RESUMEN

REVISAR LINK

CURSO BASICO DE PYTHON

CURSO PLATZI

COMANDOS DE TERMINAL

CTRL + L = LIMPIAR TODO

CD.. = REGRESAR A CARPETA ANTERIOR

CD+NOMBRE DE LA CARPETA (PARA INGRESAR A LA CARPETA)

LS = CONTENIDO DE UNA CARPETA

MKDIR = CREAR UNA NUEVA CARPETA

TOUCH + NOMBRE DEL ARCHIVO.EXT = CREAR UN NUEVO ARCHIVO

RM + NOMBRE DEL ARCHIVO.EXT= ELIMINAR ARCHIVO

flecha arriba = escribe el ultimo comando utilizado

tab autocompleta lo q estes escribiendo

COMANDOS BASICOS DE PYTHON

www.youtube.com/c/ProgramacionDesdeCero

Python 3 • *hoja de referencia: lo básico*

TIPOS DE DATOS

bool = True / False
int = 5
float = 29.60
str = “¡hola!”

OPERADORES NUMÉRICOS

+	suma
-	resta
*	multiplicación
/	división
**	potencia
%	módulo
//	división entera

OPERADORES COMPARACIÓN

==	igual
!=	distinto
>	mayor
<	menor
>=	mayor o igual
<=	menor o igual

OPERADORES BOOLEANOS

and	“y” lógico
or	“o” lógico
not	negación lógica

VARIABLES

CREACIÓN

```
radio=20
```

USO

```
radio**2
```

STRINGS

CONCATENAR

```
"Hola " + "mundo"
```

OBTENER LONGITUD

```
len("chocolate")
```

CARÁCTER EN POSICIÓN 0

```
"Música"[0]
```

REBANADA

```
"margarita"[2:6]
```

ENTRADA / SALIDA DE DATOS

IMPRIMIR (MOSTRAR) DATOS

```
print("¡Hoy es un gran día!")
```

IMPRIMIR MÁS DE UN VALOR

```
h=8
print("Debemos descansar", h, "horas")
```

LEER Y GUARDAR UN STRING QUE INGRESA EL USUARIO

```
nombre=input("Tu nombre: ")
```

LEER Y GUARDAR UN NÚMERO QUE INGRESA EL USUARIO

```
edad=int(input("Tu edad: "))
```

DECISIONES

SIMPLES (DOS POSIBILIDADES)

```
n=int(input("Adivina el número: "))
if n==9:
  print("Ganaste un premio")
else:
   print("No adivinaste")
```

MÚLTIPLES (MÁS DE DOS POSIBILIDADES)

```
if n==9:
   print("Ganaste un premio")
elif n<9:
   print("Tu número es menor")
else:
   print("Tu número es mayor")
```

👉else es siempre opcional | puede haber varios elif👈

BUCLES

FIJOS CON NÚMEROS (IMPRIMIR NÚMEROS ENTRE 10 Y 99)

```
for n in range(10,100):
   print(n)
```

FIJOS CON SECUENCIA (IMPRIMIR CARACTERES DE STRING)

```
for caracter in "1, 2, 3, ¡allá vamos!":
   print(caracter)
```

CONDICIONALES

```
nombre=input("Nombre: ")
while nombre!="Luis":
   print("Esta persona no es Luis")
   nombre=input("Nombre: ")
```

FUNCIONES

```
def funcion(parametro1, parametro2):
   #cuerpo
   return #valor
```

CONTENEDORES

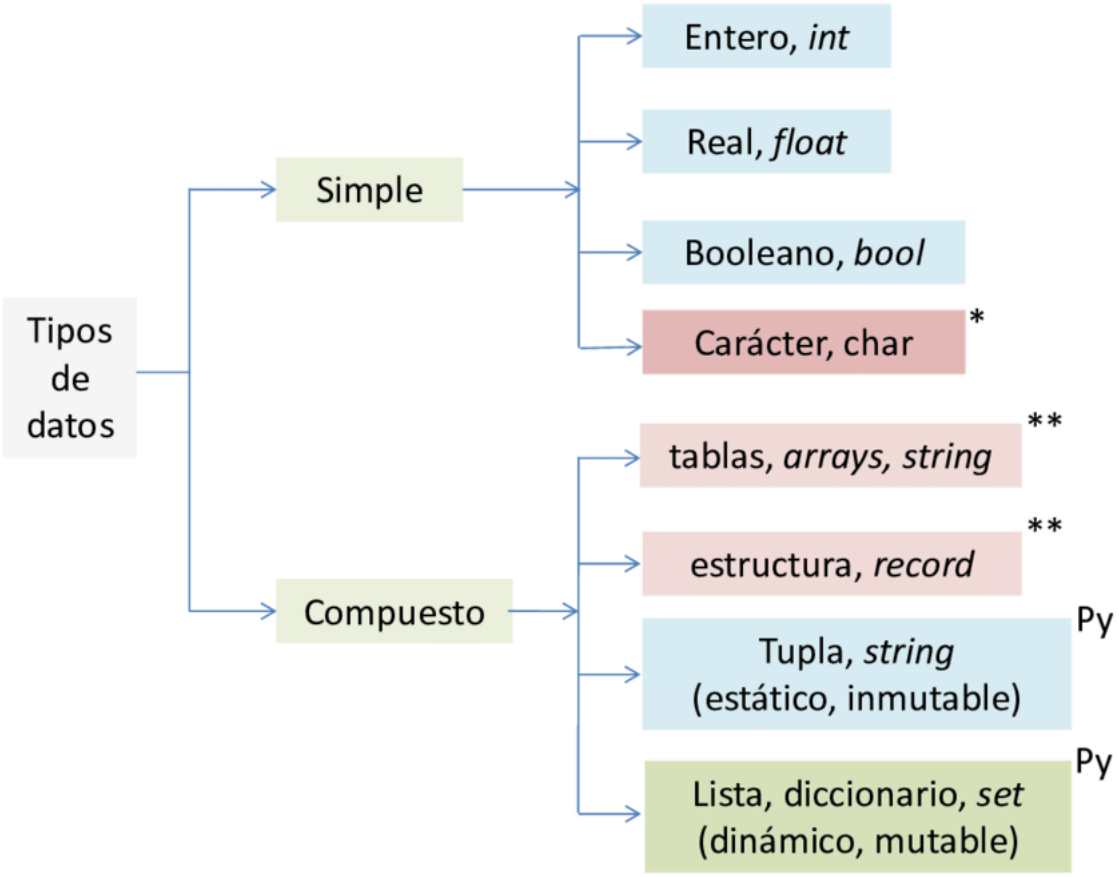
LISTAS

```
nums=[1,2,3]
nums[0]
nums.append(4)
del nums(0)
```

DICCIONARIOS

```
meses={"ene":1, "feb":2}
meses["ene"]          👈 obtener
meses["mar"]=3        👈 agregar
del meses["feb"]      👈 borrar
```

TIPOS DE DATOS



INPUTS

*Input("") para pedirle al usuario que introduzca datos.

*int() con datos o variables dentro de parentesis para convertirlo en número entero.

numero3 = int(input('Escribe un número: '))

*str() para convertir números tanto decimales como enteros a strings, QUITAR DECIMALES

OPERADORES

and para comparar si dos valores son verdaderos.

or para comparar si dos valores son falsos.

not para invertir el valor booleano.

== Compara dos valores y te dice si son iguales o no.

!= Compara dos valores y te dice sin son diferentes o no.

> Compara si es mayor que otro valor.

> Compara si es menor que otro valor.

>= igual o mayor que el valor a comparar.

<= igual o menor que el valor a comparar.

funcionan igual con strin o cualquier variable del mismo tipo

PRIMERO CODIGO "CONVERSOR DE MONEDAS"

```
<pesos = float(input("Ingrese Valor = ")) ##EL VALOR SE HACE FLOTANTE(DECIMAL)
valor_dolar = 3500
dolares = pesos / valor_dolar
dolares = round(dolares ,2) ##quitar muchos decimales dejar solo 2
print("tienes = $" + str(dolares)) ##SE IMPRIMI PERO PREVIAMENTE EL VALOR SE HACE STRIN ( POR ESO SE
CONCATENA CON UNA +)>
```

ATAJOS DE TECLADO VSC

CTLR + {} PARA COMENTAR

CONDICIONALES

```
numero = int(input("Ingrese un numero = "))
if numero > 5 :  #Condional si
    print("mayor")
elif numero == 5: # Condicional sino
    print("igual")
else :
    print("menor") # Condicional entonces
```

METODOS EN PYTHON

TAMBIEN CONOCIDO COMO **Built-in Functions**

```
variable.upper() = 'todos los caracteres en MAYÚSCULAS'
variable.capitalize() = 'solo la primera en MAYÚSCULA'
variable.lower() = 'todos los caracteres en minúscula'
variable.strip() = 'eliminar espacios basura del string'
variable.replace('caractera a cambiar', 'caracter por poner') = remplazar caracter
```

SLICES

Los slices, traducidos al español como “rebanadas”, nos permiten dividir los caracteres de un string de múltiples formas. A continuación, un ejemplo de estos:

```
nombre = "Francisco"
nombre
"Francisco"
nombre[0 : 3) //Arranca desde el primer índice hasta llegar antes del 3° índice.
"Fra"
nombre[:3] //Va desde el principio hasta antes de llegar del 3° índice. Cómo no hay ningún parámetro
en el primer lugar, se interpreta que arranca desde el principio.
"Fra"
nombre[1:7] //Arranca desde el índice 1 hasta llegar antes del 7.
"rancis"
nombre[1:7:2] //Arranca desde el índice 1 hasta llegar antes del 7, pero pasos de 2 en 2,
"rni"
nombre[1::3] //Arranca desde el índice 1 hasta el final del string (al no haber ningún 2° parámetro,
significa que va hasta el final), pero en pasos de 3 en 3.
"rcc"
nombre[::-1]  //Al no haber parámetro en los 2 primeros lugares, se interpreta que se arranca desde el
inicio hasta el final, pero en pasos de 1 en 1 con la palabra al revés, porque es -1.
"ocsicnarF"
```

EJEMPLO DE METODOS Y SLIDES

```
def palindromo (palabra) :
    palabra = palabra.replace(" " , "")#quita el espacio entre palabras reemplaza
    palabra = palabra.lower()
    palabra_invertida = palabra[::-1]# hace que las palabra se lea al revez
    if palabra == palabra_invertida : # se compara la palabra inicial con la invertida
        return True     #si es igual devulve True
    else :
        return False    # si es diferente devuelve False
        ef run():
    palabra = input("escribe una palabra : ")
    es_palindromo = palindromo(palabra)
    if es_palindromo == True:
        print("Es Palindromo")
    else :
        print("No es Palindromo")

if __name__ == '__main__' :
    run()
        ~~~
```

CICLOS

Bucles Definidos

- Con bastante frecuencia tenemos una lista de los ítems de las líneas en un archivo, es decir un conjunto finito de cosas
- Podemos escribir un bucle para ejecutar el bucle una vez para cada uno de los ítems de un conjunto utilizando la secuencia for de Python
- Estos bucles se denominan "bucles definidos" porque se ejecutan una cantidad exacta de veces
- Decimos que los "bucles definidos iteran a través de los miembros de un conjunto"

Un Bucle Definido Simple

```
for i in [5, 4, 3, 2, 1] :
    print(i)
print('Blastoff')
```

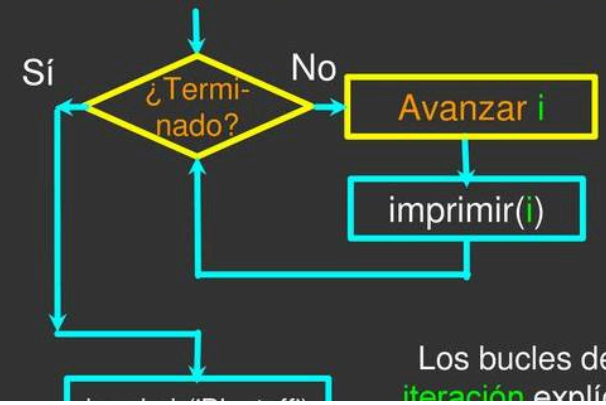
```
5
4
3
2
1
Blastoff
```

Un Bucle Definido con Cadenas

```
amigos = ['Joseph', 'Glenn', 'Sally']
for amigos in amigos :
   print('Feliz año nuevo:', amigo)
print('Terminado')
```

```
Feliz año nuevo: Joseph
Feliz año nuevo: Glenn
Feliz año nuevo: Sally
¡Terminado!
```

Un Bucle Definido Simple




```
for i in [5, 4, 3, 2, 1] :
    print(i)
print('Blastoff')
```

```
5
4
3
2
1
Blastoff
```

Los bucles definidos (bucles for) tienen variables de iteración explícitas que cambian cada vez a través del

EJEMPLO DE BUCLES WHILE

ELEVAR UN NUMERO A UNA POTENCIA DADA

```
def run():
   LIMITE = 10000000 # VALOR LIMITE DEL BUCLE (SE PONE EN MAYUSCULA XQ ES UNA CONSTANTE)
   contador = 0
   potencia_2 = 2**contador # **signo de la potenciacion
   while potencia_2 < LIMITE: # bucle while (mientras potencia_2 sea menor a limite ejecuta lo q dice
abajo)
       print("2 elevado a " + str(contador) + " es iagual a " + str(potencia_2) )
       contador+=1 # para que el valor vaya aumentando 1 cada vez q se ejecute
       potencia_2 = 2**contador
       ~~~
```

CICLO FOR

```
for contador in range(1,1001,5 ) : # 3 parametros de range (inicio, fin, pasos)
        print(contador)         # se puede leer imprimir el contador duarnte el rango de 1 a 1000 en
pasos de 5
```

RECORRIENDO CARACTERES CON FOR

```
def run():
    frase = input("Escribe una frase: ")
    for caracter in frase:  # toma todos los caracteres de la variable frase
        print(caracter.upper())
```

INTERRUMPIR CICLOS BREAK Y CONTINUE

In [270]:
```
for n in range(10000):
    print(n)
    if n == 3:
        break
```

```
0
1
2
3
```

In [271]:
```
for n in range(5):
    if n == 2:
        continue
    print(n)
```

```
0
1
3
4
```

CONTINUE

SE SALTA LOS VALORES QUE SE CONDICIONAN

```
for contador in range(1000):
        if contador % 2 != 0 : ## variable contador divide para 2 y modulo es diferenite a cero se
detiene y
            continue    #pasa al siguiente solo imprime numeros pares (se salta los numeros impares)
        print(contador)
```

BREAK

DETIENE EL CONDIGO CUANDO SE CUMPLE LA CONDICION (LO ROMPE)

```
for i in range(10000):
        if i == 5674 :
            break   # IMPRIME HASTA QUE SE CUMPLE LA CONDICION
        print(i)
```

Numero aleatorios Funciones

pagina de funciones numeros aleatorios

```
numero_aleatorio = random.randint(1,100)
    numero_elegido = int(input("ingrese numero del 1 al 100: "))
    while numero_elegido != numero_aleatorio:
        if numero_elegido < numero_aleatorio:
            print("Elige un numero mayor")
            numero_elegido = int(input("eleige otro numero "))
        else:
            print("Elige un numero menor")
            numero_elegido = int(input("eleige otro numero "))
    print("Ganaste")
```

listas en Python

LISTAS LEER

Suma(+) Concatena dos o más listas:
a=[1,2]
b=[3,4]
a + b ⟶ [1,2,3,4]

Multiplicación(*) Repite la misma lista:
a=[1,2]
a * 2 ⟶ [1,2,1,2]

Añadir elemento al final de la lista:
a=[1]
a.append(2)=[1,2]

Eliminar elemento al final de la lista:
a=[1,2]
b=a.pop()
a=[1]

Eliminar elemento dado un indice:
a=[1,2]
b=a.pop(1)
a=[2]

Ordenar lista de menor a mayor, esto modifica la lista inicial
a=[3,8,1]
a.sort() ⟶ [1,3,8]

Ordenar lista de menor a mayor, esto NO modifica la lista inicial
a=[3,8,1]
a.sorted() ⟶ [1,3,8]

Eliminar elementos de lista Elimina el elemento de la lista dado su indice
a=[1,2,3]
del a[0] ⟶ a[2,3]

Eliminar elementos de lista Elimina el elemento de la lista dado su valor
a=[0, 2, 4, 6, 8]
a.remove(6)
a=[0, 2, 4, 8]

Range creacion de listas en un rango determinado
a=(list(range(0,10,2))) ⟶crea un conteo desde 0 hasta 10 en pasos de 2 en 2.
a=[0,2,4,6,8]

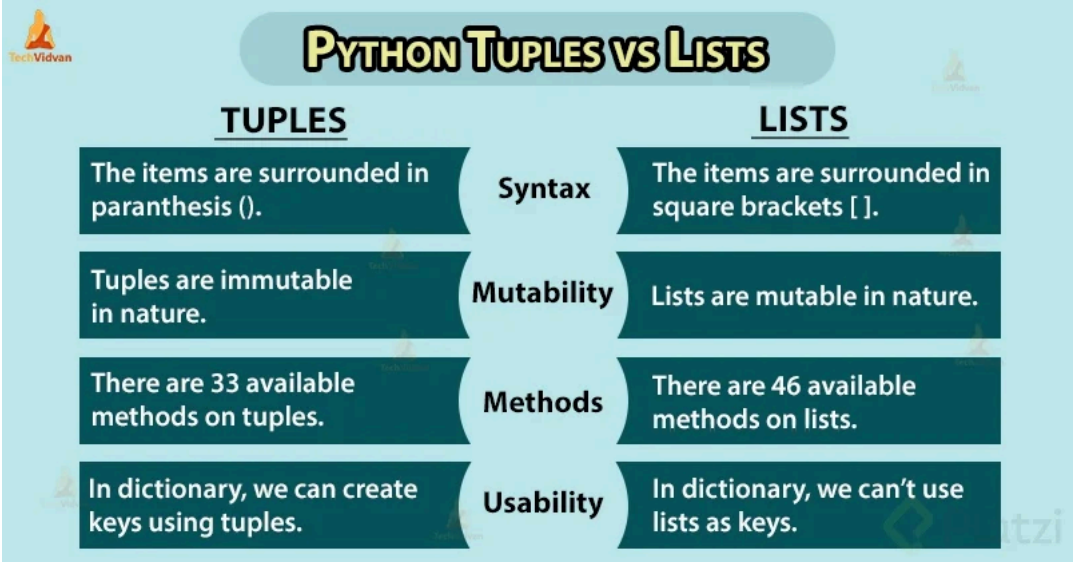
Tamaño lista len Devuelve el valor del tamaño de la lista:
a=[0,2,4,6,8]
len(a)=5

TUPLAS

SON COMO LAS LISTAS PERO NO SE PUEDEN VARIAR LOS VALORES DENTRO

SON INMUTABLES

LEER DUPLAS



count()

Este método recibe un elemento como argumento, y cuenta la cantidad de veces que aparece en la tupla.

```
valores = ("Python", True, "Zope", 5)
print "True →", valores.count(True)
```

True → 1

```
print "'Zope' →", valores.count('Zope')
```

'Zope' → 1

```
print "5 →", valores.count(5)
```

5 → 1

index()

Comparte el mismo método index() del tipo lista. Este método recibe un elemento como argumento, y devuelve el índice de su primera aparición en la tupla.

```
valores = ("Python", True, "Zope", 5)
print valores.index(True)
```

1

```
print valores.index(5)
```

3
El método devuelve un excepción ValueError si el elemento no se encuentra en la tupla, o en el entorno definido.

```
valores = ("Python", True, "Zope", 5)
print valores.index(4)
```

Traceback (most recent call last):
File "", line 1, in
ValueError: tuple.index(x): x not in tuple