
TAP version 13
(anonymous)
ok 1 Solution loaded
ok 2 The structure is correct
1..2
tests 2
pass 2
ok
Success!
You correctly created the variable bar, lexically scoped inside the function foo(), great work.
Solution
For comparison, here is a possible solution, so you can compare notes:
function foo() {
var bar;
}
# Next lesson
Execute @workshoppers/scope-chains-closures to move on to the next lesson: Scope Chains.

```
TAP version 13
# (anonymous)
ok 1 Solution loaded
ok 2 The structure is correct

1..2
# tests 2
# pass  2

# ok


# Success!

You created a scope chain using lexical scoping and var statements!



--------------------------------------------------------------------------------

# Solution

The scope chain you created now looks like this:

    (global)
        ↑
        |
      foo()
     var bar
        ↑
        |
       zip()
     var quux

By following the arrows, we can see zip() has access to var bar, but not the
other way around.
```

```
TAP version 13
# (anonymous)
ok 1 Solution loaded
ok 2 The structure is correct

1..2
# tests 2
# pass  2

# ok


# Success!

You assigned a value to quux, even though foo() doesn't have access to the
quux inside zip().



--------------------------------------------------------------------------------


# Solution

The scope chain of the solution looks like this:

    (global)
      quux
        ↑
        |
      foo()
     var bar
        ↑
        |
       zip()
     var quux
```

```
TAP version 13
# (anonymous)
ok 1 Solution loaded
ok 2 The structure is correct

1..2
# tests 2
# pass  2

# ok


# Success!

Awesome stuff - you closed over the variable bar inside zip, then returned
zip.



--------------------------------------------------------------------------------

# Solution

Let's look at the scope chain for your solution:

      foo()
     var bar
    return zip
        ↑
        |
      zip()
    bar = true

By referencing bar within zip, we have created a Closure where zip() closes over
 the variable bar from its parent scope foo().

Since we are returning the function zip, the reference to bar is maintained (and
 hence the closure is maintained) until zip is no longer required.
```