EXAMPLE 21: UNDERSTAND THE 2 WAYS OF ACCESSING OBJECT PROPERTIES.

ADDING A NEW PROPERTY IS EASY

```
rectangle["OutlineColor"] =
"Black";
```

```
rectangle.OutlineColor =
"Black";
```

REMOVING AN EXISTING PROPERTY IS EASY TOO

```
delete
rectangle.OutlineColor
```

```
delete
rectangle["OutlineColor"];
```

RECAP

ADDING A NEW PROPERTY IS EASY

SYNTAX #1

```
tangle["OutlineColor"] =
"Black";
```

```
rectangle.OutlineColor =
"Black"
```

REMOVING AN EXISTING PROPERTY IS EASY TOO

```
delete
rectangle.OutlineColor
```

```
delete
tangle["OutlineColo
```

SYNTAX #1

RECAP

ADDING A NEW PROPERTY IS EASY

```
rectangle[
```

SYNTAX #2

```
rectangle.OutlineColor =
"Black";
```

REMOVING AN EXISTING PROPERTY IS EASY TOO

```
   delete
rectangle.OutlineColor
```

SYNTAX #2

```
   delete
 rectangle[
```

RECAP

EXAMPLE 21: UNDERSTAND THE 2 WAYS OF ACCESSING OBJECT PROPERTIES.

EXAMPLE 21: UNDERSTAND THE 2 WAYS OF ACCESSING OBJECT PROPERTIES.

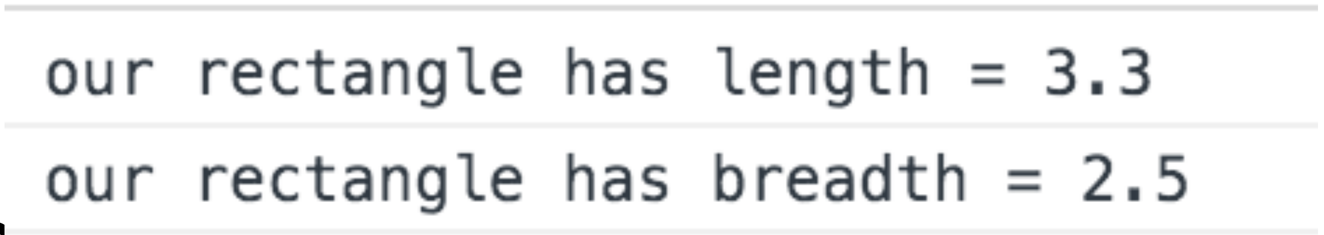
CLEARLY, THERE ARE 2 DIFFERENT WAYS OF ACCESSING THE PROPERTIES OF AN OBJECT.

CLEARLY, THERE ARE 2 DIFFERENT WAYS OF ACCESSING THE PROPERTIES OF AN OBJECT.

THIS IS ONE INSTANCE WHERE THE SEMANTICS ARE DIFFERENT FOR FUNCTIONS VERSUS FOR OTHER TYPES OF PROPERTIES.

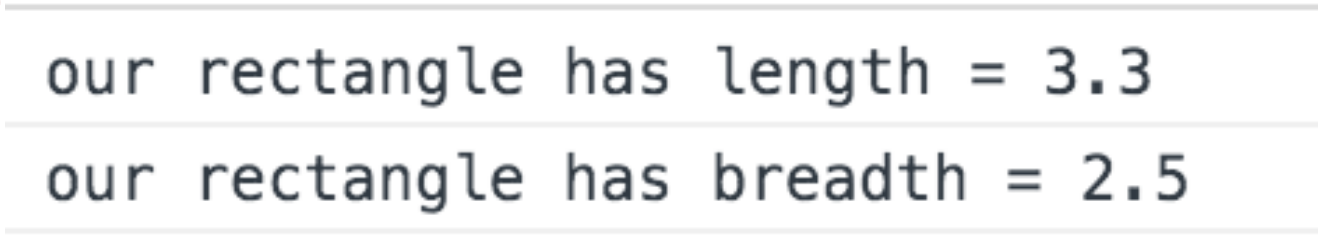
THIS IS ONE INSTANCE WHERE THE SEMANTICS ARE DIFFERENT FOR FUNCTIONS VERSUS FOR OTHER TYPES OF PROPERTIES.

```
  console.log("our rectangle has length = " +
rectangle.length);
  console.log("our rectangle has breadth = " +
rectangle.b
```



```
our rectangle has length = 3.3
our rectangle has breadth = 2.5
```

```
  console.log("our rectangle has length = " +
rectangle["length"]);
  console.log("our rectangle has breadth = " +
rectangle["b
```



```
our rectangle has length = 3.3
our rectangle has breadth = 2.5
```

THIS IS ONE INSTANCE WHERE THE SEMANTICS ARE DIFFERENT FOR FUNCTIONS VERSUS FOR OTHER TYPES OF PROPERTIES.

```
  console.log("our rectangle has length = " +
rectangle.length);
  console.log("our rectangle has breadth = " +
rectangle.
```

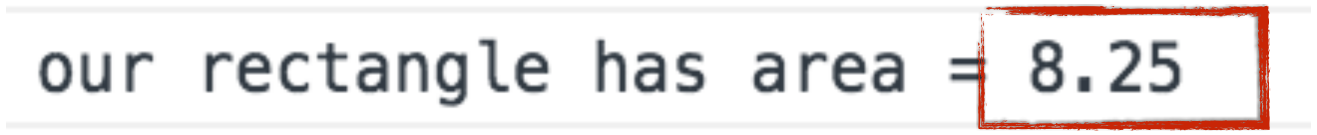
```
our rectangle has length = 3.3
our rectangle has breadth = 2.5
```

```
  console.l                                    " +
rectangle["length"]);
  console.log("our rectangle has breadth = " +
rectangle["breadth"]);
```

FOR NON-FUNCTION PROPERTIES, THE 2 ARE EQUIVALENT

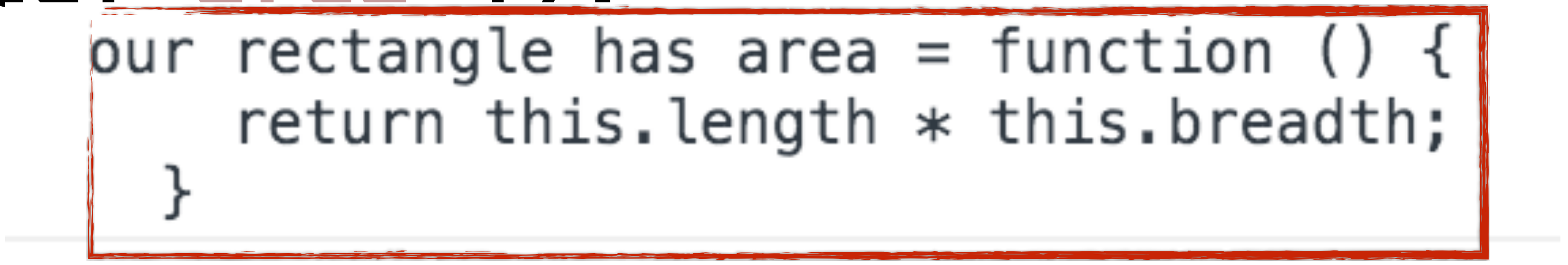
BUT THE SEMANTICS ARE QUITE DIFFERENT FOR PROPERTIES THAT ARE FUNCTIONS

```
console.log("our rectangle has area = " +
rectangle.area());
```



```
our rectangle has area = 8.25
```

```
console.log("our rectangle has area = " +
rectangle["area"]);
```



```
our rectangle has area = function () {
    return this.length * this.breadth;
  }
```