Web Development

Javascript - Review

Javascript Basics

The syntax of Javascript is very similar to PHP with a few exceptions

- Declaration of variables, no $, use **var** keyword
- String Concatenation ‘+’ instead of ‘.’
- strlen() vs .length

Javascript runs on the client side **in the browser**

Javascript code is **interpreted** by the browser

Javascript Basics

For Javascript the most important HTML tag is **<script>.**

This tag allows us to include a piece of Javascript in a document.

```
<h1>Testing alert</h1>
<script>alert("hello!");</script>
```

Such a script will run as soon as its <script> tag is encountered as the browser reads the HTML.

The page shown above will pop up an alert dialog when encountered.

Javascript Basics

Including large programs directly in HTML documents is often impractical. The **<script>** tag can be given an src attribute in order to fetch a script file (a text file containing a JavaScript program) from a URL.

```
<h1>Testing alert</h1>
<script src="js/myfile.js"></script>
```

The *js/myfile.js* file included here contains any javascript code. When an HTML page references other URLs as part of itself, for example an image file or a script—web browsers will retrieve them immediately and include them in the page.

DOM

When a web page is shown in the browser, each of the different elements e.g. Input Field, Text Field, Radio Buttons etc are all considered “Elements”.

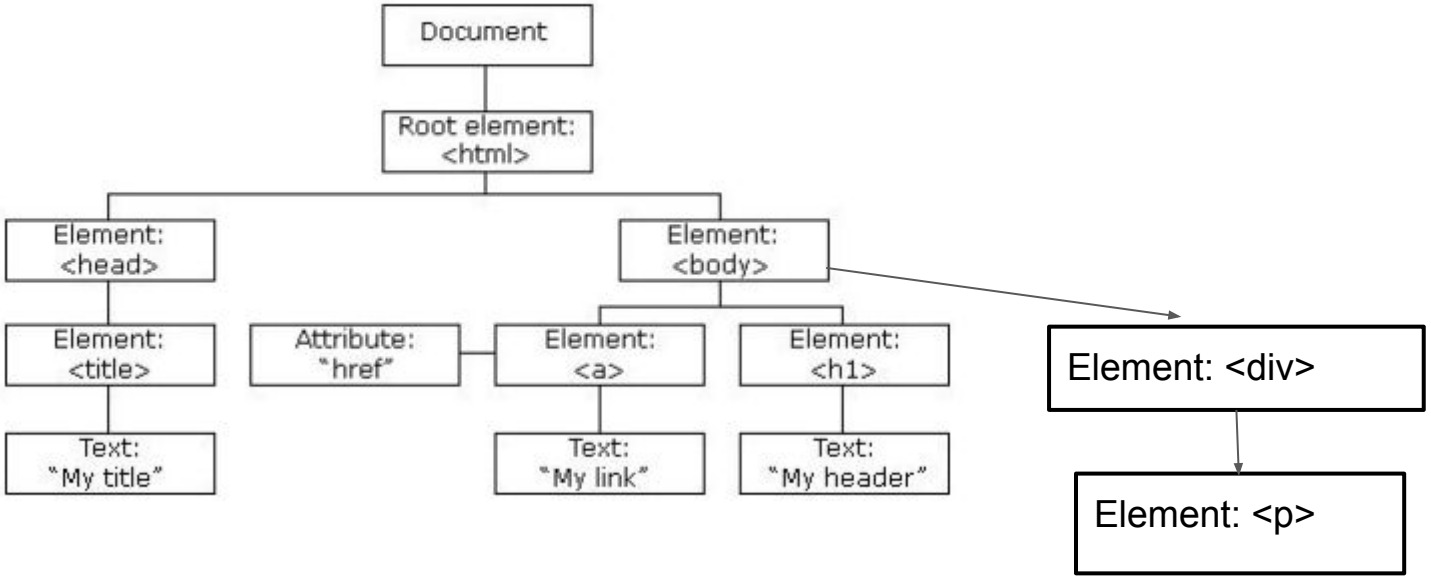
Each of these different elements are added to a Tree type structure called the **Document Object Model** (DOM).

Think about the web page being the root of the tree and each of the elements added to the tree are different branches coming off the tree. Each of the elements has different attributes such as the name, id, value.

DOM

```
<html>
<head>
<title>My Title</title>
</head>
<body>
<h1>My Header</h1>
<a href='www.google.com'>my link</a>
<div>
    <p> this is a paragraph</p>
</div>
</body>
</html>
```

DOM


Document
Root element:
<html>
Element:
<head>
Element:
<body>
Element:
<title>
Attribute:
"href"
Element:
<a>
Element:
<h1>
Text:
"My title"
Text:
"My link"
Text:
"My header"
Element:
Element: <p>

DOM

There is a parent/child relationship between elements

In the previous example <h1> is said to be a **child** of the <body> element

<body> is a **parent** of the <h1> element

DOM

In javascript the root element is defined by the javascript object

document.

There are a number of functions we can use to ‘traverse’ or interrogate the document object i.e. elements of the DOM

DOM

If we want, after the web page has loaded, we can pick one element from the DOM tree and check to see what the values current are.

we first start off by using the **getElementById** function. This will go off and find the element in the tree for us .

e.g. ***document.getElementById('id')*** // we pass the id of the element

Javascript Events

When we want a function to be called when something happens on a page, we use event handlers.

These are just like the ActionListener and ActionPerformed in Java.

When a button is pressed we can call a function:

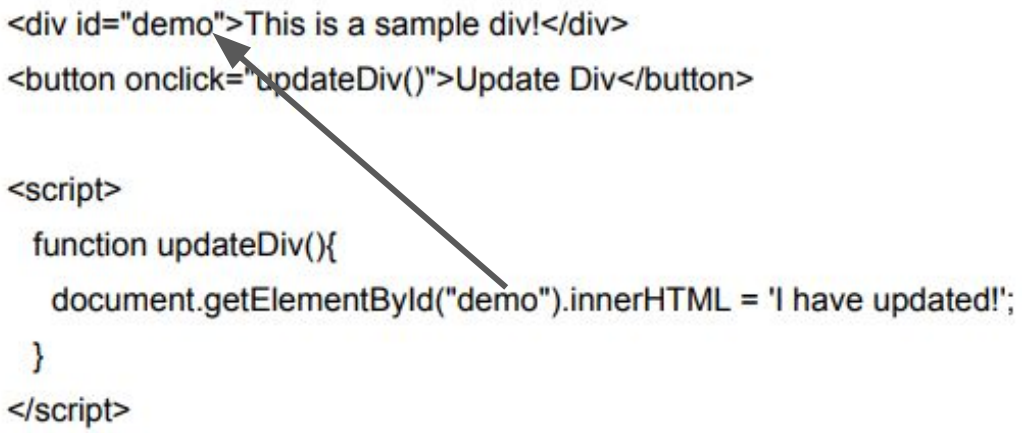
*<button **onclick**=’callFunction()’>Submit</button>*

DOM

When we call the function document.getElementById() it will return a reference to the element in the DOM tree where that element is.

Once we get this reference, we can then make changes to the element. The **innerHTML** function allows us to set the HTML of an element dynamically.

In this example we are just adding text but we could add any html we like e;g; <p>, , <a> etc,



DOM

```
1  <html>
2  <head>
3  </head>
4  <body>
5  <div id='demo'>
6  This is a sample div
7  </div>
8  <button onclick='updateDiv()'>Submit</button>
9  </body>
10 <script>
11 function updateDiv(){
12   document.getElementById('demo').innerHTML='i have updated';
13 }
14 </script>
15 </html>
```

DOM

Updating Style

```
<div id="demo">This is a sample div!</div>
<button onclick="updateDiv()">Update Div</button>

<script>
  function updateDiv(){
    document.getElementById("demo").style.color = "blue";
  }
</script>
```

DOM - Value

We can extract the value of an element e.g. input element in a form using the

value method

```
document.getElementById(‘id’).value
```

DOM - Validation

Execute function on submit

```
1  <html>
2  <body>
3  <form id='form1' action='login.php' method='post' onsubmit="return validate()">
4      username:  <input type='text' name='username' id='username'><br/>
5      email: <input type='text' name='email' id='email'><br/>
6      <input type='submit' value='Submit'>
7  </form>
8  </body>
9  </html>
10
11 <script>
12 function validate(){
13   var test=document.getElementById('username');
14   if (test.value == "") {
15       alert("Please Enter A Value");
16       return false;
17     }
18    else
19       return true;
20 }
21 </script>
```

Javascript Validation

Return the number of characters in a string:

```
var str = "Hello World!";
var n = str.length;
```

The result of *n* will be:

```
12
```

Exercise:

Amend the validate function to check that username is more than 3 characters

Javascript Validation

```
<html>
<head>
</head>
<body>
<form id='form1' action='jslogin.php' method='post' onsubmit='return validate()'>
username: <input type='text' id='username' name='username' ><br/>
email: <input type='email' name='email'  ><br/>
<input type='submit' value='submit'>
</form>
<script>
function validate(){
  var test= document.getElementById('username');
  if (test.value == "" || test.value.length < 4 ){
     alert("Please Enter Username");
     return false;
  }
  else
     return true;
}
</script>
</html>
```

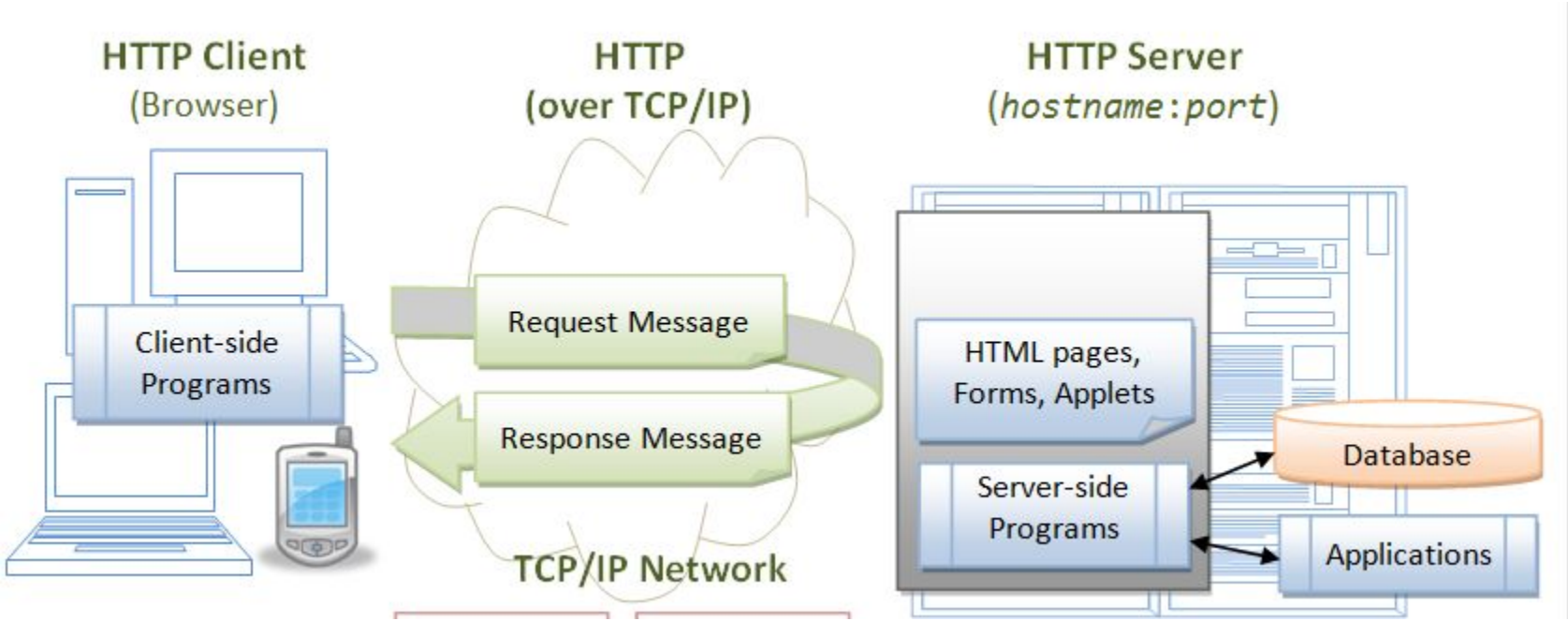
Javascript/Client Side Summary

Things to concentrate on

- Understand the DOM tree and how to draw it from a HTML document
- Understand how to add javascript validation to a html page
- How do client side and server side programming differ
- Where and how do client side and server side programs get executed
- Client Side Validation HTML5 Vs Javascript

    HTML5 limited options i.e. number of attributes min,max,required

    Javascript, more customizable, better error messages e.g

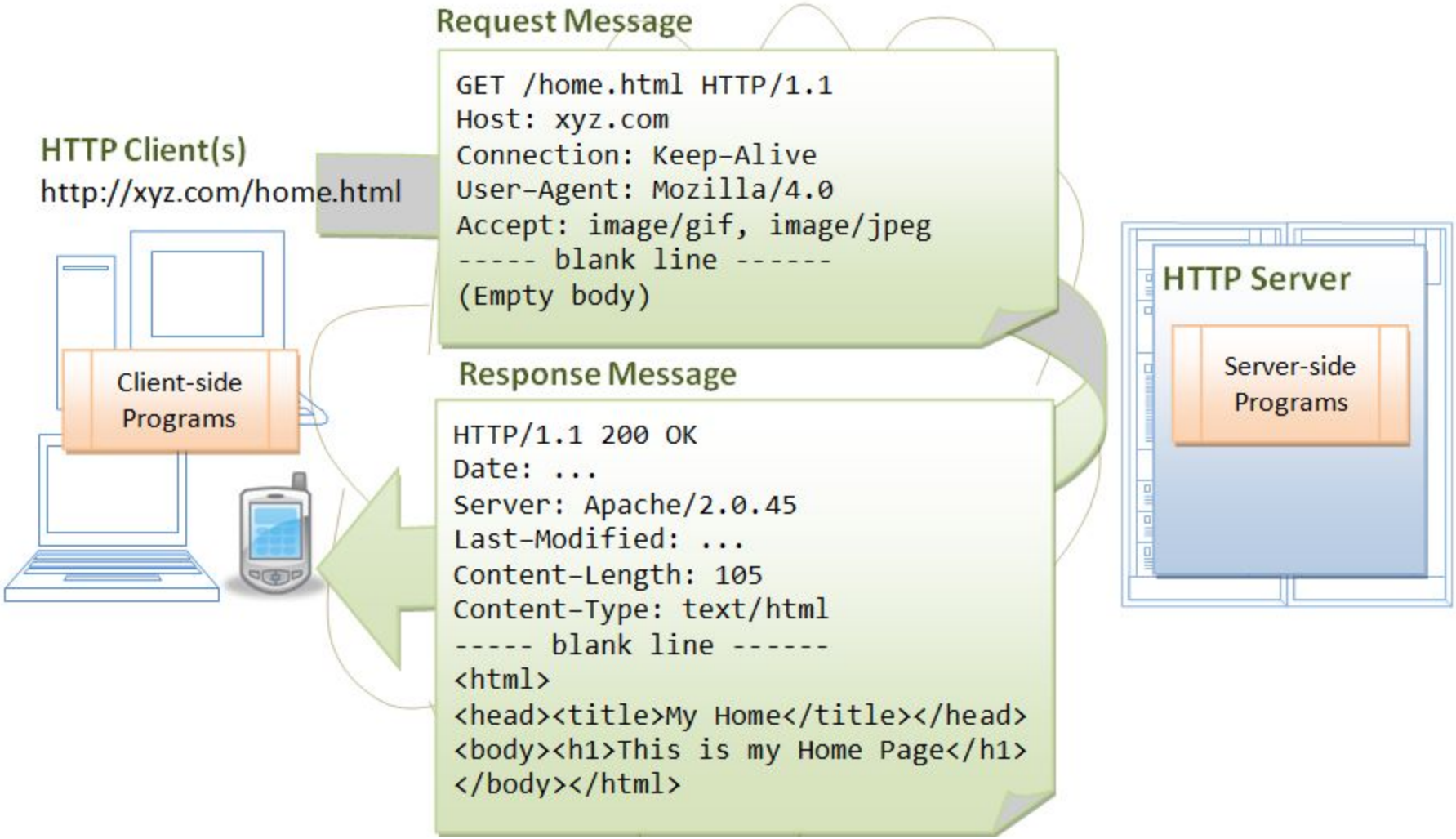

HTTP Client
(Browser)
HTTP
(over TCP/IP)
HTTP Server
(hostname:port)
Client-side
Programs
Request Message
Response Message
TCP/IP Network
HTML pages,
Forms, Applets
Server-side
Programs
Database
Applications

HTTP Request

Step 1 Web address entered by a user is translated into an I.P address using a Domain Name Server (DNS) who has the job of translating the text the user has entered into the machine readable I.P address

Step 2 The I.P address of the website we are looking for is given to us and then the web browser makes a connection to the web server at the location specified by the address and asks for the page we are requesting.

Step 3 The web server looks for the page we are requesting and then sends it back to the user after executing any server side code e.g. PHP within the file (response).


Request Message
GET /home.html HTTP/1.1
Host: xyz.com
Connection: Keep-Alive
User-Agent: Mozilla/4.0
Accept: image/gif, image/jpeg
----- blank line ------
(Empty body)
HTTP Client(s)
http://xyz.com/home.html
Client-side Programs
Response Message
HTTP/1.1 200 OK
Date: ...
Server: Apache/2.0.45
Last-Modified: ...
Content-Length: 105
Content-Type: text/html
----- blank line ------
<html>
<head><title>My Home</title></head>
<body><h1>This is my Home Page</h1>
</body></html>
HTTP Server
Server-side Programs