```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
function outer(init) {
→ var x = init;                    ←
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```
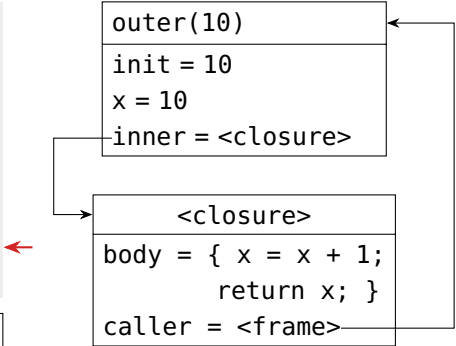
```
> var f = outer(10);
```

outer(10)
init = 10
x = 10
inner = ?

```
function outer(init) {
    var x = init;
    function inner() {
        x = x + 1;
        return x;
    }
    return inner;
}
```

```
> var f = outer(10);
```

```
outer(10)
init = 10
x = 10
inner = <closure>
```

```
<closure>
body = { x = x + 1;
         return x; }
caller = <frame>
```
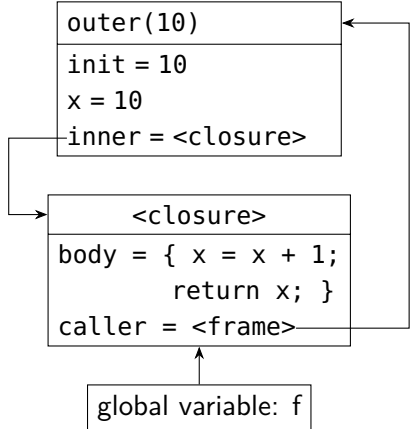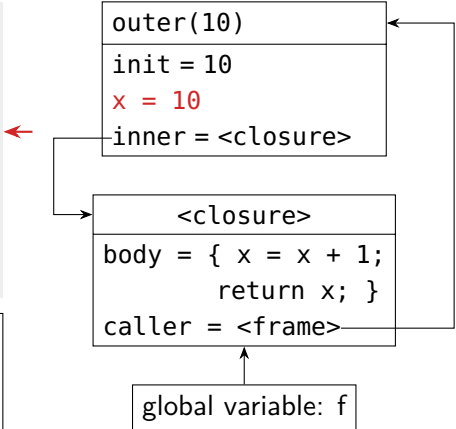
```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
> var f = outer(10);
<closure>
>
```

```
outer(10)
init = 10
x = 10
inner = <closure>
```

```
<closure>
body = { x = x + 1;
         return x; }
caller = <frame>
```

global variable: f

```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
> var f = outer(10);
<closure>
>[f(), f(), f()]
```

```
outer(10)
init = 10
x = 10
inner = <closure>
```

```
<closure>
body = { x = x + 1;
         return x; }
caller = <frame>
```

global variable: f
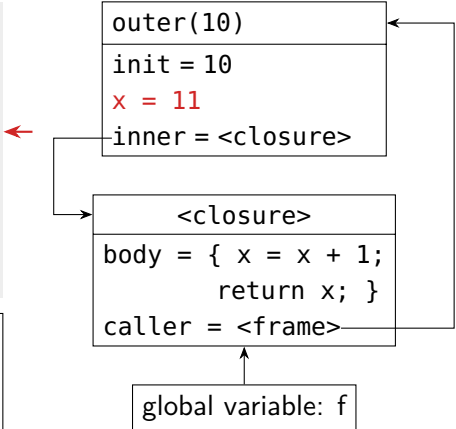
```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
> var f = outer(10);
<closure>
>[f(), f(), f()]
```

```
outer(10)
init = 10
x = 11
inner = <closure>
```

```
<closure>
body = { x = x + 1;
         return x; }
caller = <frame>
```

global variable: f
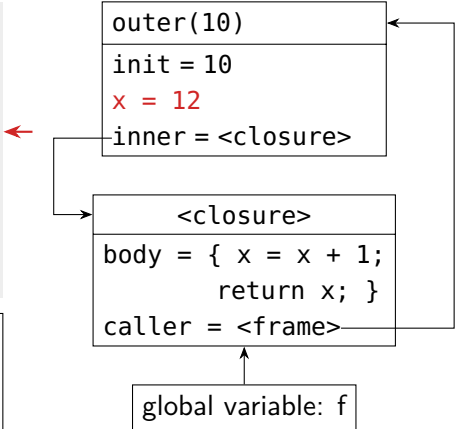
```
function outer(init) {
    var x = init;
    function inner() {
        x = x + 1;
        return x;
    }
    return inner;
}
```

```
> var f = outer(10);
<closure>
>[f(), f(), f()]
```

outer(10)
init = 10
x = 12
inner = <closure>

<closure>
body = { x = x + 1;
return x; }
caller = <frame>

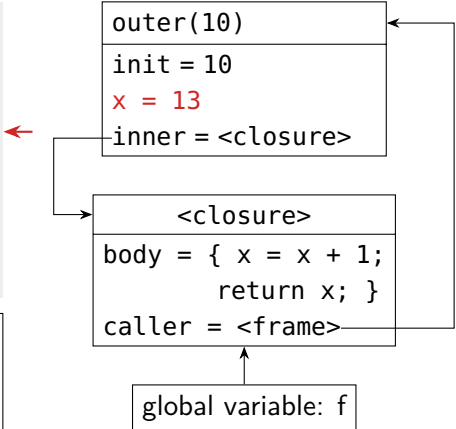global variable: f

```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
> var f = outer(10);
<closure>
>[f(), f(), f()]
```

```
outer(10)
init = 10
x = 13
inner = <closure>
```

```
<closure>
body = { x = x + 1;
         return x; }
caller = <frame>
```

global variable: f

```
function outer(init) {
  var x = init;
  function inner() {
    x = x + 1;
    return x;
  }
  return inner;
}
```

```
> var f = outer(10);
<closure>
>[f(), f(), f()]
[11, 12, 13]
```

outer(10)

init = 10
x = 13
inner = <closure>

<closure>

body = { x = x + 1;
         return x; }
caller = <frame>

global variable: f