

# JavaScript Avançado I

Capítulo 2. JavaScript: Funções

Prof. Bruno no Augusto Teixeira

# JavaScript Avançado I

---

## 2.1 Escopos

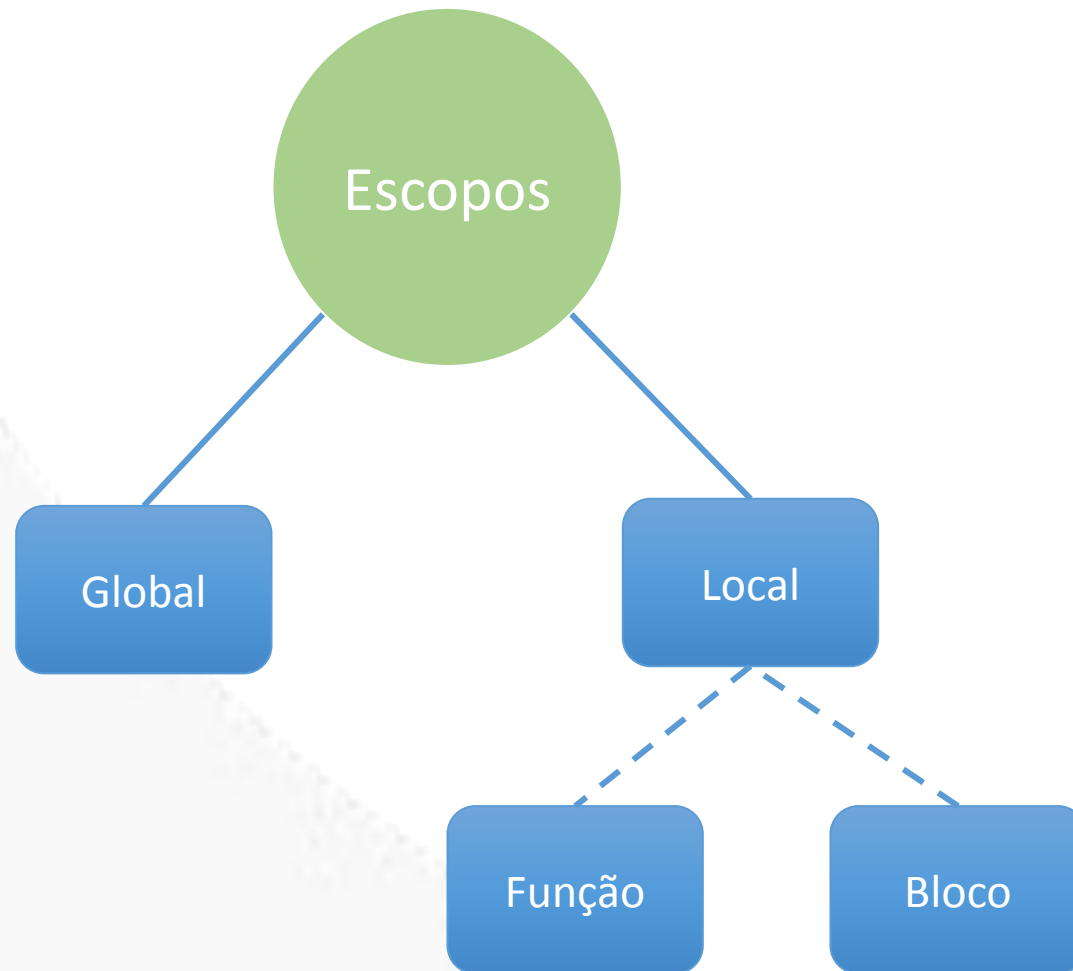
Prof. Bruno no Augusto Teixeira

# Nesta aula



- ❑ Escopos.

# Escopos



# Conclusão



- ☑ Escopos:
  - ☑ Global.
  - ☑ Local.
  - ☑ Bloco.
  - ☑ Função.
  - ☑ Hoisting.

# Próxima aula



- ❑ Closures.

# JavaScript Avançado I

---

## 2.2 Closures

Prof. Bruno no Augusto Teixeira



# Nesta aula



- ❑ Closures.



# Closures



- Encapsulamento.



```
var msg = 1;  
var car = 2;
```

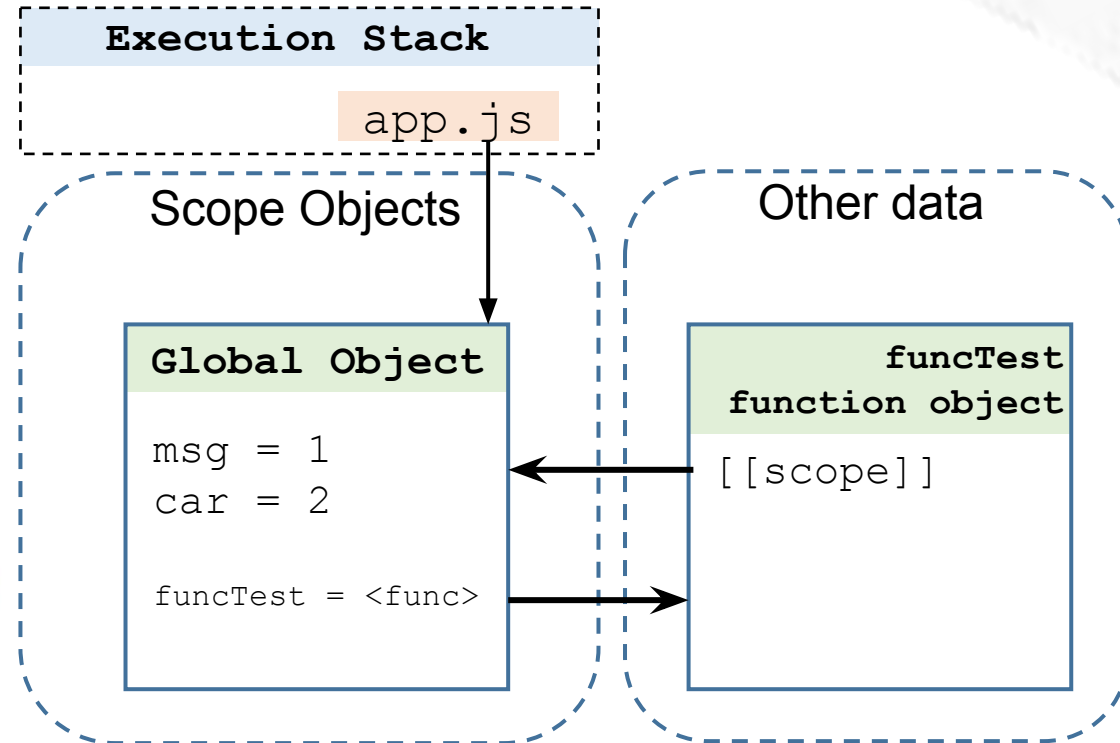


```
var msg = 1;
var car = 2;

function funcTest() {

  var a = 1;
  var b = 2;
  var msg = 3;

  console.log("Dentro");
}
```



```

var msg = 1;
var car = 2;

function funcTest() {

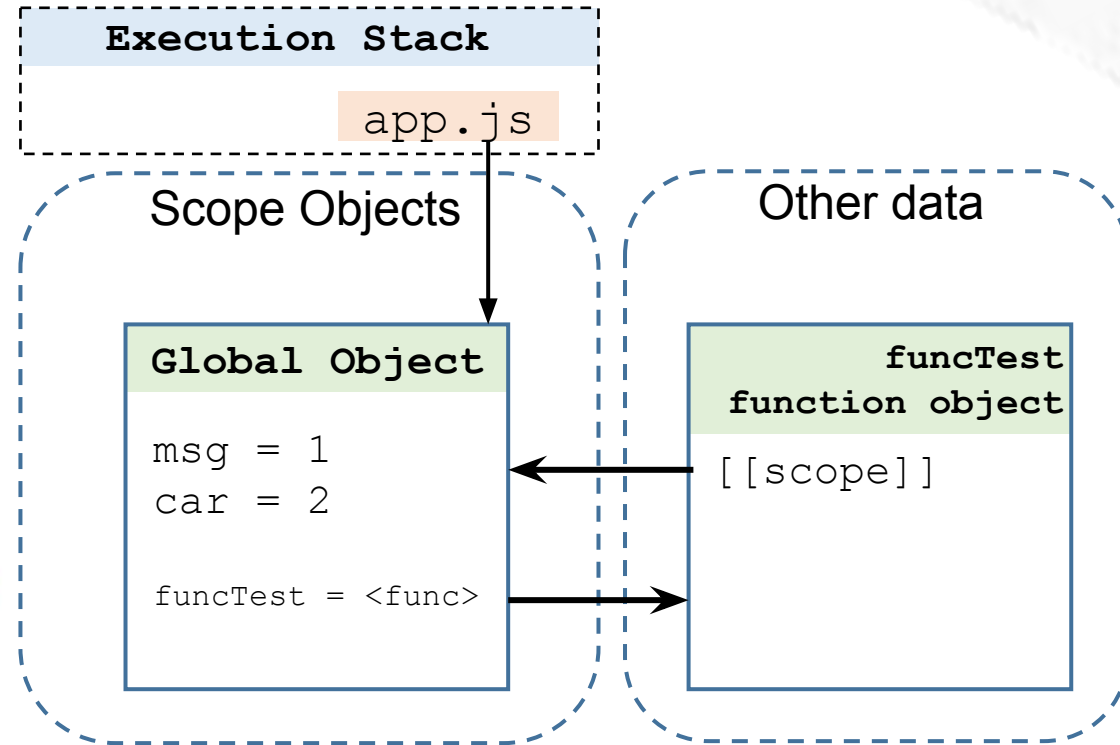
    var a = 1;
    var b = 2;
    var msg = 3;

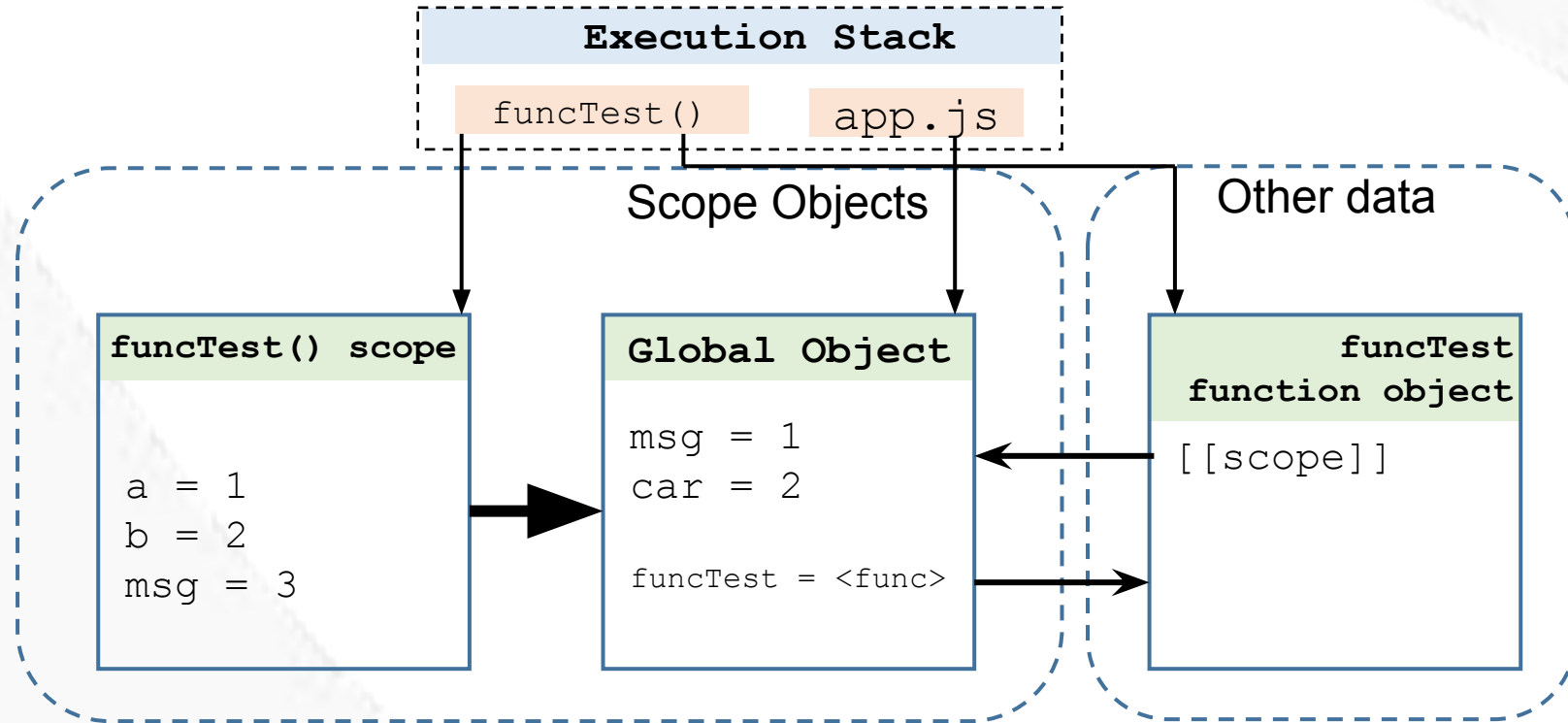
    console.log("Dentro");
}

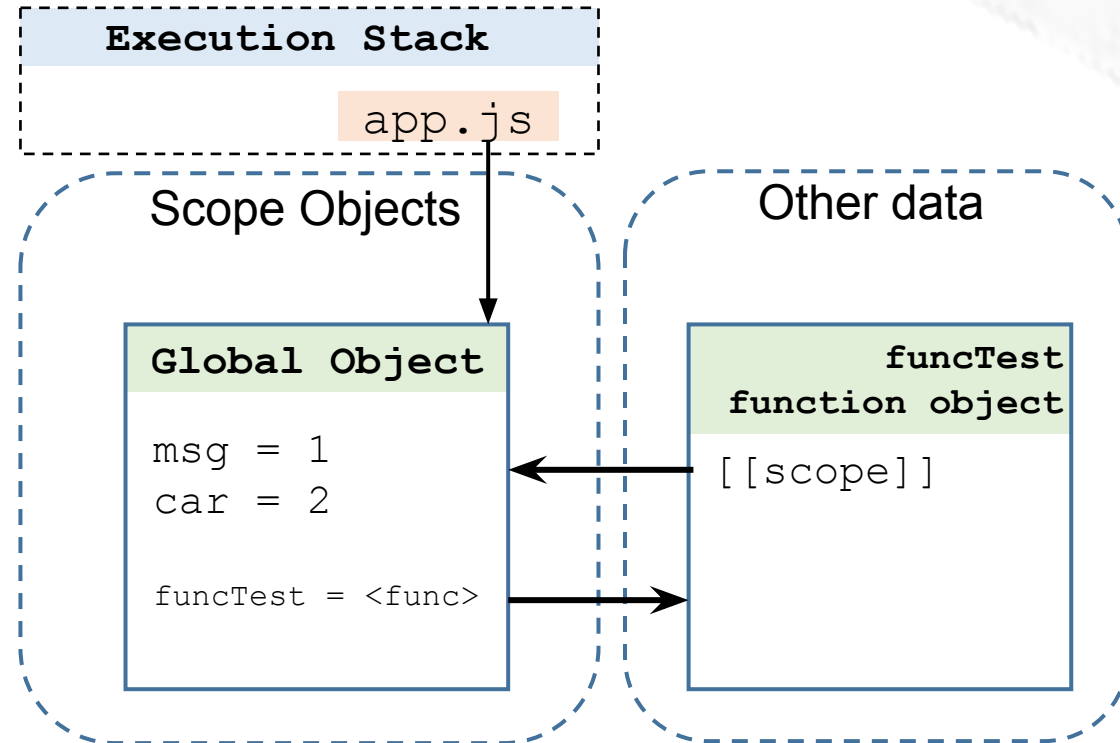
console.log("Fora");

funcTest();

```







```
function createCounter(initial) {
  var counter = initial;

  function increment(value) {
    if (!isFinite(value) || value < 1){
      value = 1;
    }
    counter += value;
  }

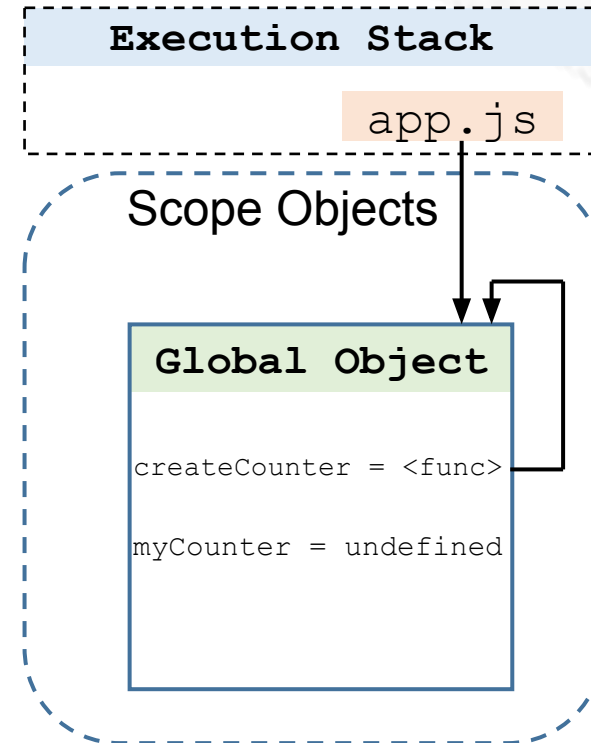
  function get() {
    return counter;
  }

  return {
    increment: increment,
    get: get
  };
}

var myCounter = createCounter(100);

console.log(myCounter.get()); // print "100"

myCounter.increment(5);
console.log(myCounter.get()); // print "105"
```



```
function createCounter(initial) {

  var counter = initial

  function increment(value) {
    if (!isFinite(value)) {
      value = 1;
    }
    counter += value;
  }

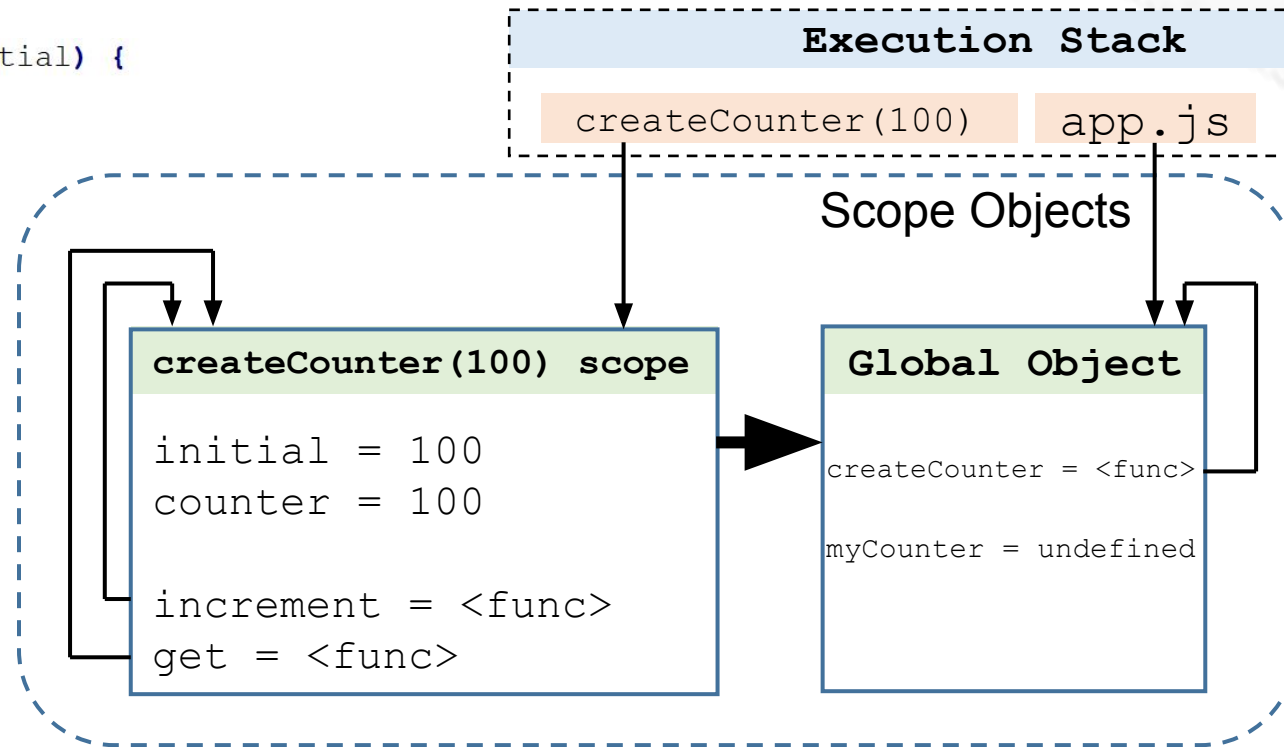
  function get() {
    return counter;
  }

  return {
    increment: increment,
    get: get
  };
}
```

```
var myCounter = createCounter(100);

console.log(myCounter.get()); // print "100"

myCounter.increment(5);
console.log(myCounter.get()); // print "105"
```





```
function createCounter(initial) {

  var counter = initial

  function increment(va
    if (!isFinite(value
      value = 1;
    }
    counter += value;
  }

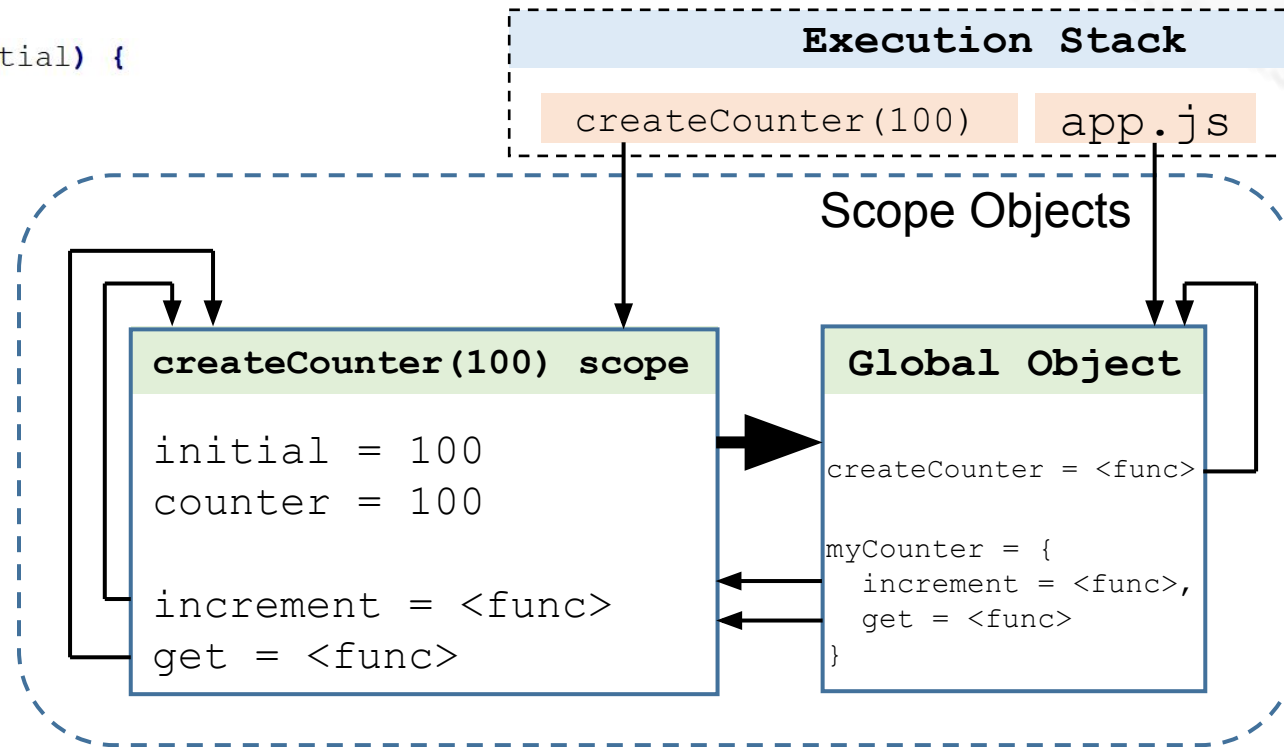
  function get() {
    return counter;
  }

  return {
    increment: incremen
    get: get
  };
}
```

```
var myCounter = createCounter(100);

console.log(myCounter.get()); // print "100"

myCounter.increment(5);
console.log(myCounter.get()); // print "105"
```



# Closures

```
function createCounter(initial) {
```

```
  var counter = initial
```

```
  function increment(value) {  
    if (!isFinite(value))  
      value = 1;  
    counter += value;  
  }  
  counter += value;  
}
```

```
function get() {  
  return counter;  
}
```

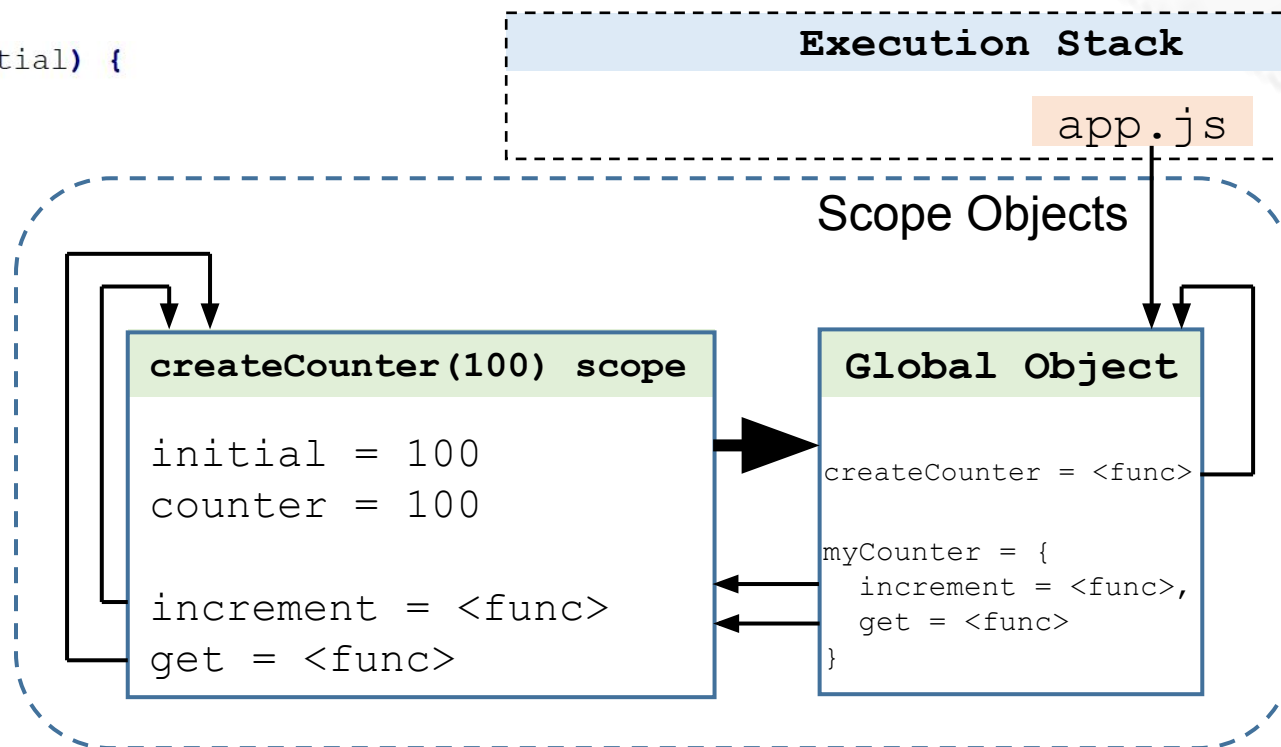
```
return {  
  increment: increment,  
  get: get  
};  
}
```

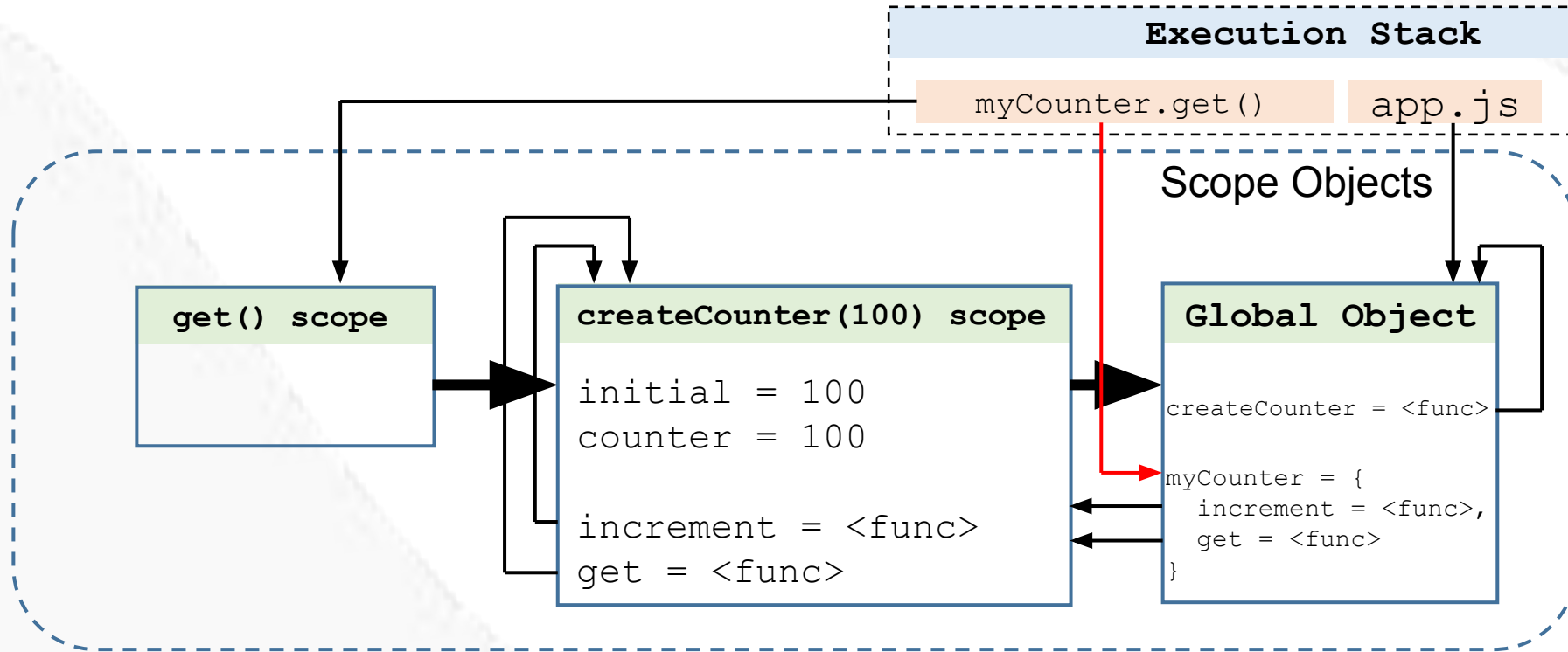
```
var myCounter = createCounter(100);
```

```
console.log(myCounter.get()); // print "100"
```

```
myCounter.increment(5);
```

```
console.log(myCounter.get()); // print "105"
```





```
function createCounter(initial) {

  var counter = initial

  function increment(va
    if (!isFinite(value
      value = 1;
    }
    counter += value;
  }

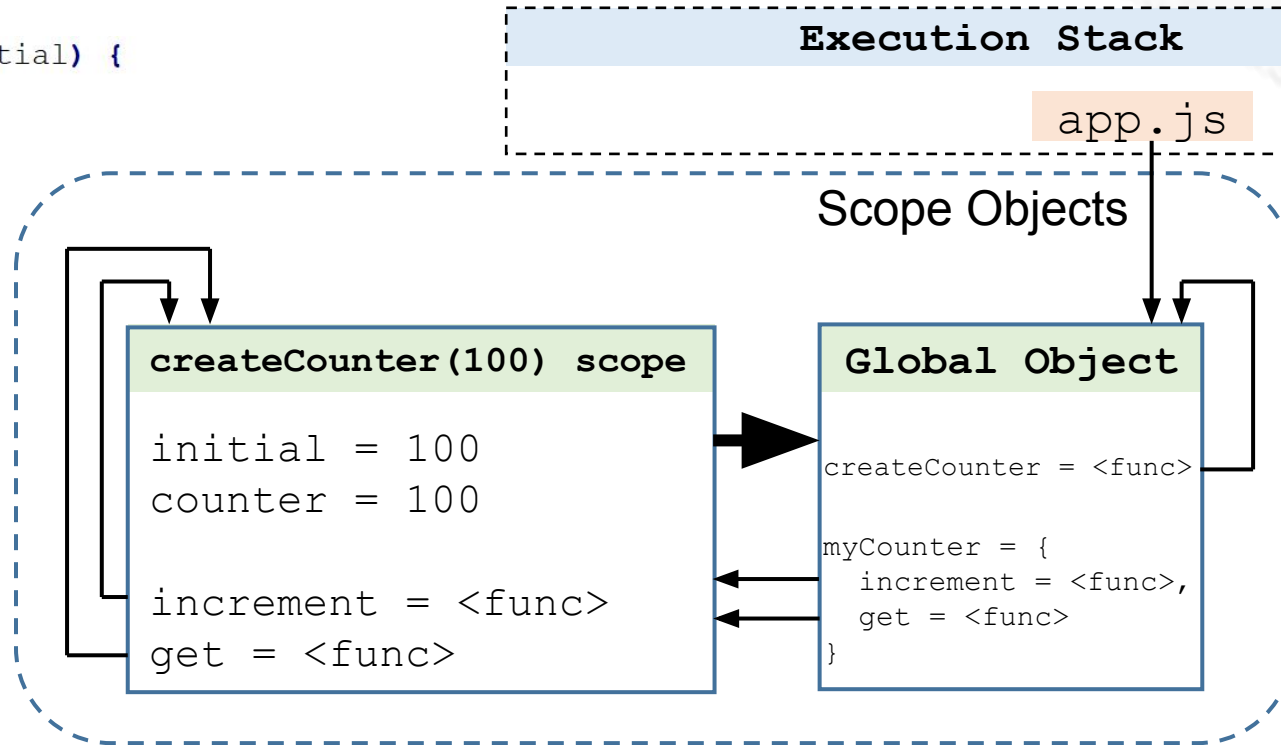
  function get() {
    return counter;
  }

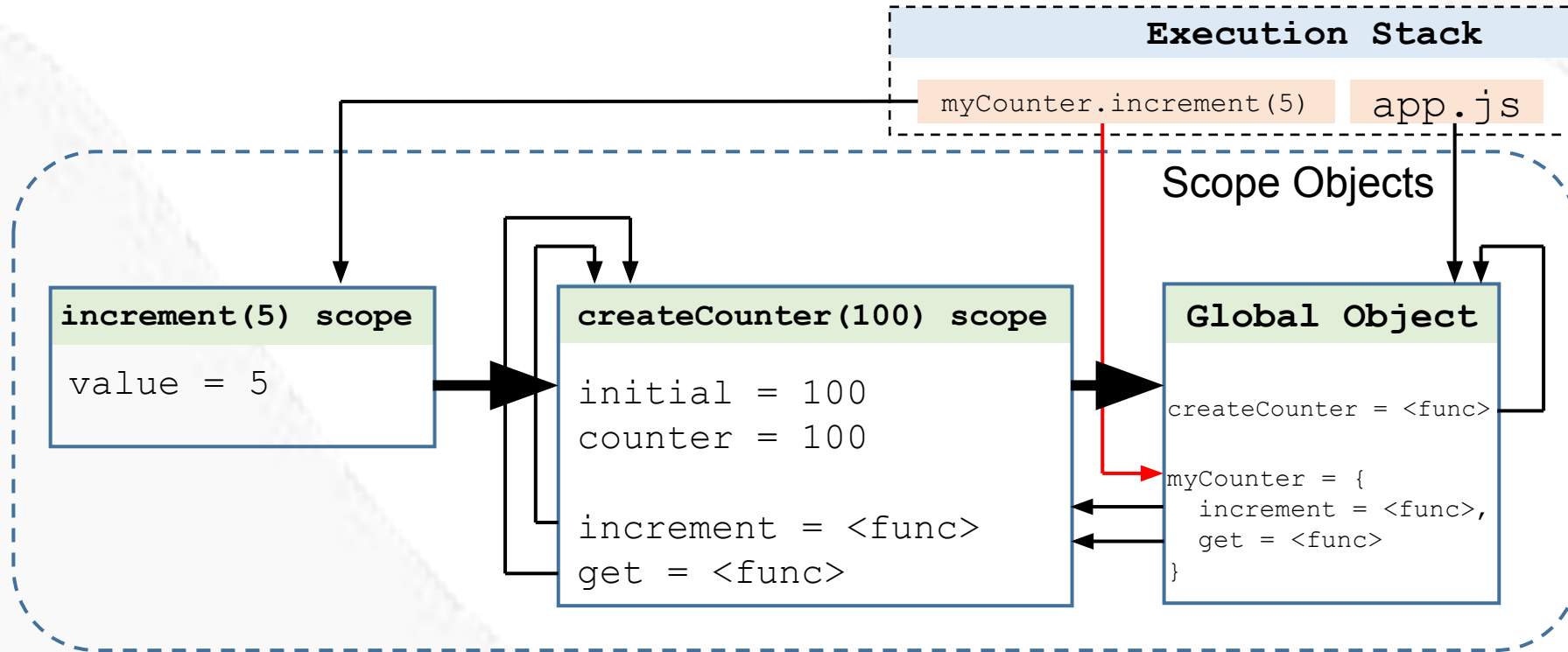
  return {
    increment: incremen
    get: get
  };
}
```

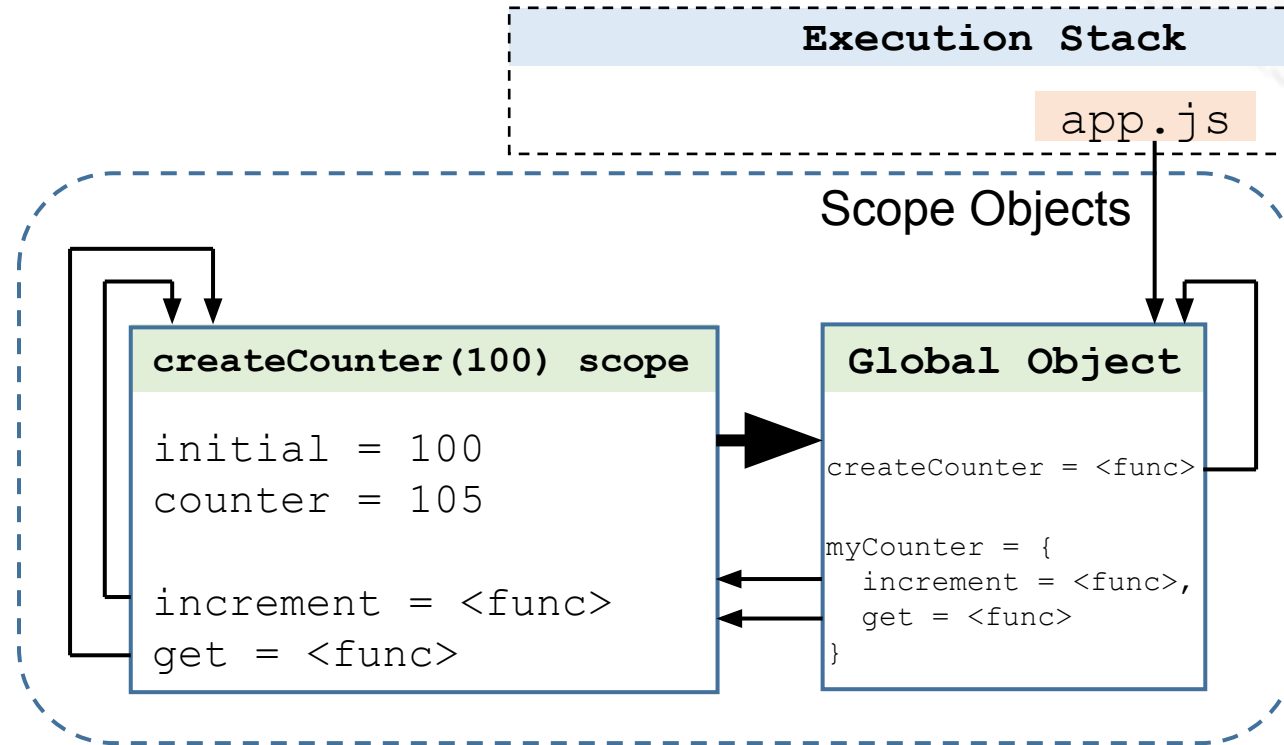
```
var myCounter = createCounter(100);

console.log(myCounter.get()); // print "100"

myCounter.increment(5);
console.log(myCounter.get()); // print "105"
```

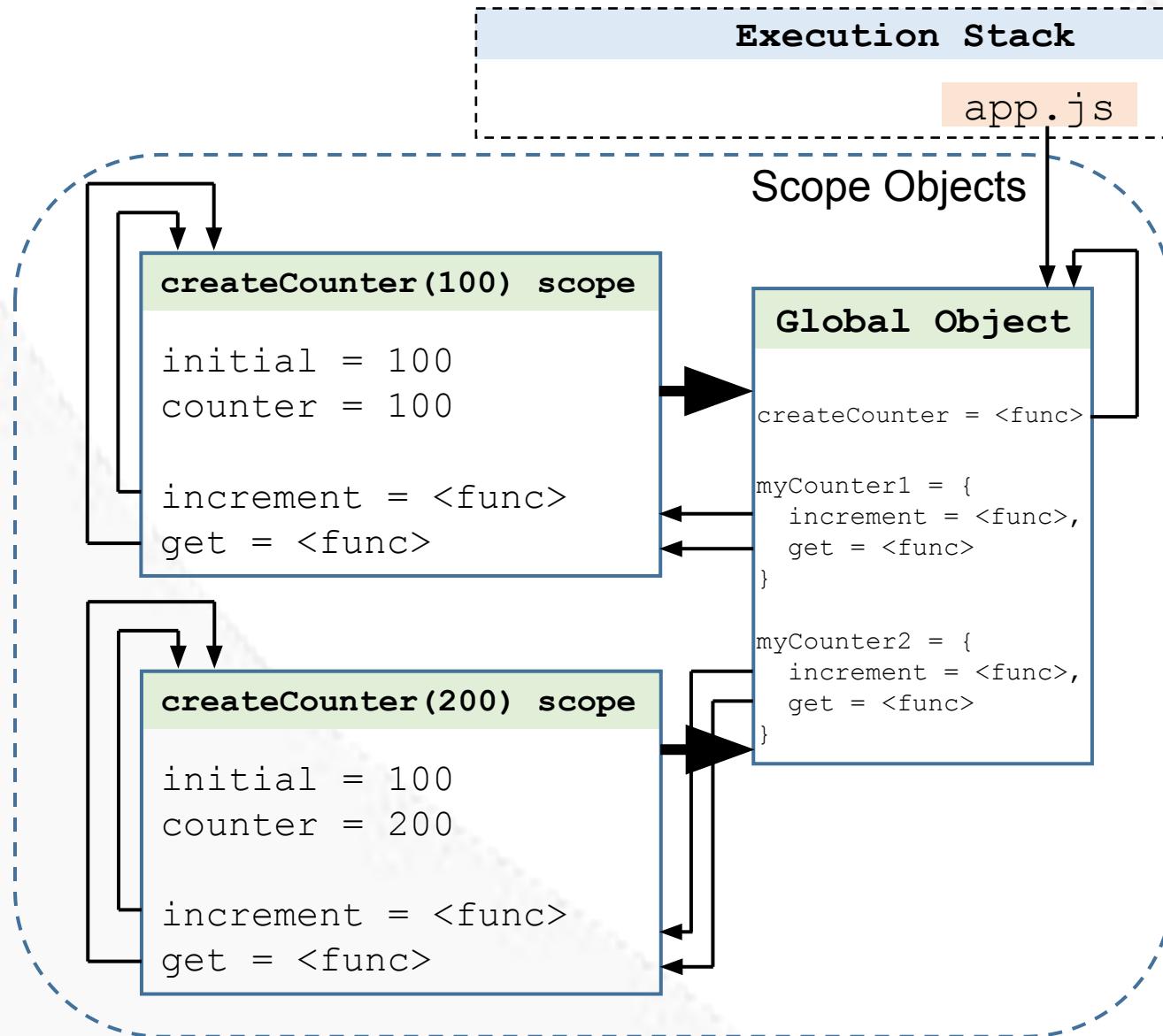






```
function createCounter(initial) {  
  /* ... implementação */  
}
```

```
var myCounter1 = createCounter(100);  
var myCounter2 = createCounter(200);
```





# Conclusão



☒ Closures.

# Próxima aula



- ❑ Prototypes.

# JavaScript Avançado I

---

## 2.3 Prototypes

Prof. Bruno no Augusto Teixeira



# Nesta aula



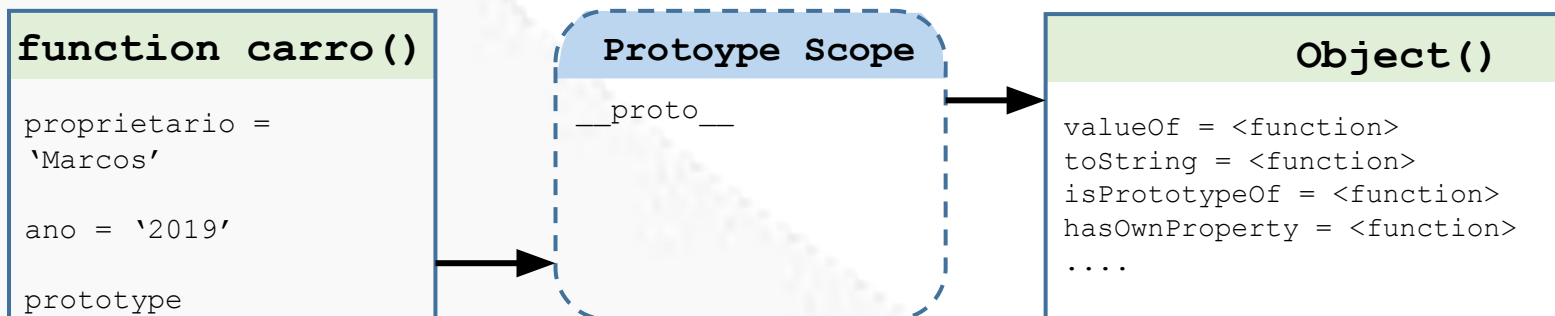
- ☐ Prototypes.

# Prototypes

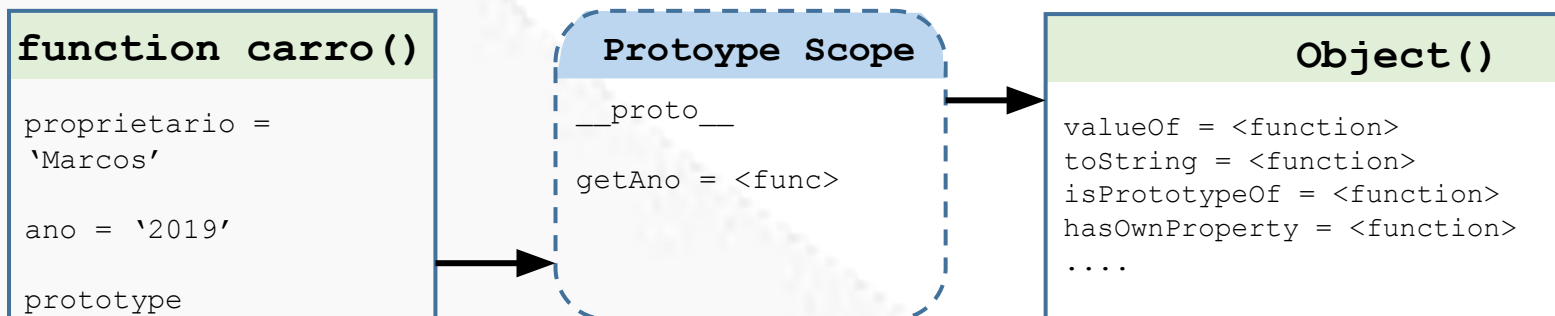


- Herança.

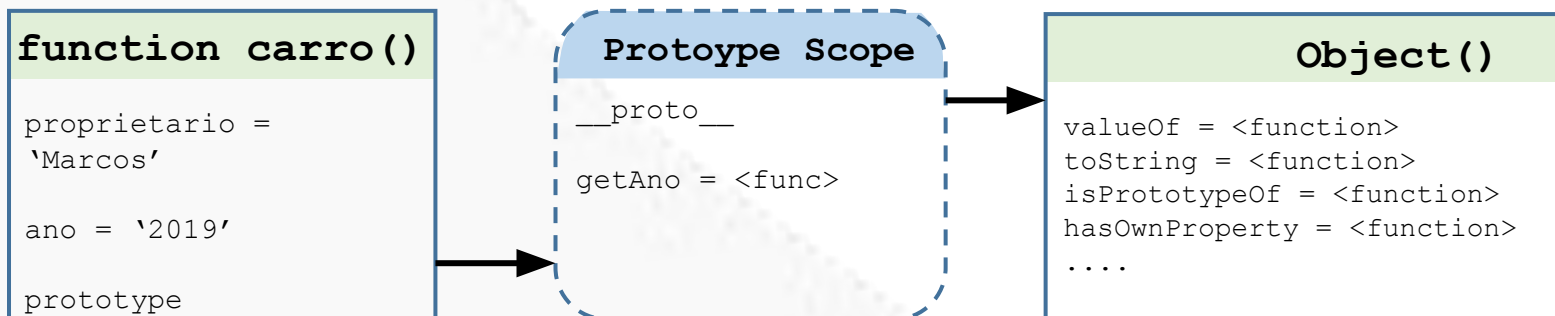
```
function Carro() {
  this.proprietario = 'Marcos';
  this.ano = 2019;
}
```



```
function Carro() {
  this.proprietario = 'Marcos';
  this.ano = 2019;
}
Carro.prototype.getAno = function () {
  console.log("Ano: " + this.ano);
  return this.ano;
};
```

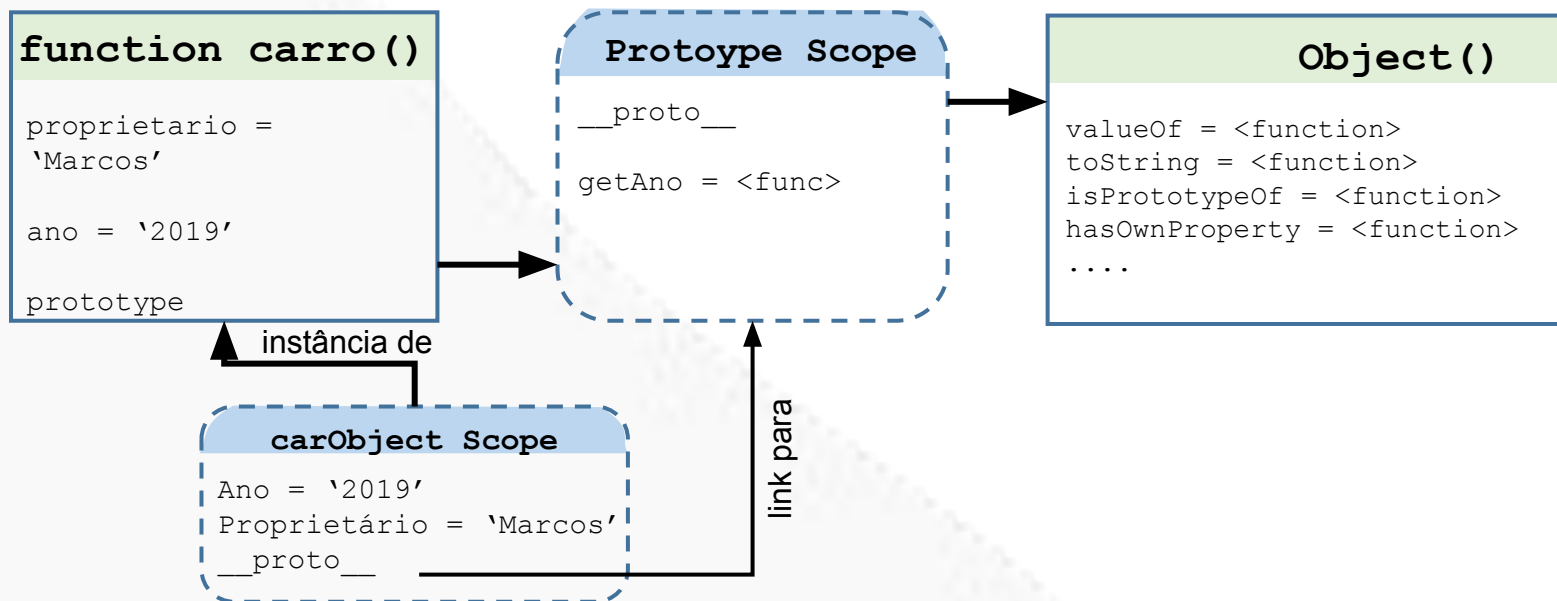


```
function Carro() {
  this.proprietario = 'Marcos';
  this.ano = 2019;
}
Carro.prototype.getAno = function () {
  console.log("Ano: " + this.Ano);
  return this.Ano;
};
let carObject = new Carro();
```

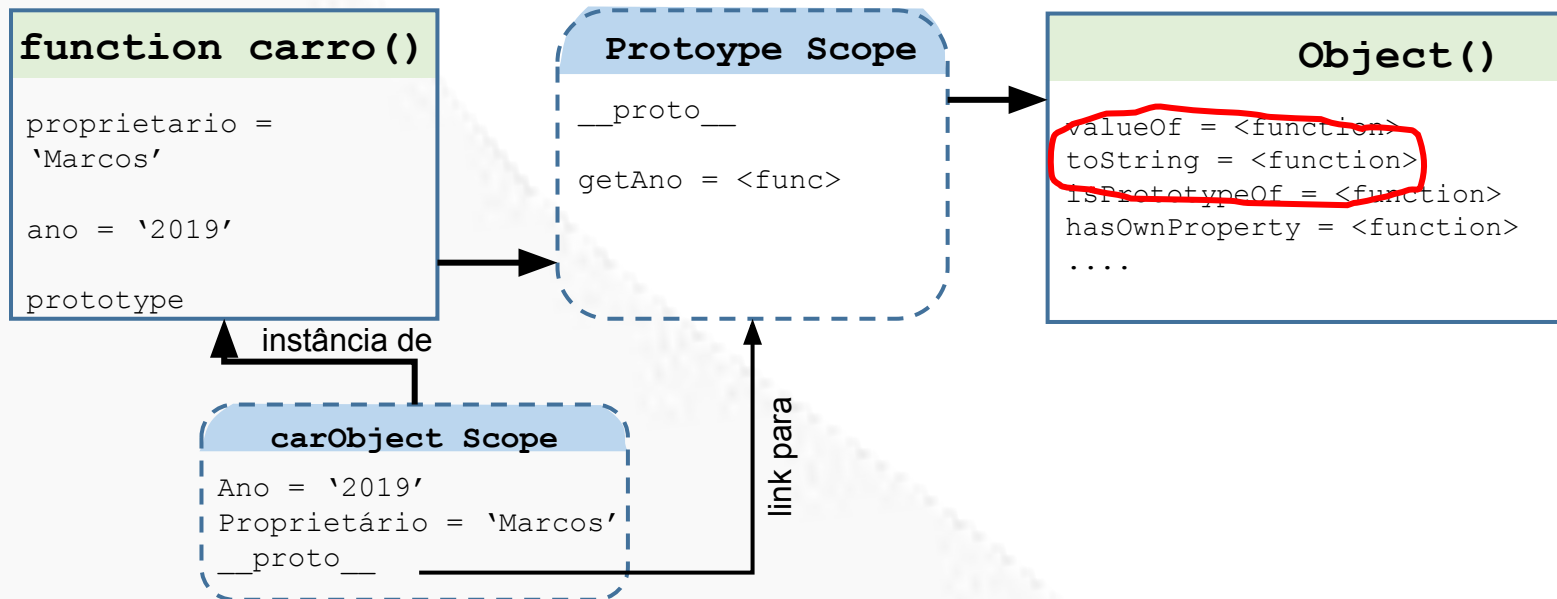




```
function Carro() {
  this.proprietario = 'Marcos';
  this.ano = 2019;
}
Carro.prototype.getAno = function () {
  console.log("Ano: " + this.Ano);
  return this.Ano;
};
let carObject = new Carro();
```



```
function Carro() {
  this.proprietario = 'Marcos';
  this.ano = 2019;
}
Carro.prototype.getAno = function () {
  console.log("Ano: " + this.Ano);
  return this.Ano;
};
let carObject = new Carro();
carObject.toString();
```



# Conclusão



☒ Prototypes.

# Próxima aula



- ❑ IIFE – Funções Imediatas.

# JavaScript Avançado I

---

## 2.4 IIFE – Funções Imediatas

Prof. Bruno no Augusto Teixeira

# Nesta aula



- ❑ IIFE – Funções Imediatas.

# IIFE

Immediately Invoked Function Expression.



# IIFE



Function Declaration:

```
function myFunction () {  
    /* código */  
}
```

Function Expression

```
let myFunction = function() {  
    /* código */  
};
```



# IIFE



Immediately Invoked Function Expression.

```
( function () {} ) ();
```

# IIFE



- Poluição do escopo global.
- Privacidade de dados.
- Closures.
- Renomear variáveis.
- Capturar o objeto Global.



# Conclusão



☑ IIFE.

# Próxima aula



☐ Proxy.

# JavaScript Avançado I

---

2.5 Proxy

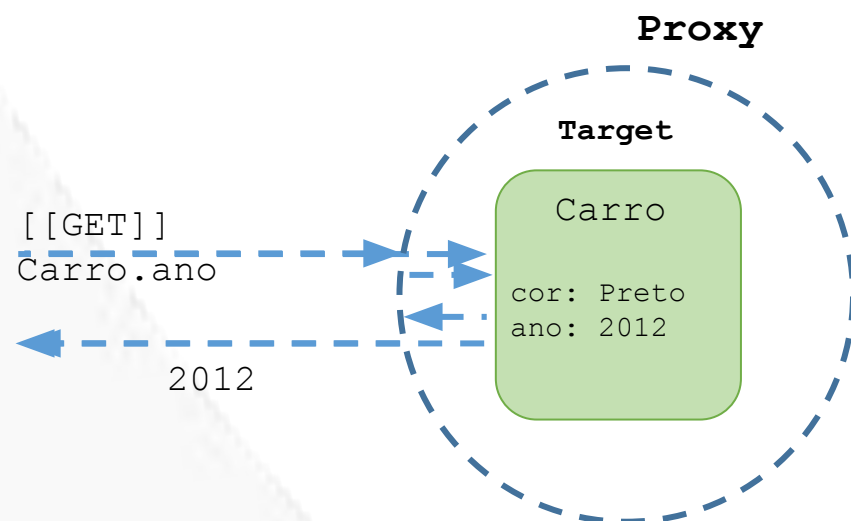
Prof. Bruno no Augusto Teixeira

# Nesta aula



- ☐ Proxy.
- ☐ Reflect.

# Proxy



# Conclusão



- ☒ Proxy.
- ☒ Reflect.



# Próxima aula

☐ Curry.



**IGTi**

# JavaScript Avançado I

---

2.6 Curry

Prof. Bruno no Augusto Teixeira

# Nesta aula



- ☐ Curry.

# Currying



# Conclusão



☒ Currying.

# Próxima aula

- ❑ JavaScript Assíncrono.

