

 **EN MINUTOS
COMENZAMOS...**

 2



Programá
tu futuro



Municipalidad de
Tres de Febrero

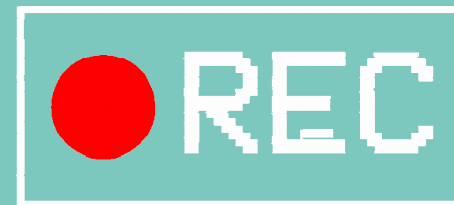




www.tresdefebrero.gov.ar/tecno3f

#ProgramáTuFuturo





VAMOS A COMENZAR A GRABAR



Programá
tu futuro



Municipalidad de
Tres de Febrero



Municipalidad de
Tres de Febrero



Programá
tu futuro



PYTHON INTERMEDIO

¡Les damos la bienvenida!





Municipalidad de
Tres de Febrero



Programá
tu futuro



TERNARIOS, *ARGS Y **KWARGS

CLASE 3



OPERADORES TERNARIOS



Programá
tu futuro

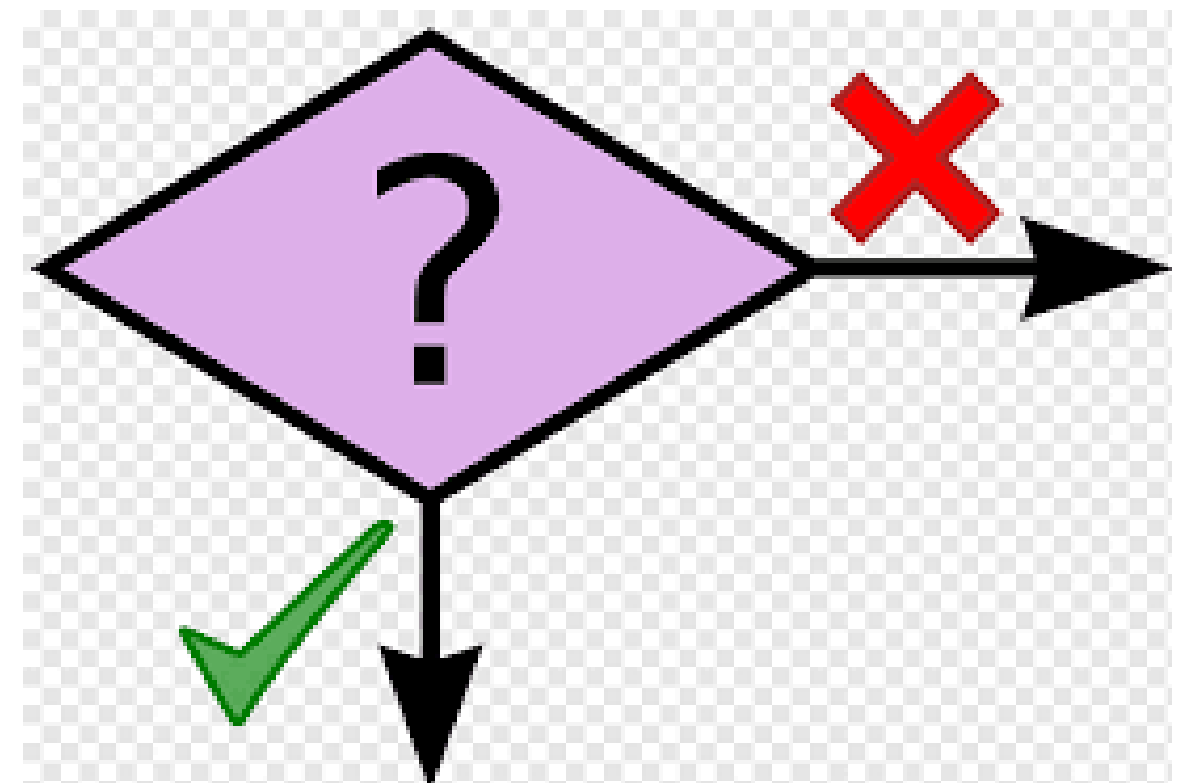


Municipalidad de
Tres de Febrero

¿QUÉ ES?

El operador ternario, también conocido como expresión condicional, es una herramienta muy útil en Python que te permite realizar una comparación y obtener un resultado en una sola línea de código.

Se asemeja a la estructura de un "if-else" pero de forma concisa..



Estructura

es una estructura bastante sencilla

Valor Verdadero **if** Condición **else** Valor Falso

Ejemplos

Como podemos ver, los ternarios se pueden ejecutar en el mismo **print**

```
edad = 18

print("Es mayor de edad" if edad >= 18 else "Es menor de edad")
```

También podemos guardar el resultado en una variable para mostrarlo cuando necesitemos

```
mostrar_edad = "Es mayor de edad" if edad >= 18 else "Es menor de edad"

print(mostrar_edad)
```

Anidando

Al igual que los condicionales, podemos anidar los operadores ternarios

```
a = 1
b = 3
c = 2

numero_mayor = a if a > b and a > c else b if b > c else c

print(numero_mayor)
```

Que valor se mostrara por consola ?

Manden sus respuestas al chat

Respuesta

Para saber la repuesta interpretamos la sentencia anterior asi:

1 si 1 es mayor a 3 y 1 mayor a 2 , sino , 3 si 3 es mayor a 2 , sino, 2

El valor que se mostrara es 3

1 si 1

* ARGS



Programá
tu futuro



Municipalidad de
Tres de Febrero

¿QUÉ ES?

El Los *ARGS (que no necesariamente deben llamarse así), permiten pasar un número variable de argumentos.

¿COMO USARLO?

Lo mas importante es respetar el *

Veamos un ejemplo

Ejemplo

```
def test_var_args(f_arg, *argv):  
    print("primer argumento normal:", f_arg)  
    print(argv) #esto regresa una tupla  
    for arg in argv:  
        print("argumentos de *argv:", arg)  
  
test_var_args('python', 'foo', 'bar')
```


***** KWARGS**



Programá
tu futuro



Municipalidad de
Tres de Febrero

¿QUÉ ES?

Al igual que con los *ARGS lo **KWARGS permiten pasar un numero variable de argumentos, con la diferencia , que en este caso se pasa key y value.

¿COMO USARLO?

Lo mas importante es respetar el **

Veamos un ejemplo

Ejemplo

```
def saludame(**kwargs):  
    print(kwargs) #esto nos regresa un diccionario  
    for key, value in kwargs.items():  
        print("{0} = {1}".format(key, value))  
  
saludame(nombre = "Gabriel", apellido = "Gomez", provincia = "Buenos Aires")
```

¿Como usarlos?

Dependerá mucho de los requisitos de tu programa, pero uno de los usos más comunes es para crear decoradores para funciones. También puede ser usado para **monkey patching**, lo que significa modificar código en tiempo de ejecución.

Considera por ejemplo que tienes una clase con una función llamada `get_info` que llama a una API que devuelve una determinada respuesta

PARA PRACTICAR



Programá
tu futuro



Municipalidad de
Tres de Febrero

Practica.. Practica.. Practica

Calcular el mayor de dos números ingresados por teclado usando un operador ternario

Buscar una palabra en una lista ingresada por teclado usando args y un operador ternario

Determinar si un número es par o impar

Calcular el promedio de una lista de números usando args y un operador ternario

Imprimir un mensaje de error si no se pasan suficientes argumentos

¿PREGUNTAS?



Programá
tu futuro



Municipalidad de
Tres de Febrero

¡MUCHAS GRACIAS!



Programá
tu futuro



Municipalidad de
Tres de Febrero