

JAVASCRIPT

CLOSURES

WHAT ARE CLOSURES?

CLOSURES

CLOSURES

- ▶ function within a function (enclosed scope)

CLOSURES

- ▶ function within a function (enclosed scope)
- ▶ inner function is returned from outer function

CLOSURES – ENCLOSED SCOPE

CLOSURES – ENCLOSED SCOPE

```
function outer() {
```

```
}
```

CLOSURES – ENCLOSED SCOPE

```
function outer() {  
    function inner() {  
  
    }  
}
```


CLOSURES – ENCLOSED SCOPE

Outer function scope

```
function outer() {  
    function inner() {  
  
    }  
}
```

CLOSURES – ENCLOSED SCOPE

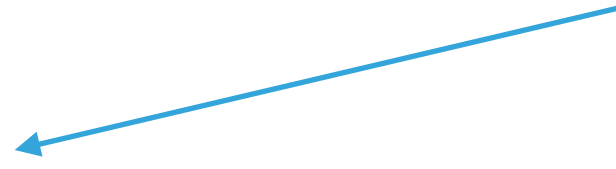
Outer function scope

```
function outer() {
```

```
    function inner() {
```

```
    }
```

```
}
```



CLOSURES – ENCLOSED SCOPE

Outer function scope

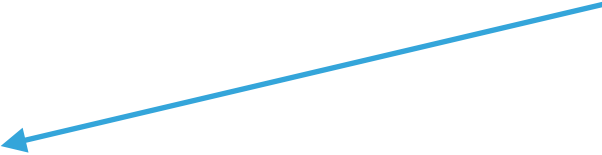
function outer() {

Inner function scope

function inner() {

}

}



CLOSURES – ENCLOSED SCOPE

Outer function scope

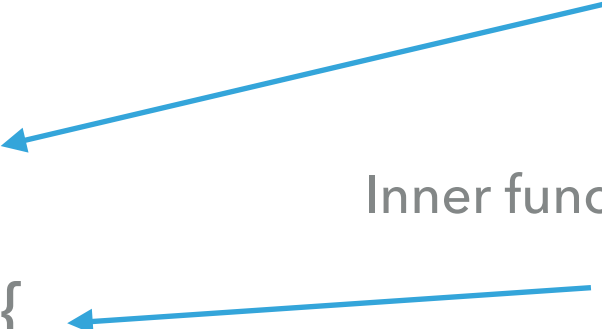
function outer() {

Inner function scope

function inner() {

}

}



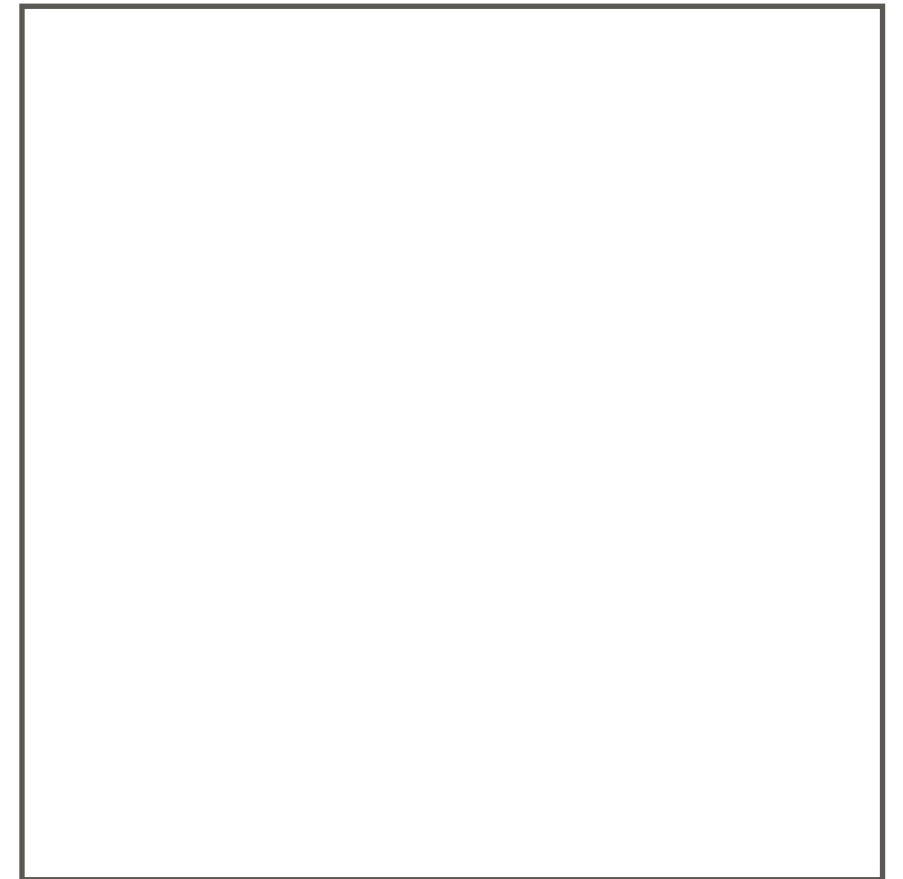
The diagram illustrates the concept of function scopes in JavaScript. It shows a function named 'outer()' which contains a nested function named 'inner()'. Two blue arrows point to the opening curly braces of these functions. The arrow pointing to 'function outer() {' is labeled 'Outer function scope', indicating that this function defines the outer scope. The arrow pointing to 'function inner() {' is labeled 'Inner function scope', indicating that this function defines an inner scope that is enclosed within the outer scope of 'outer()'.

CLOSURES – ENCLOSED SCOPE

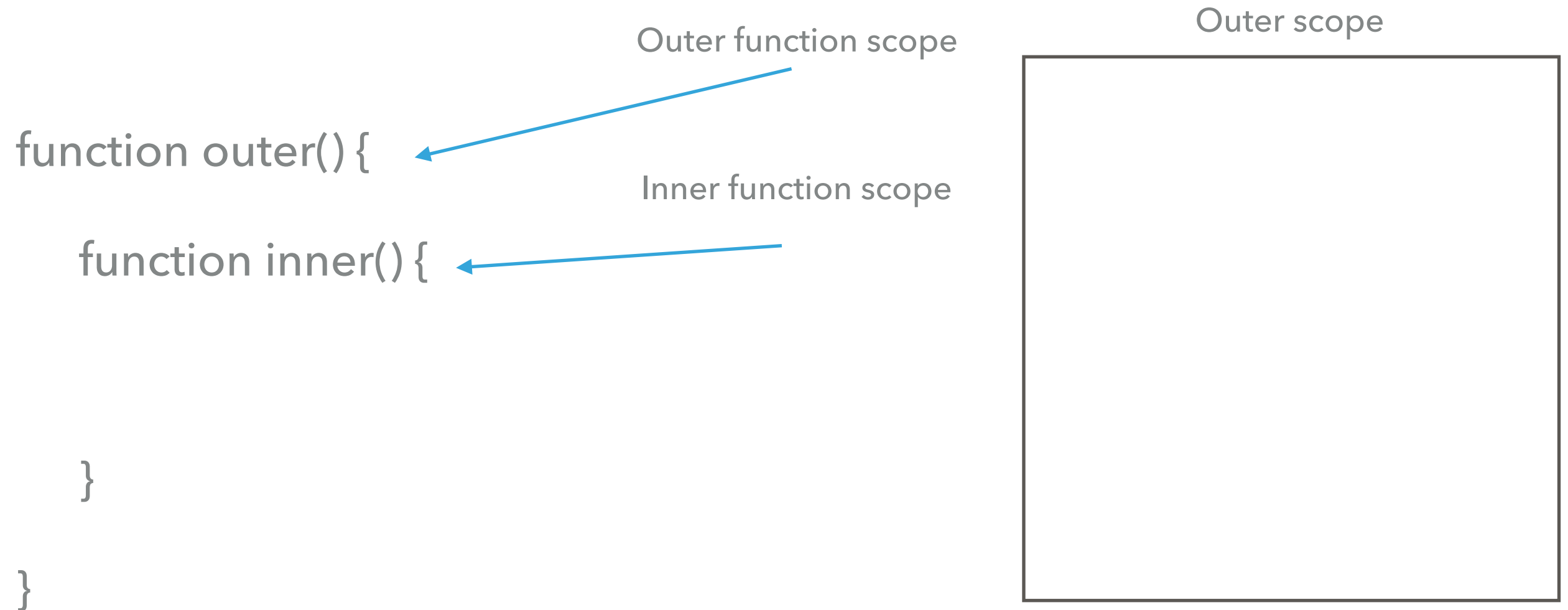
Outer function scope

```
function outer() {  
  Inner function scope  
  function inner() {  
  
  }  
}
```

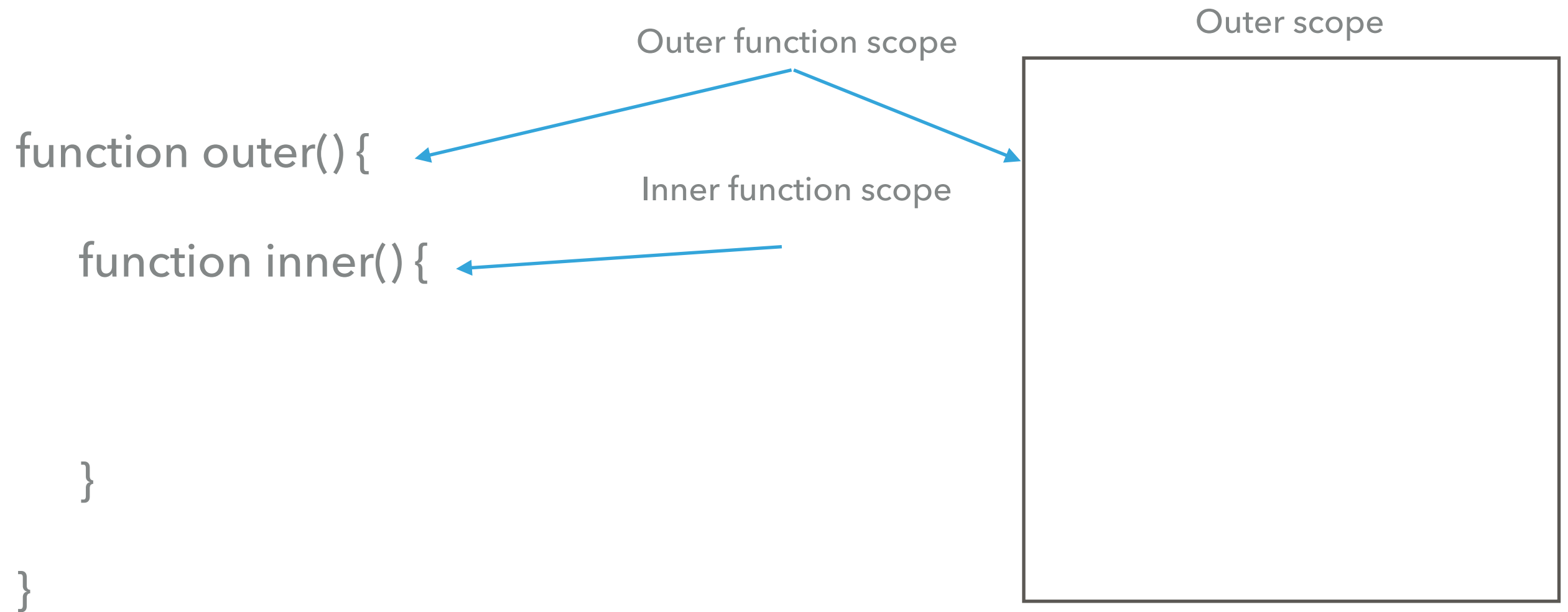
Inner function scope



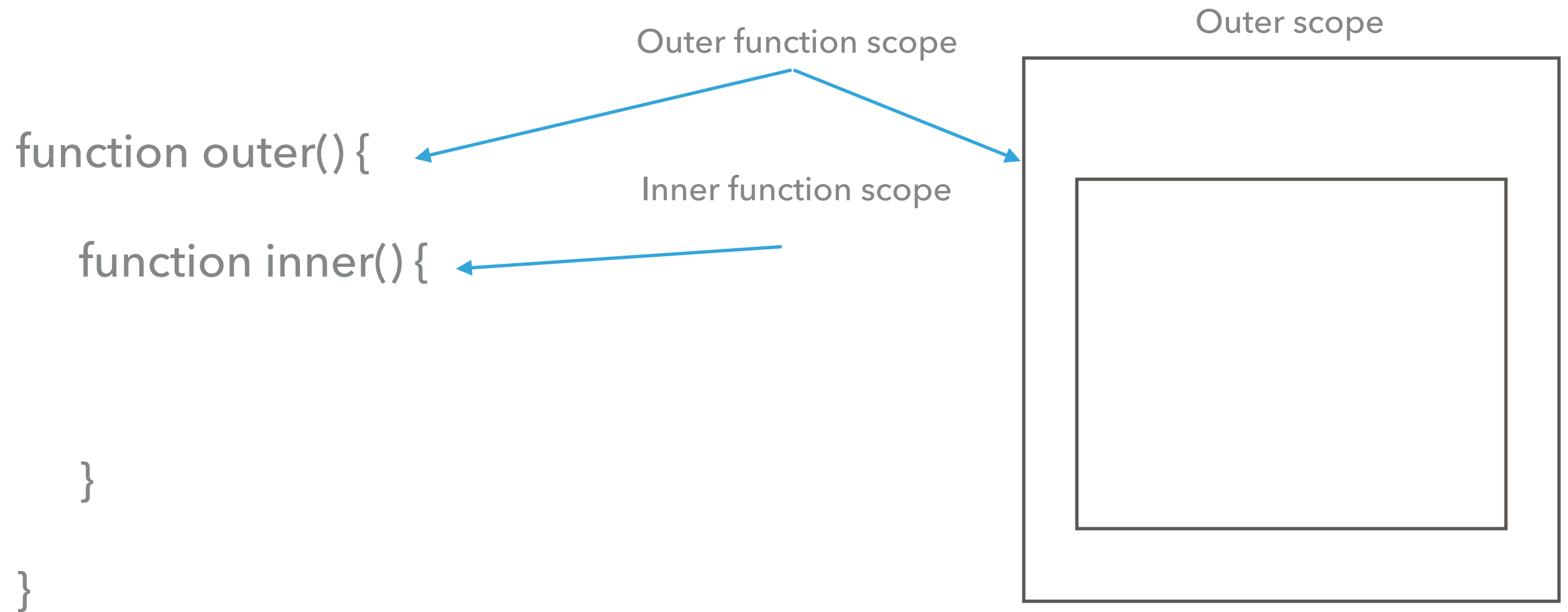
CLOSURES – ENCLOSED SCOPE



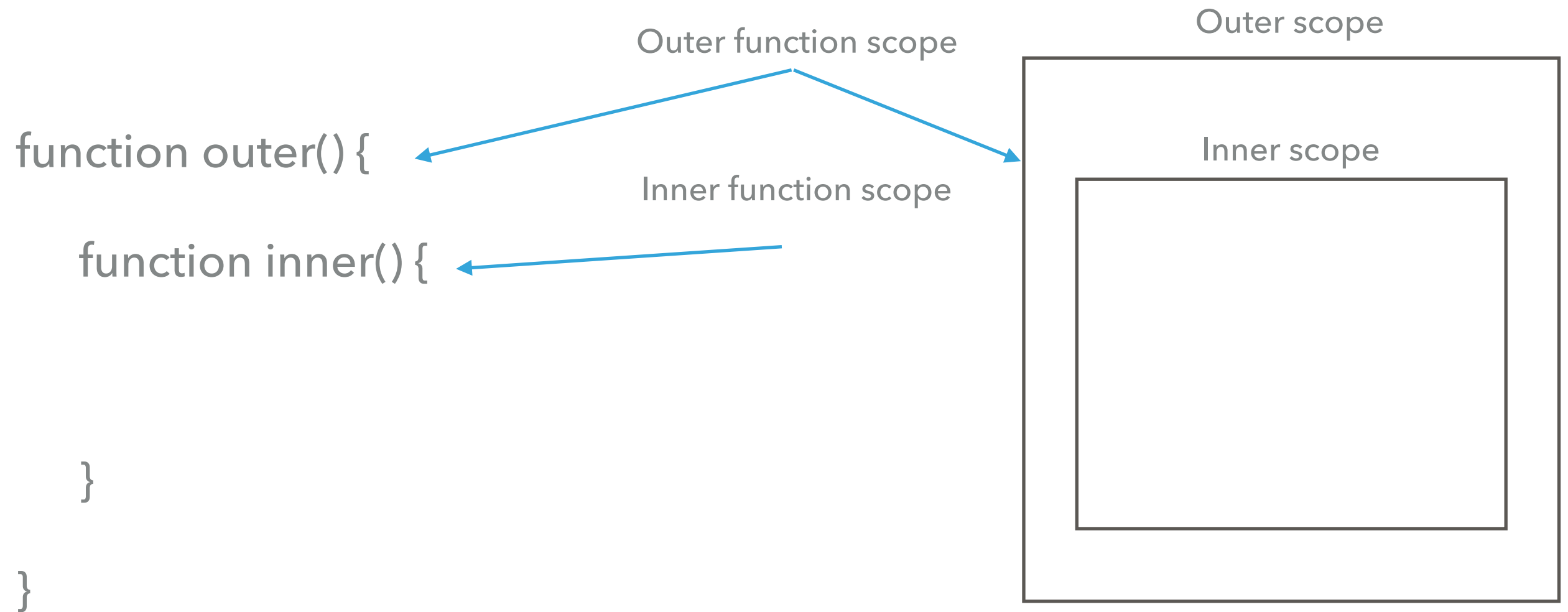
CLOSURES – ENCLOSED SCOPE



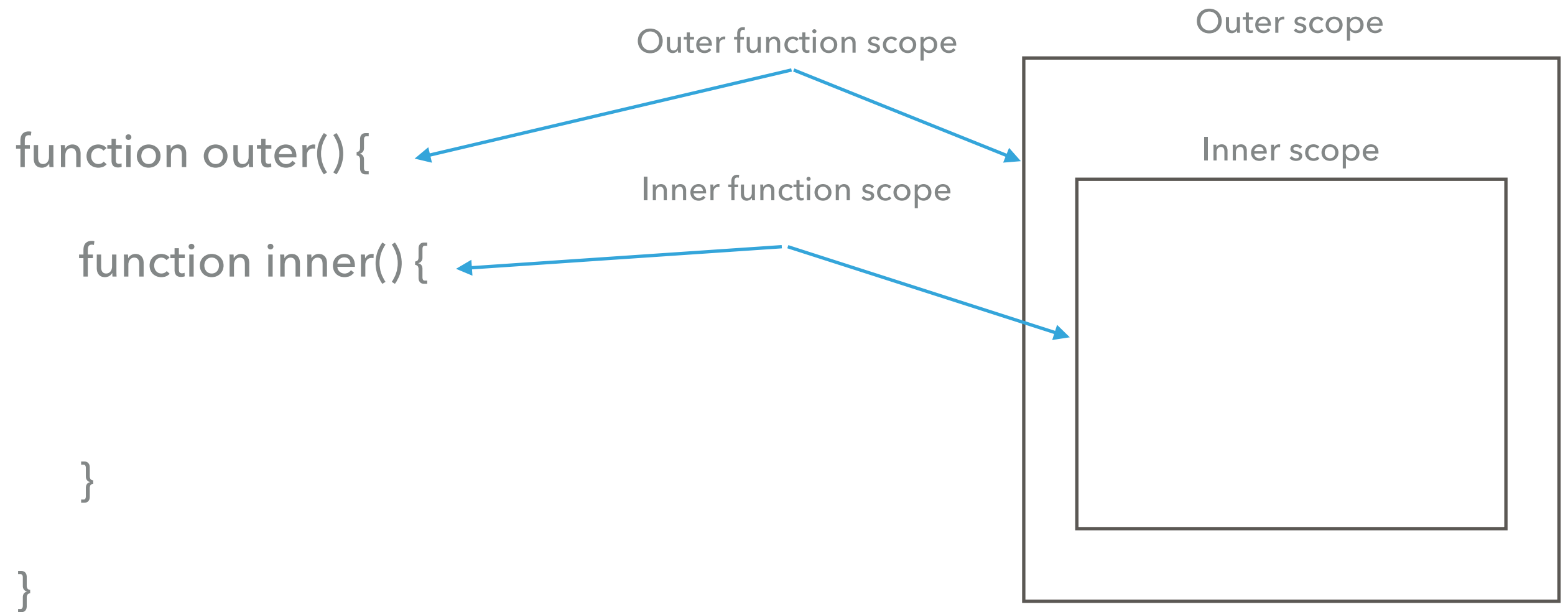
CLOSURES – ENCLOSED SCOPE



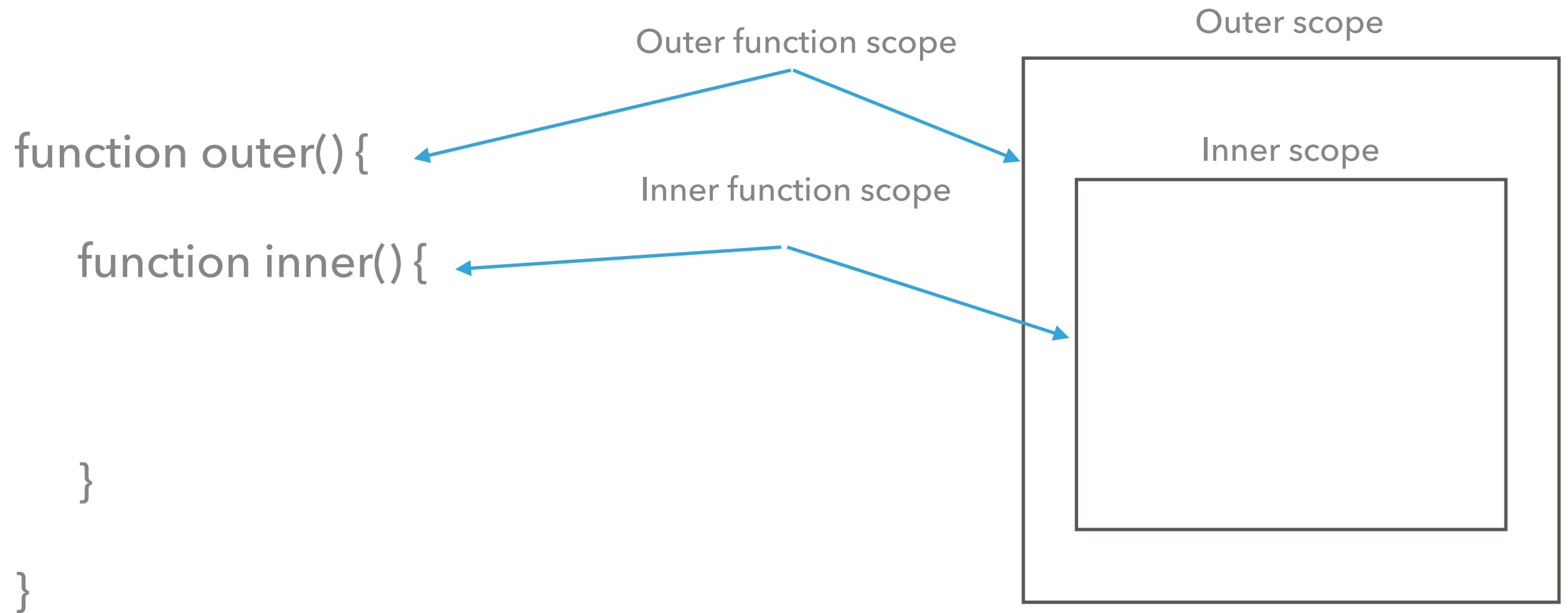
CLOSURES – ENCLOSED SCOPE



CLOSURES – ENCLOSED SCOPE

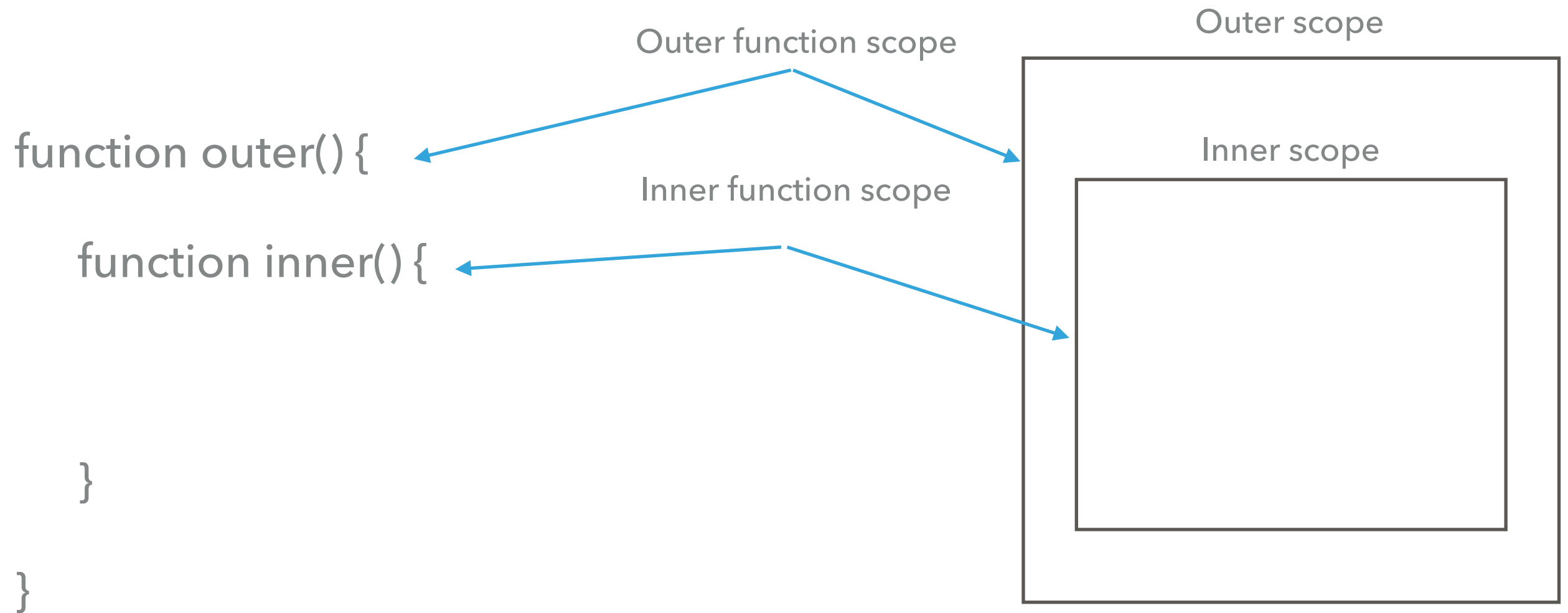


CLOSURES – ENCLOSED SCOPE



The outer scope has enclosed the inner scope.

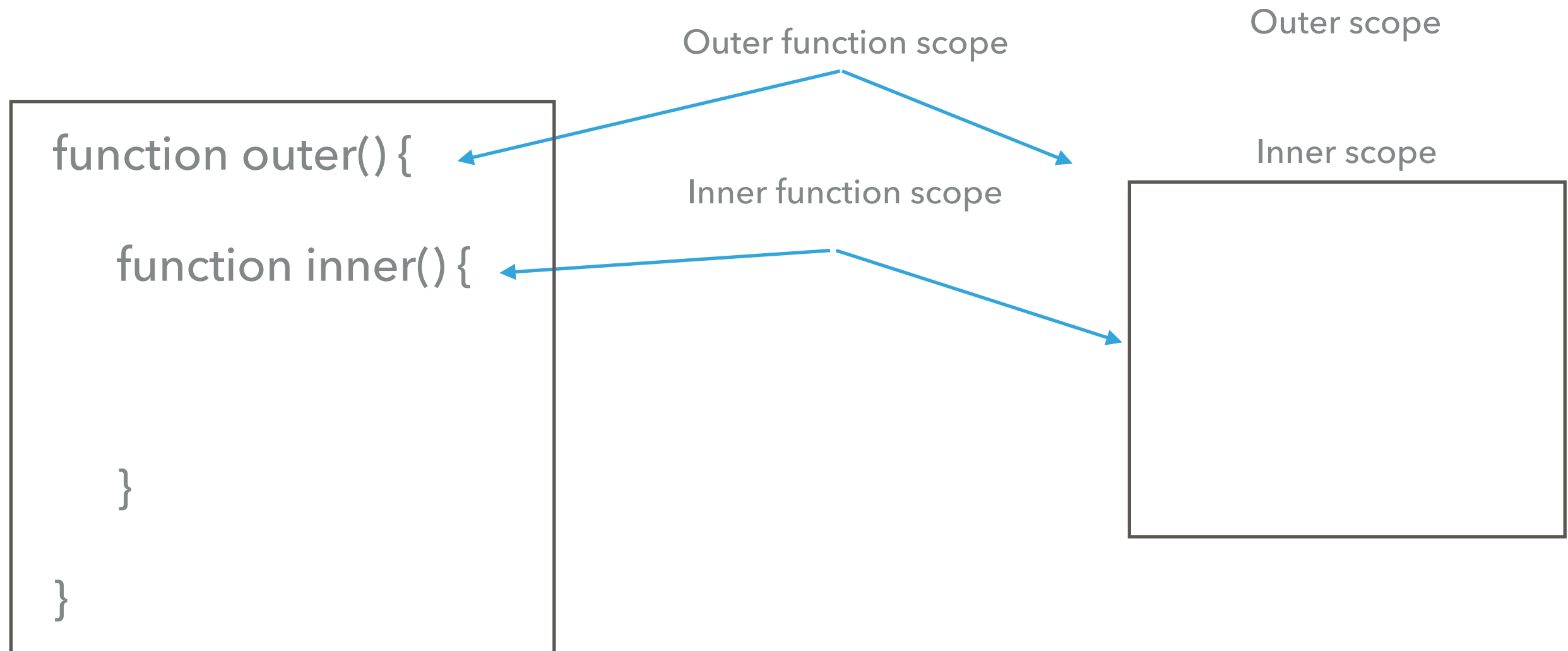
CLOSURES – ENCLOSED SCOPE



The outer scope has enclosed the inner scope.

Simple terms: **function within a function**

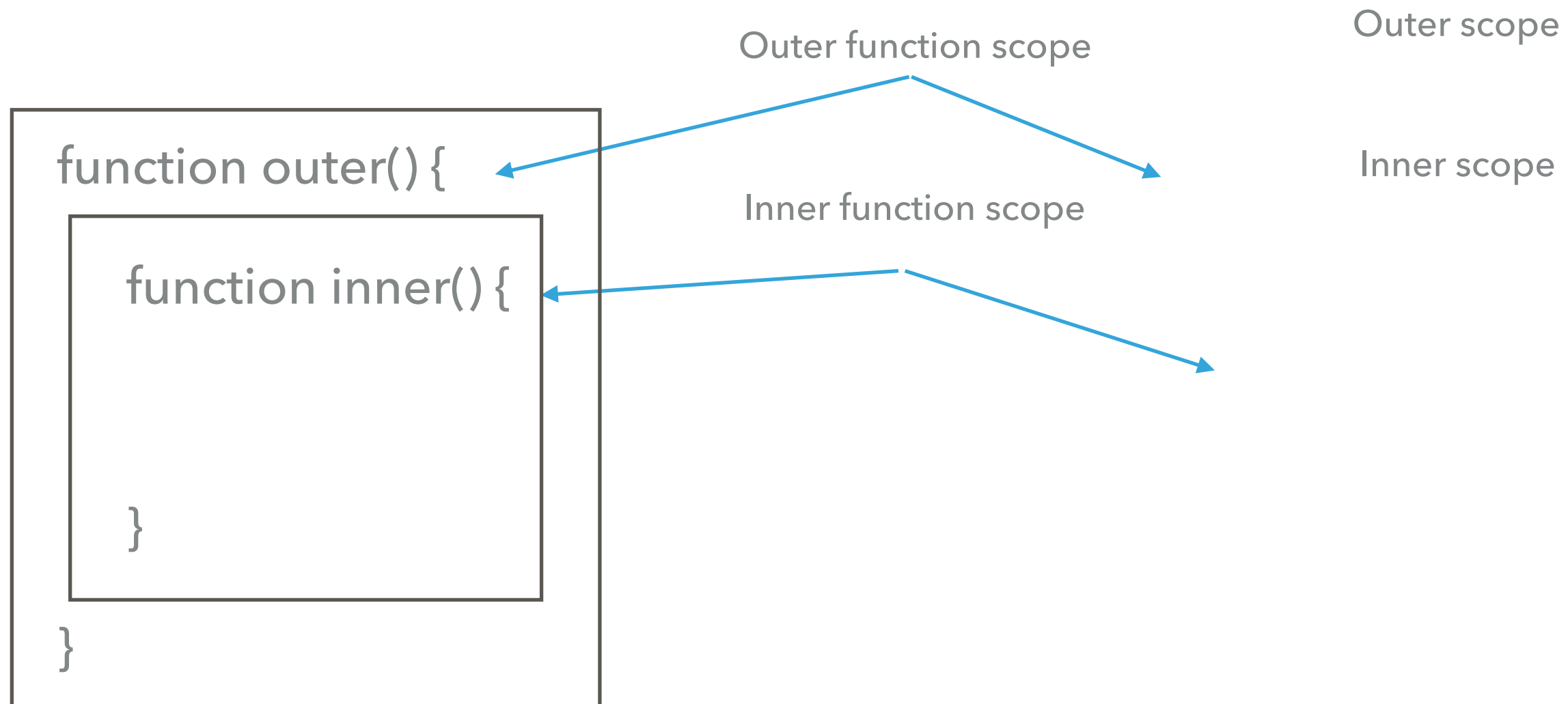
CLOSURES – ENCLOSED SCOPE



The outer scope has enclosed the inner scope.

Simple terms: **function within a function**

CLOSURES – ENCLOSED SCOPE



The outer scope has enclosed the inner scope.

Simple terms: **function within a function**

CLOSURES – SCOPE REVIEW

CLOSURES – SCOPE REVIEW

```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```

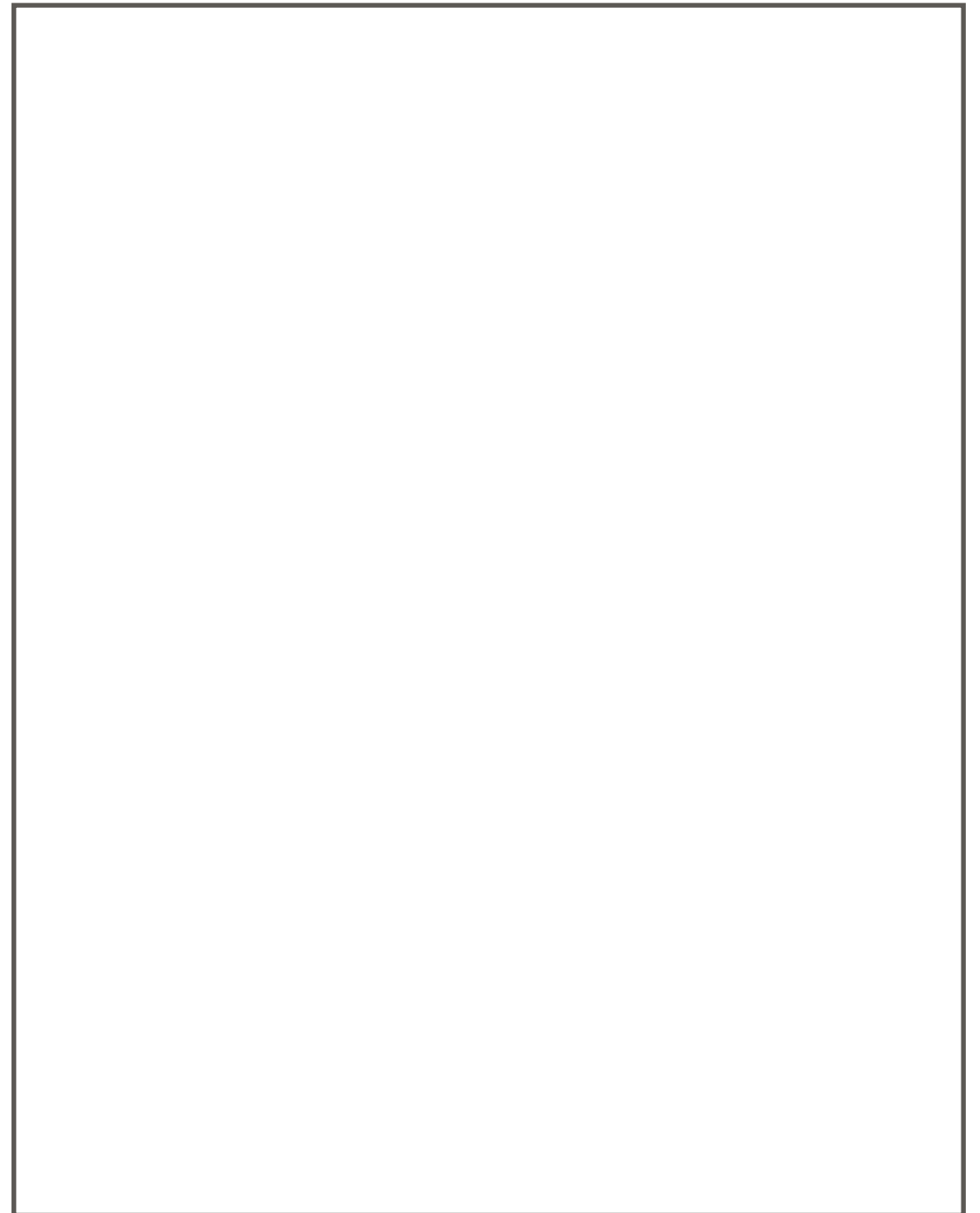

CLOSURES – SCOPE REVIEW

```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```

Can inner function access num?

CLOSURES – SCOPE REVIEW

```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```



Can inner function access num?

CLOSURES – SCOPE REVIEW

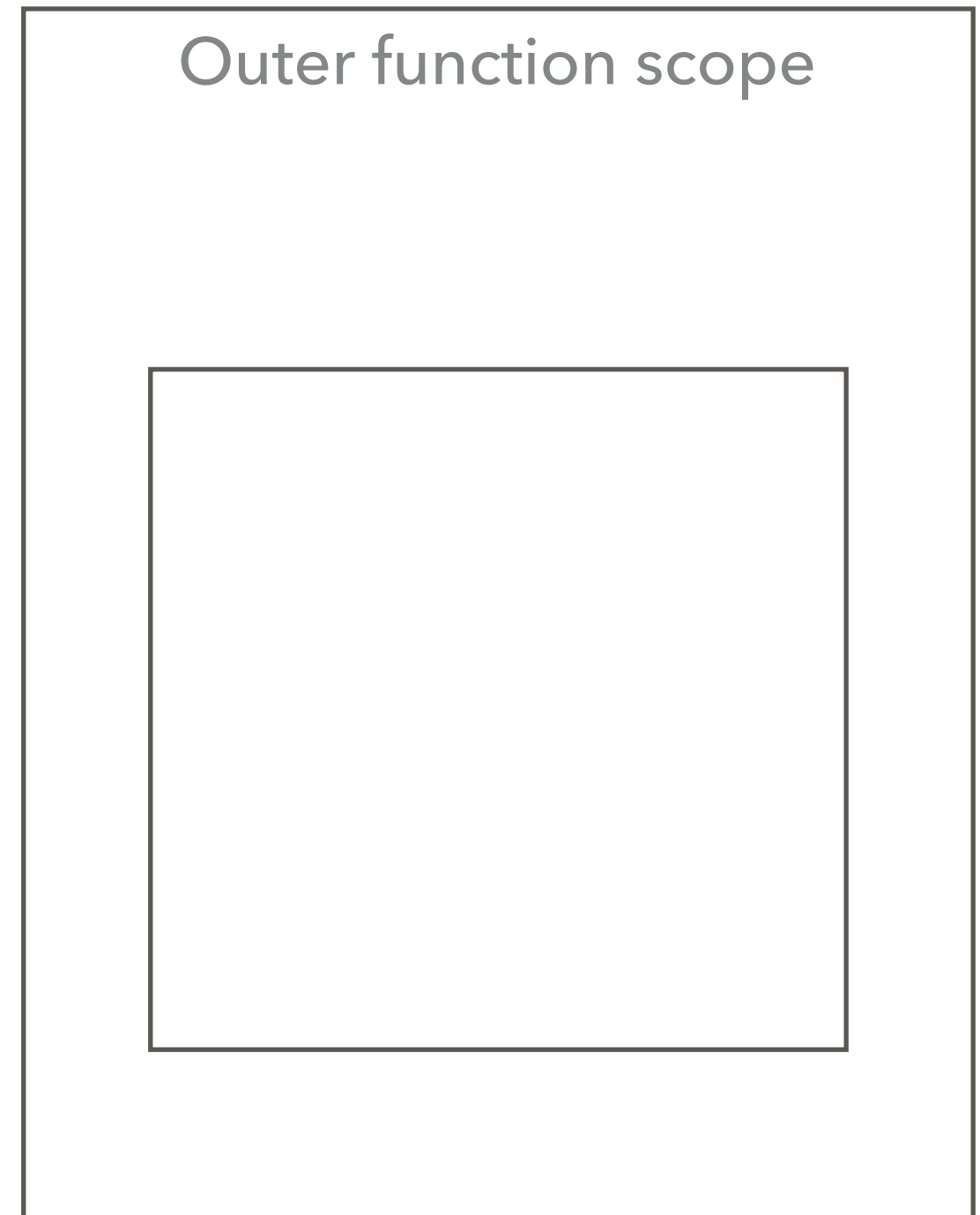
```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```

Outer function scope

Can inner function access num?

CLOSURES – SCOPE REVIEW

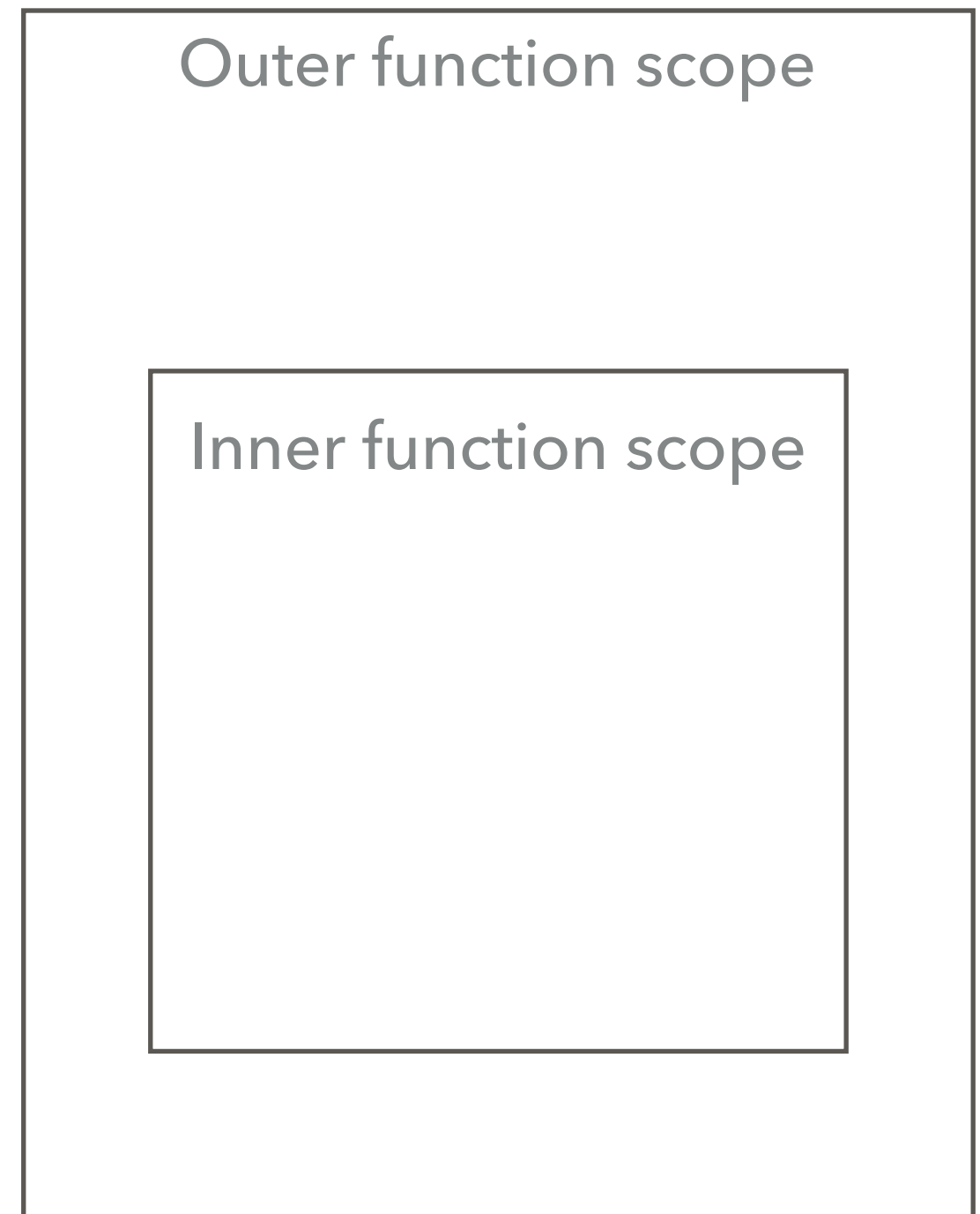
```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```



Can inner function access num?

CLOSURES – SCOPE REVIEW

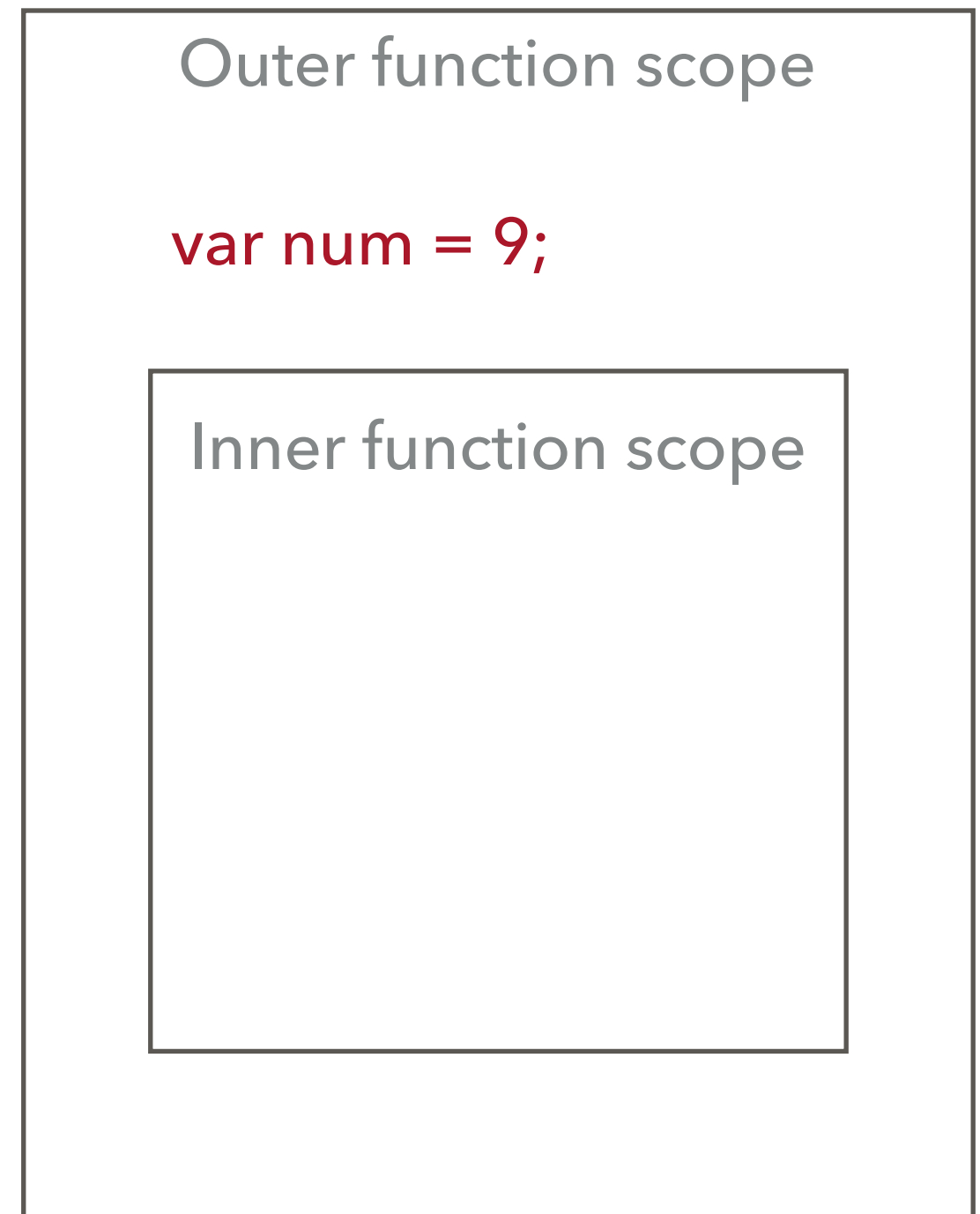
```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}  
outer();
```



Can inner function access num?

CLOSURES – SCOPE REVIEW

```
function outer() {  
  var num = 9;  
  function inner() {  
    num += 10;  
  }  
  inner();  
}  
outer();
```



Can inner function access num?

CLOSURES – SCOPE REVIEW

```
function outer() {  
    var num = 9;  
    function inner() {  
        num += 10;  
    }  
    inner();  
}
```

outer();

Outer function scope

var num = 9;

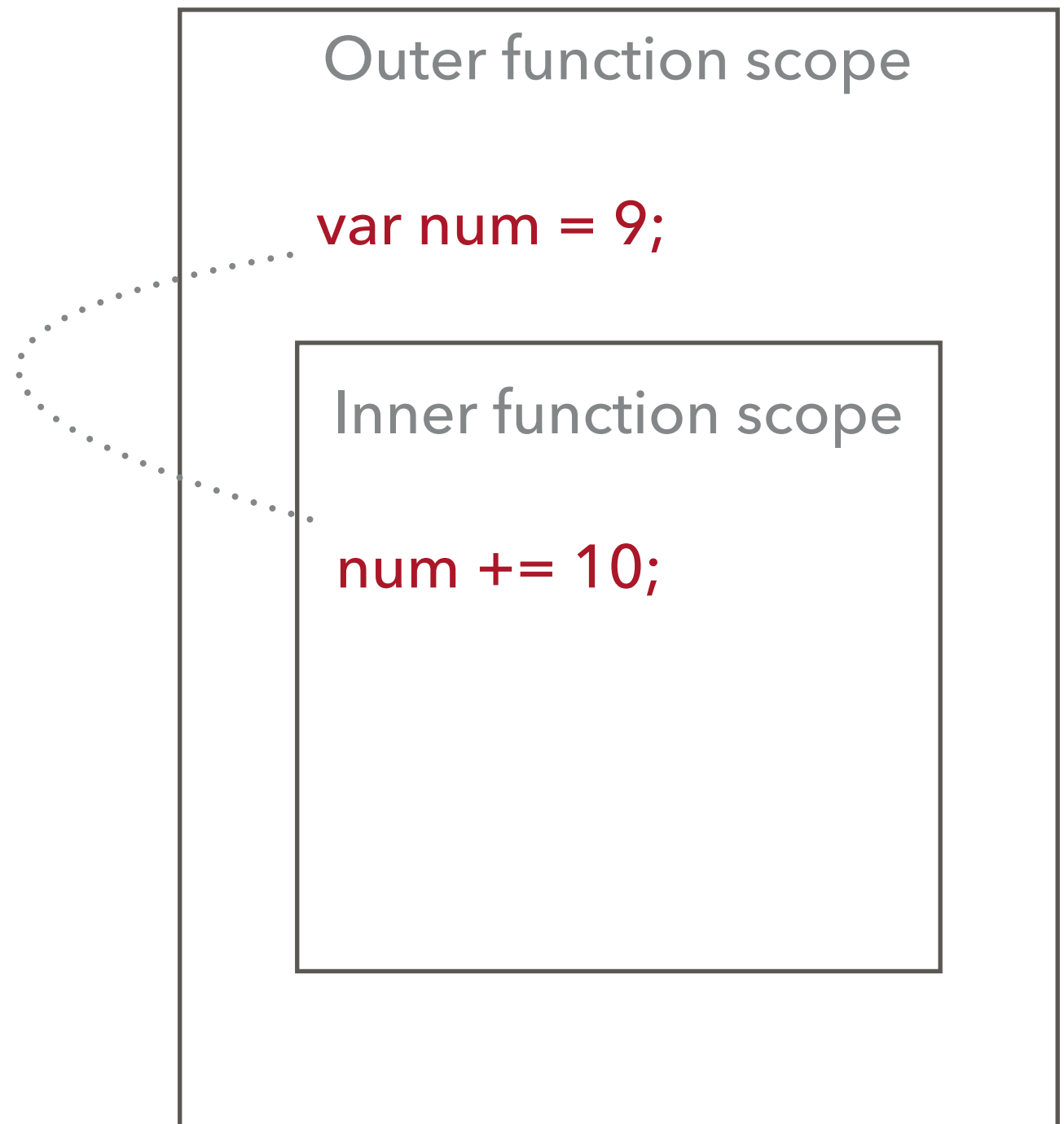
Inner function scope

num += 10;

Can inner function access num?

CLOSURES – SCOPE REVIEW

```
function outer() {  
  var num = 9;  
  function inner() {  
    num += 10;  
  }  
  inner();  
}  
outer();
```



Can inner function access num?

CLOSURES

Inner function returned.

CLOSURES

```
function outer() {
```

```
    var num = 9;
```

Inner function returned.

```
}
```

CLOSURES

Inner function returned.

```
function outer() {  
    var num = 9;  
    function inner() {  
        num++;  
        return num;  
    }  
}
```

CLOSURES

Inner function returned.

```
function outer() {  
    var num = 9;  
    function inner() {  
        num++;  
        return num;  
    }  
}
```

CLOSURES

Inner function returned.

```
function outer() {  
    var num = 9;  
    return function inner() {  
        num++;  
        return num;  
    }  
}
```

CLOSURES

Inner function returned.

```
function outer() {  
    var num = 9;  
    return function inner() {  
        num++;  
        return num;  
    }  
}  
  
var innerFn = outer();
```

CLOSURES

Inner function returned.

```
function outer() {
```

```
    var num = 9;
```

```
    return function inner() {
```

```
        num++;
```

```
        return num;
```

```
    }
```

```
}
```

```
var innerFn = outer();
```

CLOSURES

Inner function returned.

```
function outer() {
```

```
  var num = 9;
```

```
  return function inner() {
```

```
    num++;
```

```
    return num;
```

```
  }
```

```
}
```

```
var innerFn = outer();
```



CLOSURES

Inner function returned.

```
function outer() {
```

```
  var num = 9;
```

```
  return function inner() {
```

```
    num++;
```

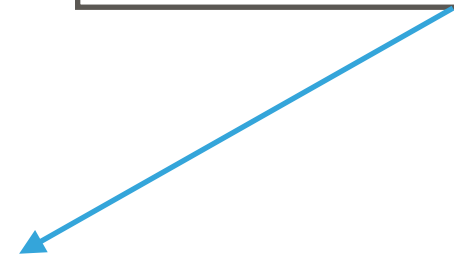
```
    return num;
```

```
  }
```

```
}
```

```
var innerFn = outer();
```

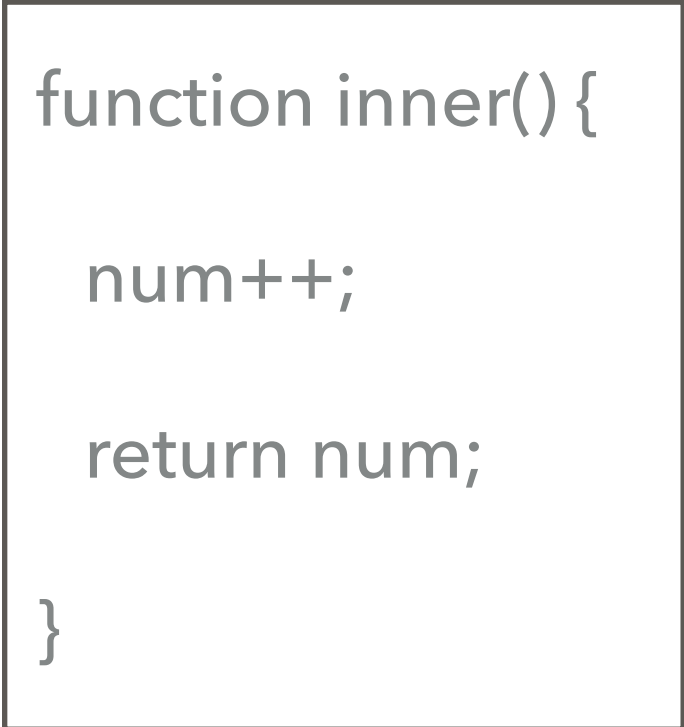
```
innerFn();
```



CLOSURES

Inner function returned.

```
function outer() {  
  var num = 9;  
  return function inner() {  
    num++;  
    return num;  
  }  
}  
  
var innerFn = outer();  
  
innerFn(); // 10
```



The diagram illustrates the concept of a closure. It shows a function `outer()` that defines a local variable `num` with the value 9. Inside `outer()`, there is an inner function `inner()`. The `return` statement of `outer()` returns the `inner` function. A blue arrow points from the closing curly brace of the `inner` function to the assignment `var innerFn = outer();`, indicating that the returned function is assigned to `innerFn`. Below this, the code `innerFn();` is shown, followed by a comment `// 10`, indicating that the inner function successfully accessed and modified the `num` variable from the `outer` function's scope.



CLOSURES – SNAPSHOT

CLOSURES – SNAPSHOT

```
function outer() {  
    var num = 0;  
    function inner() {  
        return ++num;  
    }  
    return inner;  
}
```

CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}
```



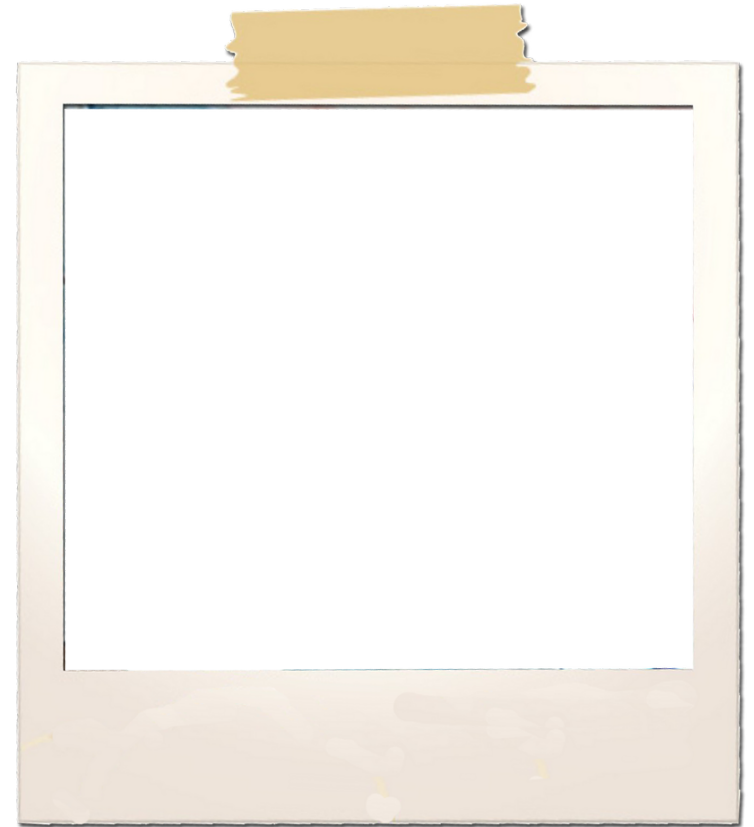
CLOSURES – SNAPSHOT

```
function outer() {  
    var num = 0;  
    function inner() {  
        return ++num;  
    }  
    return inner;  
}  
var one = outer();
```



CLOSURES – SNAPSHOT

```
function outer() {  
    var num = 0;  
    function inner() {  
        return ++num;  
    }  
    return inner;  
}  
var one = outer();
```

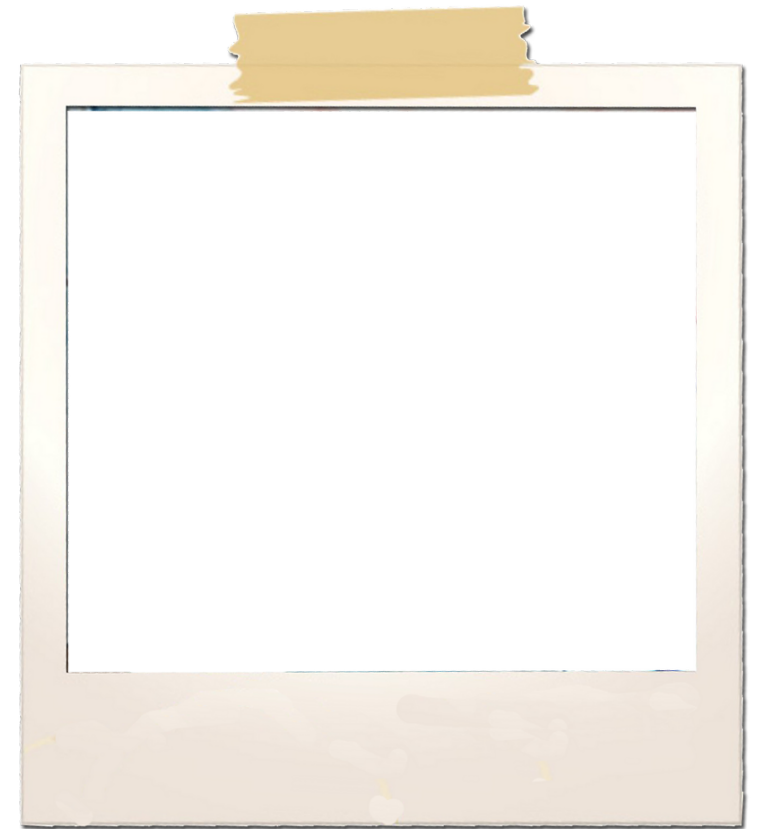


CLOSURES – SNAPSHOT

```
function outer() {  
    var num = 0;  
    function inner() {  
        return ++num;  
    }  
    return inner;  
}  
  
var one = outer();
```



snapshot of lexical
environment

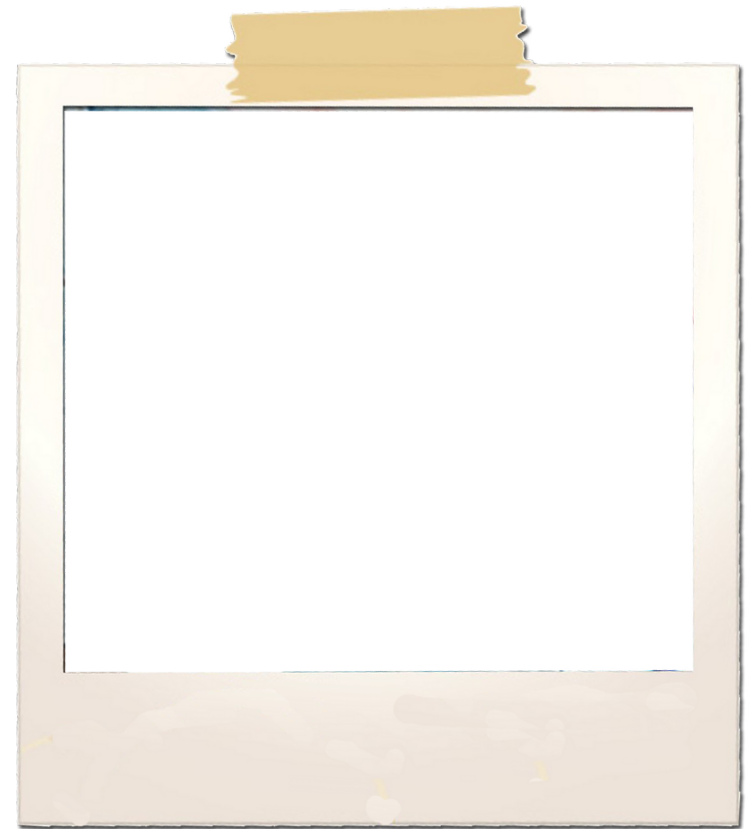


CLOSURES – SNAPSHOT

```
function outer() {  
    var num = 0;  
    function inner() {  
        return ++num;  
    }  
    return inner;  
}  
  
var one = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();  
two();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();  
two();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();  
two();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}  
  
var one = outer();  
one();  
var two = outer();  
two();  
one();
```



snapshot of lexical
environment



CLOSURES – SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}
```

```
var one = outer();  
one();  
var two = outer();  
two();  
one();
```



snapshot of lexical
environment



CLOSURES - SNAPSHOT

```
function outer() {  
  var num = 0;  
  function inner() {  
    return ++num;  
  }  
  return inner;  
}
```

```
var one = outer();  
one();  
var two = outer();  
two();  
one();
```



snapshot of lexical
environment



[HTTPS://REPL.IT/K8SB/4](https://repl.it/k8sb/4)

PRACTICE:

[HTTPS://REPL.IT/K9EP/43](https://repl.it/k9EP/43)

JAVASCRIPT

CONSTRUCTOR FUNCTIONS

[HTTPS://REPL.IT/K9EZ/21](https://repl.it/k9EZ/21)

