

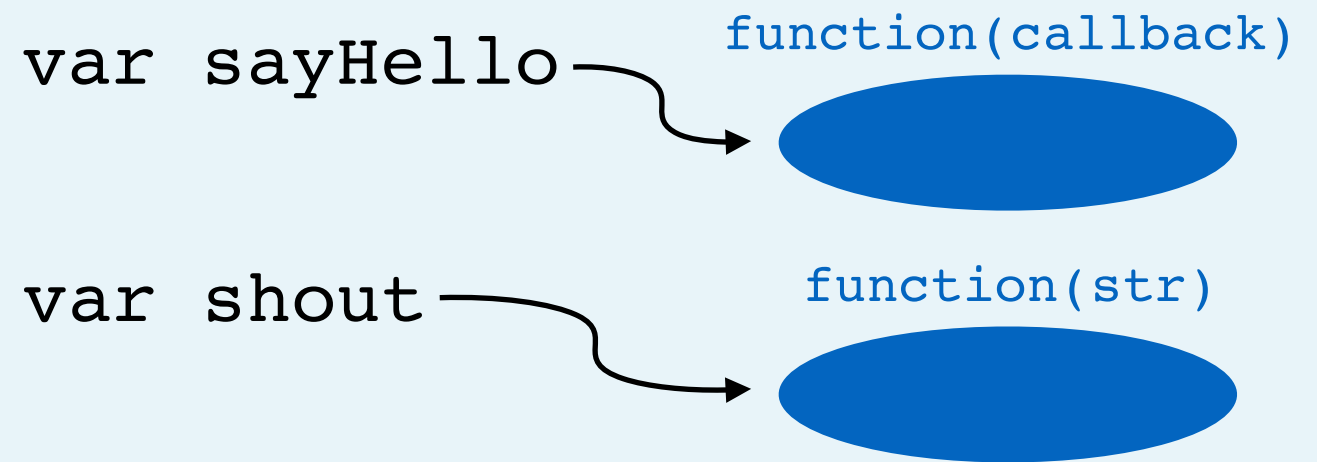
JSDN

(by example)

Functions as Arguments

Callbacks

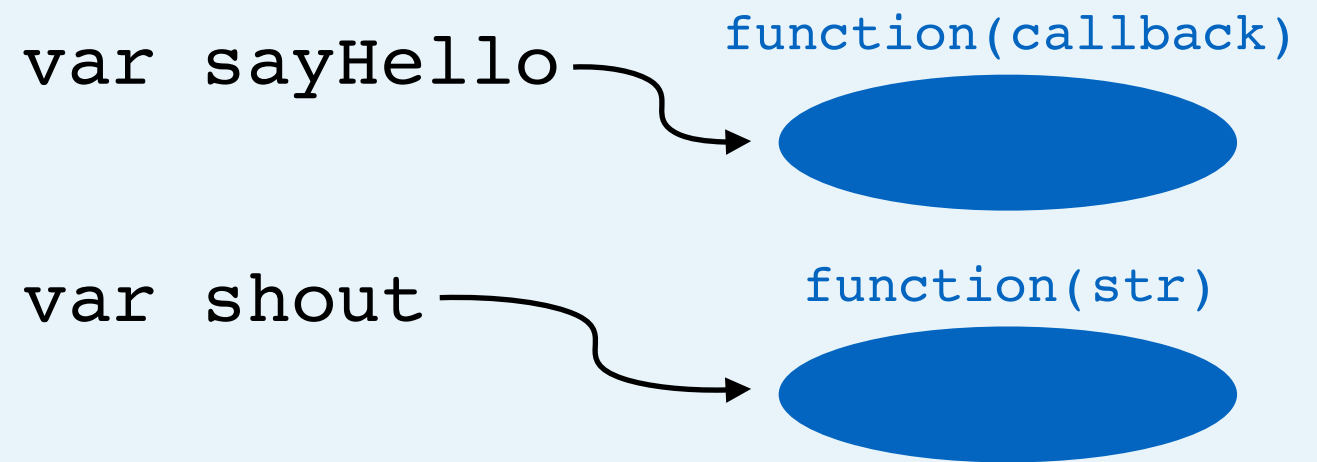
```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

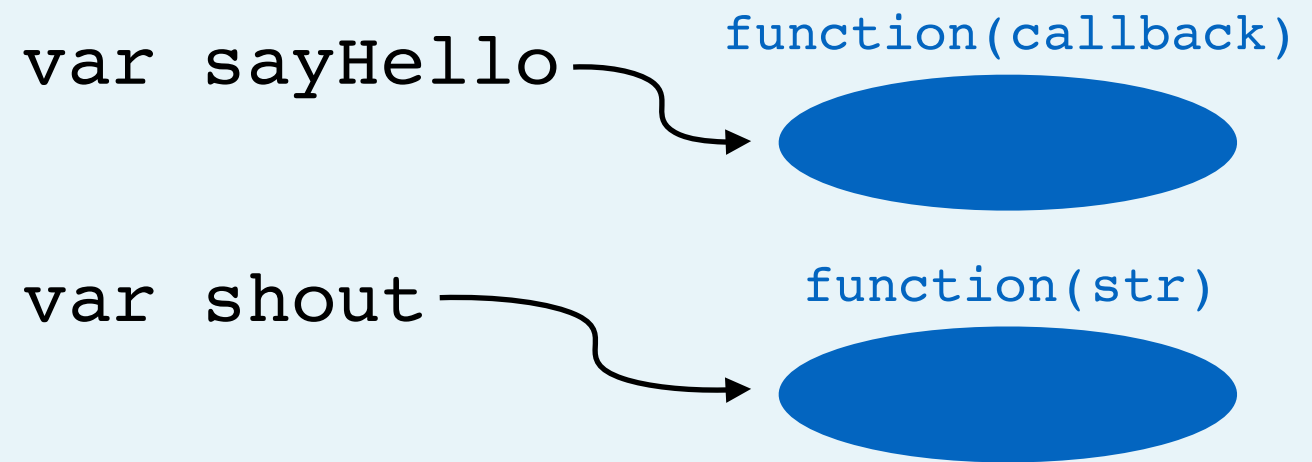
a. Assignment



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

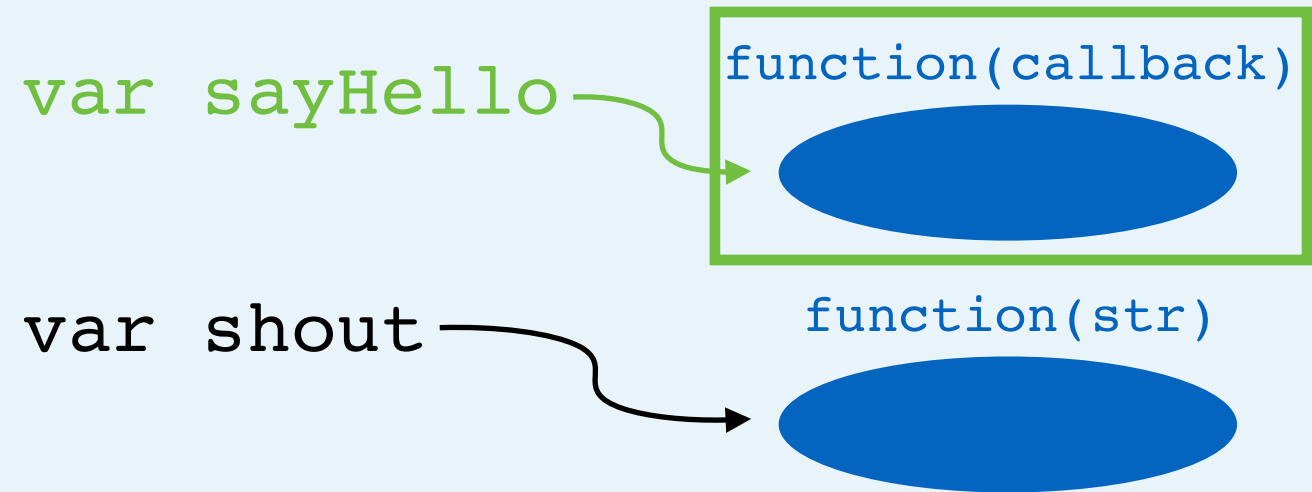
- a. Assignment
 - a. Evaluate right side



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

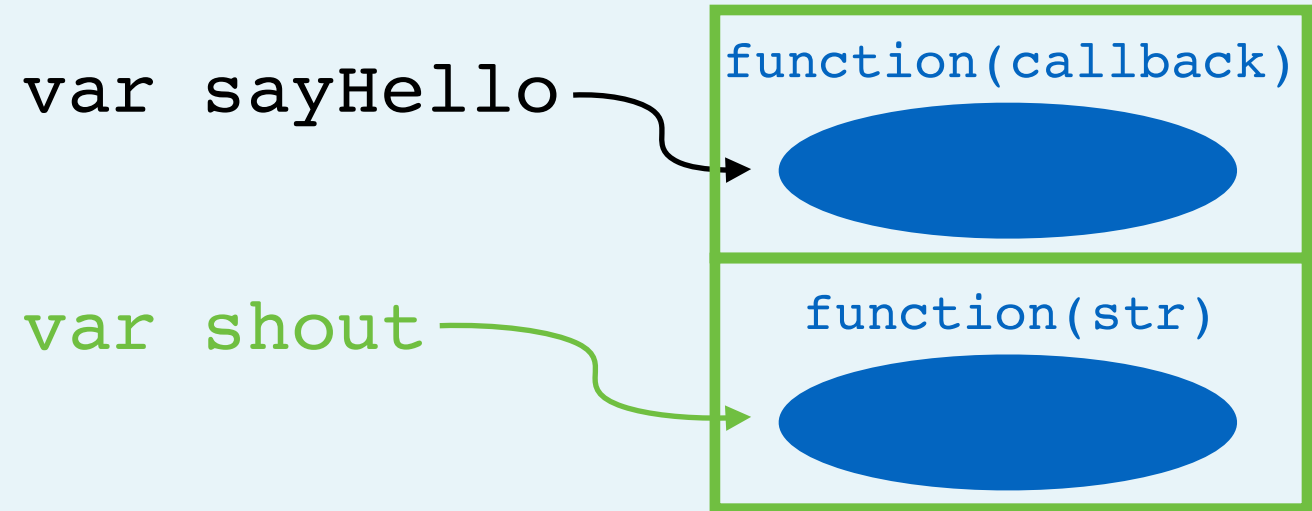
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

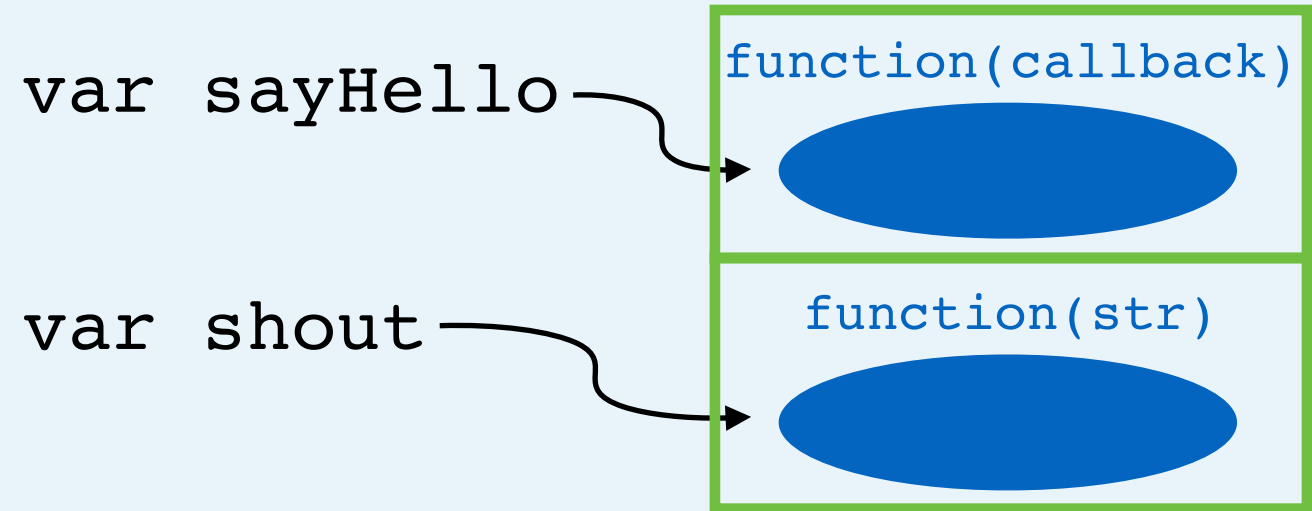
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

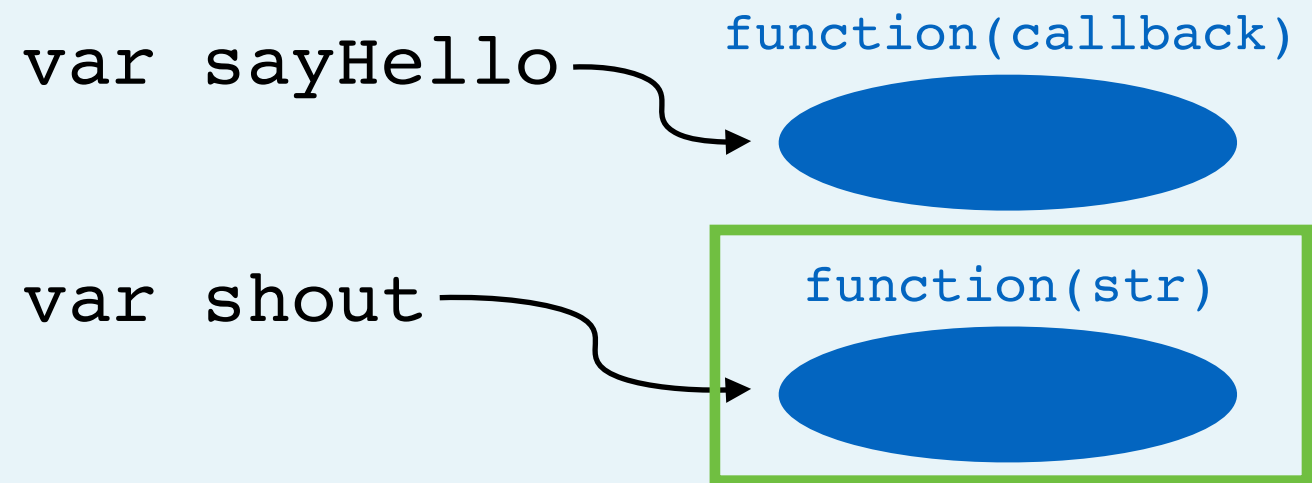
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

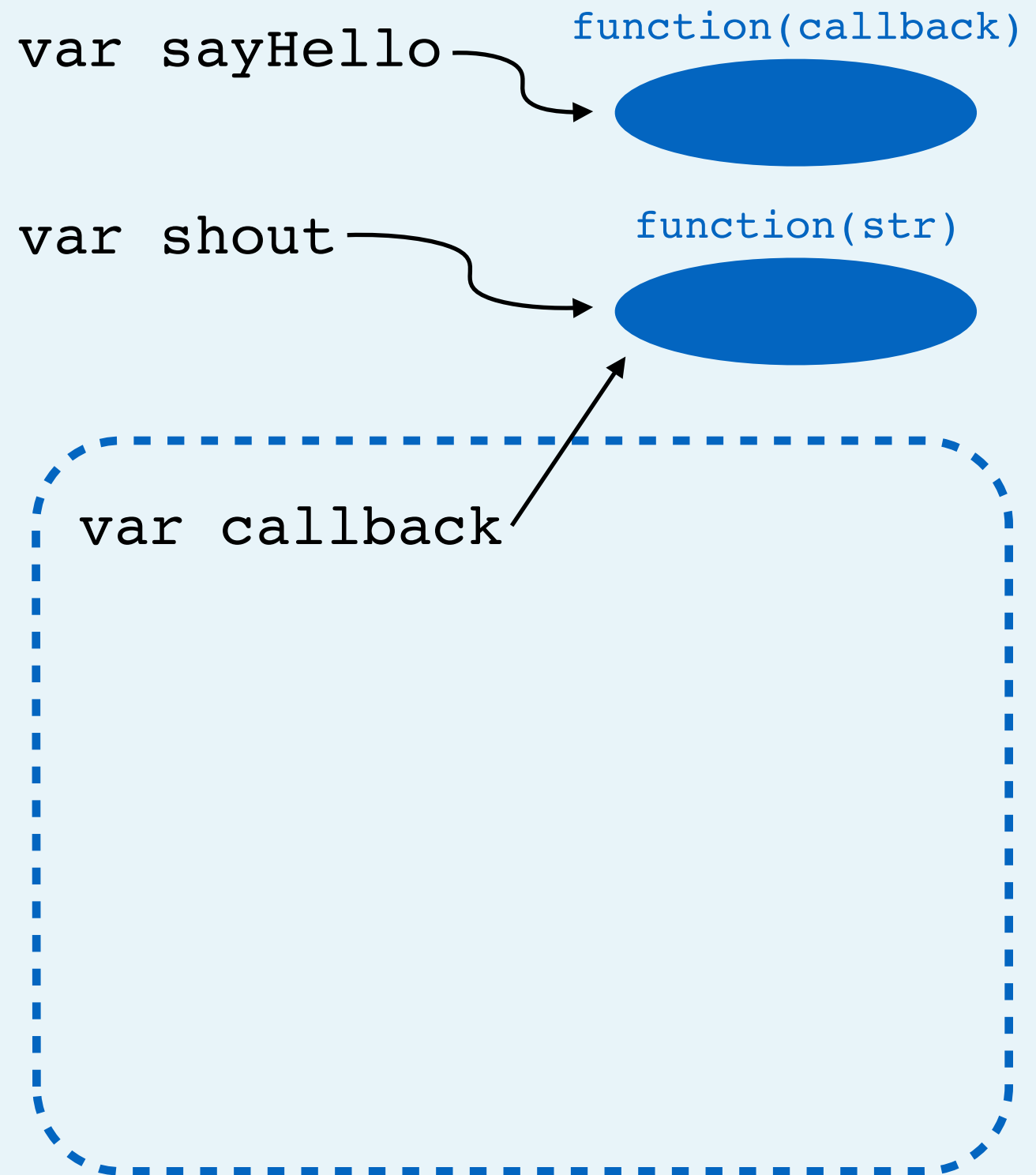
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

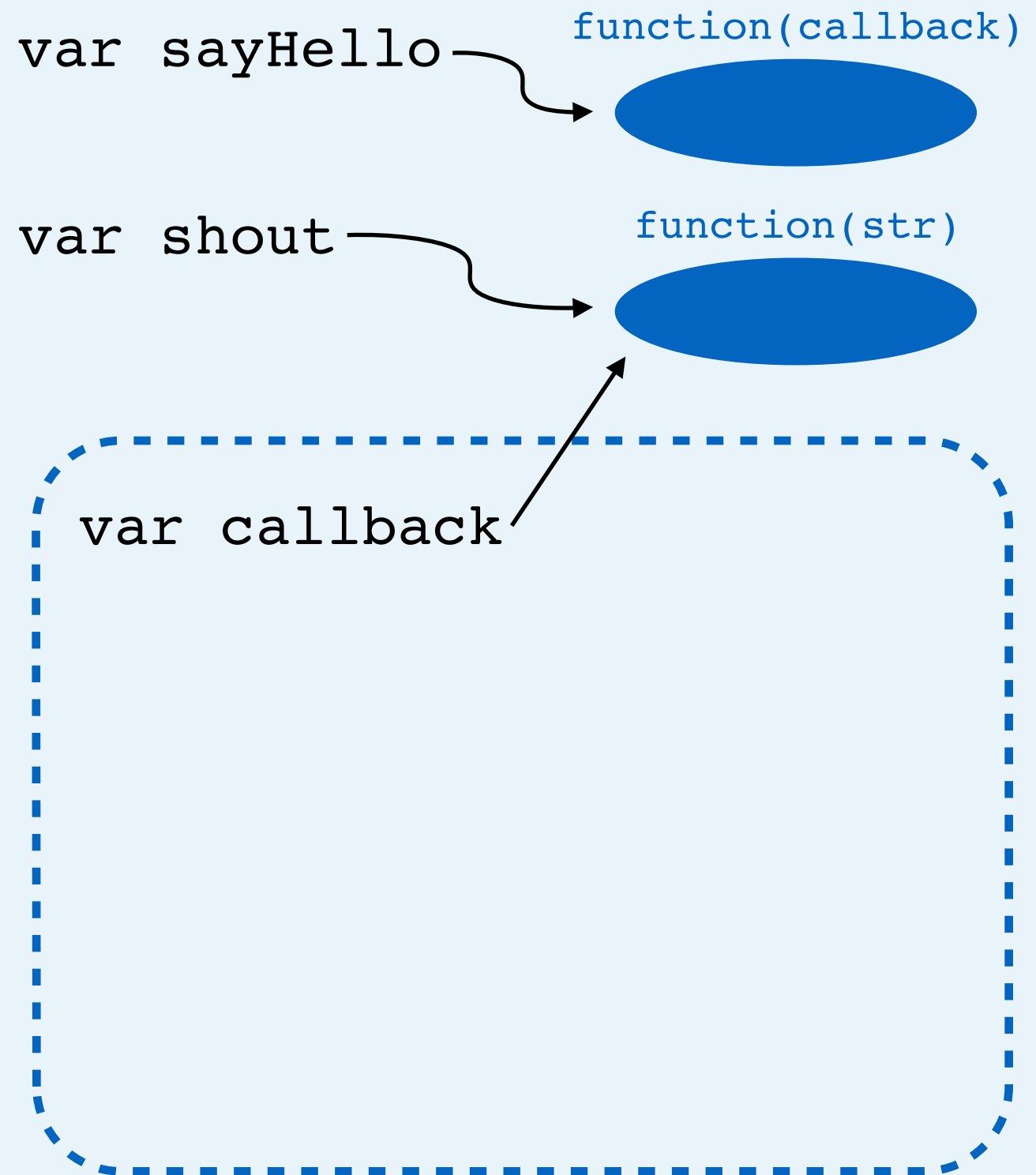
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

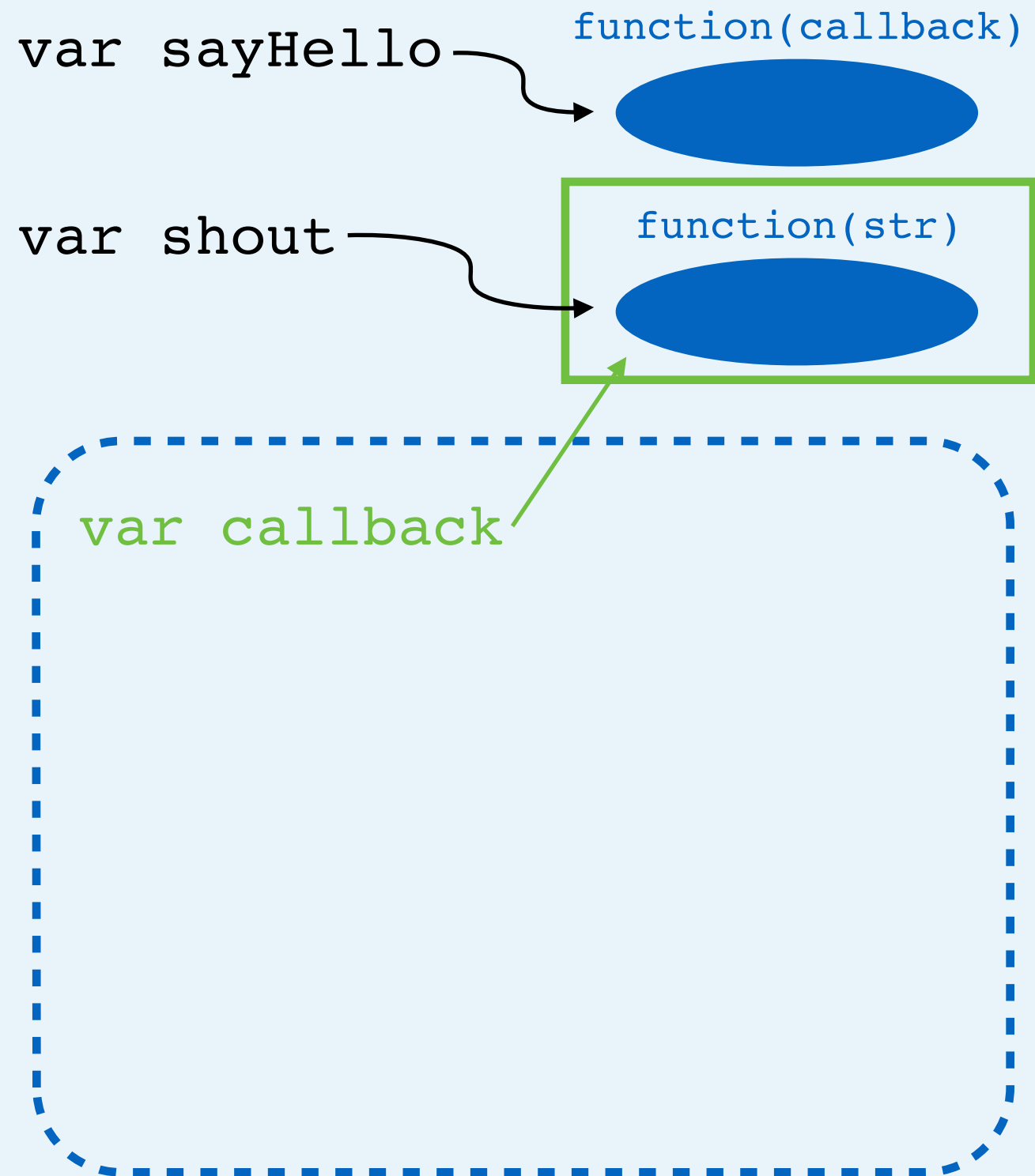
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

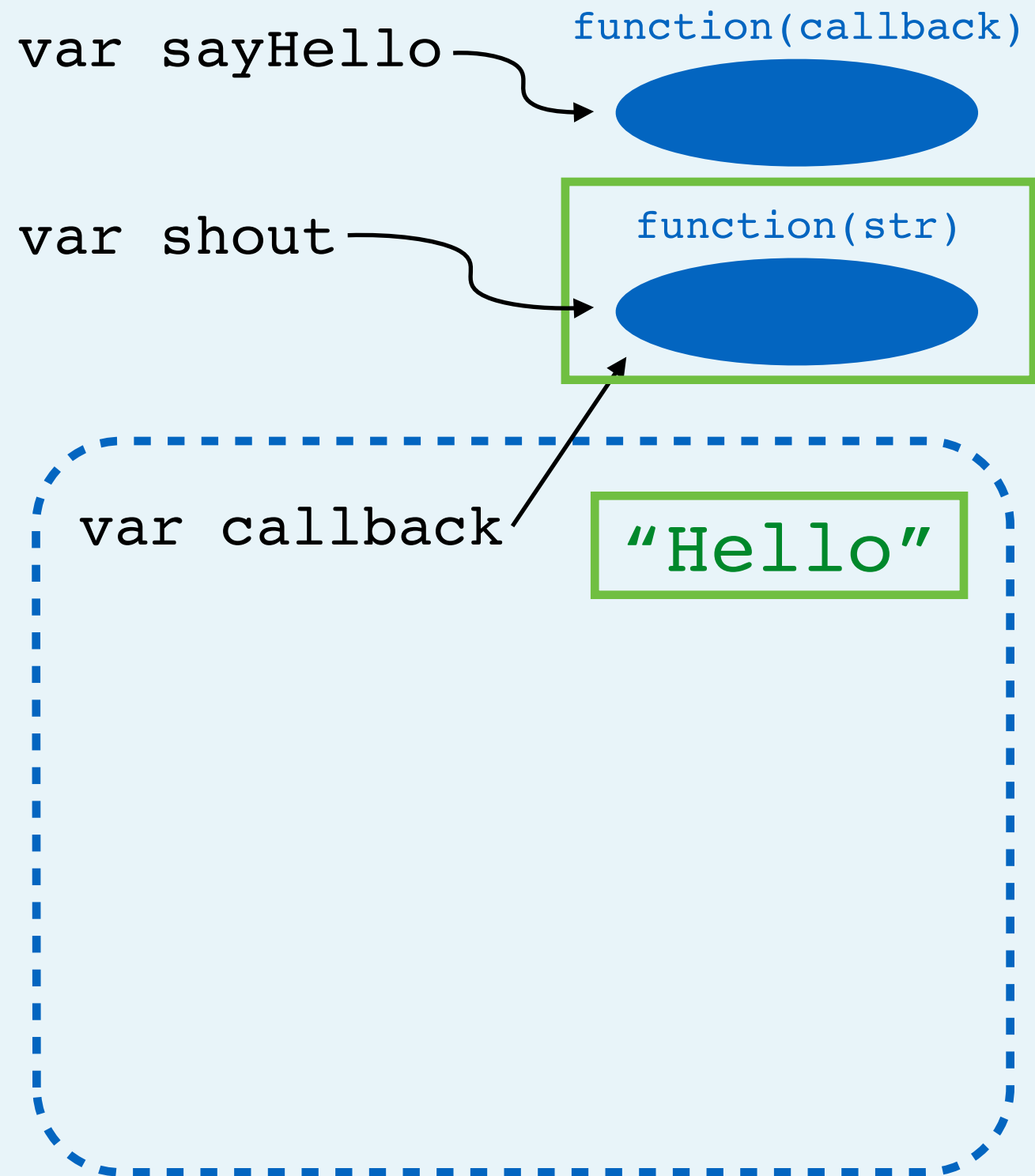
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of callback



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

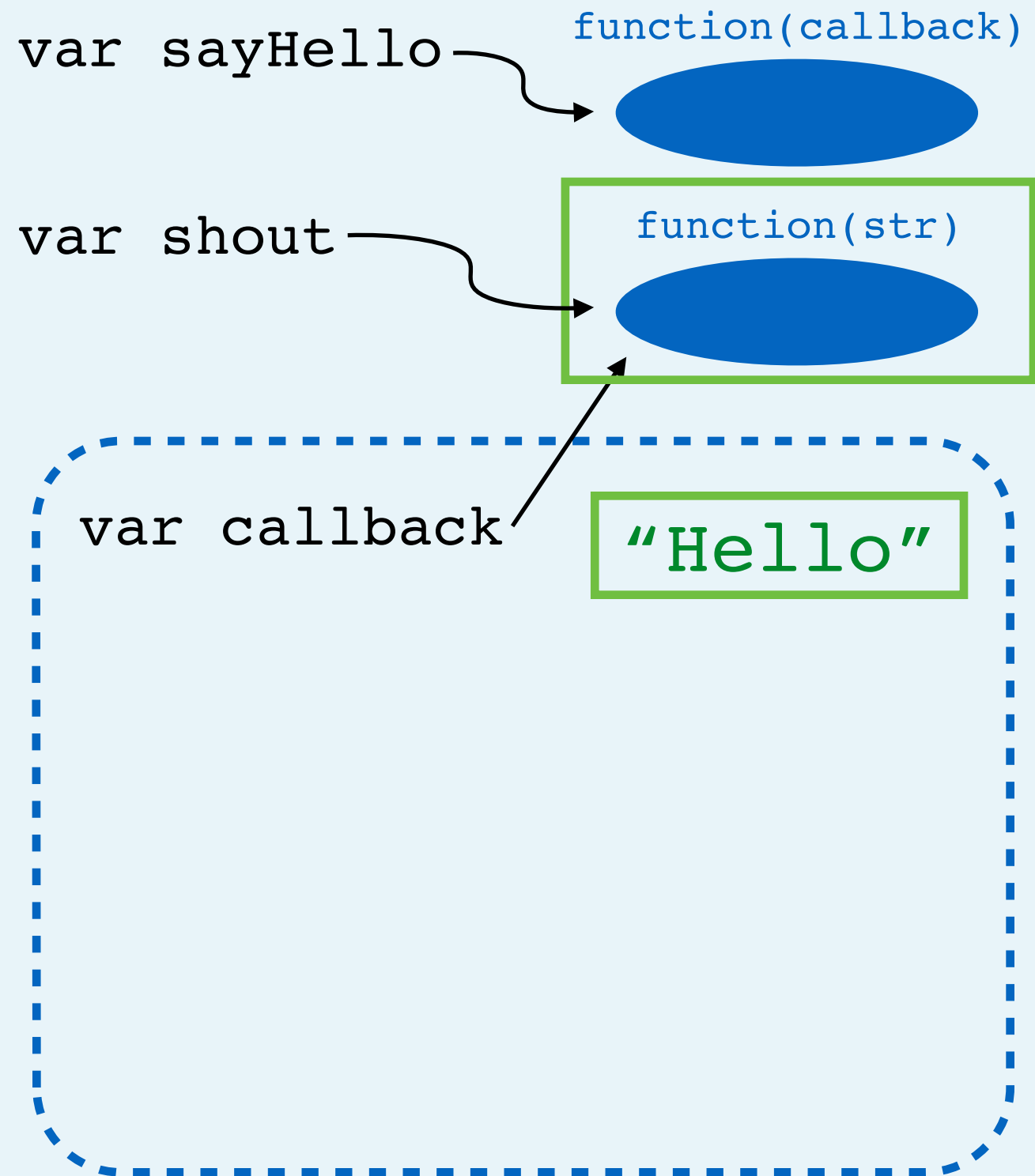
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

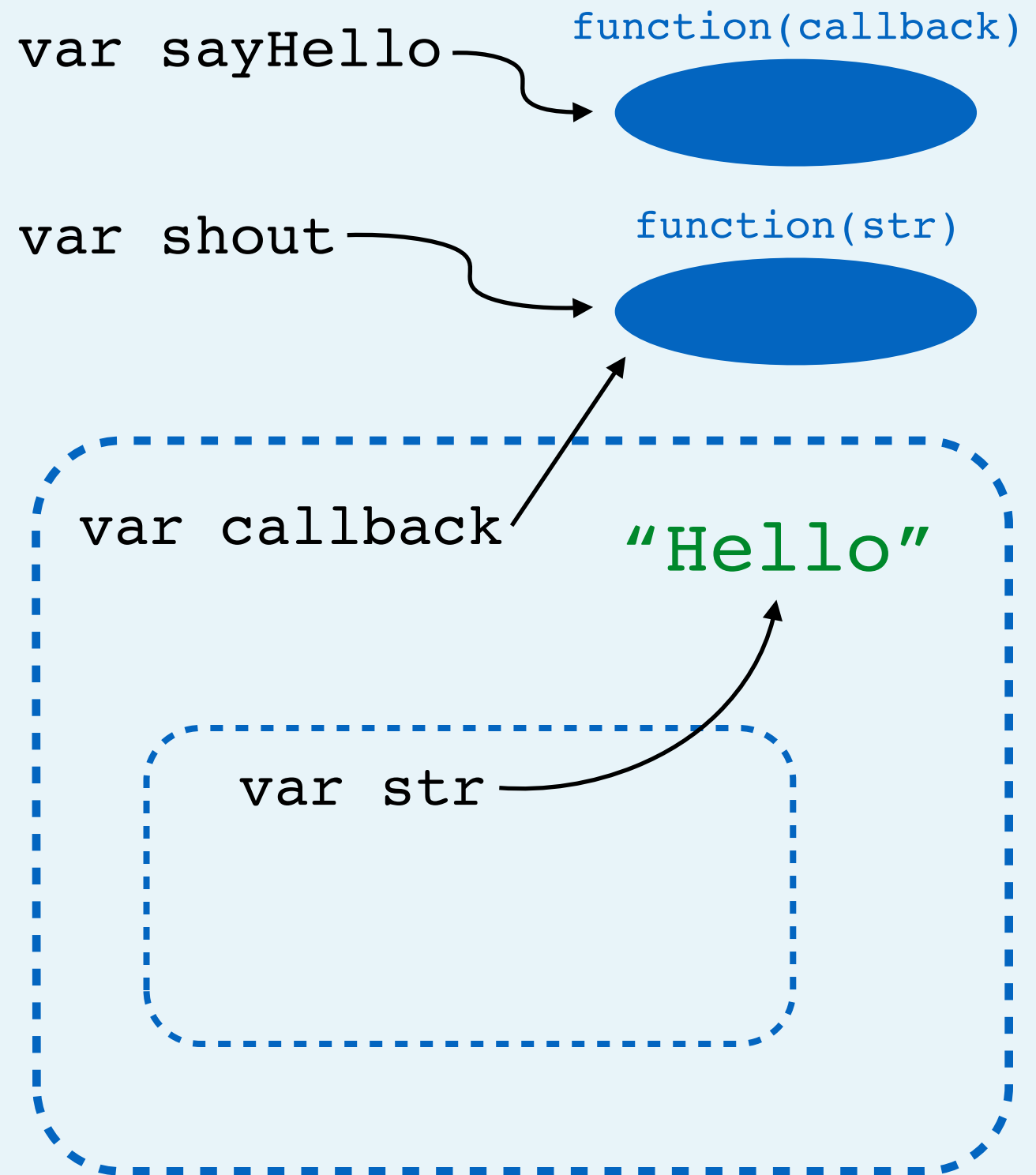
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

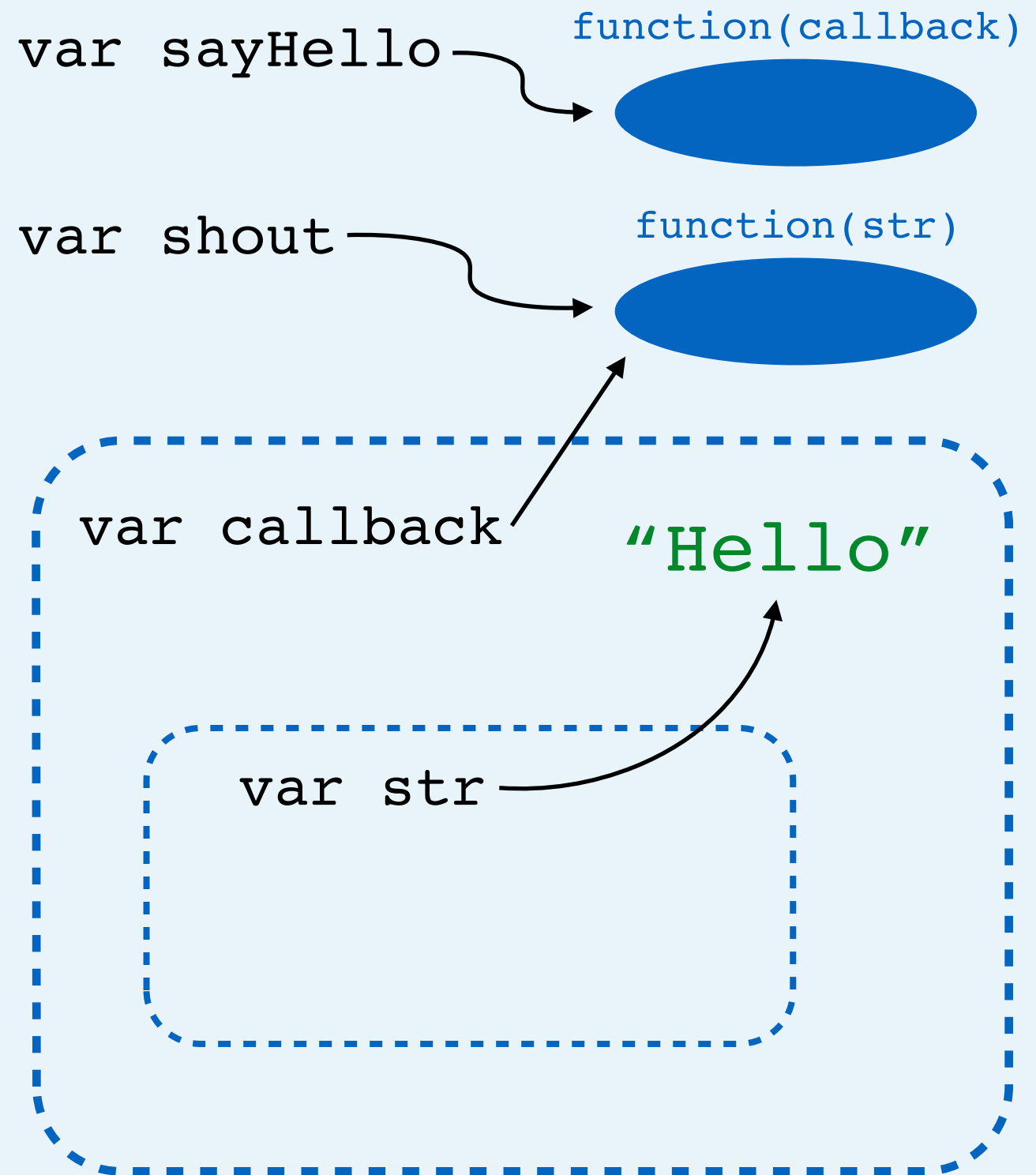
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

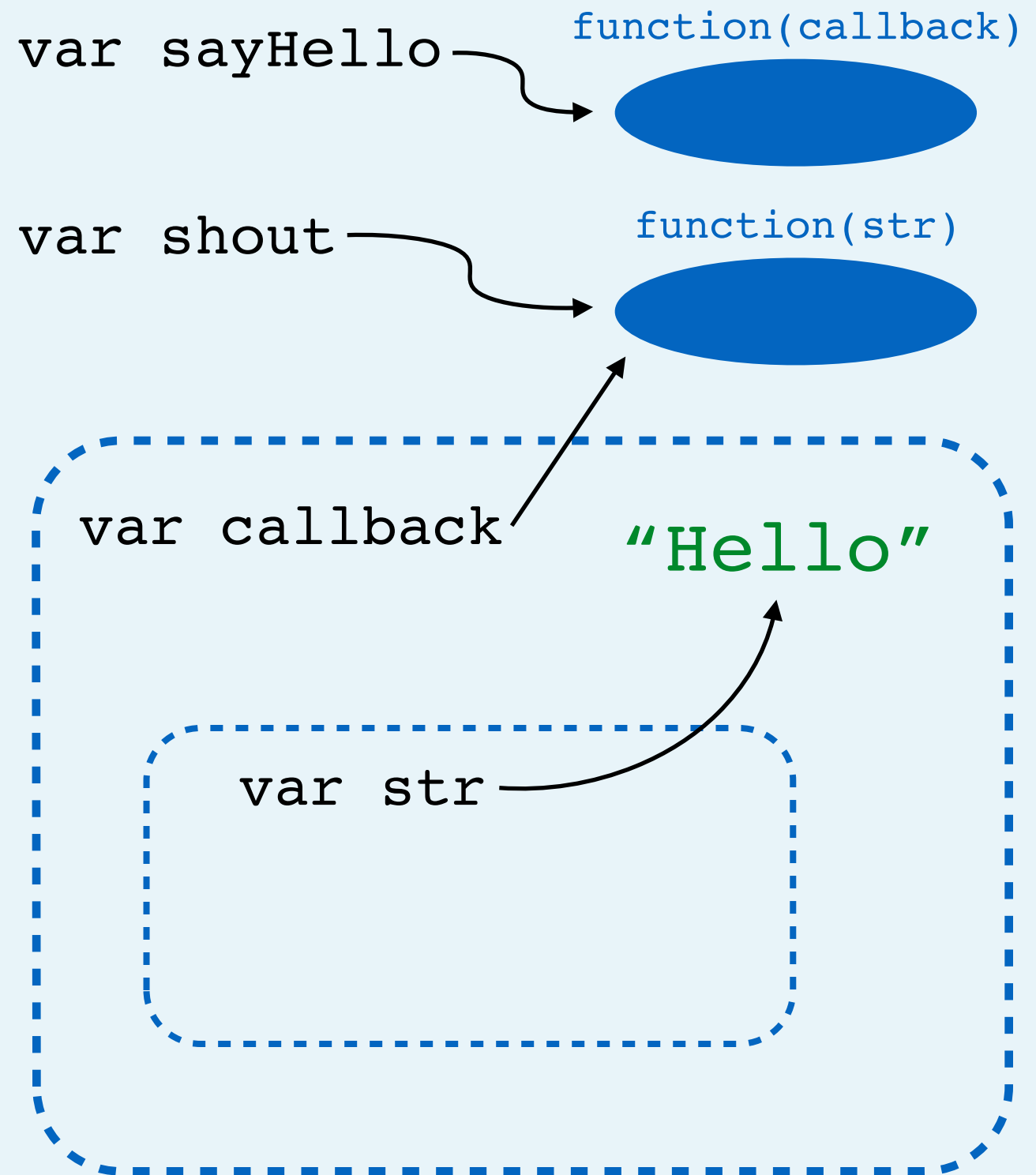
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

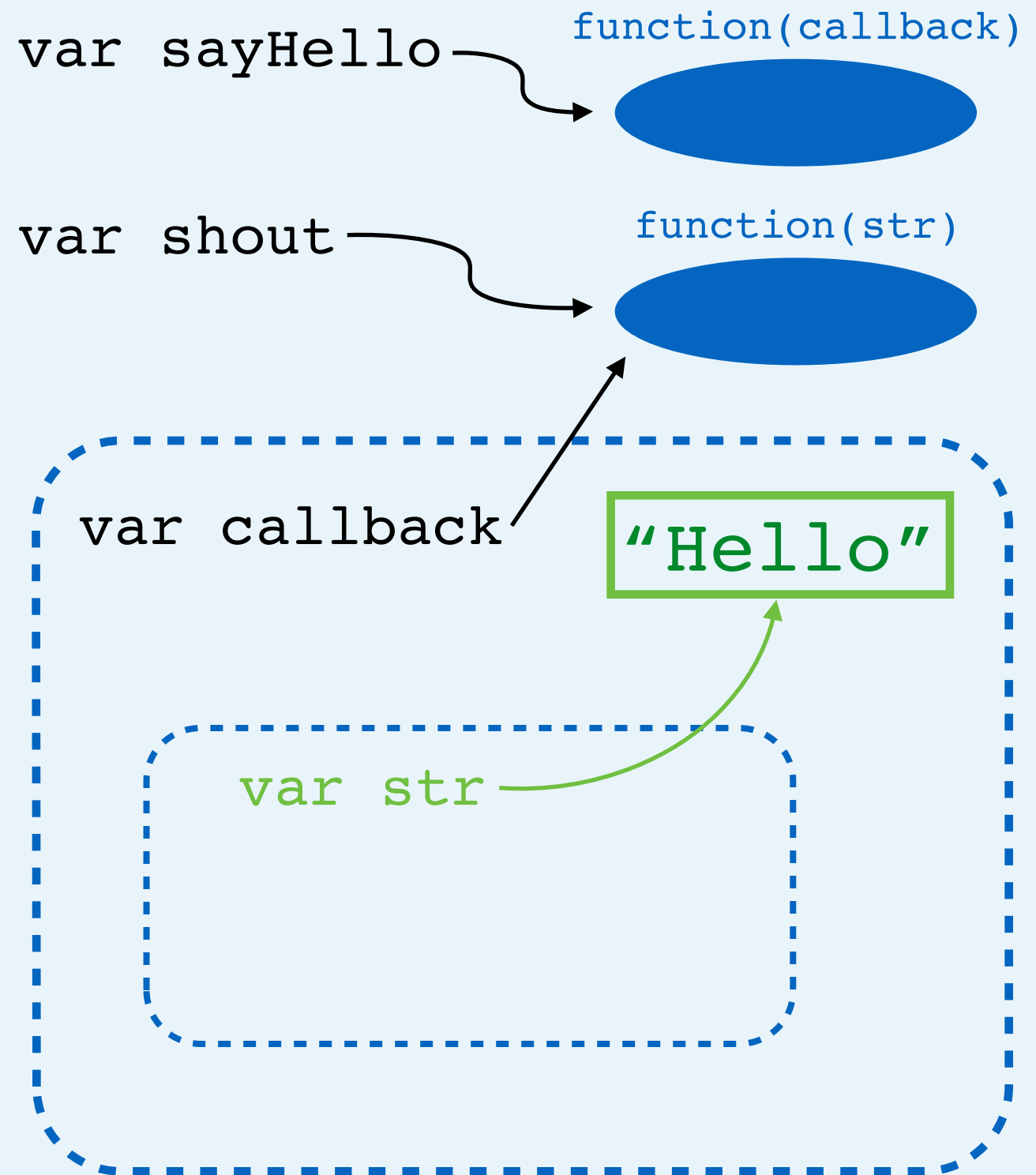
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

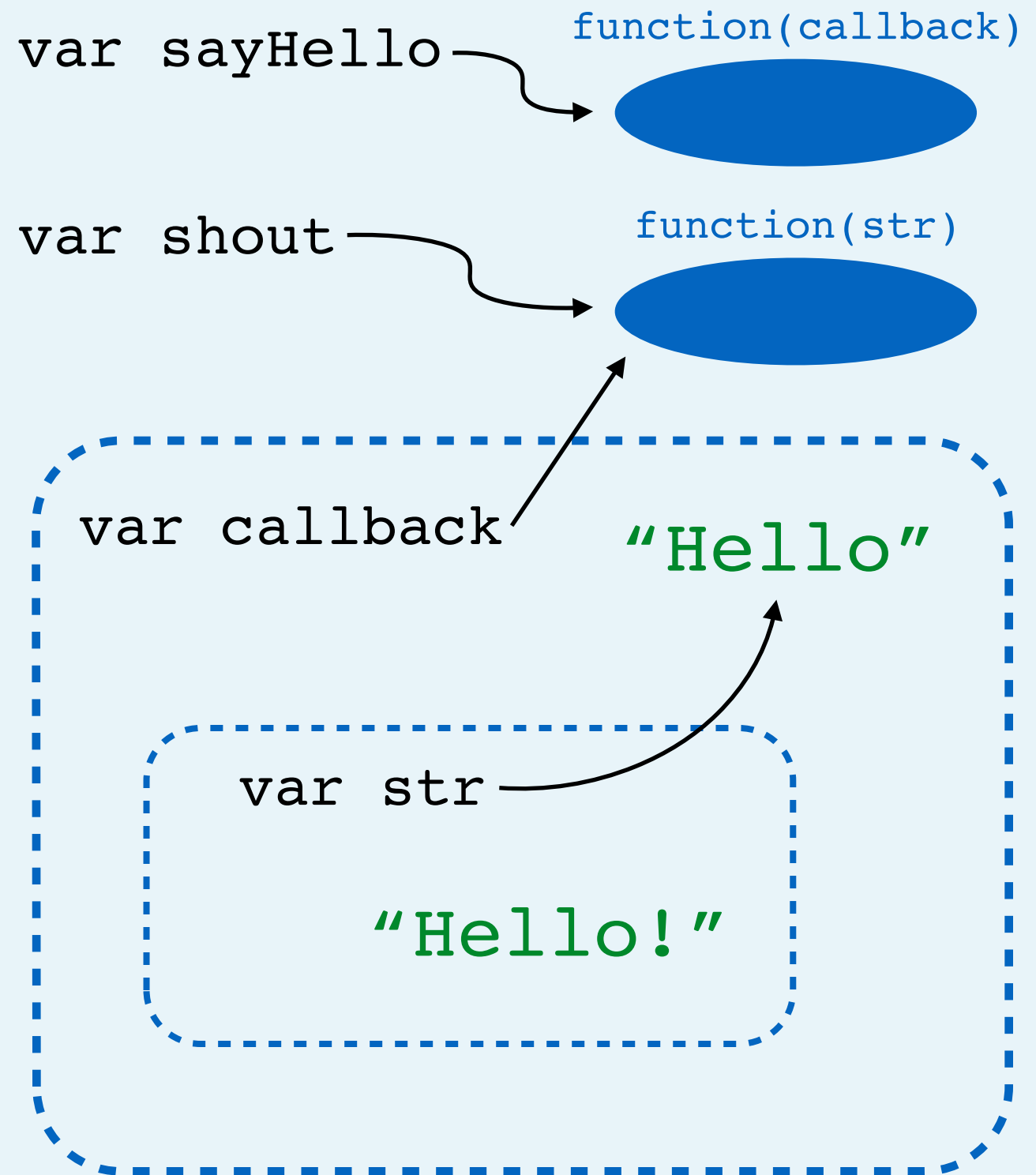
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

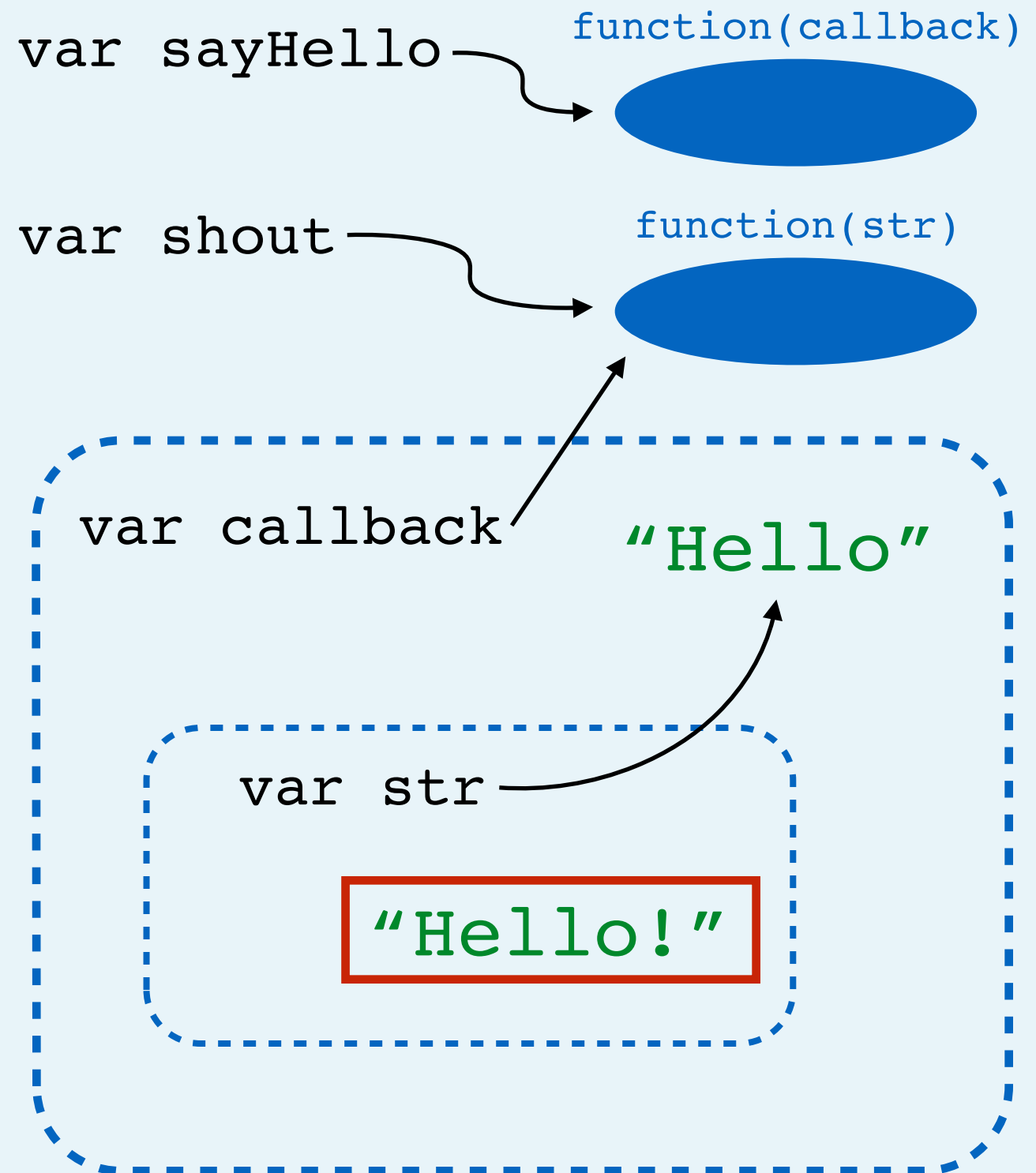
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

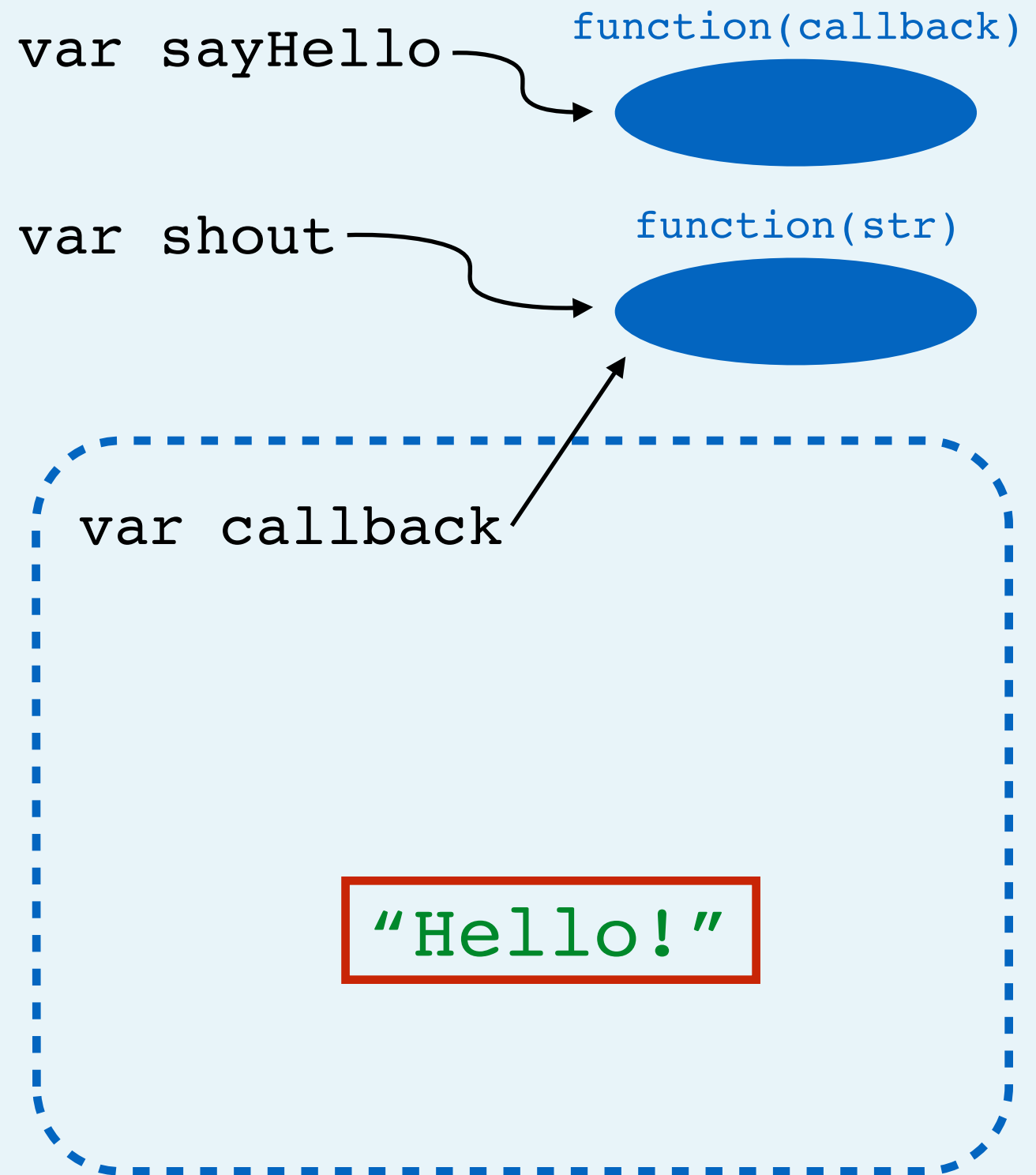
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value
 - b. Mark as return value



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

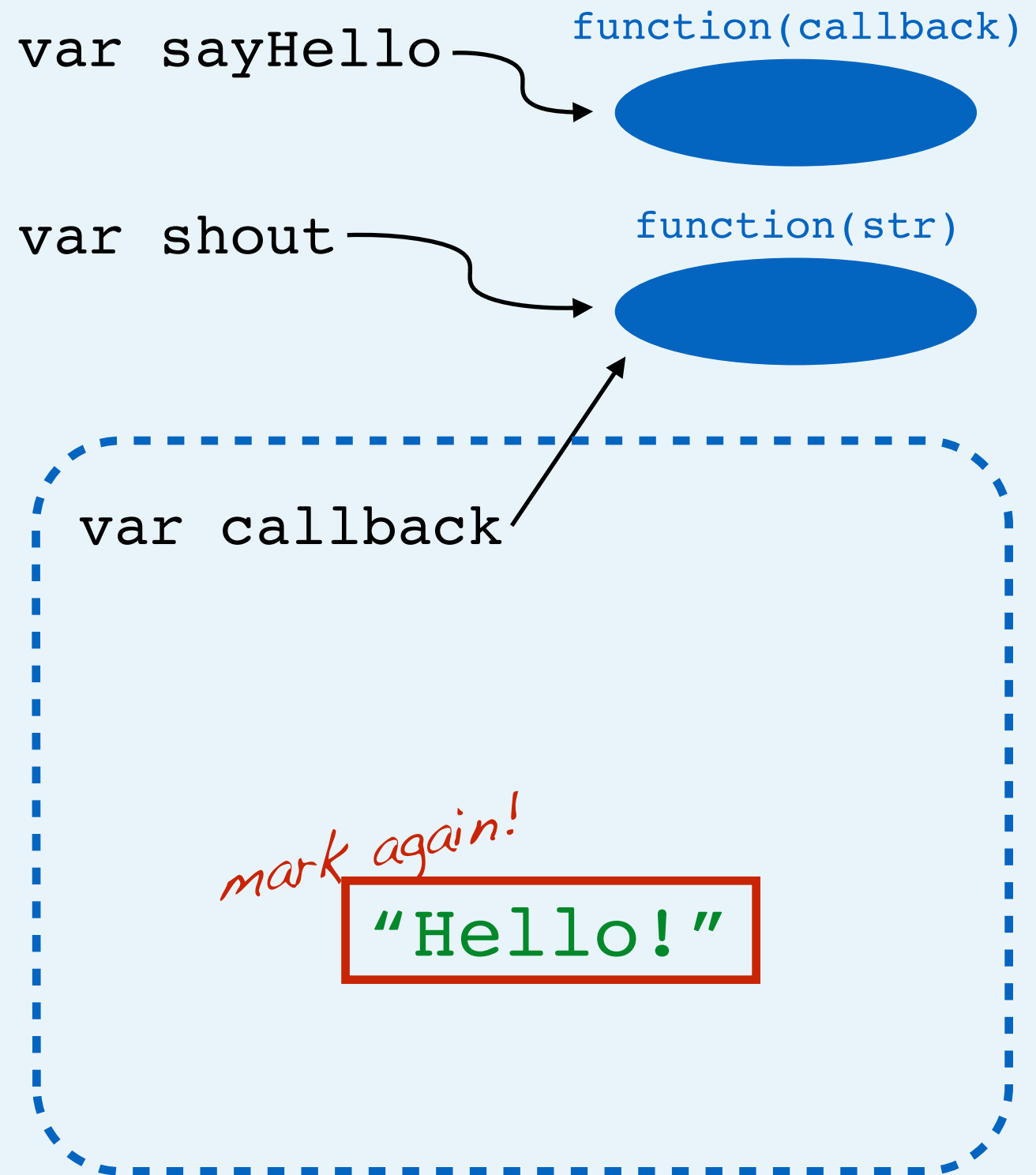
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value
 - b. Mark as return value
 - c. Garbage collect scope



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

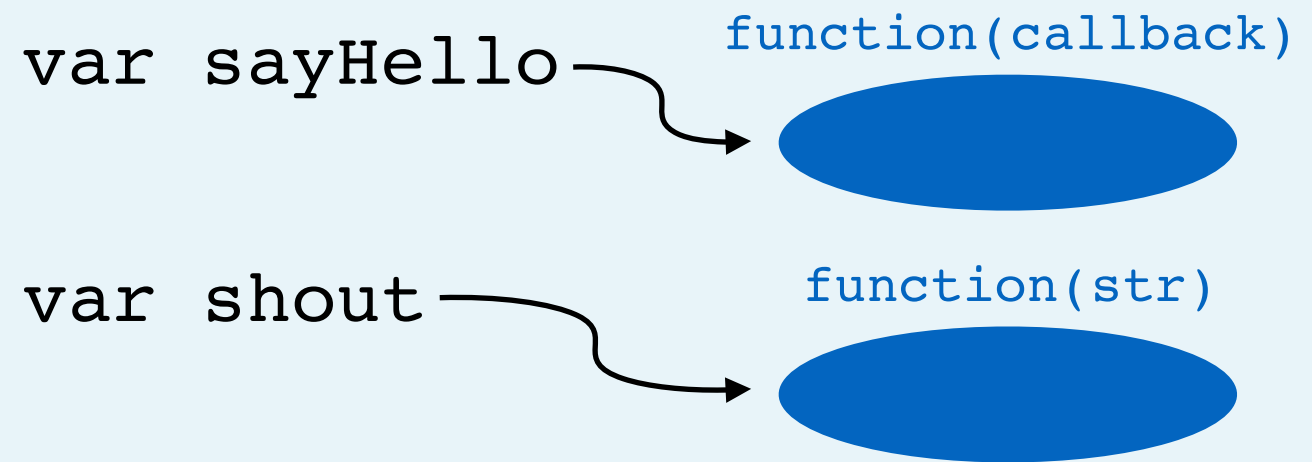
- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value
 - b. Mark as return value
 - c. Garbage collect scope
 - d. Mark as return value



Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value
 - b. Mark as return value
 - c. Garbage collect scope
 - d. Mark as return value
 - d. Garbage collect scope

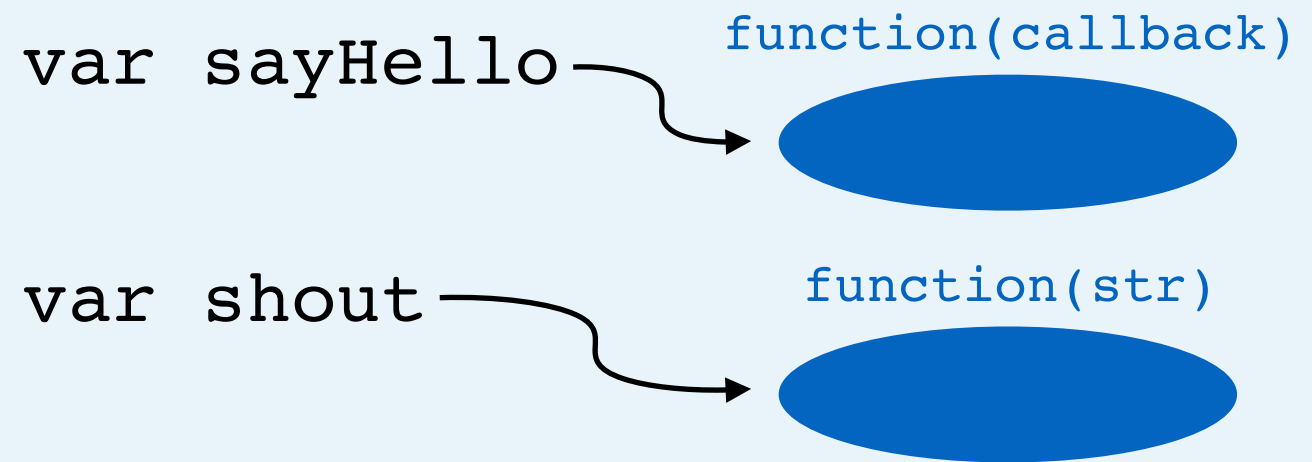


"Hello!"

Callbacks

```
var sayHello = function (callback) {  
  return callback("Hello")  
}  
var shout = function (str) {  
  return str + "!"  
}  
var result = sayHello(shout)
```

- a. Assignment
 - a. Evaluate right side
 - b. Look up value of sayHello
 - c. Look up value of shout (resolve argument)
 - d. Call function
 - a. Create scope
 - b. Create parameters
 - c. Return statement
 - a. Look up value of modifier
 - b. Create value (resolve argument)
 - c. Call function
 - a. Create scope & parameters
 - b. Return statement
 - a. Binary Operation
 - a. Look up value of str
 - b. Create value
 - b. Mark as return value
 - c. Garbage collect scope
 - d. Mark as return value
 - d. Garbage collect scope
 - e. Create varresult, point to value



`var result` → `"Hello!"`