

# Práctica 02. De la programación de bloques a Python (Tema 3)

## Enunciado de las prácticas

**Programación:** convertir los programas realizados en Scratch al lenguaje de programación Python. Se tomará como punto de partida, la solución propuesta por el profesor.

**IMPORTANTE:** Cada programa se escribirá en una celda y se solucionará en este mismo notebook en la celda indicada.

Como venimos de un lenguaje muy visual como Scratch, haremos algunas simplificaciones:

- El gato será un asterisco: \*.
- Un gato grande, serán dos asteriscos: \*\*.
- Si se esconde: (escondido)
- Cuando hable el gato, se mostrar por pantalla: \*. ¡Hola!
- Cuando avanza el gato un paso, se motrará un . *Si da 3 pasos*: \*\*.
- Cuando retroceda el gato, se motrará un -. Si retrocede 3 pasos: ---.
- El proceso de espera se ejecutará de la siguiente forma: se escribirá en la parte superior de la celda la intrucción "import time", ya veremos más adelante lo que hace. Para crear la espera, escribiremos "time.sleep(segundos)".

```
In [3]: import time
print("Hola")
time.sleep(1)
print("Ha pasado 1 segundo")

Hola
Ha pasado 1 segundo
```

## Programa #1: Programación imperativa básica



### Secuencializar o agrupar tareas manuales para que sean ejecutadas en un orden concreto

- El gato debe decir "Hola" durante 1 segundo.
- A continuación se va hacer grande fijando su tamaño a 200%.
- Esperaremos 1 segundo.
- Ocultaremos el gato.
- Esperaremos 1 segundo.
- Mostraremos el gato.
- Lo dejaremos en su tamaño original, 100%.
- Dirá (de forma pensativa) "Uy, ¿qué ha pasado?" durante 1 segundo.

### Ejemplo de ejecución

```
*: ¡Hola!
**
(escondido)
**
*
Uy, ¿qué ha pasado?
```

### Solución en la siguiente celda

```
In [ ]:
```

## Programa #2: Condicionales



### Generar distintas salidas en función de la información de entrada

- El gato debe decir "¡Hola, soy el gato IA!" durante 1 segundo.
- Preguntará "Dime una palabra en español y la traduciré al inglés": se pedirá por teclado al usuario.
- Se evaluará la palabra introducida (almacenada en la variable respuesta). Si su contenido es igual a Hola:
  - El gato debe decir "Hola se dice en inglés: hello" durante 2 segundo.
- Si no...
  - El gato debe decir "¡Vaya! Ahí me has pillado" durante 2 segundo.

### Ejemplo de ejecución

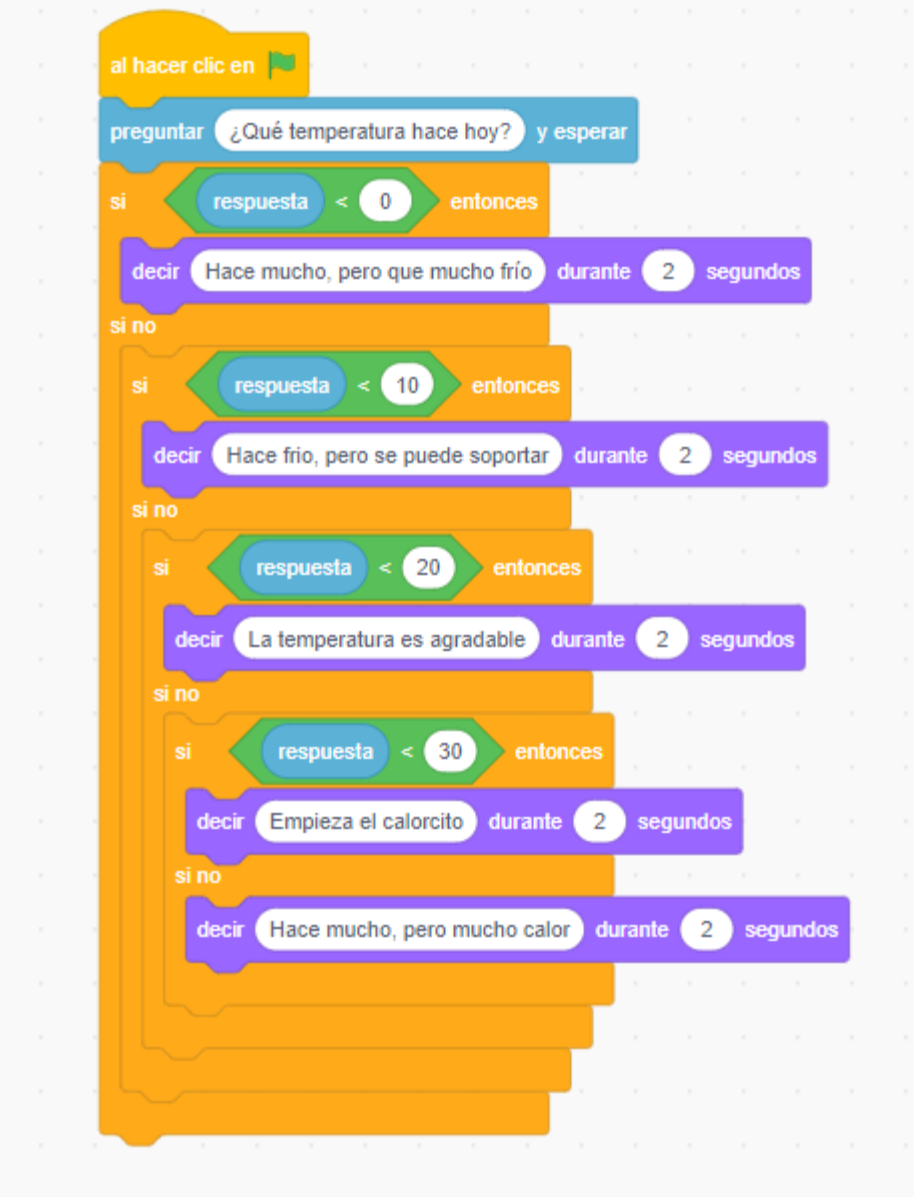
Se leerá la palabra por teclado.

```
¡Hola, soy el gato IA!
Dime una palabra en español y la traduciré al inglés: Hola
Hola se dice en inglés: hello
```

### Solución en la siguiente celda

```
In [ ]:
```

## Programa #3: Condicionales anidados



### Generar distintas salidas en función de la información de entrada - Avanzado

Se creará un sistema inteligente basado en reglas:

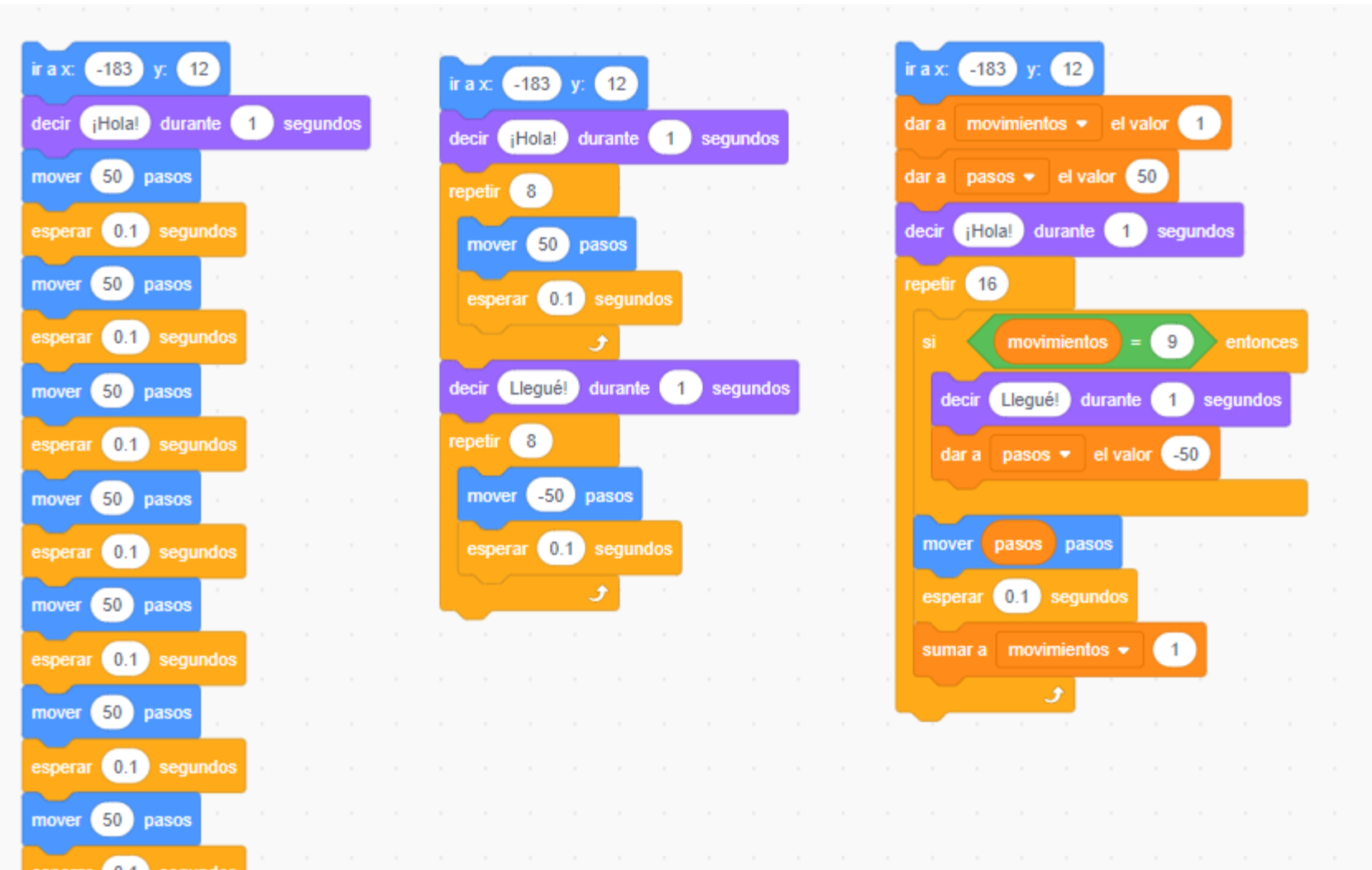
- Preguntará "¿Qué temperatura hace hoy?": La temperatura se inicializará directamente en una variable, no se pedirá por teclado.  
temperatura = 25
- Si la temperatura es inferior a 0 grados el gato dirá: "Hace mucho, pero que mucho frío" durante 2 segundos.
- Si la temperatura está entre [0, 9] el gato dirá: "Hace frío, pero se puede soportar" durante 2 segundos.
- Si la temperatura está entre [10, 19] el gato dirá: "La temperatura es agradable" durante 2 segundos.
- Si la temperatura está entre [20, 29] el gato dirá: "Empieza el calorcito" durante 2 segundos.
- Si la temperatura es superior a 30 grados el gato dirá: "Hace mucho, pero mucho calor" durante 2 segundos.

### Trabajar con if anidados, como en el pseudocódigo

### Solución en la siguiente celda

```
In [ ]:
```

## Programa #4: Bucles



### Automatizar y optimizar la repetición de una serie de tareas

Se creará un programa, similar al desarrollado en Scratch, en el cuál se diga en primer lugar "¡Hola!", a continuación se mostrará un "\*" por cada paso hasta dar 8 y se mostrará el texto "¡Llegué!". Se mostrará el texto "Volviendo" y de vuelta se mostrará un "-" por cada paso de vuelta. Al finalizar se mostrará el texto "¡Volví!".

### Ejemplo de ejecución

Se mostrará dejando 1 segundo entre cada paso (vez que se muestra al gato). Es como si el gato se moviera verticalmente, en lugar de horizontal.

```
¡Hola!
*
*
*
*
*
*
*
*
¡Llegué! Volviendo
-
-
-
-
-
-
-
¡Volví!
```

### Solución en la siguiente celda sin utilizar bucles

```
In [ ]:
```

### Solución en la siguiente celda con bucles

```
In [ ]:
```

### Solución en la siguiente celda con un solo bucle

```
In [ ]:
```