



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

INTRODUCCIÓN A LA PROGRAMACIÓN



Alternativas. Ejemplo 1:

```
x = input()
if x == "auto":
    print("Escribiste",x)
```

Alternativas. Ejemplo 2:

```
x = int(input())
if x > 0:
    print("Nro positivo")
elif x < 0:
    print("Nro negativo")
else:
    print("Cero")
```

While. Ejemplo 1:

```
i = 10
while i > 0:
    print("T -",i)
    i = i-1
print("Despegue!")
```

While. Ejemplo 2:

```
x = "si"
while x == "si":
    print("En el while.")
    print("Continuar?(si)")
    x = input()
    print()
    print("Salimos.")
```

While. Ejemplo 3:

```
suma = 0
seguir = True
while seguir:
    x = input()
    if x == "":
        seguir = False
    else:
        suma = suma +
        float(x)
print("Suma:", suma)
```

For-range. Ejemplo 1:

```
for i in range(15):
    print("Iteracion",i)
```

For-range. Ejemplo 2:

```
for i in range(10):
    print("T -", 10-i)
print("Despegue!")
```

For-range. Ejemplo 3:

```
print("Escriba un nro")
nro = int(input())
dvs = 0
for i in range(2,nro):
    if nro%i == 0:
        print(i,"es divisor")
        dvs = dvs + 1
print("el numero", nro,
      "tiene",dvs,divisores)
```

Lógica y ejecución condicional (alternativas)

- Clase anterior
 - Ejercicio: “*x* es menor a *y*”
 - Ejercicio: *piedra-papel-tijeras*
- Esta clase
 - Objetivo 1: **if-elif-else** depende de booleanos
 - Objetivo 2: conocer las operaciones booleanas
 - Ejemplo: *piedra-papel-tijeras* con booleanos
 - Objetivo 3: operaciones aritméticas (postergadas)
- Práctica control



Escriba un programa de Python que

- Solicite dos números
- Imprima en pantalla si es que son iguales
 - Debe decir: *ambos números son iguales*
- Si no son iguales, que imprima algo como
 - *17 es mayor a 10*
 - (pero con los números ingresados)



Resuelto en clases

Escriba un programa para el juego de **piedra-papel-tijera**, tal que:

- Pregunte la opción del jugador1 y luego la opción del jugador2
 - Que el programa solicite textos
 - El usuario debe escribir *piedra, papel o tijera*
- Luego el programa debe escribir en pantalla **qué jugador resultó ganador** o bien si la conclusión fue de **empate**



```
1  j1 = input()
2  j2 = input()
3  if j1 == "piedra":
4      if j2 == "piedra":
5          print("empate")
6      elif j2 == "papel":
7          print("j2 gana")
8      else:
9          print("j1 gana")
10 elif j1 == "papel":
11     if j2 == "piedra":
12         print("j1 gana")
13     elif j2 == "papel":
14         print("empate")
15     else:
16         print("j2 gana")
17 else:
18     # caso j1 == "tijera"
```

Piedra-papel-tijera

- Termine de desarrollar este código
- Falta caso **j1** dice **tijera**, que se aborda tal como los demás
- Lección: se puede usar **if-elif-else** dentro de bloques indentados
 - Hay segundo nivel de indentación



```
1  j1 = input()
2  j2 = input()
3  if j1 == "piedra" and j2 == "piedra":
4      print("empate")
5  elif j1 == "piedra" and j2 == "papel":
6      print("j2 gana")
7  elif j1 == "piedra" and j2 == "tijera":
8      print("j1 gana")
9  elif j1 == "papel" and j2 == "piedra":
10     print("j1 gana")
11 elif j1 == "papel" and j2 == "papel":
12     print("empate")
13 elif j1 == "papel" and j2 == "tijera":
14     print("j2 gana")
15 elif j1 == "tijera" and j2 == "papel":
16     # ...
```

Piedra-papel-tijera

- Termine de desarrollar este código
- Ahora usamos la operación lógica **and** para combinar comparaciones



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Lógica



Operaciones booleanas (prioridad baja)

p	q	p or q
False	False	False
False	True	True
True	False	True
True	True	True

p	q	p and q
False	False	False
False	True	False
True	False	False
True	True	True

p	not p
False	True
True	False

```
p = True                                # es True
q = (-10 > 5.5)                          # evalúa a False
print( p or q )                           # imprime True
print( p and q )                          # imprime False
print( (p or q) and (not q) )             # imprime True
```

Comparaciones (prioridad media)

- Permiten ver si dos valores son iguales, distintos o tienen algún orden (*mayor que*, *menor que*, etc)
- Su prioridad es mayor a la lógica (`and`, `or`, `not`)
- Su prioridad es menor a la aritmética (`+`, `-`, `*`, ...)

```
x = 10
print( x == 10 )      # imprime True; == chequea igualdad
print( x != 10 )      # imprime False; != chequea desigualdad
print( x > 11 )       # imprime False; > es mayor que
print( x <= 12 )      # imprime True; <= es menor o igual que
Print( 5 < x < 20 )  # imprime True
```



Aritmética (prioridad alta)

- Operaciones típicas (suma $+$, resta $-$, multiplicación $*$)
- Divisiones (estándar $/$, entera $//$, resto $\%$)
- Potenciación (ej. 10^{**4}) tiene la prioridad más alta, luego sigue la multiplicación y divisiones
- Suma y resta tienen prioridad mínima

```
print( 10 + 7 )      # imprime 17
print( 5 - 9.9 )    # imprime -4.9
print( 8.1 * 3 )    # imprime 24.3
print( 10 / 3 )     # 3.3333333...
print( 10 / 5 )     # 2.0
print( 3**4 )        # 81
print( 81**.5 )     # 9.0

print( 13 / 4 )     # 3.25
print( 13 // 4 )    # 3
print( 13 % 4 )     # 1
# es resto de división entera
```



Investigue este programa

```
print("ingrese un numero entero")
a = int( input() )
print( a%2 == 0 )
print( (a%2 == 0) == True )
```

- ¿Es cierto que este programa imprime siempre lo mismo?
- ¿Por qué funciona así?



```
j1 = input()  
j2 = input()
```

```
j1_gana = ...
```

```
if j1 == j2:  
    print("empate")  
elif j1_gana:  
    print("j1 gana")  
else:  
    print("j2 gana")
```

Piedra-papel-tijera

- ¿Cuándo gana J1?
- Complete la línea

```
j1_gana = ...
```



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Lecciones