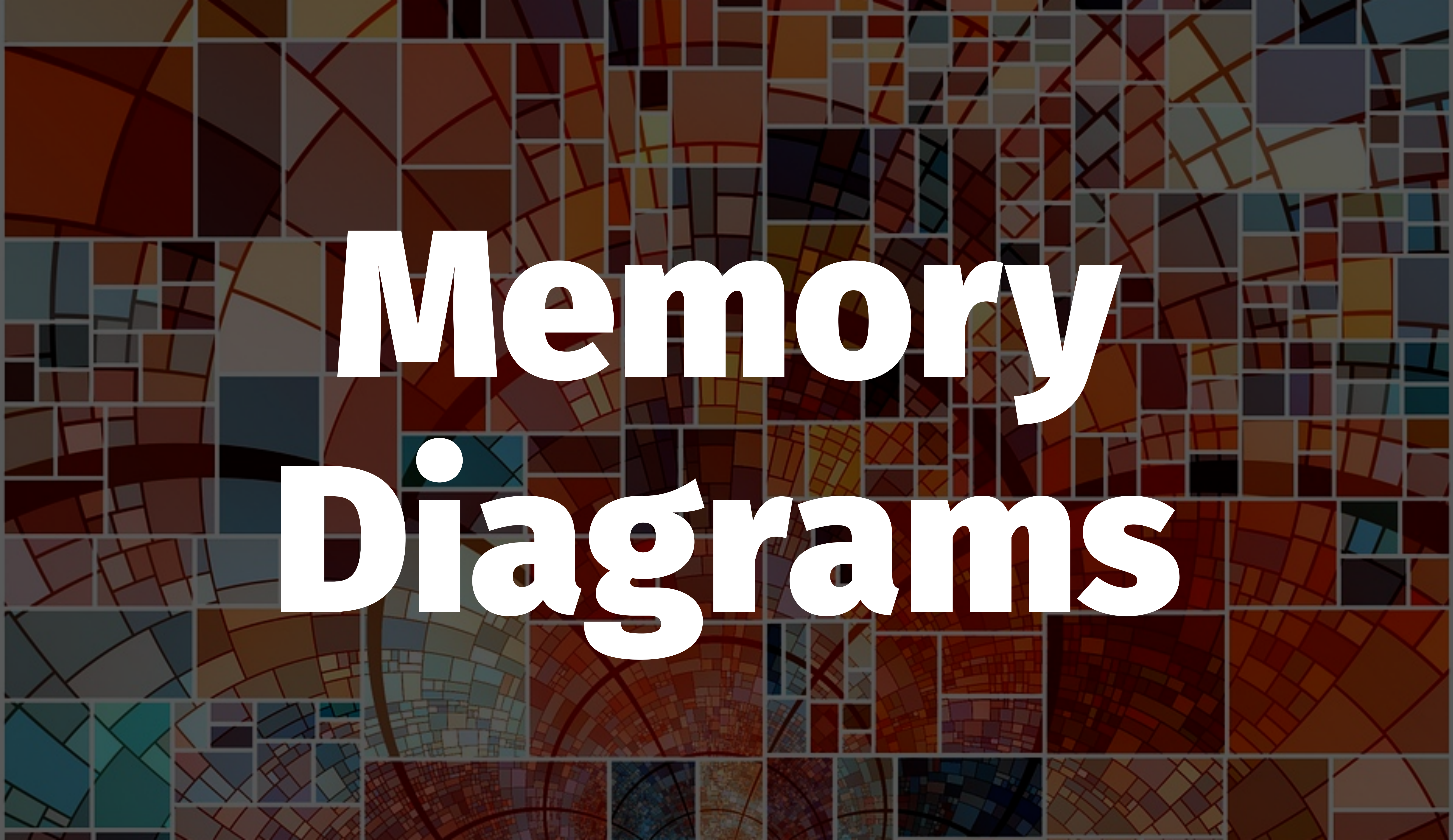
Memory Diagrams



Objectives

→ Represent "static" objects, arrays, functions and variables in diagrams.

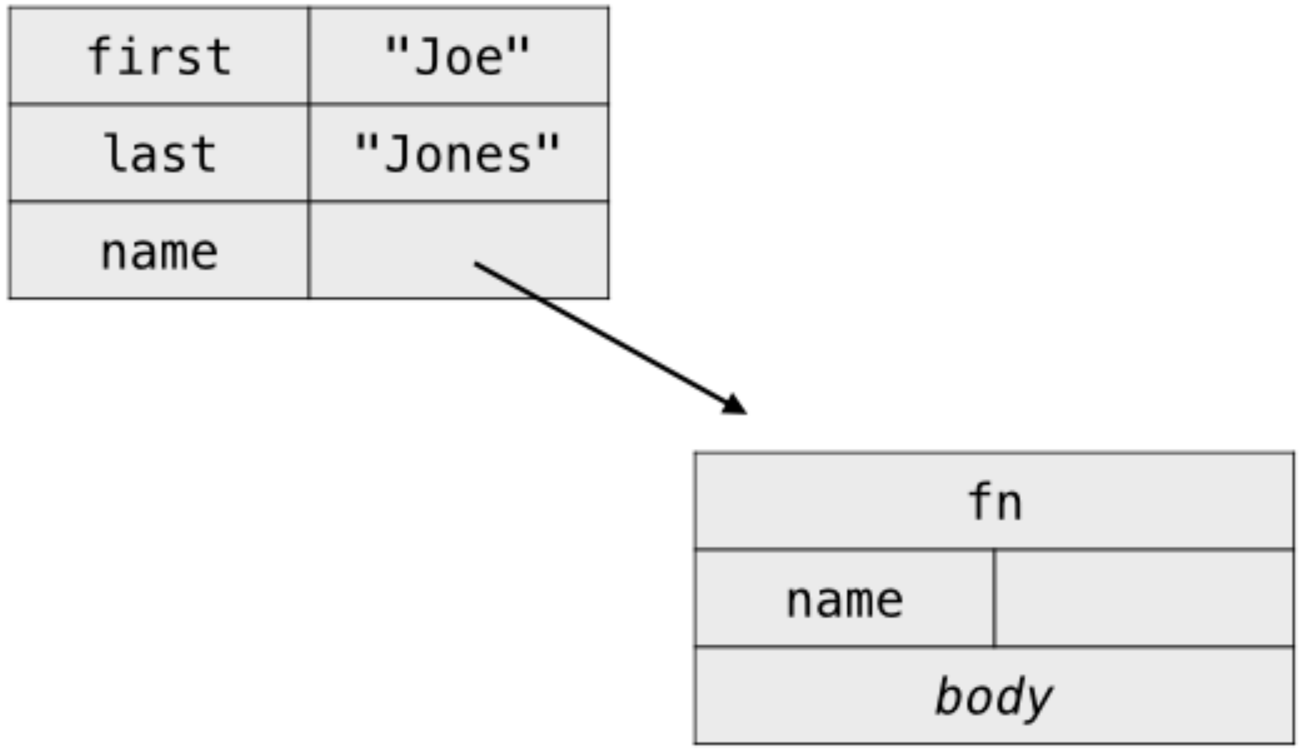
→ Draw a diagram of the result of calling a function.

→ Explain the difference between mutable values and immutable values.

How

Object

```
{
 first:'Joe',
 last: 'Jones',
 name: function(){
  return [
   this.first,
   this.last
  ].join(' ');
 }
}
```

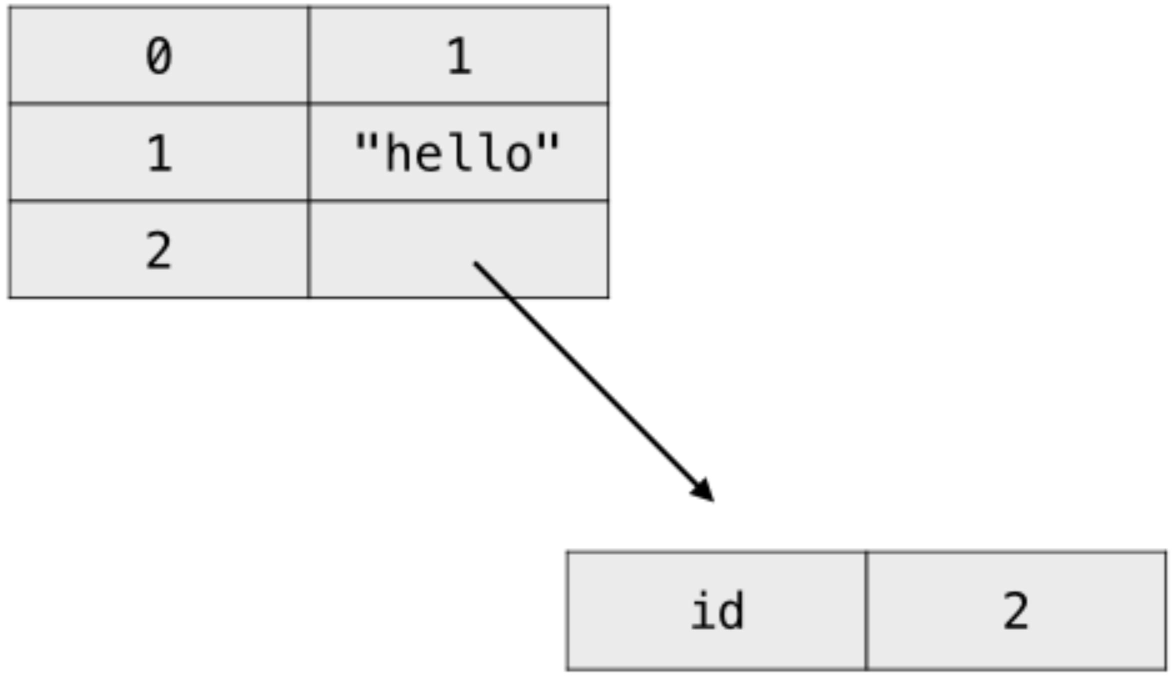


first	"Joe"
last	"Jones"
name	

fn	
name	
body	

Array

```
[
 1,
 "hello",
 {id: 2},
]
```

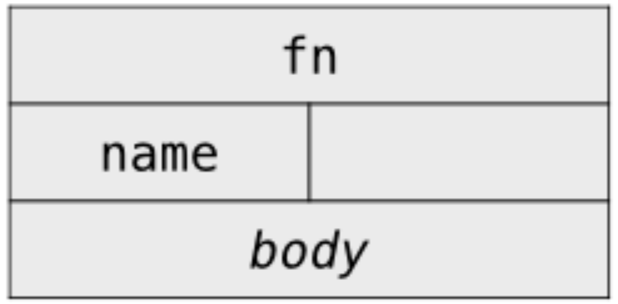


0	1
1	"hello"
2	

id	2

Function

```
function(){
  return [
   this.first,
   this.last
  ].join(' ');
}
```

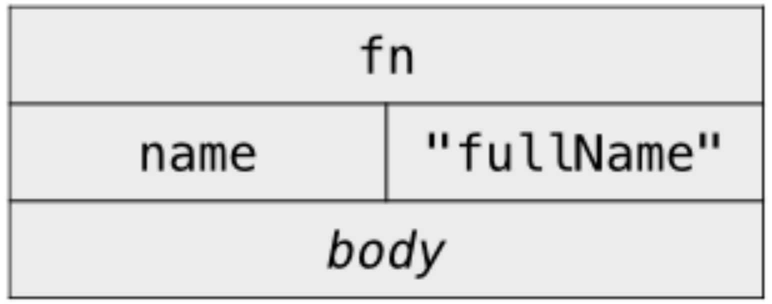


fn

name

body

```
function fullName(){
  return [
   this.first,
   this.last
  ].join(' ');
}
```



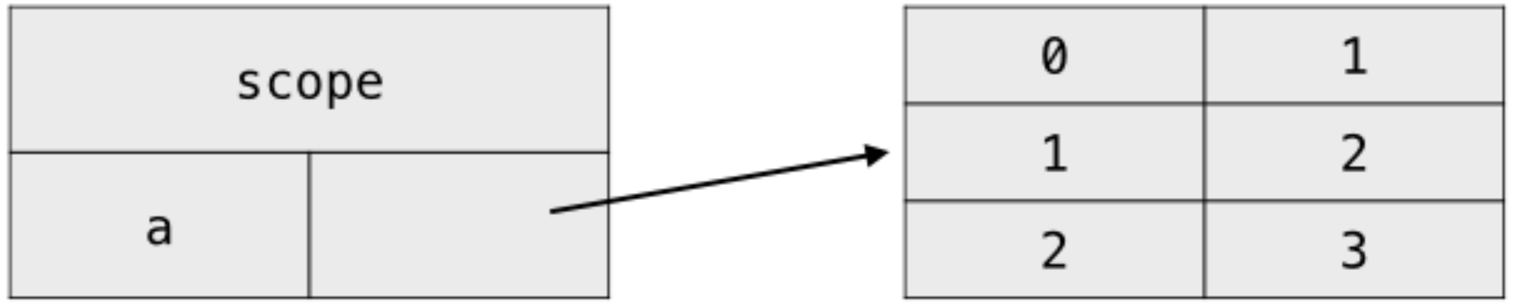
fn

name "fullName"

body

Scope (variables)

```
var a = [1,2,3]
```

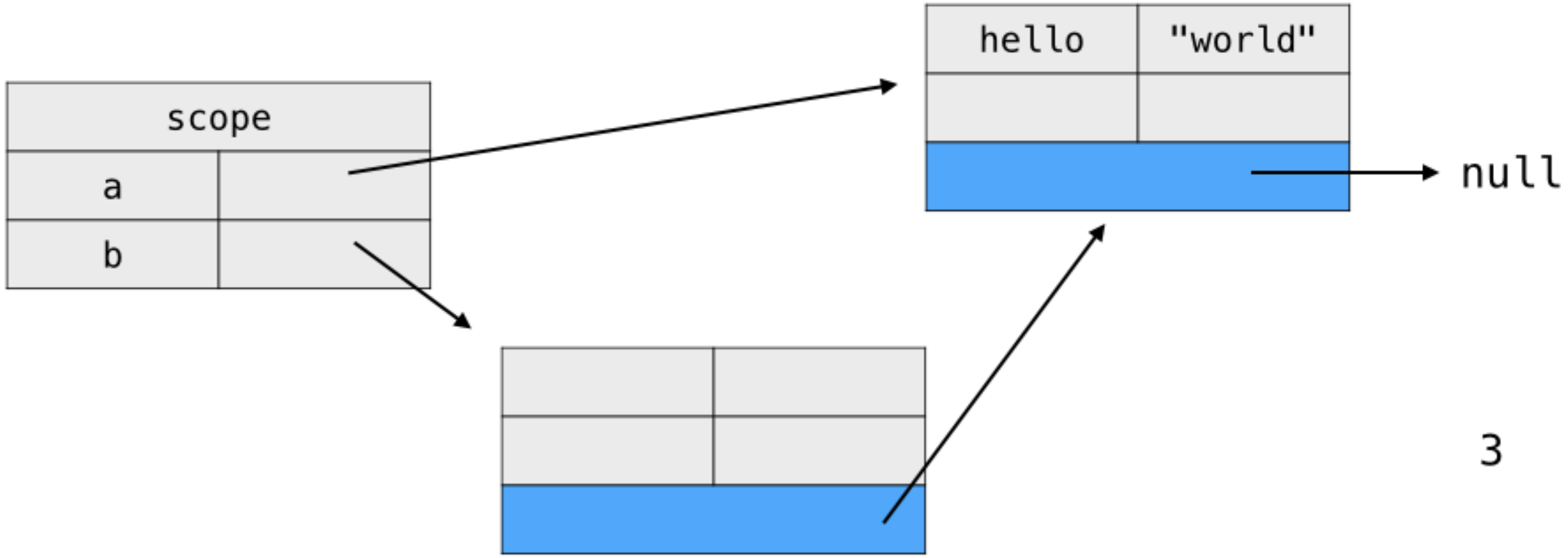


scope	
a	

0	1
1	2
2	3

Prototypes

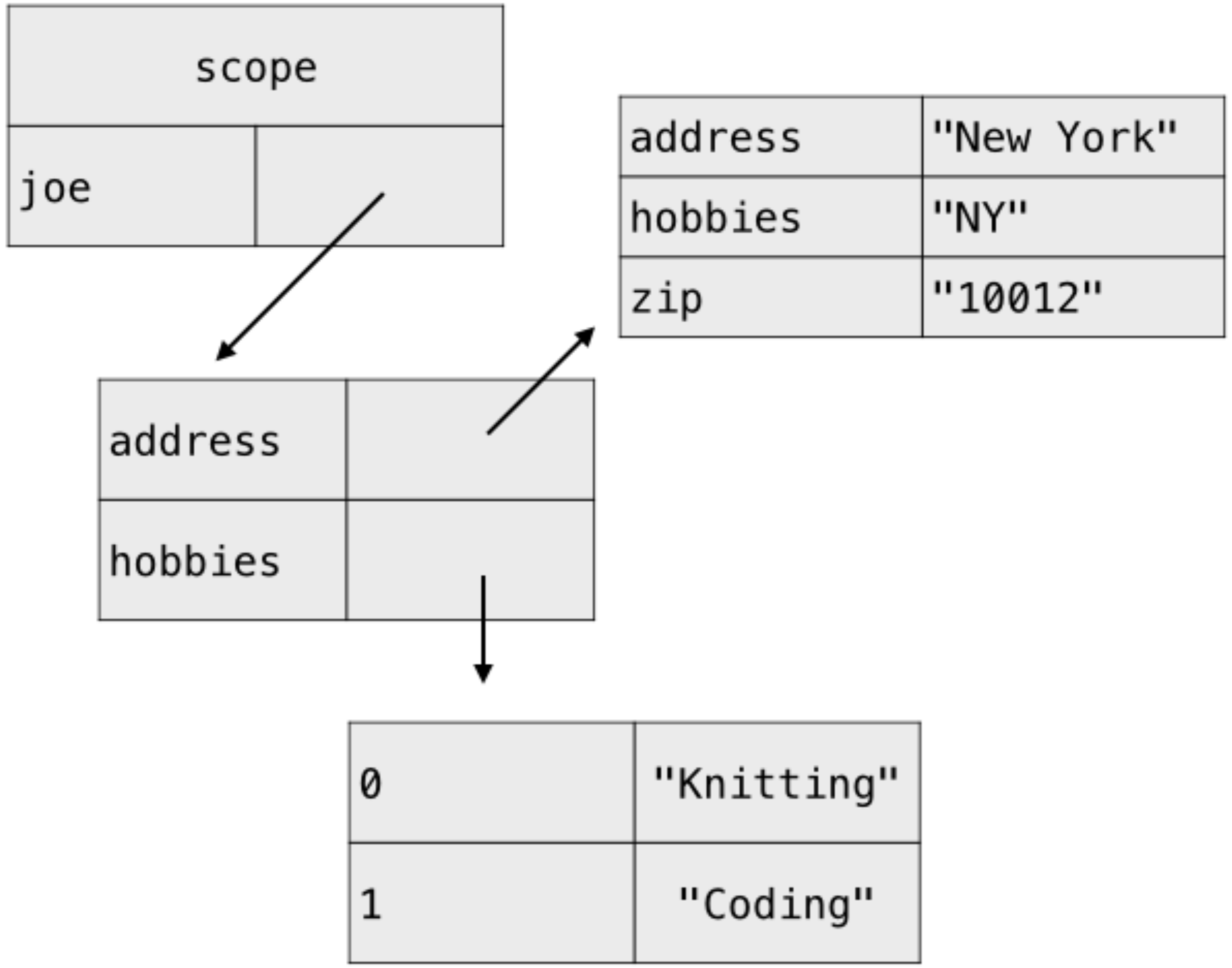
```
var a = {hello: "world"}

var b = Object.create(a)
```



```
var joe = {
    address: {
        city: "New York",
        state: "NY",
        zip: "10012"
    },
    hobbies: [
        "knitting",
        "coding"
    ]
}
```

Example

```
var joe = {
 address: {
  city: "New York",
  state: "NY",
  zip: "10012"
 },
 hobbies: [
  "knitting",
  "coding"
 ]
}
```



scope	
joe	

address	
hobbies	

address	"New York"
hobbies	"NY"
zip	"10012"

0	"Knitting"
1	"Coding"