

Javascript Basic

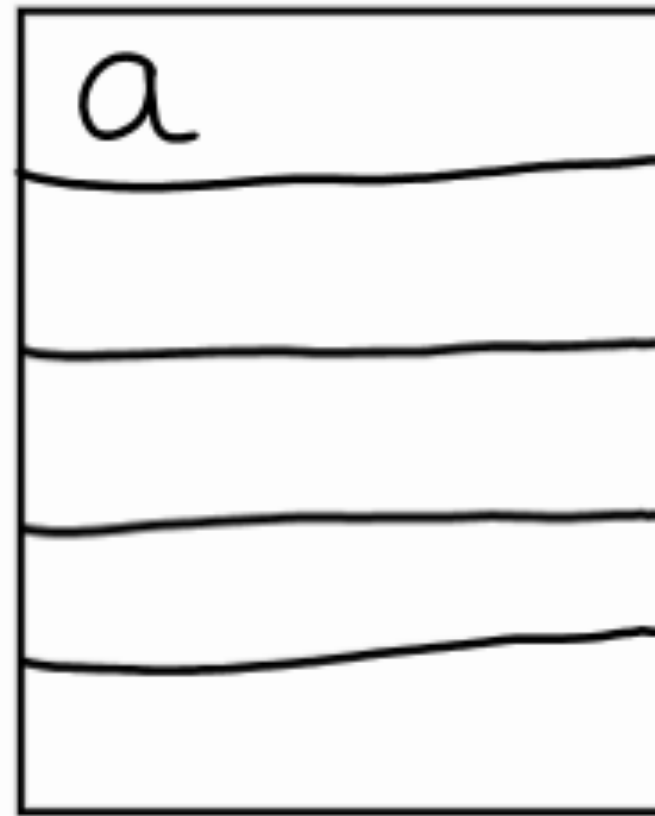
Internal Working of Javascript

Agenda

- JavaScript Interpretation and Memory Model
- Compilation and Variable Hoisting
- Function Execution
- Scope Chains and Closure
- JavaScript Runtime and Event loop
- Async Examples
- Single Threaded parallel execution

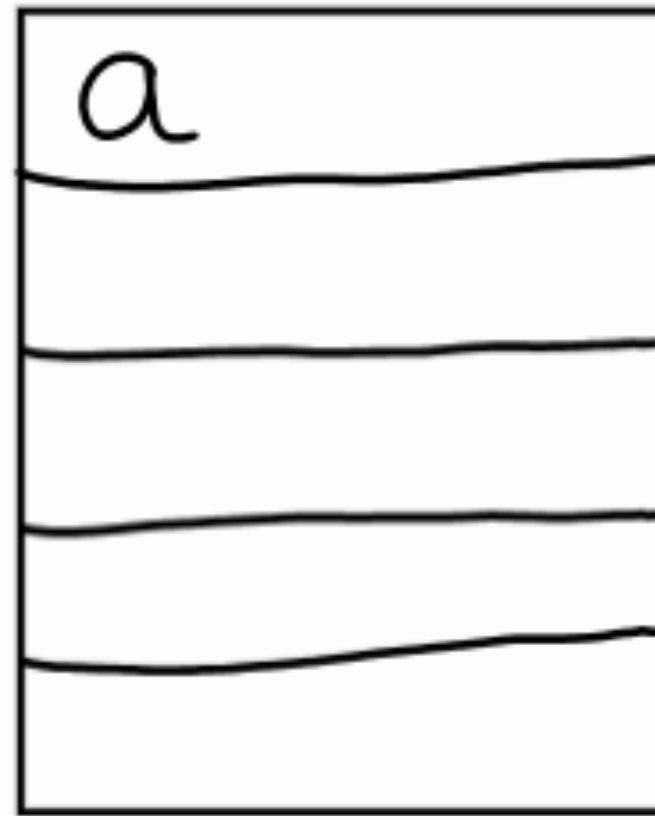
```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)



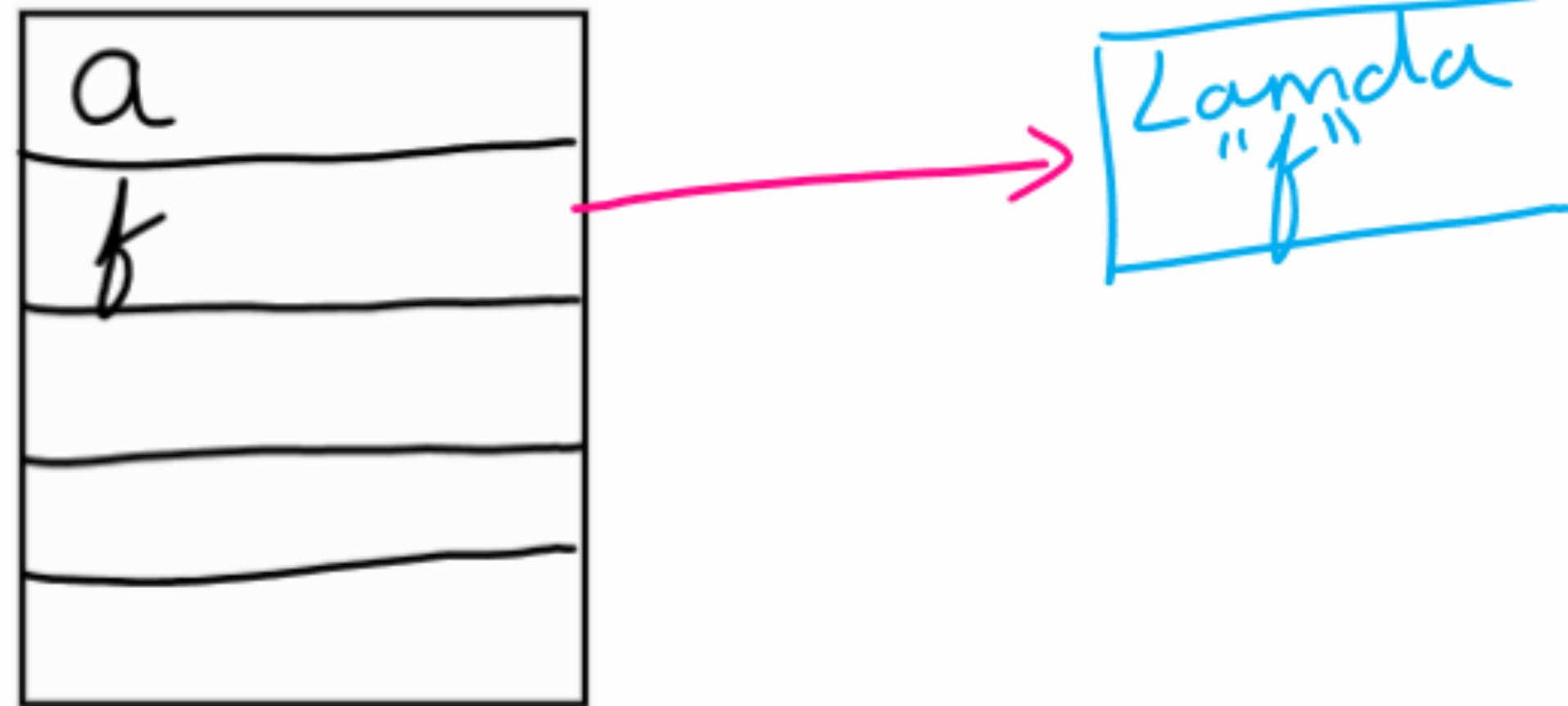
```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)



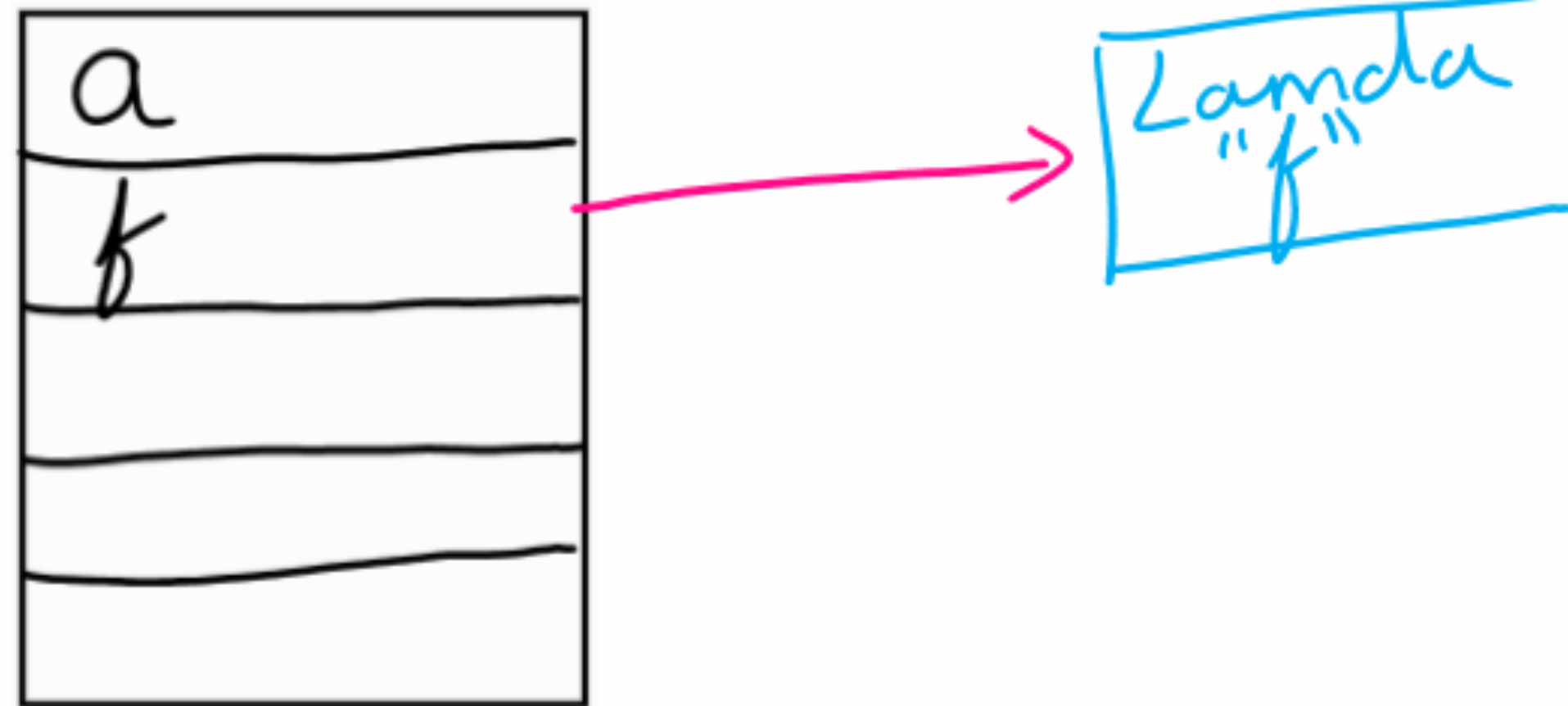
```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)



```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)



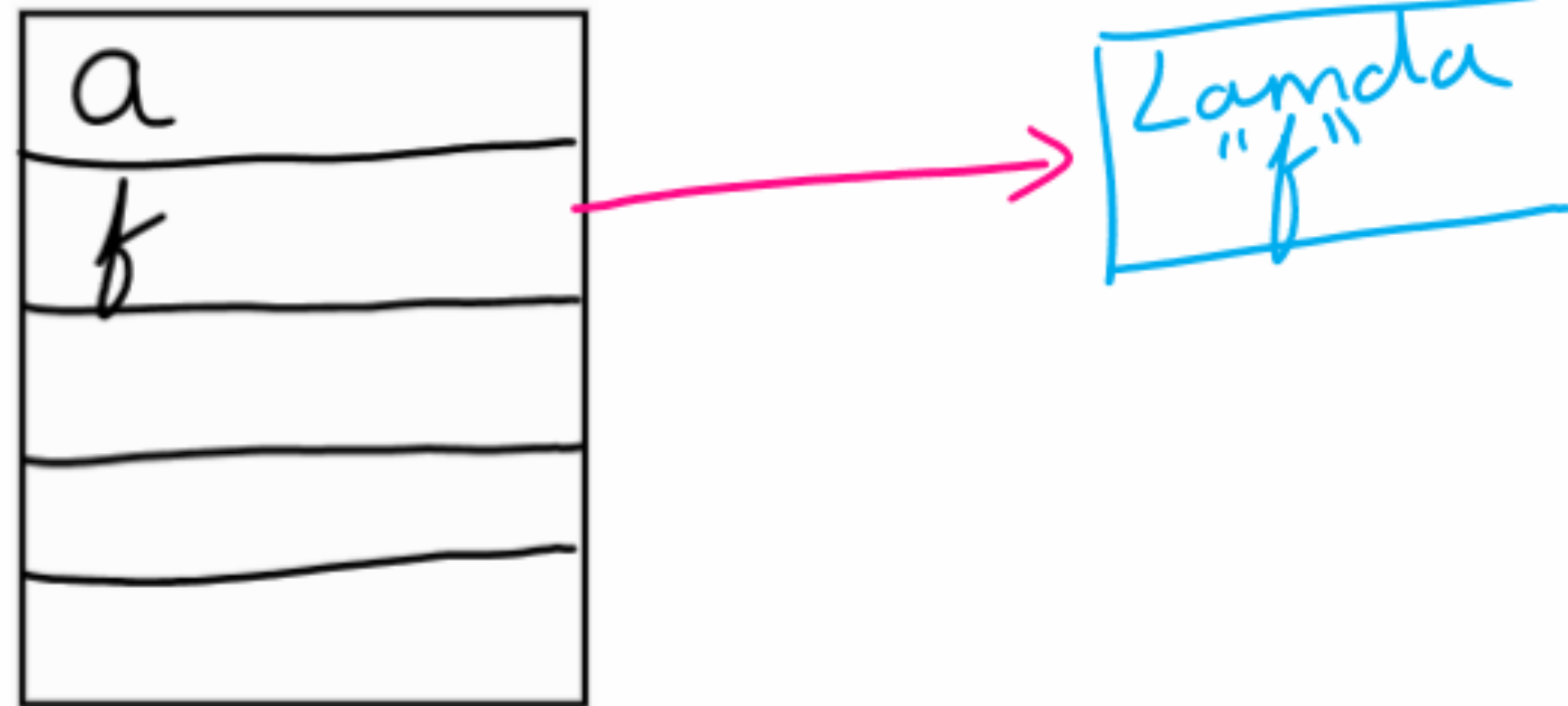
```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }

```

▶ 20 f(1); //18

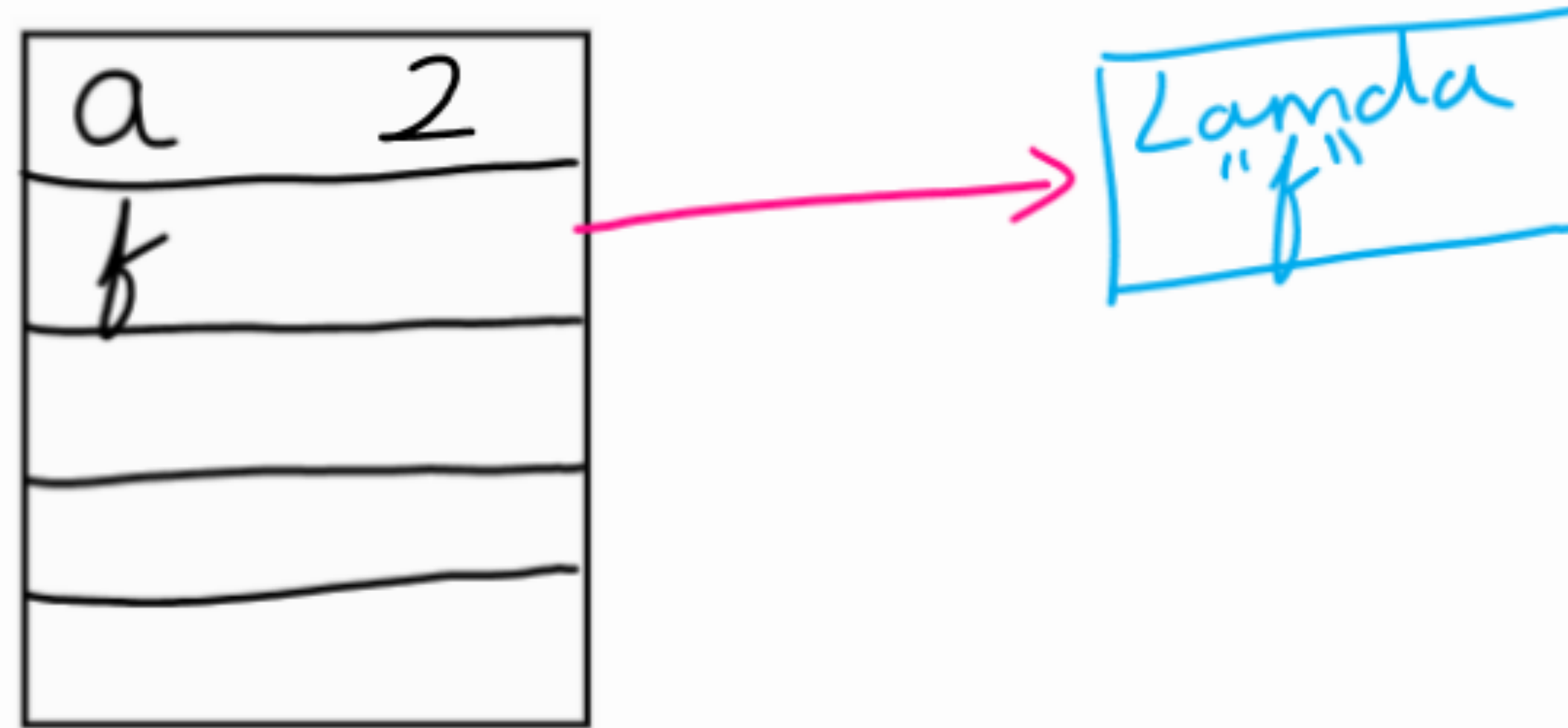
Global Scope (Window)



Compilation of main block is done!

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

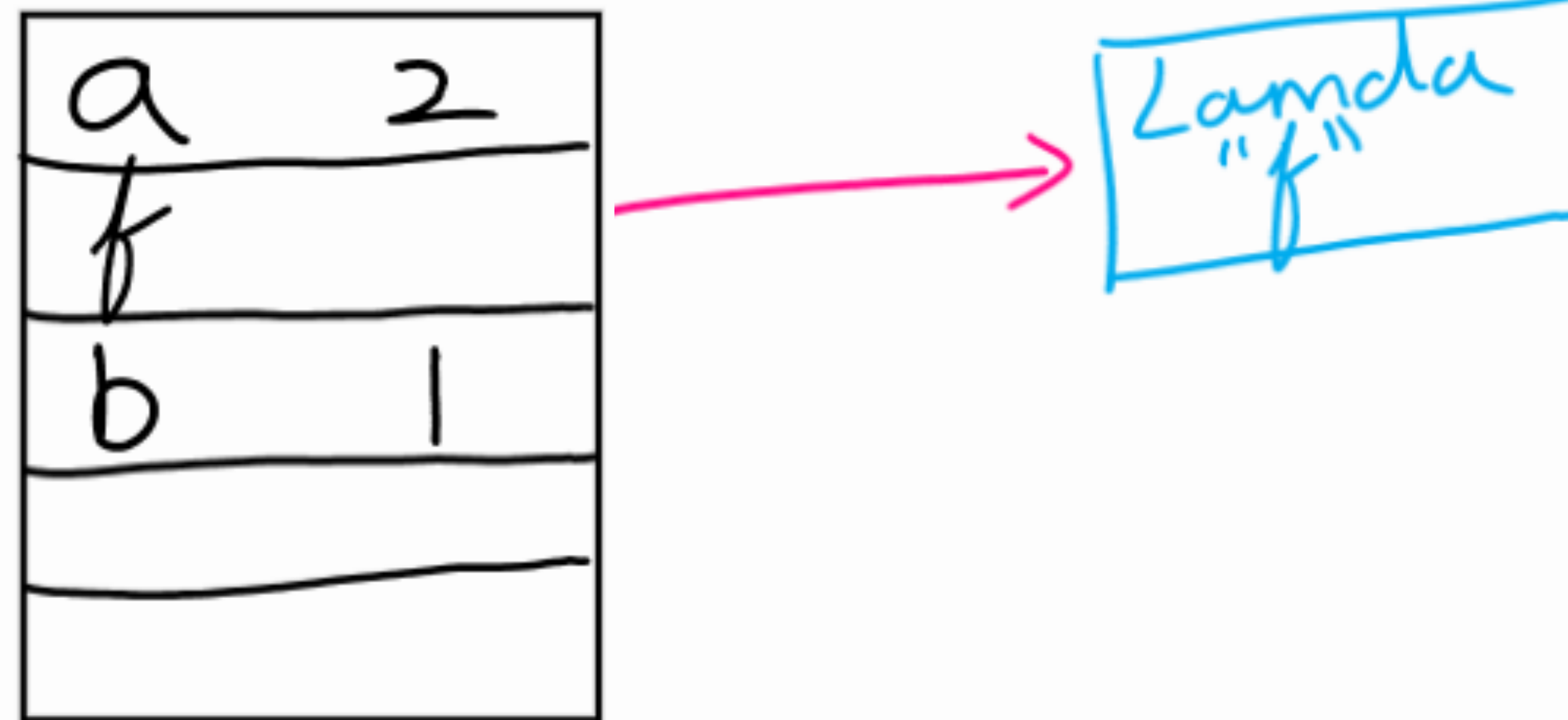



```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18

```

Global Scope (Window)

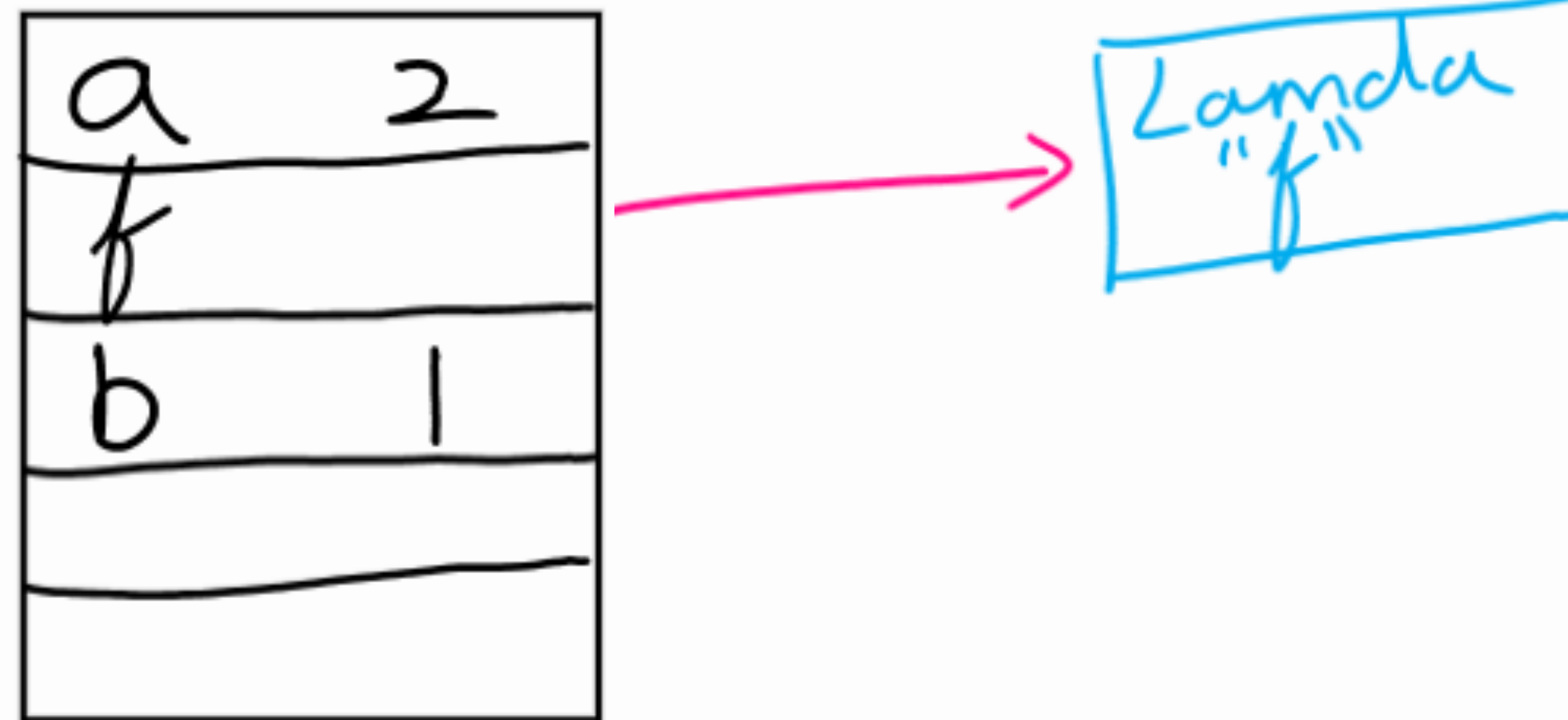


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18

```

Global Scope (Window)



Local execution Scope f()



```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z


```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z
d

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z
d

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z
d
g

Lambda
"g"


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z
d
g

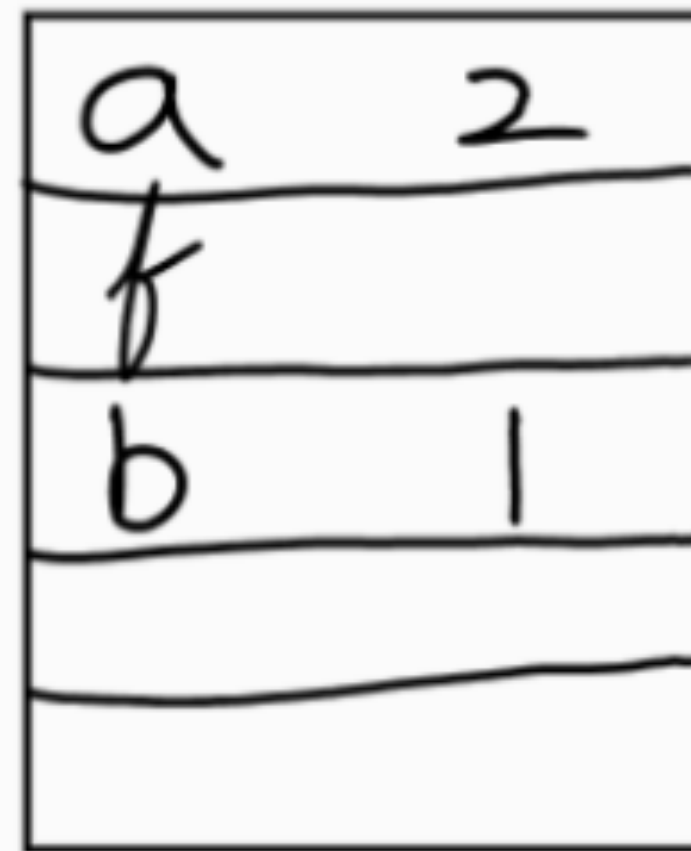
Lambda
"g"

```

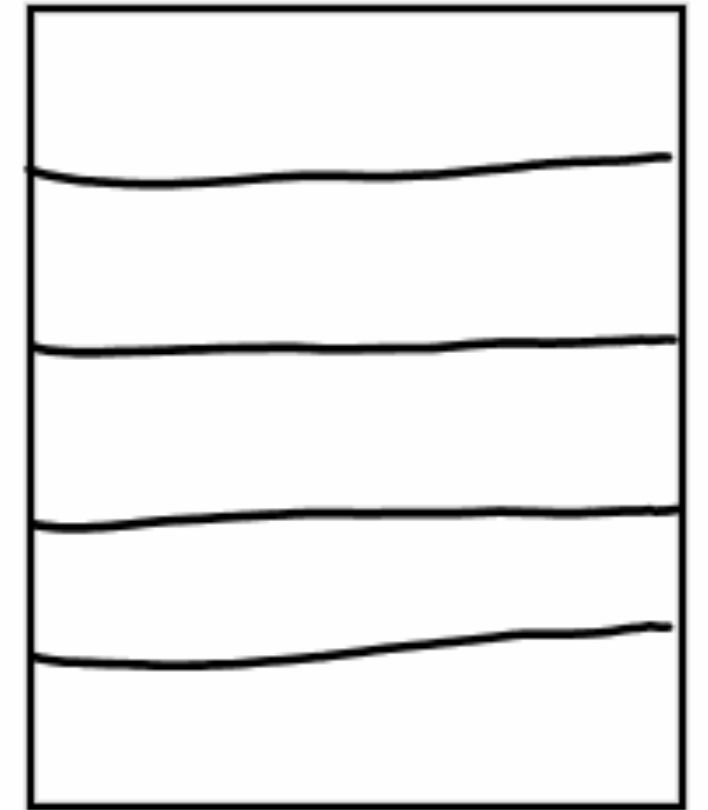
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;
18 }
19
20 f(1); //18

```

Global Scope (Window)

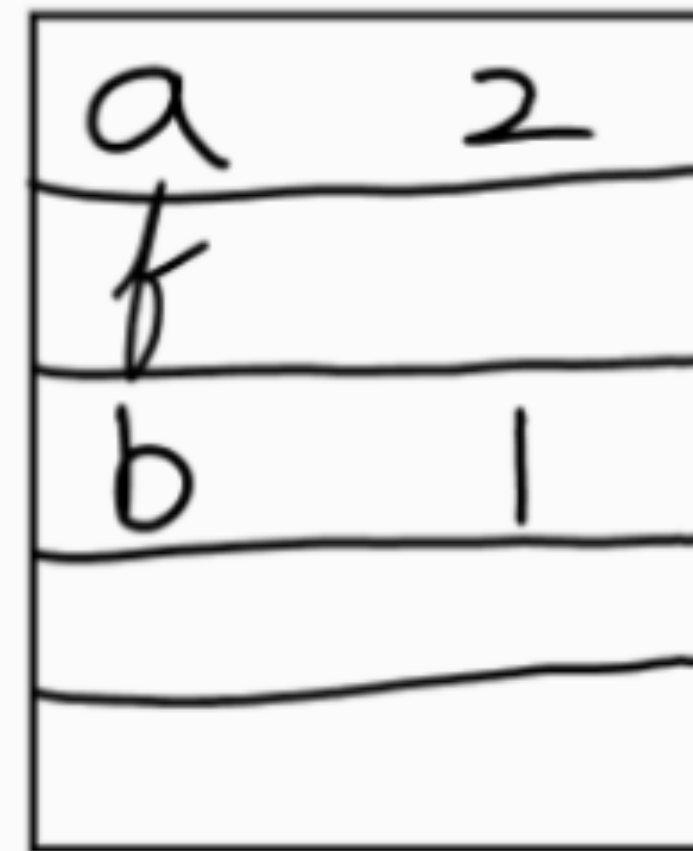


Local execution Scope f()



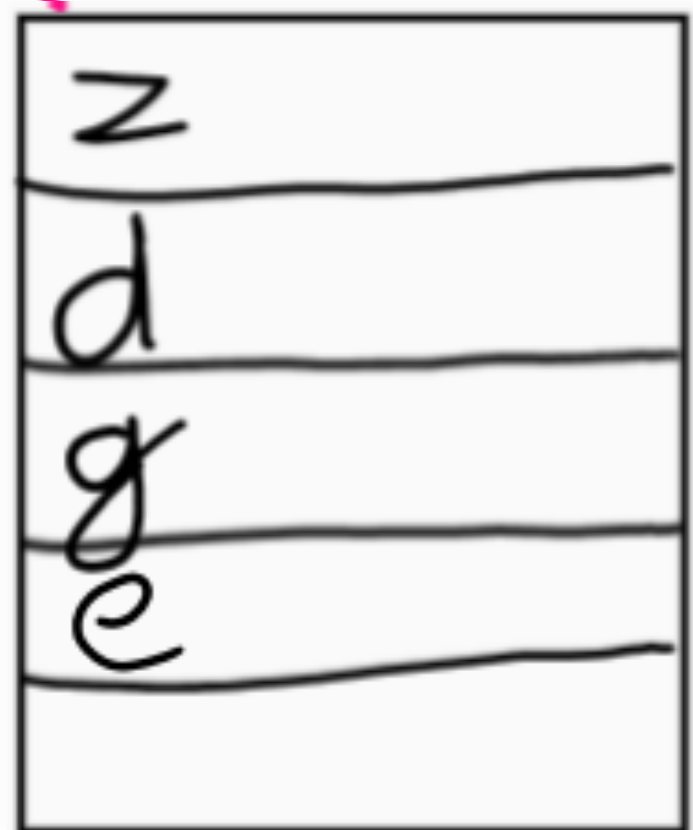
```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e; Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)



Lambda
"f"

Local execution Scope f()



Lambda
"g"

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e; Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1

Lambda
"f"

Local execution Scope f()

z	1
d	
g	
e	

Lambda
"g"


```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	3

Lambda
"f"

Local execution Scope f()

z	1
d	
g	
e	

Lambda
"g"

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	
g	
e	

Lambda
"g"

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e; Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	

Lambda
"g"

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e; Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()


```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e; Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e

Variable Shadowing


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e

Variable Shadowing

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e

Variable Shadowing

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e

Variable Shadowing


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Variable Shadowing

```
1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18
```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6 18
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Variable Shadowing


```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6 18
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Variable Shadowing

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	1 3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6 18
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Variable Shadowing

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g();
17     var e;    Variable Hoisting
18 }
19
20 f(1); //18

```

Global Scope (Window)

a	2
f	
b	3
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	18
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Garbage Collected

Variable Shadowing

Lets make a slight change in the program...


```
▶ 1  var a = 2;
   2  b = 1;
   3
   4  function f(z) {
   5      b = 3;
   6      c = 4;
   7      var d = 6;
   8      e = 1;
   9
  10      function g() {
  11          var e = 0;
  12          d = 3*d;
  13          return d;
  14      }
  15
  16      return g;  //We are returning function itself
  17      var e;
  18  }
  19
  20  var myG = f(1); //We have global reference to the function object
  21  myG() //18
```

```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g;
17     var e;
18 }
19
20 var myG = f(1);
21 myG() //18

```

Global Scope (Window)

a	2
f	
b	3
myb	
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0



```

1  var a = 2;
2  b = 1;
3
4  function f(z) {
5      b = 3;
6      c = 4;
7      var d = 6;
8      e = 1;
9
10     function g() {
11         var e = 0;
12         d = 3*d;
13         return d;
14     }
15
16     return g;
17     var e;
18 }
19
20 var myG = f(1);
21 myG() //18

```

Global Scope (Window)

a	2
f	
b	3
myb	
c	4

Lambda
"f"

Local execution Scope f()

z	1
d	6
g	
e	1

Lambda
"g"

Local execution Scope g()

e	0

Garbage Collected

Closures

- Variables holds on to the function reference which gets defined in such a deep function execution chain.
- Its an implicit, permanent link between function and its scope chain.
- Prevents garbage collection
- It is used and copied as a outer environment reference at runtime

Closures

- Implicit Closures

```
1  var data = "My Data"
2  setTimeout(function(){
3    |    console.log(data); // prints My Data
4    |  },8000);
```

- Explicit Closures

```
1  function makeAdder(n){
2    |    var inc=n;
3    |    var sum=0;
4    |    return function add(){
5    |    |    sum = sum + inc;
6    |    |    return sum;
7    |    };
8  |  }
9
10 var adder3 = makeAdder(3);
```