El Zen de Python
Beautiful is better than ugly.

 Explicit is better than implicit.

 Simple is better than complex.
 Complex is better than complicated.

 Flat is better than nested.

 Sparse is better than dense.

 Readability counts.

 Special cases aren't special enough to break the rules.
 Although practicality beats purity.

 Errors should never pass silently.
 Unless explicitly silenced.

 In the face of ambiguity, refuse the temptation to guess.

 There should be one-- and preferably only one --obvious way to do it.
 Although that way may not be obvious at first unless you're Dutch.

 Now is better than never.
 Although never is often better than *right* now.

 If the implementation is hard to explain, it's a bad idea.
 If the implementation is easy to explain, it may be a good idea.

 Namespaces are one honking great idea -- let's do more of those!
Bello > Feo

Beautiful is better than ugly.

Castillo Neuschwanstein. Bavaria, Alemania.

Museo Cantonal de Bellas Artes. Lausanne, Suiza.
def total_articulos_en_carrito(carrito_json: str) -> int:
    """Calcula costo total del carrito de compras. """
    total = 0
    for costo_articulo in costos_por_articulo(carrito_json):
        total += costo_articulo
    return total

def costos_por_articulo(carrito_json: str) -> List[int]:
    """Obtiene costos de cada artículo del carrito.
    Regresa una lista con los costos individuales. """
    articulos = json.loads(carrito_json).get("articulos")
    return [articulo["costo"] for articulo in articulos]

def calcular(datos):
    resultado = 0
    for i in procesar(datos):
        resultado += i
    return resultado

def procesar(datos):
    return [d["costo"] for d in loads(datos).get("articulos")]

Explicit is better than implicit.
Simple > Complejo > Complicado

Simple is better than complex.
Complex is better than complicated.

KISS - Keep it simple, stupid.
Mantenlo estúpidamente simple
Plano > Anidado

Flat is better than nested.

McCabe

Caminos Independientes ➞ Complejidad
Plano > Anidado

Flat is better than nested.

Jerarquías

Acceso jerárquico  ➡️ Complicado e ineficiente
Escaso (Esparcido) > Denso

Sparse is better than dense.
Explica por qué

De forma evidente

Hace lo que debe
Reglas & Practicalidad

Special cases aren’t special enough to break the rules. Although practicality beats purity.
Errores ruidosos, a consideración.

Errors should never pass silently. Unless explicitly silenced.
No adivinar

In the face of ambiguity, refuse the temptation to guess.

```python
>>> "a" > 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: '>' not supported between instances of 'str' and 'int'
```
Una manera de hacerlo

There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you’re Dutch.
Ahora > Nunca > Justo ahora

Now is better than never. Although never is often better than *right* now.

YAGNI = “You ain’t gonna need it”

No lo vas a necesitar
Fácil de explicar

If the implementation is hard to explain, it’s a bad idea.
If the implementation is easy to explain, it may be a good idea.
Espacios de nombre

Namespaces are one honking great idea – let’s do more of those!