Javascript

January 2020

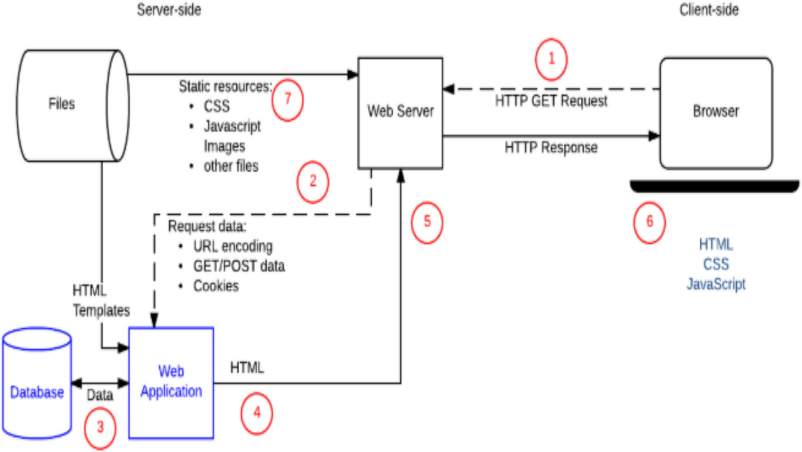
Introduction

- *JavaScript* was initially created to develop interactive/reactive pages.
- Modern JavaScript is a "safe" programming language. It does not provide low-level access to memory or CPU, because it was initially created for browsers which do not require it.
- JavaScript's capabilities greatly depend on the environment it's running in. For instance, Node.js supports functions that allow JavaScript to read/write arbitrary files, perform network requests, etc.
- In-browser JavaScript can do everything related to validation of input, interact with server, handling cookies and time-sensitive events.

Limitations of client-side scripting

- Since script code is embedded in the page, viewable to the world
- For security reasons, scripts are limited in what they can do. e.g., can't access the client's hard drive
- Since designed to run on any machine platform, scripts do not contain platform specific commands.
- Script languages are not full-featured. e.g., JavaScript objects are crude, not good for large project development

Simple architecture for a *dynamic website*



Example

<script> and </script> tells where the JavaScript starts and ends.

```
<html>
<body>
    <h1>My First Web Page</h1>
    <p id="demo">This is a paragraph.</p>
    <script type="text/javascript">

    </script>
</body>
</html>
```

Common HTML 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

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

JavaScript Functions and Events

```
<!DOCTYPE html>
<html><head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try</button>

</body>
</html>
```

JavaScript Popup Boxes

- Alert Box - An alert box is often used if you want to make sure information comes through to the user.

  ```
  alert("I am an alert box!");
  ```

- Confirm Box - A confirm box is often used if you want the user to verify or accept something.

  ```
  if (confirm("Press a button!")) {
          txt = "You pressed OK!";
  } else {
          txt = "You pressed Cancel!";
  }
  ```

- Prompt Box - A prompt box is often used if you want the user to input a value before entering a page.

  ```
  var person = prompt("Please enter your name", "Harry Potter");

  if (person == null || person == "") {
    txt = "User cancelled the prompt.";
  } else {
    txt = "Hello " + person + "! How are you today?";
  }
  ```

JavaScript Timing Events

The two key methods to use with JavaScript are:

- **setTimeout(function, milliseconds)** - Executes a function, after waiting a specified number of milliseconds.
- **setInterval(function, milliseconds)** - Same as setTimeout(), but repeats the execution of the function continuously.

setTimeout() Method

```html
<!DOCTYPE html>
<html>
<head>
    <script>
        function myFunction() {
            alert('Hello');
        }
    </script>
</head>
<body>
    <p>Wait 3 seconds, and the page will alert "Hello".</p>
    <button onclick="setTimeout(myFunction, 3000);">Try it</button>
</body>
</html>
```

setInterval() Method

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function myFunction() {
            alert('Hello');
        }
    </script>
</head>
<body>
    <p>The page will alert Hello every 3 seconds.</p>
    <button onclick="setInterval(myFunction, 3000);">Try it</button>
</body>
</html>
```

Client-side Validation

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title></title>
    <script type="text/javascript">
        function validate() {
            if (document.getElementById("txtName").value == "") {
                return false;
            }
            else {
                return true;
            }
        }
    </script>
</head>
<body>
    <form onsubmit="return validate()">
        Name: <input type="text" id="txtName" /><br/>
        <input type="submit" value="SUBMIT" />
    </form>
</body>
</html>
```