Object Methods, "this"

Object methods

Look at this object

```
> let user = {
... name: "John",
... age: 30
... };
>
```

You're familiar with methods from Java:

```
user.sayHi = function() {
  console.log("Hello!");
};

user.sayHi();
```

You can use a pre-declared function:

```
> let user = {
    name: "John",
    age: 30
    };
>

user.sayHi = function() {
  console.log("Hello!");
};

user.sayHi = sayHi; //add it as a method

user.sayHi(); //call it
```

Method shorthand

```
> user = {
... sayHi: function() {
..... console.log("Hello");
..... }
... };

// Below does the same thing
```

```
// but without 'function()'


> user = {
... sayHi: function() {
..... console.log("Hello");
..... }
... };
```

Stay tuned. We'll find out later they're not identical.

What is "this"?

this: a keyword that refers to the object making the method call. Again, like Java.

```
> let user = {
    name: "John",
    age: 30


    sayName() {
        console.log(this.name);
        }
};
```

Special in JavaScript: 'this' can be used in any function, even if it's not a method of an object

```
> function sayHi() {
... console.log(this.name);
... }
```

No syntax error

Arrow Function Have No "this"

If an arrow function references "this" it is taken to be outer context.

```
> let user = {
... firstName: "Paul",
... sayName() {
..... let name = () => console.log(this.firstName);
..... name();
..... }
... };
```

```
> user.sayName();
Paul
```

It returned user.firstName which is the outer context.