



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

| | |
|--|---|
| <code>class</code> <i>name</i> <code>extends</code> <i>prototype</i> | <i>Creates a class called name based on prototype</i> |
|--|---|

| | |
|---|---|
| <code>super</code> (<i>arguments</i>) | <i>Calls the constructor of the parent class, passing it arguments and returning the result</i> |
|---|---|
