



Pensamiento Computacional

Decisiones
Manejo de errores

Booleanos, comparaciones y expresiones condicionales simples

¿Cómo programamos decisiones?



Valores Booleanos

- Recuerdas el tipo de dato Boolean.

boolean

- true
- false

- Lo utilizamos cuando evaluamos operadores Relacionales o Lógicos.

Signo	Significado
==	Igual
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
!=	Diferente

Operador	Significado
and	¿Recuerdas las tablas de verdad?
or	
not	

Tablas de verdad

AND

A	B	A and B
T	T	T
T	F	F
F	T	F
F	F	F

OR

A	B	A or B
T	T	T
T	F	T
F	T	T
F	F	F

NOT

A	not A
T	F
F	T

Expresiones booleanas

Operador	Expresión	Resultado
And	$(4 > 2) \text{ and } (5 > 8)$	False
Or	$(12 > 2) \text{ or } (23 < 1)$	True
Not	$\text{not}(45 < 5)$	True

Expresiones relacionales

Símbolo	Significado	Ejemplo	Resultado
>	Mayor que	$12 > 7$	True
<	Menor que	$8 < 12$	True
>=	Mayor o igual que	$4 >= 5$	False
<=	Menor o igual que	$12 <= 12$	True
==	Igual que	$5 == 7$	False
!=	Distinto que	rojo != verde	True

Uso de variables

```
z = True
```

```
x = ( 5 == 2)
```

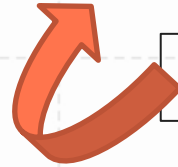
```
Y = (z or x)
```

¿Cuál es el valor final de Y?

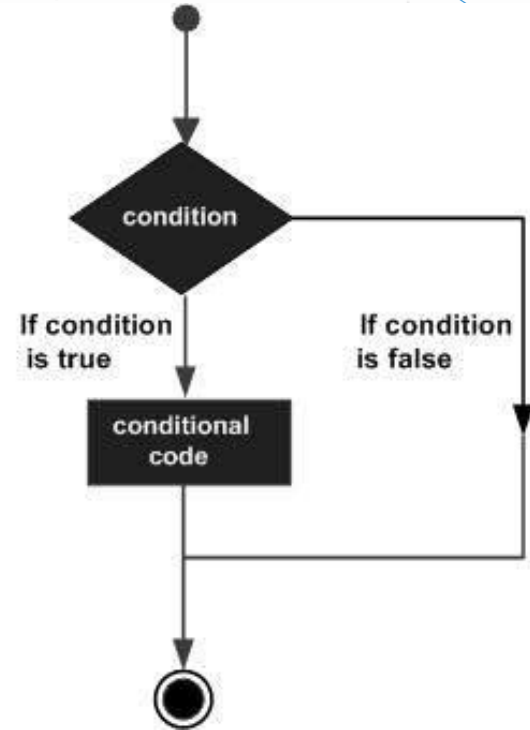
Condicionales

- Al elaborar programas es muy útil checar alguna condición y cambiar el comportamiento del programa de acuerdo al cumplimiento o no de la misma.
- En Python tenemos la sentencia condicional if

```
if x>0:  
    print (x, "es un número positivo")
```



Observa que el código debajo del IF está IDENTADO con 4 espacios



¿Cuándo uso un condicional?

Cuando la ejecución del programa depende de que una condición se cumpla.

- Ejemplo: Realiza un programa para indicar si el usuario es mayor de edad.

Entrada	Proceso	Salida
edad	<ol style="list-style-type: none">1. Solicitar al usuario su edad2. Si la edad es mayor o igual a 18 el usuario es mayor de edad, de lo contrario, es menor de edad.	Imprimir el mensaje de acuerdo con la condición que se haya cumplido.

Algoritmo

1. Pedir la edad al usuario
2. Guardarlo en la variable `edad`
3. Si `edad` es mayor o igual a 18
 Imprimir "Eres mayor de edad"
sino
 imprimir "Eres menor de edad"

Pseudocódigo

inicio

```
imprime "Dame tu edad"
lee edad
si (edad >= 18)
    imprime "Eres mayor de edad"
sino
    imprime "Eres menor de edad"
```

fin

Python

edad.py - /Users/AnaRaquel/OneDrive-Tec/OneDrive - Instituto Tecnológico y de...

```
#Este programa identifica si el usuario es mayor de edad
edad=int(input("dime ¿cuantos años tienes? "))
if edad>=18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
```

A programar



Ejercicio:

Desarrolla el código para resolver los siguientes problemas.

1. Par o impar

- Escribe un programa que identifique si un número es par o impar

2. Múltiplo

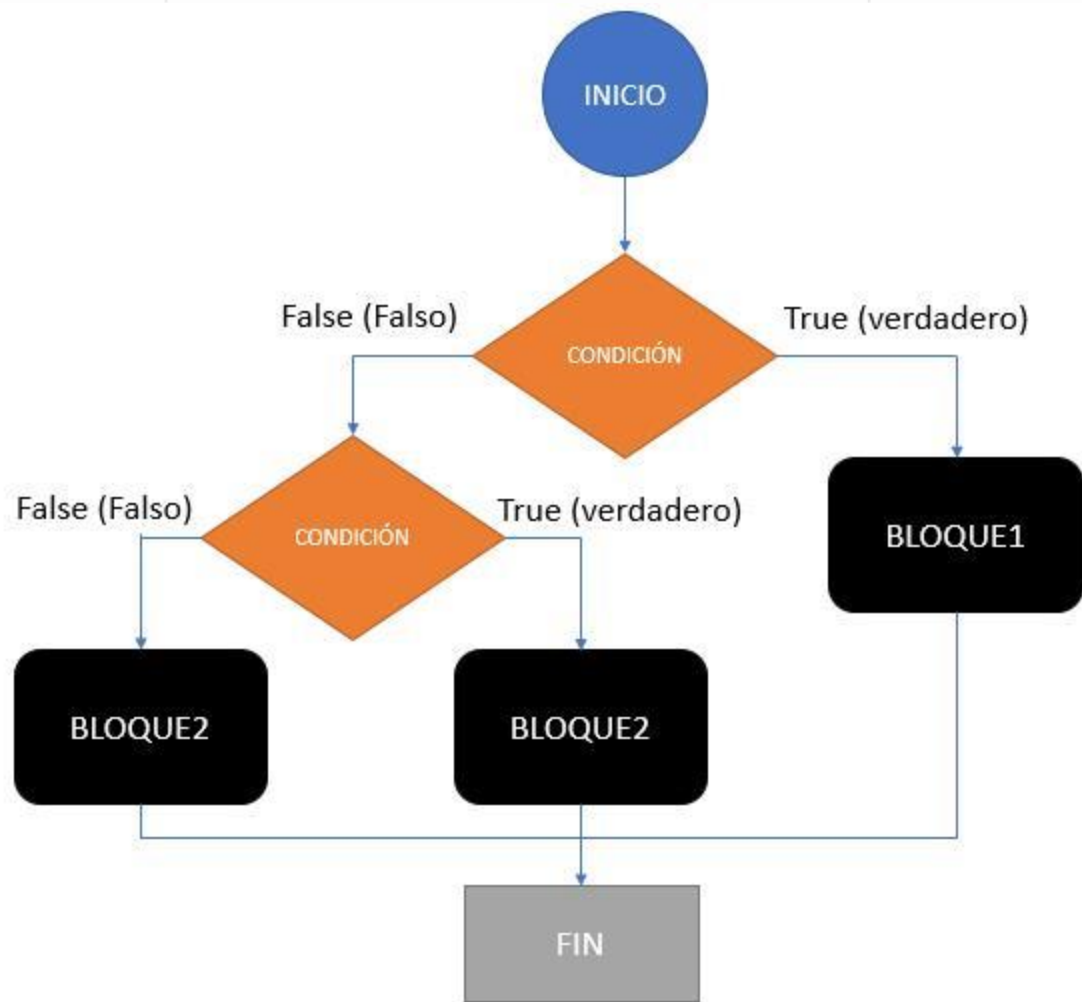
- Recibe un número e indica si es múltiplo de 7 o no.

3. División entre 0

- Escribe un programa que reciba dos números. Si el divisor es 0 envía un mensaje de error e informa que no se puede realizar la división entre 0. De otra forma, presenta el resultado de la operación.



Condicionales
anidadas y
validaciones




Condicional anidado

Puedo tener un if dentro de otro if tantas veces como sea necesario.

```
edad.py - /Users/AnaRaquel/OneDrive-Tec/OneDrive - Instituto Tecnológico y de...
#Este programa identifica el numero mayor
a=int(input("dame el valor de a: "))
b=int(input("dame el valor de b: "))
if a < b:
    print(a, " es menor que ", b)
else:
    if a > b:
        print(a, " es mayor que ", b)
    else:
        print(a, " y ", b, "son iguales")
```

Ln: 2 Col: 36

Y puedo abreviarlo con la condicional `if elif else`



```
edad.py - /Users/AnaRaquel/OneDrive-Tec/On
#Este programa identifica el numero mayor
a=int(input("dame el valor de a: "))
b=int(input("dame el valor de b: "))
if a < b:
    print(a, " es menor que ", b)
elif a > b:
    print(a, " es mayor que ", b)
else:
    print(a, " y ", b, "son iguales")
```

Este else le
pertenece al if
del elif

```
dame el valor de a: 67
dame el valor de b: 56
67 es mayor que 56
>>>
```


Puntos importantes de condicionales

- El **if** va acompañado de una condición que se valida como verdadera o falsa y : **(dos puntos)**
- El **else**, que también tiene : **(dos puntos)** NO tiene condición, es un complemento que tiene el código de lo que se ejecutará cuando la condición del if resulte falsa.
- Si necesitas plantear otra condición después de un **else**, puedes utilizar otro **if** dentro de él. Esto se conoce como *if-anidado*.
- Python maneja el **if** anidado con la instrucción **elif** que resume dos renglones de código, la del else y la del if.
- El **elif**, al igual que un **if**, tiene su condición a evaluar y sus dos puntos.

```
edad.py - /Users/AnaRaquel/OneDrive-Tec/On
#Este programa identifica el numero mayor
a=int(input("dame el valor de a: "))
b=int(input("dame el valor de b: "))
if a < b:
    print(a, " es menor que ", b)
elif a > b:
    print(a, " es mayor que ", b)
else:
    print(a, " y ", b, "son iguales")
```

A programar

THINK



CODE

Ejercicio:

- Crea un programa que dado el valor de x , te despliegue su correspondiente valor de y de acuerdo a la siguiente función por intervalos:

$$y = f(x) = \begin{cases} 2x & \text{si } x \leq -1 \\ 2x + 1 & \text{si } -1 < x < 1 \\ -x + 4 & \text{si } 1 \leq x < 4 \\ x - 1 & \text{si } x \geq 4 \end{cases}$$

Ejercicios condicional anidadas simples:

Escribe un programa que reciba la calificación del examen y la del proyecto.

Si la calificación tanto del examen como del proyecto es mayor o igual a 80 despliega la palabra APROBADO, Si la calificación del examen es mayor a 70 y del proyecto mayor a 50 despliega, **CONDICIONADO**, en caso contrario la palabra mostrada es **REPROBADO**

Ejercicios condicional anidadas con validación:

Realiza un programa que simule una calculadora.

El programa deberá recibir dos números y calcular:

- Suma
- Resta
- Multiplicación
- División
- Para el caso de la división, deberás revisar que el divisor no sea 0, de lo contrario deberás mandar el mensaje de error correspondiente.

Otro condicional anidado:

Actualmente el clima cambia de frío a calor intempestivamente. Un programador ha colocado un termómetro digital en la puerta de su casa que le indica la temperatura exacta momentos antes de salir a la calle y con ello podrá vestir de forma adecuada

Según la temperatura :

- ▣ A 30 o más grados centígrados vestirá “playera sin mangas y short”
- ▣ Si la temperatura marca menos de 30 pero más de 15 grados vestirá “playera y jeans”
- ▣ Si el termómetro marca temperatura negativa (< 0) usará su “gabardina y calentadores”
- ▣ Si la temperatura es cualquier otra, vestirá “sweter y pantalón”

Escribir un programa que pida la temperatura y regrese la sugerencia de vestimenta

Realiza el programa
que resuelve
ecuaciones
cuadráticas por la
fórmula general.

1. Se solicitan los valores de a , b y c
2. Si a es igual a 0
 - a) Si b es igual a 0
 - i. “No es una ecuación”
 - b) De lo contrario
 - i. $x = -c / b$
3. De lo contrario
 - a) Si $b^2 - 4ac$ es menor que 0
 - i. “Solución imaginaria”
 - b) Sino, si $b^2 - 4ac$ es mayor que 0
 - i. $X1 = (-b + \sqrt{b^2 - 4ac}) / (2a)$
 - ii. $X2 = (-b - \sqrt{b^2 - 4ac}) / (2a)$
 - c) Sino
 - a) “Ambas raíces son iguales a:”
 $-b/2a$

Depuración y prueba de programas

Si **tu** quieres programar sin conocer el planteamiento algorítmico de un problema, es muy probable que **tu experiencia** de programar sea muy frustrante.



Mensajes de error:

```
raiz.py x Aprobado.py x edad.py x
1 """
2 Primer programa con condicionales
3 Ana Raquel
4 """
5 print("Este programa indica si eres mayor de edad")
6 edad=int(input("Dame tu edad "))
7 if edadi>=18:
8     print("Eres mayor de edad")
9 else:
10    print("NO eres mayor de edad")
```

```
Shell x
Este programa indica si eres mayor de edad
Dame tu edad 45
Traceback (most recent call last):
  File "C:\Users\L00629725\OneDrive - Instituto Tecnológico y de Estudios Superiores de Monterrey\MateriasTEC\Thonny\edad.py", line 9, in <module>
    if edadi>=18:
NameError: name 'edadi' is not defined
```

```
raiz.py x Aprobado.py x edad.py x
1 """
2 Primer programa con condicionales
3 Ana Raquel
4 """
5 print("Este programa indica si eres mayor de edad")
6 edad=int(input("Dame tu edad "))
7 if edad>=18:
8     print("Eres mayor de edad")
9 else
10    print("NO eres mayor de edad")
```

```
Shell x
NameError: name 'edadi' is not defined

>>> %Run edad.py
Traceback (most recent call last):
  File "C:\Users\L00629725\OneDrive - Instituto Tecnológico y de Estudios Superiores de Monterrey\MateriasTEC\Thonny\edad.py", line 9
    else
    ^
SyntaxError: invalid syntax
```

```
raiz.py x
1 """
2 Programa que calcula la raíz cuadrada de un número dado por el u
3 """
4 import math
5 print("Este programa calcula la raíz cuadrada de un número")
6 # Se solicita el ingreso de un número y se guarda en num
7 num=int(input("Dame un número para obtener su raíz "))
8 # Se calcula la raíz cuadrada con la función sqrt de la libreria
9 raiz=sqrt(num)
10 print(f"La raíz cuadrada de {num} es {raiz:.3f}")
11 # Insertamos el valor de num y de raíz y formateamos raíz con 3
```

```
Shell x
Python 3.7.7 (bundled)
>>> %Run raiz.py

Este programa calcula la raíz cuadrada de un número
Dame un número para obtener su raíz 34
Traceback (most recent call last):
  File "C:\Users\L00629725\OneDrive - Instituto Tecnológico y de Estudios Superiores de Monterrey\MateriasTEC\Thonny\raiz.py", line 9, in <module>
    raiz=sqrt(num)
NameError: name 'sqrt' is not defined
```

Archivo Editar Ver Ejecutar Device Herramientas Ayuda

raiz.py

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

"""
Programa para determinar el estatus de un estudiante
Entrada: Examen y Proyecto
Proceso: Calcular el promedio y determinar el estatus
Salida: Estatus del estudiante
"""
#Entrada
examen = float(input("Dame la calificación obtenida en el examen final "))
proyecto = float(input("Dame la calificación obtenida en el proyecto final "))
#Proceso
if examen >= 80 and proyecto >= 80:
 estatus = "APROBADO"
elif examen >= 60 and proyecto >= 60:
 estatus = "CONDICIONADO"
else:
 estatus = "REPROBADO"

- Archivos
- Asistente
- Ayuda
- Boceto
- Excepción
- Inspector de objetos
- Montículo (heap)
- Notas
- Program tree
- Shell
- Stack
- Variables
- Argumentos del programa
- Plotter
- Aumentar tamaño de fuente Ctrl++
- Disminuir tamaño de la fuente Ctrl+-
- Cambiar foco al editor Alt+E
- Cambiar foco al terminal Alt+S

```
es, proyecto final y examen final.  
=80 entonces imprimo APROBADO  
proyecto>=50 entonces imprimo CONDICIONADO  
s y serían: APROBADO, CONDICIONADO O REPROBADO  
us de tu calificación final")  
ción obtenida en el examen final ")  
cación obtenida en el proyecto final ")
```

Variables	
Name	Value
examen	34.0
proyecto	43.0

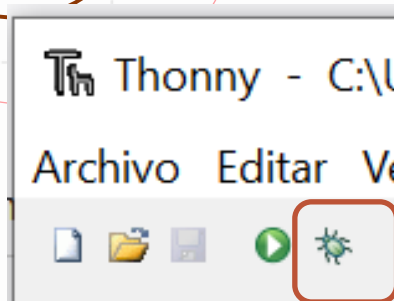
Shell

```
SyntaxError: invalid syntax  
>>> %Run Aprobado.py  
Este programa te da un estatus de tu calificación final  
Dame la calificación obtenida en el examen final 34  
Dame la calificación obtenida en el proyecto final 43  
REPROBADO  
>>>
```

Activa la opción de ver
cómo se asignan los valores
a las variables

¿Y cuando no es TAN SIMPLE?

Corre el programa con el debugger



Podrás ir viendo cómo se están recibiendo los datos, dónde se almacenan y como va avanzando el programa paso a paso. Puedes brincar pedazos de código que no quieres revisar con Step over.

```
Thonny - C:\Users\L00629725\OneDrive - Instituto Tecnologico y de Estudios Superiores de Monterrey\MateriasTEC\Thonny\Aprobado.py @ 7:28
chivo Editar Ver Ejecutar Device Herramientas Ayuda
raiz.py Aprobado.py edad.py
Step into (F7)
1 """
2 Programa que recibe dos calificaciones, proyecto final y examen final.
3 ENTRADAS: proyecto, examen
4 PROCESO: si proyecto>=80 Y examen >=80 entonces imprimo APROBADO
5          sino si examen >=70 Y proyecto>=50 entonces imprimo CONDICIONADO
6          sino imprimo REPROBADO
7 SALIDAS: Dependen de las condiciones y serían: APROBADO, CONDICIONADO O REPROBADO
8 """
9 print ("Este programa te da un estatus de tu calificación final")
10 #ENTRADAS
11 examen=float(input("Dame la calificación obtenida en el examen final "))
12 proyecto= float('90') (Dame la calificación obtenida en el proyecto final ")
13 #proceso y salidas
14 if examen>=80 and proyecto>=80:
15     print("APROBADO")
16 elif examen>=70 and proyecto>=50:
17     print("CONDICIONADO")
18 else:
19     print("REPROBADO")
20

Variables
Name Value
examen 56.0

Shell
REPROBADO
>>> %Debug Aprobado.py
Este programa te da un estatus de tu calificación final
Dame la calificación obtenida en el examen final 56
Dame la calificación obtenida en el proyecto final 90
```

Buenas prácticas de programación

1. Documentación: Que el programa esté comentado con tu nombre y al menos algo del código explicado. Además el programa utiliza nombres de variables que representan para qué están siendo utilizadas.
2. Usabilidad: Que el programa interactúe apropiadamente con el usuario, le dé instrucciones de uso y el texto aparezca con el espaciado apropiado tanto en las capturas como en la impresión de resultados.
3. Eficiencia: El programa funciona y presenta los resultados esperados, si hay unidades, las menciona.
4. Formato: El programa utiliza los formatos de impresión vistos en clase.
5. Pruebas: Los ejemplos planteados muestran la efectividad del programa y que está validado adecuadamente.

Qué aprendimos...



Decisiones If

Las condiciones se plantean con if y :
Lo que sucederá si la condición es verdadera se escribe debajo del if, con una indentación de 4 espacios.



Else

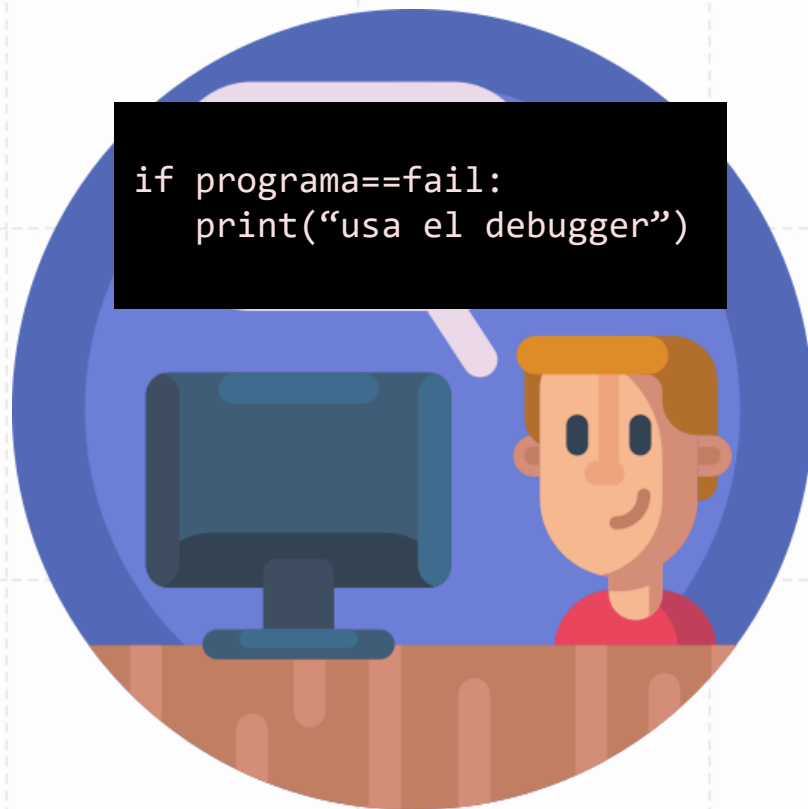
1. Si quiero que ocurra algo diferente cuando la condición NO ES VERDADERA, lo escribo debajo del else:
2. El else NUNCA lleva condición.



Depuración: Debug

El *debugger* me permite ver la corrida del programa paso a paso, de manera que puedo ver lo que ocurre con las variables y si se están asignando correctamente.

```
if programa==fail:  
    print("usa el debugger")
```



Recursos recomendados

- Python: Encuentra los errores en tu programa.
 - <https://youtu.be/NYYT0J8nf3o>
 - Video desarrollado por Yolanda Martínez.
- 