

Local Functions so Far

Boring!

No interaction with the enclosing scope

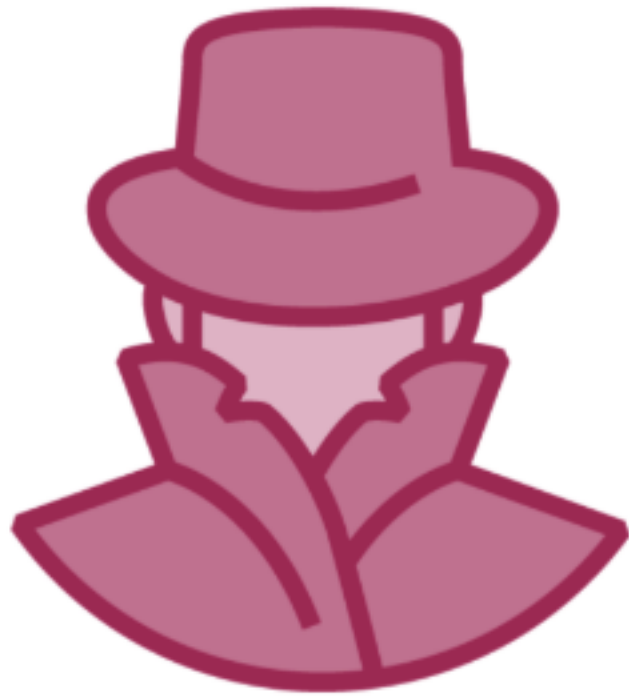
Enclosing scope

We've seen that they can access the enclosing scope

Returned

We've seen that they can be returned from the enclosing function

A Small Mystery



How does a returned local function retain access to its enclosing scope?

Once the local function is returned, the enclosing scope is gone!

How can the returned local function continue to operate?

Closure

Records objects from enclosing scopes

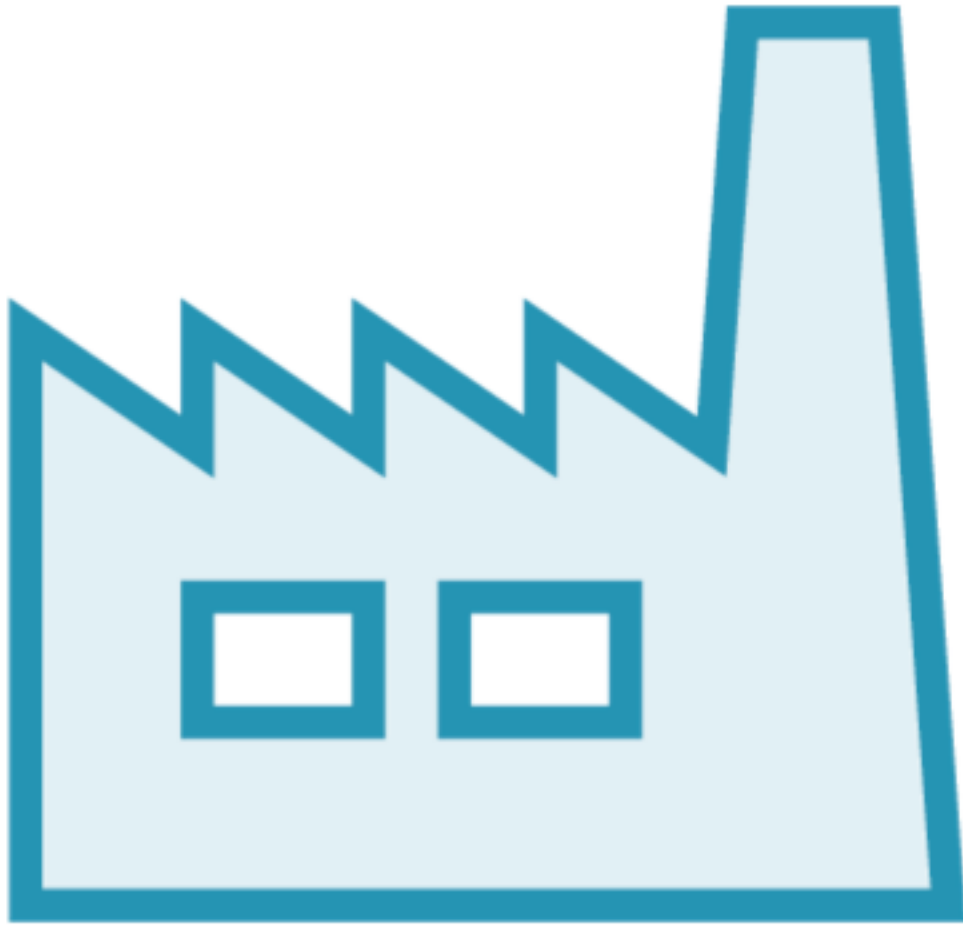
Keeps recorded objects alive for use after the enclosing scope is gone

Implemented with the `__closure__` attribute

Closures

```
>>> def enclosing():
...     x = 'closed over'
...     def local_func():
...         print(x)
...     return local_func
...
>>> lf = enclosing()
>>> lf()
closed over
>>> lf.__closure__
(<cell at 0x10377eb80: str object at 0x1038647b0>, )
>>>
```

Function Factories



Functions that return other functions

Returned functions use both their own arguments as well as arguments to the factory

Combination of runtime function definition and closures

Function Factories

```
>>> def raise_to(exp):
...     def raise_to_exp(x):
...         return pow(x, exp)
...     return raise_to_exp
...
>>> square = raise_to(2)
>>> square.__closure__
(<cell at 0x10e0055e0: int object at 0x10dd939d0>,)
>>> square(5)
25
>>> square(9)
81
>>> square(1234)
1522756
>>> cube = raise_to(3)
>>> cube(3)
27
>>> cube(10)
1000
>>> cube(23)
12167
>>>
```