




if

Clase #5

IIC1103 – Introducción a la Programación

El plan de hoy es...

- Conocer Clearn
- Sets = entrega TODOS los lunes a las 19:00
 - Set 1 = Lunes 27 Marzo a las 19:00
 - Hoy llega aviso sobre clearn
- Salas SAP y SAV en Canvas

Distribución de Salas				
Lunes	Martes	Miércoles	Jueves	Viernes
K203	B12	BC22	A3	LAB 
	B21		A5	
K204	B25	LAB 	B11	
LAB 	LAB 		C002	SAV 
			K203	
SAV 	SAV 	SAV 	LAB 	
			SAV 	

El plan de hoy es...

- Cambio en fecha I1: 18 mayo a las 18:30
- if / else!
- Descargar pycharm (community edition) -> <http://www.jetbrains.com/pycharm/>

Problema 0 (Repaso)

- Cara (1) o Sello (2) ? **1**
- Salió 1.
- True



Solución

- `import random`
- `n = int(input("Cara (1) o Sello (2)?"))`
- `moneda = random.randint(1,2)`
- `print("Salio",moneda)`
- `print(moneda==n)`

Repaso clase pasada: **bool**

- **bool**: valor de verdad, solo dos valores posibles: True y False
- `a = 5`
- `b = 3`
- `a > b` #True
- `b > a` #False
- `c = a > b` # c es True
- `d = "hola"`
- `d == "chao"` #False
- `d != "chao"` #True

operadores booleanos

<code>==</code>	son iguales?
<code>!=</code>	son distintos?
<code><</code>	menor
<code>></code>	mayor
<code><=</code>	menor o igual
<code>>=</code>	mayor o igual

Clase pasada: **random.randint**

- `print("hola")`
- `x = random.randint(1,6)`

Algunas funciones retornan un valor, otras no devuelven nada, simplemente ejecutan alguna acción (como imprimir)

Vamos a...



Problema #1

- Juguemos cachipún!
- Elige: 1=Piedra, 2=Papel, 3=Tijeras: 2
- Yo saqué: 3!!!!
- Ganaste? False

Piedra, papel o tijera, conocido también como cachipún, jankenpón, dum-kin-voy, yan ken po, pin pon papas, chis bun papas, hakembó, How-are-you-speak o kokepon, es un **juego infantil**, un juego de manos en el que



Condición simple o comparación

Sintaxis: expresión1
operador-de-relación
expresión2

Semántica:

- evaluar expresiones
- comparar resultados de expresiones
- si condición se cumple entregar valor verdadero (True)
- si condición no se cumple entregar valor falso (False)

op	Descripción
<	menor
>	mayor
<=	menor o igual
>=	mayor o igual
==	igual
!=	distinto

Condición compuesta

Sintaxis

- condición1 operador-lógico condición2 . . .
- operador lógico:
 - **and**: y, conjunción
 - **or**: o, disyunción

Semántica

c1	c2	c1 and c2	c1 or c2
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F
		V si ambos V	V si alguno V

Además:

- **not**: negación

c1	not c1
V	F
F	V

Solución

- `import random`
- `print("Juguemos cachipún!")`
- `jugador = int(input("Elige: 1=Piedra, 2=Papel, 3=Tijeras: "))`
- `computador = random.randint(1,3)`
- `print("Yo saque",computador,"!!!!!!")`
- **`G = (jugador==1 and computador==3) or (jugador==2 and computador==1) or (jugador==3 and computador==2)`**
- `print("Ganaste?",G)`

Problema #2

- Voy a tirar un dado
- Adivina cual numero me sale? 3
- Salio un 3
- Si, adivinaste!

Instrucción if-else

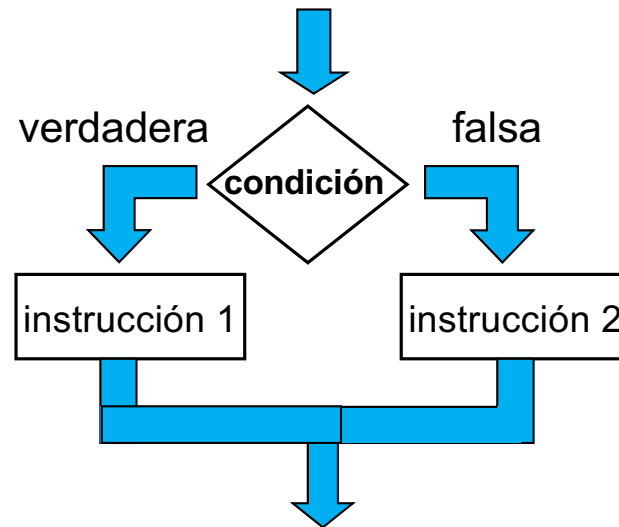
Sintaxis

```
if condición:  
    instrucción1  
else:  
    instrucción2
```

Semántica

- Si condición se cumple (es verdadera), ejecutar instrucción1
- Si condición no se cumple (es falsa), ejecutar instrucción2

Gráficamente:



Instrucción if-else

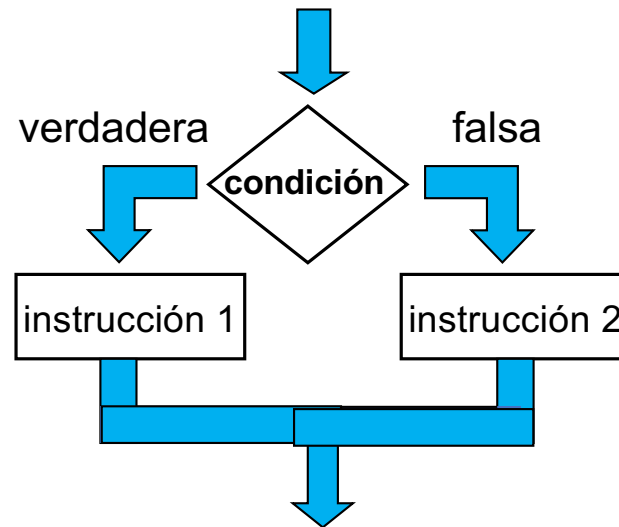
Sintaxis

```
if condición:  
    instrucción1  
else:  
    instrucción2
```

Semántica

- Si condición se cumple (es verdadera), ejecutar instrucción1
- Si condición no se cumple (es falsa), ejecutar instrucción2

Gráficamente:



Solución

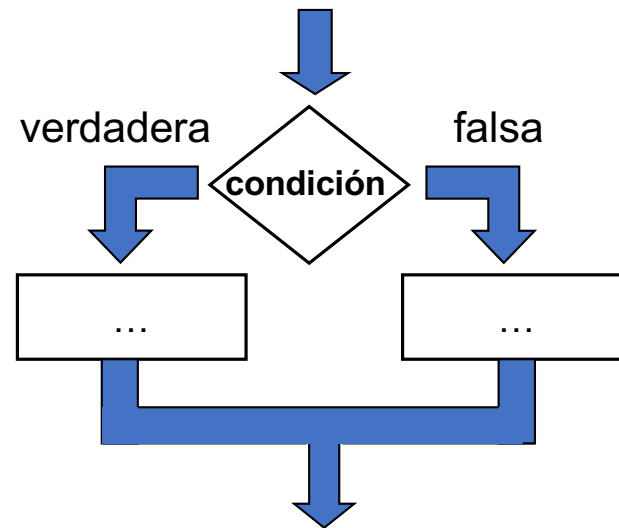
```
• import random
  print("Voy a tirar un dado")
  n = int(input("Adivina cual numero me sale?"))
  dado = random.randint(1,6)
  print("Salio un "+str(dado))
  if dado == n:
      print("Si, adivinaste!")
  else:
      print("No adivinaste")
```


Instrucción if-else

Sintaxis

```
if condición:  
    instrucción  
    instrucción  
    ...  
else:  
    instrucción  
    instrucción
```

Gráficamente:



Uso de indentación en Python

- Los espacios en blanco/tabs **IMPORTANT!**

if condición:

instrucción
instrucción
...



estas instrucciones
corresponden al if

else:

instrucción
instrucción



estas instrucciones
corresponden al else

instrucción

Python 



**A PYTHON DEVELOPER
MEASURING THE LEVEL OF
INDENTATION**

Python is the
easier language
to learn.
No brackets,
no main.



You get errors
for writing an
extra space



INSTA: RATHAN.CAGE

Problema #3

- Cuántos chocolates hay? 4
- Cuántas personas hay? 5
- Nadie va a comer chocolate!
- Repartir 0 chocolates para cada uno, sobra 4



Solución

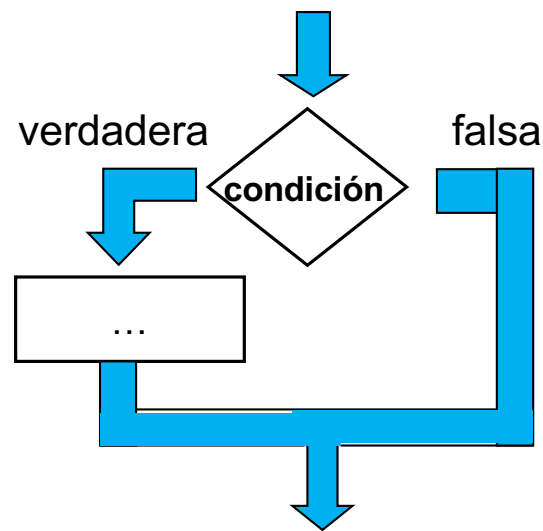
¿Cómo le agregamos la advertencia?

- `choc = int(input("Cuántos chocolates hay? "))`
- `pers = int(input("Cuántas personas hay?"))`
- `print("Repartir", choc//pers, "chocolates para cada uno, sobra", choc%pers)`

Solución: else puede omitirse

```
if choc < pers:  
    print("Nadie va a comer chocolate!")
```

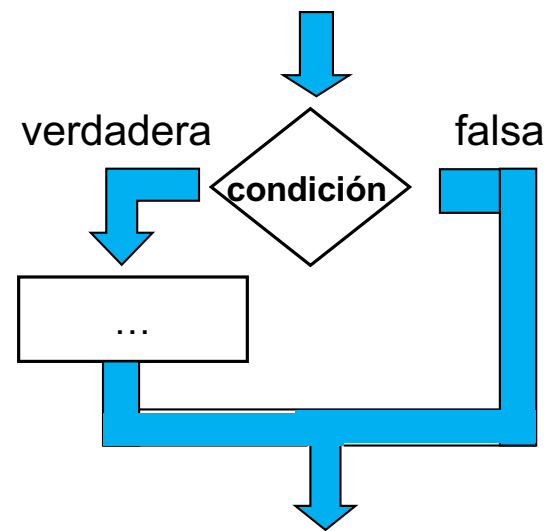
Ojo: Puede existir un
if sin else,
pero no un else sin
if!!!



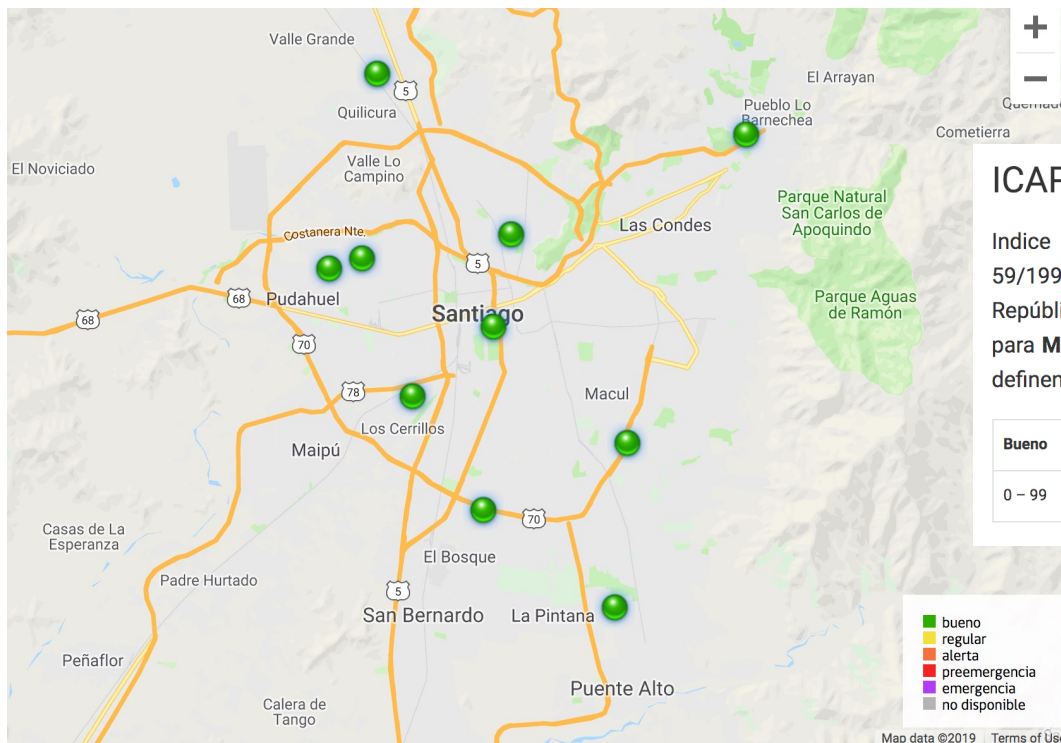
Solución: else puede omitirse

```
if choc < pers:  
    print("Nadie va a comer chocolate!")
```

Ojo: Puede existir un
if sin else,
pero no un else sin
if!!!



En la vida real...



ICAP

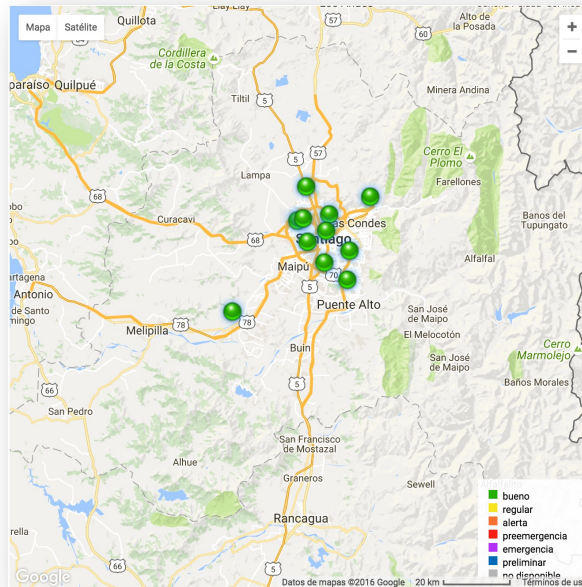
Indice de **Calidad del Aire** referido a Partículas (ICAP) según D.S. N° 59/1998 del Ministerio Secretaría General de la Presidencia de la República (MINSEGPRES) que establece la **Norma de Calidad Primaria** para **Material Particulado Respirable** MP10 y en especial los niveles que definen situaciones de Emergencia Ambiental.

Bueno	Regular	Alerta	Preemergencia	Emergencia
0 – 99	100 – 199	200 – 299	300 – 499	500 – Superior

<https://sinca.mma.gob.cl/> - <http://airesantiago.gob.cl/balance-1997-2017/calidad-del-aire/>

Problema 4 => menti

- Ingresar valor Estación:
124
- Regular



Bueno: 0 - 99

Regular: 100 - 199

Condición compuesta

```
• valor = int(input("Ingresar valor  
estacion: "))  
  
if valor >= 0 and valor <=99:  
    print("Bueno")
```

Condición compuesta

Sintaxis

- condición1 operador-lógico condición2 . . .
- operador lógico:
 - **and**: y, conjunción
 - **or**: o, disyunción

Semántica

c1	c2	c1 and c2	c1 or c2
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F
		V si ambos V	V si alguno V

Además:

- **not**: negación

c1	not c1
V	F
F	V

Pregunta:

En vez de $a \leq b$ and $b \leq c$?????

Comparisons yield boolean values: `True` or `False`.

Comparisons can be chained arbitrarily, e.g., `x < y <= z` is equivalent to `x < y` and `y <= z`, except that `y` is evaluated only once (but in both cases `z` is not evaluated at all when `x < y` is found to be false).

Formally, if `a`, `b`, `c`, ..., `y`, `z` are expressions and `op1`, `op2`, ..., `opN` are comparison operators, then `a op1 b op2 c ... y opN z` is equivalent to `a op1 b` and `b op2 c` and ... `y opN z`, except that each expression is evaluated at most once.

Note that `a op1 b op2 c` doesn't imply any kind of comparison between `a` and `c`, so that, e.g., `x < y > z` is perfectly legal (though perhaps not pretty).

Fuente: <https://docs.python.org/3/reference/expressions.html>

if's anidados

- `valor = int(input("Ingresar valor
estacion: "))`

- `if valor >= 0 and valor <=99:`
 `print("Bueno")`

if pueden estar
anidados

- `else:`
 - `if valor>=100 and valor<=199:`
 `print("Regular")`
 - `else:`
 `print("Invalido")`

¿Practicar más?

Problema 1

- Las monedas en EEUU son de 1, 5, 10, y 25 centavos. Escribe un programa que siga el siguiente diálogo (ejemplo):
- > Cuanto vuelto necesitas? (en centavos): 45
- > 25c: 1
- > 10c: 2

Resumen de hoy

- Hoy vi

```
if condicion:
    instrucciones
else: #puede omitirse
    instrucciones
```

```
condicion simple:
==, !=, <, >, <=, >=
condicion compuesta:
and, or, not
```

```
if condicion1:
    instrucciones
elif condicion2:
    instrucciones
elif condicion3:
    instrucciones
...
else:
    instrucciones
```