

CODI TRIANGLE AMB FUNCIONS

https://py3.codeskulptor.org/#user310_VkyCJgHAwg_0.py

CODI TRiangle amb classes

https://py3.codeskulptor.org/#user310_VkyCJgHAwg_3.py

ANÀLISI COMPARATIVA: FUNCIONS VS POO (TRIANGLE)

OBJECTIU DEL PROGRAMA

Calcular el perímetre d'un triangle i verificar si és vàlid (segons la desigualtat triangular).

VERSIÓ 1: FUNCIONS

```
def estriangulo(lados): ...
```

```
def perimetro(l): ...
```

Observacions:

- Es treballa amb tuples o llistes com a contenidors genèrics de dades.
- Les funcions són externes: no pertanyen als triangles.
- Cal passar explícitament els valors (lados) a cada funció.
- El triangle no és una entitat clara: només un conjunt de números.
- Si tenim diversos triangles, hem de gestionar múltiples tuples i passar-les una a una.

VERSIÓ 2: CLASSE Triangle (POO)

```
class Triangle:
```

```
    def __init__(self, c1, c2, c3): ...
```

```
    def estriangulo(self): ...
```

```
    def perimetro(self): ...
```

Observacions:

- S'encapsulen les dades i els mètodes en una entitat concreta: Triangle.
- Cada triangle és un objecte que guarda estat propi.

- Més llegible i natural: `t1.perimetro()` en comptes de `perimetro(lados)`.
- Es pot modificar internament (`t2.costat1 = 8`) i l'objecte conserva coherència.
- Permet treballar amb diversos triangles sense repetir codi o perdre claredat.

AVANTATGES DE LA POO EN AQUEST CAS

Aspecte	Funcions	Classes (POO)
-----	-----	-----
Modelatge conceptual	Tupla anònima	Objecte Triangle ben definit
Llegibilitat	<code>perimetro(l)</code>	<code>t1.perimetro()</code>
Escalabilitat	Difícil de mantenir	Afegir mètodes nous és directe
Reutilització	Poc reutilitzable	Objectes reusables i combinables
Mantenibilitat	Codi dispers	Codi agrupat i coherent

CONCLUSIÓ PEDAGÒGICA

Quan un conjunt de dades i operacions tenen coherència semàntica (com en el cas d'un triangle), crear una classe no només fa el codi més llegible, sinó més robust, escalable i natural.