

# Introducción a la Programación

Prof. Agustín Gravano

Primer semestre de 2022

Clase teórica 7: Condicionales y tipo bool

# Control del flujo de ejecución de un programa

En el **paradigma de programación imperativa**, un **programa** es una secuencia finita de instrucciones: operaciones que transforman datos (el *estado* del programa), o bien **modifican el flujo de ejecución**.

## Estructuras de control de flujo:

### Funciones



### Condicionales



(hoy)

### Ciclos



(un poco hoy, y  
en próximas clases)

# Condicionales

**if** *CONDICIÓN*:  
*BLOQUE*

*CONDICIÓN* es una expresión de tipo bool  
(que puede ser *verdadera* o *falsa*).  
*BLOQUE* es un bloque de código.

*BLOQUE* se ejecuta sii *CONDICIÓN* es *verdadera*.

## Ejemplo:

```
1  def f(n:int):  
2      ''' Imprime una descripción del número n. '''  
3      if n > 5:  
4          print(n, 'es mayor que 5.')
```

# Condicionales

**if** *CONDICIÓN*:  
    *BLOQUE1*  
**else**:  
    *BLOQUE2*

*CONDICIÓN* es una expresión de tipo bool  
(que puede ser **verdadera** o **falsa**).  
*BLOQUE1/2* son bloques de código.

*BLOQUE1* se ejecuta si *CONDICIÓN* es **verdadera**.  
*BLOQUE2* se ejecuta si *CONDICIÓN* es **falsa**.

## Ejemplo:

```
1  def f(n:int):  
2      ''' Imprime una descripción del número n. '''  
3      if n > 5:  
4          print(n, 'es mayor que 5.')  
5      else:  
6          print(n, 'es menor o igual que 5.')
```

# Condicionales

**if** *CONDICIÓN1*:

*BLOQUE1*

**elif** *CONDICIÓN2*:

*BLOQUE2*

**else**:

*BLOQUE3*

*CONDICIÓN1/2* son expresiones de tipo bool.

*BLOQUE1/2/3* son bloques de código.

*BLOQUE1* se ejecuta sii *CONDICIÓN1* es **verdadera**.

*BLOQUE2* se ejecuta sii *CONDICIÓN1* es **falsa** y  
además *CONDICIÓN2* es **verdadera**.

*BLOQUE3* se ejecuta sii *CONDICIÓN1* y  
*CONDICIÓN2* son **falsas**.

## Ejemplo:

```
1 def f(n:int):  
2     ''' Imprime una descripción del número n. '''  
3     if n > 5:  
4         print(n, 'es mayor que 5.')  
5     elif n == 5:  
6         print(n, 'es igual a 5.')  
7     else:  
8         print(n, 'es menor que 5.')
```

Podemos incluir cualquier número de **elif**'s.

## Ejercicio:

¿Qué imprime por pantalla este código?

```
1  a:int = 1234
2  b:int = 987
3  c:int = 0
4
5  if a > b:
6      c = a
7  else:
8      c = b
9  print(c)
```

Pensar la respuesta; no ejecutar el código en la computadora.

# Tipo de datos `bool`

La condición de los condicionales es una expresión de tipo `bool` (o una expresión *booleana*, o de tipo *lógico*).

Hay dos valores de verdad posibles: `verdadero` (`True`) y `falso` (`False`).

**Operaciones:** `Negación` (`not`), `conjunción` (`and`) y `disyunción` (`or`).

Estas operaciones se definen mediante **tablas de verdad**:

<code>p</code>	<code>not p</code>
<code>True</code>	<code>False</code>
<code>False</code>	<code>True</code>

<code>p</code>	<code>q</code>	<code>p and q</code>	<code>p or q</code>
<code>True</code>	<code>True</code>	<code>True</code>	<code>True</code>
<code>True</code>	<code>False</code>	<code>False</code>	<code>True</code>
<code>False</code>	<code>True</code>	<code>False</code>	<code>True</code>
<code>False</code>	<code>False</code>	<code>False</code>	<code>False</code>

**Ejercicios:** ¿Qué valor devuelve cada una de estas expresiones? (¡Pensar!)

- a) `len('hola')>1 and 1+1==3`
- b) `len('hola')>1 or 1+1==3`
- c) `len('hola')>1 and not (1+1==3)`

## Tipo de datos `bool` · Ejercicio

Tenemos definida la siguiente función `es_par`:

```
def es_par(n:int) -> bool:
    ''' Requiere: nada.
        Devuelve: True si n es par, False si es impar.
    '''
    b:bool = (n % 2 == 0)
    return b

print(es_par(7)) # imprime False
print(es_par(8)) # imprime True
```

Escribir una función que reciba un número entero y devuelva el string **El número es par** o bien **El número es impar**, según corresponda.

(Sugerencia: usar una instrucción `if` que invoque a la función `es_par` en la condición.)



# Ciclos

```
1 i:int = 10
2 while i>=1:
3     print(i)
4     i = i - 1
5 print('despegue!')
```



**while** *CONDICIÓN*:  
*BLOQUE*

*CONDICIÓN* es una expresión de tipo bool.  
*BLOQUE* es un bloque de código.

## Flujo de ejecución:

1. Se evalúa *CONDICIÓN*.
2. Si es *verdadera*: se ejecuta *BLOQUE* y se vuelve al paso 1.
3. Si es *falsa*: se termina.

*CONTINUARÁ...*

# Repaso de la clase de hoy

- ▶ Condicionales: `if/elif/else`
- ▶ Tipo de datos `bool`:
  - ▶ Valores `True`, `False`.
  - ▶ Operaciones `not`, `and`, `or`.
  - ▶ Funciones que devuelven `bool`.
- ▶ Un poquito de ciclos...

## **Bibliografía complementaria:**

- ▶ APPP2, secciones 4.1 a 4.7.
- ▶ HTCSP3, secciones 5.1 a 5.8.

Con lo visto, ya pueden resolver hasta el Ejercicio 2 (inclusive) de la Guía de Ejercicios 3.