



# PROTOTYPAL INHERITANCE

## CONSTRUCTOR FUNCTION

---

```
function Person(initialName) {  
  this.name = initialName;  
  this.species = 'Homo Sapiens';  
  this.speak = function() {  
    return "Hello! I'm " + this.name;  
  };  
}
```

```
let adam = new Person('Adam');  
let steve = new Person('Steve');
```

## CREATING A PROTOTYPE CHAIN

---

```
Child.prototype = new Parent();
```

*Specifies parent's prototype as  
prototype of child constructor*

---

# CLASSES

---

```
class Vehicle {  
  constructor(color) {  
    this.color = color;  
  }  
  checkSpeed() {  
    console.log('Not speeding!');  
  }  
}
```

```
class Car extends Vehicle() {  
  constructor(color) {  
    super(color);  
  }  
}
```

## CLASS SYNTAX

---

`class name extends prototype`      *Creates a class called name based on prototype*

`super(arguments)`      *Calls the constructor of the parent class, passing it arguments and returning the result*

---