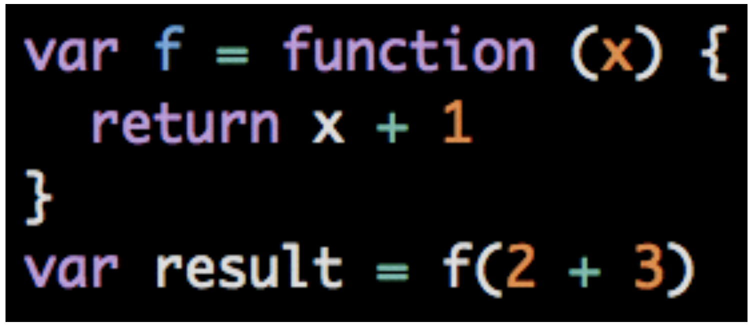
JSDN

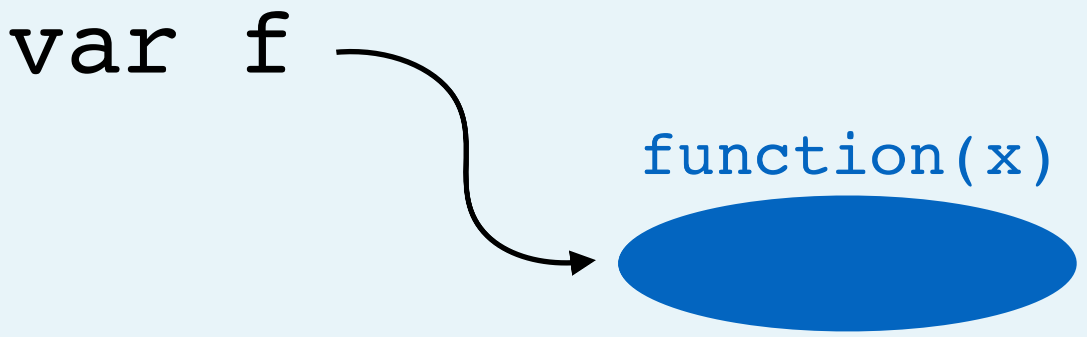
(by example, part 2)

[more]
Function Calls

Arguments Resolve First



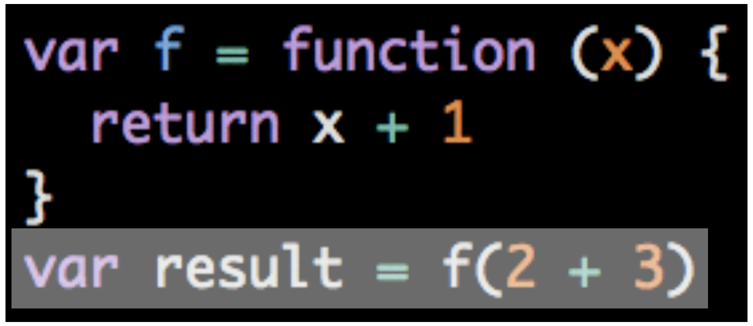
```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```



var f

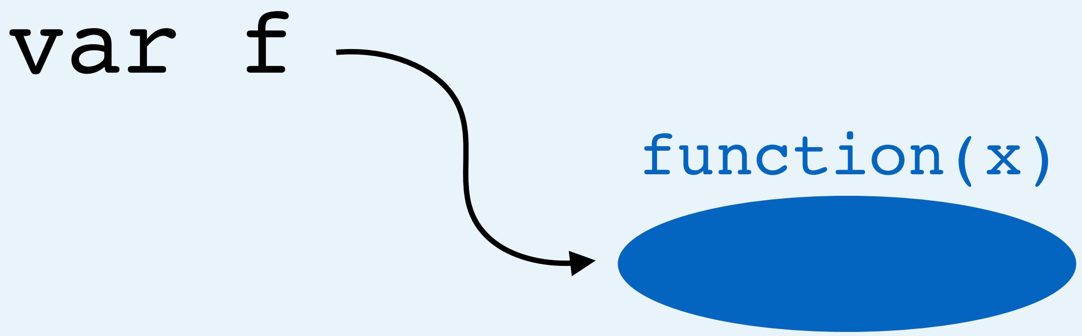
function(x)

Arguments Resolve First



```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

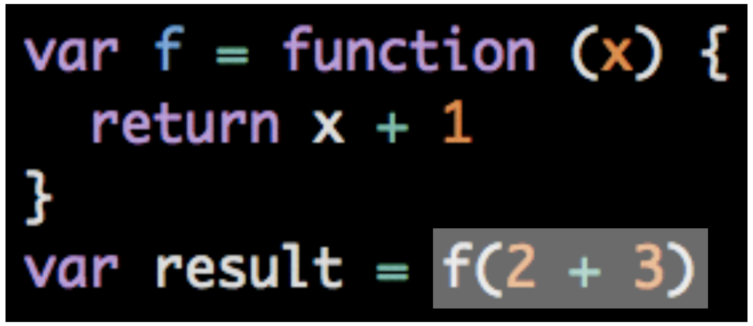
a. Assignment



var f

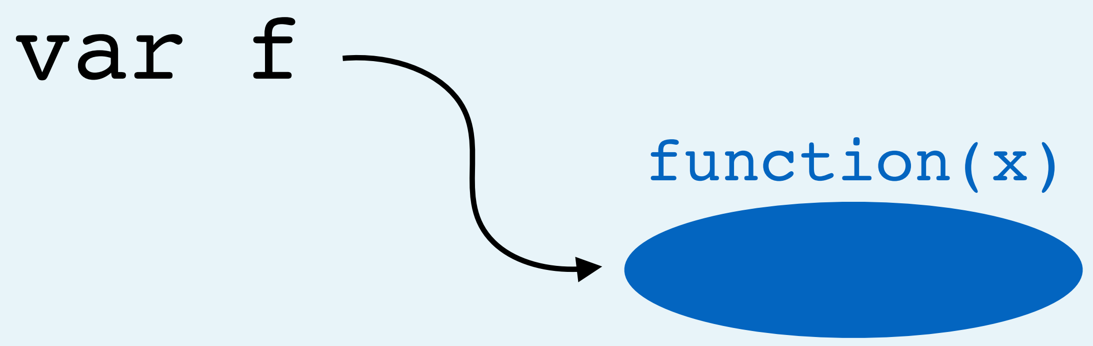
function(x)

Arguments Resolve First



```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

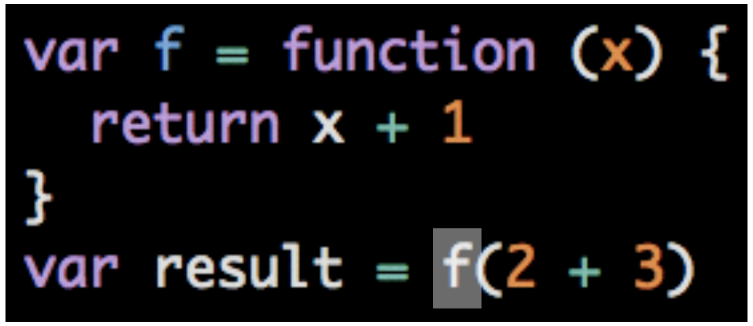
a. Assignment
    a. Evaluate right side



var f

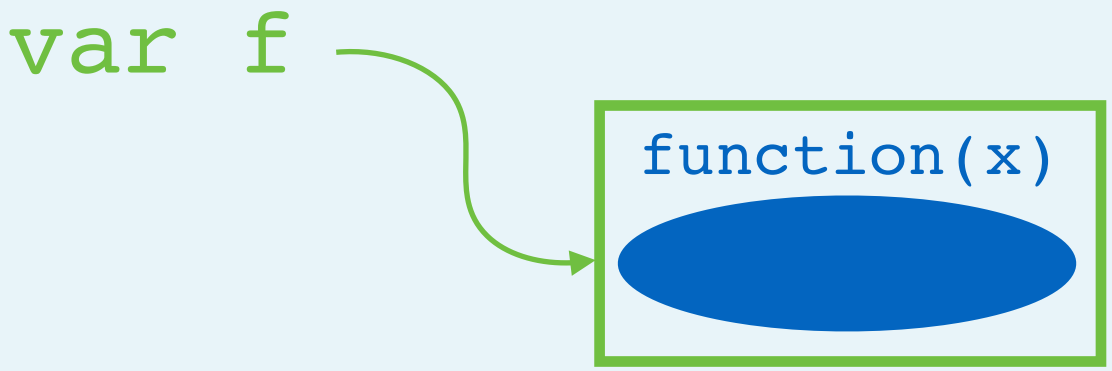
function(x)

Arguments Resolve First

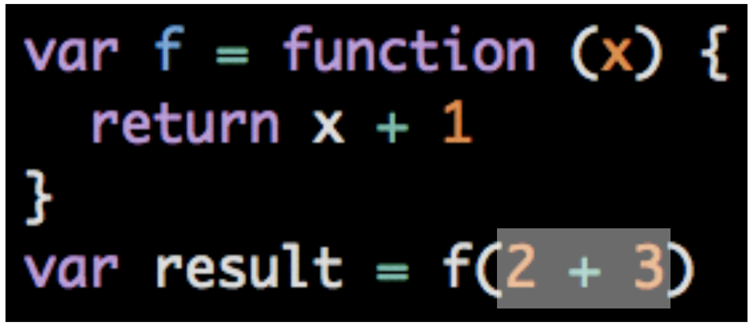


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it's a function!)

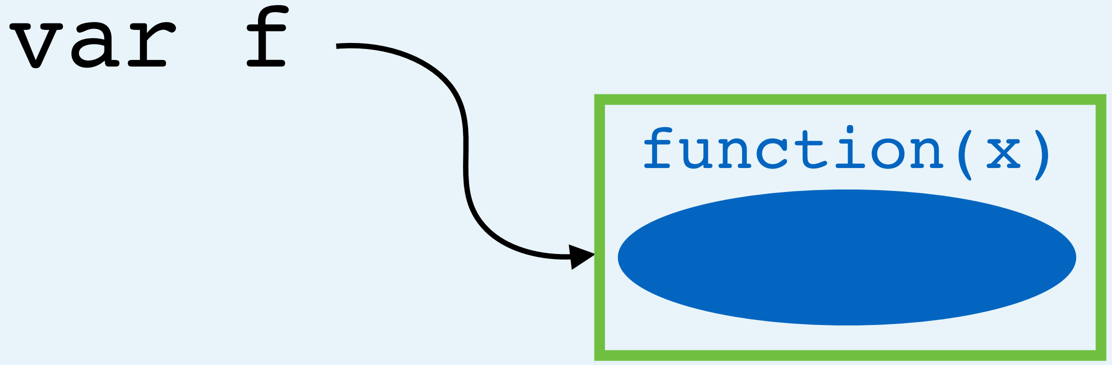


Arguments Resolve First

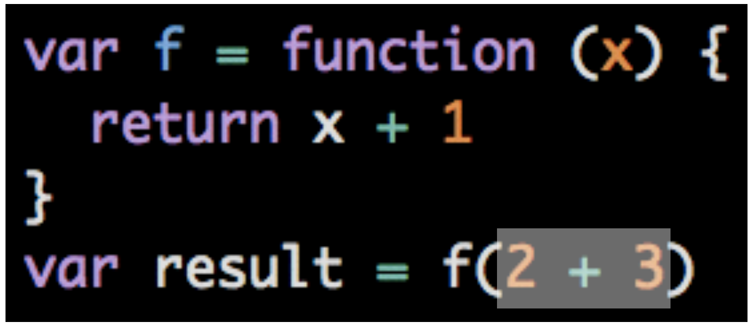


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it's a function!)
    c. Resolve argument

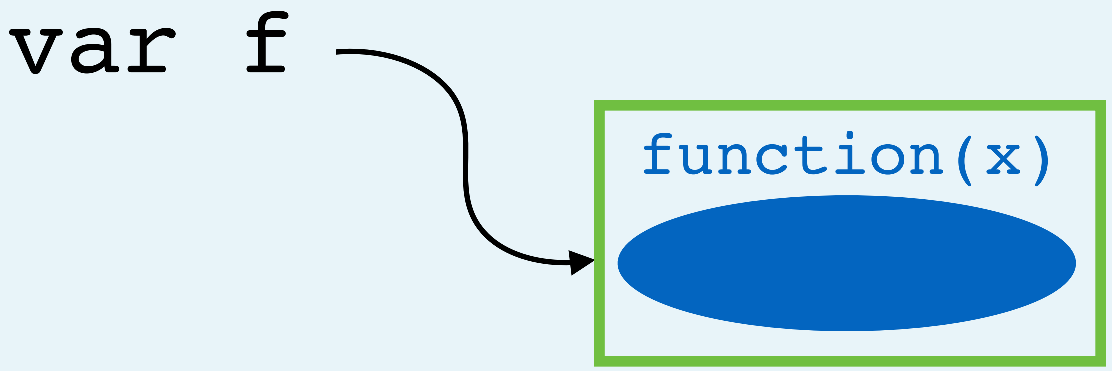


Arguments Resolve First

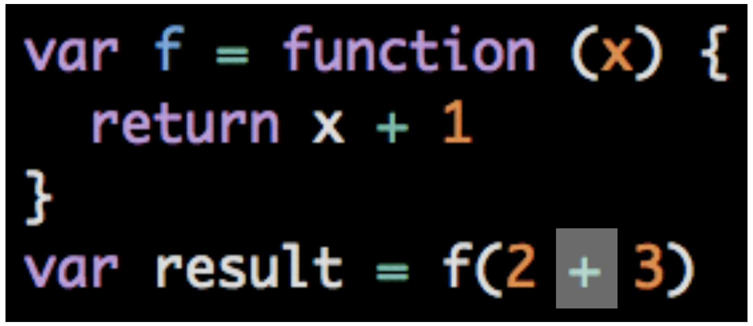


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
   a. Evaluate right side
   b. Look up value of f (it’s a function!)
   c. Resolve argument
      a. Binary Operation (addition)

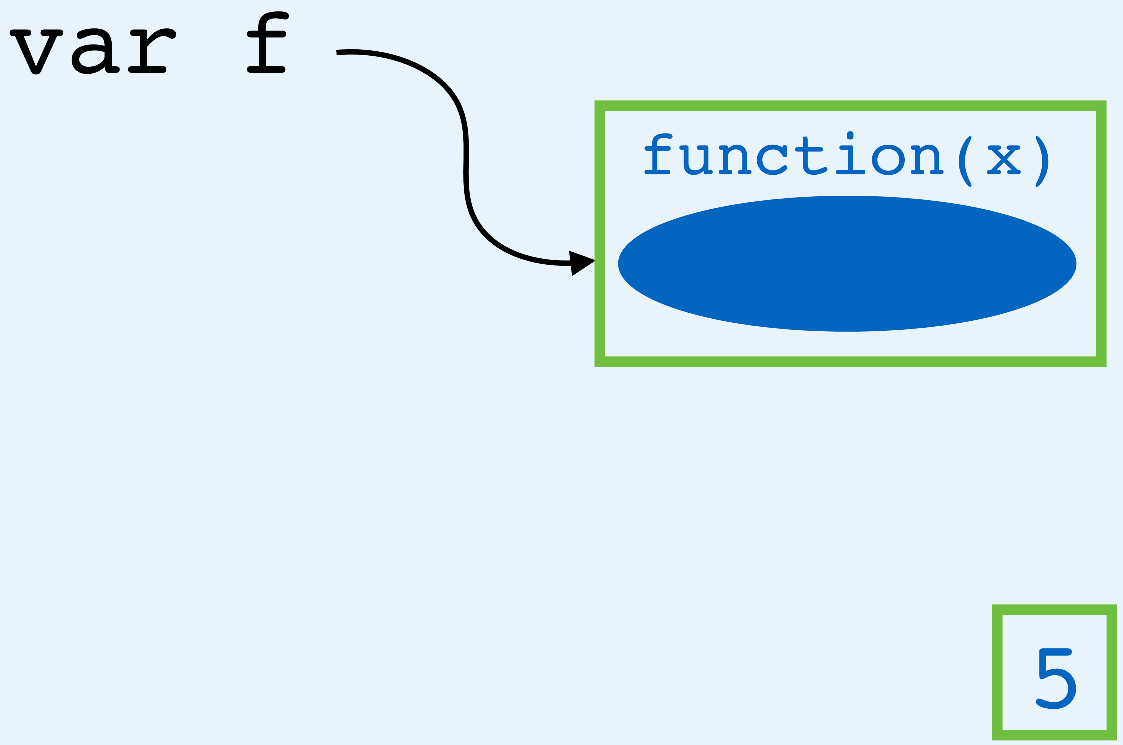


Arguments Resolve First

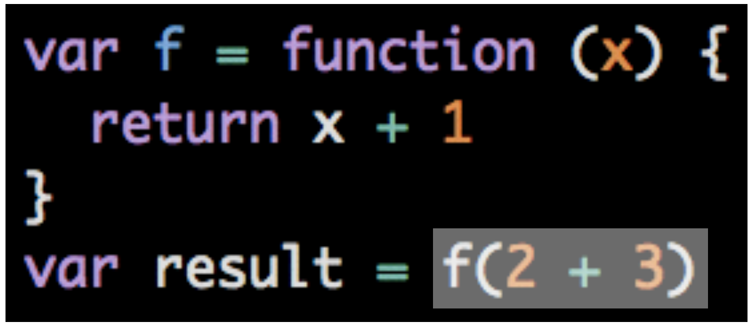


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

- a. Assignment
    - a. Evaluate right side
    - b. Look up value of f (it’s a function!)
    - c. Resolve argument
        - a. Binary Operation (addition)
            - a. Create value

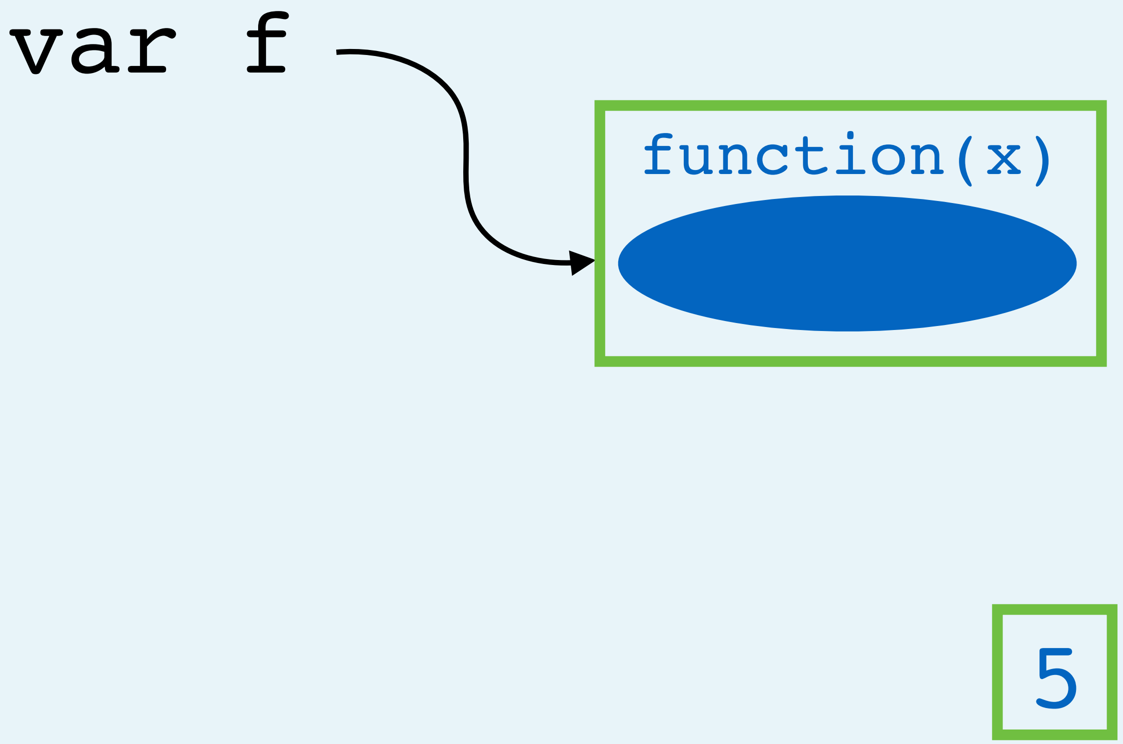


Arguments Resolve First

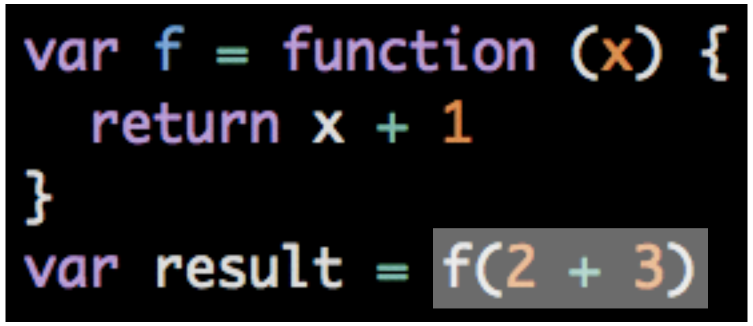


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it’s a function!)
    c. Resolve argument
        a. Binary Operation (addition)
            a. Create value
    d. Call function

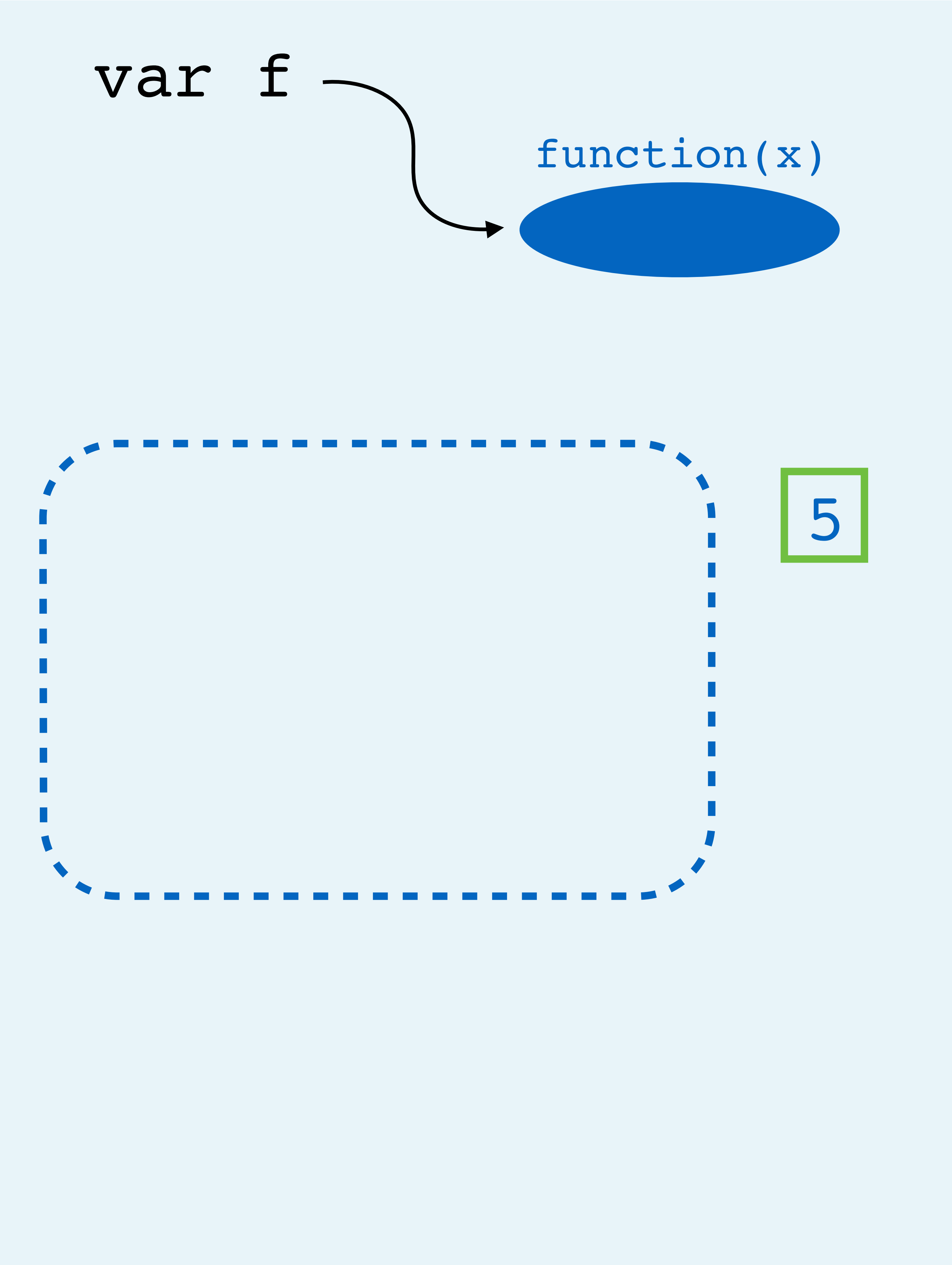


Arguments Resolve First

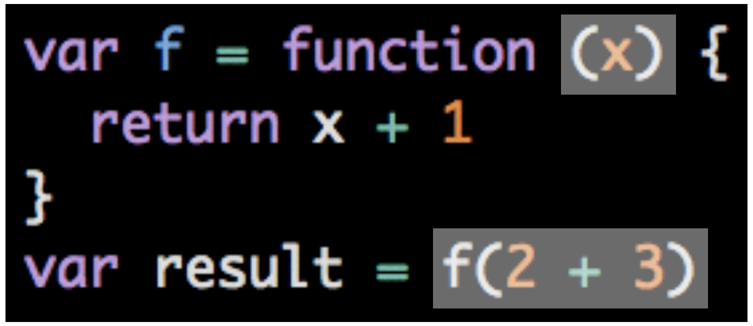


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it's a function!)
    c. Resolve argument
        a. Binary Operation (addition)
            a. Create value
    d. Call function
        a. Create scope

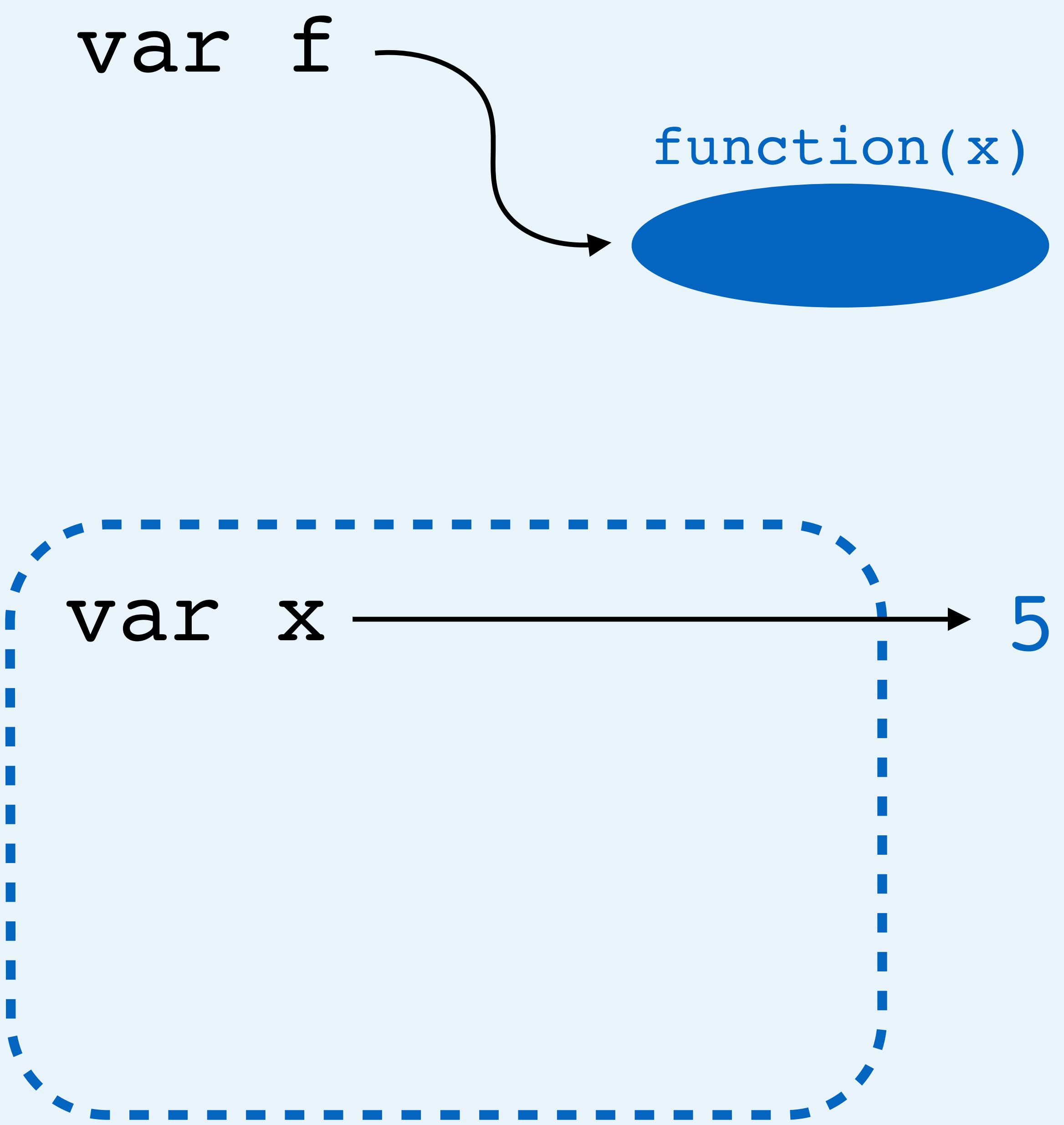


Arguments Resolve First

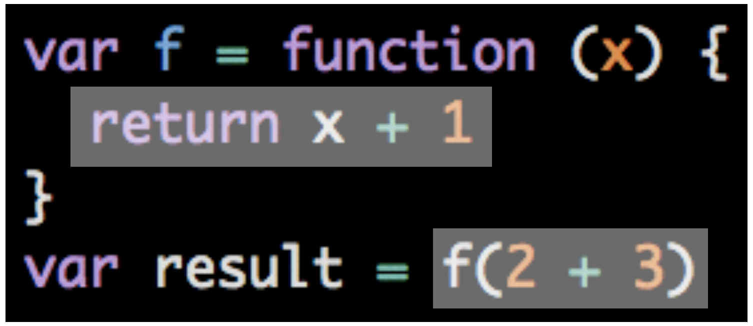


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it’s a function!)
    c. Resolve argument
        a. Binary Operation (addition)
            a. Create value
    d. Call function
        a. Create scope
        b. Create parameter(s)

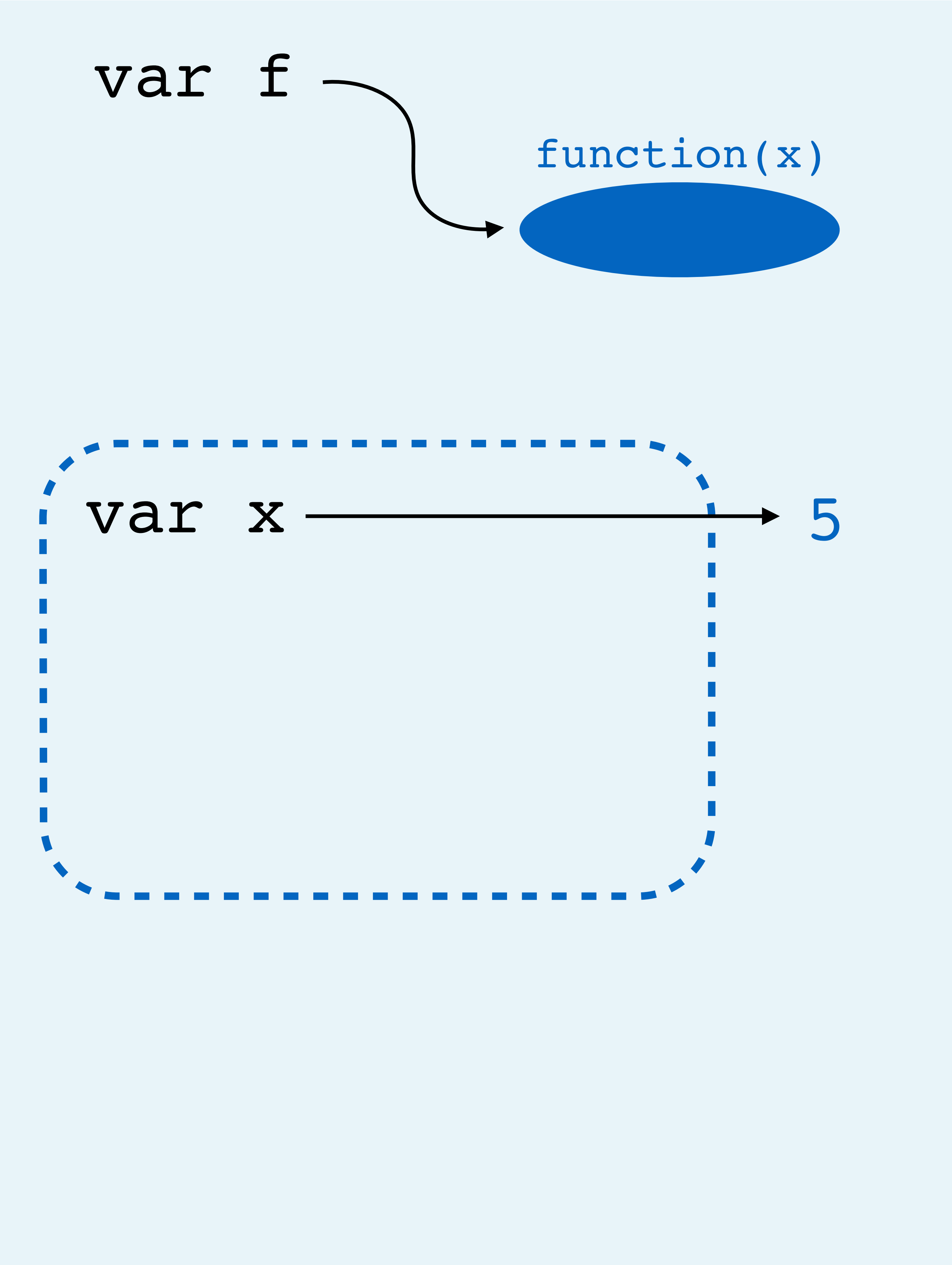


Arguments Resolve First

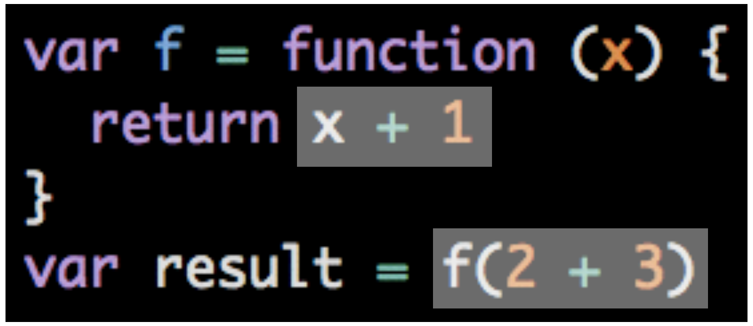


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

- a. Assignment
    - a. Evaluate right side
    - b. Look up value of f (it’s a function!)
    - c. Resolve argument
        - a. Binary Operation (addition)
            - a. Create value
    - d. Call function
        - a. Create scope
        - b. Create parameter(s)
        - c. Return statement

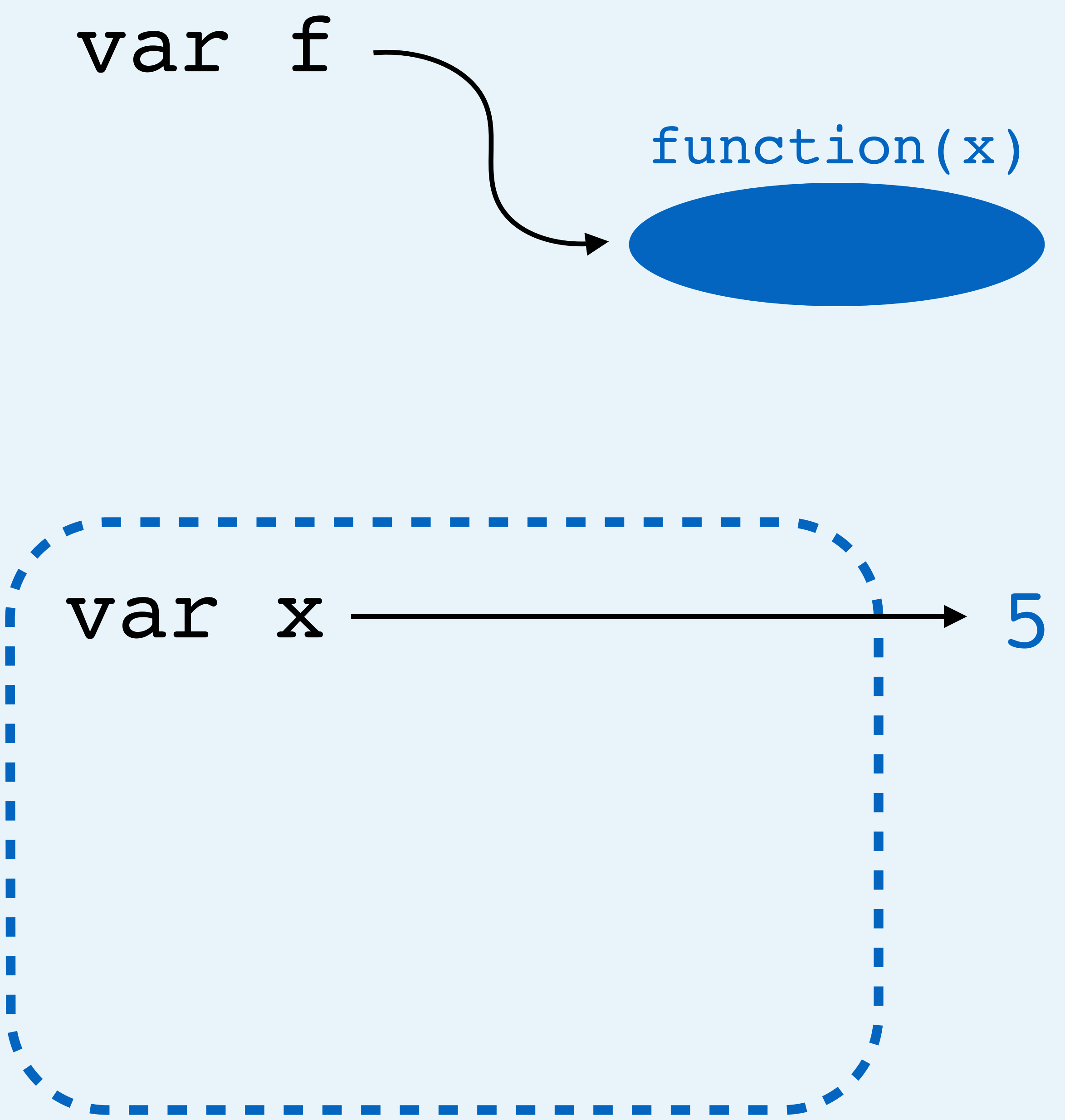


Arguments Resolve First

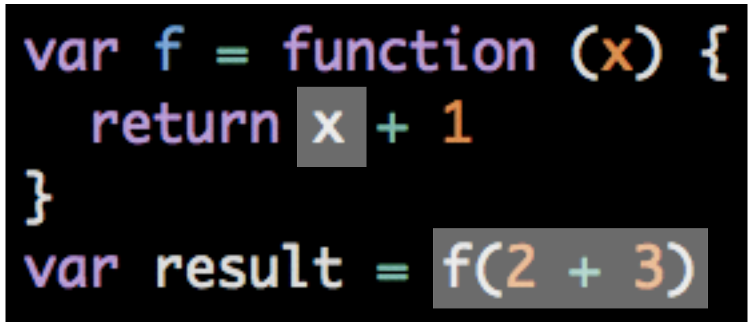


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it’s a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)

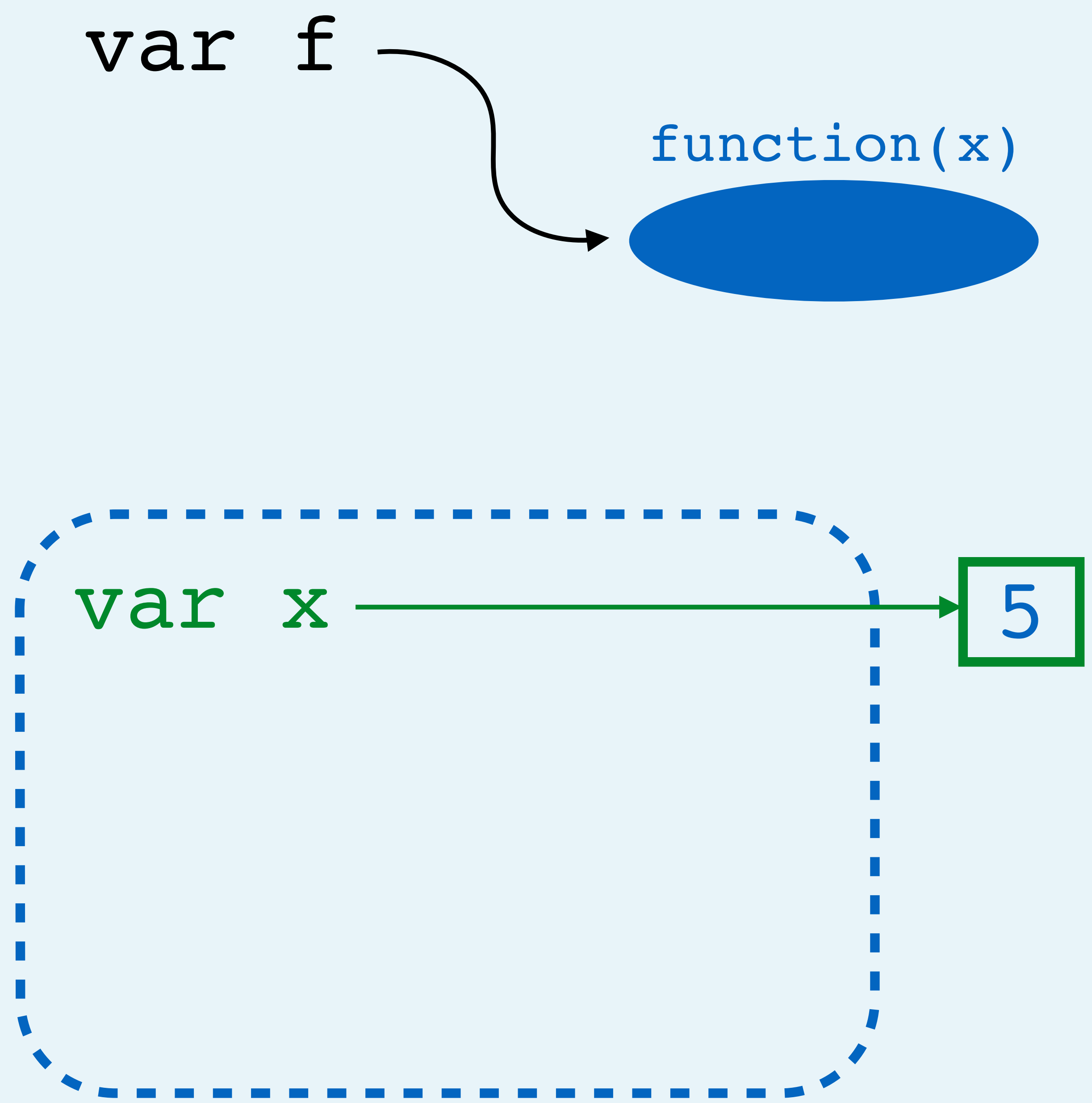


Arguments Resolve First

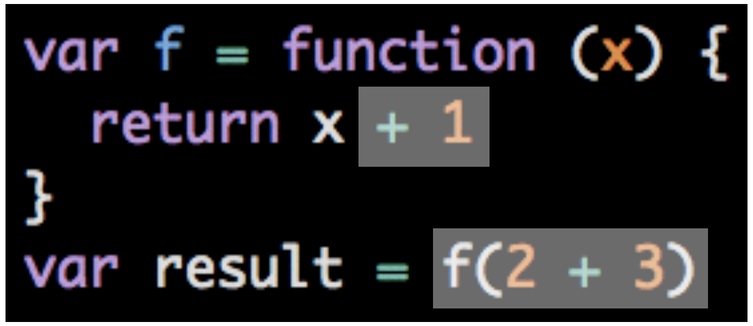


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it’s a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)
        - a. Look up value of x

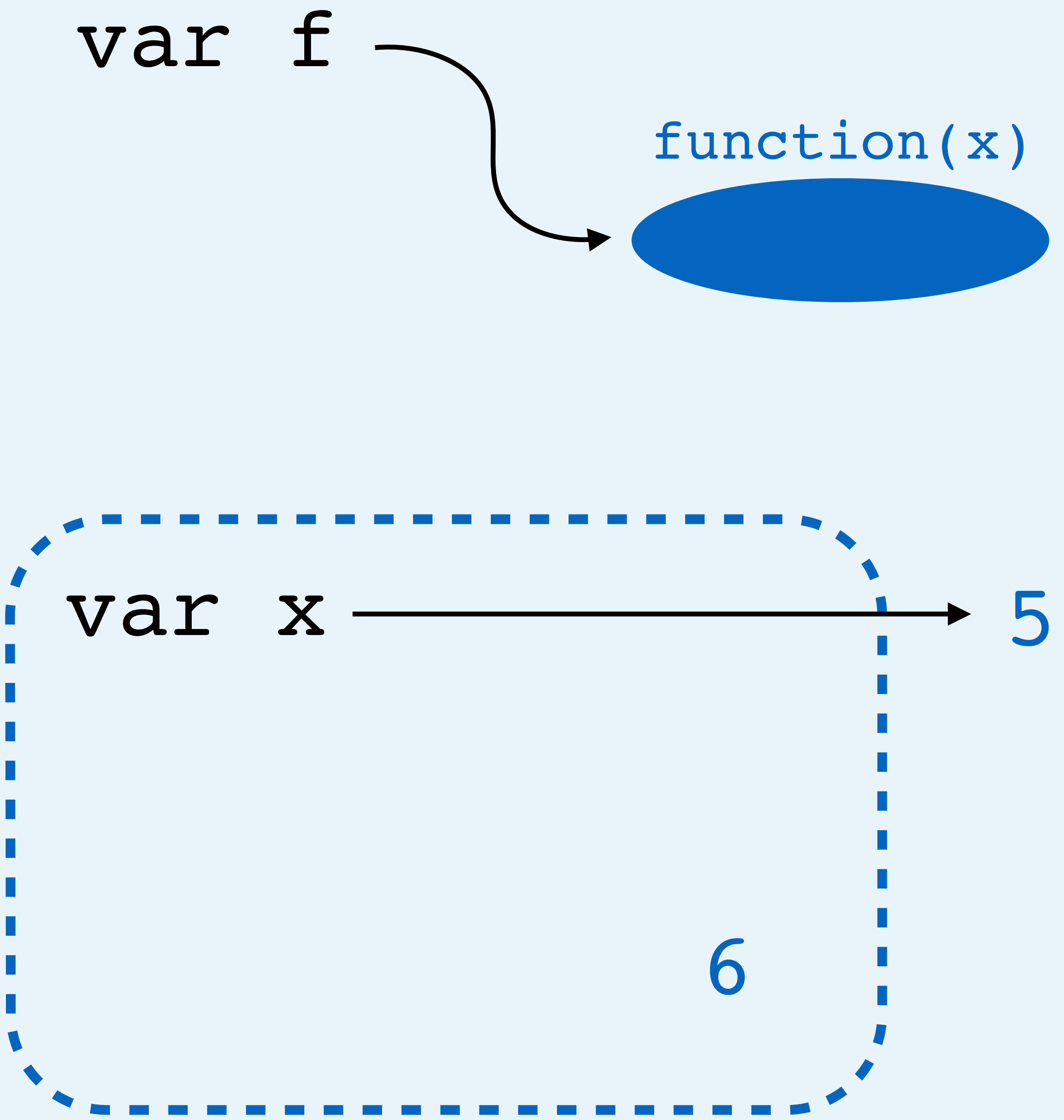


Arguments Resolve First

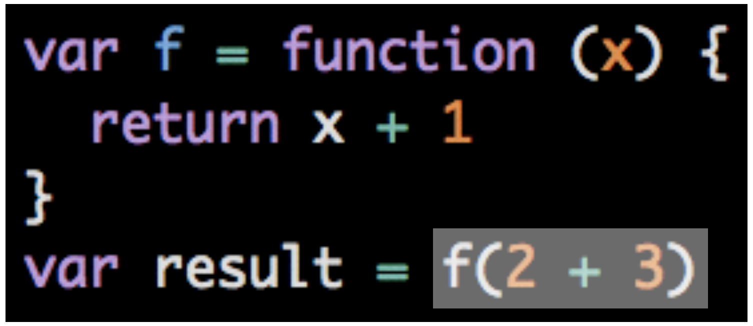


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

1. Assignment
   1. Evaluate right side
   2. Look up value of f (it’s a function!)
   3. Resolve argument
      1. Binary Operation (addition)
         1. Create value
   4. Call function
      1. Create scope
      2. Create parameter(s)
      3. Return statement
         1. Binary Operation (addition)
            1. Look up value of x
            2. Create value

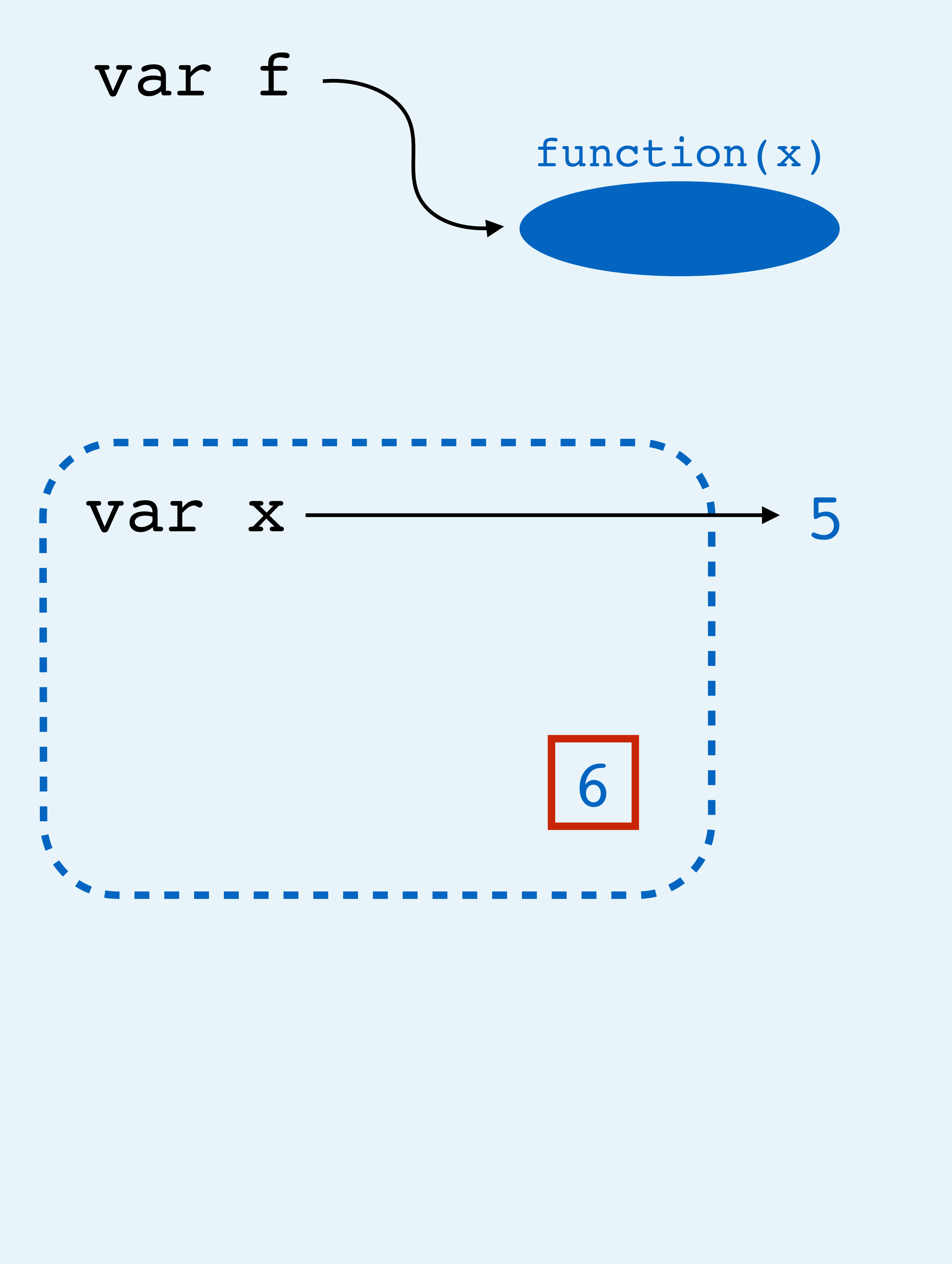


Arguments Resolve First

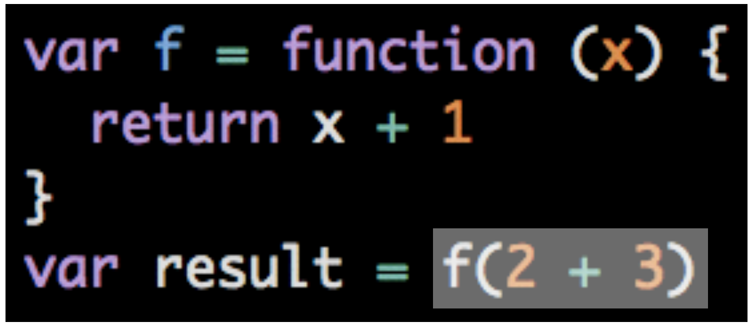


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it’s a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)
        - a. Look up value of x
        - b. Create value
      - b. Mark as return value

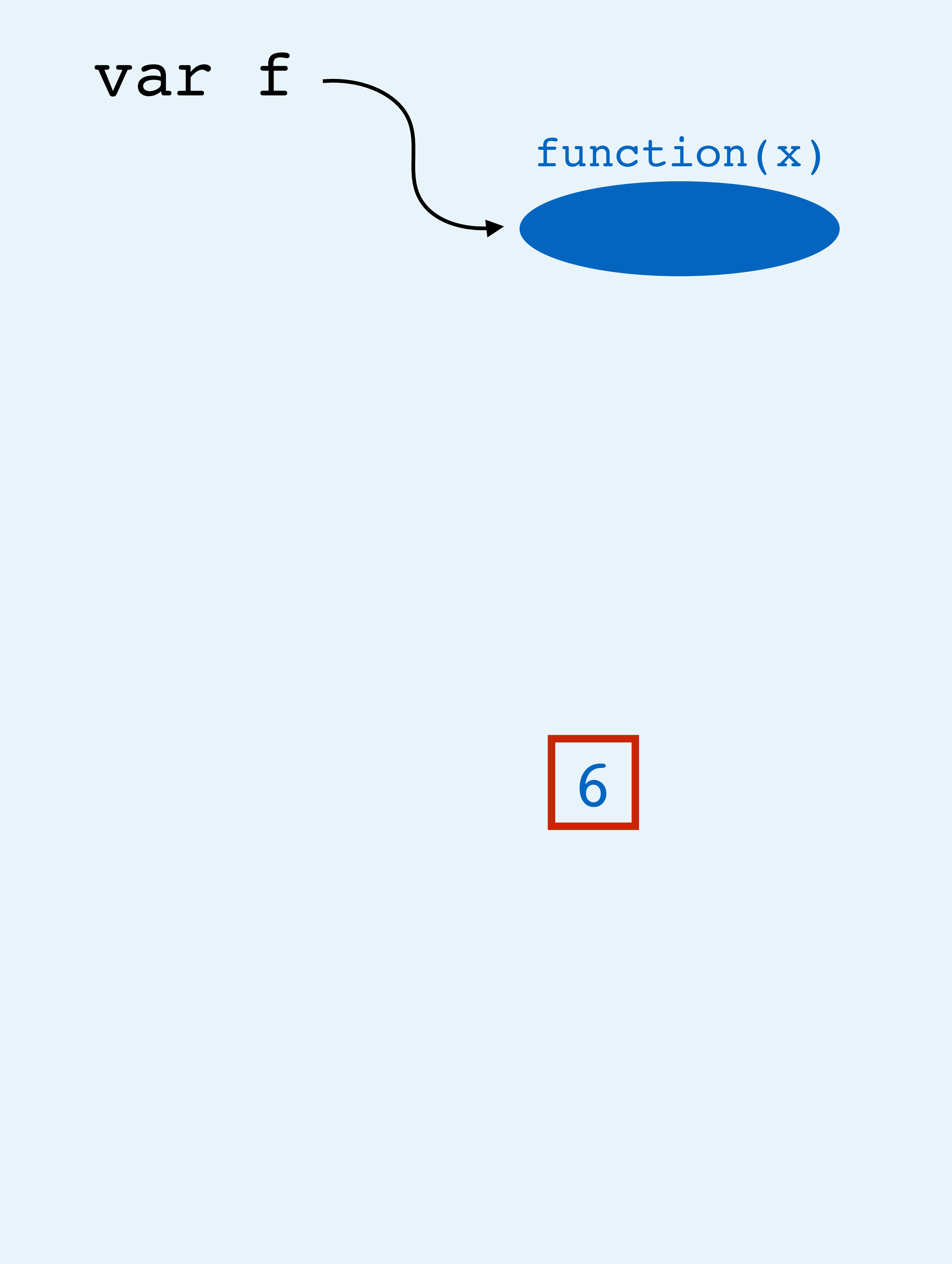


Arguments Resolve First

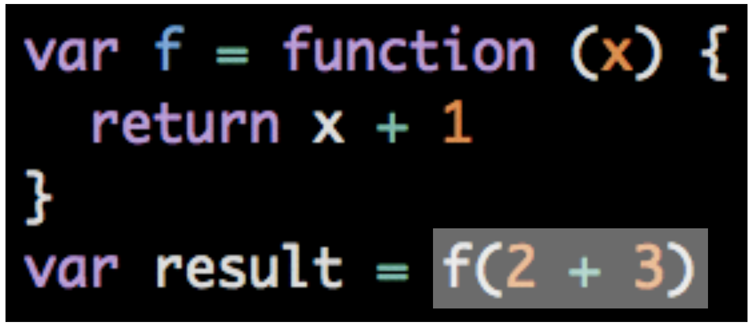


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

a. Assignment
    a. Evaluate right side
    b. Look up value of f (it's a function!)
    c. Resolve argument
        a. Binary Operation (addition)
            a. Create value
    d. Call function
        a. Create scope
        b. Create parameter(s)
        c. Return statement
            a. Binary Operation (addition)
                a. Look up value of x
                b. Create value
            b. Mark as return value
        d. Garbage collect scope

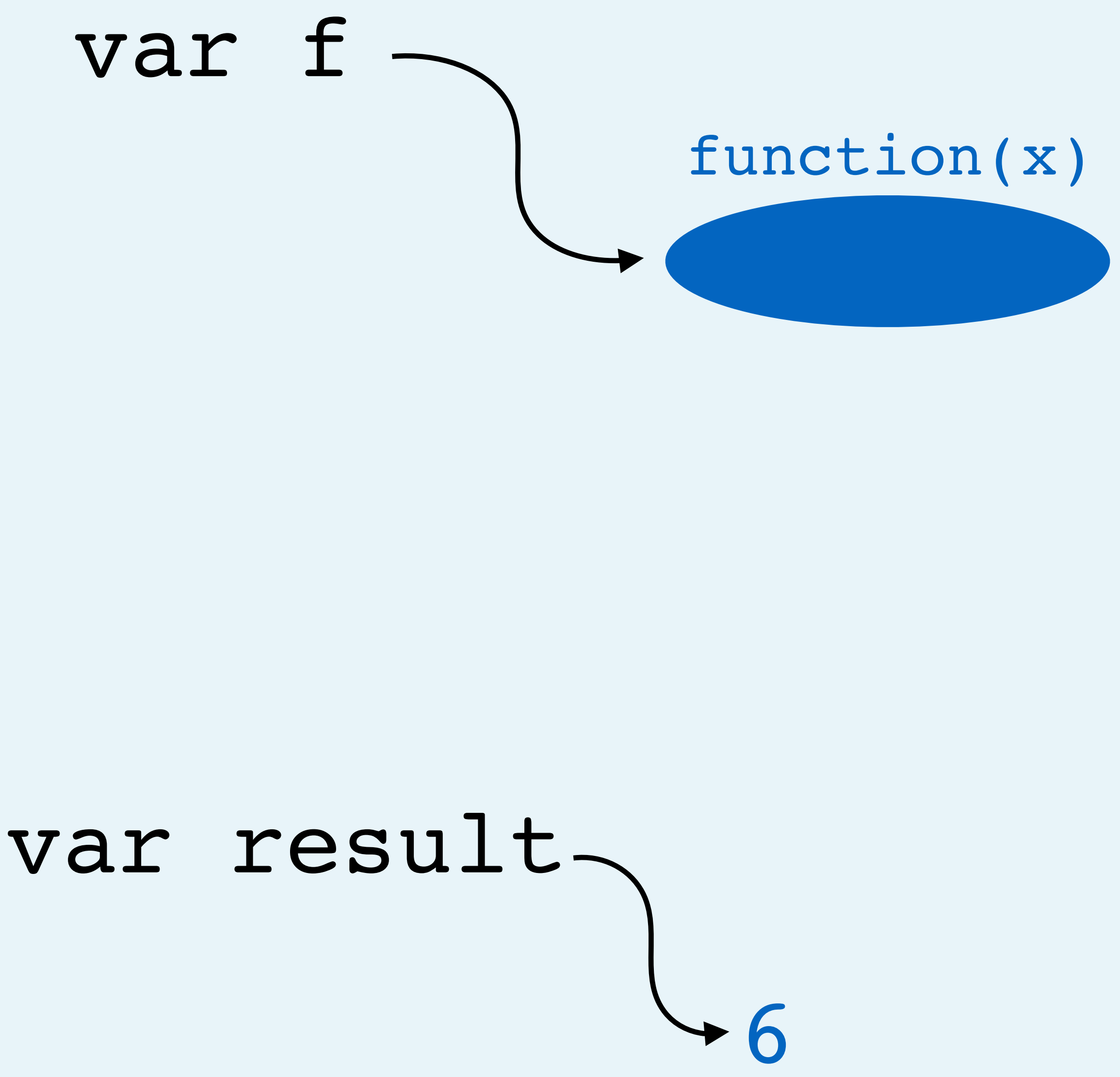


Arguments Resolve First

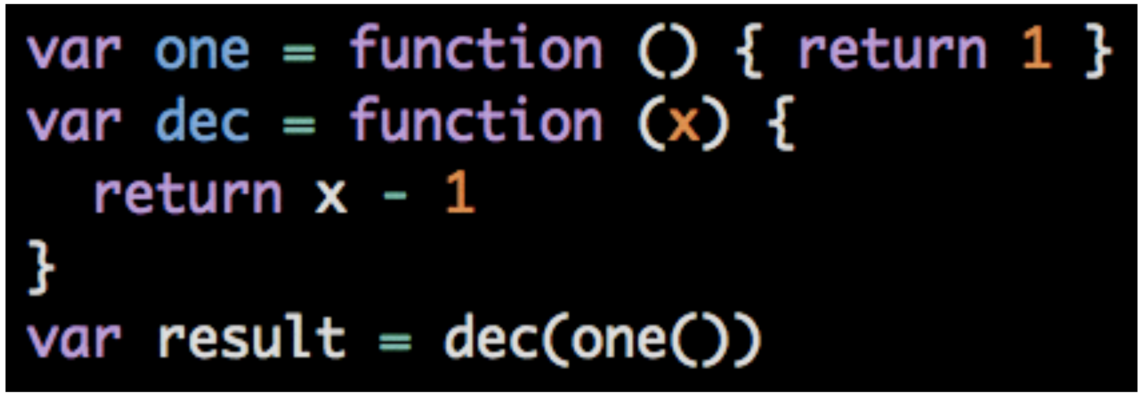


```
var f = function (x) {
  return x + 1
}
var result = f(2 + 3)
```

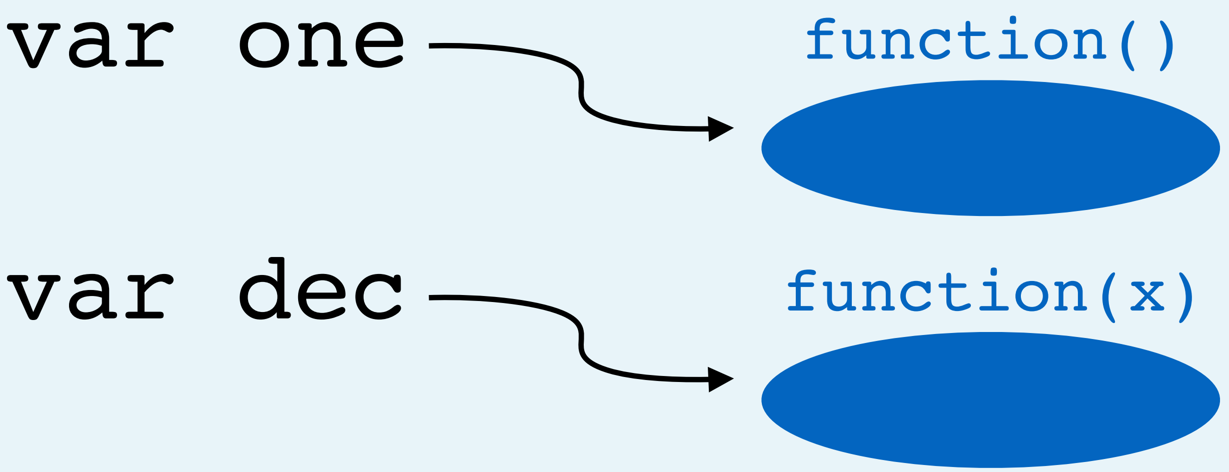
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it’s a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)
        - a. Look up value of x
        - b. Create value
      - b. Mark as return value
    - d. Garbage collect scope
  - e. Create var result, point to value



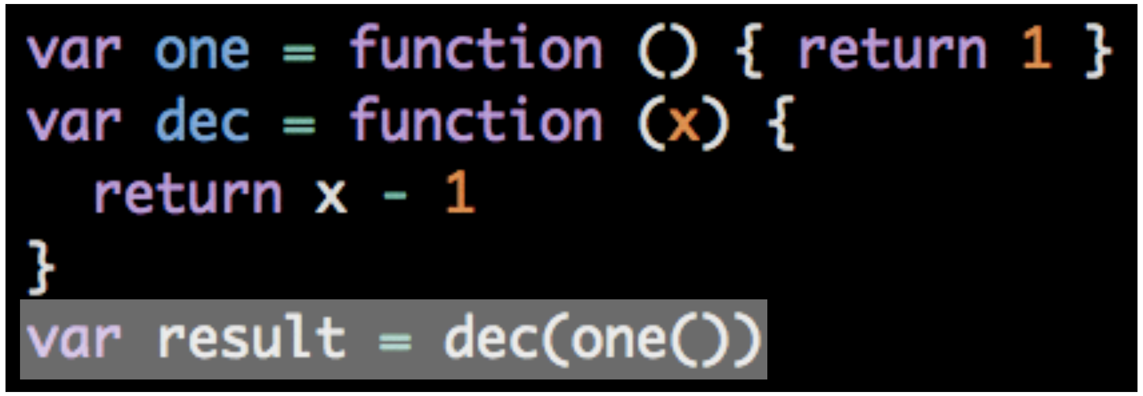
Arguments Resolve First (2)



```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

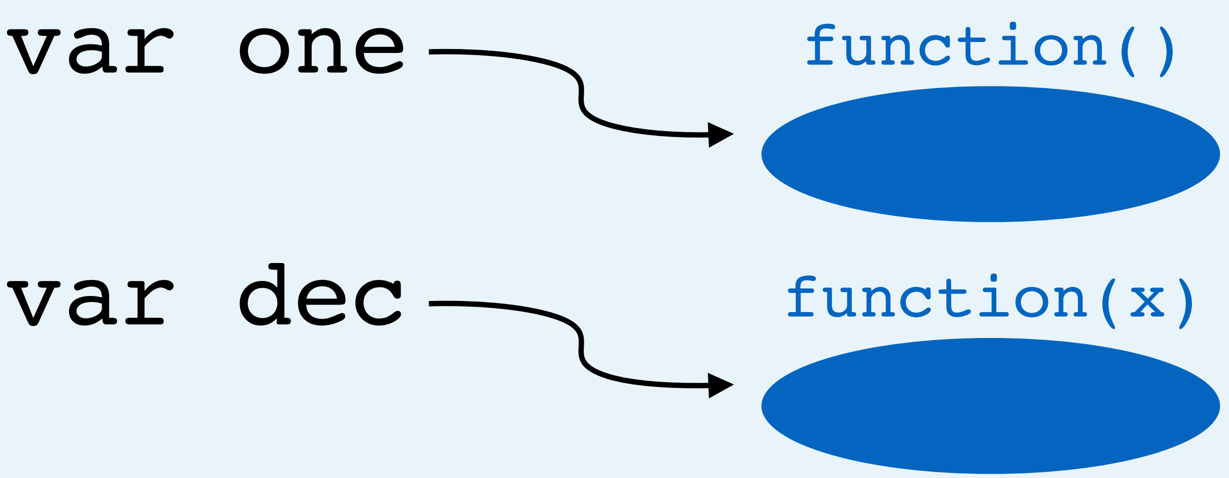


Arguments Resolve First (2)



```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment



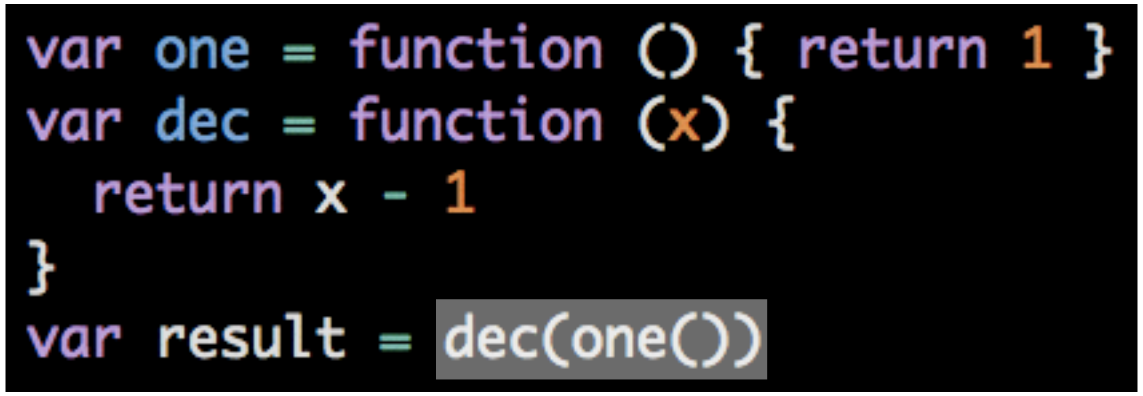
var one

function()

var dec

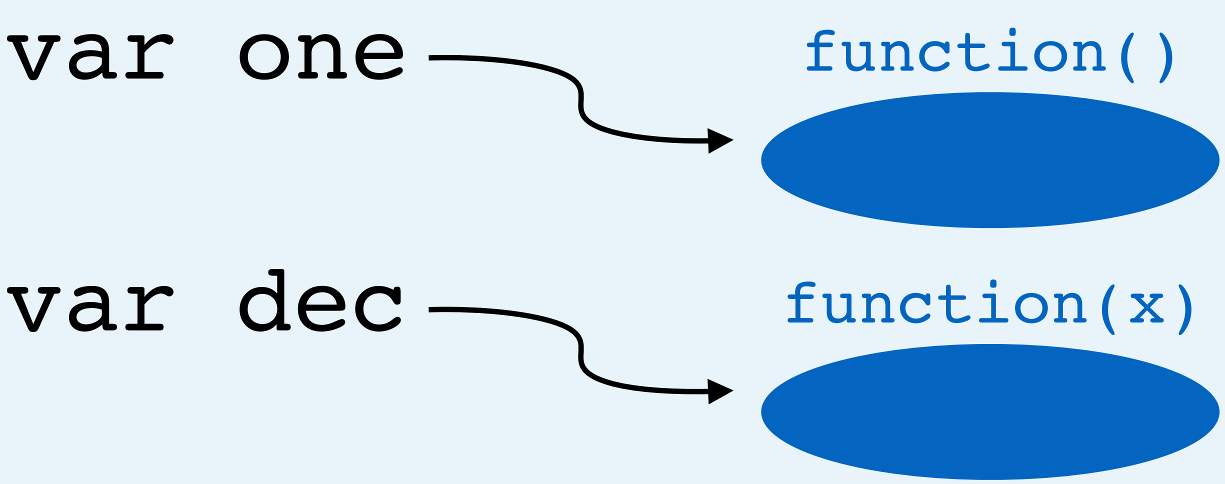
function(x)

Arguments Resolve First (2)

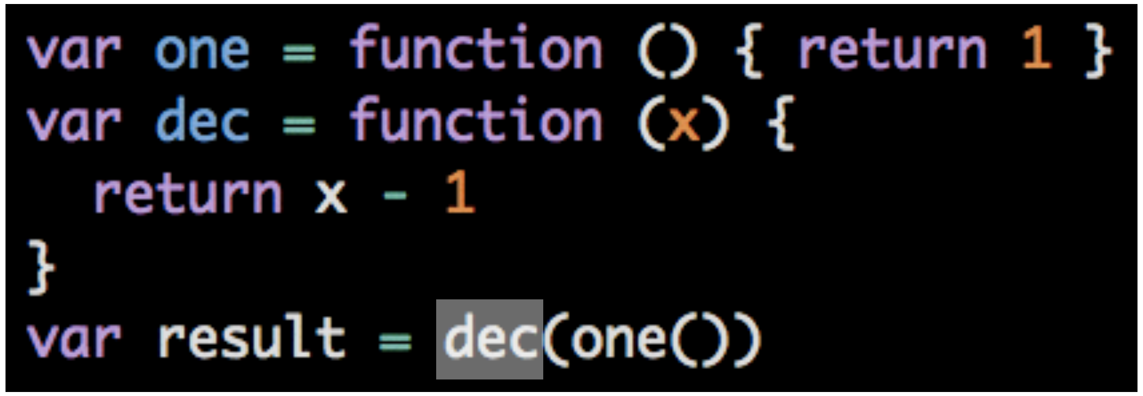


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side

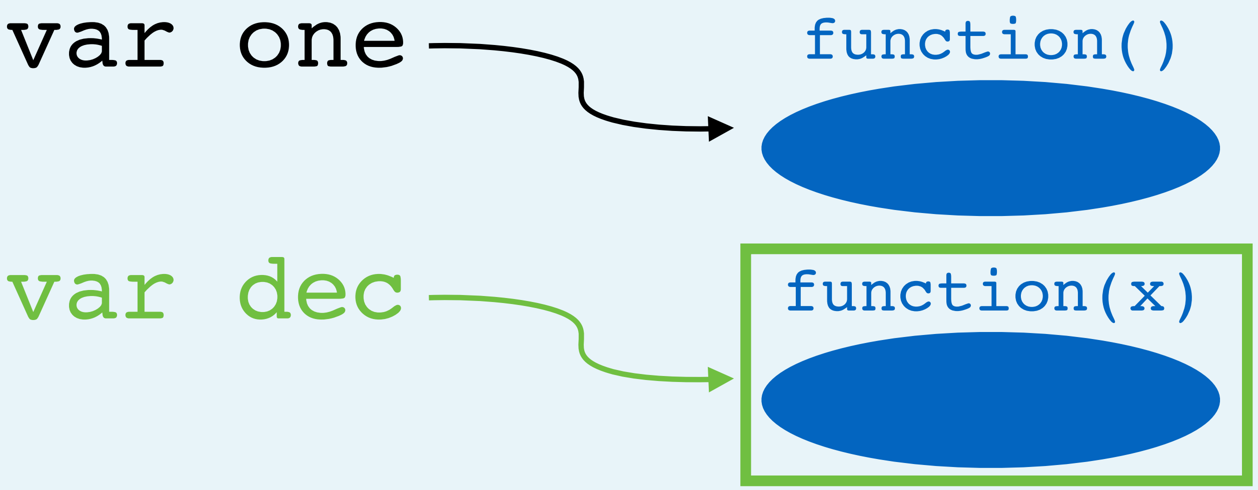


Arguments Resolve First (2)

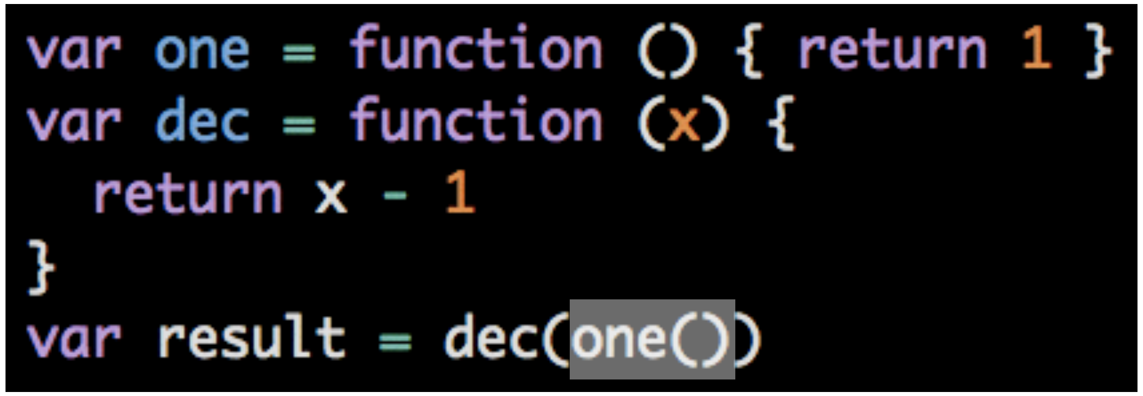


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec

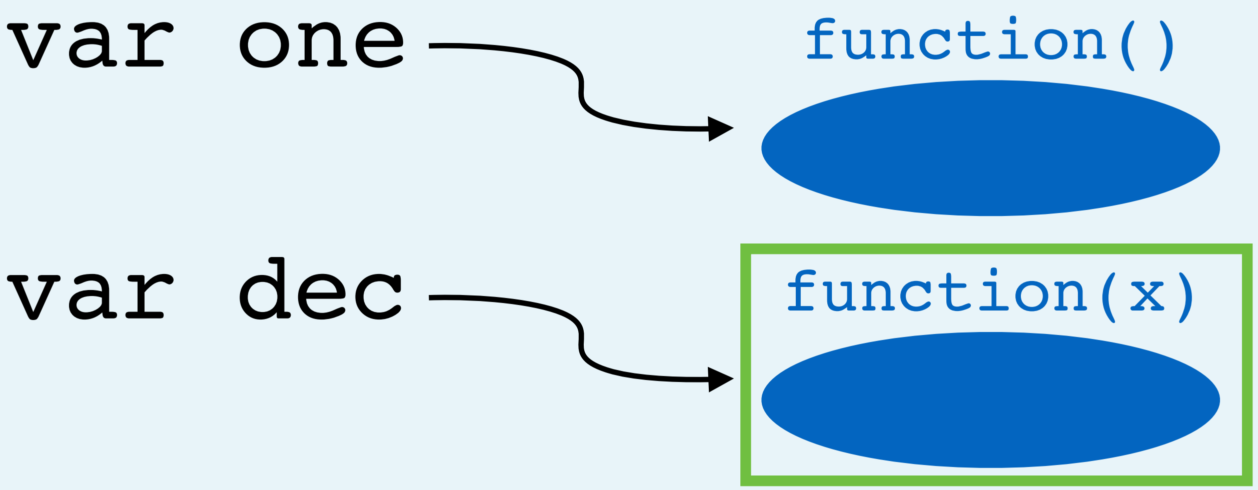


Arguments Resolve First (2)

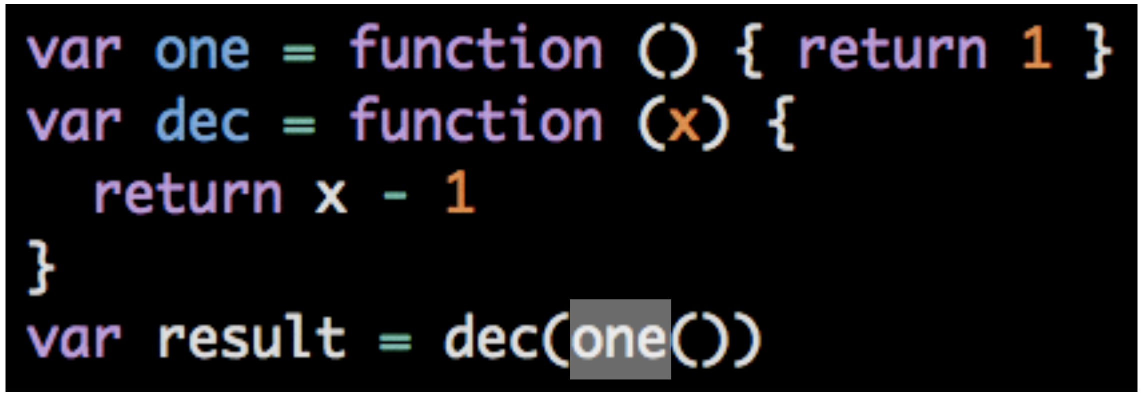


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec
    c. Resolve argument

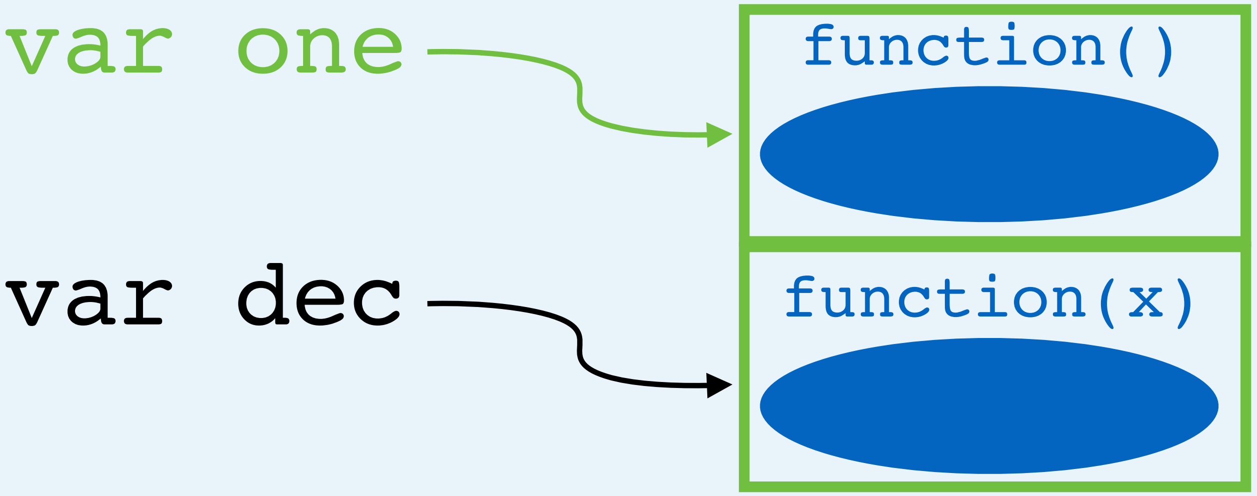


Arguments Resolve First (2)

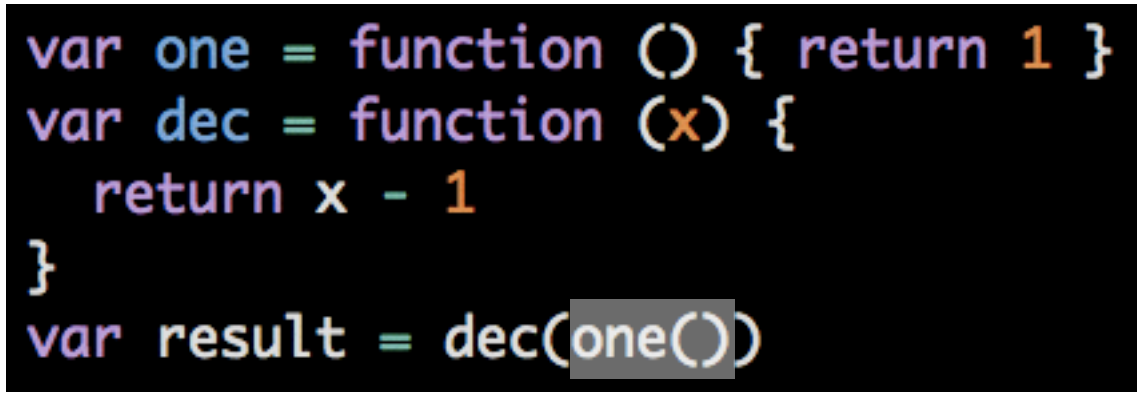


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec
    c. Resolve argument
        a. Look up value of one

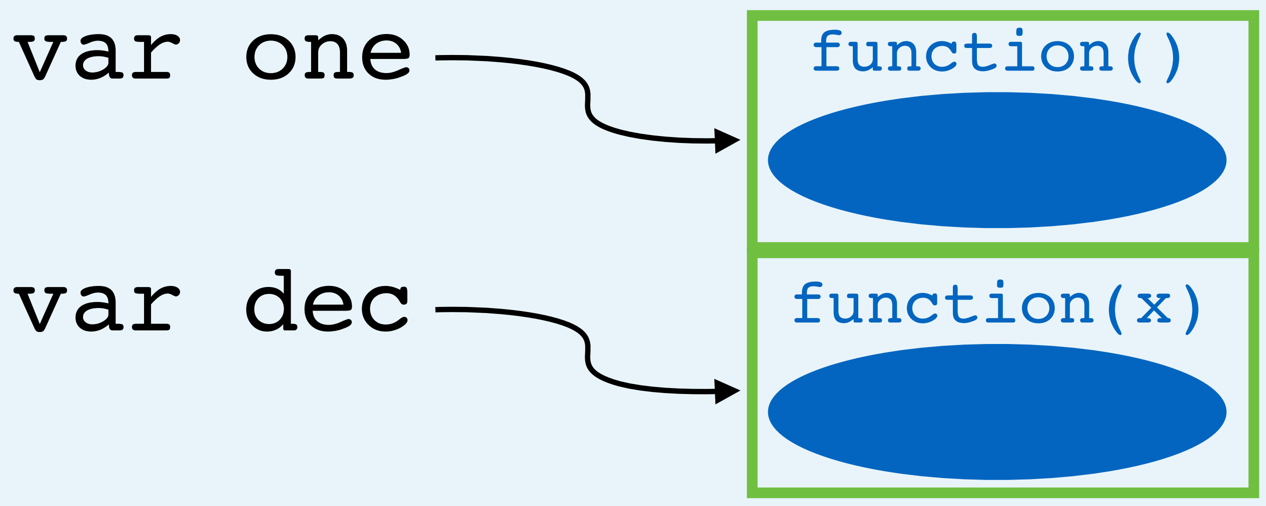


Arguments Resolve First (2)

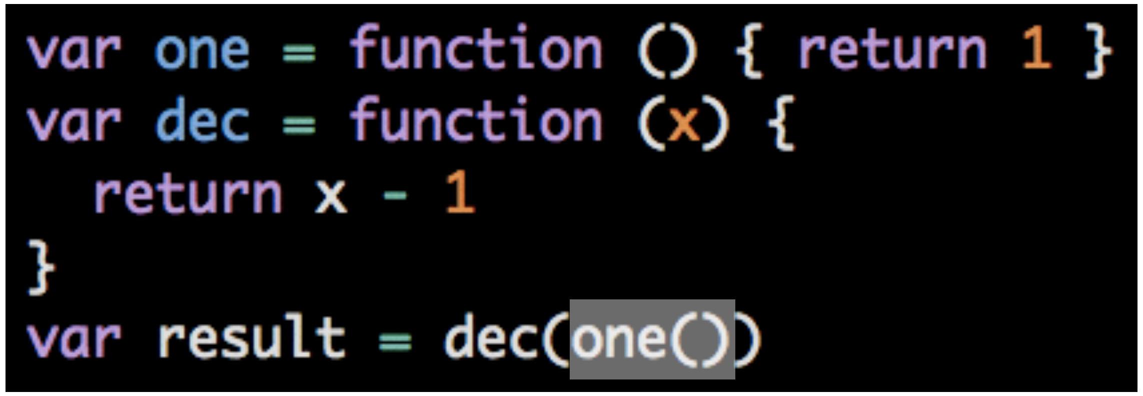


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec
    c. Resolve argument
        a. Look up value of one
        b. Call function

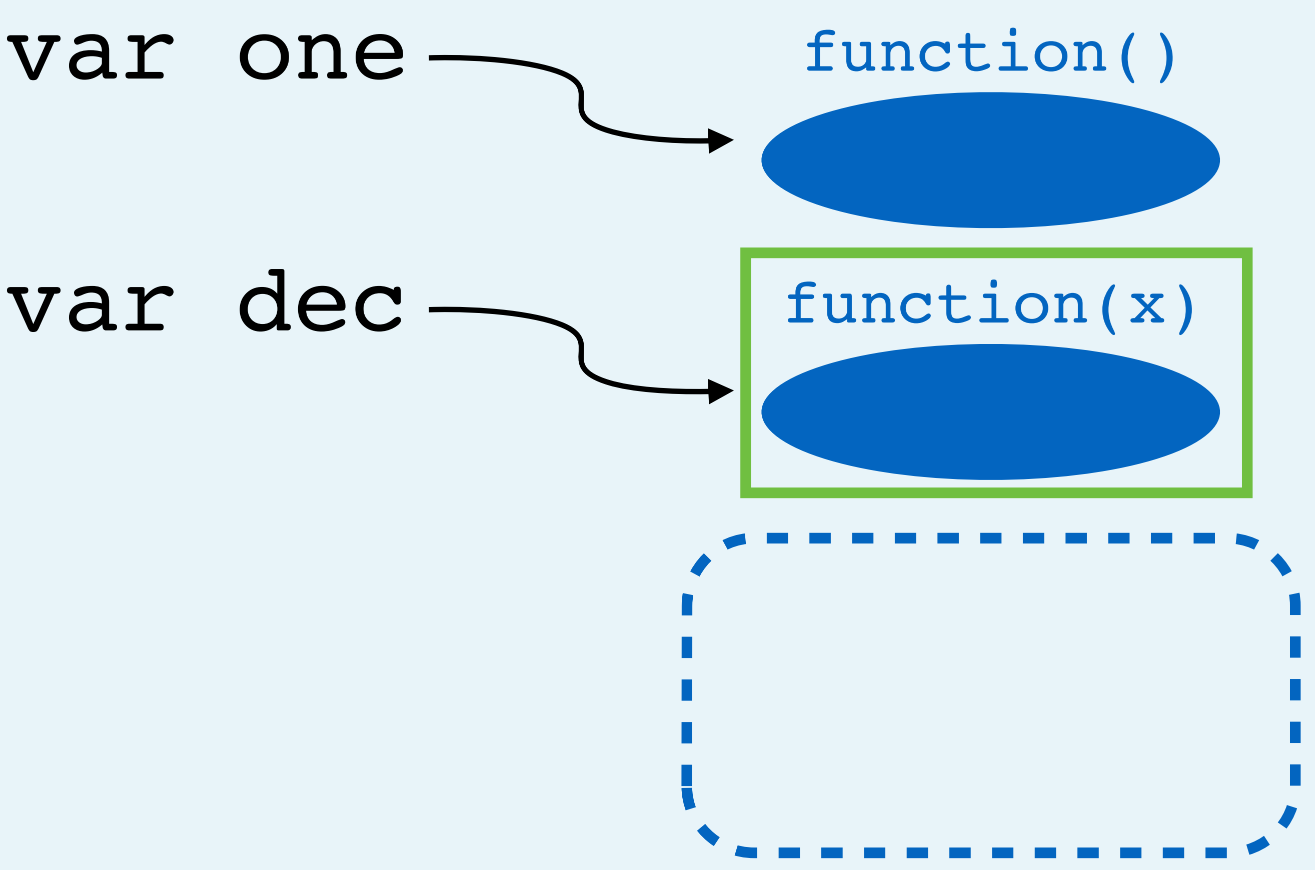


Arguments Resolve First (2)

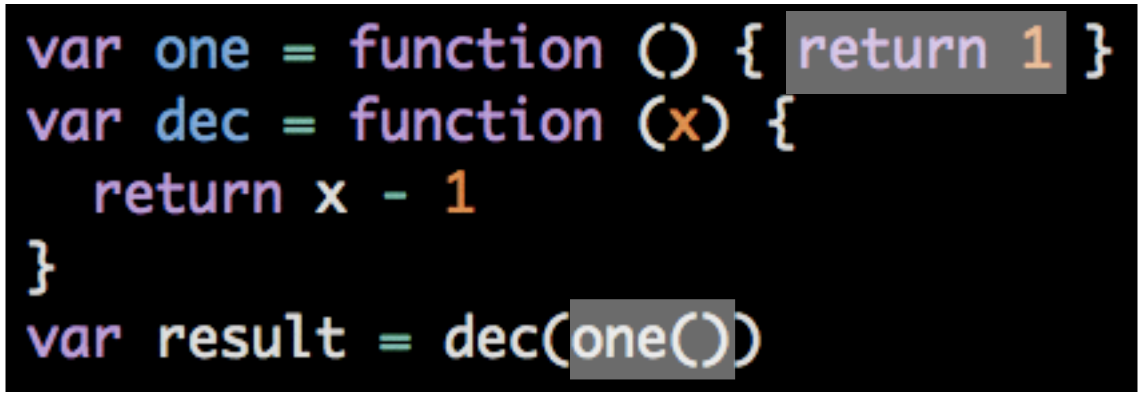


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec
    c. Resolve argument
        a. Look up value of one
        b. Call function
            a. Create scope

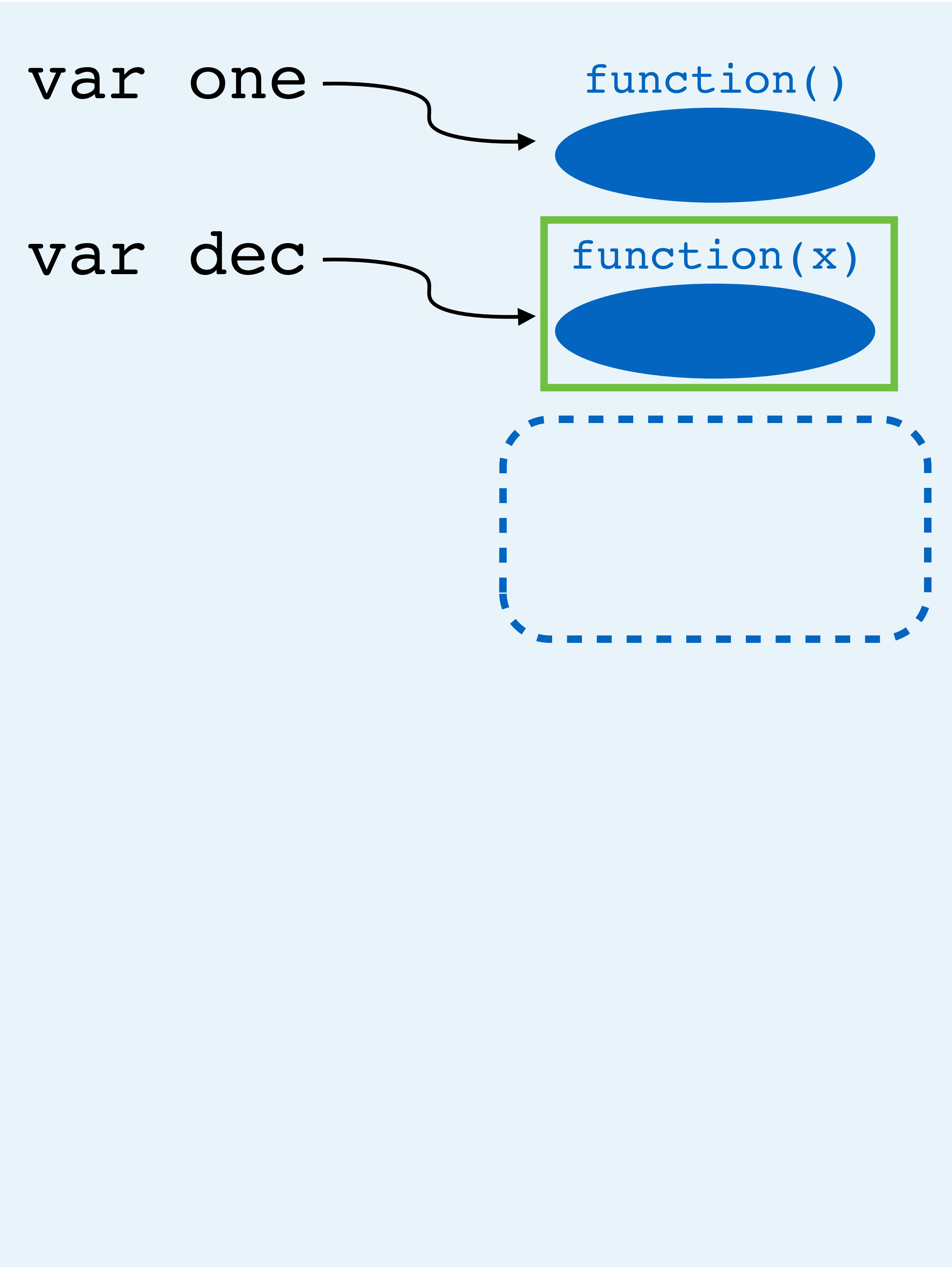


Arguments Resolve First (2)



```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement



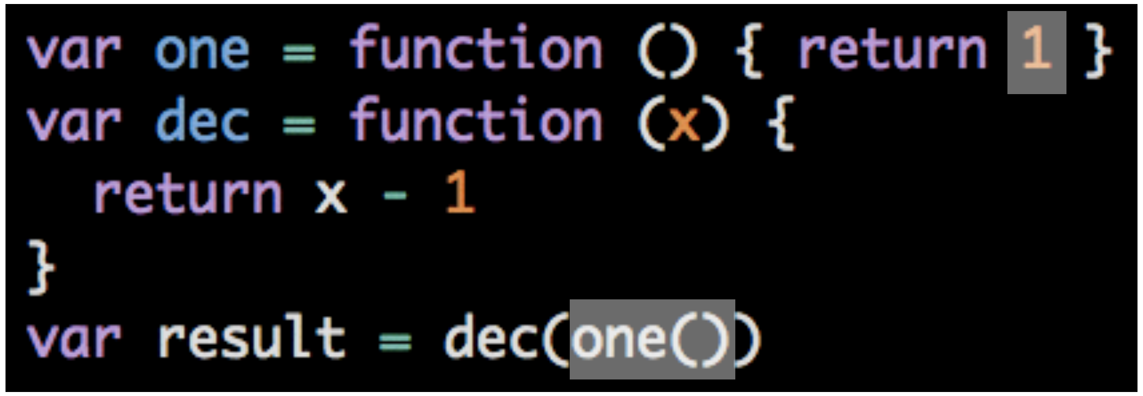
var one

function()

var dec

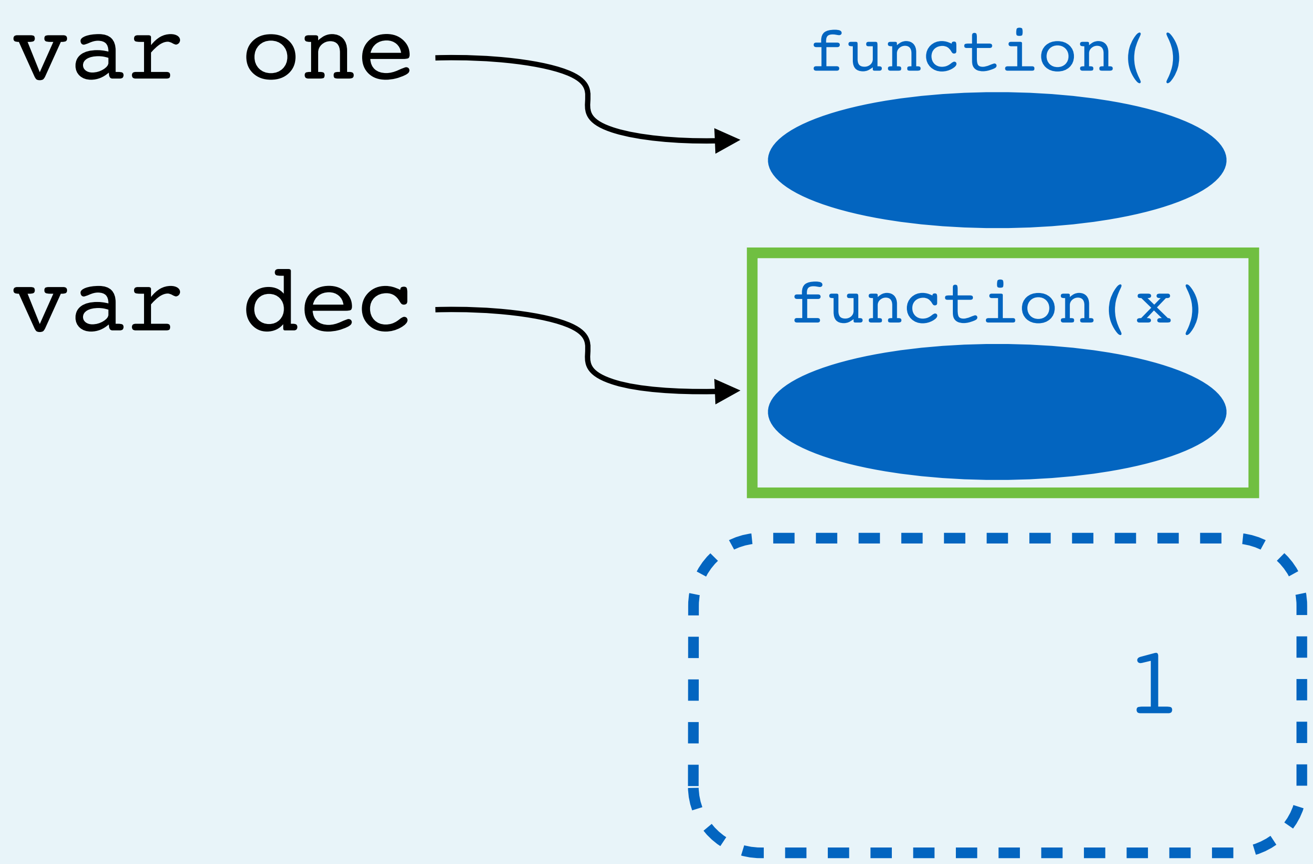
function(x)

Arguments Resolve First (2)

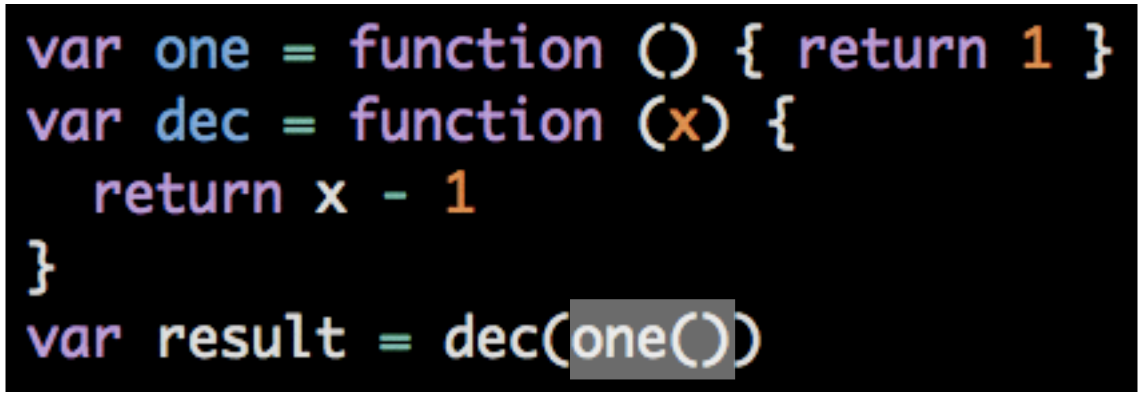


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number

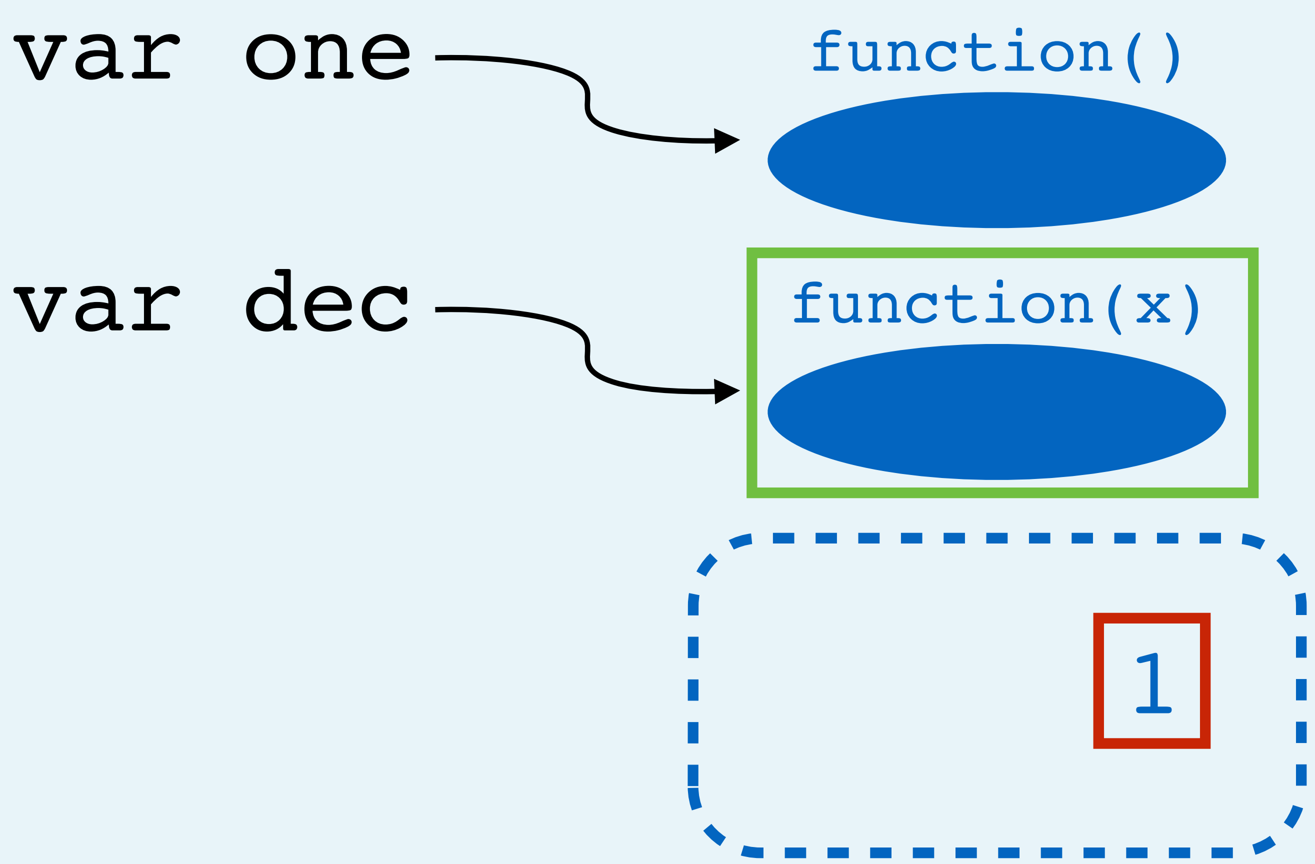


Arguments Resolve First (2)

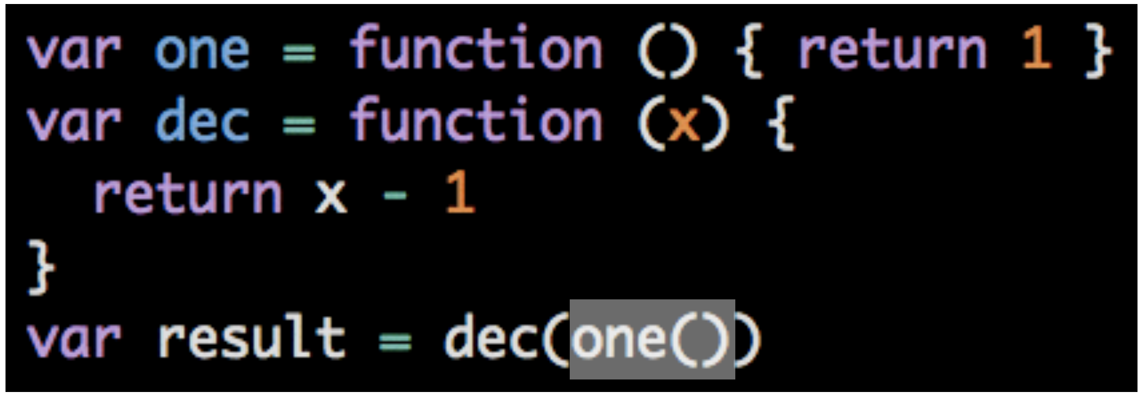


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value

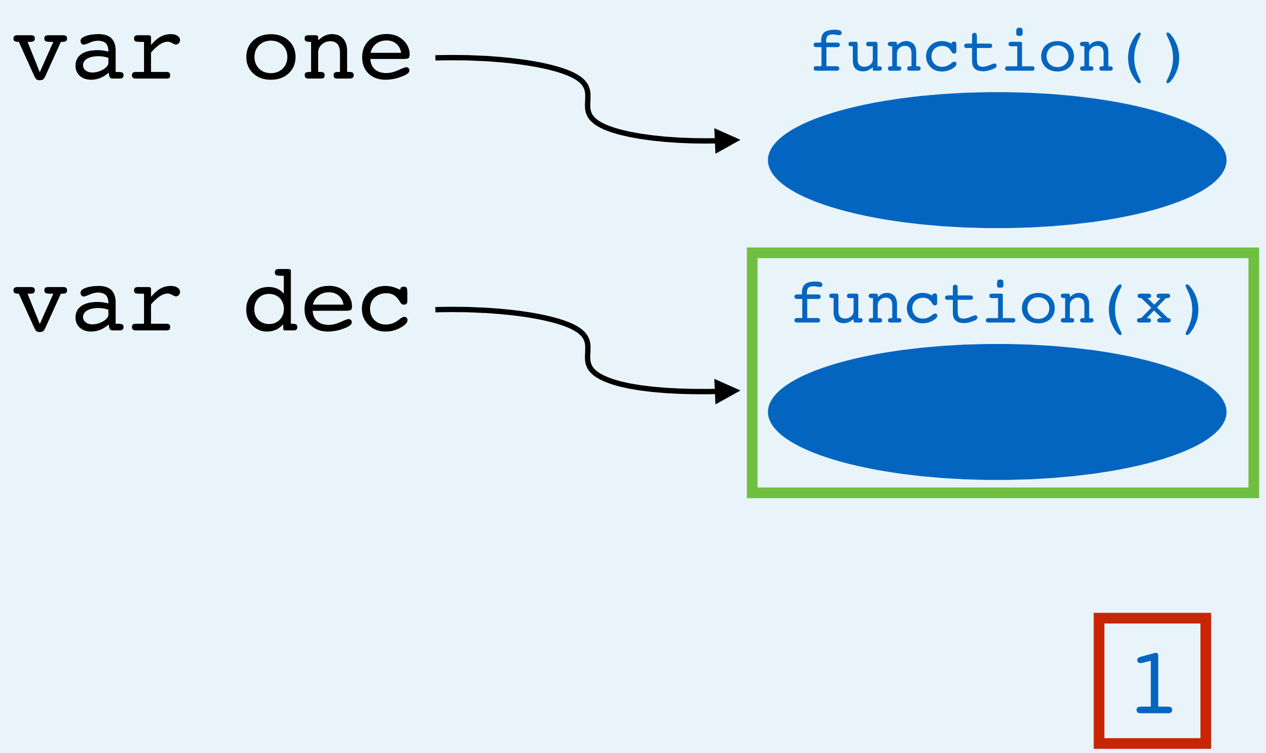


Arguments Resolve First (2)

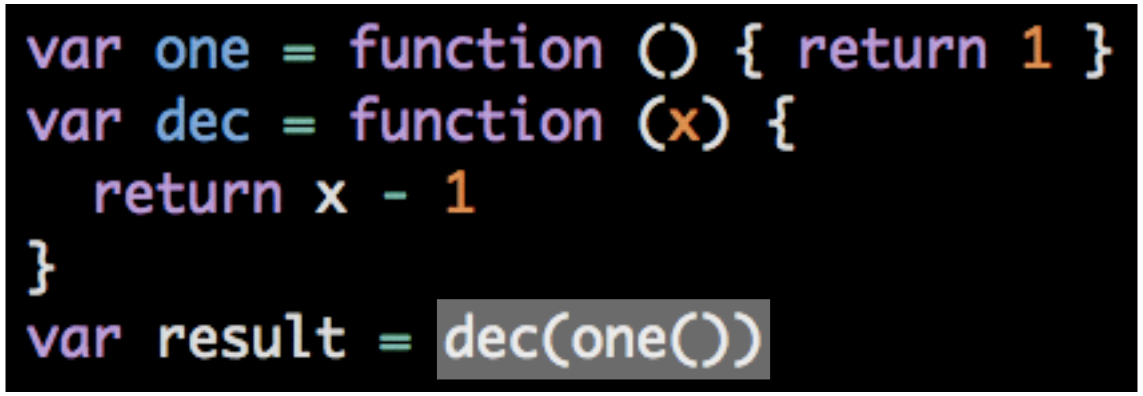


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope

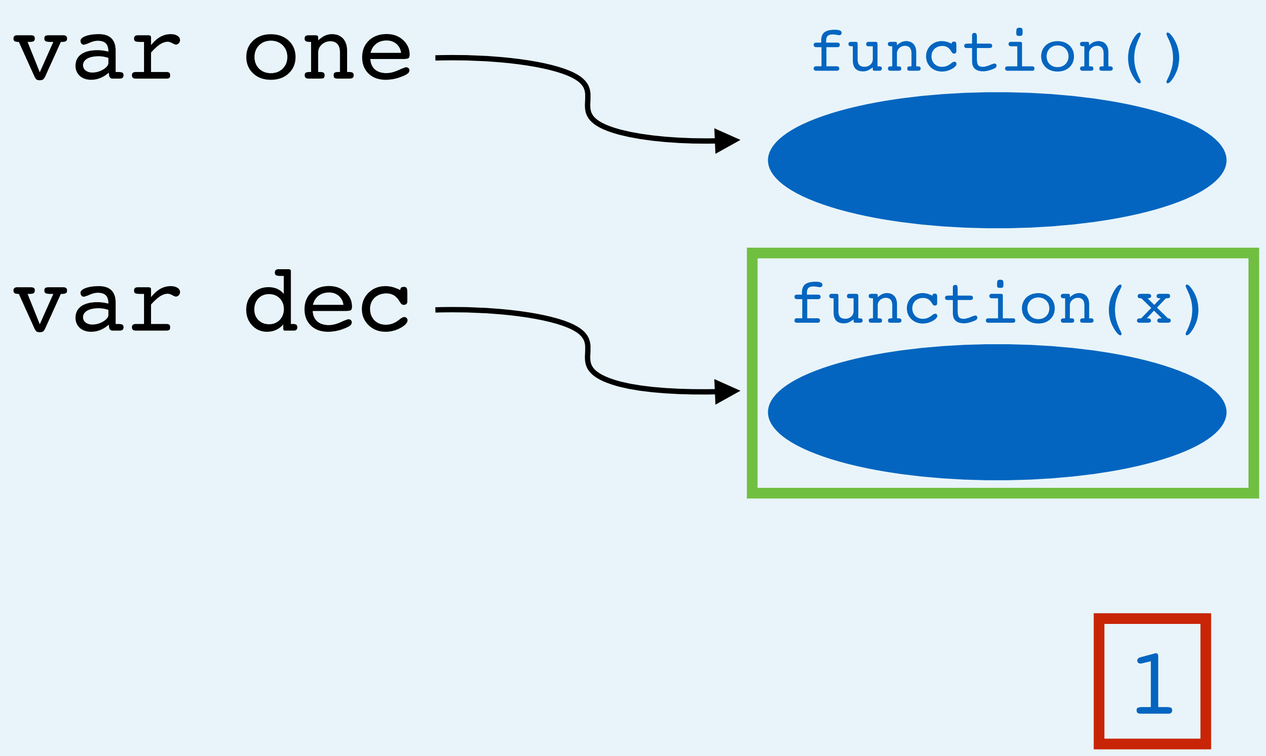


Arguments Resolve First (2)

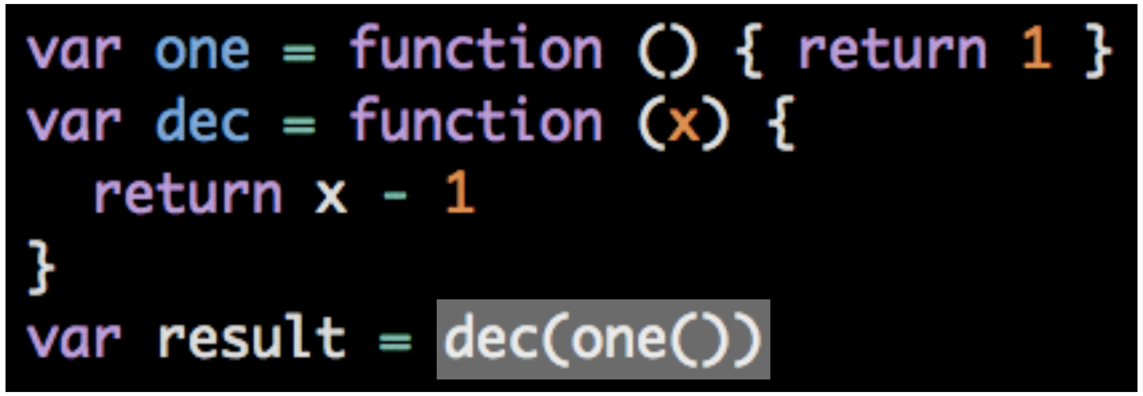


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function

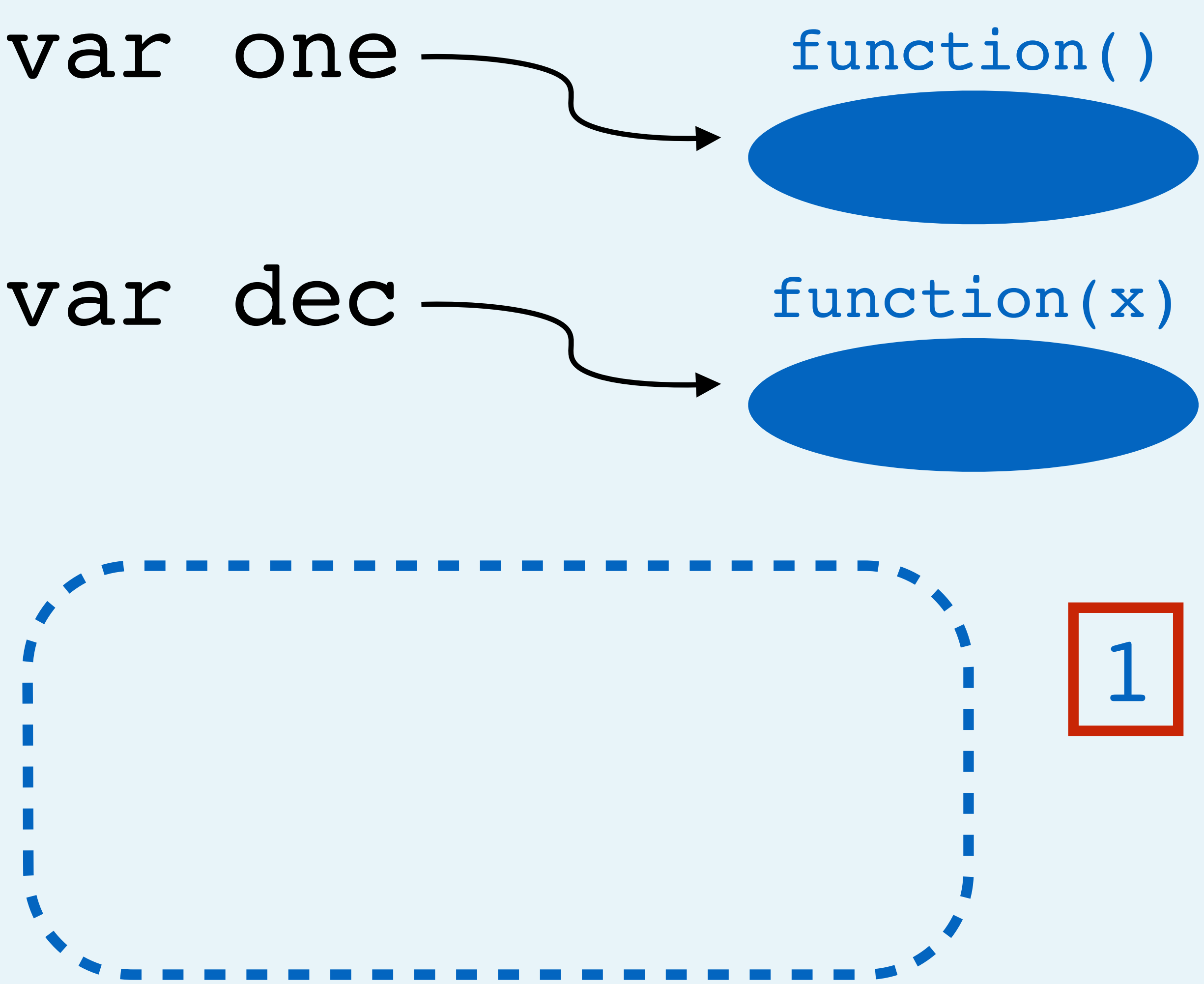


Arguments Resolve First (2)

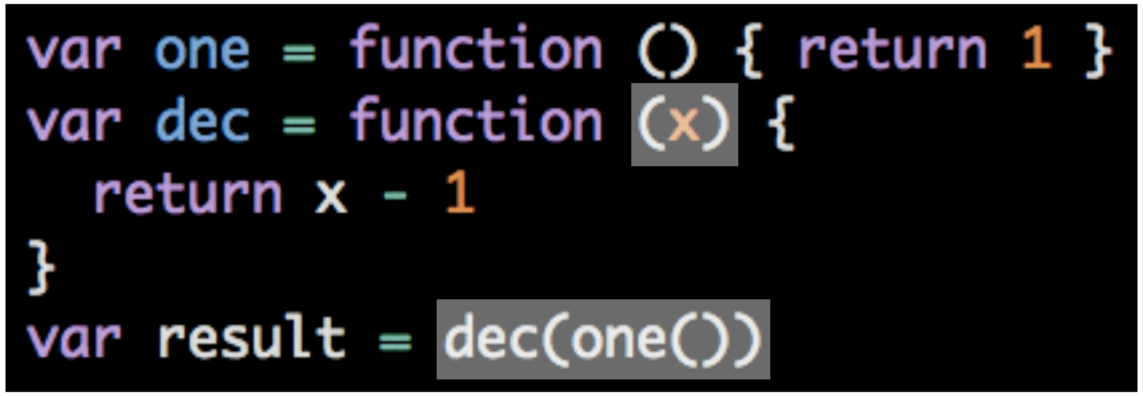


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

a. Assignment
    a. Evaluate right side
    b. Look up value of dec
    c. Resolve argument
        a. Look up value of one
        b. Call function
            a. Create scope
            b. Return statement
                a. Create number
                b. Mark as return value
        c. Garbage collect scope
    d. Call function
        a. Create scope

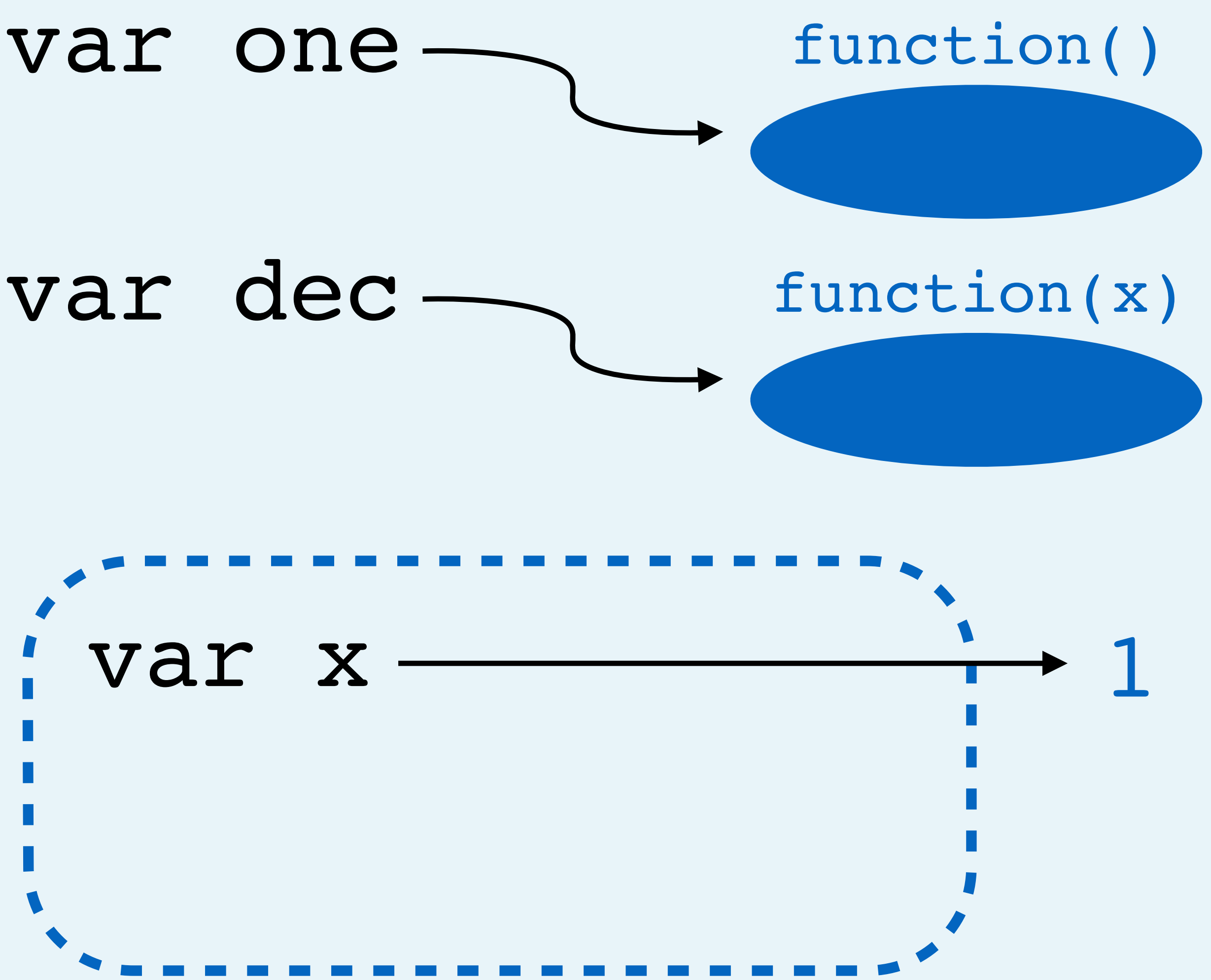


Arguments Resolve First (2)

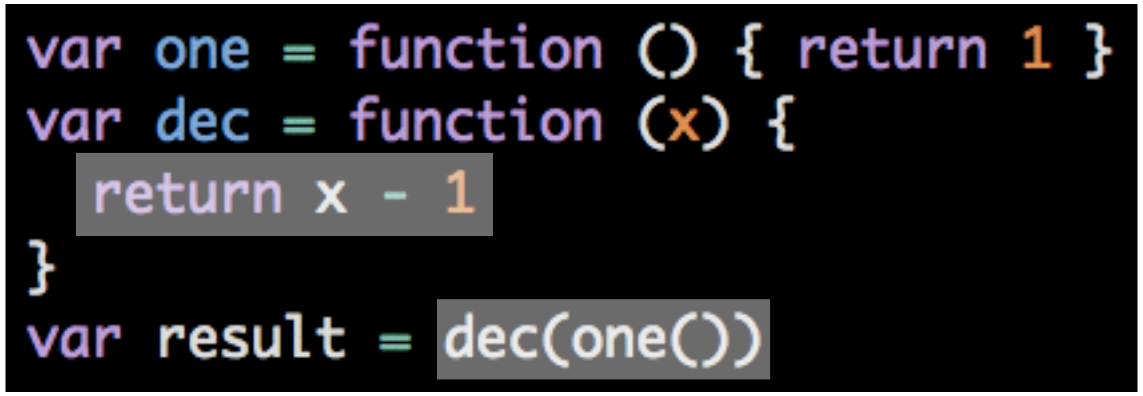


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)

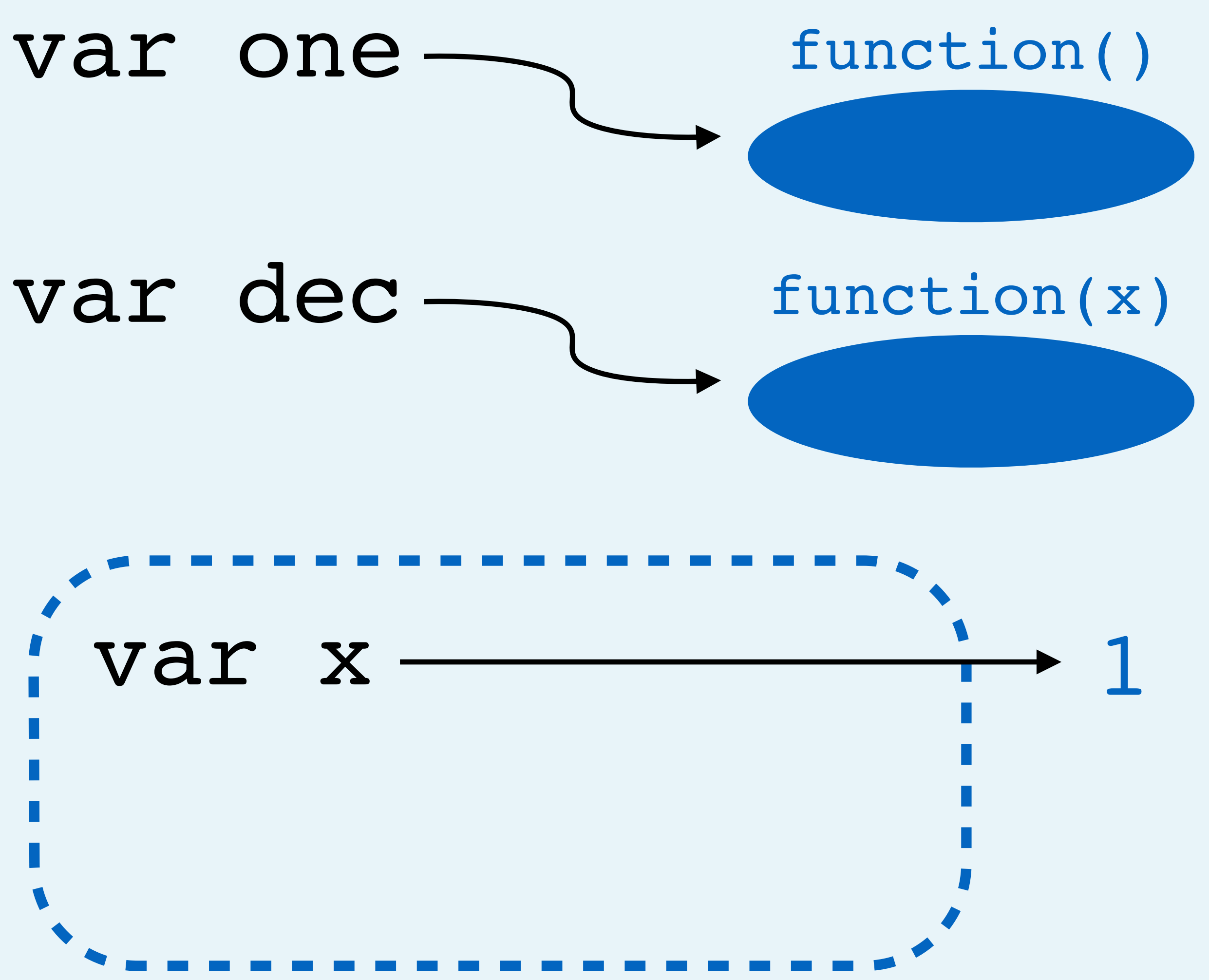


Arguments Resolve First (2)

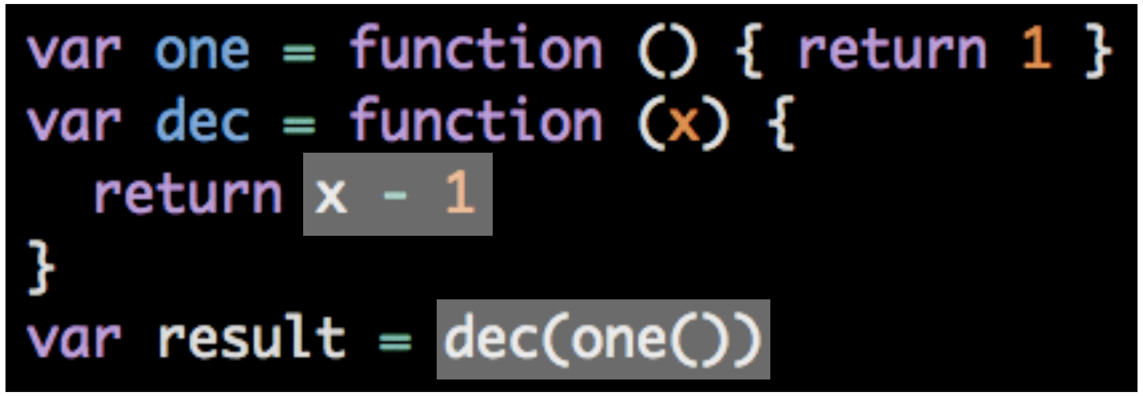


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement

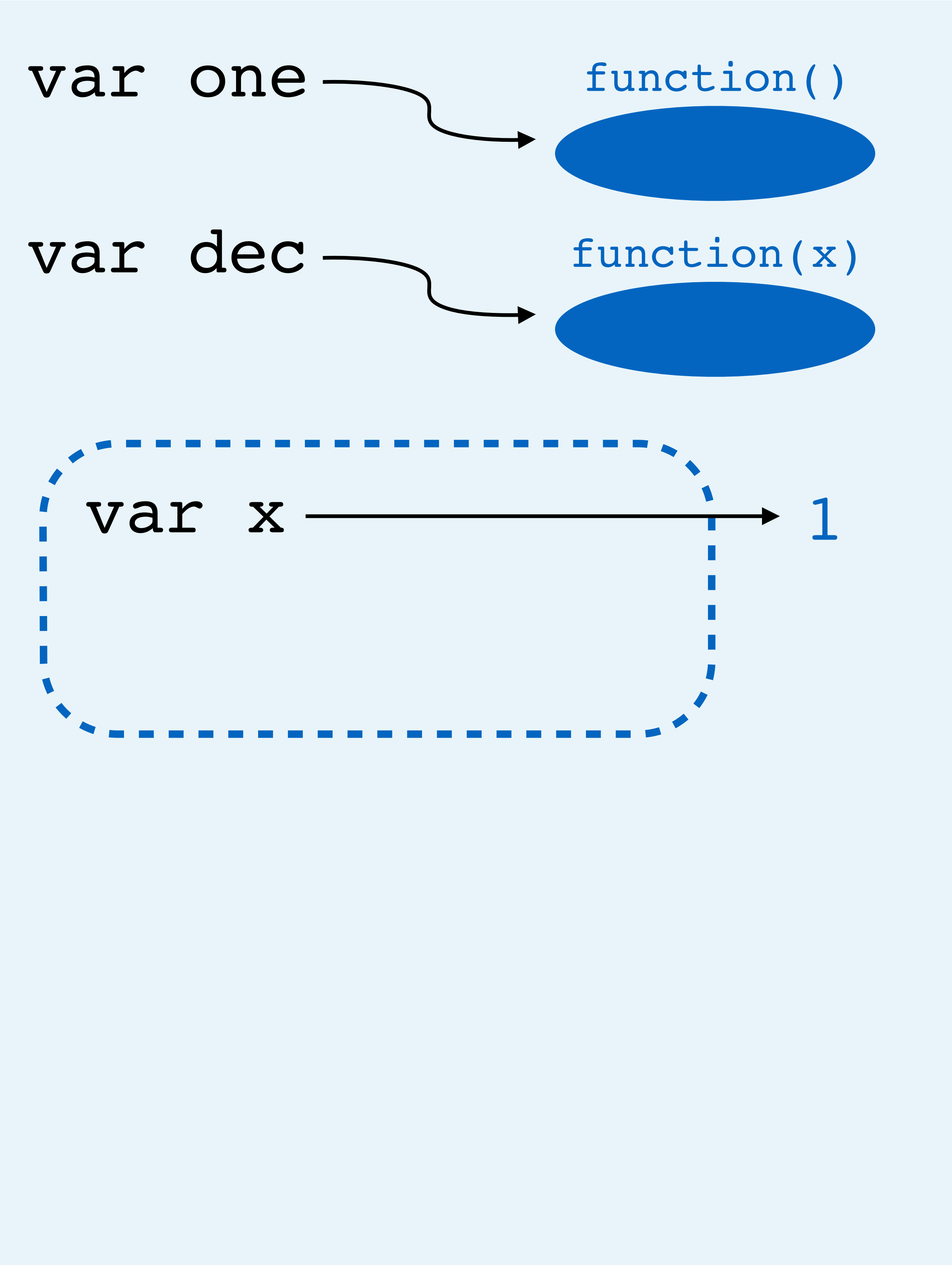


Arguments Resolve First (2)

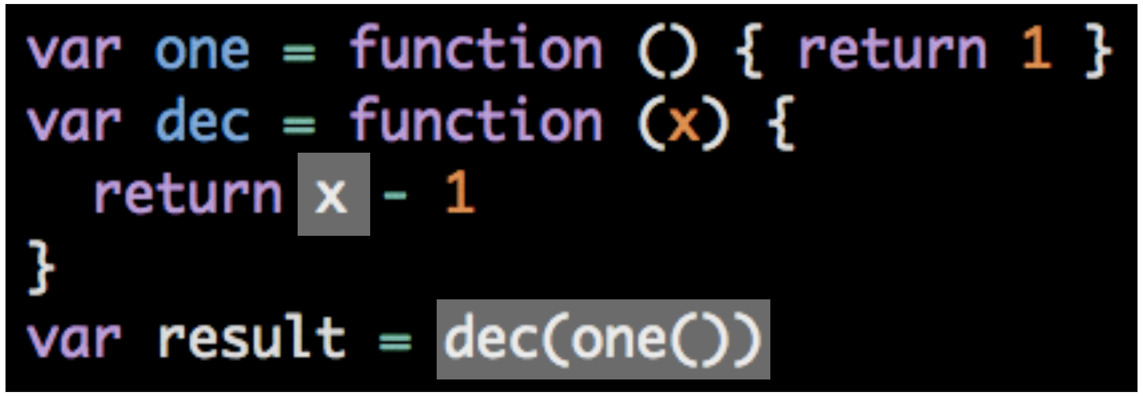


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)

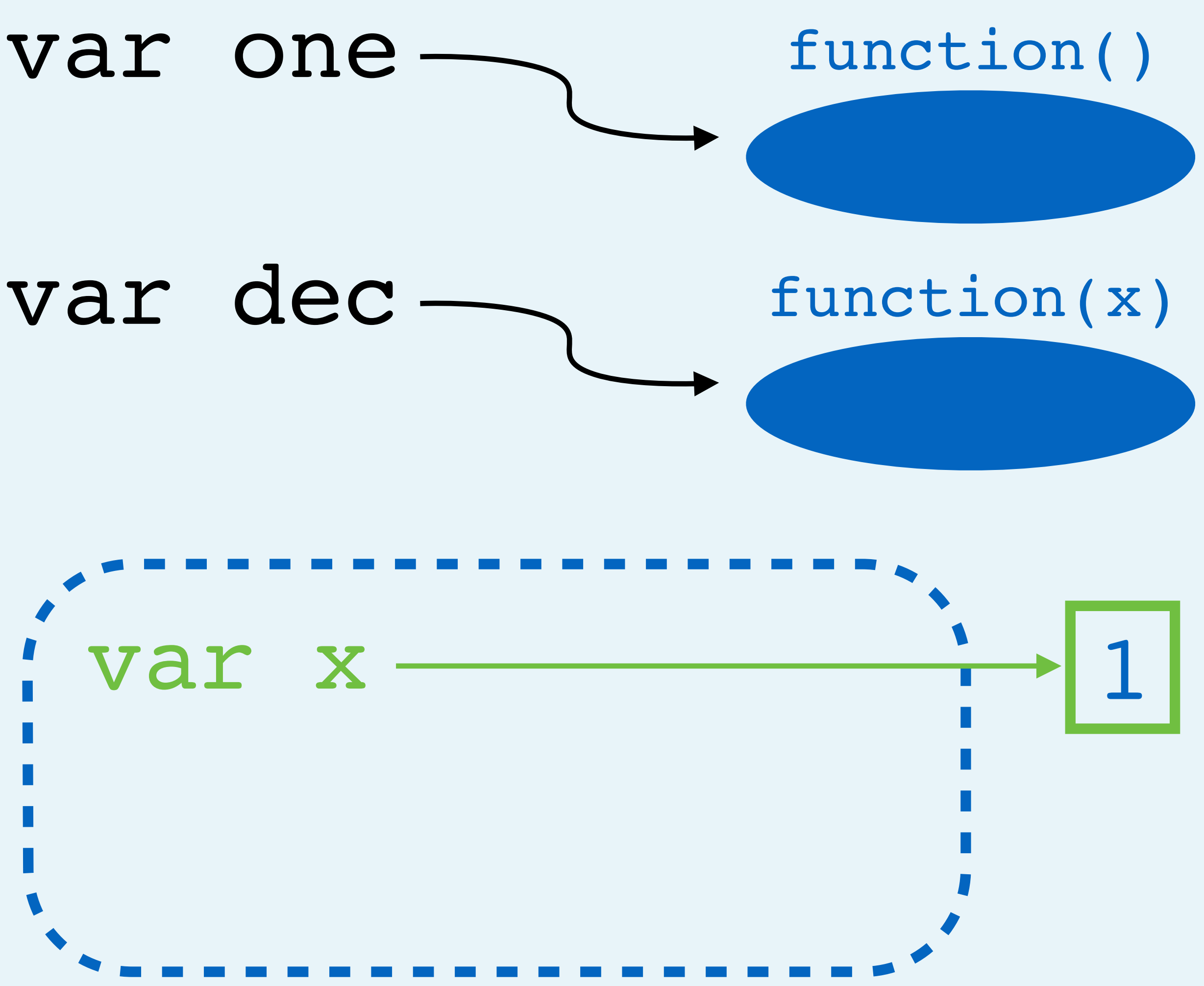


Arguments Resolve First (2)

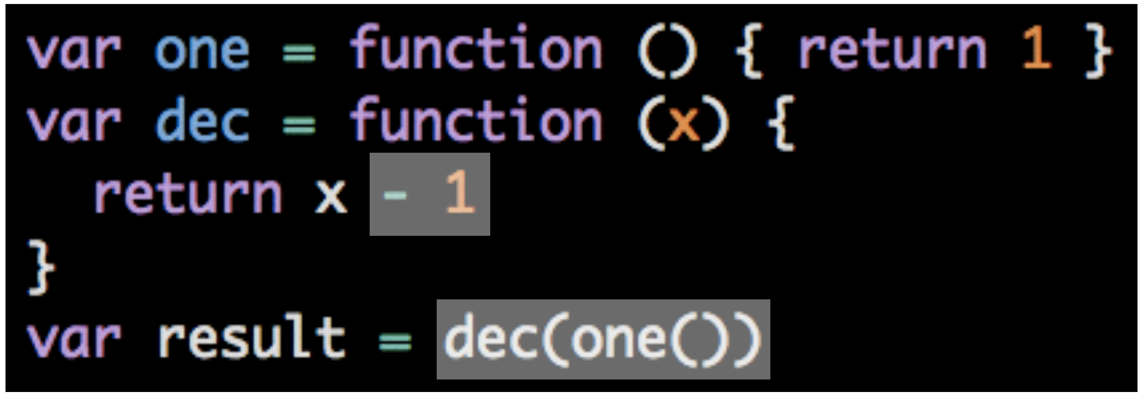


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)
        - a. Look up value of x

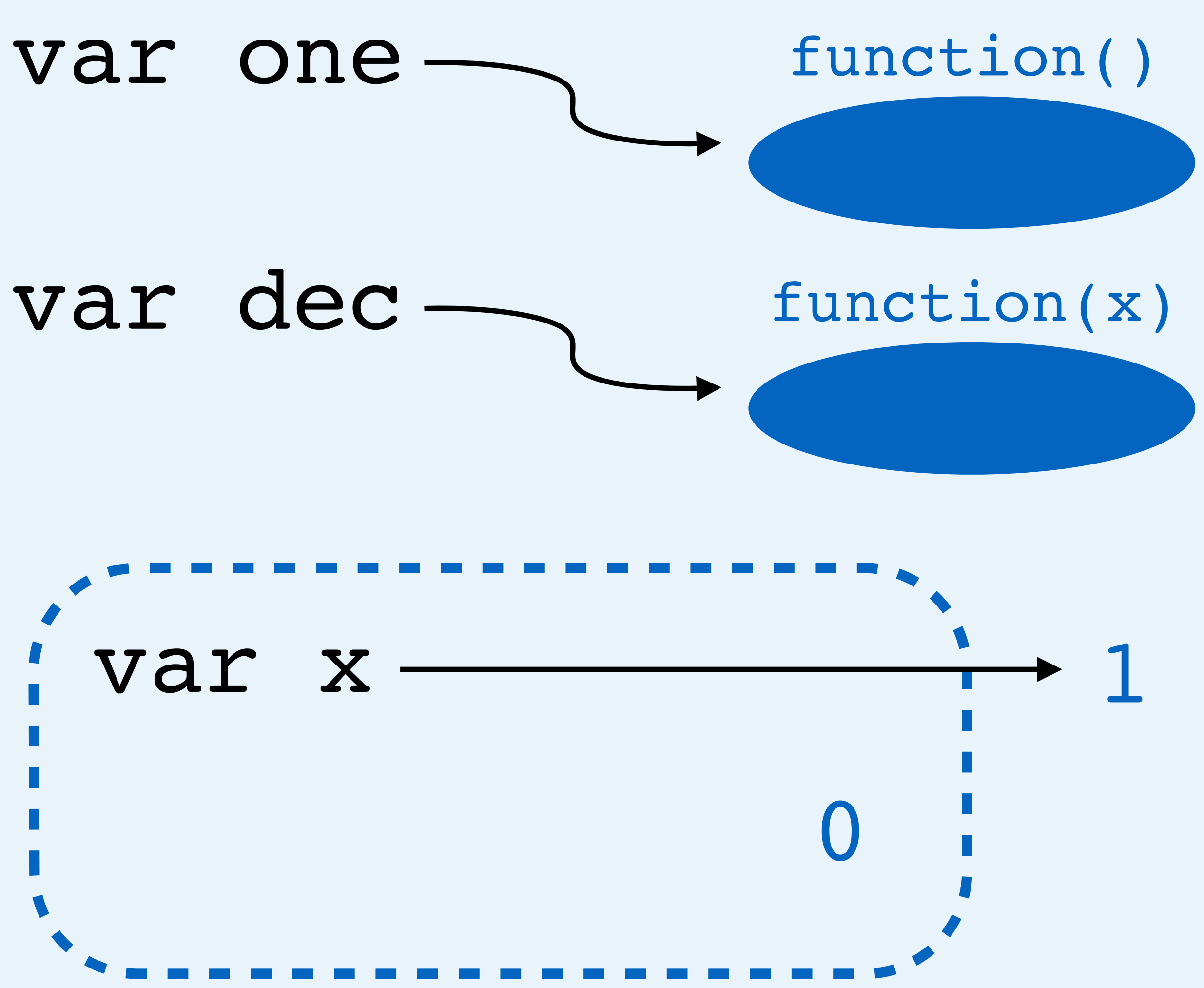


Arguments Resolve First (2)

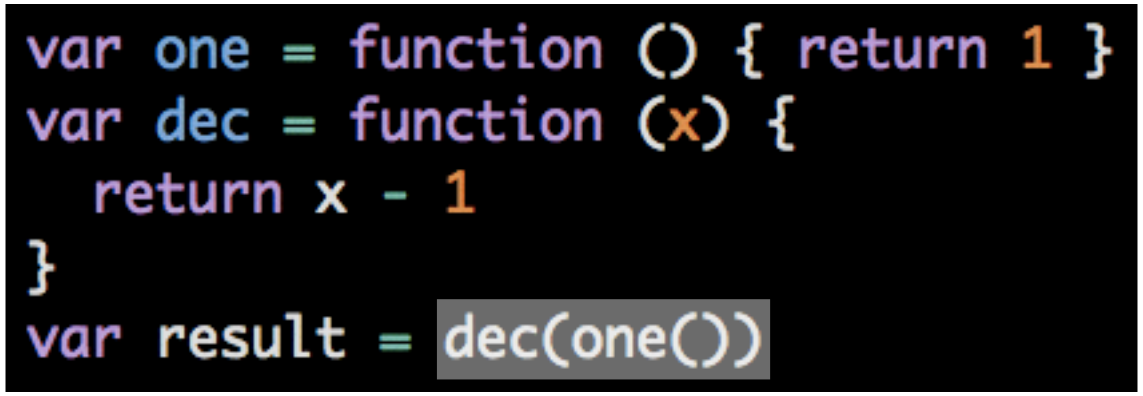


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)
        - a. Look up value of x
        - b. Create value

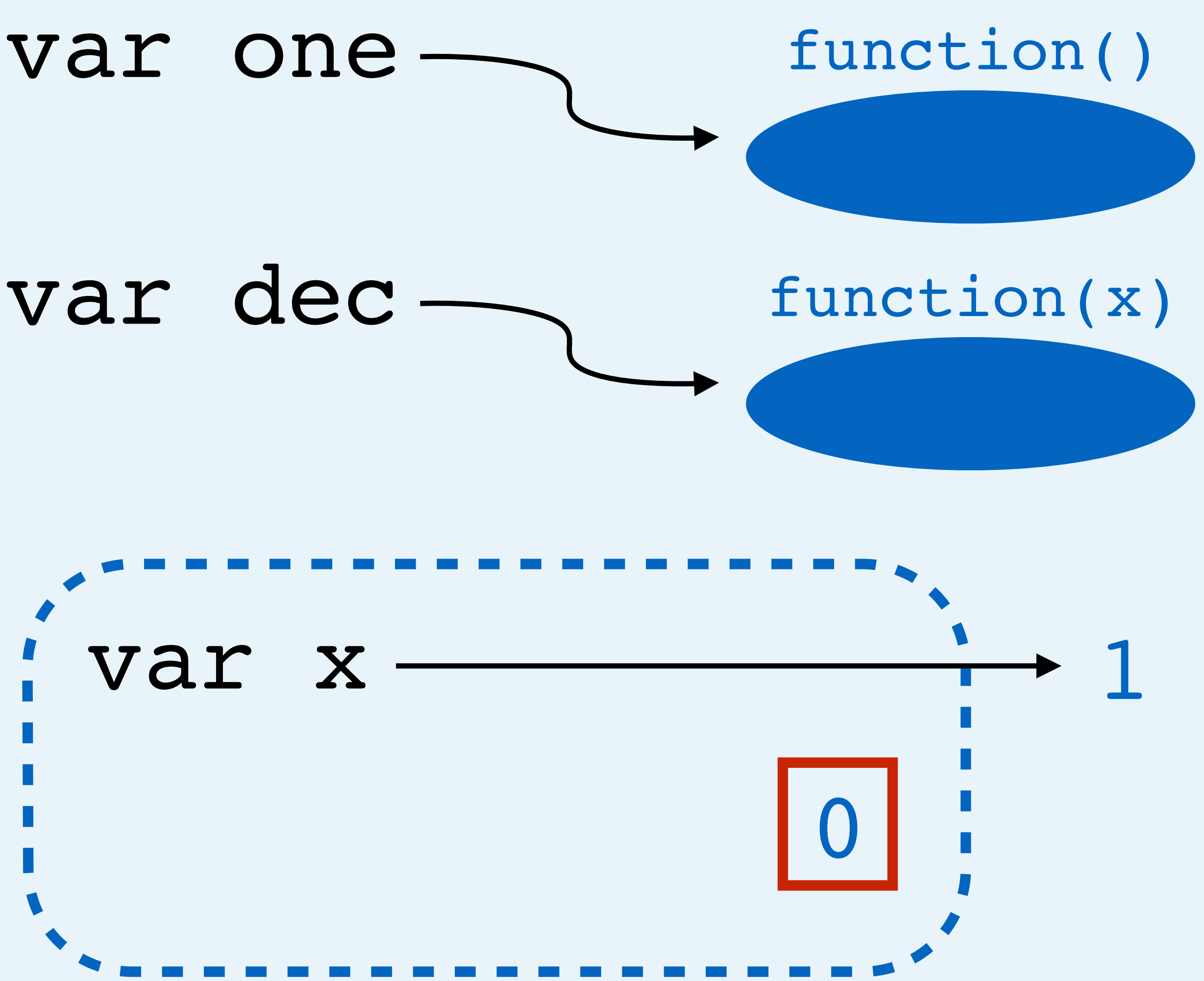


Arguments Resolve First (2)



```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)
        - a. Look up value of x
        - b. Create value
      - b. Mark as return value



var one

function()

var dec

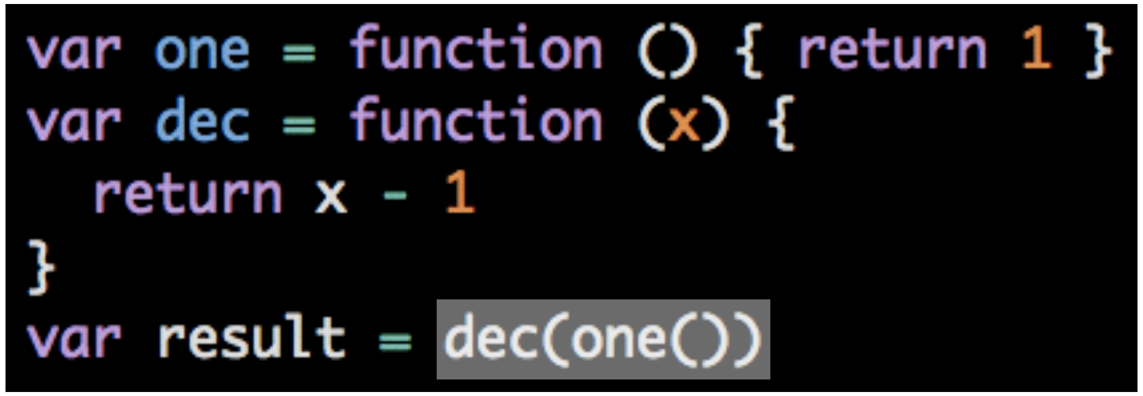
function(x)

var x

1

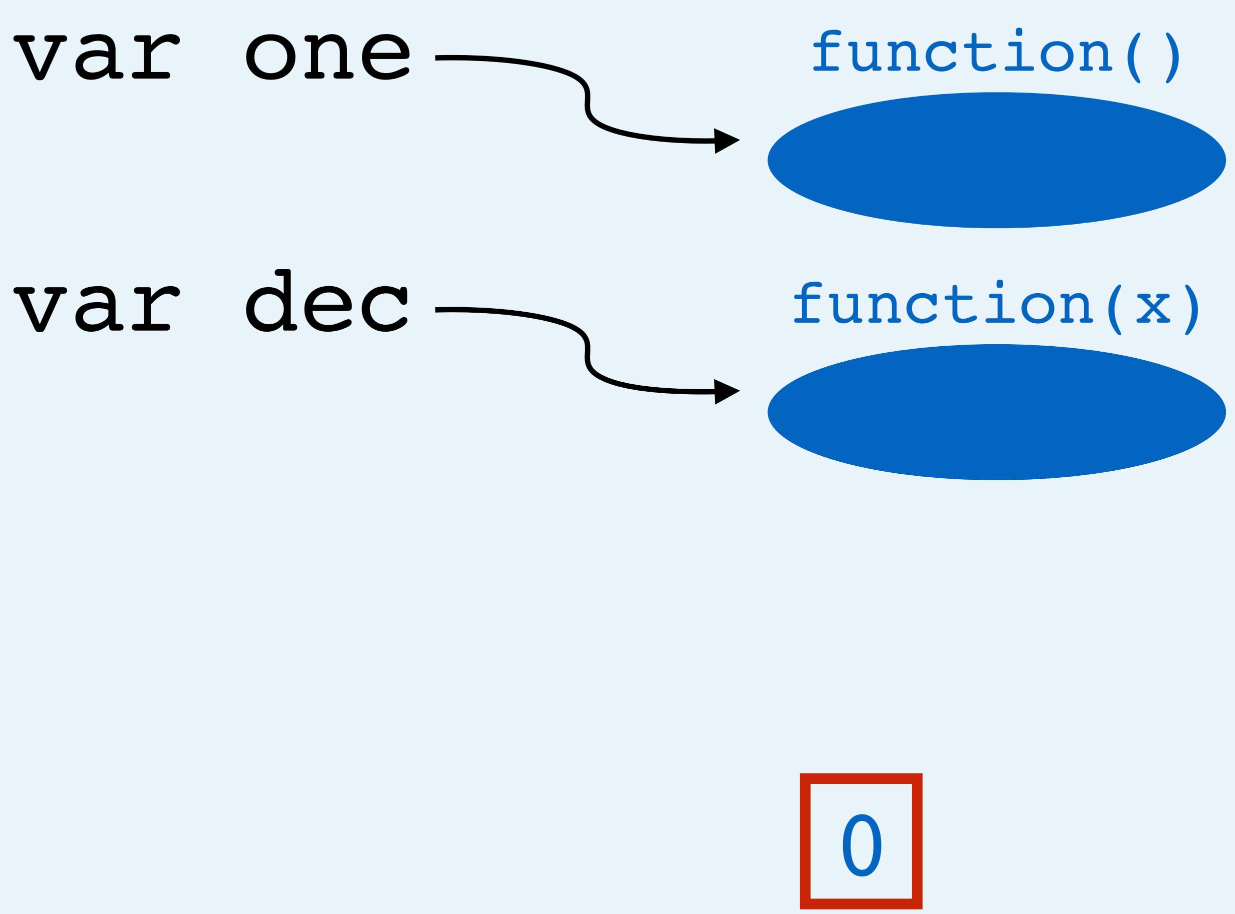
0

Arguments Resolve First (2)

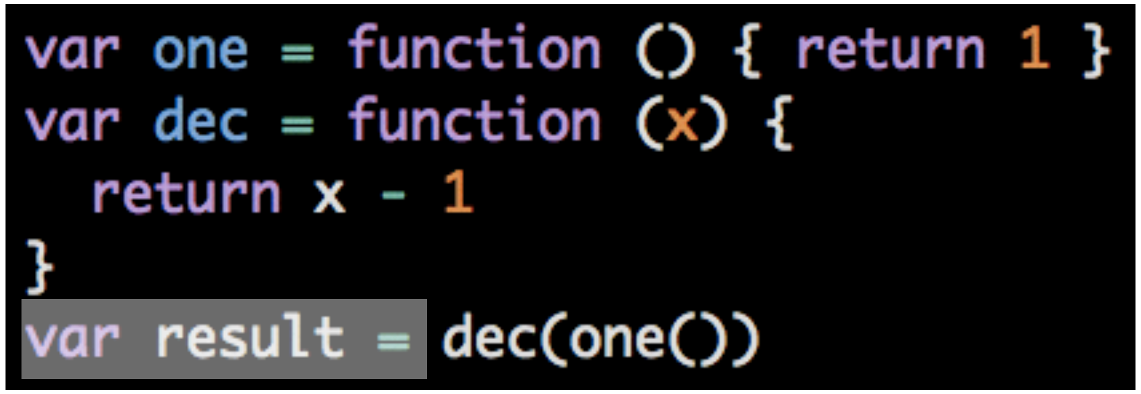


```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)
        - a. Look up value of x
        - b. Create value
      - b. Mark as return value
    - d. Garbage collect scope

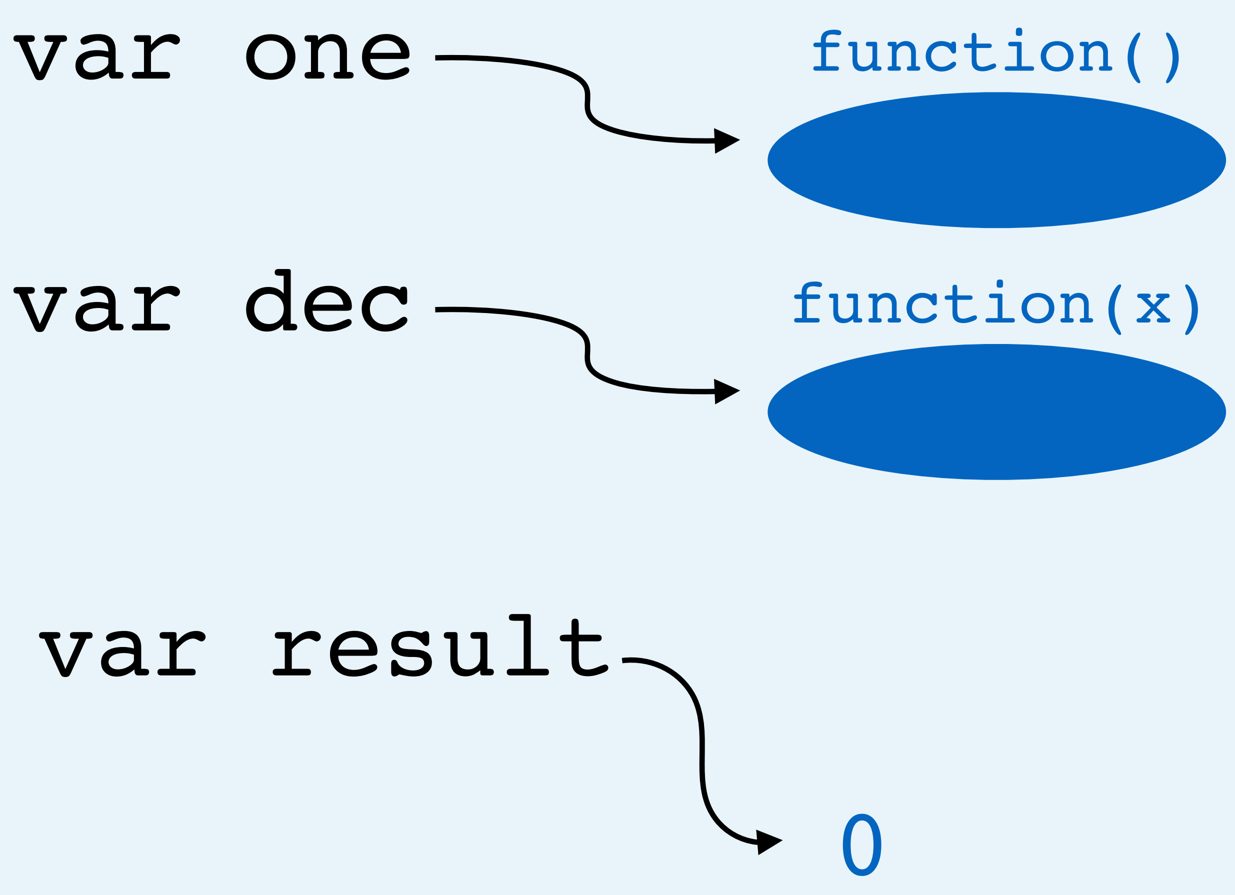


Arguments Resolve First (2)



```
var one = function () { return 1 }
var dec = function (x) {
  return x - 1
}
var result = dec(one())
```

- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (subtraction)
        - a. Look up value of x
        - b. Create value
      - b. Mark as return value
    - d. Garbage collect scope
  - e. Create var result, point to value


var one
function()
var dec
function(x)
var result
0