Track_Calls Decorating the **factorial** Function

First **factorial** is defined, which binds its name to a function object.

Writing

```
factorial = Track_Calls(factorial)
```

(or using `@Track_Calls`) rebinds **factorial** to refer to a **Track_Calls** object, whose `f` attribute refers to **factorial**’s original function object (and `calls` is bound to 0, which `__call__` will examine and update).

Any direct call to `factorial` (or recursive call to `factorial` inside the function object) will look up the **Track_Calls** object bound to `factorial` and execute the `__call__` method defined in that class.

```python
def factorial(n):
    if n == 0
        return 1
    else:
        return n*factorial(n-1)
```

```
f0 = factorial
```
Memoize Decorating the \texttt{fib} Function

First \texttt{fib} is defined, which binds its name to a function object.

Writing \texttt{fib = Memoize(fib)} (or using \texttt{@Memoize}) rebinds \texttt{fib} to refer to a \texttt{Memoize} object, whose \texttt{f} attribute refers to \texttt{fib}'s original function object (and \texttt{cache} is bound to an empty \texttt{dict}, which \texttt{__call__} will examine and update).

Any direct call to \texttt{fib} (or recursive call to \texttt{fib} inside the function object) will look up the \texttt{Memoize} object bound to \texttt{fib} and execute the \texttt{__call__} method defined in that class.