



UNIVERSIDAD DE SANTIAGO DE CHILE  
FACULTAD DE INGENIERÍA  
Fundamentos de Computación y Programación (10110-1)



# CLASE N°7

## EXPRESIONES BOOLEANAS Y ESTRUCTURAS DE DECISIÓN


## VALORES BOOLEANOS



- Al realizar una comparación, Python entrega como resultado los valores `True` o `False`
- Estos valores son denominados **booleanos**
  - Representan el resultado de una evaluación lógica
  - Son a un **tipo de dato nativo** en Python
  - No hay más posibilidades: verdadero o falso
- Las expresiones que devuelven valores lógicos se denominan **expresiones booleanas**

2


## EXPRESIONES BOOLEANAS



- Python maneja varios **operadores de comparación**:
  - Mayor  $>$   $4.3 > 3.2$
  - Mayor o igual  $>=$   $4.0 >= 4.1$
  - Menor  $<$   $-2 < 0$
  - Menor o igual  $<=$   $-3.14 <= -3.2$
  - Igual  $==$   $2.0 == 2$
  - Distinto  $!=$   $-2 != -2.1$

3

## EXPRESIONES BOOLEANAS



- Existen además **operadores para componer** expresiones booleanas:
  - La **conjunción** o “y lógico” (and), que resulta verdadero si y sólo **todas** las sub-expresiones son verdaderas
    - $x > y$  and  $y <= z$
  - La **disyunción** u “o lógico” (or), que resulta verdadero si **al menos una** de las sub-expresiones lo es
    - $x != y$  or  $x <= z$  or  $y < z$

4

## EXPRESIONES BOOLEANAS



- Finalmente existe el operador **negación** (not) para **invertir** el valor de verdad
  - `not 50 > 4` → `not True` → `False`
  - `not -40.4 >= 44.5` → `not False` → `True`

5

## EXPRESIONES BOOLEANAS



- Los operadores booleanos también tienen **reglas de precedencia**
  - Operadores de comparación tienen **menor precedencia** que los operadores aritméticos
  - Luego tiene precedencia el operador de negación, luego las conjunciones y finalmente las disyunciones
- Podemos alterar esta precedencia usando paréntesis

6

## SENTENCIA IF



- Permite **condicionar la ejecución** de un bloque de sentencias al cumplimiento de una **condición**
  - La condición es una expresión booleana
  - Sintaxis:

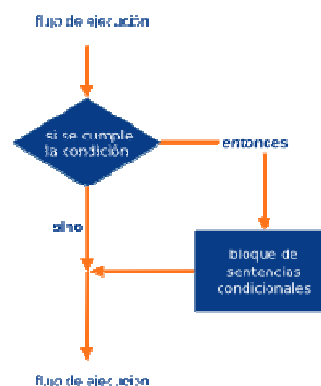
```
if <condición>:  
    <Bloque de sentencias condicionales>  
    <Bloque de sentencias que sigue>
```
  - El fin del bloque condicionado se marca con un **retroceso** en la indentación

7

## SENTENCIA IF



- Como diagrama:
  - El **flujo de ejecución** (como que fuera flujo de agua en tuberías) **pasa de largo** si la condición no se cumple
  - Pero **pasa por el cuerpo** de la sentencia **if** si la condición resulta con valor verdadero



8

## SENTENCIA IF - ELSE



- Permite **condicionar la ejecución** de **dos bloques** de sentencias al cumplimiento de una **condición**

- Sintaxis:

```
if <condición>:  
    # Se ejecuta si la condición se cumple  
    <Bloque de sentencias condicionales>  
else:  
    # Se ejecuta si la condición no se cumple  
    <Bloque de sentencias alternativo>  
    <Bloque de sentencias que sigue>
```

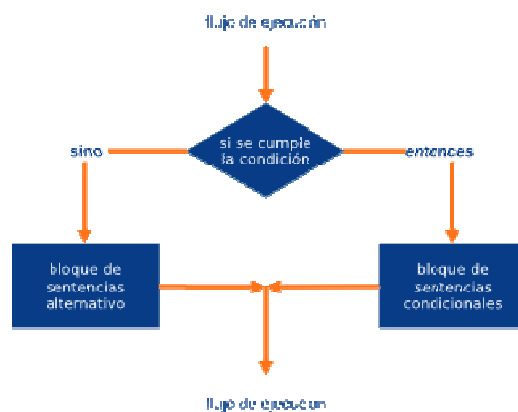
9

## SENTENCIA IF - ELSE




- Como diagrama:

- El flujo de ejecución **pasa por el bloque condicional** si la condición se cumple
- Pero si no se cumple, el flujo **pasa por el bloque alternativo**




10

## PARA LA PRÓXIMA CLASE



- Revisar guía de ejercicios propuestos:
  - Profundizar en los contenidos vistos hoy
  - Averiguar que son las **decisiones anidadas** y la sintaxis de la construcción **if-elif-else**
- Practicaremos creando programas en Python que toman decisiones anidadas

11



## ¿CONSULTAS?

12