Step-by-Step

1. Define a class
   - Must implement **`__call__`** method
2. Create an instance of the class
   - This is the callable object
   - Similar to a **`def`**, but…

Dun!

Dun!


Dun!

Step-by-Step

1. Define a class
   - Must implement **`__call__`** method
2. Create an instance of the class
   - This is the callable object
   - Similar to a **`def`**, but…

Dun! Dun! Dun!

Callables Can Have Hysteresis

We'll Break This Down Below

```
class Factorial:
    def __init__(self):
        self.previous = {}
    def __call__(self, n: int) -> int:
        if n not in self.previous:
            self.previous[n] = self.compute(n)
        return self.previous[n]
    def compute(self, n: int) -> int:
        if n == 0: return 1
        return n*self.__call__(n-1)
```