

# JSDN

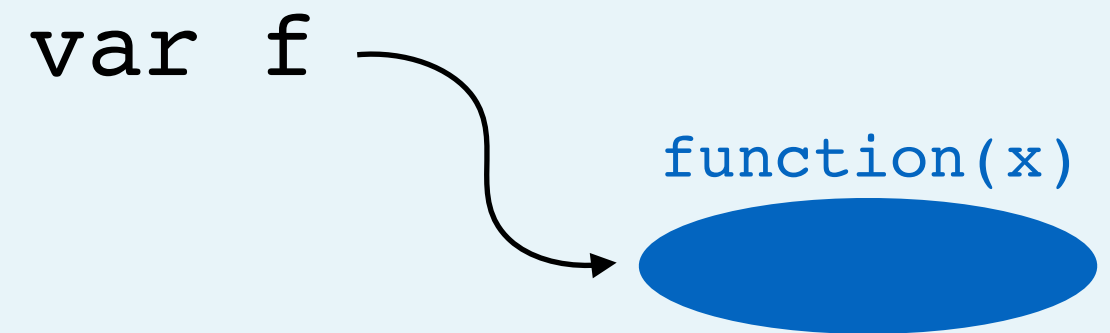
(by example, part 2)

[more]

# Function Calls

# Arguments Resolve First

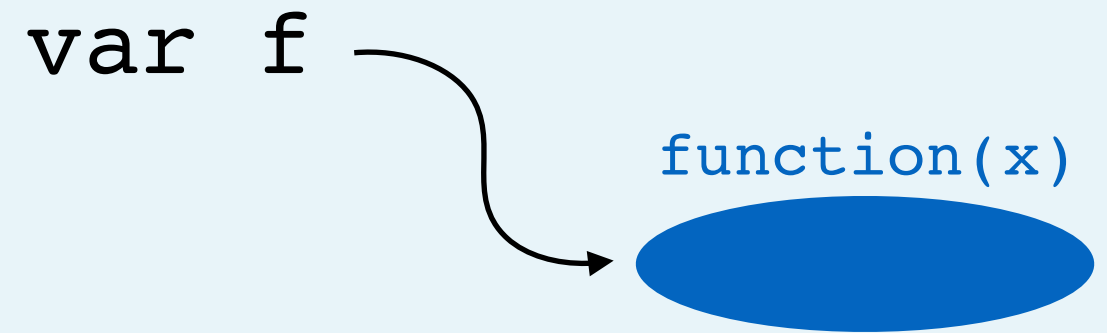
```
var f = function (x) {  
  return x + 1  
}  
var result = f(2 + 3)
```



# Arguments Resolve First

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

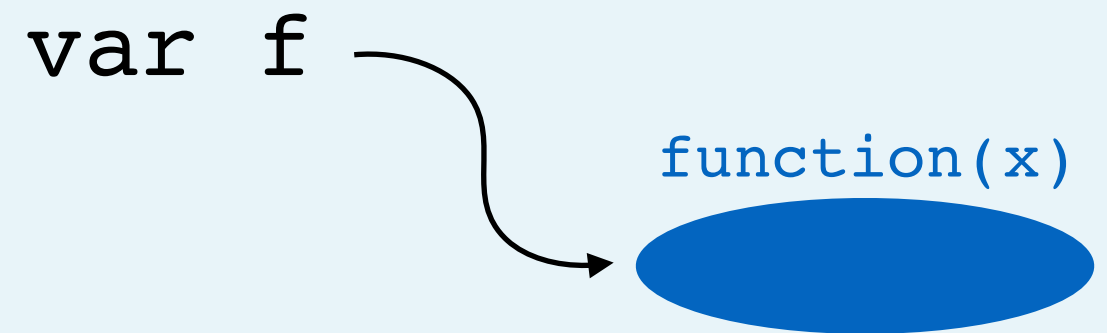
a. Assignment



# Arguments Resolve First

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

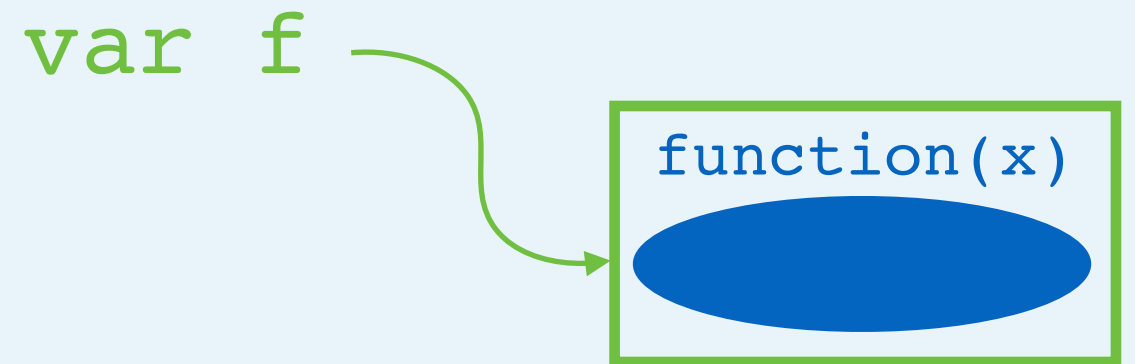
- a. Assignment
  - a. Evaluate right side



# Arguments Resolve First

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

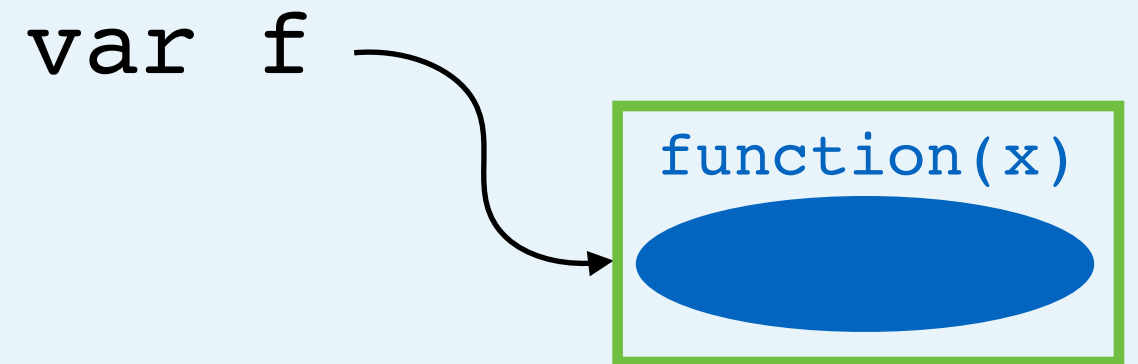
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)



# Arguments Resolve First

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

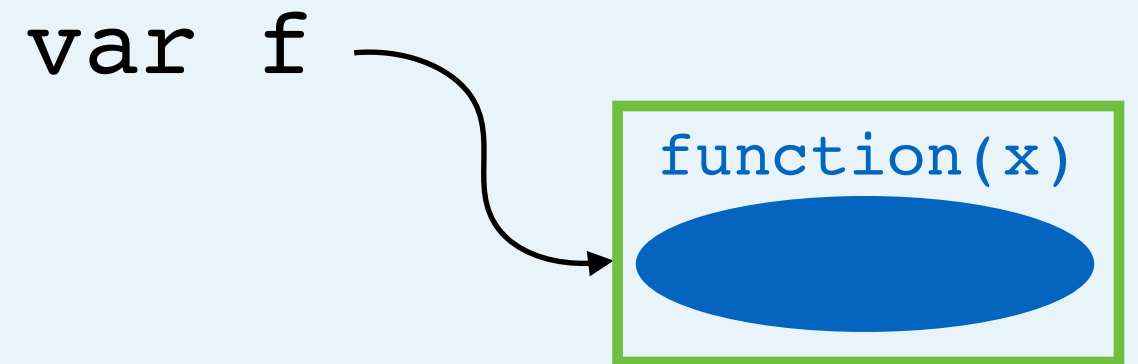
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument



# 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)



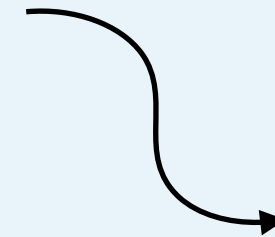


# Arguments Resolve First

```
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

var f



function(x)



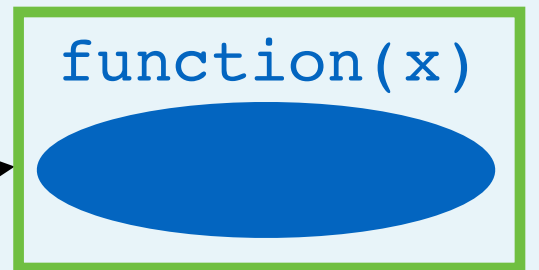
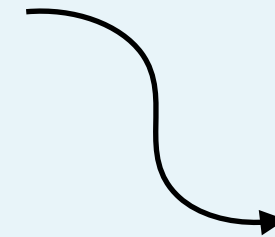
5

# 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

var f

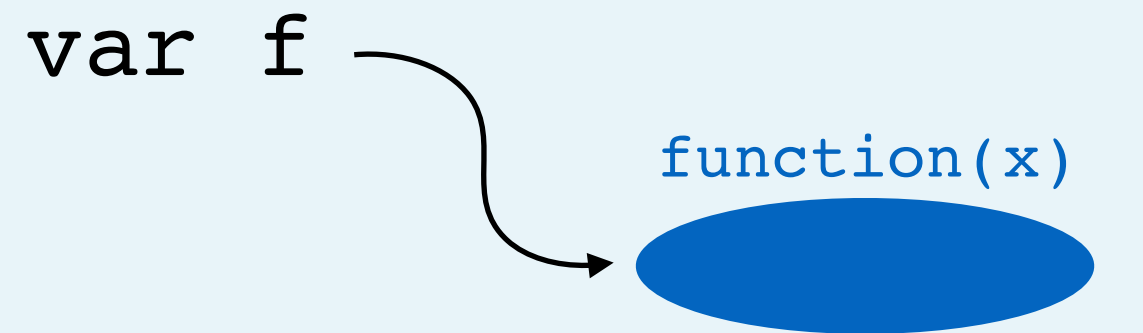


5

# Arguments Resolve First

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

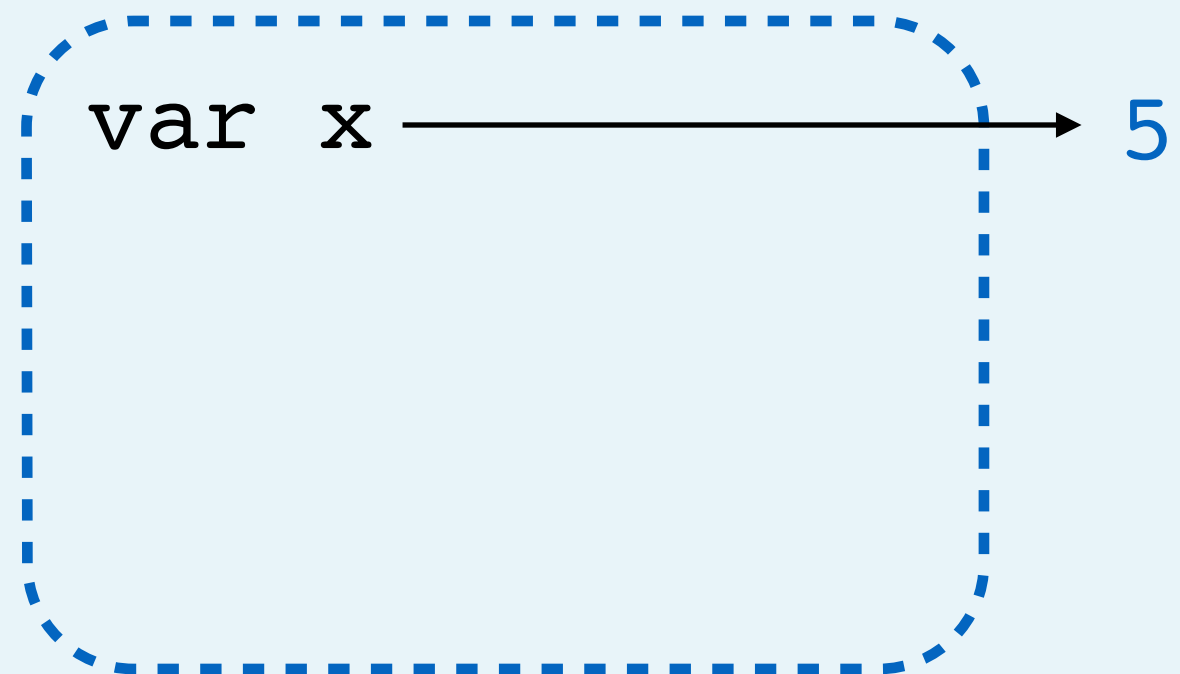
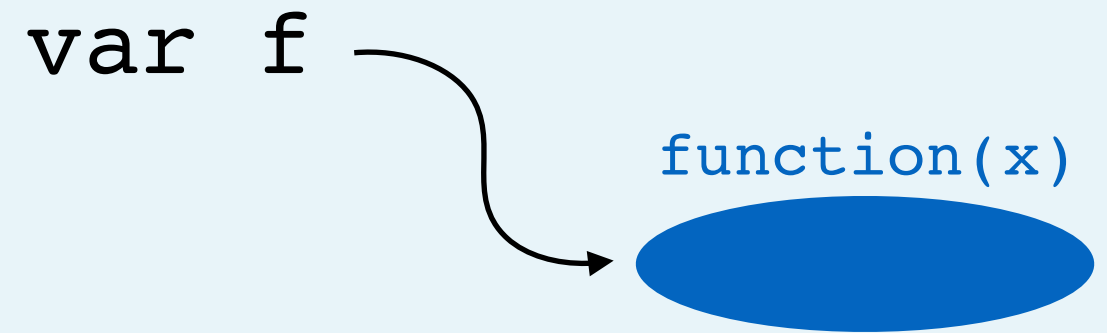
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope



# Arguments Resolve First

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

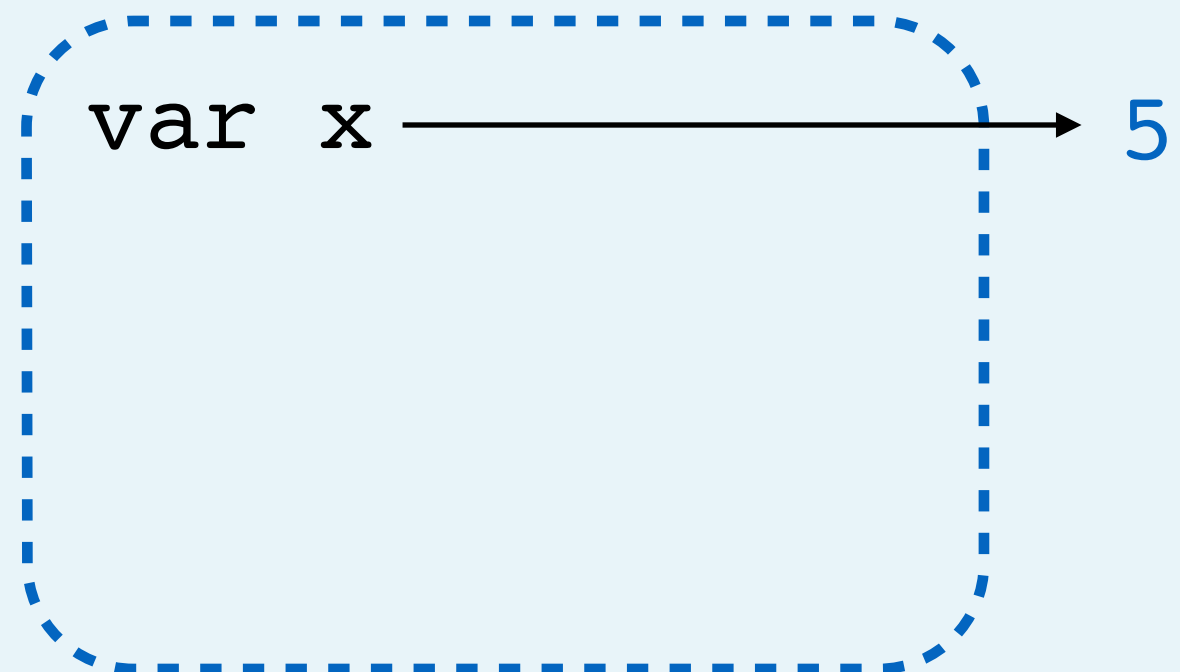
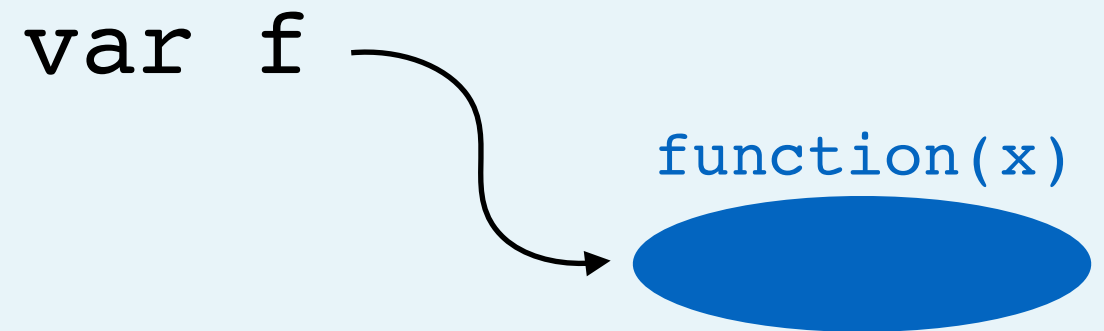
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)



# Arguments Resolve First

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

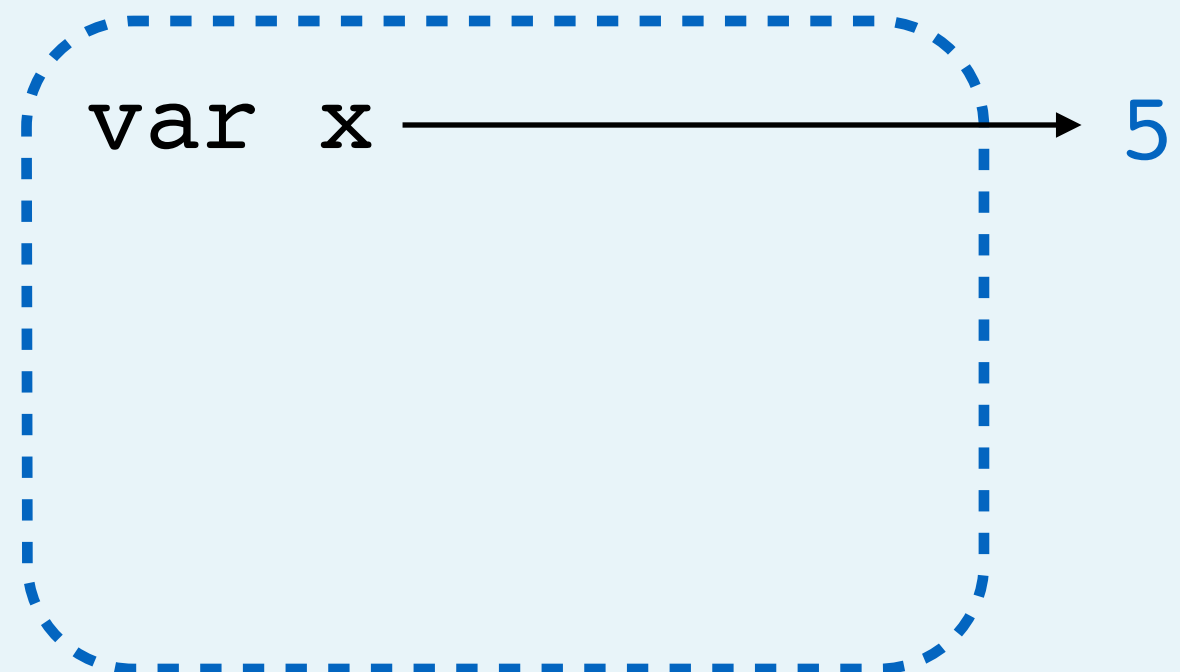
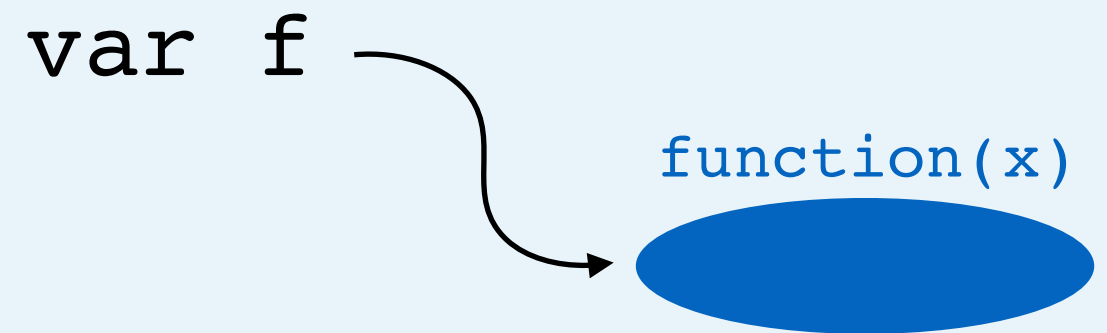
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement



# Arguments Resolve First

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

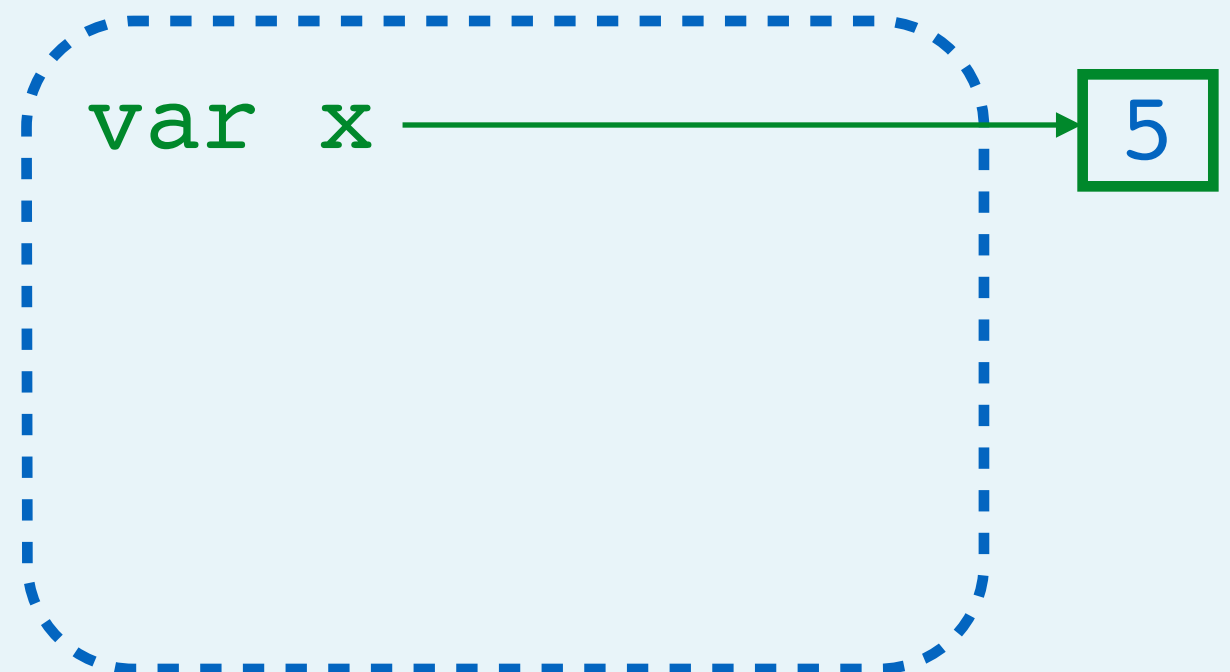
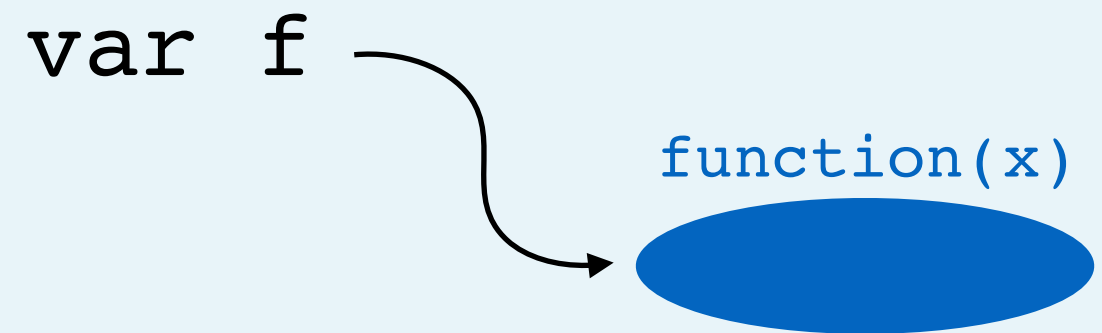
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)



# Arguments Resolve First

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

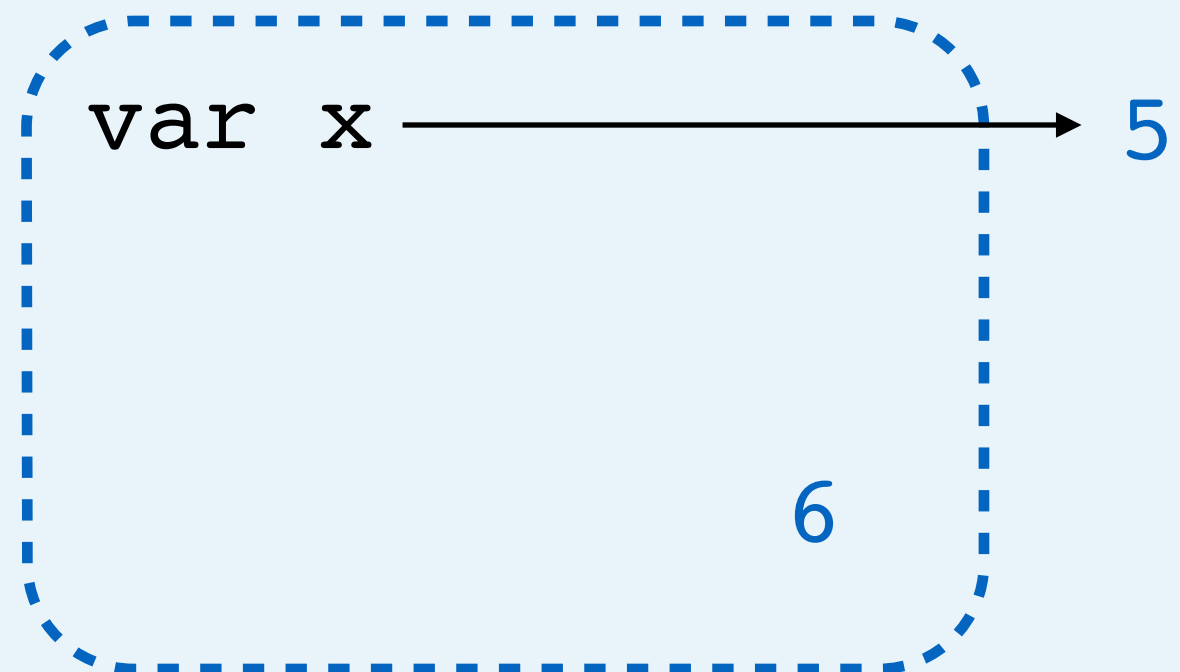
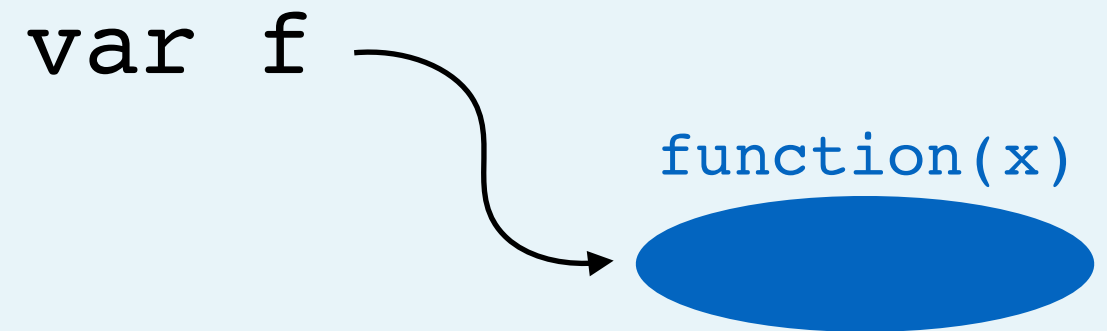
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)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



# Arguments Resolve First

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

- 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

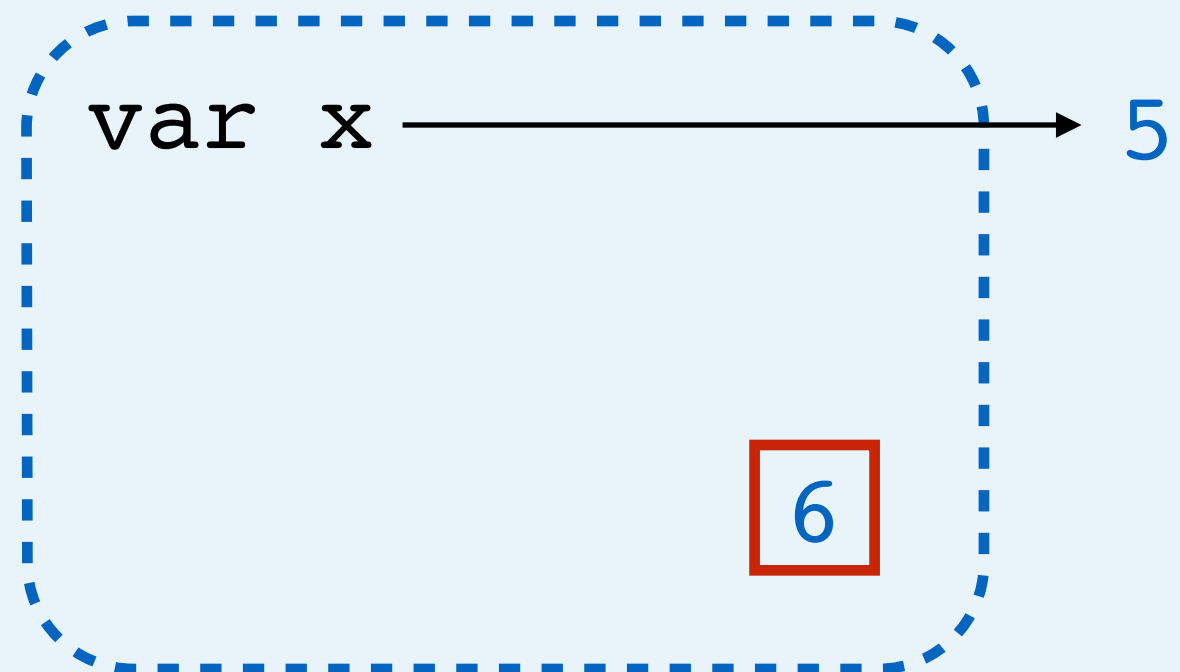
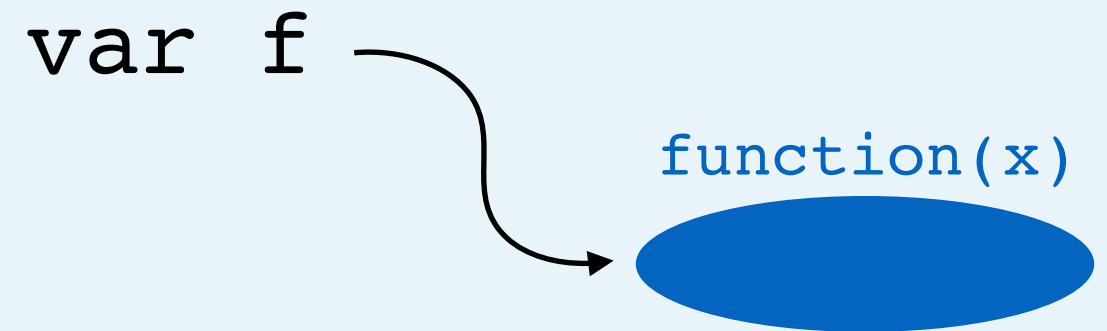




# Arguments Resolve First

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

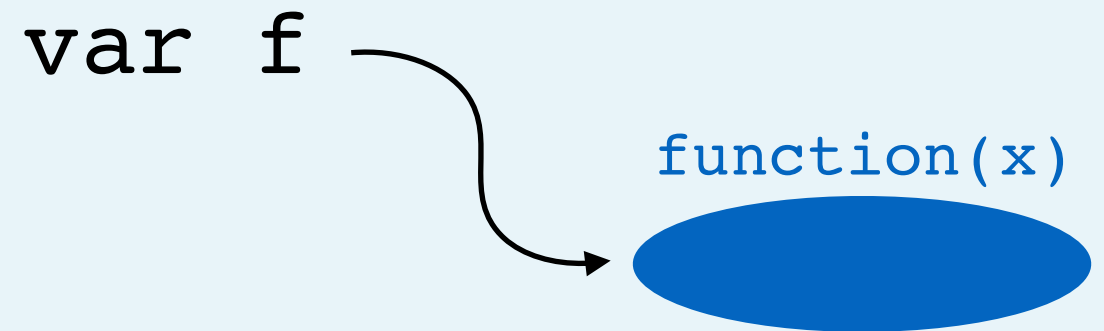
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of f (it's a function!)
  - c. Resolve argument
    - a. Binary Operation (addition)
      - a. Create value
  - d. Call function
    - a. Create scope
    - b. Create parameter(s)
    - c. Return statement
      - a. Binary Operation (addition)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

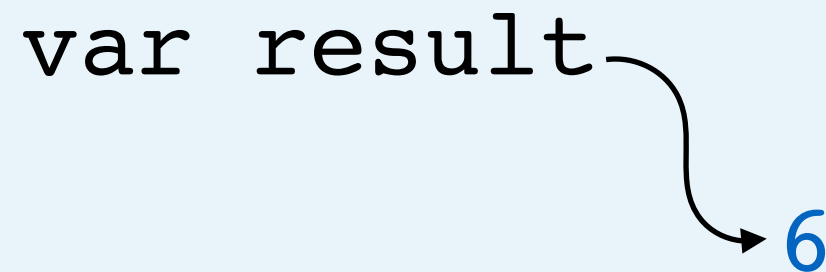
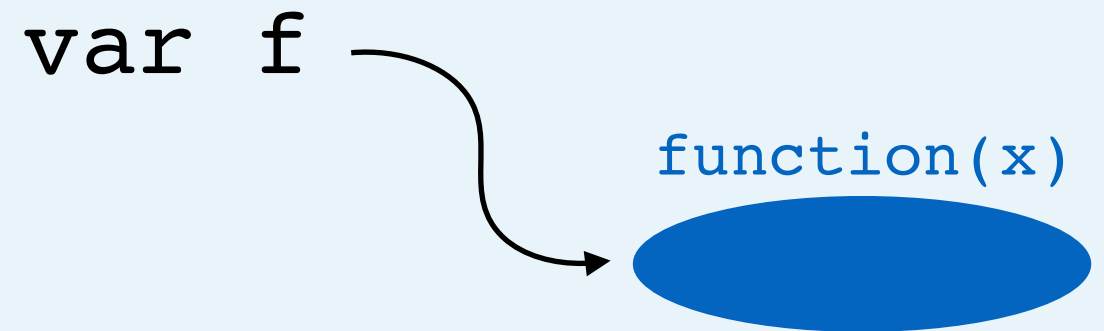


6

# Arguments Resolve First

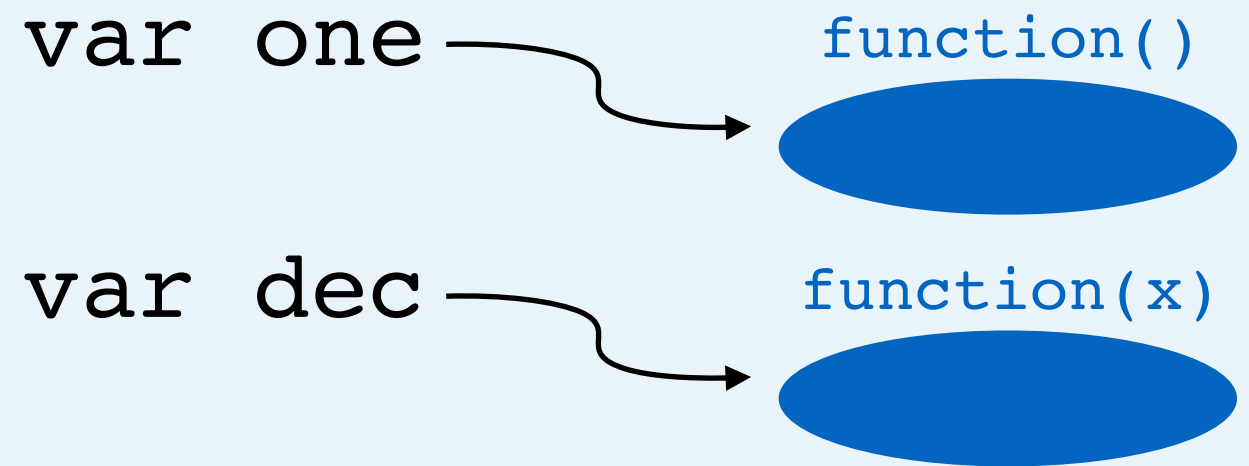
```
var f = function (x) {  
  return x + 1  
}  
var result = f(2 + 3)
```

- 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)

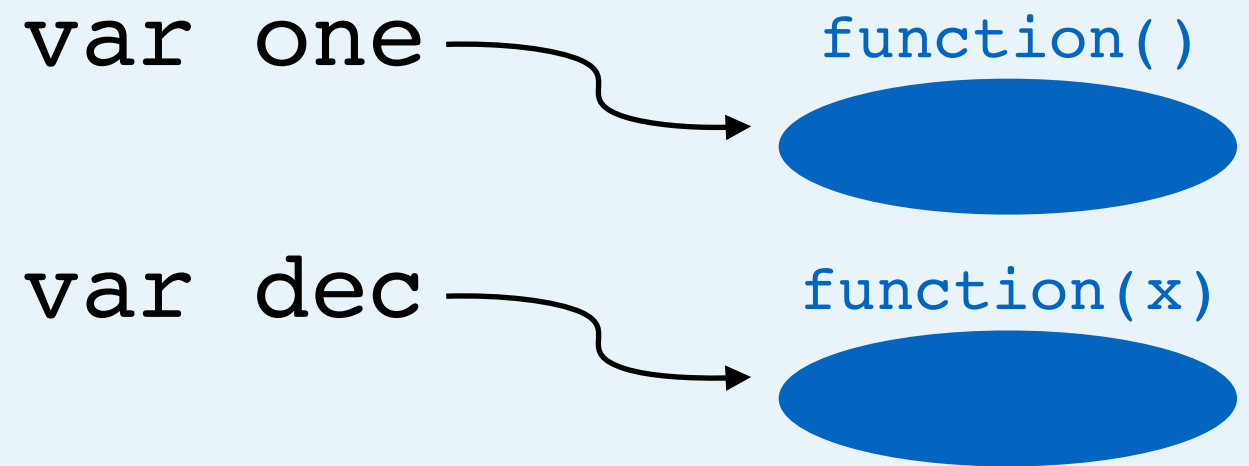
```
var one = function () { return 1 }  
var dec = function (x) {  
  return x - 1  
}  
var result = dec(one())
```



# Arguments Resolve First (2)

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

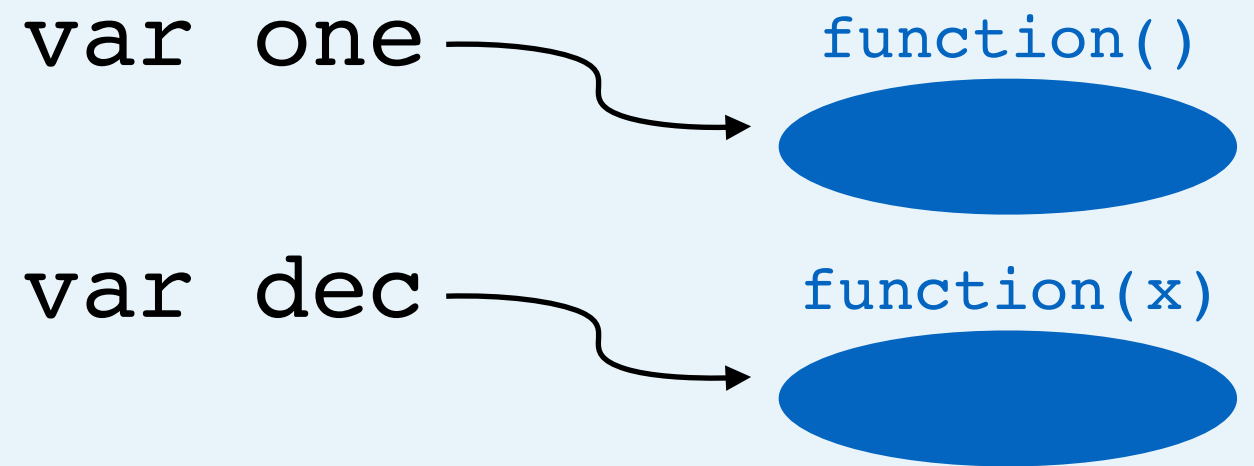
a. Assignment



# Arguments Resolve First (2)

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

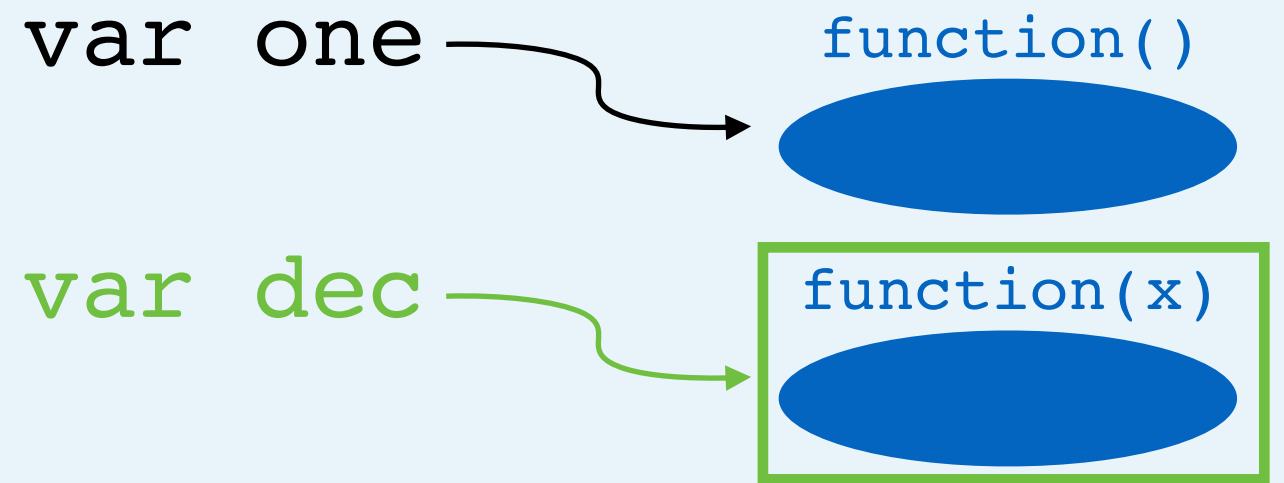
- a. Assignment
  - a. Evaluate right side



# Arguments Resolve First (2)

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

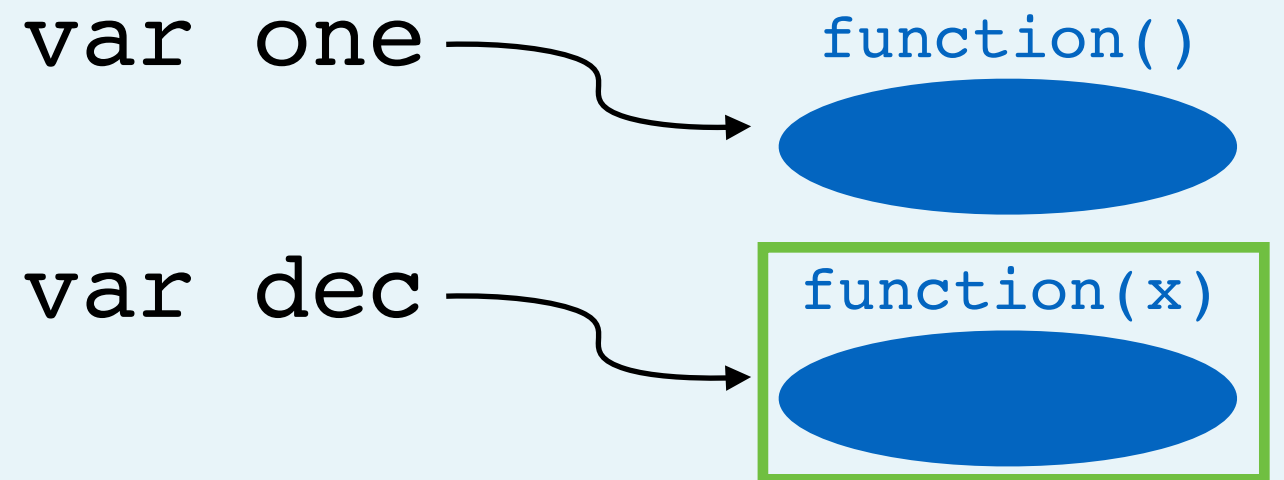
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec



# 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

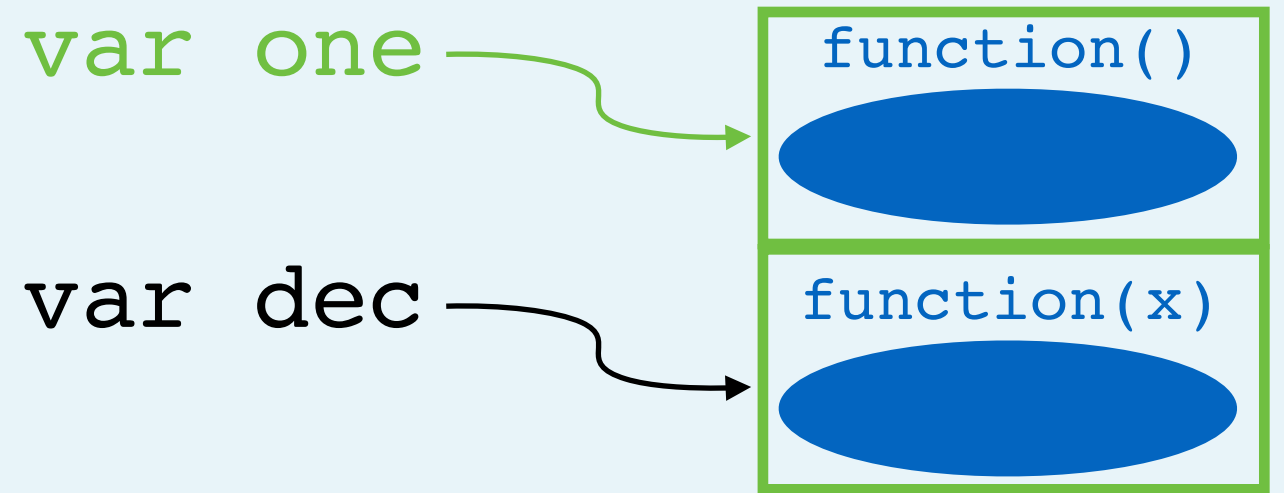




# Arguments Resolve First (2)

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

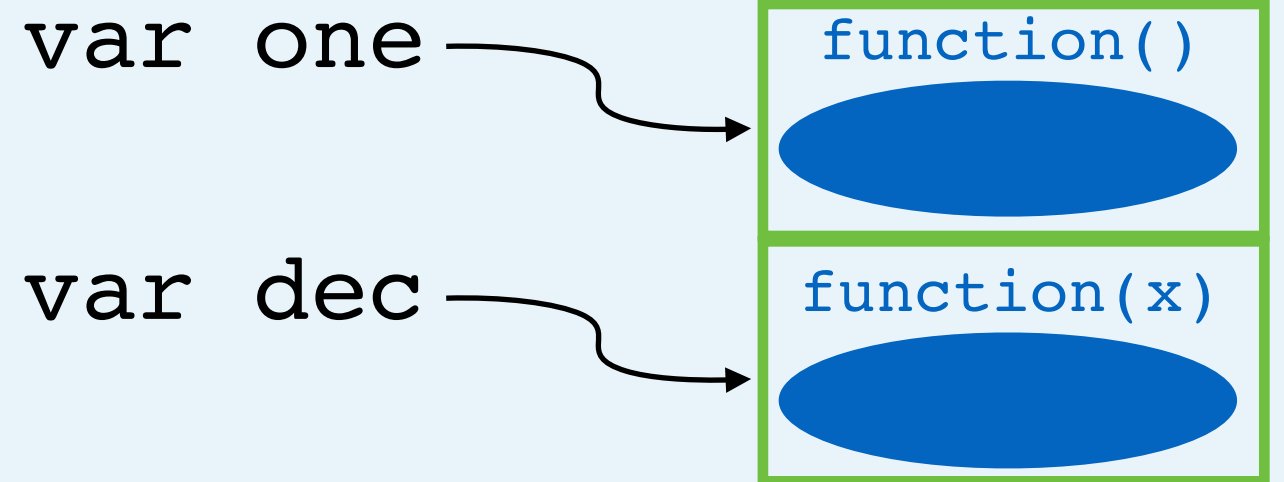
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one



# Arguments Resolve First (2)

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

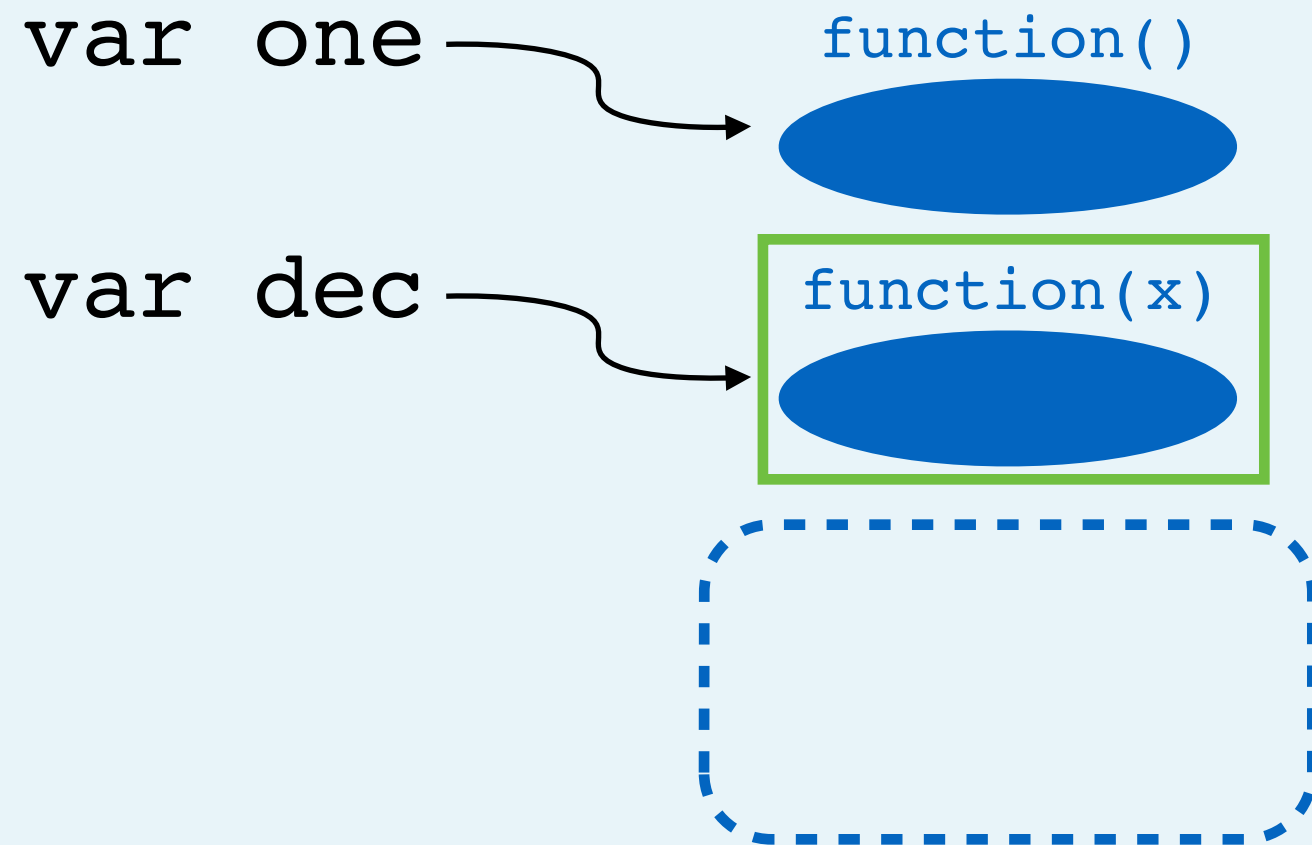
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function



# 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

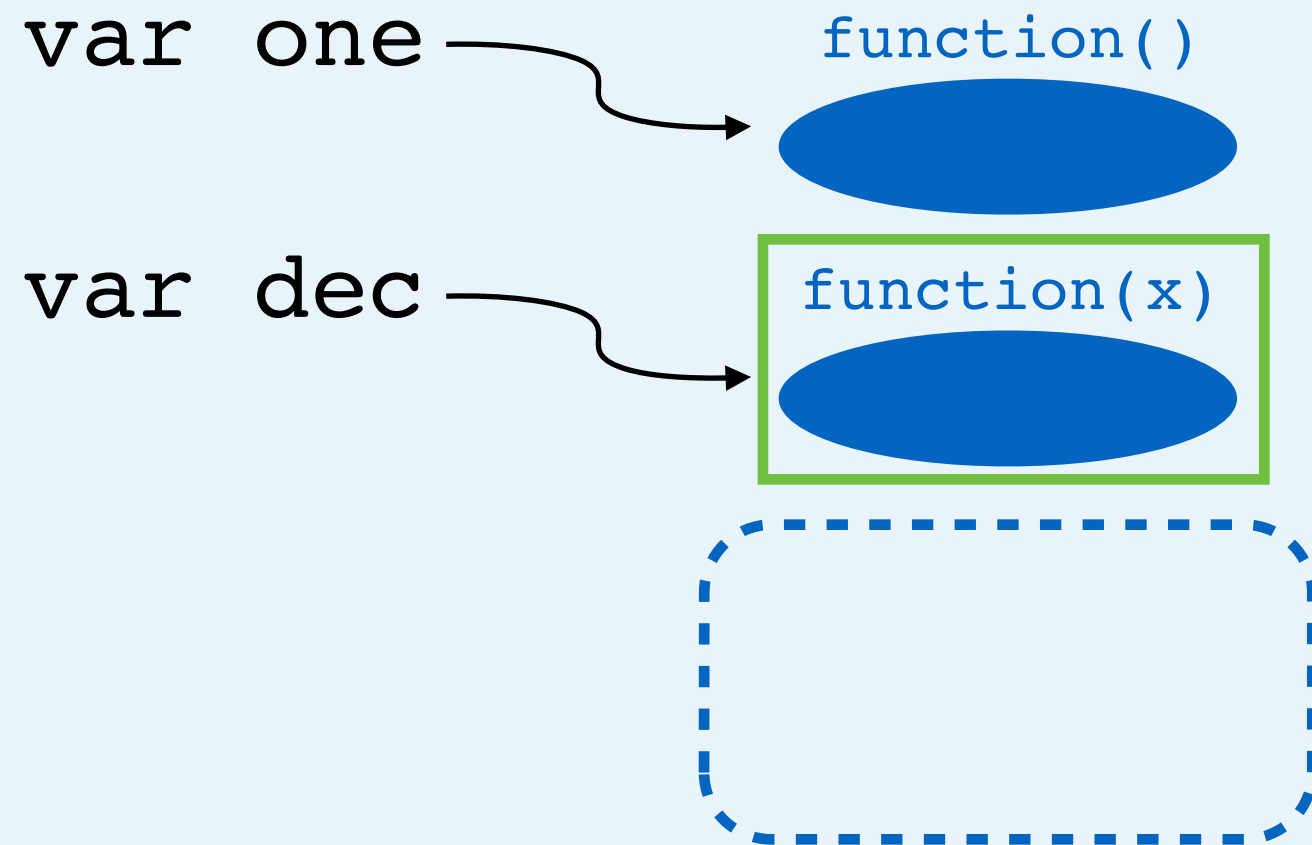


# Arguments Resolve First (2)

```
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



# 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

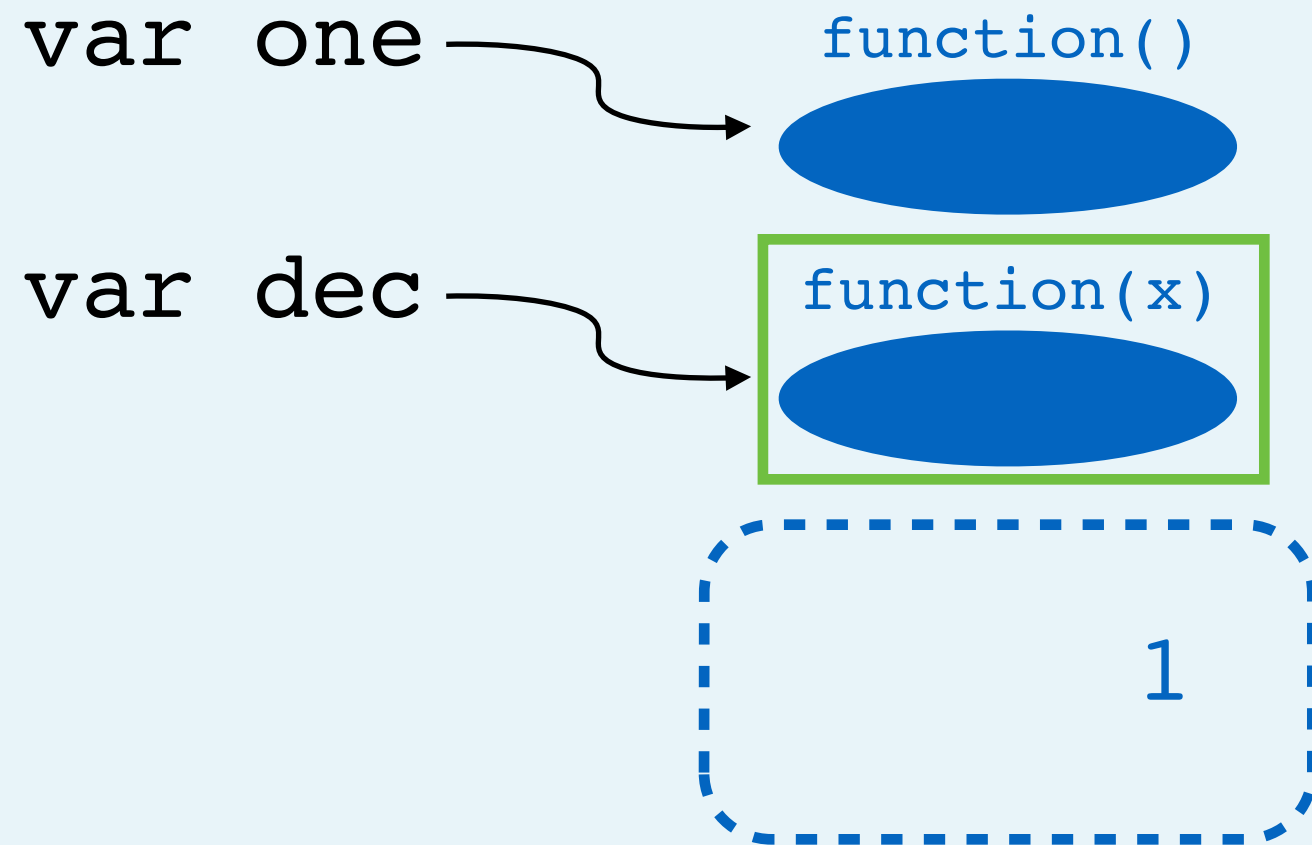
a. Look up value of one

b. Call function

a. Create scope

b. Return statement

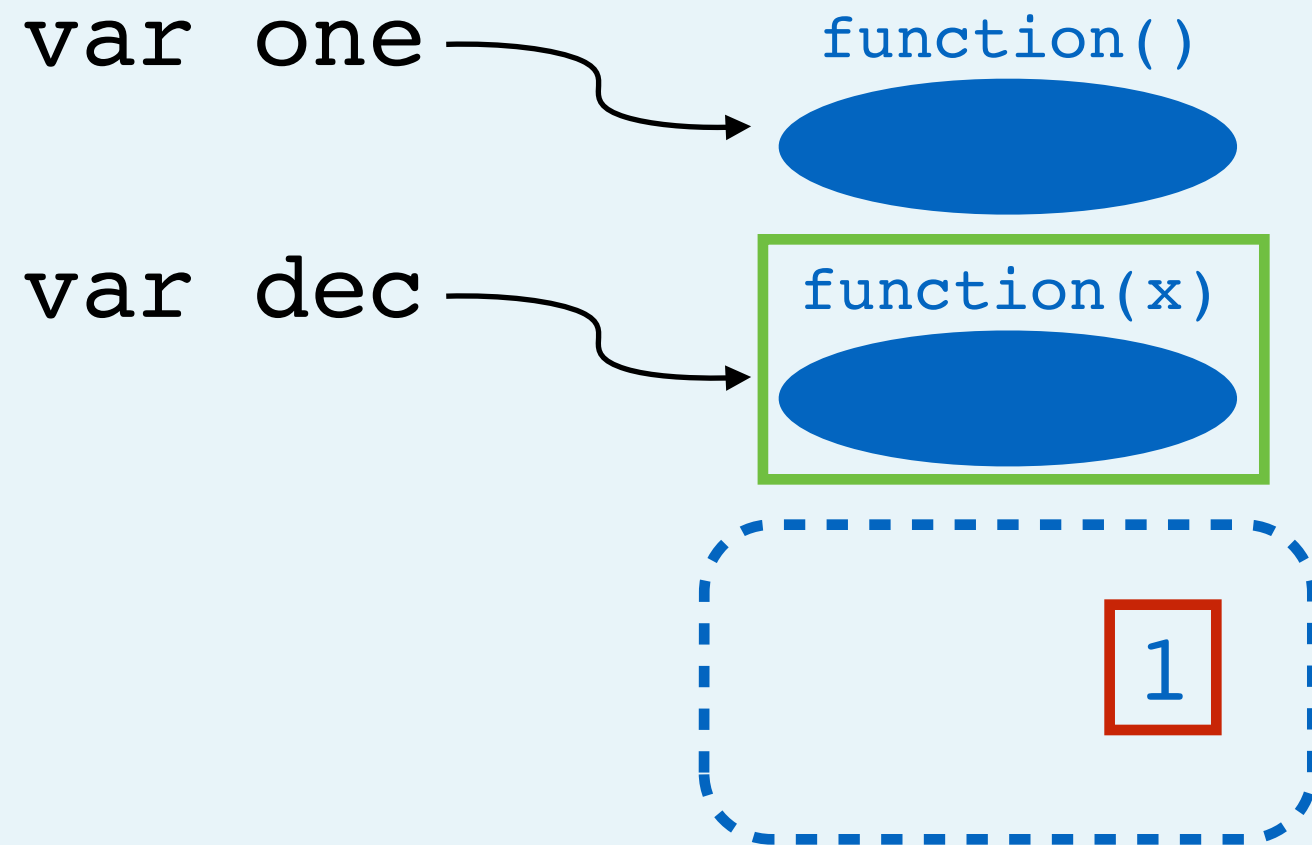
a. Create number



# Arguments Resolve First (2)

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

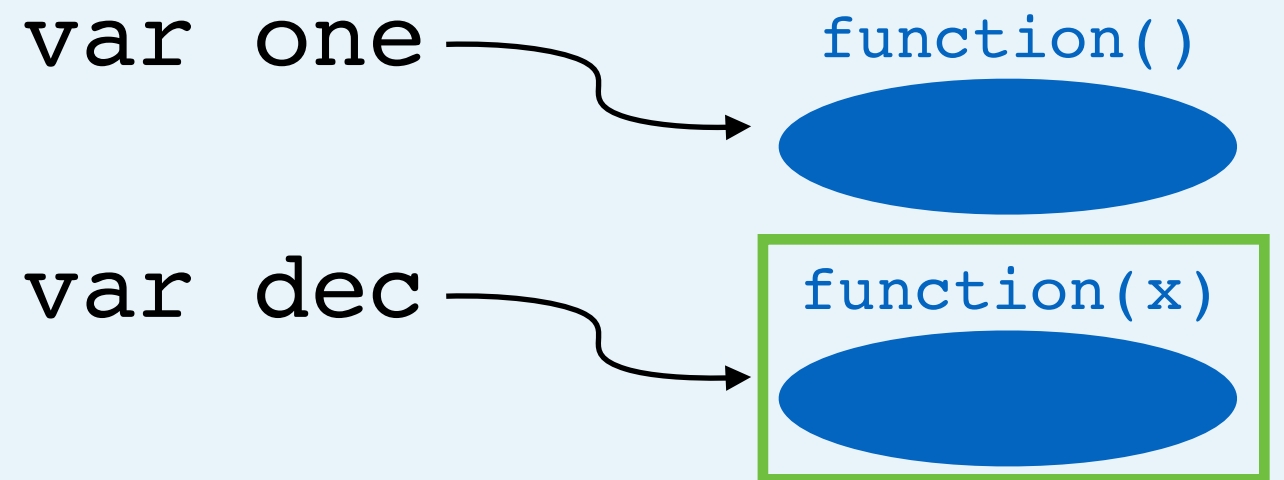
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value



# 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

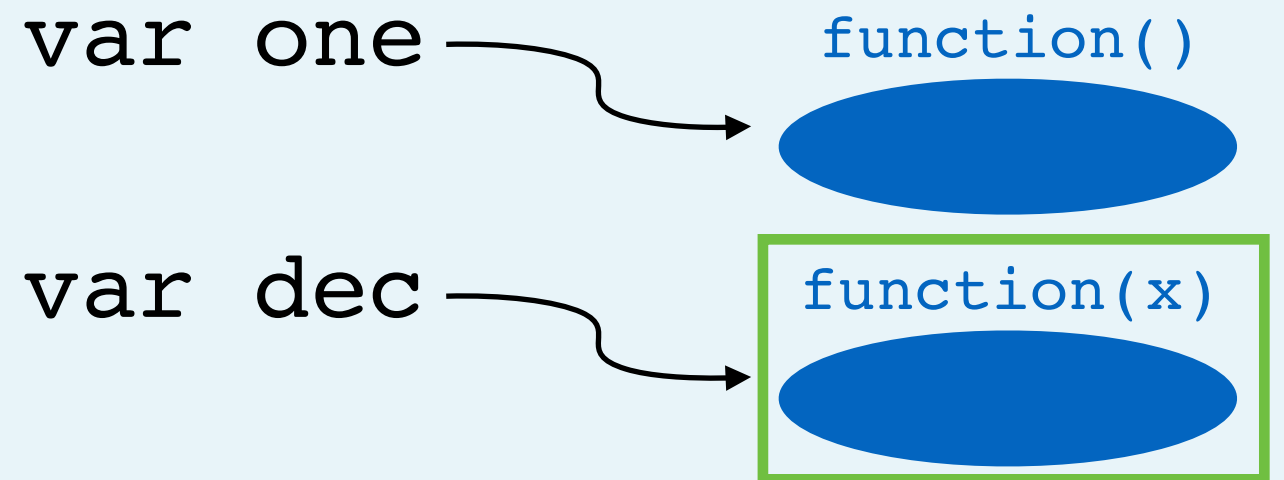


1

# 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



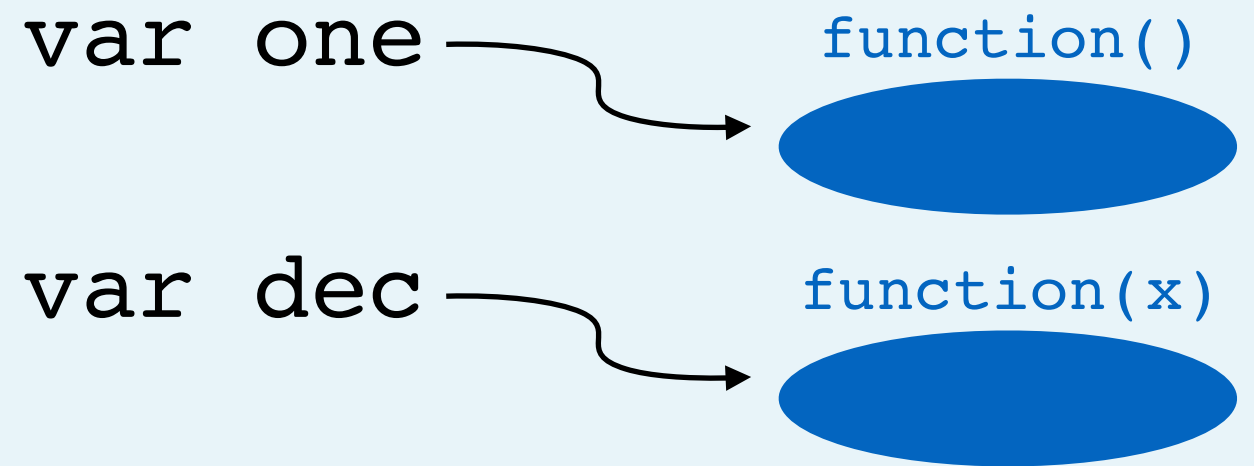
1



# Arguments Resolve First (2)

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

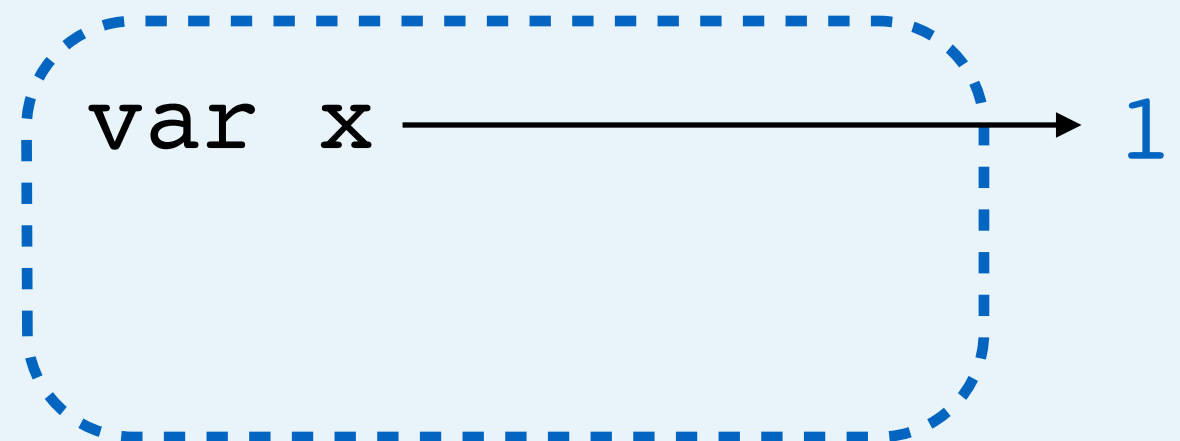
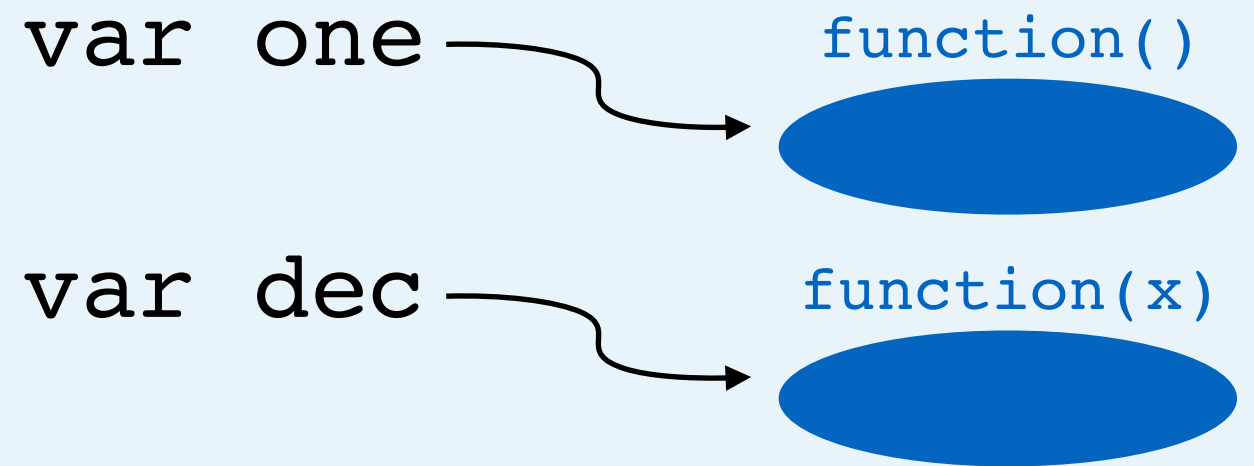
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
  - c. Garbage collect scope
- d. Call function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



# Arguments Resolve First (2)

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

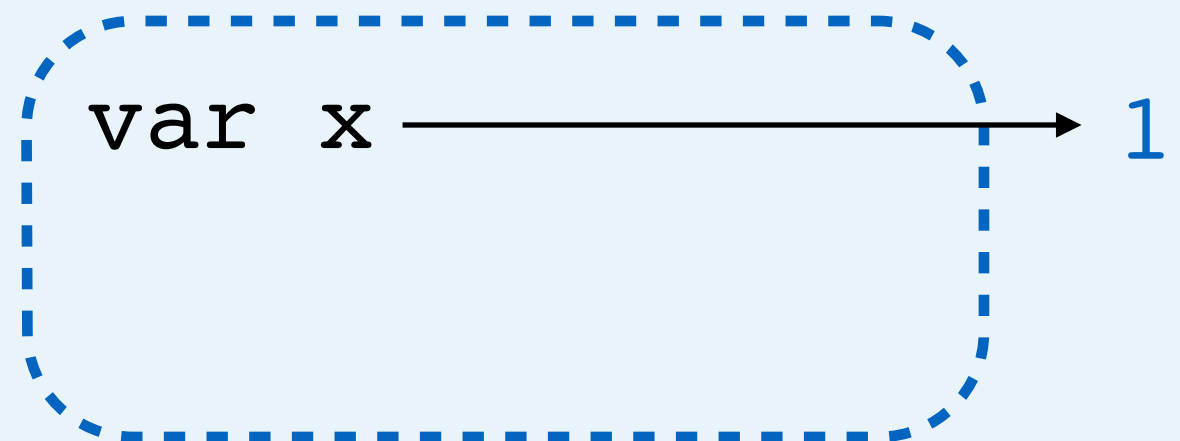
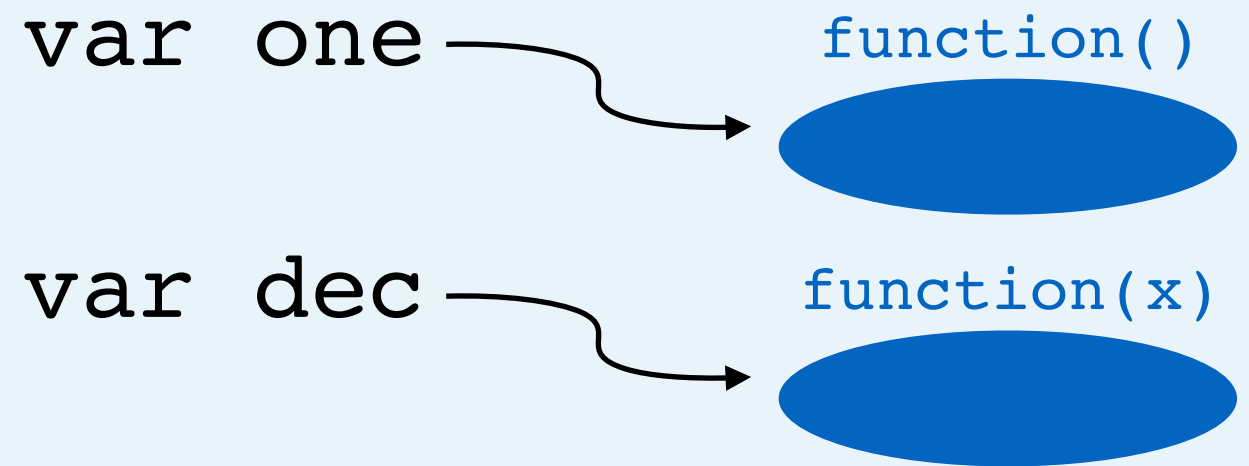
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
  - c. Garbage collect scope
- d. Call function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

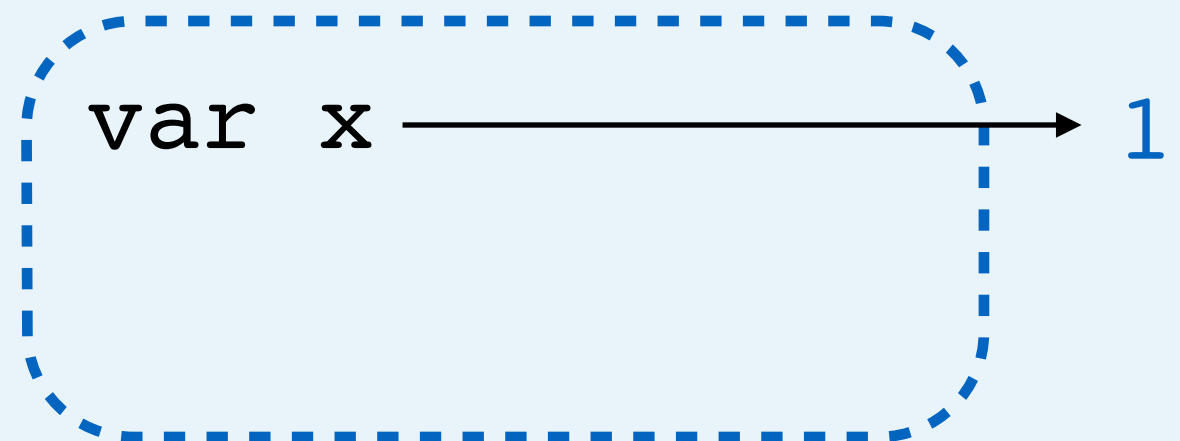
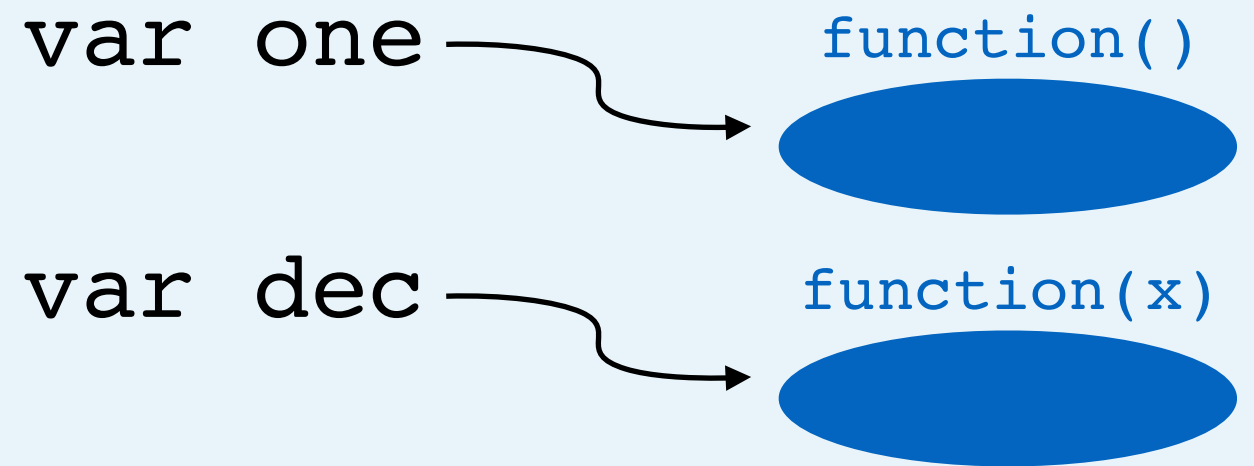
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call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

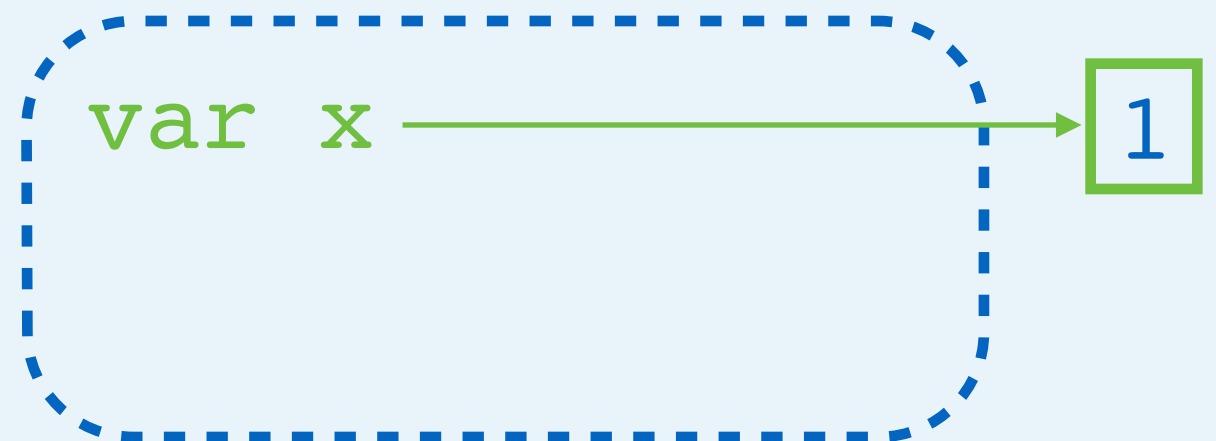
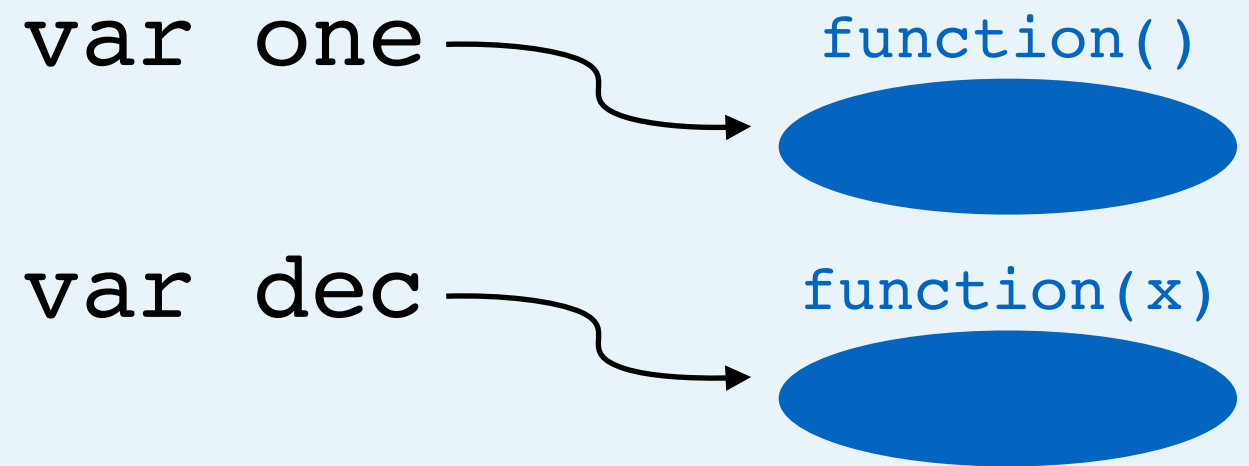
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
  - c. Garbage collect scope
- d. Call function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

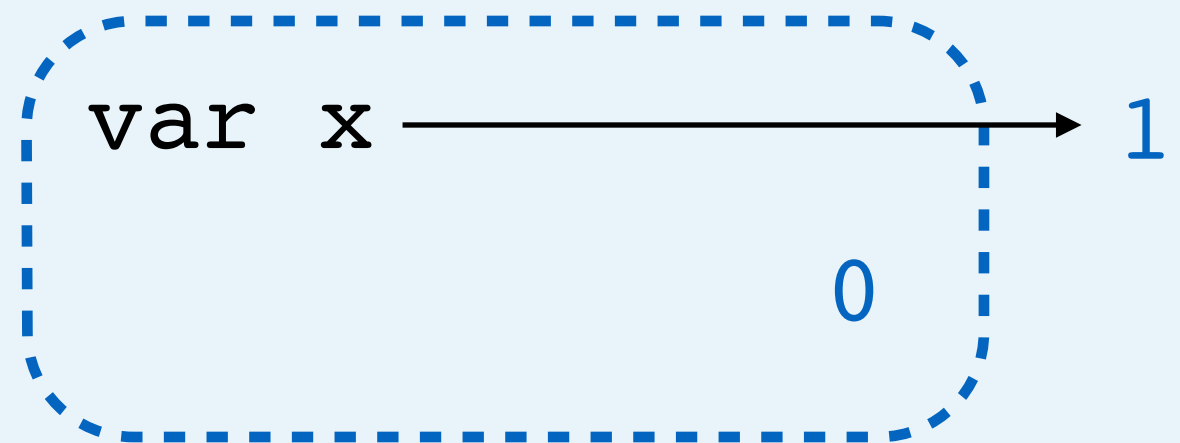
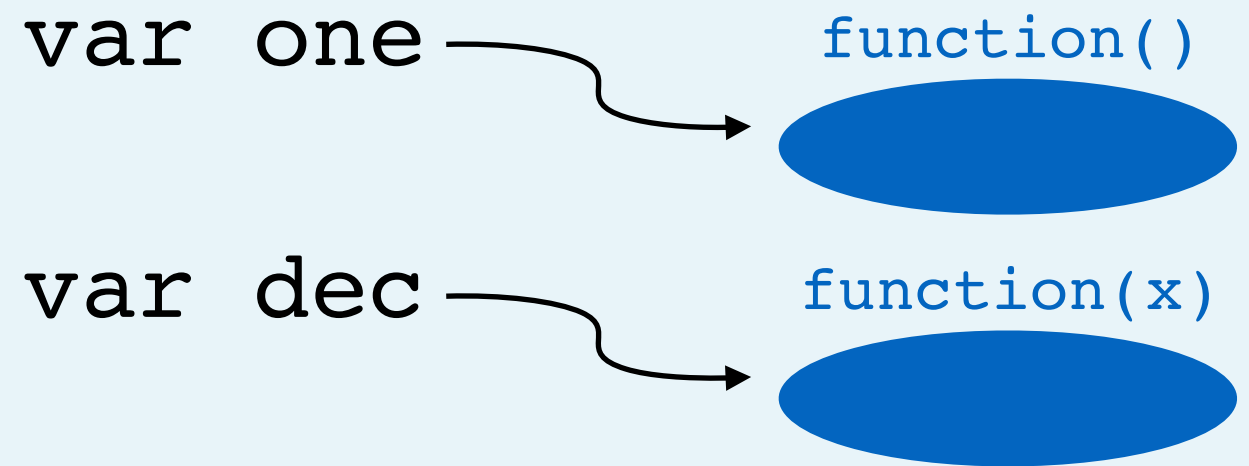
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
  - c. Garbage collect scope
- d. Call function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

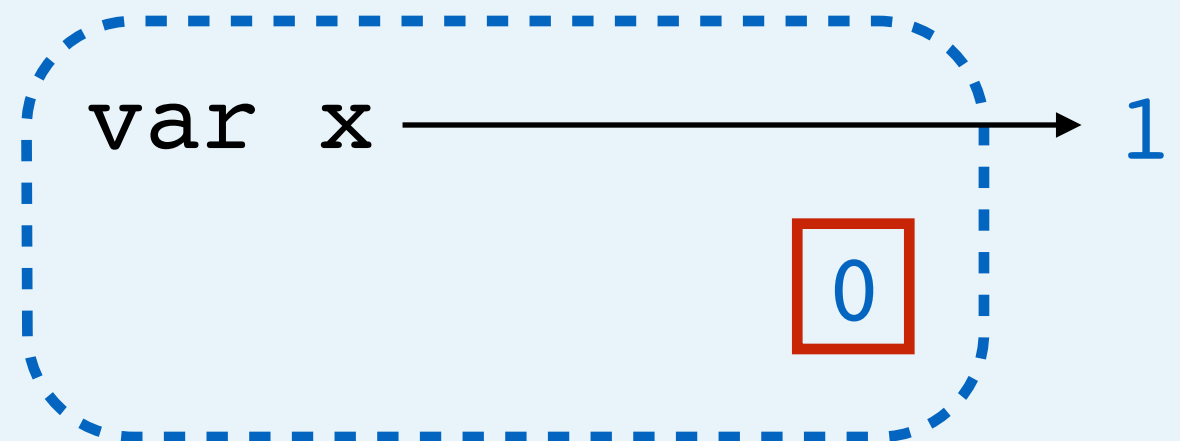
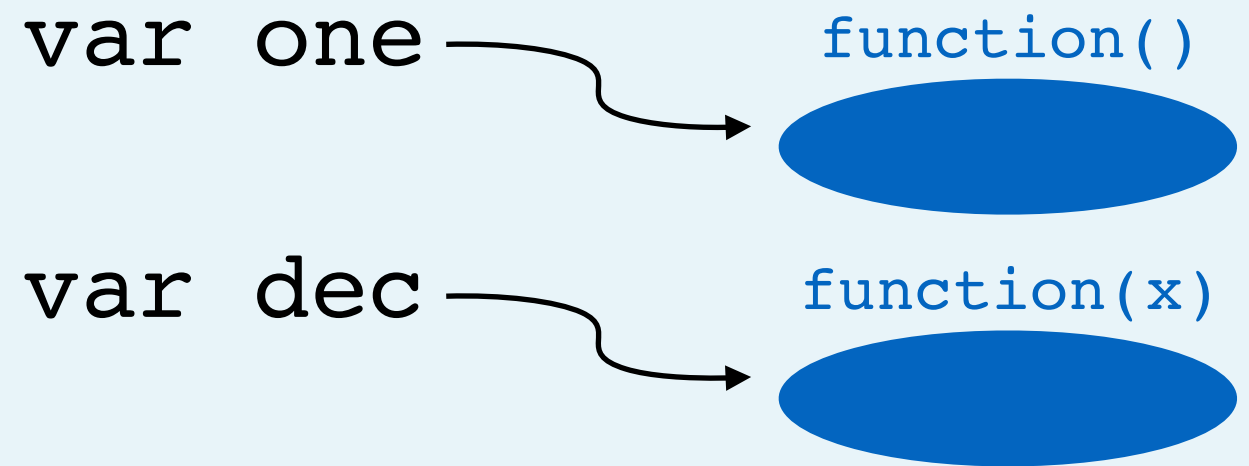
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
  - c. Garbage collect scope
- d. Call function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

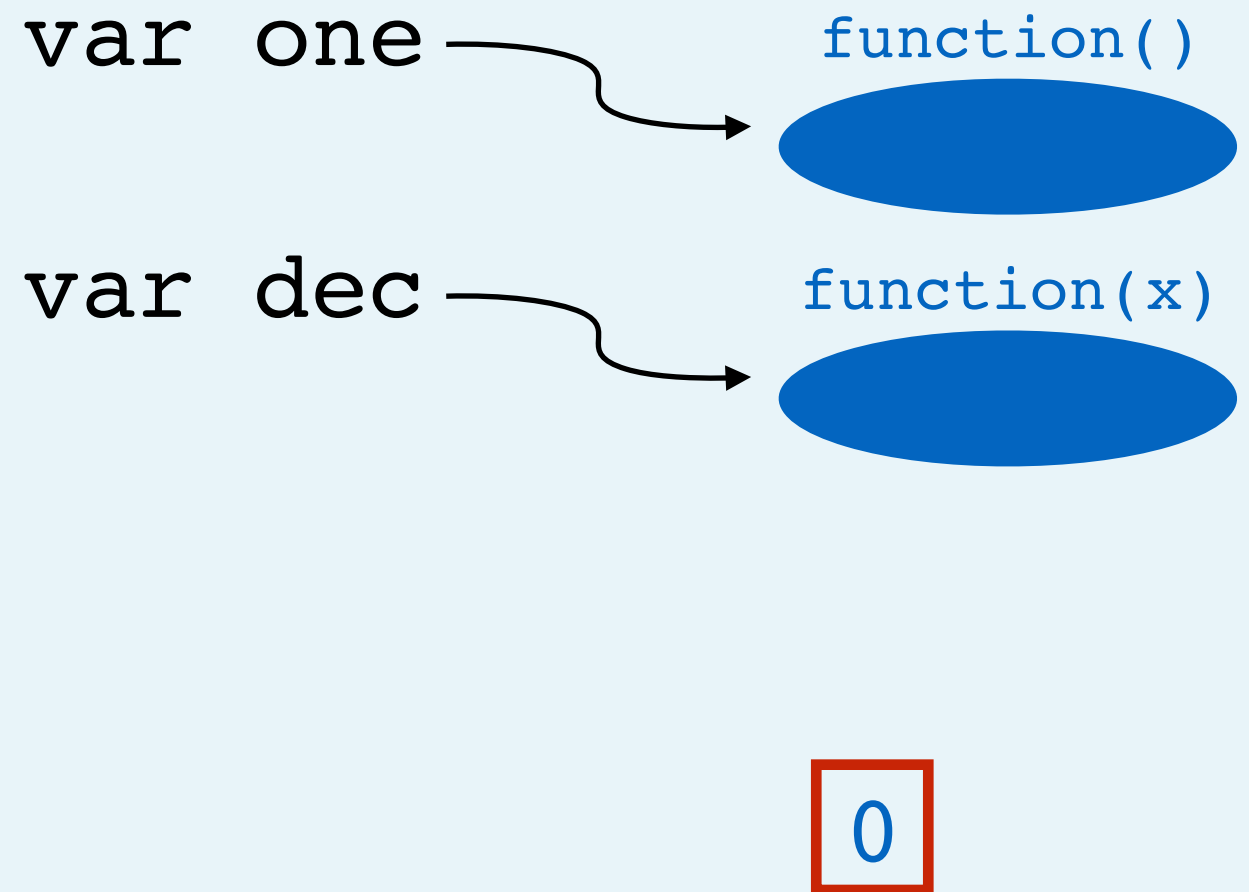
- a. Assignment
  - a. Evaluate right side
  - b. Look up value of dec
  - c. Resolve argument
    - a. Look up value of one
    - b. Call function
      - a. Create scope
      - b. Return statement
        - a. Create number
        - b. Mark as return value
    - c. Garbage collect scope
  - d. Call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



# 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

