

Methods

- Methods are functions associated to an object.

```
const circle = {  
    radius: 10,  
    area() {return Math.PI * this.radius * this.radius},  
    circumference() {return 2 * Math.PI * this.radius},  
    expand(scale) {this.radius *= scale}  
};
```

- They are called through an object using dot notation:
`circle.area();`
- The object used to call the function is called **receiver**.

Methods

- **this** takes on the value of the method's receiver as determined at runtime.

```
function talkTo(message, suffix)  
{ return message + ', ' + this.name + ". " + suffix; }
```

```
const alice = {name: 'Alice', address: talkTo};
```

```
const bob = {name: 'Bob', address: talkTo};
```

```
console.log(alice.address("Hi", "Bye."));
```

```
console.log(bob.address("Bonjour", "Au revoir."));
```

Hi, Alice. Bye.

Bonjour, Bob. Au revoir.

Prototypes

In JavaScript, we start with an initial (prototypical) object, then derive additional objects from it. These new objects have the original object as their **prototype**.

- The expression **Object.create(p)** creates a new object whose prototype is p.
- Our script creates a black circle of radius 1, centered at the origin, as the **prototype** of three other circles (c1, c2, and c3).
- In c1, x and color are called **own properties**, while y and radius are called **inherited properties**.

```
const unitCircle = {
  x: 0,
  y: 0,
  radius: 1,
  color: 'black',
  area() {return Math.PI * this.radius * this.radius},
  circumference() {return 2 * Math.PI * this.radius}
};

const c1 = Object.create(unitCircle);
c1.x = 3;
c1.color = 'green';

const c2 = Object.create(unitCircle);
c2.radius = 5;

const c3 = Object.create(unitCircle);
console.log(c2.color);          //black
console.log(c2.area());         //25*PI
console.log(c3.y);              //0
console.log(c3.area());         //PI
```

Prototypes defined as “Classes”

Javascript provides syntax for creating prototypes using the **class** keyword and a **constructor** function. The constructor function can be used to initialize the properties of the object to either default values or values given as arguments.

```
class Circle {  
    constructor( x=0, y=0, r=1, c= "black") {  
        this.x=x;  
        this.y=y;  
        this.radius=r;  
        this.color=c;  
    }  
    area() {return Math.PI * this.radius * this.radius;}  
};  
  
const c1= new Circle();  
c1.radius = 10;  
c1.color="green";  
  
const c3= new Circle (10, 10);  
  
console.log(c1);  
console.log("Area: " + c1.area()); }  
console.log();  
console.log(c3);  
console.log("Area: " + c3.area()); }
```

```
Circle { x: 0, y: 0, radius: 10, color: 'green' }  
Area: 314.1592653589793
```

```
Circle { x: 10, y: 10, radius: 1, color: 'black' }  
Area: 3.141592653589793
```