Introduction to JS

Aleksander Fabijan

Recap from last time

1. We **introduced** JavaScript and its history,
2. We learned how to **debug** JavaScript,
3. We introduced **Variables**, **Statements** and **Expressions**,
4. We introduced **Data Types** (String, objects)
5. We introduced **functions**.

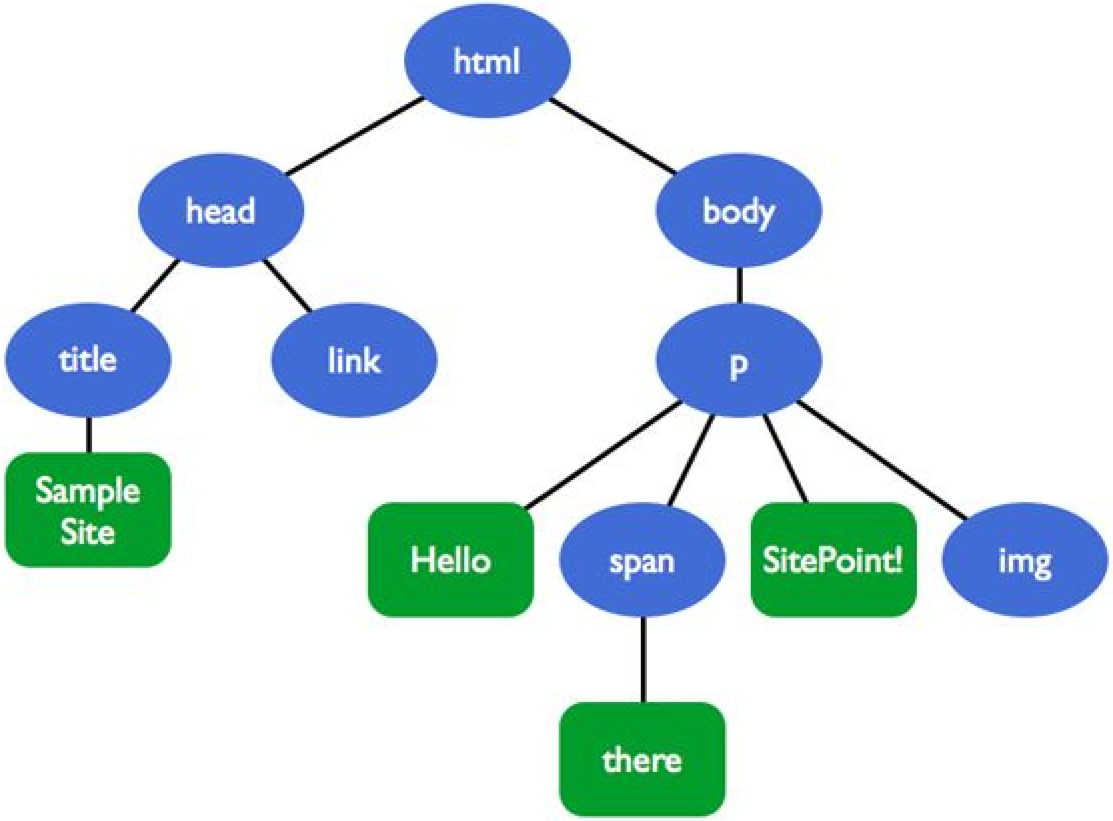
3.

DOM

The HTML DOM (Document Object Model)

When an HTML document is loaded into a web browser, it becomes a **document object**.

Every element in an HTML document is a node.



html

head

body

title

link

p

Sample Site

Hello

span

SitePoint!

img

there

Accessing Html elements/nodes

To access a certain node, we can help ourselves with already available functions.

Access by id:

```
var myNode = document.querySelector("#someID");
```

… this stores the reference no the node with id == “someID” to the variable “myNode”

ID vs CLASS in HTML

A typical HTML file will have two types of identification elements:

(1) **ID**

   (a) Unique and can only appear once!!!
   (b) We refer to ID tags with “#id_name.”

(2) **CLASS**

   (a) Class (not unique and can/should be used multiple times)
   (b) We refer to “class” elements with “.class”

Use a class when you want to consistently style multiple elements throughout the page/site.

Example use of an ID

```
<html>
<body>

    <h1 id="myHeader">Hello World!</h1>
    <button  onclick="displayResult()">Change text</button>

    <script>
        function displayResult() {
        document.getElementById("myHeader").innerHTML ="ok";
        }
    </script>

</body>
</html>
```

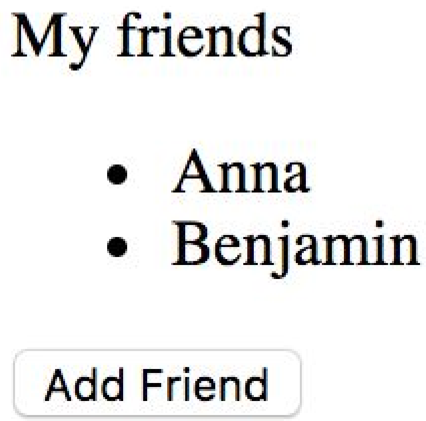
Example use of a CLASS

```html
<!DOCTYPE html>
<html><head><style>
div.cities {
     background-color: black;
     color: white;
     margin: 20px 0 20px 0;
     padding: 20px;
}
</style></head><body>
     <div class="cities">
     <h2>London</h2>
     <p>London is the capital of England. It is the most populous city in the United Kingdom,
     with a metropolitan area of over 13 million inhabitants.</p>
     </div>

     <div class="cities">
     <h2>Paris</h2>
     <p>Paris is the capital and most populous city of France.</p>
     </div>
</body>

</html>
```

Example with both



My friends

- Anna
- Benjamin

```
…
<body>
<p id =”myFriendsText”> My friends: </p>
<ul id =”myFriendsList”>
    <li class=”blue”>Lilly </li>
    <li class=”blue”>Sara </li>
</ul>
</body>
```

http://html5-demos.appspot.com/shadowdom-visualizer

> *How do we programmatically add a new element(=node) to a webpage?*

In natural language...

(1) We create a new element of type Text and give it a value

(2) We create a new element of type <li>.

(3) We attach the text node to <li> node

(4) We attach the <li> node to the document <ul> node.

```
document.createTextNode(“...”);   //creates a text node
document.createElement("li");    //creates an element node
parentNode.appendChild(childNode);
```

Adding elements/nodes

Example 2: Add new list item to #myFriends <ul>

```
//creating a new text node
var textNode = document.createTextNode(“Andrew”);

//creating a new list node
 var liNode = document.createElement("li");

//appending the list tex to the list node
 liNode.appendChild(textNode);

//appending the list item to the list itself
 document.querySelector("#myFriendsList").appendChild(liNode);
```

Appending a new element/node / example 2

To add new elements to the document we (1) need to create them, and (2) use appendChild method to actually append them.

```
//Creating a new paragraph node
var pNode = document.createElement(“p”);

//Create a new text node
var textNode = document.createTextNode(“Just some text”);

//append the text node to the pnode
pNode.appendChild(textNode);

//appending a paragraph node to the body
 document.querySelector("body").appendChild(pNode);
```

3.

Event listeners

Reacting on Events in JS

JavaScript needs a way of detecting user actions so that it knows when to react.

For that, we use event listeners.

http://www.quirksmode.org/js/introevents.html

.addEventListener method

```
element.addEventListener(event,
function, useCapture);
```

- The first parameter is the type of the event (like "click" or "mousedown").
- The second parameter is the function we want to call when the event occurs.
- The third parameter is a boolean value specifying whether to use event bubbling or event capturing. This parameter is optional.

Example

```
<!DOCTYPE html>
<html>
<body>

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>
document.getElementById("myBtn").addEventListener("click", displayDate);

function displayDate() {
    document.getElementById("demo").innerHTML = Date();
}
</script>
</body>
</html>
```

Example from Before

```
function myAddFunction(randomText)
{
  var pNode = document.createTextNode(randomText);
  var liNode = document.createElement("li");
  liNode.appendChild(pNode);

  document.querySelector("#myFriendsList").appendChild(liNode);
}



document.querySelector("#addFriend").addEventListener("click", function() {
myAddFunction("random_text_here");}, false);
```

Summary

(1) We introduced the DOM,
(2) We repeated what #id and .class are,
(3) We learned how to create new nodes and add them to the DOM,
(4) We learned how to handle events.

4.

Resources

Useful resources

https://jsbin.com (write and execute JS online)

https://jsfiddle.net/ (JS, CSS and HTML playground)

https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript (a re-intro to JS)

https://www.kirupa.com/html5/event_capturing_bubbling_javascript.htm (event bubling)

Thanks!

You can find me at:

aleksander.fabijan@mah.se