



**Multitarea**



# Multitarea con threading



## Índice

- La clase Thread
- Ejemplo de tarea
- Otros métodos de threading



## La clase Thread

- El módulo threading, incorporado en Python a partir de la versión 3, proporciona la clase Thread para implementación de aplicaciones multitarea
- De cara a crear un tipo de tarea, se creará una subclase de Thread y se sobrescribirá el método ***run()***
- Para crear una tarea y ponerla en ejecución, se creará un objeto de esta subclase de Thread y se llamará al método ***start()***



## Ejemplo de tarea

- Creación de una tarea que se encargue de imprimir la tabla de multiplicar de un número
- Se crearán tres tareas para imprimir de forma concurrente las tablas de multiplicar de tres números



## Otros métodos de threading

➤ Además del acceso a la clase Thread, la librería threading proporciona otros métodos de interés:

- **activeCount()**. Devuelve el total de threads activos
- **currentThread()**. Devuelve una referencia al objeto Thread que está actualmente en ejecución
- **enumerate()**. Devuelve una lista con todos los Thread que están actualmente activos:

```
#muestra los nombres de todos los thread activos
for t in threading.enumerate():
    print(t.getName())
```