


Reto 1: Contador simple

Descripción: Muestra los números del 1 al 10 usando un bucle for.

Código:

```
UT3 >  Reto1ContadorSimple.py > ...  
1  
2     for numero in range(1,11):  
3         print(numero, end=" ")
```


Salida por terminal:

```
PS F:\tomas\Documents\Segundo  
Reto1ContadorSimple.py  
1 2 3 4 5 6 7 8 9 10
```

Reto 2: Cuenta atrás

Descripción: Muestra los números del 10 al 1 usando range() con paso negativo

Código:

```
UT3 >  Reto2CuentaAtras.py > ...  
1     for numero in range(10, 0, -1):  
2         print(numero, end=" ")
```

Salida por terminal:

```
PS F:\tomas\Documents\Segundo  
Reto2CuentaAtras.py  
10 9 8 7 6 5 4 3 2 1
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Reto 3: Suma de los primeros 100 números

Descripción: Usa un bucle for para sumar del 1 al 100.

Código:

```
2 suma = 0
3
4 for number in range(1,101):
5     suma+=number
6
7 print(f"Resultado de la suma: {suma}")
```

Salida por terminal:

```
PS F:\tomas\Documents\SegundoD
00.py"
Resultado de la suma: 5050
```

Reto 4: Validación de entrada

Descripción: Pide una contraseña hasta que el usuario escriba la correcta

Código:

```
1
2 contra = "casa"
3 usuario = input("Introduce la contraseña: ")
4
5 while(usuario!=contra):
6     usuario = input("Error. Vuelve a introducir la contraseña: ")
7
8 print("Has acertado.")
```

Salida por terminal:

```
Introduce la contraseña: jose
Error. Vuelve a introducir la contraseña: arbol
Error. Vuelve a introducir la contraseña: casa
Has acertado.
```

Reto 5: Números pares del 1 al 20

Descripción: Se meterán en un array los números pares del 1 al 20.

Código:

```
2 pares = []
3
4 for numero in range(1,21):
5     if(numero%2==0):
6         pares.append(numero)
7
8 print("Números pares del 1 al 20: ")
9 for numero in pares:
10     print(numero, end=" ")
```

Salida por terminal:

```
Números pares del 1 al 20:
2 4 6 8 10 12 14 16 18 20
```

Reto 6: Busca el primer múltiplo de 7 mayor que 100

Descripción: Se irá multiplicando la tabla del 7 hasta que el número sea mayor que 100 y se mostrará por pantalla.

Código:

```
2 i = 1
3 resultado = 0
4
5 while resultado < 100:
6     resultado = 7 * i
7     i+=1
8
9 print(f"Primer multiplo de 7 mayor a 100: {resultado}")
```

Salida por terminal:

```
Primer multiplo de 7 mayor a 100: 105
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Reto 7: Contador de vocales

Descripción: Se le pedirá al usuario una palabra y veremos cuántas vocales tiene dicha palabra.

Código:

```
1  vocales = "aeiouAEIOU"
2
3  usuario = input("Introduce una palabra: ")
4
5  contadorVocales = 0
6
7  for letra in usuario:
8      if letra in vocales:
9          contadorVocales+=1
10
11 print(f"La palabra '{usuario}' tiene {contadorVocales} vocales.")
```

Salida por terminal:

```
Introduce una palabra: Electricista
La palabra 'Electricista' tiene 5 vocales.
```

Reto 8: Menú interactivo

Descripción: Se creará un menú simple que permite 3 opciones. Dicho menú se cerrará cuando el usuario elija la 3a opción (salir).

Código:

```
1  def saludar():
2      print("Saludos, portate bien.")
3
4  def sumar(numero1, numero2):
5      return numero1 + numero2
6
7  seguir = True
8
9  while(seguir):
10     usuario = int(input("Bienvenido al menu interactivo. Qué desea:\n1. Saludar \n2. Sumar \n3. Salir\n-----\n"))
11
12     if usuario==1:
13         saludar()
14     elif usuario==2:
15         num1 = int(input("Introduce numero 1: "))
16         num2 = int(input("Introduce numero 2: "))
17         print(f"Resultado de la suma: {int(sumar(num1, num2))}")
18     elif usuario == 3:
19         print("Hasta luego.")
20         seguir = False
21     else:
22         print("Error. Introduce una opción válida.")
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Salida por terminal:

```
Bienvenido al menu interactivo. Qué desea:
```

```
1. Saludar
```

```
2. Sumar
```

```
3. Salir
```

```
-----
```

```
1
```

```
Saludos, portate bien.
```

```
Bienvenido al menu interactivo. Qué desea:
```

```
1. Saludar
```

```
2. Sumar
```

```
3. Salir
```

```
-----
```

```
2
```

```
Introduce numero 1: 5
```

```
Introduce numero 2: 10
```

```
Resultado de la suma: 15
```

```
Bienvenido al menu interactivo. Qué desea:
```

```
1. Saludar
```

```
2. Sumar
```

```
3. Salir
```

```
-----
```

```
4
```

```
Error. Introduce una opción válida.
```

```
Bienvenido al menu interactivo. Qué desea:
```

```
1. Saludar
```

```
2. Sumar
```

```
3. Salir
```

```
-----
```

```
3
```

```
Hasta luego.
```

Reto 9: Bucle con else

Descripción: Recorre una lista de números y busca el 0. Si no lo encuentra, muestra un mensaje con else.

Código:

```
lista = [1, 2, 0, 4, 10, 12, 20, 5, 72]

for numero in lista:
    if numero == 0:
        print("¡Encontrado! El 0 se encuentra en la lista.")
        break

    else:
        print("No se encontró el 0 en la lista.")
```

Salida por terminal:

```
¡Encontrado! El 0 se encuentra en la lista.
```

Si quito el 0 de la lista:

```
lista = [1, 2, 4, 10, 12, 20, 5, 72]
```

```
No se encontró el 0 en la lista.
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Reto 10: Procesar una lista de alumnos

Descripción: Dada una lista de nombres, muestra solo los que empiezan por vocal.

Código:

```
1 lista_Alumnos = {"Juan", "Pepe", "Ander", "Julio", "Amador", "Eduardo", "Ulises"}
2 vocales = "aeiouAEIOU"
3
4 for alumno in lista_Alumnos:
5     if alumno[0] in vocales:
6         print(alumno)
7
```

Salida por terminal:

```
Amador
Ulises
Eduardo
Ander
```

Reto 11: Simulación de intentos de login

Descripción: Permite 3 intentos para introducir usuario y contraseña correctos. Se utilizan variables auxiliares booleanas para que el flujo del programa sea el adecuado. El programa para cuando se cometan 3 errores en cualquiera de las peticiones (usuario o contraseña) y también cuando el login sea válido.

Código:

```
1  oportunidades = 3
2
3  usuarioValido = "Tomas"
4  contrasenaValida = "12345"
5
6  condicion1 = False
7  condicion2 = False
8
9  while(oportunidades > 0):
10     usuario = input("Introduce el usuario: ")
11
12     if usuario == usuarioValido:
13         print("Usuario válido")
14         condicion1 = True
15     else:
16         print("Usuario incorrecto.")
17         oportunidades -=1
18
19     if condicion1 == True:
20         contraUsuario = input("Introduce la contraseña: ")
21         if contraUsuario == contrasenaValida:
22             print("Contraseña correcta.")
23             condicion2 = True
24         else:
25             print("Contraseña errónea.")
26             oportunidades -=1
27
28
29     if condicion1 and condicion2 == True:
30         print("Sesión iniciada con éxito")
31         break
32
33 if oportunidades == 0:
34     print("Te quedaste sin intentos")
35
```


Tomás Pérez Carrillo

2º DAM

21-11-2025

Salida por terminal:

Caso 1 (Login correcto):

```
Introduce el usuario: Tomas
Usuario válido
Introduce la contraseña: 12345
Contraseña correcta.
Sesión iniciada con éxito
```

Caso 2 (Usuario erróneo):

```
Introduce el usuario: usu
Usuario incorrecto.
Introduce el usuario: usu
Usuario incorrecto.
Introduce el usuario: usu
Usuario incorrecto.
Te quedaste sin intentos
```

Caso 3 (Contraseña errónea y usuario válido):

```
Introduce el usuario: Tomas
Usuario válido
Introduce la contraseña: 123
Contraseña errónea.
Introduce el usuario: Tomas
Usuario válido
Introduce la contraseña: 123
Contraseña errónea.
Introduce el usuario: Tomas
Usuario válido
Introduce la contraseña: 123
Contraseña errónea.
Te quedaste sin intentos
```

Reto 12: Generador de contraseñas

Descripción: Genera 5 contraseñas aleatorias de 8 caracteres usando un bucle. He utilizado dos librerías (random,string) para crearlas aleatoriamente.

Código:

```
1  import random
2  import string
3
4  listaContra = []
5
6  tamanoContra = 8
7
8  letras = string.ascii_letters + string.digits
9
10 for contra in range(0,5):
11     contra = "".join(random.choices(letras, k=tamanoContra))
12     listaContra.append(contra)
13
14 print("Lista de 5 contraseñas: ")
15 for contrasenias in listaContra:
16     print(contrasenias)
17
```

Salida por terminal:

```
Lista de 5 contraseñas:
GWbS7jTi
Di1AcMRA
6z7HQyrw
v38QMTNl
EstJoV4j
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Reto 13: Filtrar archivos por extensión

Descripción: Filtrar archivos por extensión

Código:

```
1  mi_lista = {"hola.txt", "resumen.pdf", "texto.txt", "musica.mp4", "python.py", "prueba.pdf", "kalewi.pdf"}
2
3
4  buscar = "pdf"
5
6  for archivo in mi_lista:
7      partesDoc = archivo.split(".")
8      extension = partesDoc[1]
9      if extension == buscar:
10         print(archivo)
11
12
13
```

Salida por terminal:

```
PS F:\tomas\Documen
kalewi.pdf
resumen.pdf
prueba.pdf
```

Reto 14: Contador de palabras

Descripción: Pide una frase y cuenta cuántas palabras contiene

Código:

```
fraseUsuario = input("Introduce una frase: ")

fraseSinEspacio = fraseUsuario.split()

contadorPalabras = len(fraseSinEspacio)

print(f"En tu frase hay: {contadorPalabras} palabras")
```

Salida por terminal:

```
Introduce una frase: hola soy pepico el de los palotes
En tu frase hay: 7 palabras
```

Reto 15: Número primo

Descripción: Pide un número y determina si es primo usando un bucle

Código:

```
2  numero = int(input("Introduce un número: "))
3
4  es_primo = True
5
6  if numero < 2:
7      es_primo = False
8  else:
9      for i in range(2, numero):
10         if numero % i == 0:
11             es_primo = False
12             break
13
14  if es_primo:
15      print(f"El {numero} ES PRIMO")
16  else:
17      print(f"El {numero} NO es primo")
18
```

Salida por terminal:

```
Introduce un número: 1
El 1 NO es primo
PS F:\tomas\Documents\Segund
Introduce un número: 2
El 2 ES PRIMO
PS F:\tomas\Documents\Segund
Introduce un número: 7
El 7 ES PRIMO
PS F:\tomas\Documents\Segund
Introduce un número: 5
El 5 ES PRIMO
PS F:\tomas\Documents\Segund
Introduce un número: 12
El 12 NO es primo
```

Reto 16: Adivina el número

Descripción: Juego donde el usuario tiene que adivinar un número secreto entre 1 y 100.

Código:

```
numeroSecreto = 21

acierto = False

while(acierto==False):
    numeroUsu = int(input("Introduce un número del 1 al 100: "))
    if numeroSecreto == numeroUsu:
        print("Has acertado.")
        acierto = True
```

Salida por terminal:

```
Introduce un número del 1 al 100: 70
Introduce un número del 1 al 100: 12
Introduce un número del 1 al 100: 45
Introduce un número del 1 al 100: 21
Has acertado.
```

Reto 17: Conversor de temperaturas

Descripción: Muestra una tabla de conversión de °C a °F del 0 al 100 en pasos de 10.

Código:

```
1  print("    °C    |    °F")
2  print("-----")
3
4
5  for c in range(0, 101, 10):
6
7      # Fórmula: (C * 1.8) + 32
8      f = (c * 1.8) + 32
9
10     print(f"    {c}\t|    {f}")
```

Tomás Pérez Carrillo

2º DAM

21-11-2025

Salida por terminal:

°C		°F

0		32.0
10		50.0
20		68.0
30		86.0
40		104.0
50		122.0
60		140.0
70		158.0
80		176.0
90		194.0
100		212.0

Reto 18: Simulación de cajero automático

Descripción: Pide un importe y muestra cuántos billetes de cada tipo se necesitan (500, 200, 100...).

Código:

```
2  importe = int(input("Introduce un importe para desglosarlo en billetes: "))
3
4  billetes = [500, 200, 100, 50, 20, 10, 5]
5
6  print(f"Desglose para {importe} euros:")
7
8  for valor in billetes:
9
10     cantidad = importe // valor
11
12     if cantidad > 0:
13         print(f"{cantidad} billete(s) de {valor} €")
14
15     importe = importe % valor
16
17 if importe > 0:
18     print(f"Sobran {importe} € en monedas.")
19
```

Salida por terminal:

```
Introduce un importe para desglosarlo en billetes: 1250
Desglose para 1250 euros:
2 billete(s) de 500 €
1 billete(s) de 200 €
1 billete(s) de 50 €
PS F:\tomas\Documents\SegundoDAM\Asignaturas\Python> python
Introduce un importe para desglosarlo en billetes: 40
Desglose para 40 euros:
2 billete(s) de 20 €
```

Reto 19: Animación de carga

Descripción: Simula una barra de carga con print() y time.sleep().

Código:

```
import time

barraCarga = ""

print("Cargando...")
for i in range(40):
    barraCarga += "#"

    print(f"\r[{barraCarga}]", end="")

    time.sleep(0.2)

print("Completado.")
```

Salida por terminal:

```
Cargando...
[#####]Completado.
```