C O R U T I N A S

```python
async def hola():
  return "Hola"

foo() # devuelve una corutina
await foo() # devuelve "Hola"
```

⚠️no se puede esperar más de una vez⚠️
⚠️si no se espera, no se empieza⚠️

F U T U R O S

```python
f = asyncio
  .get_running_loop()
  .create_future()
f.set_result(...) # set_exception()
```

🪦de bajo nivel!🪦
🎨todo a mano!🎨

T A R E A S

```python
# el argumento es una corutina
t = await asyncio.create_task(foo())
t.add_done_callback(fn_normal)
t.done() # booleano
t.cancelled() # booleano
t.exception() # None | Exception
t.result() # el resultado
```

💥empieza a ejecutarse de una, no se espera💥
💥se puede esperar > 1 vez💥

G A T H E R

```python
await asyncio.gather( # devuelve una lista
  tarea1,
  tarea2,
  ...,
  return_exceptions=True,
)
```

T A S K G R O U P

```python
with asyncio.TaskGroup() as tg:
  tg.create_task(...)
```

🐍python 3.11🐍
🧹se cancela toda tarea del grupo si hay error🧹

⚠️ Evite lavavajillas!