

19 Constructor

What is a constructor?

In JavaScript, a constructor is a special function that acts as a mold to create new objects.

There are numerous built-in constructors in JavaScript, such as [String](#), [Number](#), [Promise](#), [Date](#), [Array](#), [Object](#), and many more.

We can create our own custom constructors if need be.

A great place to use a constructor is when you are creating multiple objects of the same kind.

There are two parts to working with a constructor:

(1) Defining a constructor

When creating a custom constructor

(2) Using a constructor

with the "new" keyword

// Defining a Constructor

```
function Car(make, model, year){  
  this.make = make;  
  this.model = model;  
  this.year = year;  
  
  this.setMiles = function(miles){  
    this.miles = miles  
    return miles;  
  }  
}
```

// Using a constructor

```
const car1 = new Car('Toyota', 'Prius', 2016);  
const car2 = new Car('Hyundai', 'Sonata', 2018);
```

// Adding method to the constructor prototype

```
Car.prototype.age = function(){  
  return (new Date()).getFullYear() - this.year;  
}
```

```
car1.age(); // 2
```

Rule of thumb:

A) Set properties inside a constructor.

B) Set methods inside the prototype property.

"new" keyword

The new keyword is used to create a new object (instance) from the constructor.

"prototype" property

prototype is a special property on every object. Properties (methods or values) attached to the prototype property get inherited to every instance of the constructor.