

JSDN:

# Closures

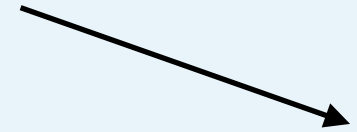
(part 2)

# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

var createIncrementer

function()



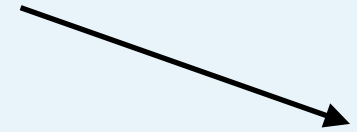
# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

a. Assignment

var createIncrementer

function()



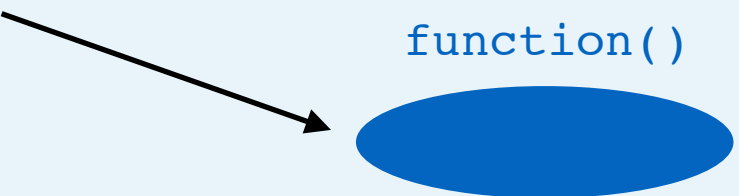
# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function

var createIncrementer

function()

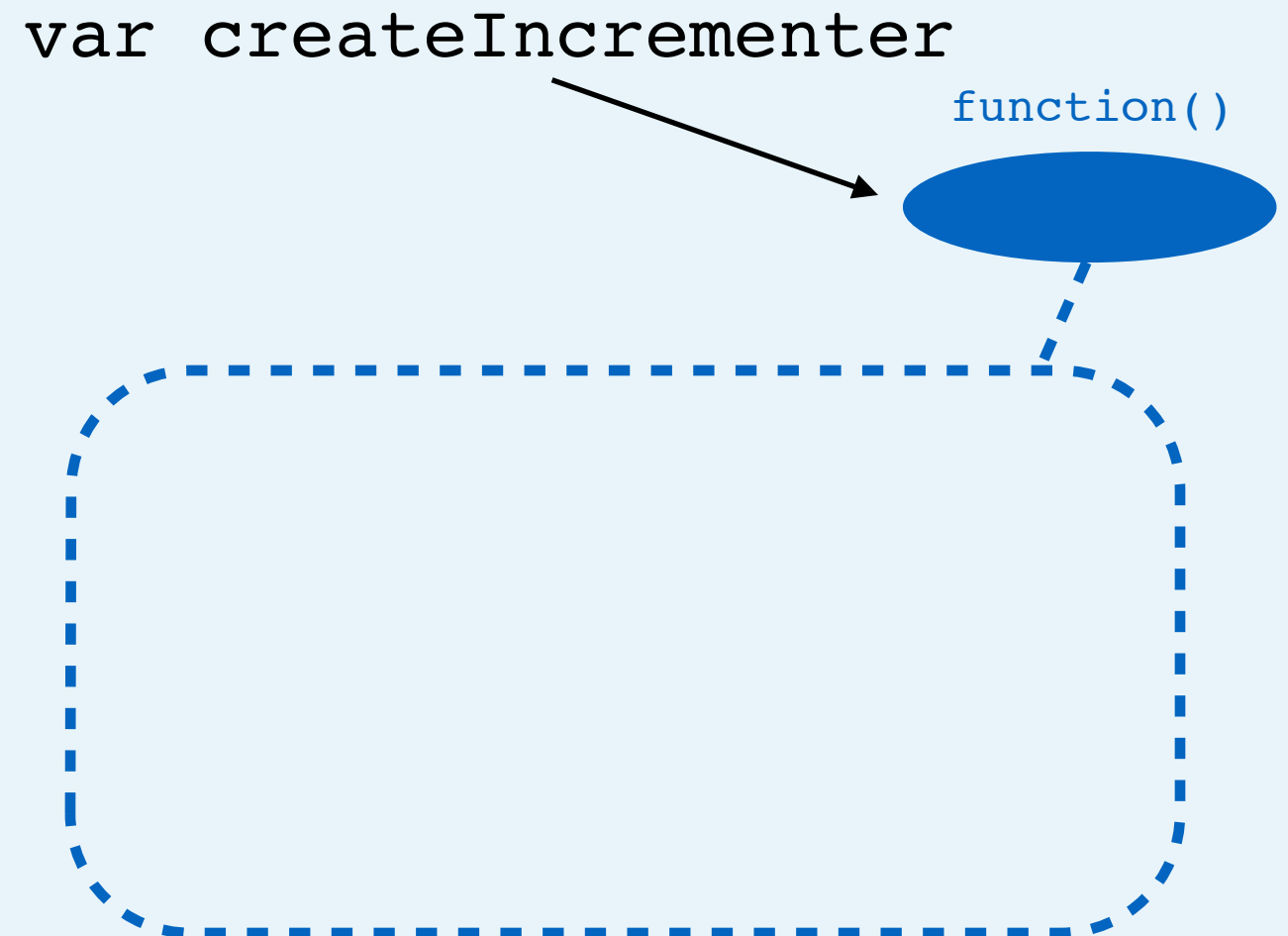


The diagram illustrates the state of the variable `createIncrementer`. It is shown as a text label with an arrow pointing to a blue oval. The label `function()` is positioned above the oval, indicating that the variable holds a reference to a function object.

# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

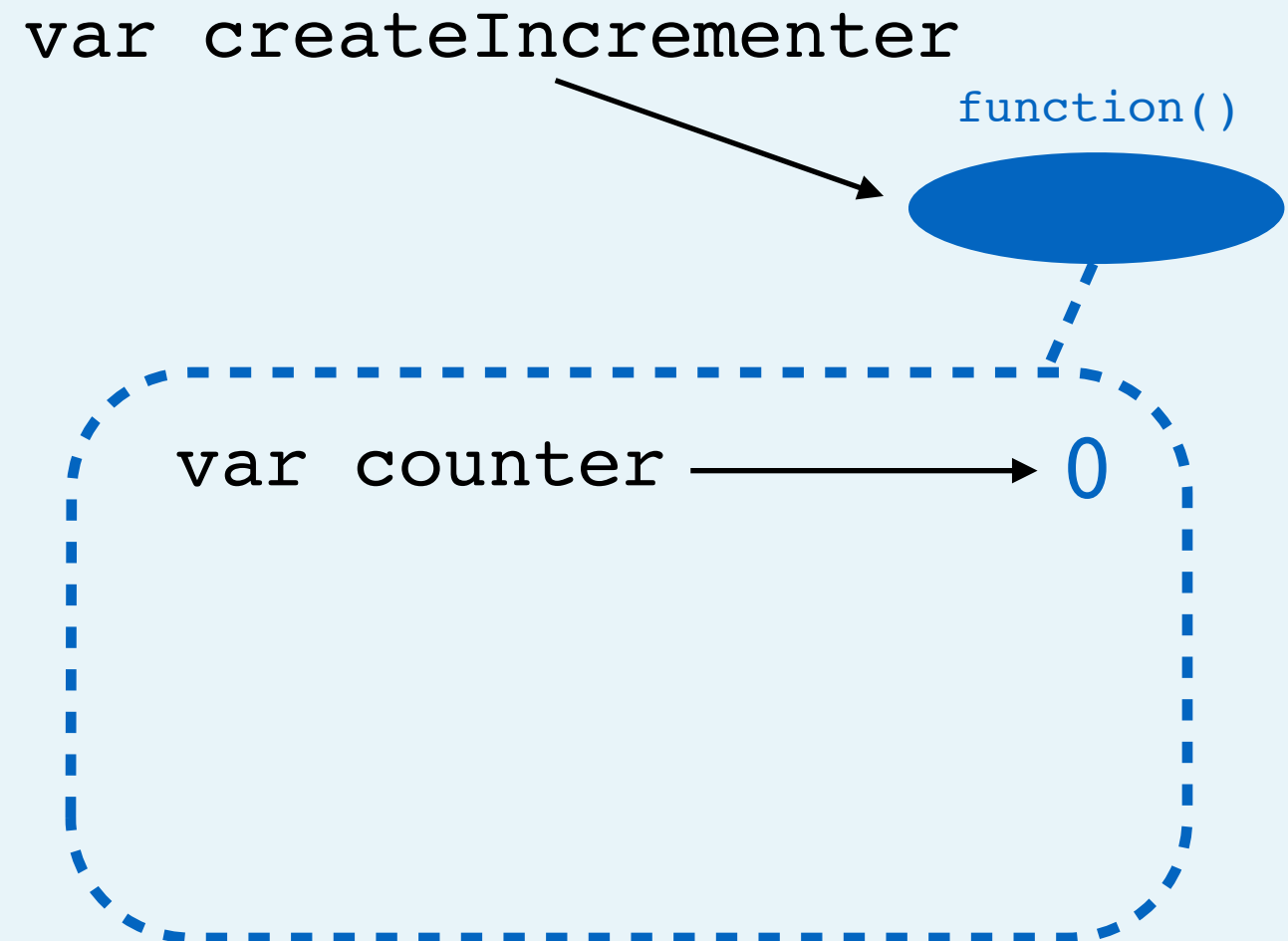
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope



# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

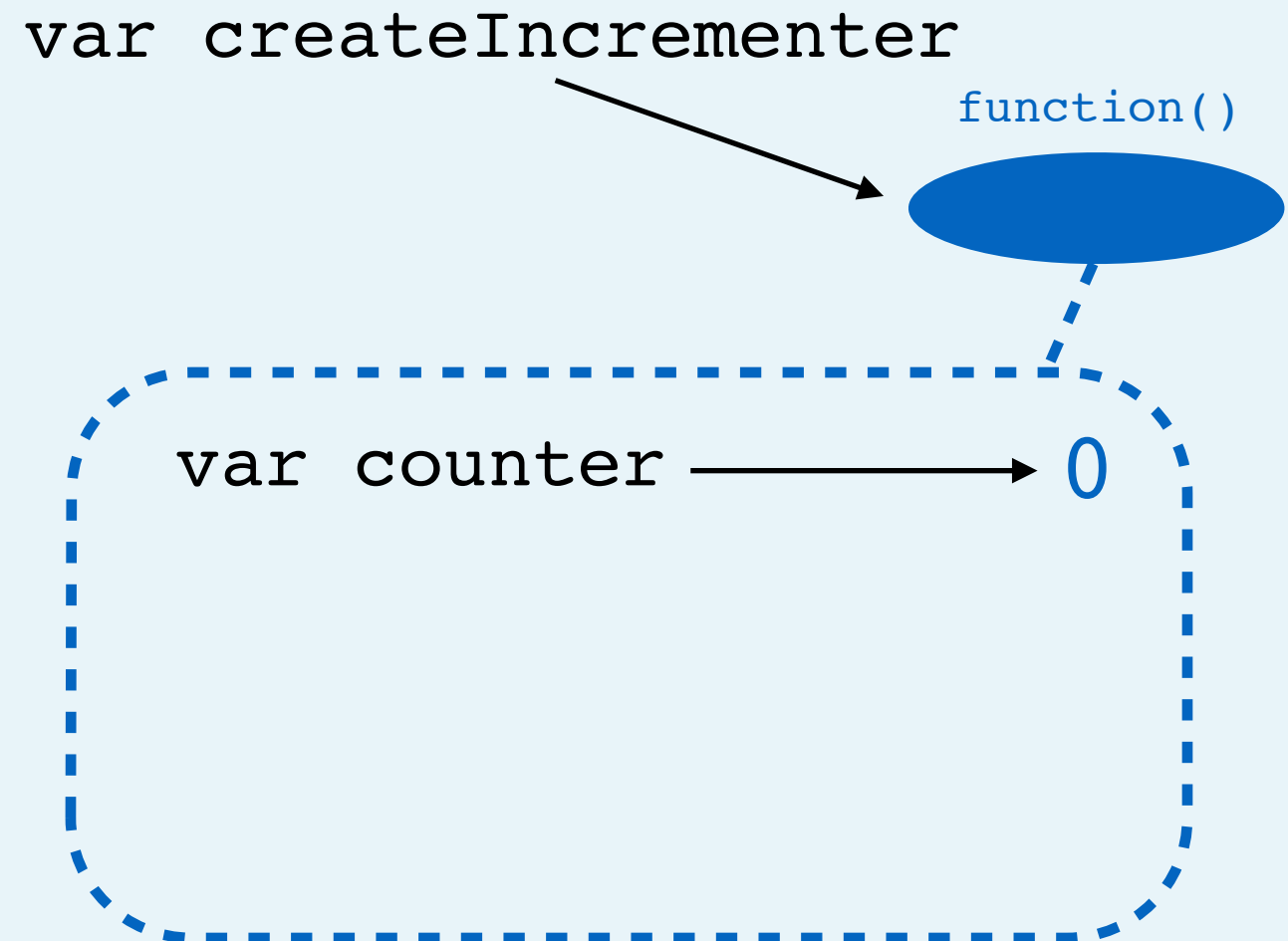
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)



# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

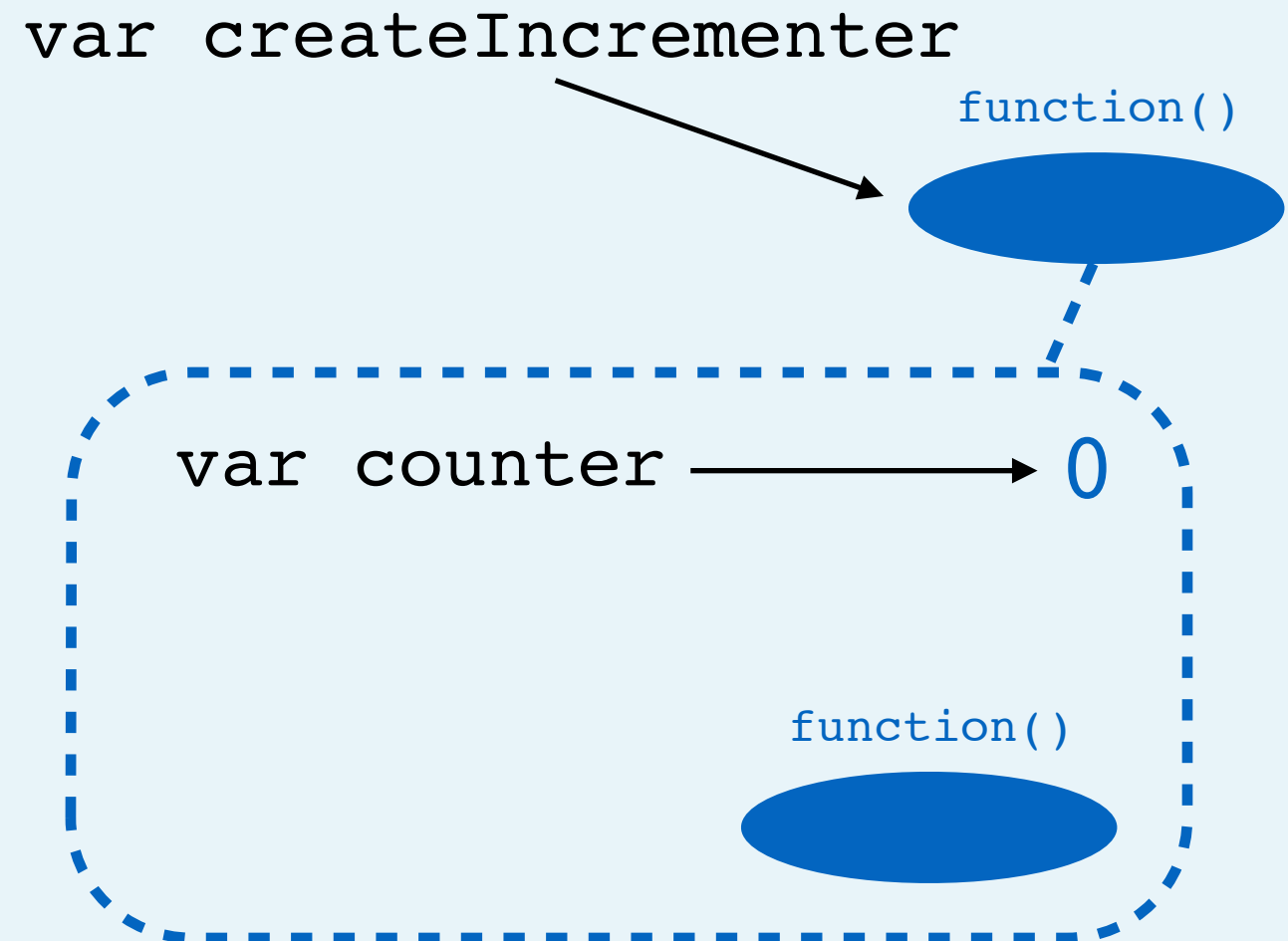
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement



# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement
      - a. Create function

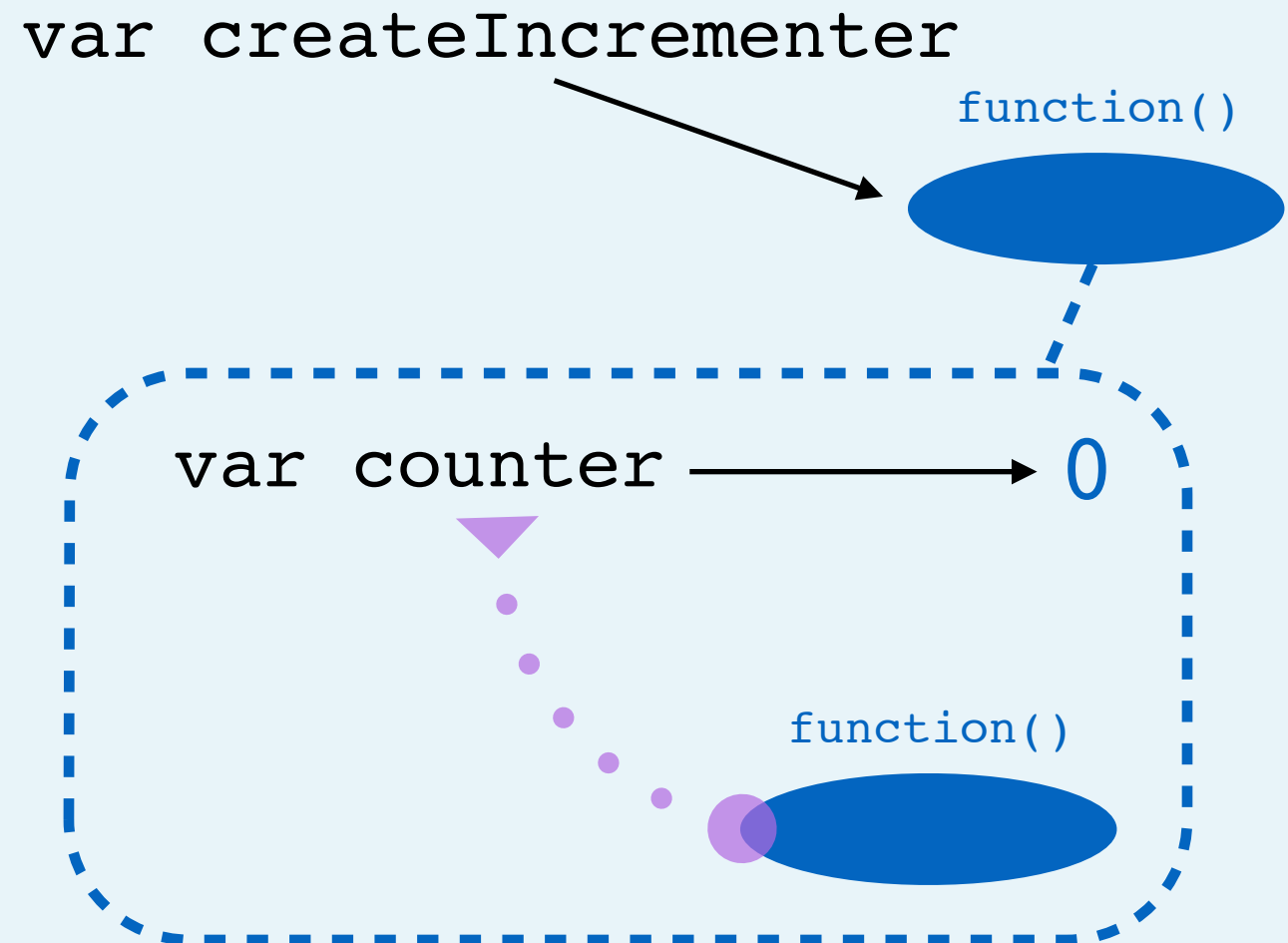




# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

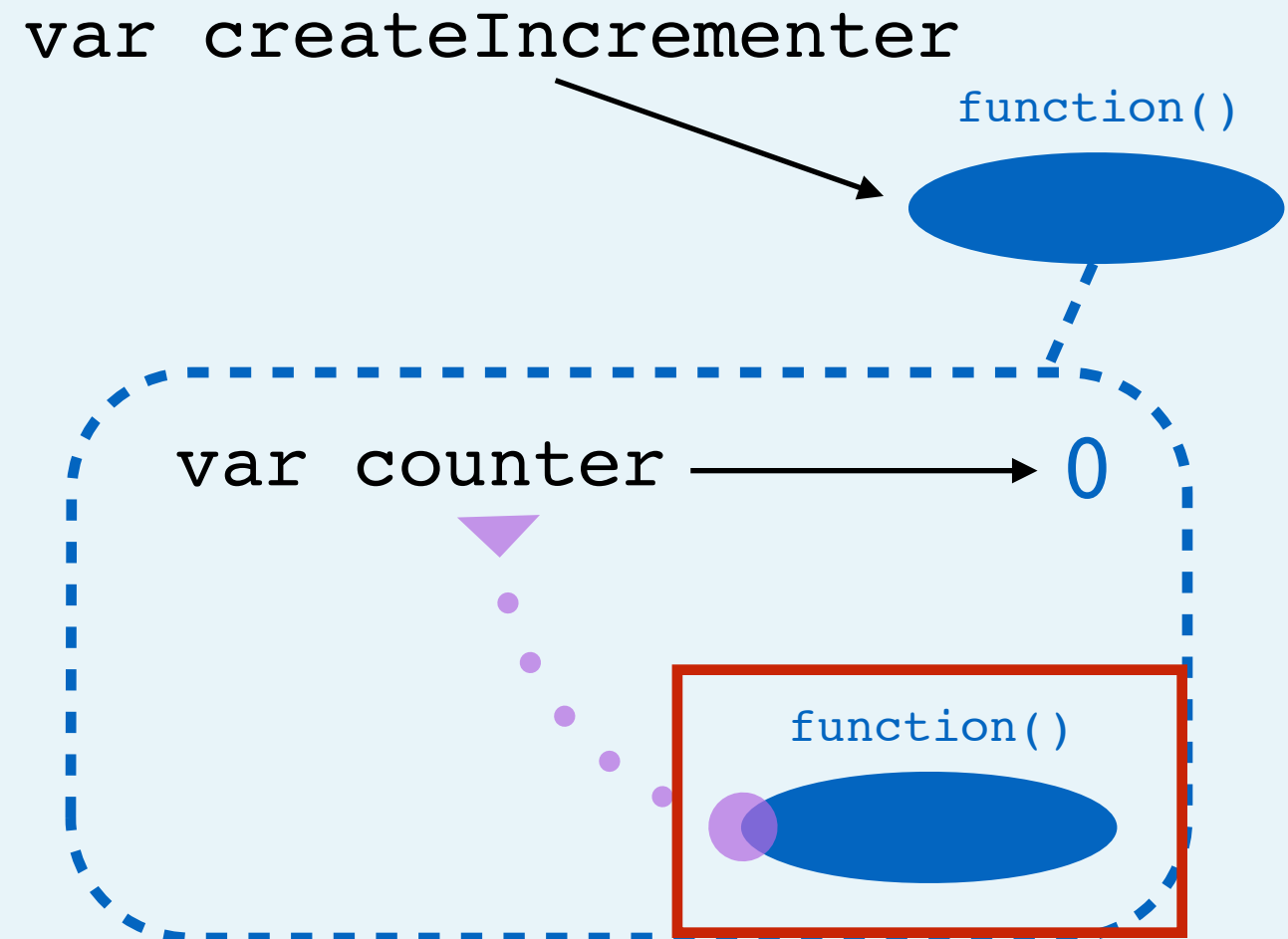
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement
      - a. Create function
        - a. Create closure (var counter)



# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

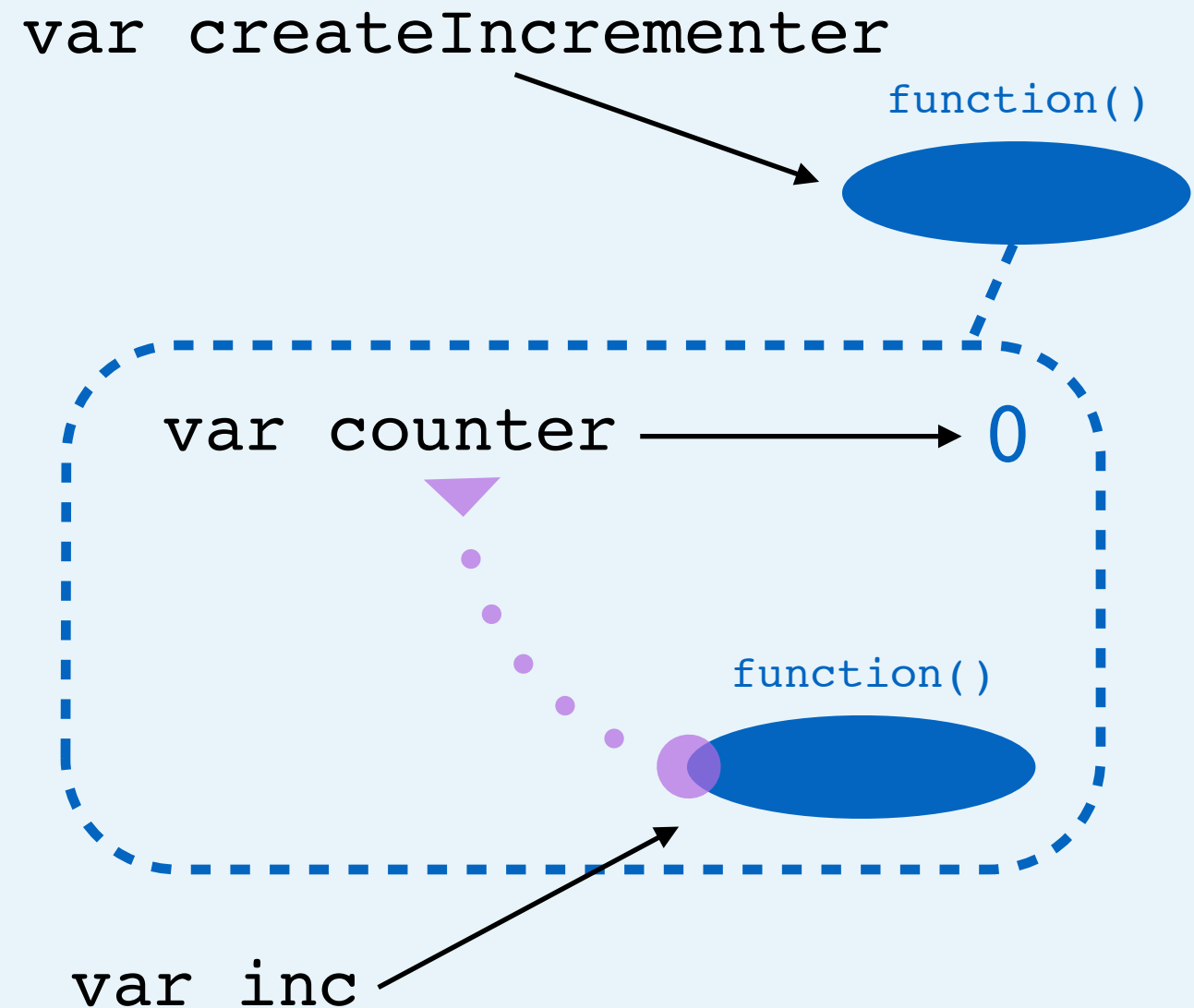
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement
      - a. Create function
        - a. Create closure (var counter)
        - b. Mark as return value



# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement
      - a. Create function
        - a. Create closure (var counter)
        - b. Mark as return value
- b. Create var, point to value

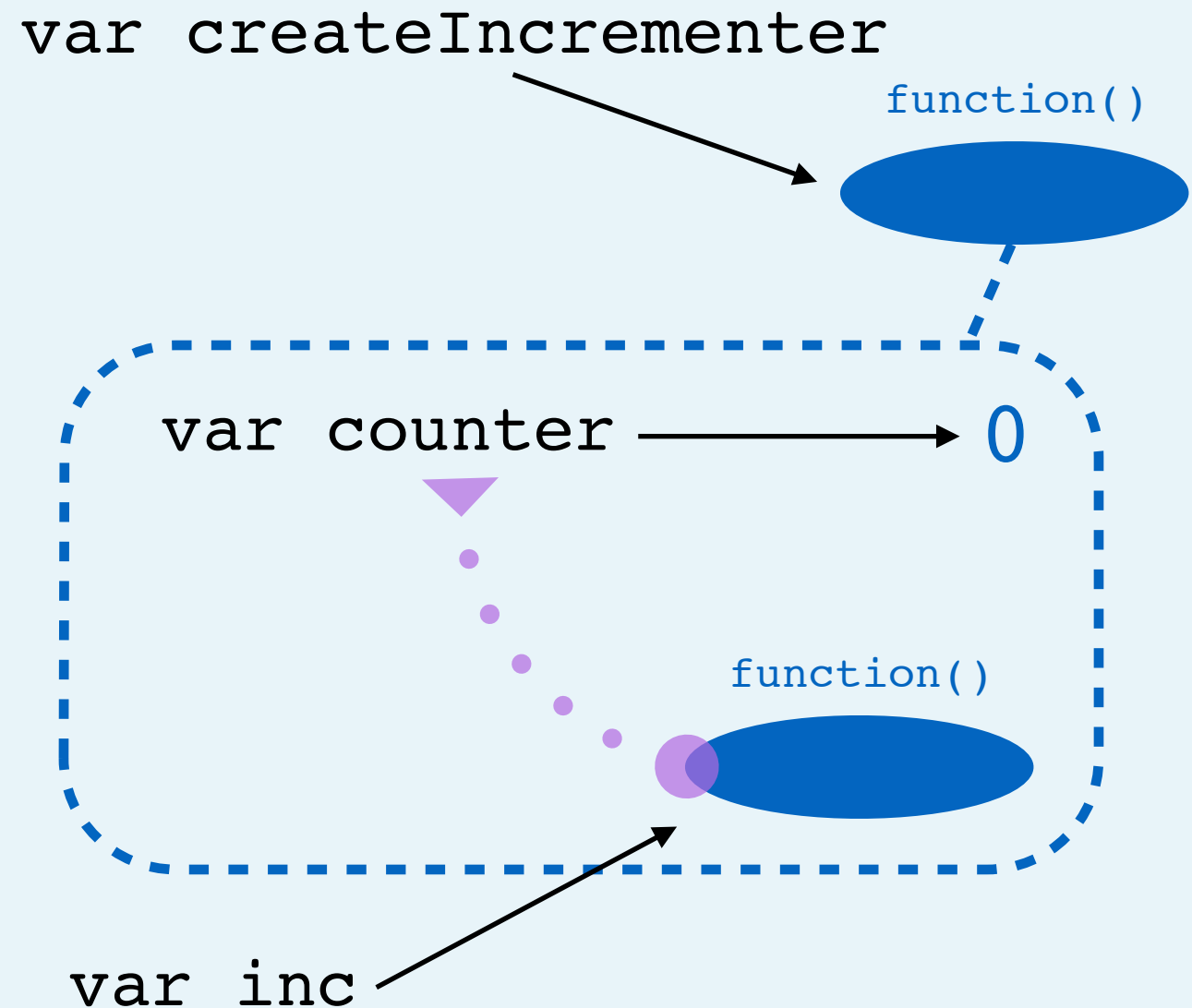


# A Closure in Action (Setup)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

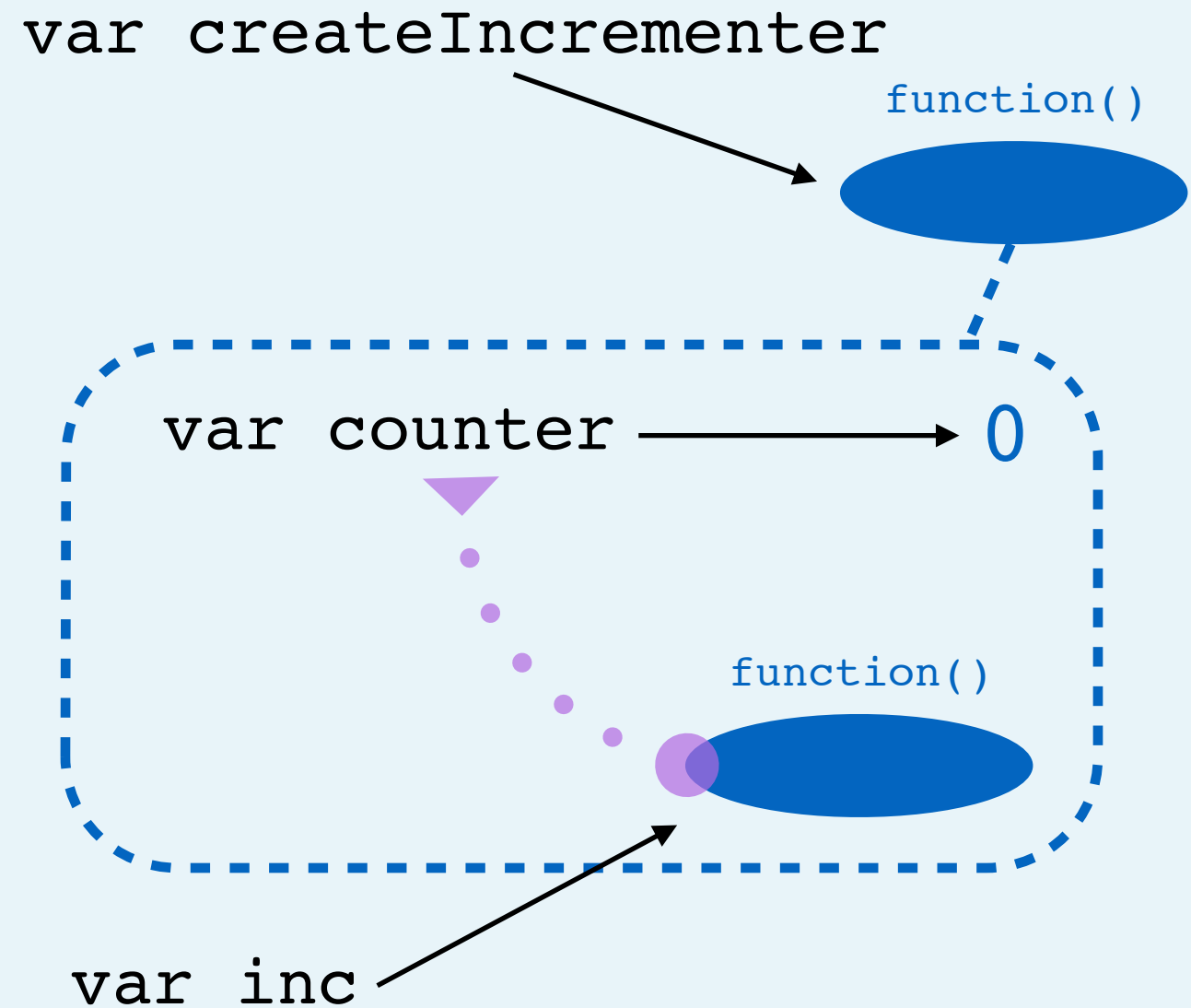
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. Assignment (steps omitted)
    - c. Return statement
      - a. Create function
        - a. Create closure (var counter)
        - b. Mark as return value
  - b. Create var, point to value

All done!



# A Closure in Action (Usage)

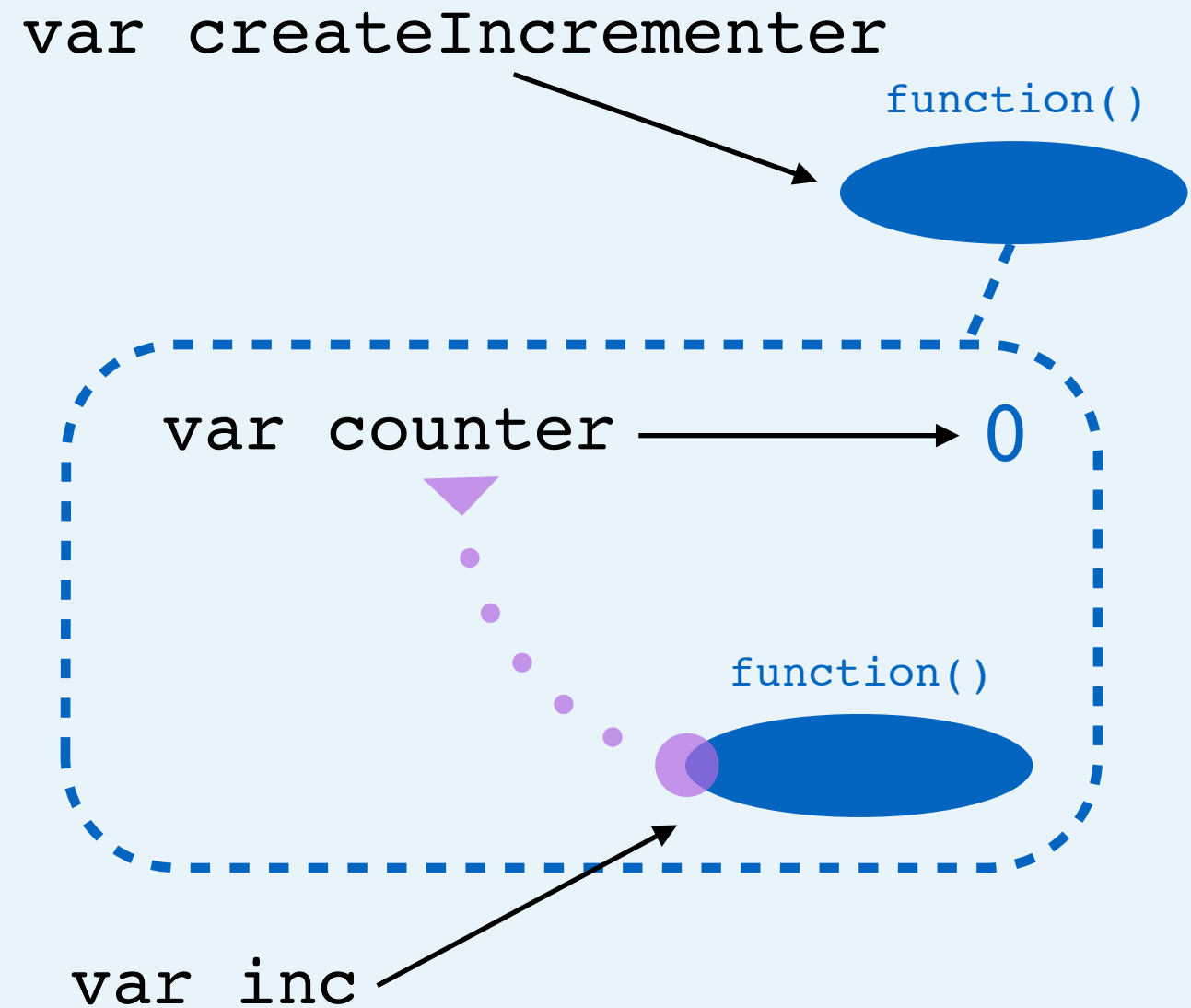
```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

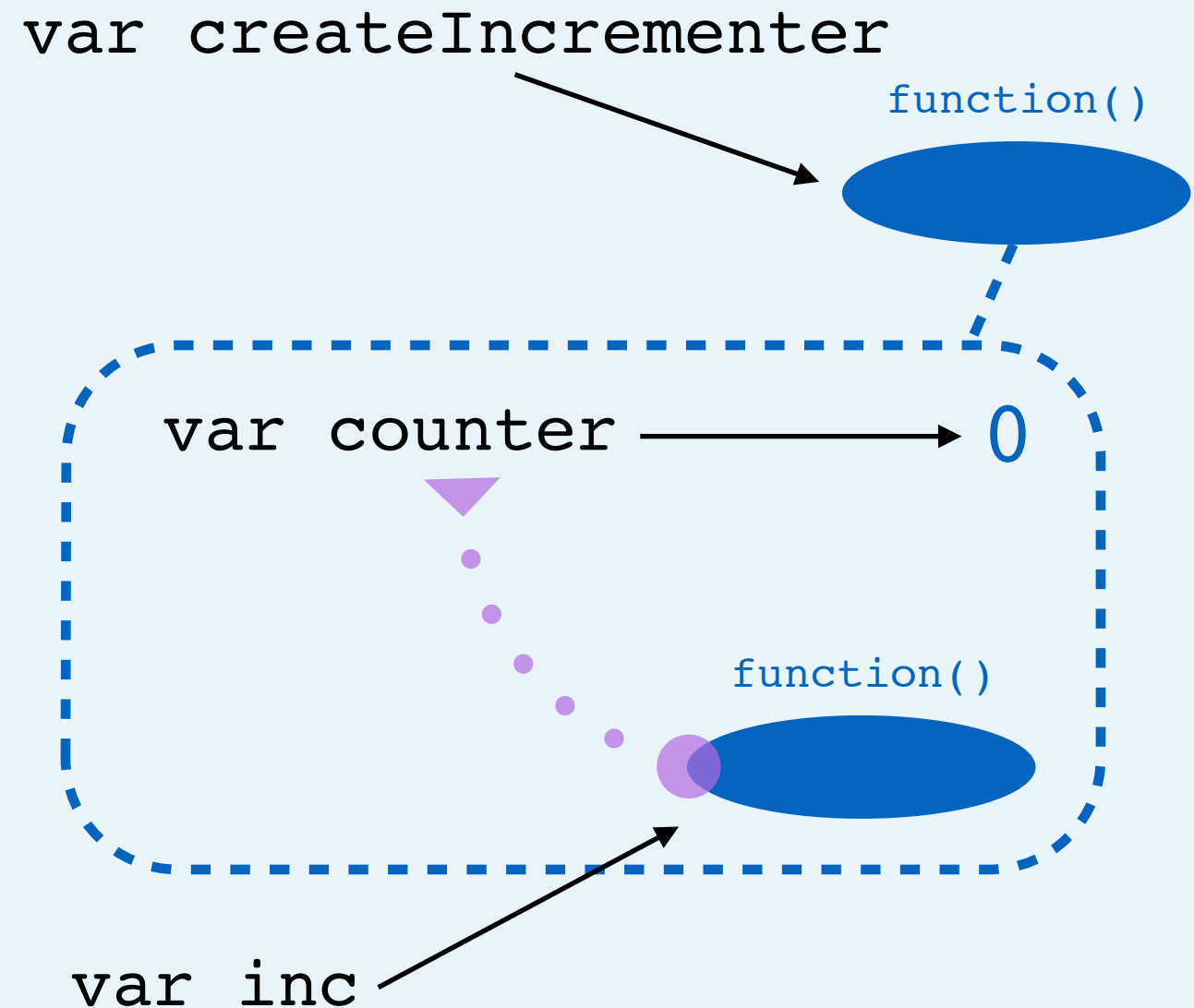
a. Assignment



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

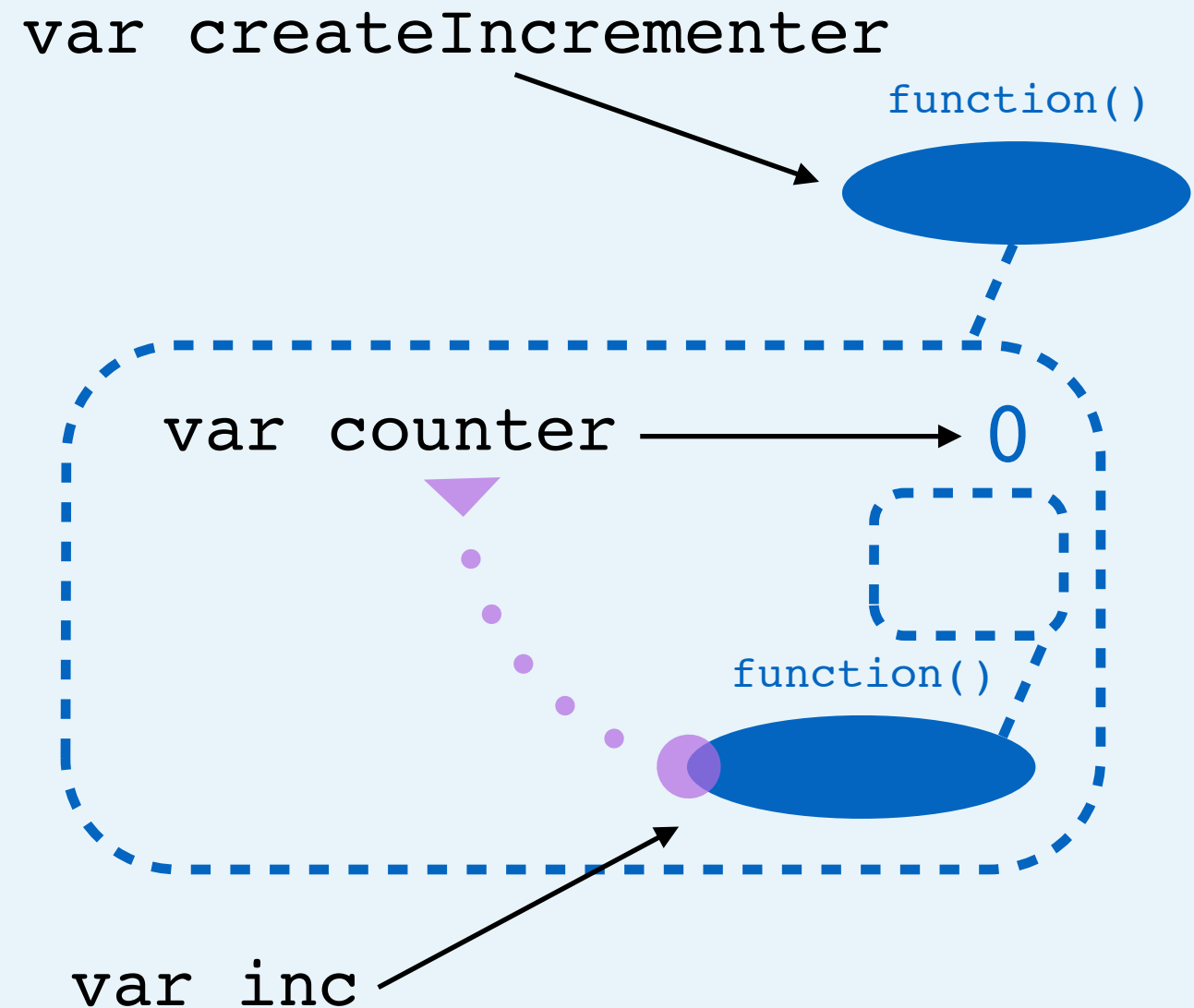
- a. Assignment
  - a. (Evaluate right side) Call function



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope

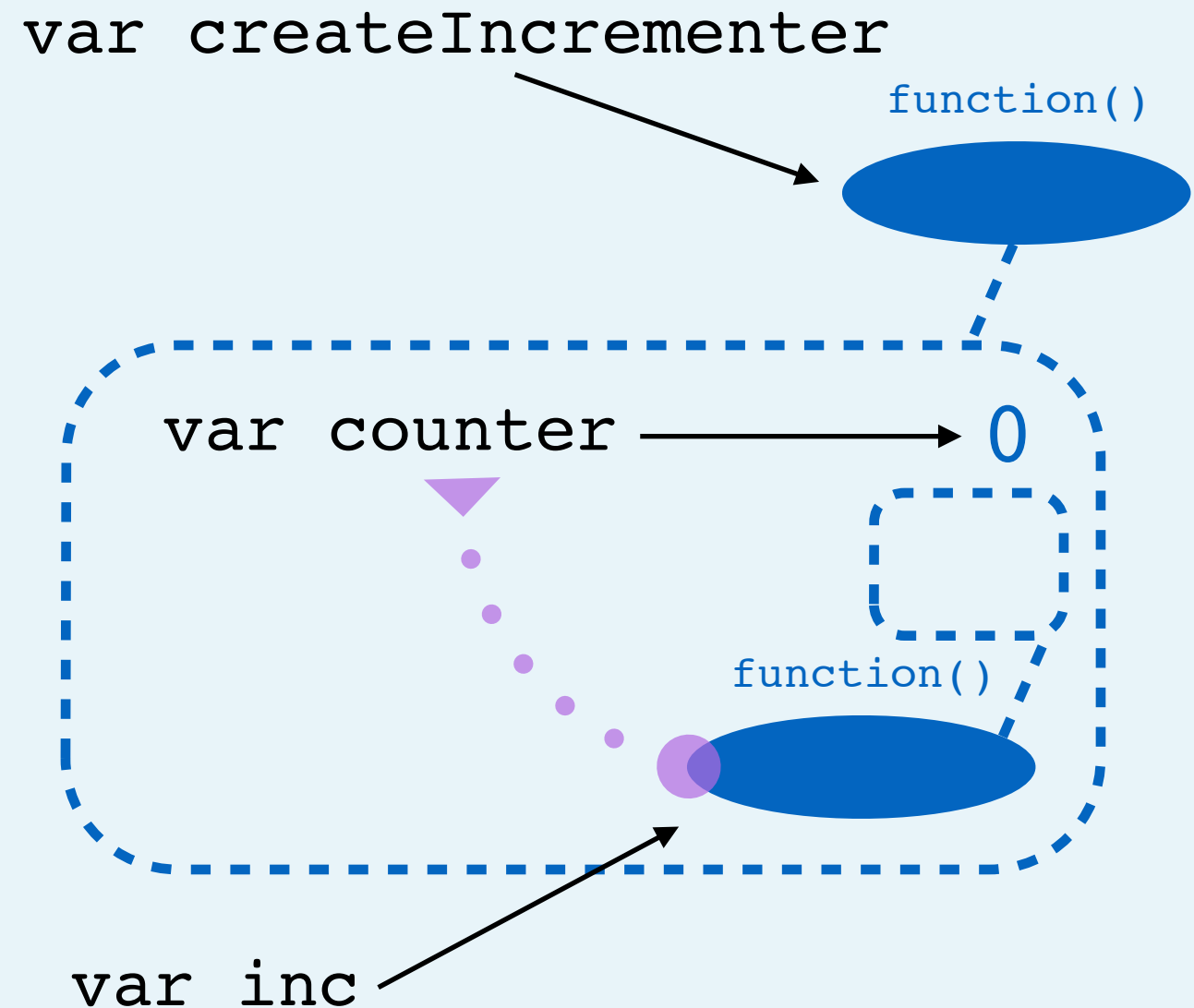




# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

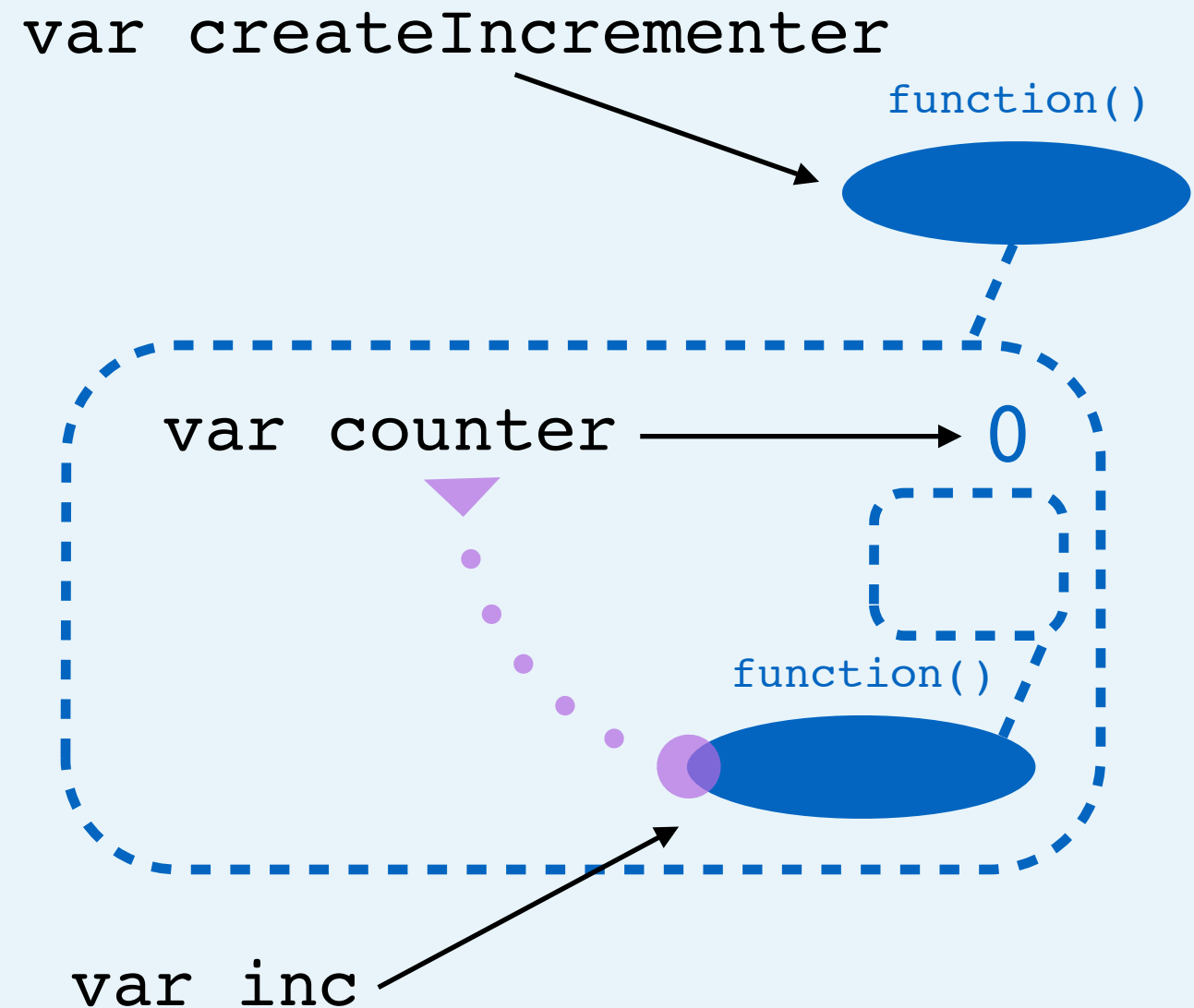
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. **Re**assignment



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

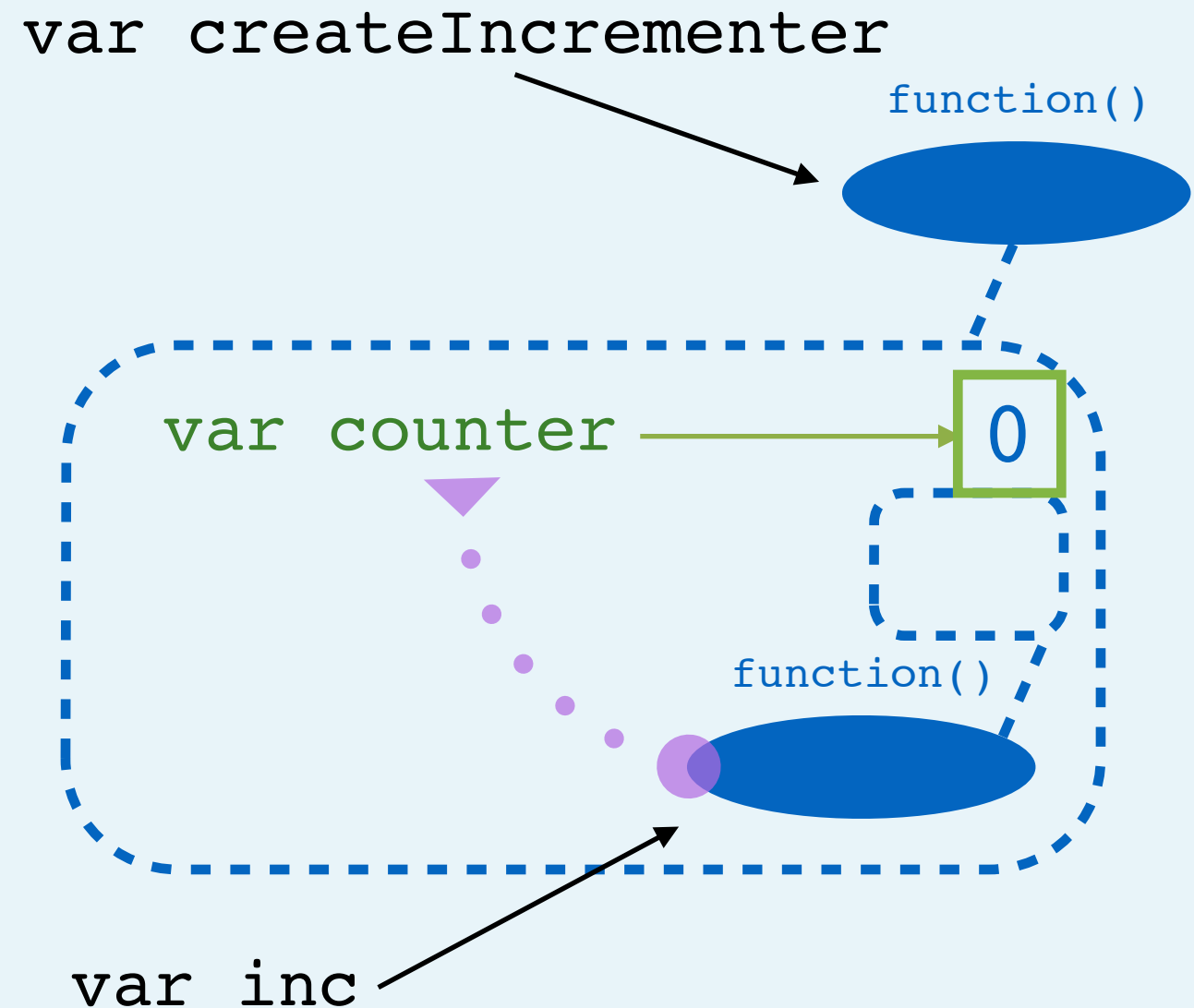
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. **Re**assignment
      - a. Binary Operation (addition)



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

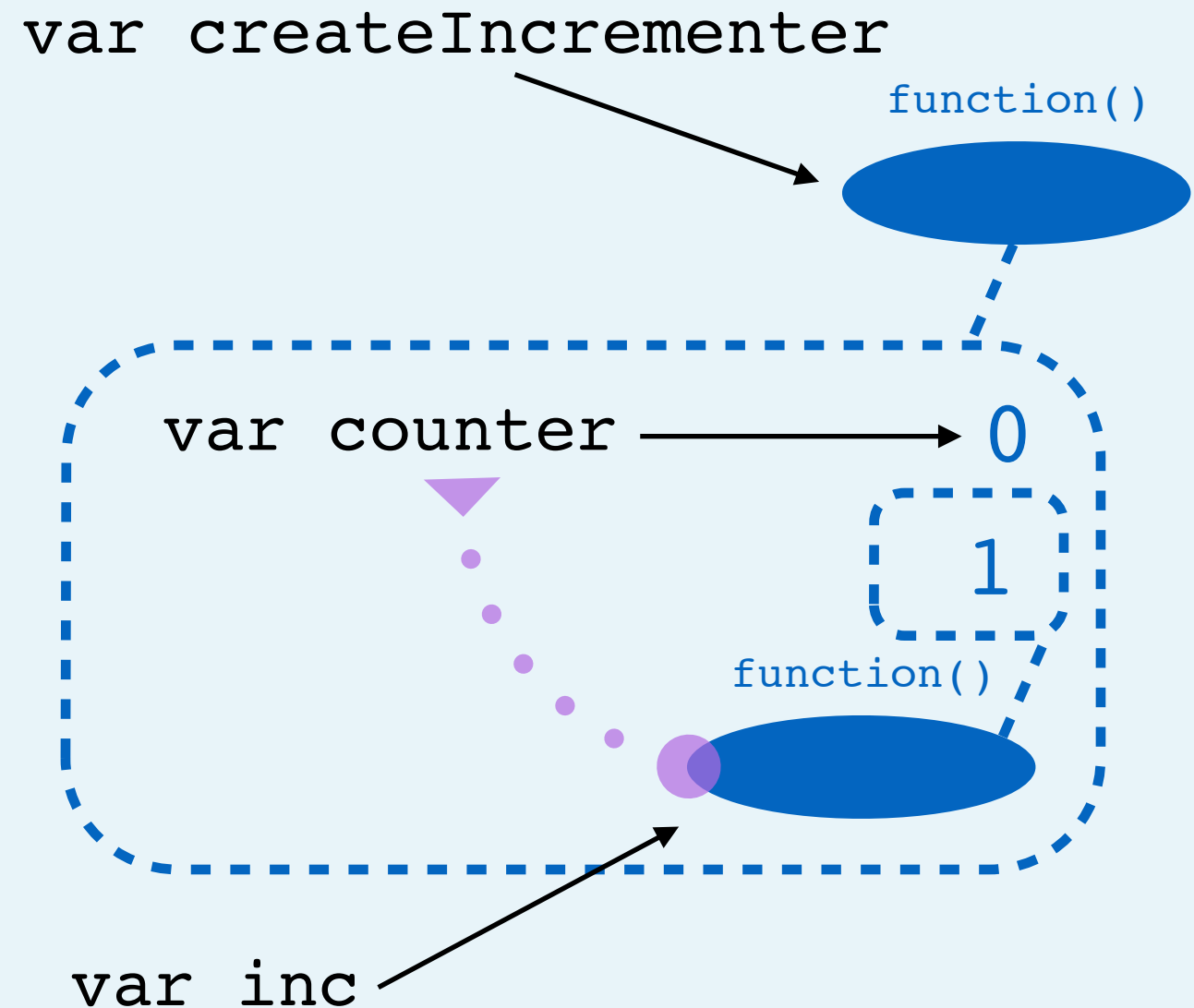
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. **Re**assignment
      - a. Binary Operation (addition)
        - a. Look up value of counter



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

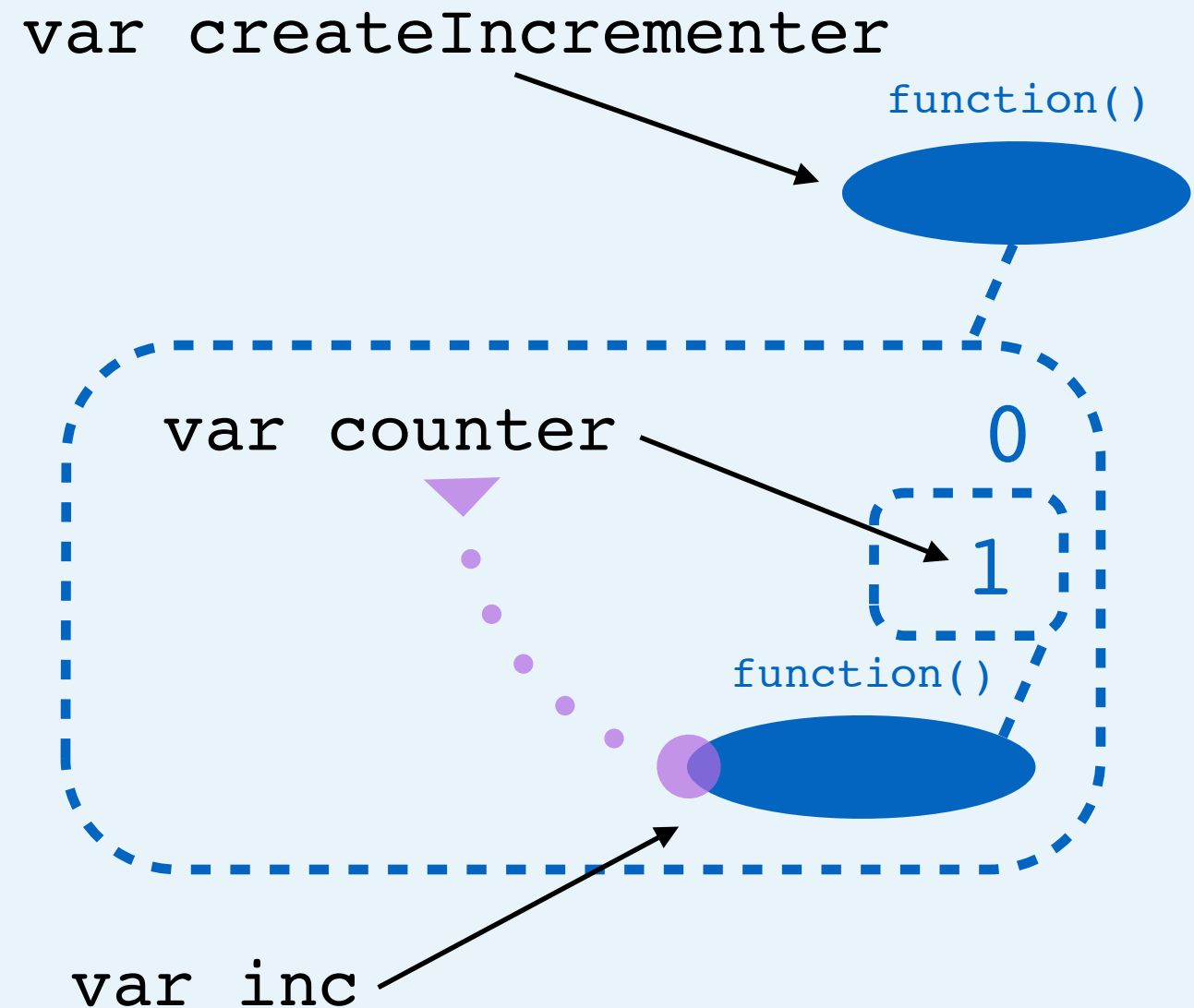
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
    - b. **Re**assignment
      - a. Binary Operation (addition)
        - a. Look up value of counter
        - b. Create value



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
      - b. **Re**assignment
        - a. Binary Operation (addition)
          - a. Look up value of counter
          - b. Create value
        - b. Set var to point to value



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

a. Assignment

a. (Evaluate right side) Call function

a. Create scope

b. **Re**assignment

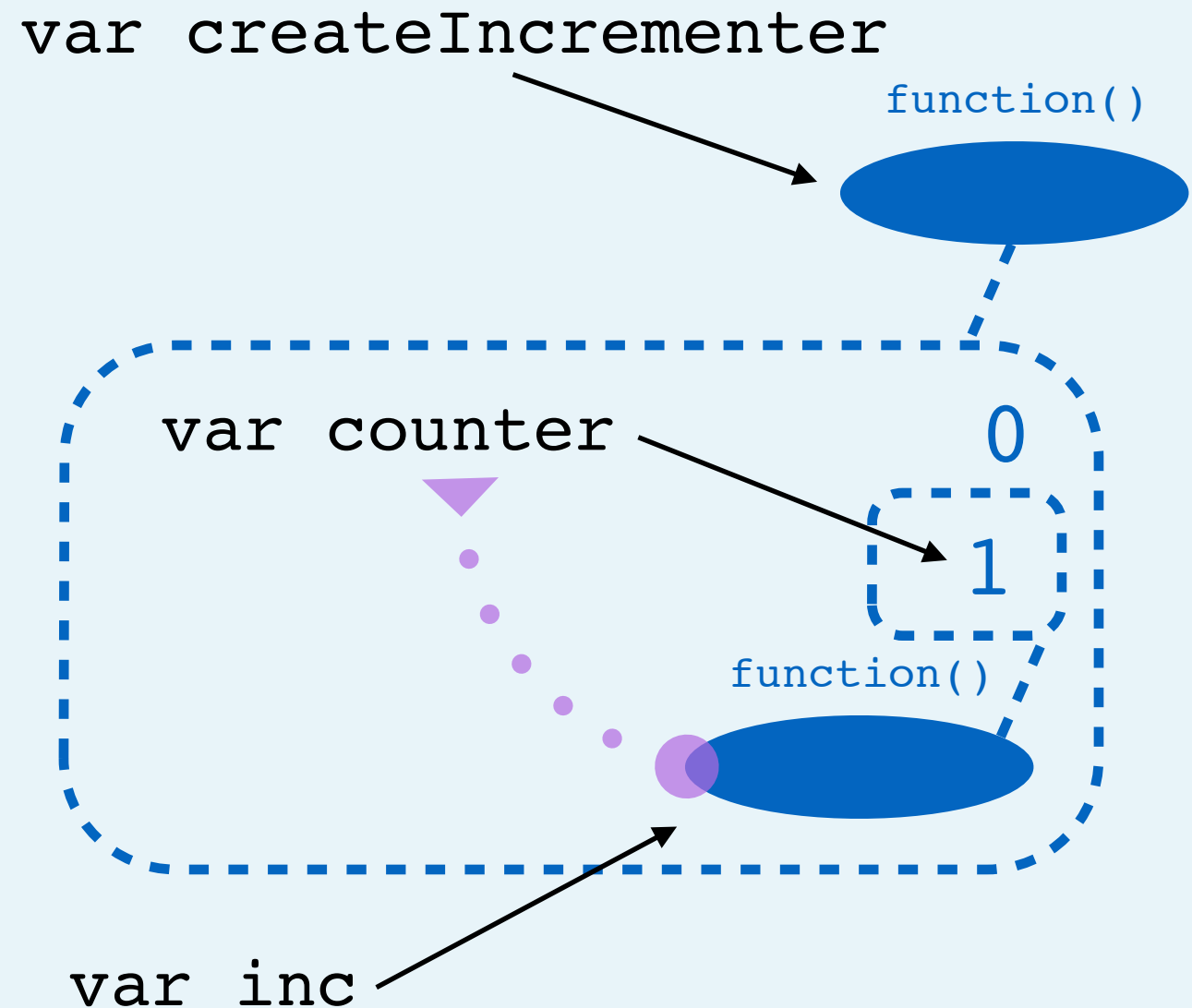
a. Binary Operation (addition)

a. Look up value of counter

b. Create value

b. Set var to point to value

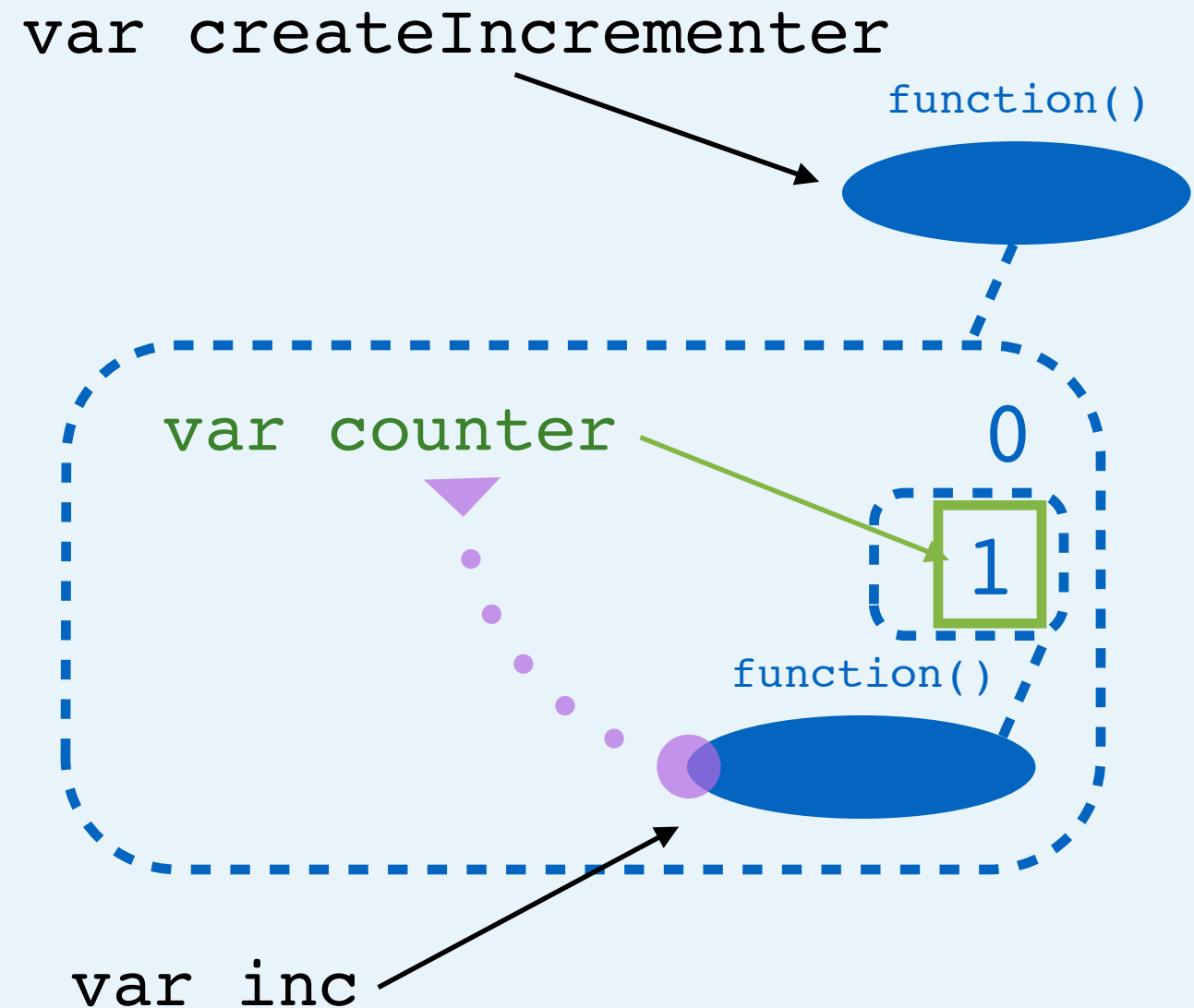
c. Return statement



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
      - b. **Re**assignment
        - a. Binary Operation (addition)
          - a. Look up value of counter
          - b. Create value
        - b. Set var to point to value
      - c. Return statement
        - a. Look up value of counter



# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

a. Assignment

a. (Evaluate right side) Call function

a. Create scope

b. **Re**assignment

a. Binary Operation (addition)

a. Look up value of counter

b. Create value

b. Set var to point to value

c. Return statement

a. Look up value of counter

b. Mark as return value

var createIncrementer

function()

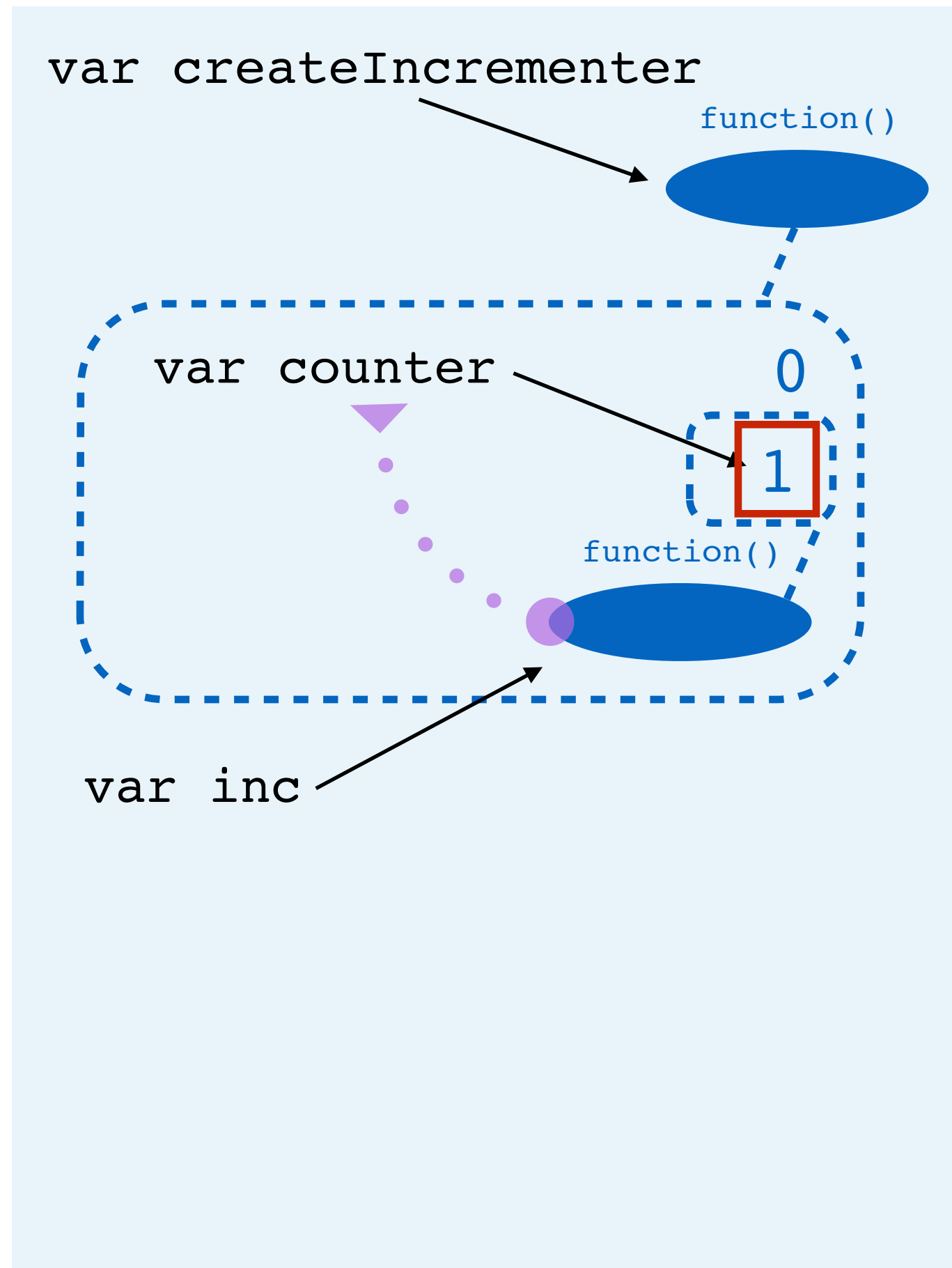
var counter

0

1

function()

var inc





# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

a. Assignment

a. (Evaluate right side) Call function

a. Create scope

b. **Re**assignment

a. Binary Operation (addition)

a. Look up value of counter

b. Create value

b. Set var to point to value

c. Return statement

a. Look up value of counter

b. Mark as return value

b. Create var, point to value

var createIncrementer

function()

var counter

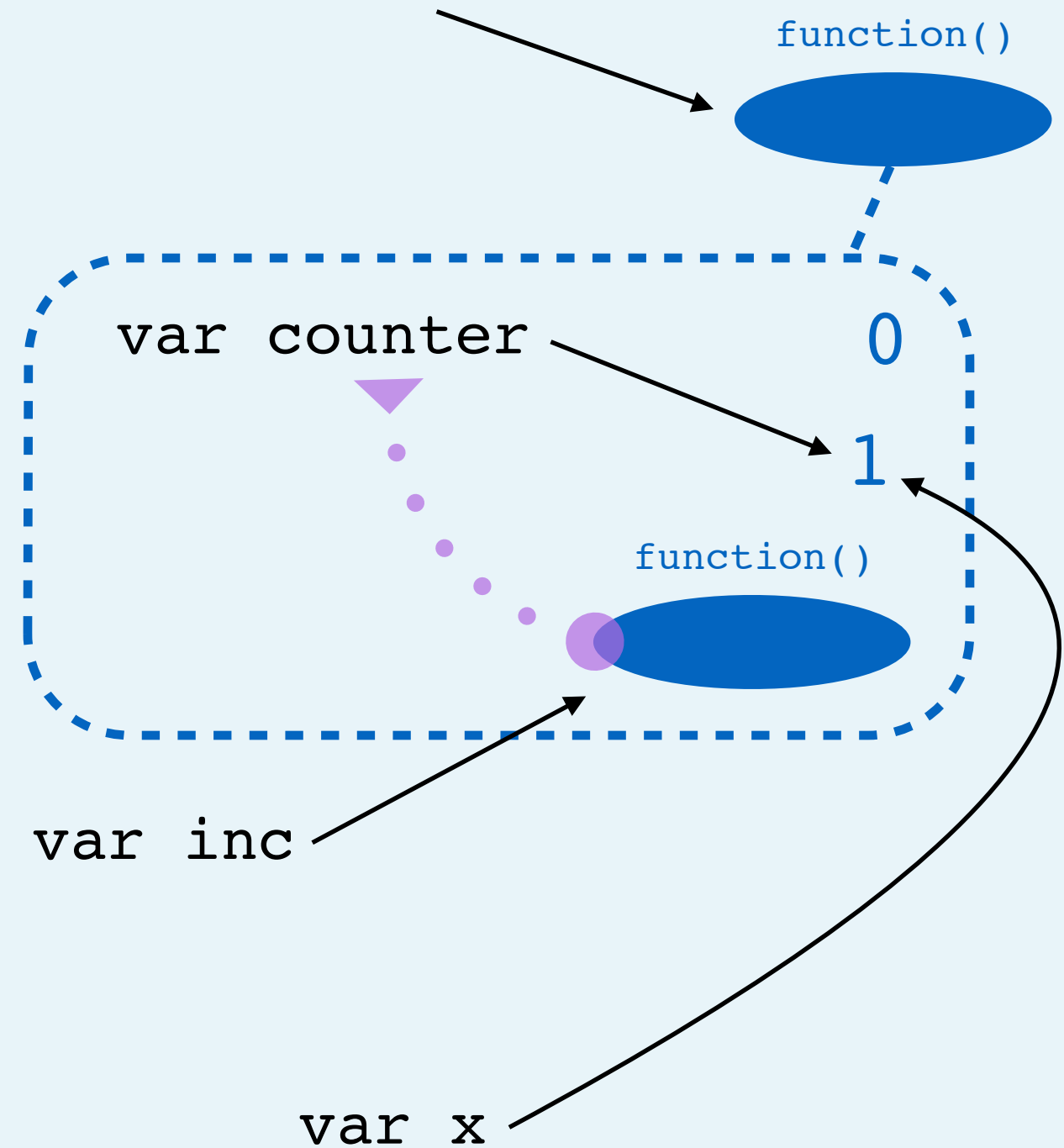
0

1

function()

var inc

var x

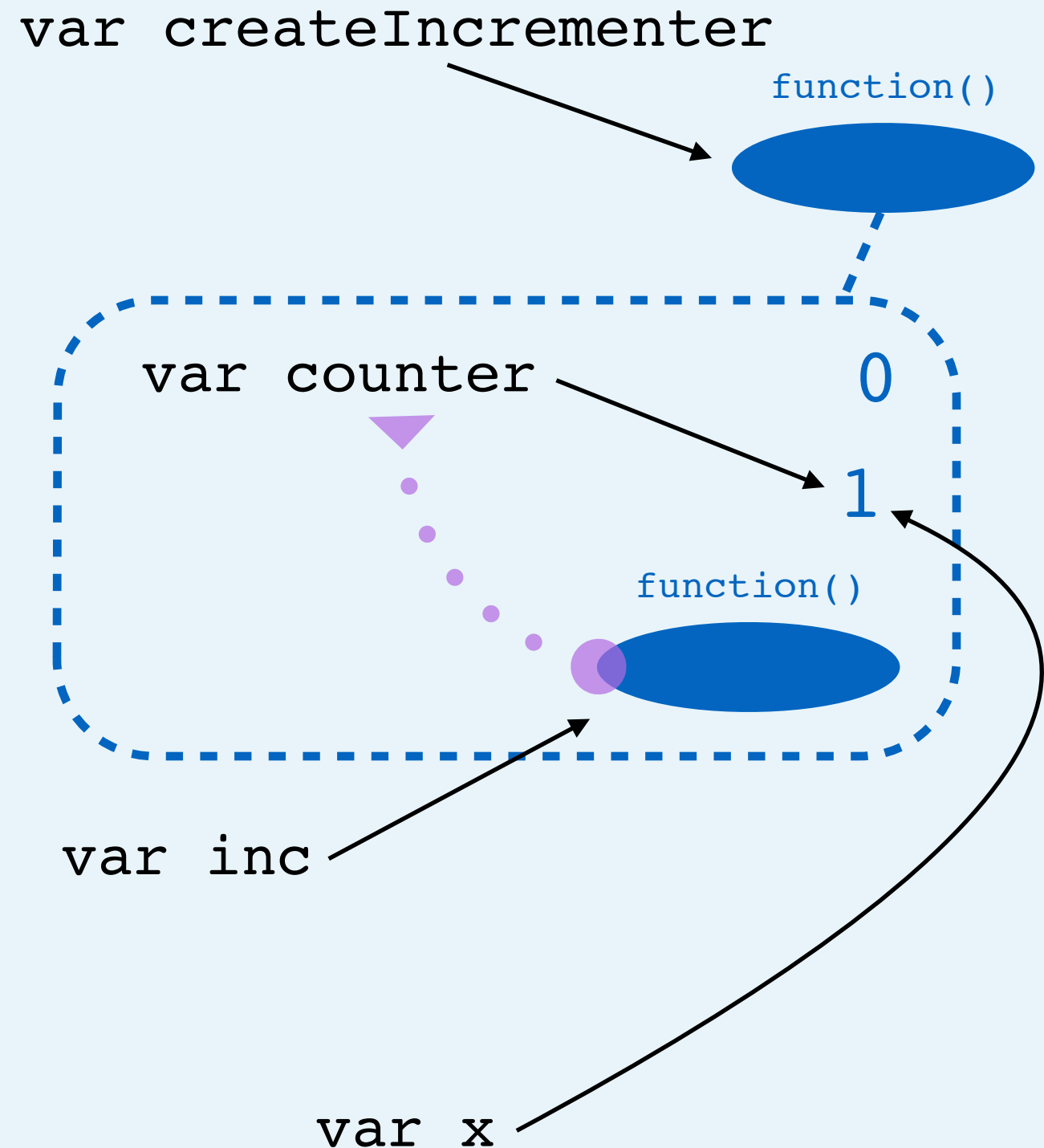


# A Closure in Action (Usage)

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var inc = createIncrementer();  
var x = inc();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create scope
      - b. **Re**assignment
        - a. Binary Operation (addition)
          - a. Look up value of counter
          - b. Create value
        - b. Set var to point to value
      - c. Return statement
        - a. Look up value of counter
        - b. Mark as return value
    - b. Create var, point to value

All done!

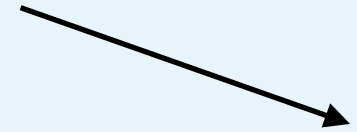


# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

var createIncrementer

function()



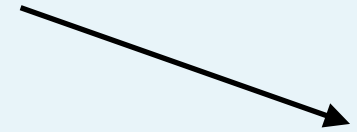
# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

a. Assignment

var createIncrementer

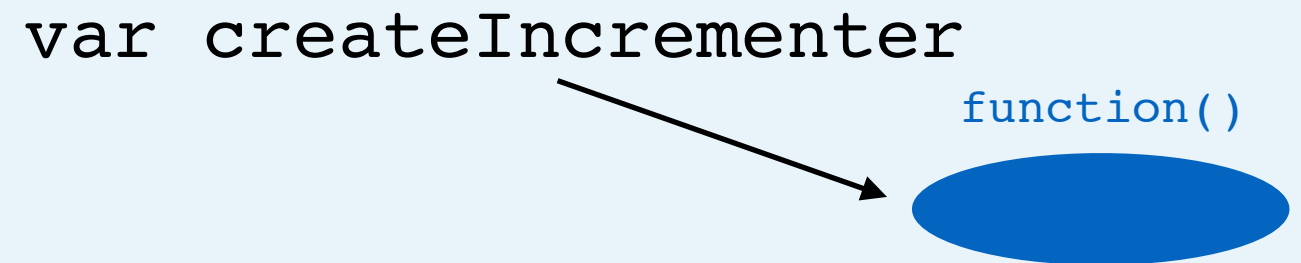
function()



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

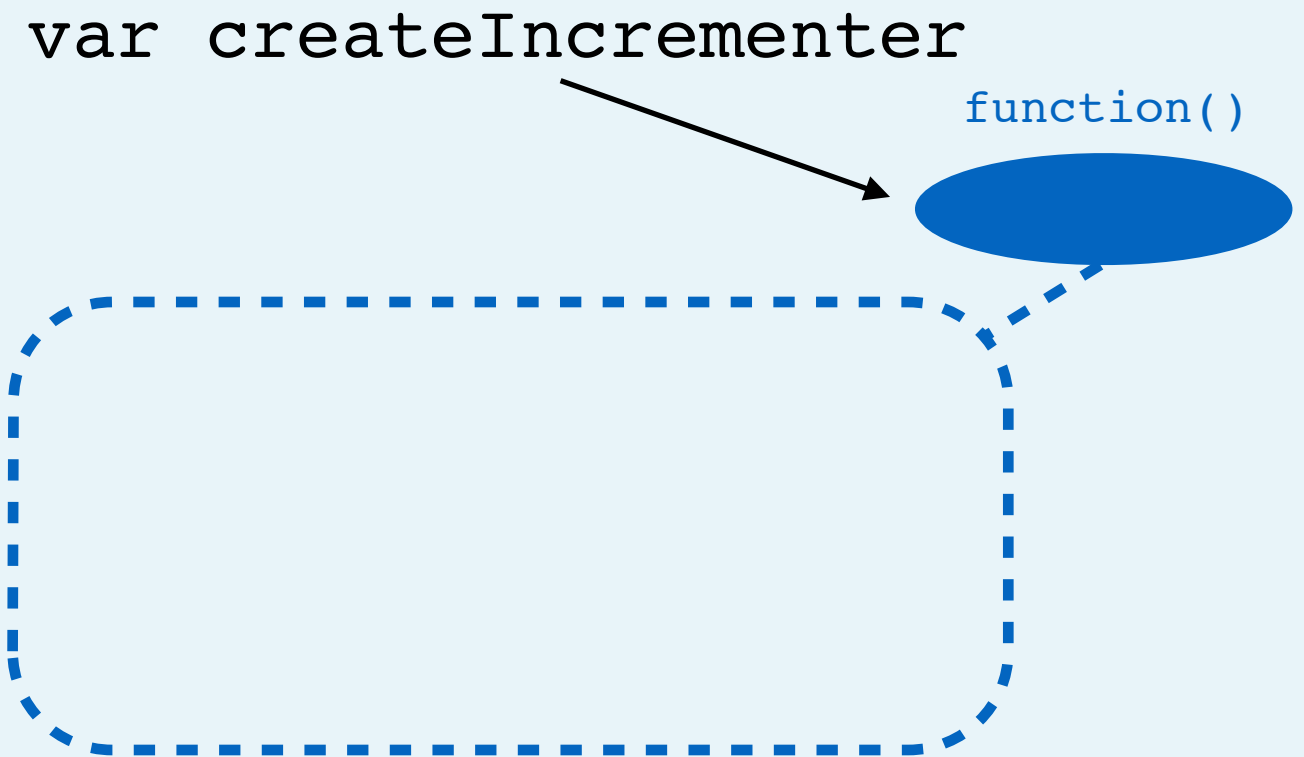
- a. Assignment
  - a. (Evaluate right side) Call function



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

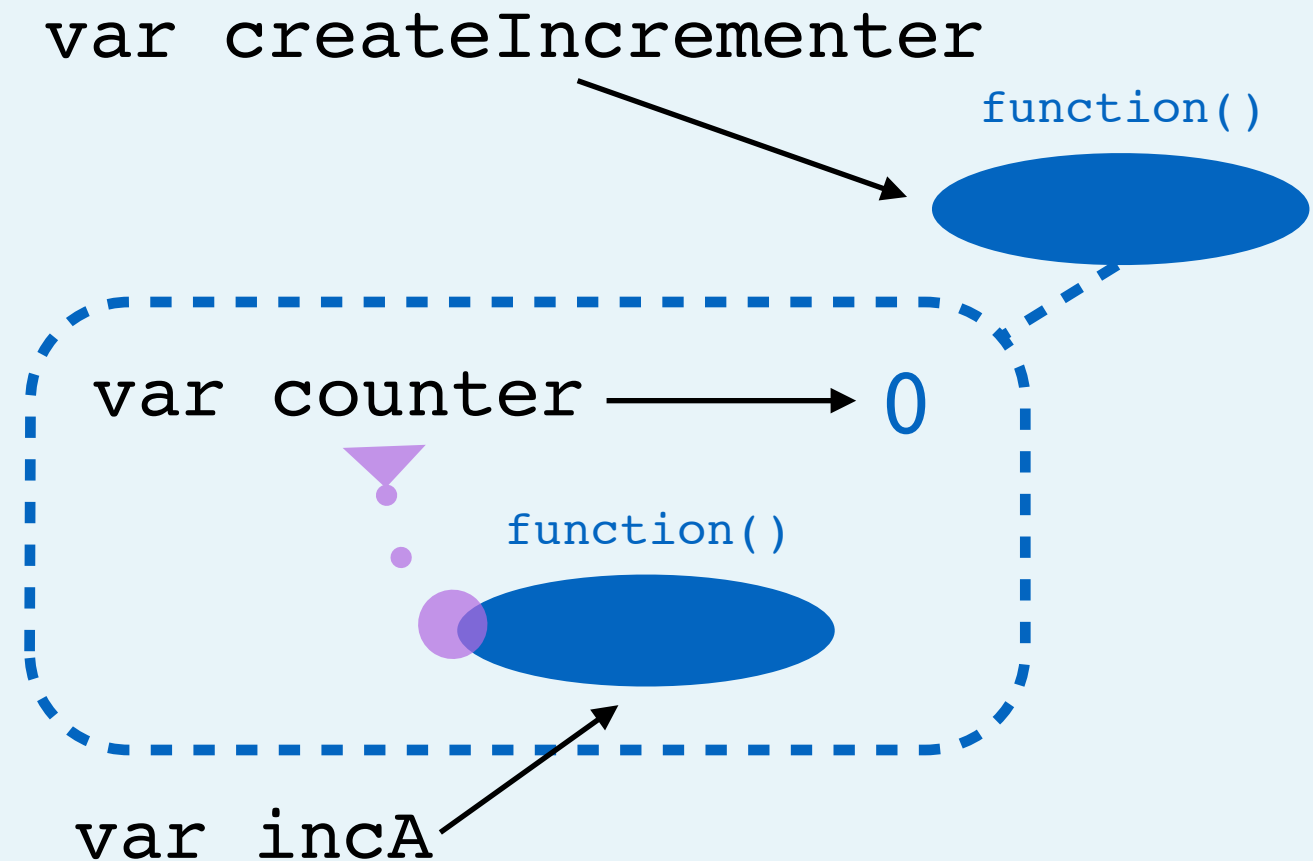
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
  
var incA = createIncrementer();  
var incB = createIncrementer();
```

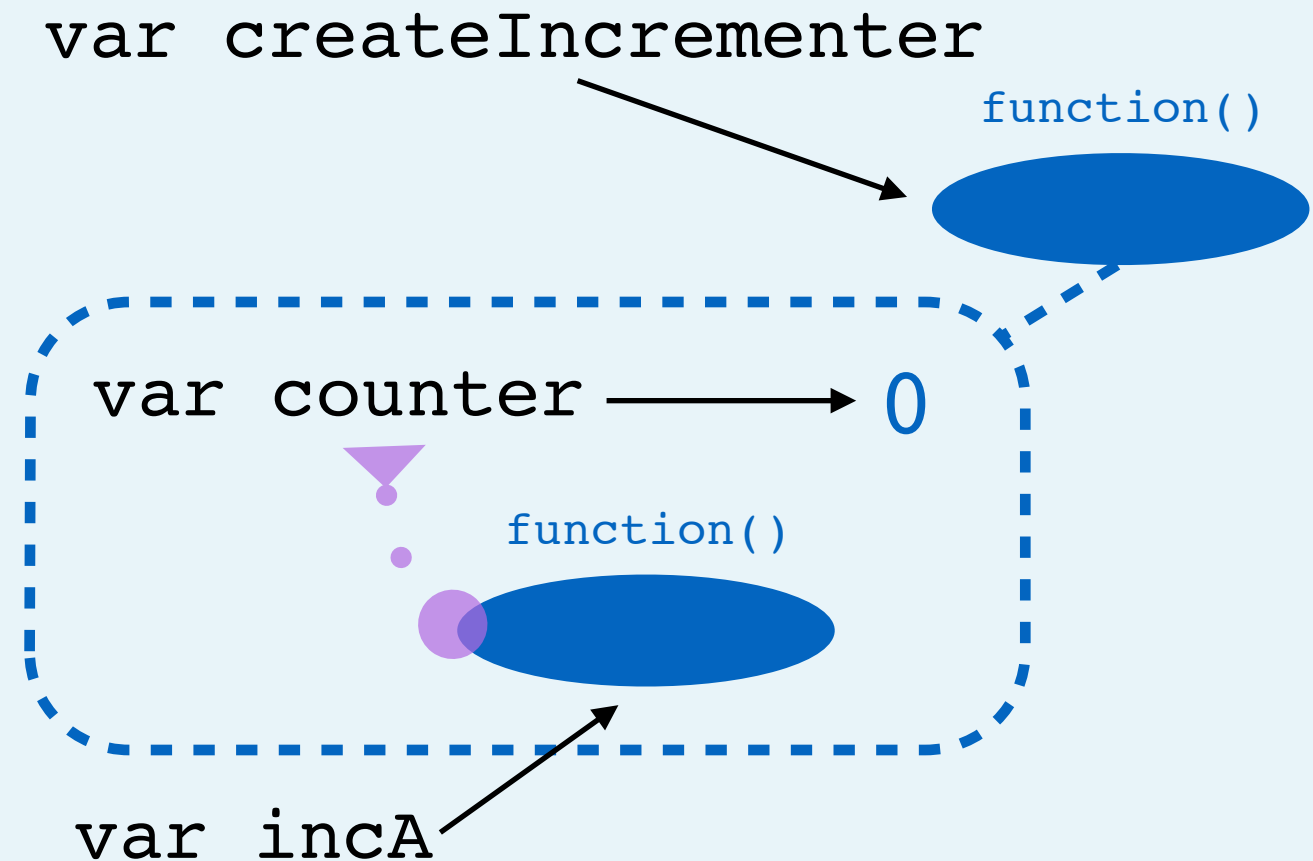
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.
  - b. Assignment

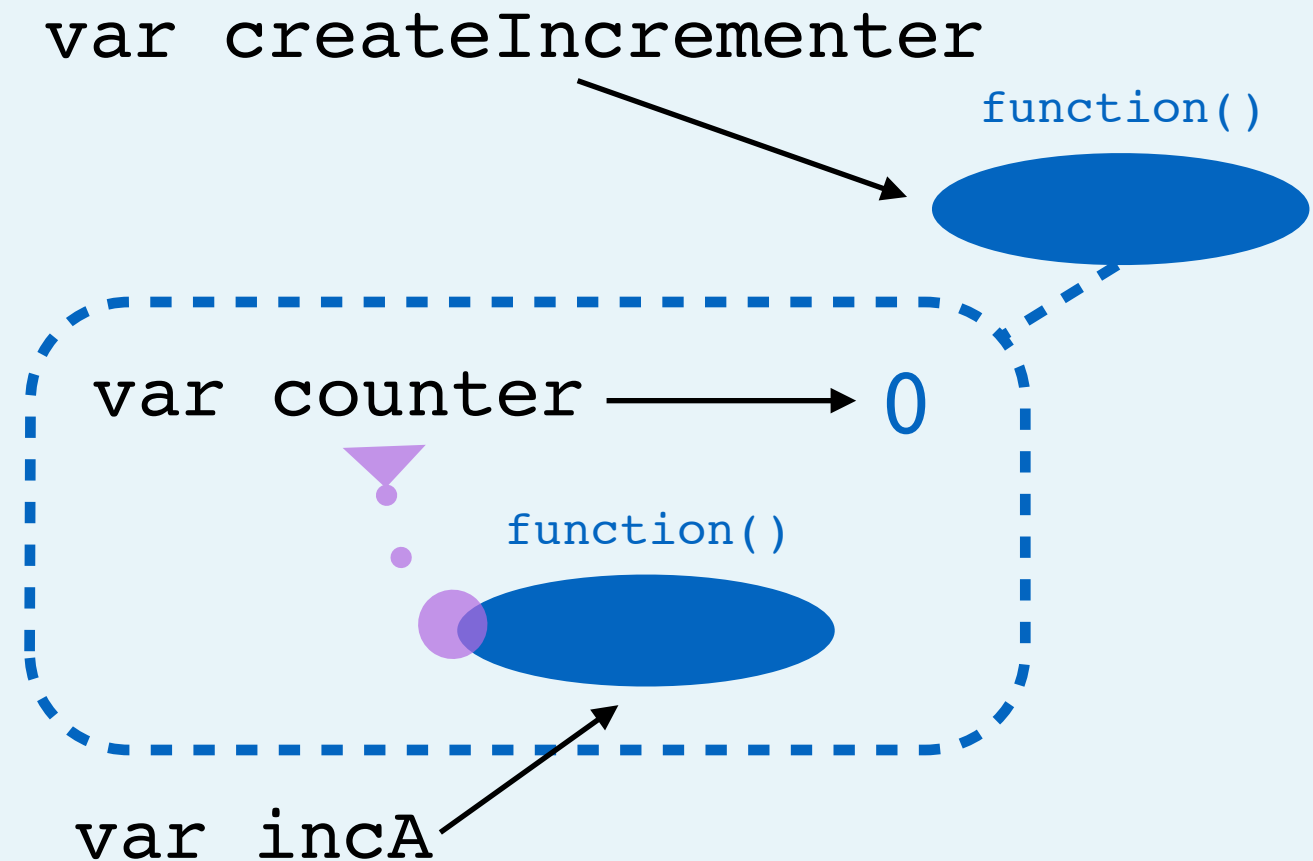




# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

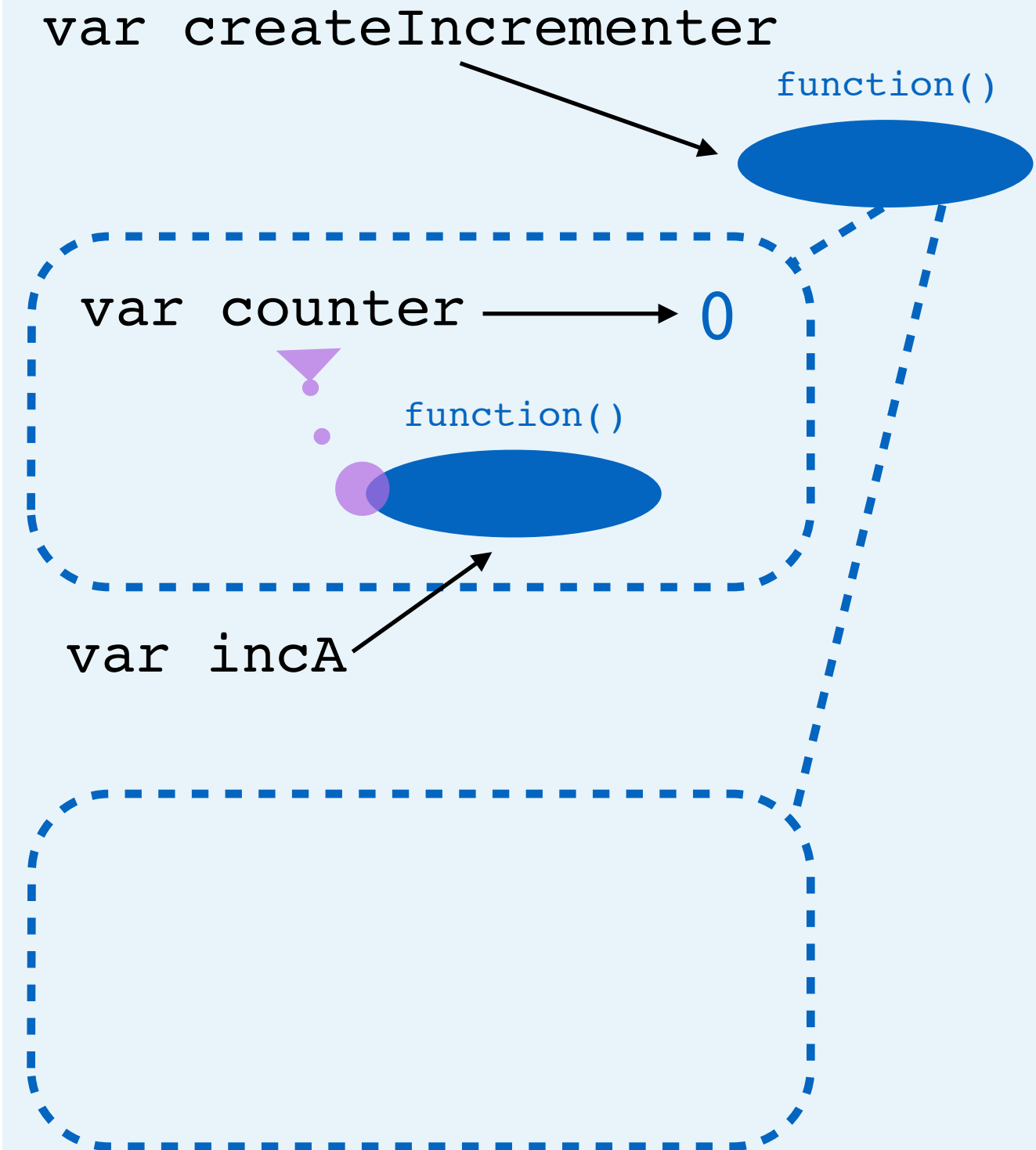
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.
- b. Assignment
  - a. (Evaluate right side) Call function



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

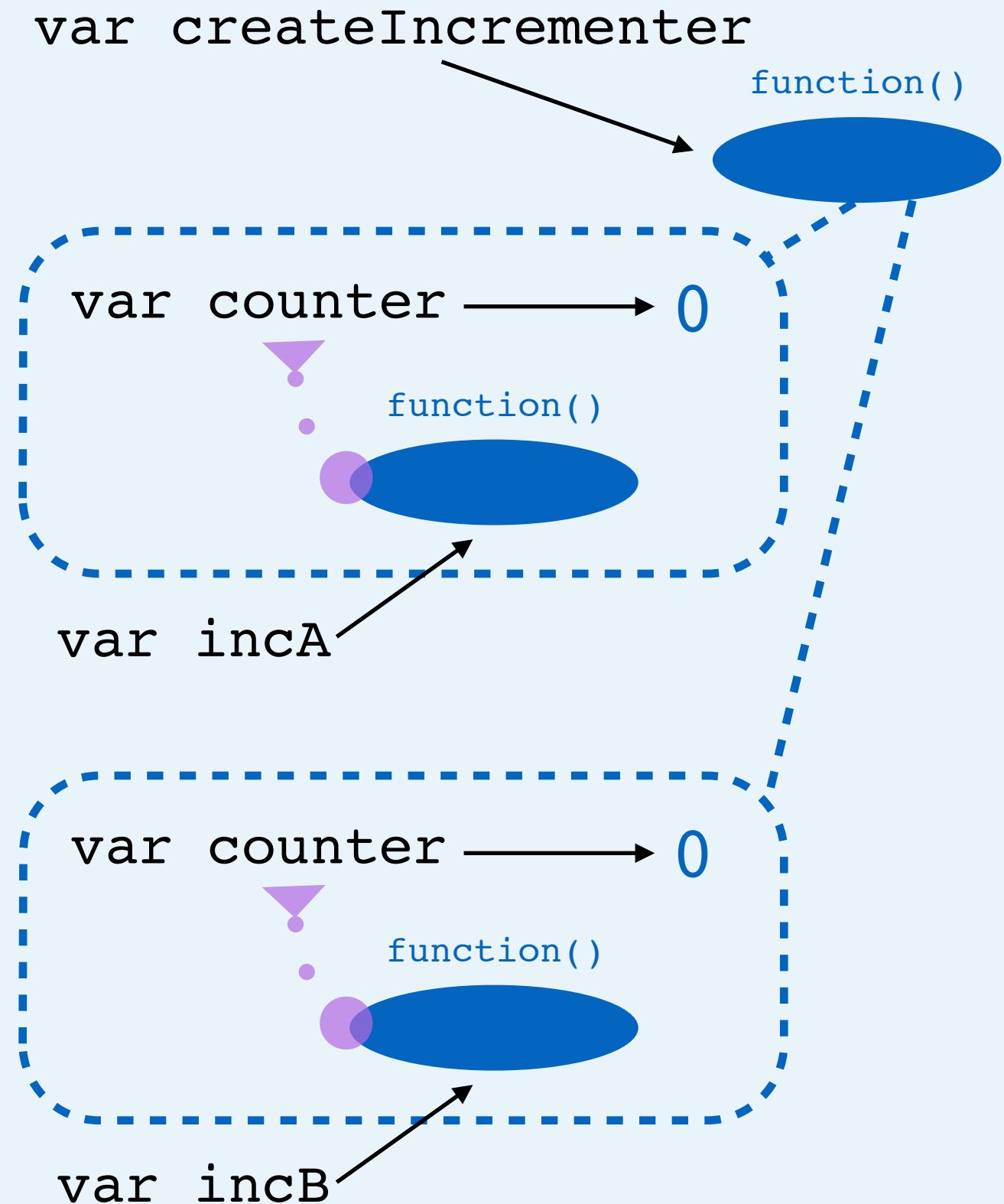
- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.
- b. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.
- b. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.



# Multiple Closure Scopes

```
var createIncrementer = function () {  
  var counter = 0;  
  return function () {  
    counter = counter + 1;  
    return counter;  
  }  
}  
var incA = createIncrementer();  
var incB = createIncrementer();
```

- a. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.
- b. Assignment
  - a. (Evaluate right side) Call function
    - a. Create Scope
    - b. Etc.

## Exercise

```
incA();  
incA();  
incB();
```

How do these function calls update the diagram to the right?

