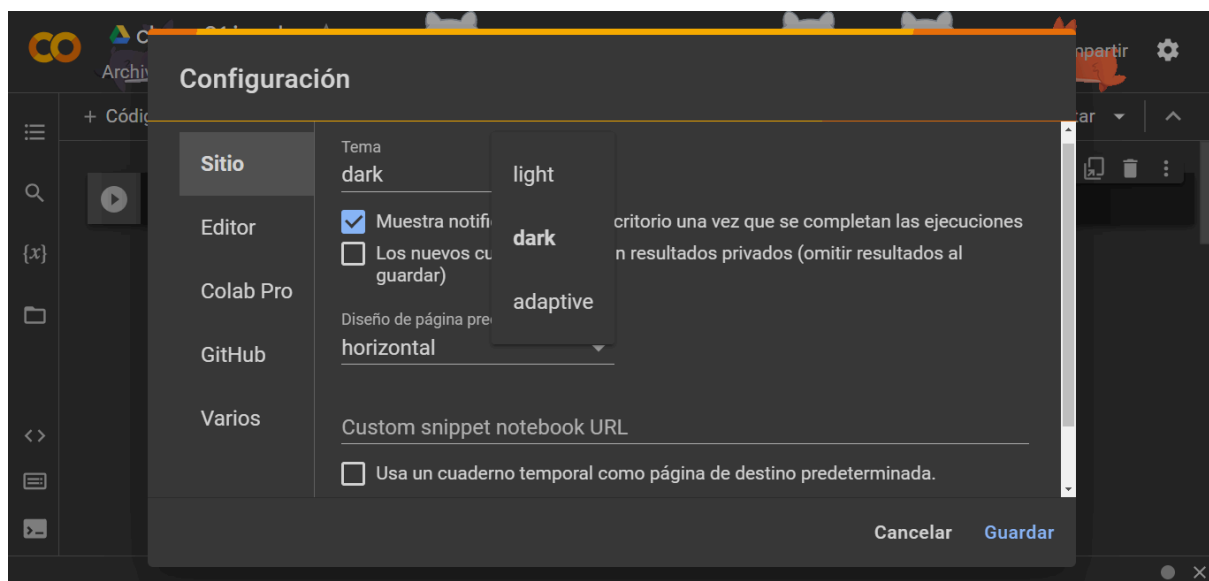
Luego verás el “cuaderno”, o notebook, donde podrás escribir el código en Python:



Haciendo click sobre el título podremos cambiar el nombre de nuestro cuaderno de Colab.



Luego podemos cambiar el modo (oscuro, claro o adaptativo) ingresando a Herramientas y luego Configuración.



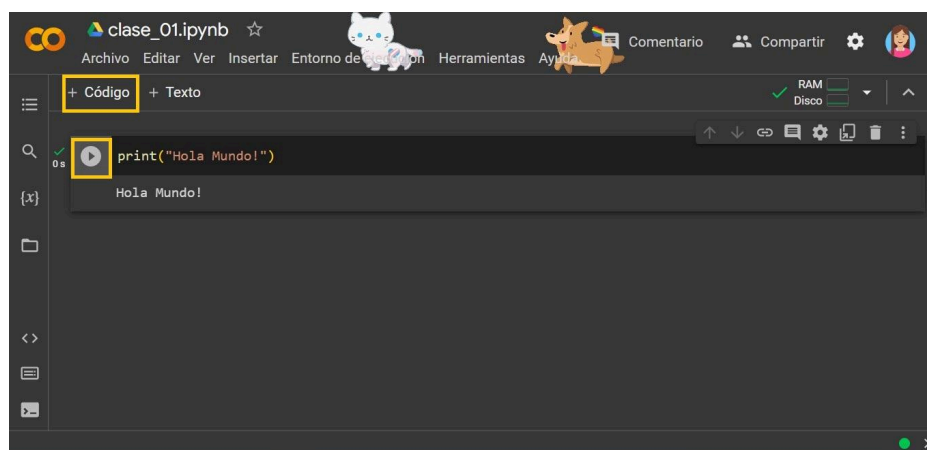


También desde Configuración, en “Varios”, podemos agregar algunos animalitos para que se paseen por encima de nuestro código y nos hagan compañía mientras programamos



Todos los cuadernos que creemos en Colab quedarán automáticamente guardados en nuestro Drive, en una carpeta llamada “Colab Notebooks”.

El código lo escribiremos en celdas, luego para ejecutarlo debemos presionar sobre el triángulo que aparece delante de la celda.



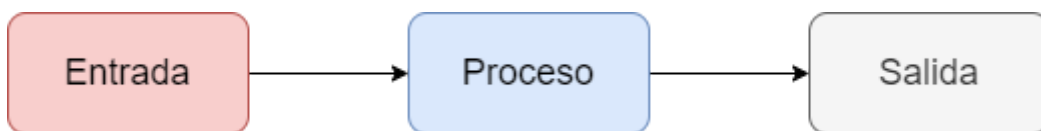
Si queremos agregar más celdas, presionamos en el botón superior “+Código”.

¡Ya estamos listos/as para trabajar con Colab!

Fundamentos de Python.

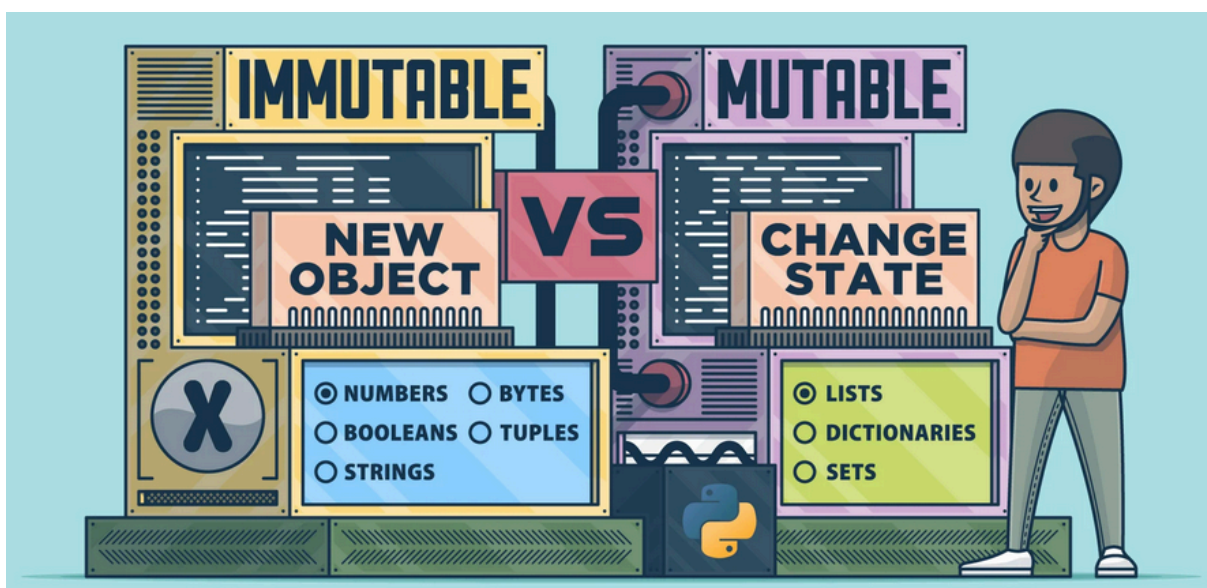
Al iniciar a programar en cualquier lenguajes de programación es necesario conocer algunos términos como *tipos de datos* y *variables*. También hablaremos de la *entrada* y *salida*.

La entrada será la información que le proporcionaremos a nuestro código. Por ejemplo, queremos saber la suma de dos números, entonces le proporcionamos los dos números. La salida será lo que nos devuelve el código luego de efectuar un proceso. En el caso anterior el proceso sería sumar, y nos devolverá el resultado de la suma.



Pero ¿Qué pasaría si en lugar de ingresar dos números ingresamos dos letras, o dos palabras? No es posible sumar dos palabras, así que obtendríamos un error. Acá es importante distinguir los distintos tipos de datos.

Tipos de Datos.





Los principales tipos de datos en Python son:

- Tipos numéricos
 - Tipo entero (int)
 - Tipo real o punto flotante (float)
 - Tipo numérico (complex) ⇒ Se usan solo para teorías físicas/matemáticas
- Tipo cadena (str)
- Tipo booleano (bool)
 - Verdadero: True
 - Falso: False
- Tipo lista (list)
- Tipo tupla (tuple)
- Tipo de datos mapas o diccionario (dict)

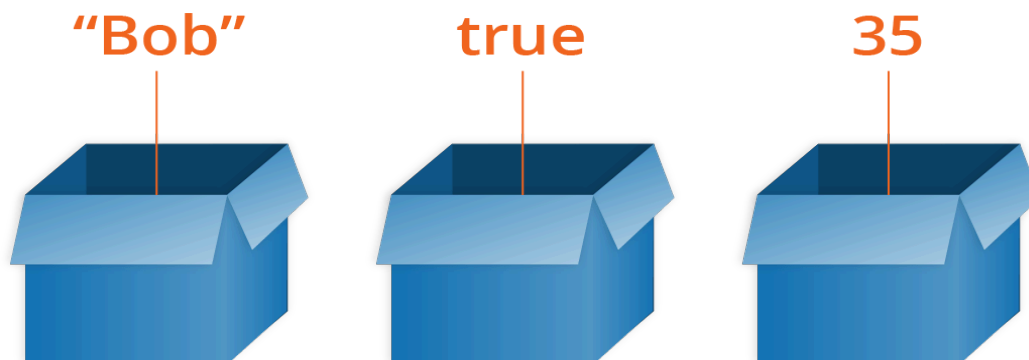
Veamos la definición de los principales tipos de datos:

Nombre	Tipo	Ejemplo	Definición	Tipo de dato
Entero	int	132	Los números enteros son aquellos que no contienen decimales, pueden ser positivos, negativos o cero.	Inmutable
Booleano	bool	True, False	Este tipo de dato se representa en lenguaje de programación como booleano y puede ser True o False.	Inmutable
Flotante	float	15e3	Tipo de dato que puede ser positivo o negativo, conteniendo uno o más decimales.	Inmutable
Cadena	str	"Hola mundo"	Permite almacenar cadenas de caracteres. Se representan entre comillas dobles.	Inmutable

Variables.

Las variables son pequeños espacios en la memoria que nos permite guardar datos (de cualquier tipo) para luego utilizarlas en el momento que necesitemos. Esto nos permite avanzar en el programa sin tener que declarar repetidas veces un mismo dato. Para guardarlas es necesario colocarles un nombre, lo que funcionará como una dirección, y cada vez que las necesitemos usar, solo tendremos que llamar a ese dato por el nombre.

Existe un tipo de variable "constante", la cual se utiliza para definir valores fijos, que no requieran ser modificados. Esto nos evitará problemas futuros, ya que nos aseguramos que su valor será siempre igual. Por ejemplo, sabemos que la variable PI será siempre 3,14.



Estos nombres pueden estar formados por letras (minúsculas y mayúsculas) de la “a” a la “z”. También pueden incluir el guión bajo, y se pueden usar números, salvo como primer carácter. Se recomienda utilizar nombres descriptivos y en minúsculas. Para nombres compuestos, separar las palabras por guiones bajos o en el formato camelCase. Antes y después del signo = , debe haber un espacio en blanco.

Mayúsculas y minúsculas son diferentes para Python. Por ejemplo, TODAS las siguientes variables son diferentes:

`nombre = 'David'`

`Nombre = 'Diego'`

`NOMBRE = 'Rosita'`

Te invito a que hagas la prueba con estas variables y crees otras para practicarlas.

Algunos ejemplos que pueden servirnos al momento de pensar nombres de variables y su descripción de lo que queremos guardar. En muchos casos será muy importante que podamos usar la imaginación para generar las variables correctas, mezclando mayúsculas y minúsculas.

Nombre	Estado	Observaciones
Altura mtr	inválido	Hay dos palabras, cuando una variable tiene que ser de una
2altura = 442	inválido	No se puede arrancar una variable con un número.
altura = 442	válido	altura y Altura son variables distintas, "altura" (con minúscula) guarda el 442 y "Altura" (con mayúscula) guarda el 25
Altura = 25		
alturaMtrs	válido	Podemos especificar un poco más con el formato camelCase.
altura_2	válido	El guión bajo aleja la palabra del número, pero sigue siendo una
nombre = "David"	válido	Las 3 variables son distintas, aunque tengan el mismo valor. Ver que puedo usar tanto comillas simples como dobles
Nombre = "David"		
NOMBRE = "David"		

Operadores básicos.

Operadores matemáticos.

Los operadores matemáticos o aritméticos se usan para calcular un valor a partir de dos o más números, incluso cambiar el signo de un número de positivo a negativo o viceversa.

Nos permiten realizar tareas como sumar, restar, multiplicar, dividir y realizar otras operaciones matemáticas básicas.

Operadores matematicos		
Operador	Descripción	Uso
+	Realiza la suma de dos o más elementos numéricos	$10 + 2 = 12$
-	Realiza la resta entre los operandos numéricos	$9 - 3 = 6$
/	Realiza División entre dos números	$15 / 3 = 5$
%	Devuelve el resto de una división	$14 \% 3 = 1$
		$16 \% 2 = 0$
		$17 \% 2 = 1$
*	Realiza la multiplicación entre dos elementos	$10 * 3 = 30$
**	Realiza la potencia de los elementos	$2 ** 3 = 8$
//	Realiza la división dando como resultado un número entero	$21 // 5 = 4$

Operadores de comparación.

Los operadores de comparación se utilizan para comparar dos valores y evaluar si una cierta relación es verdadera o falsa. Estos operadores se aplican comúnmente en estructuras condicionales y expresiones lógicas. En Python, los operadores de comparación incluyen:

Operadores de comparación		
Operador	Propósito	Ejemplo
<	Devuelve True si el primer valor es menor que el segundo valor	Valor1 < Valor2
<=	Devuelve True si el primer valor es menor o igual que el segundo valor.	Valor1 <= Valor2
>	Devuelve True si el primer valor es mayor que el segundo valor.	Valor1 > Valor2
>=	Devuelve True si el primer valor es mayor o igual que el segundo valor	Valor1 >= Valor2
==	Devuelve True si el primer valor es igual al segundo valor.	Valor1 = Valor2
!=	Devuelve True si el primer valor no es igual al segundo valor.	Valor1 != Valor2

Python utiliza estos operadores para evaluar expresiones y devolver un valor booleano, es decir, True (verdadero) o False (falso).

Cuando se utiliza un operador de comparación, la expresión se evalúa y produce un resultado lógico en función de la relación entre los valores involucrados.

Mostrar salida en pantalla.

En los programas, para que python nos muestre texto o variables hay que utilizar la función `print()`.

La función **print()** permite mostrar texto en pantalla. Las cadenas se pueden delimitar tanto por comillas dobles (") como por comillas simples ('). Puede admitir varios argumentos seguidos de cualquier tipo.

Ejemplo:

Vamos a mostrar en pantalla el famoso **¡Hola Mundo!**

Código:

##

```
print('¡Hola Mundo!')
```

##

Concatenar

Podemos crear un mensaje concatenando variables.

Hay varias formas de hacer esto en Python:

Utilizando el signo +:

Código:

##

```
nombre = "Lucas"  
print("Mi nombre es " + nombre)
```

##

Esto será útil solamente cuando tengamos que concatenar cadenas de caracteres.

¿Qué pasa si queremos concatenar cadenas con otro tipo de datos?

Código:

##

```
nombre = "Lucas"  
edad = 17
```




```
print("Mi nombre es " + nombre + " y tengo " + edad)
##
```

¡Nos devuelve un error!

Para lograr esto podemos usar el modo **f-string**. Agregamos una **f** antes de iniciar la cadena y las variables las colocamos dentro de las comillas y entre llaves:

Código:

```
##
nombre = "Lucas"
edad = 17
print(f"Mi nombre es {nombre} y tengo {edad}")
##
```

Esto nos permitirá concatenar distintos tipos de datos. ¿Qué otra forma se te ocurre?

Ingresar Datos por teclado.

El comando **input()** en Python se utiliza para leer datos del usuario y almacenarlos en una variable. Es una palabra clave fundamental para crear programas interactivos que puedan pedir información al usuario accionar en base a ella.

¿Cómo se utiliza?

La sintaxis básica de **input()** es la siguiente:

```
variable = input("Mensaje al usuario")
```

"Mensaje al usuario": Es un mensaje que se muestra al usuario para indicarle qué tipo de información debe introducir.

Veamos un ejemplo:

Código:

```
##
nombre = input("¿Cómo te llamás?")
```

```
print(f"Hola {nombre}, ¡Bienvenido!")
```

##

En este ejemplo, le pedimos al usuario que conteste la pregunta: ¿Cómo te llamas? Lo que responda el usuario será lo que se guardará en la variable nombre, que luego usamos al imprimir un mensaje.

Ingresar distintos tipos de datos

Es **importante aclarar que input() siempre devuelve una cadena de texto (str)**. Si necesitamos almacenar un dato de otro tipo, como un número entero (int) o un número decimal (float), deberíamos **convertir la cadena de texto a ese tipo de dato**.

Ejemplos de conversiones:

Código:

##

```
edad = int(input("¿Qué edad tienes?"))  
altura = float(input("¿Cuánto mides?"))
```

##

Acá el programa primero solicita al usuario que introduzca su edad y altura como cadenas de texto (por defecto). Luego, las convierte a los tipos de datos int y float respectivamente.

Comentarios.

Como cualquier otro lenguaje de programación, Python permite escribir comentarios en el código. Los comentarios son útiles para explicar por qué estamos programando algo de un modo concreto o añadir indicaciones. Te aseguro que son de utilidad cuando se retoma un programa o aplicación en el futuro

Los comentarios son ignorados por el intérprete de Python. Solo tienen sentido para los programadores.

Para añadir un comentario a tu código simplemente comienza una línea con el carácter #:



Código:

```
##
```

```
# Esto es un comentario
```

```
edad = int(input("¿Qué edad tienes?"))
```

```
altura = float(input("¿Cuánto mides?"))
```

```
##
```

Podemos agregar comentarios al finalizar la línea de código:

Código:

```
##
```

```
edad = int(input("¿Qué edad tienes?")) # ingreso la edad por teclado
```

```
altura = float(input("¿Cuánto mides?")) # ingreso la altura por teclado
```

```
##
```

Los comentarios son útiles para aclarar qué es lo que realiza nuestro código.



Desafío N° 1:

Escribir un programa simple que tome la entrada del usuario, realice algunos cálculos básicos y muestre el resultado.

[Desafío 1](#)



A large, stylized wireframe dome structure, resembling a geodesic dome, is positioned on the left side of the page. It is composed of numerous interconnected lines forming a series of triangles and polygons, creating a complex, geometric pattern. The dome is rendered in a light gray color against a dark blue background.

Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad