

Curso básico de python

Ctrl+L → limpia la pantalla en Cmder

cd .. → decirle a la consola muévete a la carpeta padre

cd Desktop\ → decirle a la consola que se mueva al escritorio

cd ejemplo → decirle a la consola que se mueva a la carpeta de ejemplo

ls → liste las carpetas

mkdir curso → crea una carpeta con el nombre curso

touch ejemplo.txt → crea un archivo llamado ejemplo de formato txt

py → inicia la consola interactiva

print() → imprime

21//5 = 4 → devuelve la parte entera de la división

21%5 = 1 → devuelve el residuo

2**2 = 4 → eleva al cuadrado

numero = 3 → dentro de la variable número voy a colocar el objeto 3

- Las variables no pueden empezar con un número, las palabras en minúsculas y se separan con guion bajo
- En Python todo es un objeto, tipos de datos: int, float, String, Boolean.
- En Python no importa si son comillas simples o dobles, entre comillas siempre hay texto

nombre = "Facundo" → variable con la cadena facundo

nombre + nombre2 → se concatenan las cadenas de caracteres

nombre * 4 → pone 4 veces la cadena

numero_decimal = 3.4 → variable con numero decimal, siempre con punto

es_estudiante = True → boolean con el valor true, primera letra en mayúscula

- Convertir datos: los números ingresados por teclado son cadenas de caracteres, por lo tanto, hay que convertirlos a int para poder operarlos
- Al convertir un punto flotante a int solo va a quedar la parte entera

numero1 = input('escribe un numero: ') → me pide que ingrese un numero por teclado

numero1 = int(numero) → convierte la variable en entero y lo vuelve a almacenar en la misma variable

int = 4.5 → devuelve la parte entera que es 4

`str(numero_decimal)` → pone el dato en un string

`es_estudiante = True`

`trabaja = False`

`es_estudiante and trabaja` → operador y, devuelve el resultado false, solo da true, cuando todo es true

`es_estudiante or trabaja` → operador o, devuelve el resultado true, solo da false cuando todo es false

`not trabaja` → le cambia el valor a los booleanos, trabaja ahora es true

`numero1 == numero2` → pregunta si ambas variables son iguales, y bota si es verdadero o falso

`numero1 > numero2` → pregunta si numero1 es mayor a numero2

`numero1 < numero2` → pregunta si numero1 es menor a numero2

`numero1 >= numero2` → mayor o igual

`numero1 <= numero2` → menor o igual

- Pass es una palabra para simbolizar la procrastinación pura
- En Python es muy importante dejar los espacios (la sangría), casi reemplaza los corchetes que se utilizan en java
- La tecla tab completa el nombre en la consola
- # para comentar líneas
- Si se selecciona todo el texto y se presiona ctrl+slash se comenta todo el texto
- Con tres comillas dobles se puede crear una cadena de varias líneas
- Def palabra para función

`nombre.upper()` → .upper es un método que lo que hace es cambiar el string a solo letras mayúsculas, no guardan el valor

`nombre.capitalize()` → .capitalize es un método que pone la primera letra en mayúscula

`nombre.strip()` → .strip quita el espacio al final de un string

`nombre.lower()` → .lower pone las letras en minúscula

`nombre.replace("a","o")` → .replace cambia todas las a por o en este caso, cambia las letras en el primer parámetro, por las que están en el segundo parámetro

`nombre[0]` → devuelve el carácter en la posición entre corchete

`len.nombre` → devuelve la cantidad de caracteres del string

`len("Hola mundo")`

- Funciones built in vienen dentro del lenguaje Python y no hay que invocarlas

`nombre[0:3]` → se obtiene todos los caracteres de antes de la posición 3

`nombre[:3]` → hace lo mismo que el de arriba

`nombre[3:]` → devuelve todos los caracteres menos los tres primeros

`nombre[3:7:2]` → devuelve del carácter 3 hasta antes del 7, pero en intervalos de dos

`nombre[::-1]` → recorre todo el string pero de atrás hacia delante

- Se crea una función principal
- Las otras funciones se crean por encima del lugar en donde se invocan
- Se dejan dos espacios entre cada función

`if __name__ == "__main__":` → es el punto de entrada de un programa de Python,

- Para definir constante se pone el nombre en mayúsculas

`a=range(1000)` → define una lista

`a=list(range(1000))` → a se convierte en todos los elementos de la lista

- Ctrl+c cancela el programa

`Import random` → aleatoriedad

`Objetos.append[False]` → se agrega el valor false a la lista

`Objetos.pop[0]` → elimina los elementos de la lista en el índice 0

`tupla = (1,2,3,4,5)` → crea una tupla, un objeto estático, no se pueden hacer las mismas operaciones que con las listas, inmutable

`tupla = [1,2,3,4,5]` → crea una lista, objeto dinámico

- En los diccionarios el contenido está definido como llaves