EXAMPLE 19: PROPERTIES CAN BE FUNCTIONS TOO!

FUNCTIONS

RECAP

FUNCTIONS

IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

RECAP

FUNCTIONS IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

YOU CAN STORE A FUNCTION IN A VARIABLE

RECAP

FUNCTIONS IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

YOU CAN STORE A FUNCTION IN A VARIABLE

YOU CAN HAVE A FUNCTION RETURN A FUNCTION

RECAP

FUNCTIONS IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

YOU CAN STORE A FUNCTION IN A VARIABLE

YOU CAN HAVE A FUNCTION RETURN A FUNCTION

YOU CAN HAVE A FUNCTION TAKE IN A FUNCTION AS AN ARGUMENT

RECAP

FUNCTIONS IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

1. YOU CAN STORE A FUNCTION IN A VARIABLE
2. YOU CAN HAVE A FUNCTION RETURN A FUNCTION
3. YOU CAN HAVE A FUNCTION TAKE IN A FUNCTION AS AN ARGUMENT

THESE 3 PROPERTIES COLLECTIVELY ARE CALLED "FIRST CLASS FUNCTIONS"

RECAP

FUNCTIONS IN JAVASCRIPT CAN BE TREATED JUST LIKE NUMBERS OR STRINGS

JAVASCRIPT IS A PROGRAMMING LANGUAGE THAT SUPPORTS FUNCTIONS

1

YOU CAN FUNCTION IN A VARIABLE

2

YOU CAN FUNCTION FUNCTION FUNCTION

3

YOU CAN TAKE IN A FUNCTION AS AN ARGUMENT

THESE 3 PROPERTIES COLLECTIVELY ARE CALLED "

RECAP

"AN OBJECT IS A SET OF KEY-VALUE PAIRS, WHERE THE VALUES CAN ALSO BE FUNCTIONS"

RECAP

"AN OBJECT IS A SET OF KEY-VALUE PAIRS, WHERE THE VALUES CAN ALSO BE FUNCTIONS"

REMEMBER THAT FUNCTIONS IN JAVASCRIPT ARE VALUES JUST LIKE STRINGS OR NUMBERS!

RECAP

"AN OBJECT IS A SET OF KEY-VALUE PAIRS, WHERE THE VALUES CAN ALSO BE FUNCTIONS"

REMEMBER THAT FUNCTIONS IN JAVASCRIPT ARE

FIRST CLASS FUNCTIONS

RECAP

REMEMBER THAT FUNCTIONS IN JAVASCRIPT ARE

FIRST CLASS FUNCTIONS

"AN OBJECT IS A SET OF KEY-VALUE PAIRS, WHERE THE VALUES CAN ALSO BE FUNCTIONS"

(ONCE YOU REMEMBER THIS, IT MAKES COMPLETE SENSE THAT VALUES IN AN OBJECT CAN BE FUNCTIONS, JUST AS THEY COULD BE STRINGS, OR NUMBERS OR WHATEVER)

RECAP

EXAMPLE 19: PROPERTIES CAN BE FUNCTIONS TOO!

EXAMPLE 19: PROPERTIES CAN BE FUNCTIONS TOO!

```
var rectangle = {
  length : 5.0,
  breadth : 3.5,
  color : 'Red',
  area : function() {
    return this.length * this.breadth;
  }
};
```

HERE IS AN OBJECT CREATED THE SIMPLE (JSON-LIKE) WAY

EXAMPLE 19: PROPERTIES CAN BE FUNCTIONS TOO!

```
var rectangle = {
    length : 5.0,
    breadth : 3.5,
    color : 'Red',
    area : function() {
        return this.length * this.breadth;
    }
};
```

IT HAS 4 PROPERTIES..

EXAMPLE 19: PROPERTIES CAN BE FUNCTIONS TOO!

```
var rectangle = {
    length : 5.0,
    breadth : 3.5,
    color : 'Red',
    area : function() {
        return this.length * this.breadth;
    }
};
```

ONE OF WHICH IS A FUNCTION!

UNDER THE HOOD, FUNCTIONS ARE OBJECTS TOO.

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

```
our rectangle has area = 17.5
```

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

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

UNDER THE HOOD, FUNCTIONS ARE OBJECTS TOO.

```
rectangle["area"]);
```

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

ACCESSING THE FUNCTION WITH THIS SYNTAX PRINTS OUT THE ACTUAL FUNCTION - IT DOES NOT EXECUTE THE FUNCTION!