

</ Clase 04 fundamentos de la programación

/>

} /> [

Nombre: Alex Mellado
Gamboa

</ ¿Que se vio anteriormente?

Estructura [if, else, elif]

Y ejemplos de la aplicaciones

Ciclo while

Ejercicios aplicando while, condicionales

Ejercicio de tipo pep

Se vio otro ejercicio de tipo pep, con un enfoque distinto a lo que se ha visto



</ Funciones ¿Que son? />

- Extracto de código cuya finalidad es realizar una operación
- Permite mejor cohesión, y hacer código de forma separada
- Mejora la estructura general del código



</ ¿Que contiene las funciones? />

- Nombre de la operación que se desea realizar a través del operando “def”
- Argumento o parametro, lo que recibe esta función para operar
- Un retorno, las funciones deben retornar alguna cosa, de acuerdo a lo que requiera el programador



</Función en vista de código/>



Def es
como
comienza
la función

```
def verificarUnoYCero(numero):  
    if(numero == 0):  
        return 0  
    else:  
        return 1
```

Número es
el
argumento
que recibe



return o retorno
es lo que
devuelve esta
función

¿Entonces... cómo llamamos esta función?

Para llamar esta función, solo basta con referenciar por su nombre, y darle el argumento que requiere

```
#Bloque principal

numero = int(input("Ingrese un 0 o 1: "))

#Se llama a la funcion para que realice la operacion

operacion = verificarUnoYCero(numero)

print("La operacion que se muestra por pantalla es: ", operacion)
```

Estructura recomendada por los programadores al diseño

#Estructura para construir funciones

#Dominio: Que es lo que recibe la funcion, en este caso es un (numero)

#Recorrido: Booleano, es decir, 0 o 1 para este caso

#Descripción: Esta funcion verifica si se ha ingresado un 1 o 0

Para generar una función, nos **vamos a registrar** por la siguiente estructura, además de las **buenas prácticas**, garantiza legibilidad al código

Ejercicios con python

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio 1

Construir un programa que realice las operaciones como suma, resta, multiplicación, y división a través de funciones

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº2

Construir un programa en python en base a funciones que permita saber si una palabra es un palíndromo o no

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº3

Construya un programa que permita reemplazar una letra de una palabra ingresada

Ejemplo: hola

Palabra a reemplazar = o

Palabra para reemplazo = v

Palabra = hvla

Ejercicio nº4 tipo pep

Uno de los algoritmos de ordenamiento más extraños en el mundo de la programación es el algoritmo conocido como random sort o shotgun sort, el cual funciona de la siguiente forma:

- Si la **lista está ordenada**, se **entrega la lista**.
- Si la **lista no está ordenada**, se **revuelven aleatoriamente sus elementos y se revisa nuevamente**. Este algoritmo es altamente ineficiente, pero se desea conocer en la práctica cuán ineficiente es y cómo varía en la medida que la lista a ordenar crece, por lo que se le solicita a usted que construya un programa que ingrese una lista, de números enteros positivos, e indique **cuántas veces debieron reordenarse los elementos** para ordenarse de menor a mayor.

Mismo ejercicio pep... Pero con funciones

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1