

Capitulo 3

Módulo 3

Módulo 3

- Toma de decisiones en Python
- Operadores relacionales
- Condiciones y ejecución condicional
- La cláusula `elif`
- Bucles de Python (`loop`)
- Operaciones lógicas y de bits
- Listas: recopilaciones de datos
- Clasificación de listas simples: el algoritmo de clasificación de burbujas
- Listas en aplicaciones avanzadas.

3.1 Toma de decisiones en Python

Preguntas y respuestas

- Un programador escribe un programa y el programa hace preguntas.
- Una computadora ejecuta el programa y proporciona las respuestas. El programa debe poder reaccionar de acuerdo con las respuestas recibidas.
- Afortunadamente, las computadoras solo conocen dos tipos de respuestas:
 - si, esto es cierto (True)
 - No, esto es falso (False)

Comparación: operador de igualdad

Pregunta: ¿son dos valores iguales?

Para hacer esta pregunta, usa el operador == (igual igual).

No olvide esta importante distinción:

= es un operador de asignación,

== es un operador de igualdad, $a == b$ compara a y b .

Es un operador binario con enlace del lado izquierdo. Necesita dos argumentos y comprueba si son iguales.

Igualdad: el operador "igual a" (==)

El operador == (igual a) **compara** los valores de dos operandos. Si son iguales, el resultado de la comparación es **Verdadero (True)** . Si no son iguales, el resultado de la comparación es **Falso (False)**.

```
var == 0  
  
var = 0  
print (var == 0)  
var = 1  
print (var == 0)
```

Desigualdad: el operador "no es igual a" (! =)

El operador != (No igual a) también **compara** los valores de dos operandos. Aquí está la diferencia: si son iguales, el resultado de la comparación es **Falso (False)**. Si no son iguales, el resultado de la comparación es **Verdadero (True)**.

```
var != 0  
  
var = 0  
print (var != 0)  
var = 1  
print (var != 0)
```

Comparación: mayor que

También puede hacer una pregunta de comparación utilizando el operador > (mayor que).

True lo confirma; False lo niega.

```
a = 3  
b = 5  
print(a > b)
```


Comparación: mayor o igual a

El operador mayor que tiene otra variante especial, no estricta, pero se denota de manera diferente que en la notación aritmética clásica: `>=` (mayor o igual que).

```
a = 3
b = 5
c = 2
print (a >= b)
print (a >= c)
```

Comparación: menor o igual a

Como probablemente ya haya adivinado, los operadores utilizados en este caso son: el operador <(menor que) y su hermano no estricto: <= (menor o igual que).

```
a = 3
b = 5
c = 2
print (a <= b)
print (a <= c)
```

Prioridades - ¡actualizado!

Priority	Operator	
1	<div>+</div> , <div>-</div>	unary
2	<div>**</div>	
3	<div>*</div> , <div>/</div> , <div>%</div>	
4	<div>+</div> , <div>-</div>	binary
5	<div><</div> , <div><=</div> , <div>></div> , <div>>=</div>	
6	<div>==</div> , <div>!=</div>	

Práctica de laboratorio 3.1.1.5

Escenario

Usando uno de los operadores de comparación en Python, escriba un programa simple de dos líneas que tome el parámetro n como entrada, que es un número entero, e imprima:

- Falso si n es menor que 100, y
- Verdadero si n es mayor o igual que 100.

Condiciones y ejecución condicional

La primera forma de una declaración condicional le permite hacer algo si se cumple una condición y no hacerlo si no se cumple.

Seudocódigo:

```
si es_verdadero_o_no:  
    hacer_esto_si_verdadero
```

Sintaxis:

```
if es_verdadero_o_no :  
    hacer_esto_si_verdadero
```

Condiciones y ejecución condicional

Esta declaración condicional consta de los siguientes elementos, estrictamente necesarios en este orden:

- La palabra clave `if`;
- Uno o más espacios en blanco;
- Una expresión (una pregunta o una respuesta)
- Dos puntos seguidos de una nueva línea;
- Una instrucción con sangría o un conjunto de instrucciones (al menos una instrucción es absolutamente necesaria)

Ejecución condicional: la declaración `if`

```
if contador_de_ovejas >= 120:  
    Hacer_la_cama()  
    bañarse()  
    Dormir_y_soñar()  
Alimentar_a_los_perros_pastor()
```

Ejecución condicional: la declaración `if-else`

```
if condición_es_verdadera_o_falsa:  
    realizar si la condición es verdadera  
else:  
    realizar si la condición es falsa
```


Ejecución condicional: la declaración `if-else`

```
si hace_buen_tiempo:
```

```
    ir_a_caminar()
```

```
Si no:
```

```
    ir_a_un_teatro()
```

```
almorzar()
```

Ejecución condicional: la declaración `if-else`

```
si hace_buen_tiempo: sal_a_caminar()  
si no: ir_a_un_teatro()  
almorzar()
```

Ejecución condicional: la declaración `if-else`

```
si hace_buen_tiempo:  
    ir_a_caminar()  
    divertirse()  
  
Si no:  
    ir_a_un_teatro()  
    disfruta_la_pelicula()  
almorzar()
```

Declaraciones if-else anidadas

SEUDOCÓDIGO:

```
si hace_buen_tiempo:
    si se_encuentra_un_buen_restaurante:
        almorzar()
    si no:
        comer_un_sandwich()
si no:
    si las_entradas_están_disponibles:
        ir_al_teatro()
    si no:
        ir_de_compras()
```

CODIFICACIÓN:

```
if hace_buen_tiempo():
    if se_encuentra_un_buen_restaurante():
        almorzar()
    else:
        comer_un_sandwich()
else:
    if las_entradas_están_disponibles():
        ir_al_teatro()
    else:
        ir_de_compras()
```

La declaración de elif

elif se usa para verificar **más de una** condición y para **detenerse** cuando se **encuentra** la primera **declaración** que es verdadera.

```
si hace_buen_tiempo:  
    sal_a_caminar()  
O sino las_entradas_están_disponibles:  
    ir_al_teatro()  
O sino la_mesa_está_disponible:  
    ir_a_almorzar()  
otra cosa: jugar_ajedrez_en_casa ()
```

La declaración de elif

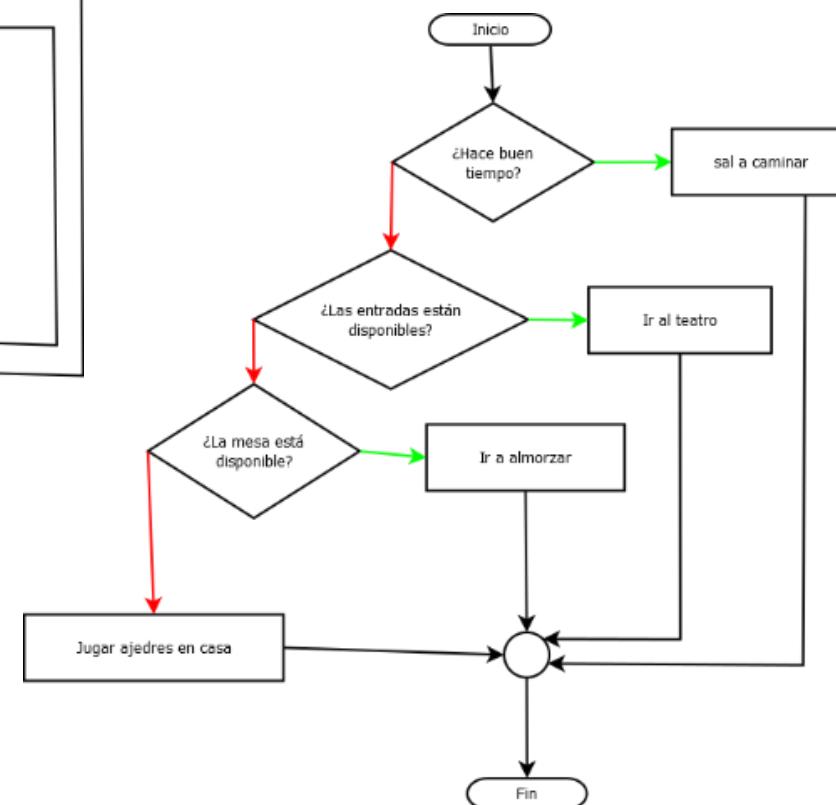
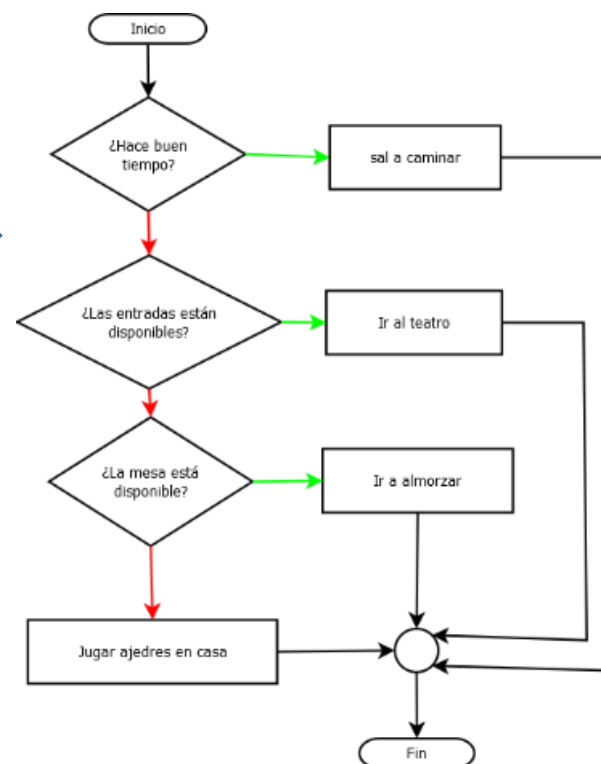
if - elif - else

```
if hace_buen_tiempo():  
    sal_a_caminar()  
elif las_entradas_están_disponibles():  
    ir_al_teatro()  
elif La_mesa_está_disponible():  
    ir_a_almorzar()  
else:  
    jugar_ajedrez_en_casa()
```

if - else anidados

```
if hace_buen_tiempo():  
    sal_a_caminar()  
else:  
    if las_entradas_están_disponibles():  
        ir_al_teatro()  
    else:  
        if La_mesa_está_disponible():  
            ir_a_almorzar()  
        else:  
            jugar_ajedrez_en_casa()
```

Las mismas decisiones expresadas en forma distinta:



La declaración de `elif`

En este caso, se debe prestar atención adicional:

- No debe utilizar `"else"` sin un `"if"` anterior
- `"else"` es siempre la última rama de la cascada, independientemente de si ha utilizado `"elif"` o no
- `"else"` es una parte opcional de la cascada y puede omitirse
- si hay una rama `"else"` en la cascada, sólo se ejecuta una de todas las ramas
- si no hay una rama `"else"`, es posible que ninguna de las ramas disponibles se ejecute.

Práctica de laboratorio 3.1.1.11 Operadores de comparación y ejecución condicional

Escenario – Spathiphyllum

Escriba un programa que utilice el concepto de ejecución condicional, tome una cadena como entrada y:

- Imprime la frase "Sí, ¡Spathiphyllum es la mejor planta que existe!" a la pantalla si la cadena ingresada es "Spathiphyllum" (mayúsculas)
- Imprime "No, quiero un gran Spathiphyllum!" si la cadena ingresada es "spathiphyllum" (minúsculas)
- Imprime "¡Spathiphyllum! ¡No [entrada]!" de otra manera. Nota: [entrada] es la cadena que se toma como entrada.

Práctica de laboratorio 3.1.1.12

Conceptos básicos de la instrucción `if-else`

Escenario – escriba una calculadora de impuestos

- Debe aceptar un valor de punto flotante: el ingreso.
- Imprima el impuesto calculado, redondeado a pesos completos de acuerdo con estas reglas:
- Si el ingreso no era superior a 85.528 pesos, el impuesto equivalía al 18% del ingreso menos 556 pesos y 2 centavos (este fue el llamado alivio fiscal)
- Si el ingreso era superior a esta cantidad, el impuesto equivalía a 14.839 pesos y 2 centavos, más el 32% del excedente sobre 85.528 pesos.