**Juan José Díaz Prada**

**Quiz 2.2**

*“First, solve the problem. Then, write the code.” - John Johnson*

1. ¿Las siguientes expresiones son equivalentes? Responde y sustenta.
2. B.



No son equivalentes, en la expresión A se evalúa primero la potencia y luego se divide en 12, en la expresión B se divide primero en 12 y luego se evalúa la potencia. Esto significa que van a tener salidas distintas.

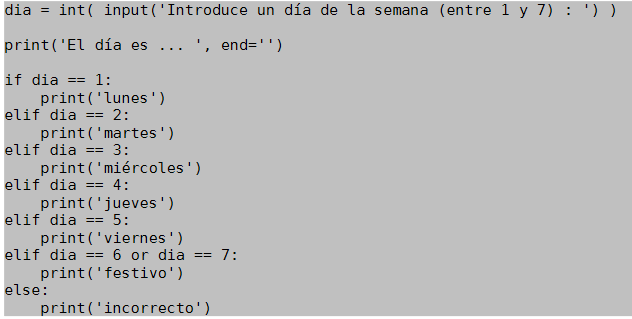
Calcula el valor de r.

|  |  |
| --- | --- |
| Código | Valor de r |
|  | 15 |
|  | 15 |
|  | -5 |
|  | 50 |
|  | 50 |
|  | 0.5 |
|  | 0.5 |
|  | 9765625 |
|  | 9765625 |

1. Asigna True o False a las siguientes expresiones según corresponda:

|  |  |
| --- | --- |
| Expresión | Valor lógico |
|  | False |
|  | False |
|  | False |
|  | True |
|  | True |
|  | True |
|  | True |
|  | False |
|  | False |

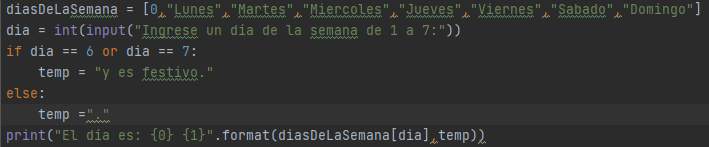
1. Observa el siguiente código en Python que a partir del número del día de la semana introducido (del 1 al 7), si dicho día es laborable escribe el nombre del día correspondiente por la pantalla, y si no escribe festivo:



¿Qué ventajas tiene la instrucción *elif*?

Después de que se evalue la condición *if* inicial como falsa, el programa se moverá a la siguiente función *elif*, y se hará esto secuencialmente hasta que una de las condiciones sea verdadera. Esto sirve para tener varios condicionales en un mismo ciclo.

¿Se te ocurre otra forma de escribir este código? Realízalo en tu entorno de programación sí es así y adjúntalo a tu respuesta.



Forma alternativa utilizando una lista. Código escrito por Sebastián Zárate.

1. Documenta el siguiente código que es una calculadora para operaciones básicas entre dos números, puedes editar sobre el siguiente código que ya tiene algunos comentarios agregados:

# calc.py  
#  
#   
#  
############################################-1-##################################################  
  
def sumar(): #Definimos la función sumar  
 x = a + b  
 print (("Resultado"), (x))  
def restar():#Definimos la función restar  
 x = a - b  
 print (("Resultado"), (x))  
def multiplicar():#Definimos la función multiplicar  
 x = a \* b  
 print (("Resultado"), (x))  
def dividir():#Definimos la función dividir  
 x = a / b  
 print (("Resultado"), (x))  
  
##########################################-2-###################################################  
while True: #  
 try: #Intentamos obtener los datos de entrada  
 a = int(input("Ingresa el primer numero: \n")) #  
 b = int(input("Ingresa el segundo numero: \n"))#  
 print (("Que cálculo quieres realizar entre"), (a), ("y"), (b), ("?\n")) #  
 op = str(input(""" #Ofrecemos las opciones de cálculo que van a llamar a las funciones  
 1- Sumar  
 2- Restar  
 3- Multiplicar  
 4- Dividir \n"""))  
 except: #En caso de error:  
 print ("Error")  
 op = '?'  
##########################################-3-##################################################  
 if op == '1':#  
 sumar()  
 break  
 elif op == '2':#Si el usuario elige opción 1 llamamos a restar  
 restar()  
 break  
 elif op == '3':#  
 multiplicar()  
 break  
 elif op == '4':#Si el usuario elige opción 1 llamamos a dividir  
 dividir()  
 break  
 else:  
 print ("""Has ingresado un numero de opcion erroneo""") #En caso que el número no  
 #se encuentre

Código mejorado:

def sumar(): # Definimos la función sumar  
 x = a + b  
 print(a, "+", b, "=", x)  
  
  
def restar(): # Definimos la función restar  
 x = a - b  
 print(a, "-", b, "=", x)  
  
  
def multiplicar(): # Definimos la función multiplicar  
 x = a \* b  
 print(a, "\*", b, "=", x)  
  
  
def dividir(): # Definimos la función dividir  
 x = a / b  
 print(a, "/", b, "=", x)  
  
  
while True: #  
 try: # Intentamos obtener los datos de entrada  
 a = int(input("Ingresa el primer numero: \n")) #  
 b = int(input("Ingresa el segundo numero: \n")) #  
 print("¿Qué cálculo quieres realizar? Escribiste:", a, "y", b, "\n")  
 op = str(input("""   
 1- Sumar  
 2- Restar  
 3- Multiplicar  
 4- Dividir \n""")) # Ofrecemos las opciones de cálculo que van a llamar a las funciones  
 except: # En caso de error:  
 print("Error")  
 op = '?'  
  
 if op == '1': # Si el usuario elige opción 1 llamamos a sumar  
 sumar()  
 break  
 elif op == '2': # Si el usuario elige opción 2 llamamos a restar  
 restar()  
 break  
 elif op == '3': # Si el usuario elige opción 3 llamamos a multiplicar  
 multiplicar()  
 break  
 elif op == '4': # Si el usuario elige opción 4 llamamos a dividir  
 dividir()  
 break  
 else:  
 print("Has ingresado un numero de opción erróneo") # En caso que el número no se encuentre

Primero se usa la función def() para definir 4 operaciones matemáticas: suma, resta, multiplicación y división. Se declaran las variables “a” y “b” como los números que escribe el usuario para operar y se le ofrecen opciones de 1 a 4 para elegir que operación desea realizar (se utiliza una variable de control “op” para esto: si es 1 es suma, si es 2 es resta, etc…). Si se escribe un valor que no sea un numero o se selecciona un valor incorrecto, se le indica al usuario que hay un error y se repite el proceso.

1. El siguiente código muestra las tablas de multiplica de cualquier número entre 1 y 10.

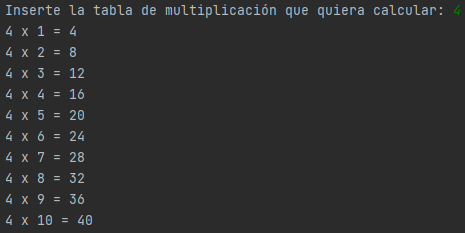
Pruébalo y mira la salida de este que se produce de esa forma porque se utiliza la función *format* para imprimir.

Anexa una captura de tu entorno con la salida del programa. Si quieres mejorarlo, adelante y por favor, comparte con todos la mejora que ideaste.

def imprimir\_tabla(numero):  
 # Se supone que las tablas llegan hasta el 10  
 LIMITE = 10  
 # Comenzar en 1  
 contador = 1  
 while contador <= LIMITE:  
 resultado = contador \* numero  
 print("{} x {} = {}".format(numero, contador, resultado))  
 # Incrementar contador para no caer en ciclo infinito  
 contador = contador + 1  
  
# Probar función  
imprimir\_tabla(1)

Código mejorado:

def imprimir\_tabla(numero):  
 # Comenzar en 1  
 contador = 1  
 while contador <= 10: # Las tablas de multiplicar son los 10 primeros valores  
 resultado = contador \* numero  
 print("{} x {} = {}".format(numero, contador, resultado))  
 # Incrementar contador para no caer en ciclo infinito  
 contador += 1  
  
  
# Se le pide al usuario escribir la tabla que quiera calcular  
imprimir\_tabla(int(input('Inserte la tabla de multiplicación que quiera calcular: ')))



Se elimina la variable “limite” y se reemplaza en el ciclo while como un 10, se reescribe la operación de la variable “contador” y finalmente el resultado de la operación se cambia para que en vez que se tenga que escribir en la entrada, el usuario lo pueda escribir el mismo cuando el programa comience a correr.