Web Development

JavaScript Arrays & Events

Arrays

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

Arrays

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
var array_name = [item1, item2, ...];
```

Access the Elements of an Array

```
var name = cars[0];
```

Arrays

The **length** property of an array returns the length of an array (the number of array elements).

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.length;                          // the length of fruits is 4
```

Arrays and Objects

In JavaScript, **arrays** use **numbered indexes**.

```
var person = [];
person[0] = "John";
```

In JavaScript, **objects** use **named indexes**.

```
var person = [];
person["firstName"] = "John";
```

Arrays are a special kind of objects, with numbered indexes.

How to Recognize an Array

The problem is that the JavaScript operator **typeof** returns "object".

1. ECMAScript 5 defines a new method **Array.isArray()**

```
Array.isArray(fruits);     // returns true
```

2. The **instanceof** operator returns true if an object is created by a given constructor.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits instanceof Array     // returns true
```

Array Methods

Example

Events

HTML events are **"things"** that happen to HTML elements.

When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

JavaScript lets you execute code when events are detected.

Event Handler Attributes

HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

```
<element event="some JavaScript">
```

```
<button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>
```

Common HTML Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

MORE

JS Hoisting

- In JavaScript, a variable can be declared after it has been used. In other words; a variable can be used before it has been declared.

- Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope (to the top of the current script or the current function).

- JavaScript only hoists declarations, not initializations.

DEMO

JavaScript strict mode

`"use strict";` Defines that JavaScript code should be executed in "strict mode". (ECMAScript version 5).

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".

With strict mode, you can not use undeclared variables.

Strict mode makes it easier to write "secure" JavaScript.

Strict mode changes previously accepted "bad syntax" into real errors.

The "use strict" directive is only recognized at the **beginning** of a script or a function.

EXAMPLE