Resumen Notas Capacitación de Python from DEVELOTECA and FREECODECAMP

VIDEO: Curso para aprender PYTHON

2023may28

Develoteca:

* Bajar Python de Python.org: versión 3.10.11
* Cargar el widget PYTHON Intellisense Pylance
* Abrimos la terminal y entramos a los comandos de Python >>>
* Para salir a la consola cmd, type exit()
* Digitemos PIP, que es un manejador de paquetes, tipo composer en PHP
* En el CMD, pip –version … nos va a dar que tenemos la versión 23.0.1 de PIP
* En el CMD, Python para entrar en comandos
* Podemos escribir los comandos como print(“Hola Mundo”) y nos desplegara el resultado
* Abrimos una carpeta en el escritorio capacitacionpython
* La ponemos como predeterminada en VS Code
* Programas en Python con extensión PY
* Los programas se pueden ejecutar desde la terminal CMD con el comando
  + Python ejercicio.py
* ejercicio1.py
* la indentación o sangrías son de vital importancia de Python, no dejar espacios innecesarios
* print("hola mundo")
* Ejercicio2.py
* Comentario con hashtag o numero, una sola línea o en paralelo
* # así se comenta en python, no se reflejan al usuario final
* print("hola mundo") # comentario al lado del comando
* Ejercicio3.py
* Comentarios de varias lineas
* """
* comentarios de varias lineas
* continua el comentario entre las triples comillas dobles
* notar que el comando print lleva (" ...........     ")
* """
* print("este es un comando de impresión, linea 1")
* print("este es un comando de impresión, linea 2")
* print("este es un comando de impresión, linea 3")
* ejercicio4.py
* variables sencillas
* """
* variables sencillas
* no se predefinen
* son de tipado fuerte
* """
* x = 5
* y = "develoteca"
* print(x)
* print(y)
* x = 3
* y = 4
* z = x + y
* print(z)
* """
* resultado:
* 5
* develoteca
* 7
* """
* Ejercicio5.py
* Case sensitive variables (variables sencibles)
* # case sensitive variables
* a=5
* A=4
* print(a)
* print(A)
* valora=6
* valorA=10
* print(valora)
* print(valorA)
* """
* resultado:
* 5
* 4
* 6
* 10
* """
* Ejercicio6.py
* Como identificar el tipo de una variable con TYPE
* x=5
* y=("develoteca")
* print( type(x) )    # dara un str
* print( type(y) )    # dara un int
* ejercicio7.py
* casting de variables
* """
* como identificar y convertir variables (casting)
* vamos a forzar una variable a un tipo de variable
* vamos a distinguir 3 cosas
* """
* x= str(5)
* y= int(3)
* z= float(4)
* print(x)
* print(y)
* print(z)
* ejercicio8.py
* recomendación de variables

"""

ciertas recomendaciones:

1. case sensitive

2. se pueden usar guiones bajos

3. no usar guiones medios -

4. no dejar espacios como valor a =

5. no usar numeros antes de nombre de variable 2valor =

"""

x=4

y=2

valora=5

valorb=3

valorA=6

valorB=4

valor\_a=5

\_valor\_b=3

* ejercicio9.py
* ingreso o input de información

# ingreso de informacion y concatenación de datos para imprimir

print("Escriba su nombre")

nombre=input()

print("Escriba su edad")

edad=input()

print("Bienvenido a la aplicación",nombre, ", de edad ",edad)

* ejercicio10.py
* casting a integers

"""

casting a integers

"""

print("Ingrese valor 1 ")

a=input()

print("Ingrese valor 1 ")

b=input()

c=int(a)+int(b)

print("resultado ... ",c)

"""

1

2

resultado ..... 3

"""

* ejercicio11.py
* multiple entrada de datos y casting

"""

como declarar variables multiples

"""

a,b,c= 5,3,2

print(a)

print(b)

print(3)

print(a,b,c)

print("deme 3 valores")

x,y,z = input(), input(), input()

suma=x+y+z # esta opcion solo añade 3 strings

print(suma)

suma=int(x)+int(y)+int(z) # esta opcion si suma los numeros

print(suma)

* ejercicio12.py
* operadores aritméticos

# operaciones aritmeticas, incluyendo: modulo, exponenciacion

# y floor

print("Ingrese valor 1 ")

a=int(input())

print("Ingrese valor 2 ")

b=int(input())

# ejemplo suma

suma=a+b

# ejemplo resta

resta=a-b

# ejemplo multiplicacion

mult=a\*b

# ejemplo division

divi=a/b

# ejemplo modulo o residuo

modo=a%b

# ejemplo exponenciacion

expo=a\*\*b

# ejemplo floor o residuo

floo=a//b

print("resultado de suma ... ",suma)

print("resultado de resta ... ",resta)

print("resultado de multiplicacion ... ",mult)

print("resultado de división ... ",divi)

print("resultado de modulo ... ",modo)

print("resultado de exponenciacion ... ",expo)

print("resultado de floor ... ",floo)

"""

1

2

suma=3

resta=-1

multiplicacion=2

division=0.5

modulo=1

exponeciacion=1

floor=0

"""

* ejercicio13.py
* operadores de comparación

"""

comparadores lógicos

==  >=  <=  !=  <   >    (los que van 2 signos, no se pueden invertir el orden)

"""

x=3

y=3

print(x!=y)                 # imprime   FALSE

print(x==y)                 # imprime   TRUE

print(x<=y)                 # imprime   TRUE

print(x>=y)                 # imprime   TRUE

print(x>y)                  # imprime   FALSE

print(x>y)                  # imprime   FALSE

* ejercicio14.py
* operadores de asignación

"""

Operadores de asignacion

"""

x=5

y=3

x=x+4

print(x)        # resultado = 9

x+=4

print(x)        # resultado = 13

x-=5

print(x)        # resultado = 8

x\*=2

print(x)        # resultado = 16

x/=4

print(x)        # resultado = 4.0

* ejercicio15.py
* operadores booleanos y lógicos

"""

variables de tipo booleana

solo pueden tener un TRUE o FALSE

"""

a=True

b=False

# operadores LOGICOS and o &,  or o |

print( a and b)     # resultado     FALSE

print( a or b)      # resultado     TRUE

print( a & b)       # resultado     FALSE

print( a | b)       # resultado     TRUE

print( not a )      # resultado     FALSE

* ejercicio16.py
* operadores de identidad

"""

 operadores de indentidad

"""

a=3

b=4

print(a is b)       #   resultado         FALSE

print(a is not b)   #   resultado         TRUE

* ejercicio17.py
* declaración de múltiples variables

"""

declaración de muúltiples variables

"""

x=4

y=3

z=2

print(x,y,z)

x,y,z=6,5,4

print(x,y,z)

x,y,z="uno","dos","tres"

print(x,y,z)

* ejercicio18.py

"""

declaración de múltiples variables

"""

x=4

y=4

z=4

x=y=z=5

print(x,y,z)

* ejercicio19.py
* concatenar texto y numeros

# concatenar texto

print("hola mundo")

y="hola mundo"

print(y)

x="yo"

y="soy"

z="el mundo"

print(x,y,z)        # este comando deja un espacio en medio

print(x+y+z)        # esta es una concatenación

x="yo"

y="soy el #"

z=1

print(x,y,z)        # este comando deja un espacio en medio

print(x+y+z)        # ERROR no se puede sumar un numero a un string

* ejercicio20.py
* crear una micro aplicación IMC

# crear una pequeña aplicacion IMC indice de masa corporal

# IMC = peso / (altura)2

print("Proporciona el peso en KG")

peso =input()

print("Proporciona la altura en metros")

altura =input()

peso=float(peso)

altura=float(altura)

imc=peso/(altura\*altura)

print("su IMC es de ",imc)

* ejercicio21.py
* descubra el tipo de datos cn TYPE

# descubra tipo de datos con TYPE

x=1

y=2.8

z=-321654

a=-87.7e100         # e de exponencial

print(type(x))      # resultado <class 'int'>

print(type(y))      # resultado <class 'float'>

print(type(z))      # resultado <class 'int'>

print(type(a))      # resultado <class 'float'>

* ejercicio22.py
* cast de números flotantes

# casting de datos con float, incluso strings

# en php se permite inclusive combinar caracteres, aqui no

x=float(1)

y=float(2.8)

z=float("3.5")

print(x,y,z)        # resultado   1.0 e.8 3.5

* ejercicio23.py
* manejo de strings

# manejo de strings

a= """

este es un mensaje que va a tener informacion"""

print(a)

* ejercicio24.py
* manejo de strings

# manejo de strings

a= """

este es un mensaje que va a imprimir"""

print(a)

frase="hola, Develoteca"

print(frase [0:4] )      # resultado       hola

print(frase [0:7] )      # resultado       hola, D

print(frase [8:] )       # resultado       veloteca

print(frase [7:] )       # resultado       eveloteca

print(frase [8:5] )      # resultado

                         # este ultimo no me funciona

* ejercicio25.py
* convertir texto a mayúsculas y minúsculas
* variable.upper y variable.lower

# manejo de strings, convirtiendo texto a mayusculas

frase="hola, Develoteca"

print(frase.upper())

fraseMayuscula = frase.upper()

print(fraseMayuscula)

# manejo de strings, convirtiendo texto a minusculas

frase="hola, Develoteca"

print(frase.lower())

fraseMinuscula = frase.lower()

print(fraseMinuscula)

* ejercicio26.py
* trim caracteres o eliminar caracteres en blanco STRIP

# manejo de strings, STRIP quita nulos al principio y al final

frase="  hola, Develoteca  "

print(frase.strip())    # resultado ....hola, Develoteca.....

fraseMayuscula = frase.strip()

print(frase.strip())    # resultado ....hola, Develoteca.....

* ejercicio27.py
* reemplazar caracteres o strings REPLACE

# manejo de strings, REPLACE reemplaza caracteres o strings

frase="  hola, Develoteca "

print(frase.replace("Develoteca", "Mundo"))   # resultado      hola, Mundo

fraseNueva = frase.replace("Develoteca", "Mundo")

print(fraseNueva)                             # resultado      hola, Mundo

* ejercicio28.py
* SPLIT separadores de un string a partir de un elemento, en este caso una ,
* Se podrá usar para archivos tipo CVS

# manejo de strings,  SPLIT divide strings en substrings, en este caso con elemento ","

# se puede utilizar para archovs tipo CVS

frase="hola, Develoteca, tengan un buen día"

print(frase.split(",")) # ['hola', ' Develoteca', ' tengan un buen día']

* Ejercicio29.py
* Concatenar strings

# manejo de strings,  concatenar texto

palabra1="hola"

palabra2="Presentador de Develoteca"

palabra3="tengan un buen día"

signo = ", "     # elemento de concatenación

frase= palabra1+signo+"soy "+palabra2+signo+palabra3

print(frase) # resultado ...hola, soy Presentador de Develoteca, tengan un buen día

* Ejercicio30.py
* String FORMAT, ordena la combinación de texto con variables.

# manejo de strings .... usando 'format' en string.

# ayuda a combinar string con variables de una manera más ordenada

cantidad=3

numeroID=123456

precio=59.90

precioDeVenta="El producto vale {} y son {} productos, el control número de contro es {}"

print(precioDeVenta.format(precio,cantidad,numeroID))

* Ejercicio31.py
* String FORMAT, ordena la combinación de texto con variables, usando indices

# manejo de strings .... usando 'format' en string, usando indices

# ayuda a combinar string con variables de una manera más ordenada

cantidad=3

numeroID=123456

precio=59.90

# siempre comenzar con el valor 0 en el arrray

precioDeVenta="El producto vale {1} y son {2} productos, el número de contro es {0}"

# el orden cambio en relación al ejercicio30.py

print(precioDeVenta.format(numeroID,precio,cantidad))

* Ejercicio32.py
* Contando numero de caracteres con LEN

# manejo de strings,  contar caracteres en un string

print("Escriba su fase para contar los caracteres")

frase=input()

print("la cantidad de carcteres es de ... ")

print(len(frase))

* Ejercicio33.py
* Comparación de valores con IF

# comparaciones con condicional IF

a=3

b=5

if b>a :

    print("el valor de b es mayor que el de a")

* Ejercicio34.py
* Comparación de valores con IF – ELSE

# comparaciones con condicional IF - ELSE

# no olvidar los : puntos

print("Escriba un mumero del 1 al 10")

numeroUsuario=int(input())

numero=5

if numeroUsuario > numero :

    print("tu numero es mayor que el de la PC")

else:

    print("tu numero es menor o igual que el de la PC")

* Ejercicio35.py
* Condicional IF ELIF ELSE

#

a=22

b=22

if b>a :

    print("B es Mayor que A")

elif b==a:

    print("B es IGUAl que A")

else:

    print("B es Menor que A")

* Ejercicio36.py
* Ejercicio inconcluso en el video

# condicional IF corto

a=22

b=22

# IF corto

print("B es Mayor que A") if a<b else print("B es Menor o Igual que A")

* Ejercicio37.py
* Condicional IF ELIF ELSE

# condicional

# comparar 2 edades

print("Escriba la edad de Pedro")

edadPedro=int(input())

print("Escriba la edad de Juanita")

edadJuanita=int(input())

if edadPedro>edadJuanita:

    print("Pedro es mayor que Juanita")

elif edadPedro==edadJuanita:

    print("Pedro y Juanita ienen la misma edad")

else:

    print("Juanita es mayor")

* Ejercicio38.py
* Ciclos WHILE

# ciclos WHILE

i=1

ciclos=6

while i<=ciclos:

    print(i, "vuelta")

    i+=1

* Ejercicio39.py
* Ciclos WIHILE con IF

# ciclos WHILE con IF adentro

# impresión solo de num pares

i=1

ciclos=10

while i<=ciclos:

    if (i%2)==0:

        print(i, "vuelta")

    i+=1

"""

respuesta:

2 vuelta

4 vuelta

6 vuelta

8 vuelta

10 vuelta

"""

* Ejercicio40.py
* Condicional WHILE - ELSE

# ciclos WHILE ELSE

i=1

ciclos=10

while i<=ciclos:

    print(i)

    i+=1

else:

    print("se acabo el ciclo")

"""

respuesta:

1

2

3

4

5

6

7

8

9

10

se acabo el ciclo

"""

* Ejercicio41.py
* Ejemplo con WHILE

# ejemplo ejercicio con WHILE

# aplicacion que le pida al usuario una cantidad

# determinada de numeros, mostar numeros impares

i=1

print("Ingrese cuanto ciclos desea ...")

ciclos=int(input())

while i<=ciclos:

    if (i%2)==1:

        print(i,"impar")

    i+=1

else:

    print("se acabo")

* Ejercicio42.py
* Ciclo FOR IN (muy bueno) pues funciona con un apuntador

# ciclos FOR IN

nombres=["Jose","Pedro","Josue"]

# el singular del nombre del arreglo es un apuntador

for nombre in nombres:

    print("Nombre...", nombre)

"""

Resultado:

Nombre... Jose

Nombre... Pedro

Nombre... Josue

"""

* Ejercicio43.py
* Ciclo FOR con strings

# ciclo FOR con strings

frase=("Hola!")

for letra in frase:

    print(letra)

"""

Resultado:

H

o

l

a

!

"""

* Ejercicio44.py
* Ciclo FOR in RANGE, numéricos

# ciclo FOR para RANGE, ller numeros

for numero in range(6): # ojo comienza en 0

    print(numero)

"""

resultado:

0

1

2

3

4

5

"""

* Ejercicio45.py
* Ciclo FOR in RANGE con intervalo

# ciclos FOR in RANGE, para leer un grupo de elementos

# comienza del 1 al 6

for numero in range(21,26):

    print(numero)

"""

resultado:

21

22

23

24

25

"""

* ejercicio46.py
* ciclos FOR ELSE

# ciclos FOR ... ELSE

for numero in range(1,6):

    print(numero)

else:

    print("se termino el programa")

# siempre se ejecuta el ELSE

# ojo ver que el último numero no se imprime....

"""

resultado:

1

2

3

4

5

se termino el programa

"""

* ejercicio47.py
* listas o arreglos o arrays

# listas o arrays

frutas=["manzana","banana","tomate"]

print(frutas)

for fruta in frutas:

    print(fruta)

# en los arrays se pueden combinar

numeros=[1,2,'3',4.1,"dato"]

print(numeros)

for numero in numeros:

    print(numero)

# SUBDATOS en los arrays

subdatos=[6,[5,6,7],7,8]

print(subdatos)

for subdato in subdatos:

    print(subdato)

"""

resultado:

['manzana', 'banana', 'tomate']

manzana

banana

tomate

[1, 2, '3', 4.1, 'dato']

1

2

3

4.1

dato

[6, [5, 6, 7], 7, 8]

6

[5, 6, 7]

7

8

"""

* ejercicio48.py
* función LEN en arrays

# uso de funcion en las listas, arrays

frutas=["manzana","pera","banana"]

print(frutas)

# funcion LEN

cantidadDeLista=len(frutas)

print(cantidadDeLista)

"""

resultado:

['manzana', 'pera', 'banana']

3

"""

* ejercicio49.py
* type en arrays

# mas funciones de listas

frutas=["manzana","pera","banana"]

print(frutas)

# funcion TYPE

print(type(frutas))

"""

resultado:

['manzana', 'pera', 'banana']

<class 'list'>

"""

* ejercicio50.py
* max y min de arrays

# listas e indices

numeros=[3,9,11,3,54,62,32]

print(numeros)

# muestra numero mas alto de la lista

print(max(numeros))

# muestra numero mas bajo de la lista

print(min(numeros))

* ejercicio51.py
* índices en arrays

# listas e indices

# indice  0       1        2

frutas=["uva","manzana","pera"]

print(frutas)

print(frutas[2])

print(frutas[0])

# impriiento con FOR e indice

for fruta in frutas:

    print(fruta)

* ejercicio52.py
* arrays e índices negativos

# listas e indices

# indice  0       1        2

frutas=["uva","manzana","pera"]

# indice negativo (o sea el ultimo indice)

print(frutas[-1])

""""

resultado:

pera

"""

* ejercicio53.py
* arrays y rangos de índices

# arrays (listas) e indices

# indice  0       1        2       3

frutas=["uva","manzana","pera","banana"]

print(frutas)

#  impresion por rangos de indices

print(frutas[1:3])

""""

resultado:

['uva', 'manzana', 'pera', 'banana']

['manzana', 'pera']

"""

* ejercicio54.py
* reemplazo de un valor en un array

# arrays (listas) e indices reemplazar un valor

# indice  0       1        2       3

frutas=["uva","manzana","pera","banana"]

print(frutas)

#  reemplazar un valor, en el indice

frutas[2]="piña"

print(frutas)

""""

resultado:

['uva', 'manzana', 'pera', 'banana']

['uva', 'manzana', 'piña', 'banana']

"""

* ejercicio55.py
* remplazos grupales en listas

# arrays (listas) e indices reemplazar un valor

# indice  0       1        2       3

frutas=["uva","manzana","pera","banana","fresa"]

print(frutas)

# reemplazamos un rango

print(frutas[1:3])      # primer numero es el indice inicial, y el segundo valor es la ultima columna

frutas[1:3]=["jocote","sandia"]

print(frutas)

#  reemplazar de varios valores, en un array

j=(len(frutas))

i=0

while i<j:

    frutas[i]="-"

    i+=1

print(frutas)

""""

resultado:

['uva', 'manzana', 'pera', 'banana', 'fresa']

['manzana', 'pera']

['uva', 'jocote', 'sandia', 'banana', 'fresa']

['-', '-', '-', '-', '-']

"""

* ejercicio56.py
* inserción de valores en arrays

# insertar valores en arrays

# indice  0       1        2       3       4

frutas=["uva","manzana","pera","banana","fresa"]

print(frutas)

# ocuparemos una funcion INSERT

frutas.insert(2,"piña") # se inserta entre el indice 1 y 2

print(frutas)

"""

resultado

['uva', 'manzana', 'pera', 'banana', 'fresa']

['uva', 'manzana', 'piña', 'pera', 'banana', 'fresa']

"""

* ejercicio57.py
* agregar un valor al final de un array

# anexar valores al final del array

# indice  0       1        2       3       4

frutas=["uva","manzana","pera","banana","fresa"]

print(frutas)

# ocuparemos una funcion APPEND, agregar al final

frutas.append("mora") # se inserta entre el indice 1 y 2

print(frutas)

"""

resultado

['uva', 'manzana', 'pera', 'banana', 'fresa']

['uva', 'manzana', 'pera', 'banana', 'fresa', 'mora']

"""

* ejercicio58.py
* unir 2 listas

# de 2 arrays crear uno solo

# indice  0       1        2       3       4     5

frutasDulces=["uva","manzana","pera","banana","fresa"]

frutasNodulces=["aguacate","tomate"]

print(frutasDulces)

print(frutasNodulces)

cestoFrutas=[]

cestoFrutas.extend(frutasDulces)

cestoFrutas.extend(frutasNodulces)

print(cestoFrutas)

"""

resultado:

['uva', 'manzana', 'pera', 'banana', 'fresa']

['aguacate', 'tomate']

['uva', 'manzana', 'pera', 'banana', 'fresa', 'aguacate', 'tomate']

"""

* ejercicio59.py
* eliminar datos de un array

# eliminar un elemento de un array con REMOVE

# indice  0       1        2       3       4     5

frutasDulces=["uva","manzana","pera","banana","fresa"]

print(frutasDulces)

frutasDulces.remove("uva") # por nombre

print(frutasDulces)

frutasDulces.pop(2) # por indice

print(frutasDulces)

"""

resultado:

['uva', 'manzana', 'pera', 'banana', 'fresa']

['manzana', 'pera', 'banana', 'fresa']

['manzana', 'pera', 'fresa']

"""

* ejercicio60.py
* borrado de elementos en listas con DEL

# borrado en listas

frutas=["manzana","uva","banana","fresa","pera"]

print(frutas)

del frutas[1] # borra el elemento 1 o sea "uva"

print(frutas)

"""

resultado:

['manzana', 'uva', 'banana', 'fresa', 'pera']

['manzana', 'banana', 'fresa', 'pera']

"""

* ejercicio61.py
* CLEAR en listas

# usando CLEAR en listas

marcaCoches=["mazda","ford","VW","chevrolet"]

print(marcaCoches)

marcaCoches.clear()

print(marcaCoches)cls

marcaCoches=["toyota","audi","mercedes","bmw"]

print(marcaCoches)

"""

resultado:

['mazda', 'ford', 'VW', 'chevrolet']

[]

['toyota', 'audi', 'mercedes', 'bmw']

"""

* Ejercicio62.py
* SORT

# usando SORT las listas, ordenamiento del contenido

# array de strings

marcaCoches=["mazda","ford","vw","chevrolet"]

print(marcaCoches)

marcaCoches.sort()

print(marcaCoches)

# array de numeros

numeros=[1,9,88,65,46,13,22,8,7]

print(numeros)

numeros.sort()

print(numeros)

"""

# array combinado

combinado=[1,9,88,65,7,"mazda","ford","vw"]

print(combinado)

#combinado.sort() # da un error pues eo ORDER no permite la combinación

print(combinado) # al menos que se conviertan los numeros a string

"""

"""

resultado:

['mazda', 'ford', 'vw', 'chevrolet']

['chevrolet', 'ford', 'mazda', 'vw']

[1, 9, 88, 65, 46, 13, 22, 8, 7]

[1, 7, 8, 9, 13, 22, 46, 65, 88]

"""

* ejercicio63.py
* xxxxx.sort(reverse=True) o xxxx.reverse()

# usando SORT las listas, ordenamiento invertido del contenido

# con reverse....... numeros.sort(reverse=True) o números.reserve()

# array de strings

marcaCoches=["mazda","ford","vw","chevrolet"]

print(marcaCoches)

marcaCoches.sort(reverse=True)

print(marcaCoches)

# array de numeros

numeros=[1,9,88,65,46,13,22,8,7]

print(numeros)

numeros.sort(reverse=True)

print(numeros)

"""

resultado:

['mazda', 'ford', 'vw', 'chevrolet']

['vw', 'mazda', 'ford', 'chevrolet']

[1, 9, 88, 65, 46, 13, 22, 8, 7]

[88, 65, 46, 22, 13, 9, 8, 7, 1]

"""

* ejercicio64.py
* copiar listas opción 1

# listas o arrays, copiar listas y agregar elementos

marcasCarros=["mazda","ford","toyota"]

print(marcasCarros)

listaTemporal=marcasCarros

listaTemporal.append("vw")

print(listaTemporal)

"""

resultado

['mazda', 'ford', 'toyota']

['mazda', 'ford', 'toyota', 'vw']

"""

* ejercicio65.py
* copiar listas opción 2

# listas o arrays, copiar listas y agregar elementos

marcasCarros=["mazda","ford","toyota"]

print(marcasCarros)

listaTemporal=list(marcasCarros)

listaTemporal.append("vw")

print(listaTemporal)

"""

resultado

['mazda', 'ford', 'toyota']

['mazda', 'ford', 'toyota', 'vw']

"""

* ejercicio66.py
* unión de arrays (listas o arreglos)
* alternativa a EXTEND pero aquí si se pueden unir números y strings

# unir 3 arrays o listas, deja unir strings con numeros.....+ + +

# por lo que es una mejor opción que EXTENT

# indice  0       1        2       3       4     5

frutasDulces=["uva","manzana","pera","banana","fresa"]

frutasNodulces=["aguacate","tomate"]

otroListado=[1,6,3]

print(frutasDulces)

print(frutasNodulces)

print(otroListado)

cestoFrutas=[]

cestoFrutas=frutasDulces+frutasNodulces+otroListado

print(cestoFrutas)

"""

resultado:

['uva', 'manzana', 'pera', 'banana', 'fresa']

['aguacate', 'tomate']

['uva', 'manzana', 'pera', 'banana', 'fresa', 'aguacate', 'tomate']

"""

* ejercicio67.py
* tuplas

# continuamos con arrays y TUPLAS

marcasCarros=["ford","mazda","vw"]      # array []

marcasCarrosTupla=("audi","opel")       # TUPLA  () no permite modificar,sort elementos

print(marcasCarros)

print(marcasCarrosTupla)

"""

resultado:

['ford', 'mazda', 'vw']

('audi', 'opel')

"""

* ejercicio68.py
* tupla

# TUPLAS el mismo tratamiento de un arreglo

marcasCarros=("mazda","ford","toyota")

print(marcasCarros)

print(marcasCarros[1])

print(marcasCarros[-1])

* ejercicio69.py
* modificación de tuplas(mas complicado pues no es la idea)
* conversión con TUPLE

# TUPLAS como modificar (pero no es la idea)

marcasCarros=("mazda","ford","toyota")

print(marcasCarros)

marcasCarrosTemporal=list(marcasCarros) # usar LIST

marcasCarrosTemporal.append("audi")  # OJO con sintaxis

marcasCarros=tuple(marcasCarrosTemporal) # usar TUPLE

print(marcasCarros)

"""

resultado

('mazda', 'ford', 'toyota')

('mazda', 'ford', 'toyota', 'audi')

"""

* ejercicio70.py
* desempaquetado o asignación a TUPLA

# TUPLAS como asignacion

marcasCarros=("mazda","ford","toyota")

print(marcasCarros)

# desempaquetar (o asignacion) datos de TUPLAS

(marca1,marca2,marca3)=marcasCarros  # marcaX como variables

print(marca1)

print(marca2)

print(marca3)

"""

resultado:

('mazda', 'ford', 'toyota')

mazda

ford

toyota

"""

* ejercicio71.py
* mas asignación de TUPLAS

# TUPLAS  una mejor asignacion de valores a tuplas

marcasCarros=("mazda","ford","toyota","vw","chevrolet")

print(marcasCarros)

# desempaquetar (o asignacion) datos de TUPLAS

(marca1,marca2,\*marca3)=marcasCarros  # marcaX como variables, observar \*

# en \*marca3  se almacenaran los datos posteriores..."toyota","vw","chevrolet"

print(marca1)

print(marca2)

print(marca3)

"""

resultado:

('mazda', 'ford', 'toyota', 'vw', 'chevrolet')

mazda

ford

['toyota', 'vw', 'chevrolet']

"""

* ejercicio72.py
* unión de 2 TUPLAS

# TUPLAS  union de varias tuplas

marcasCarros=("mazda","ford","toyota")

placasCarros=("CXM-123","WER-798","JHG-369")

datosCarros=marcasCarros+placasCarros

print(datosCarros)

"""

resultado:

('mazda', 'ford', 'toyota', 'CXM-123', 'WER-798', 'JHG-369')

"""

* ejercicio73.py
* duplicar contenidos TUPLAS

# TUPLAS duplica el array en sus contenidos

numeros=(4,5,6,1)

print(numeros)

nuevaTupla=numeros\*2  # duplica cantidad de datos

print(nuevaTupla)

nuevaTupla=numeros\*3  # duplica cantidad de datos

print(nuevaTupla)

"""

resultado:

(4, 5, 6, 1)

(4, 5, 6, 1, 4, 5, 6, 1)

(4, 5, 6, 1, 4, 5, 6, 1, 4, 5, 6, 1)

"""

* ejercicio74.py
* diccionarios

# diccionarios o SETS son como arrays con {  }

frutas={"zapote","sandia","jocote","guayaba"}

print(frutas)

numeros=(1,2,3,4)

print(numeros)

valores=(True,False,False)      #valores booleanos

print(valores)

valoreVariados={"zapote",2,True,20}

print(valoreVariados)

"""

resultado:

{'sandia', 'guayaba', 'zapote', 'jocote'}

(1, 2, 3, 4)

(True, False, False)

{True, 2, 20, 'zapote'}

"""

* ejercicio75.py
* add para SETS { }

# diccionarios o SETS son como arrays con {  }

frutas={"zapote","sandia"}

print(frutas)

frutas.add("jocote")    # podemos agregar mas elemnto con ADD, no APPEND.

print(frutas)

frutas.add("guayaba")   # un elemento a la vez

print(frutas)

"""

resultaod:

{'zapote', 'sandia'}

{'zapote', 'sandia', 'jocote'}

{'zapote', 'sandia', 'jocote', 'guayaba'}

"""

* ejercicio76.py
* ADD

# diccionarios o SETS son como arrays con {  }

frutas={"zapote","sandia"}

print(frutas)

frutas.add("jocote")    # podemos agregar mas elemnto con ADD, no APPEND.

print("Escriba una fruta a agregar: ")

nuevaFruta=input()

frutas.add(nuevaFruta)   #

print(frutas)

"""

resultado(favor revisen el orden en que se da todo, pues no hay un orden como en el APPEND)

{'zapote', 'sandia'}

Escriba una fruta a agregar:

granadilla

{'zapote', 'granadilla', 'jocote', 'sandia'}

"""

* Ejercicio77.py

# uniendo los diccionarios, atencion cuando {} y cuando()

# se pueden poner varios valores

frutasVerdes={"manzana","pera"}

frutasMaduras={"sandia","jocote"}

# unir diccionarios

frutasMaduras.update(frutasVerdes)

print(frutasMaduras)

"""

resultado:

{'jocote', 'manzana', 'sandia', 'pera'}

"""

* Ejercicio77B.py
* Diccionarios

# diccionarios

# nombre:valor

# nombres son claves

# valores : oscar, develoteca.com, develoteca, 39

nombres={"nombre":"Oscar","sitio":"develoteca.com","canal":"develoteca","edad":39 }

print(nombres)

print(nombres["canal"])

print(nombres["sitio"])

print(nombres.get("edad"))

"""

resultado:

{'nombre': 'Oscar', 'sitio': 'develoteca.com', 'canal': 'develoteca', 'edad': 39}

develoteca

develoteca.com

39

"""

* Ejercicio78.py