Date: 05-6-2020
Morning Session: 9 am – 11 PM
By ~ Sundeep Charan Ramkumar Today

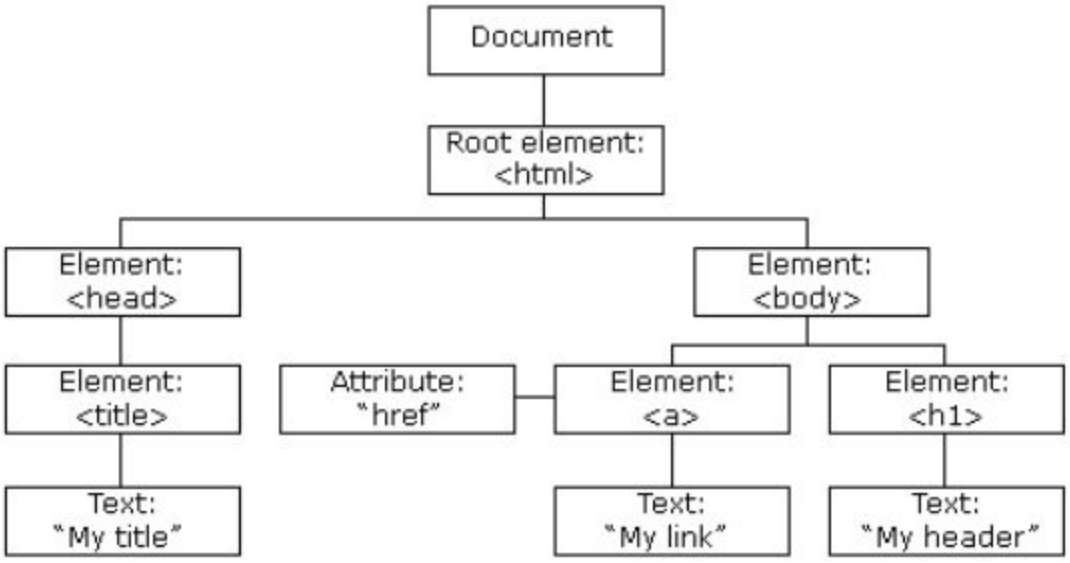
Topics: Introduction to DOM

DOM: Document Object Model (DOM) we are modeling HTML documents in a way to make sure that even that last most element can be accessible in object format, That is called a DOM.

DOM converts HTML to javascript native elements..

Binary tree data structure is the container is the ay to javascript could able to reach the element structure

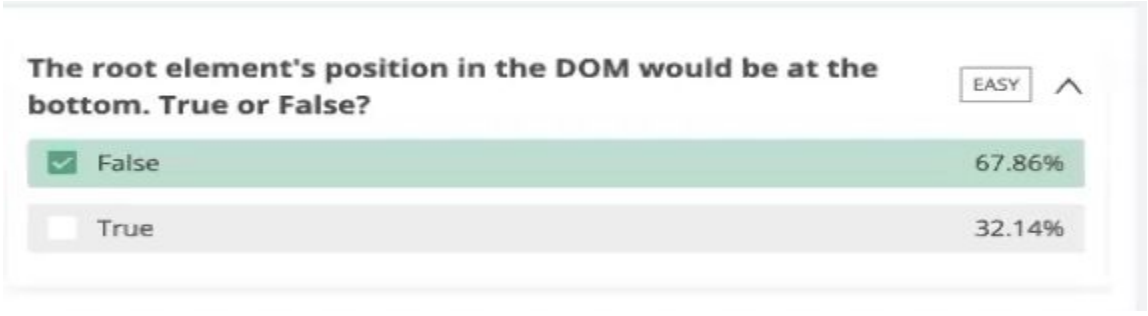
DOM TREE:



DOM Selectors:

Note: always use id's for javascript purposes.

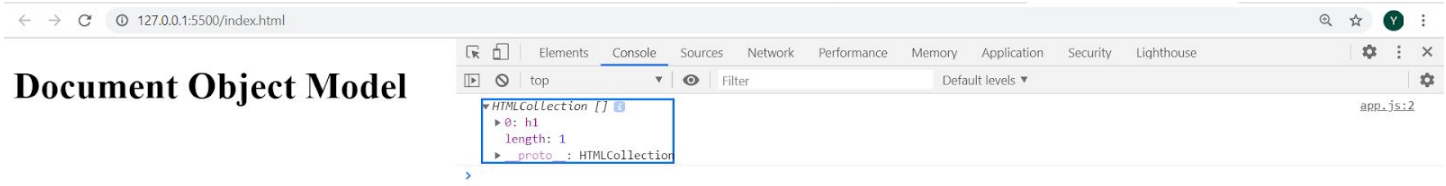
MCQ1:



Getting Elements BY Tag Name:

- tag names are unique

```
JS app.js > ...
1   var h1Elemennts = document.getElementsByTagName('h1')
2   console.log(h1Elemennts);
```



In the console we found a special type called HTML collections. It might look like a mimit array but it doesn't have freedom of having all those methods and which array has.

How To Converting to Array:

ES5 Method

```
> Array.from(h1Elemennts)
< [h1]
    0: h1
    length: 1
  ▶ __proto__: Array(0)
>
```

Whenever we see the key word called iterable it means that it has an index otherwise it’s not able to traverse through the entire collection.

ES6 Method:

```
[...h1Elements]
▶ [h1]
```

Note: elements do not change but the type of collection will change.

Getting Elements BY Class:

```
var divContainers = document.getElementsByClassName("container");
console.log(divContainers);
```

Getting Elements BY ID:

```
var spanUnique = document.getElementById("unique");
console.log(spanUnique);
```

⚠ **Don't forget the `"s"` letter!**

Novice developers sometimes forget the letter `"s"`. That is, they try to call `getElementByTagName` instead of `getElementsByTagName`.

The `"s"` letter is absent in `getElementById`, because it returns a single element. But `getElementsByTagName` returns a collection of elements, so there's `"s"` inside.

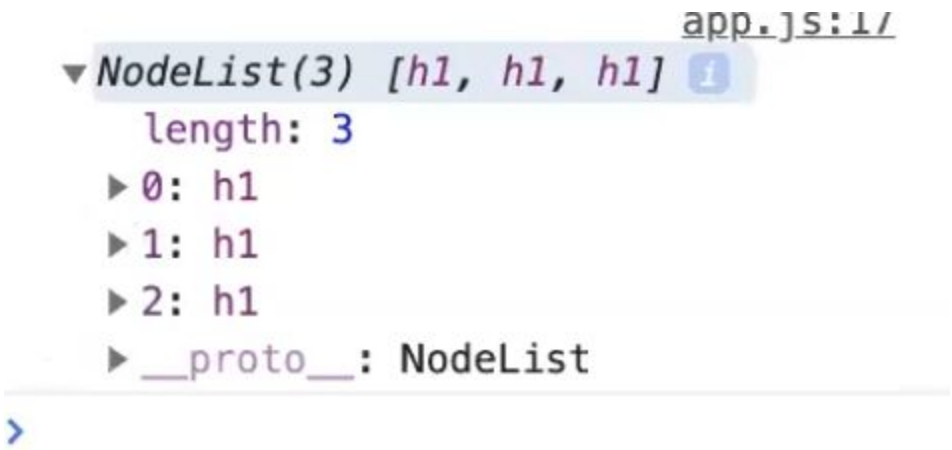
querySelector: it will return one element irrespective of duplication

```
var h1Element = document.querySelector("h1");
console.log(h1Element);
```

querySelectorAll: it will find all the duplicates and return all the collections.

```
var h1Elements = document.querySelectorAll("h1");
console.log(h1Elements);
```

we found another special type **NODEList**



MCQ 2:

CALLBACKS: A Way to Automatically invoke a function that has been passed as an argument to another function.

```
function checkNumber(callBackFunction, number) {
  // Checks the number position between 10 and 20
  var errorMessage = null;
  if (number < 10) {
    // The number is less than 10
    errorMessage = "The number is less than 10";
    callBackFunction(errorMessage, null);
  } else if (number > 20) {
    // The number is greater than 20
    errorMessage = "The number is greater than 20";
    callBackFunction(errorMessage, null);
  } else {
    // The number is between 10 and 20.
    callBackFunction(null, "The number " + number + " is between 10 and 20");
  }
}
```

```
// Callback function
function callBackFunction(err, result) {
  console.log(err, result);
  if (err !== null) {
    alert("ERROR: " + err);
  } else {
    console.log(result);
  }
}

checkNumber(callBackFunction, 9);
```

Events: is a way to capture user interaction.

Resource: https://developer.mozilla.org/en-US/docs/Web/API/Event

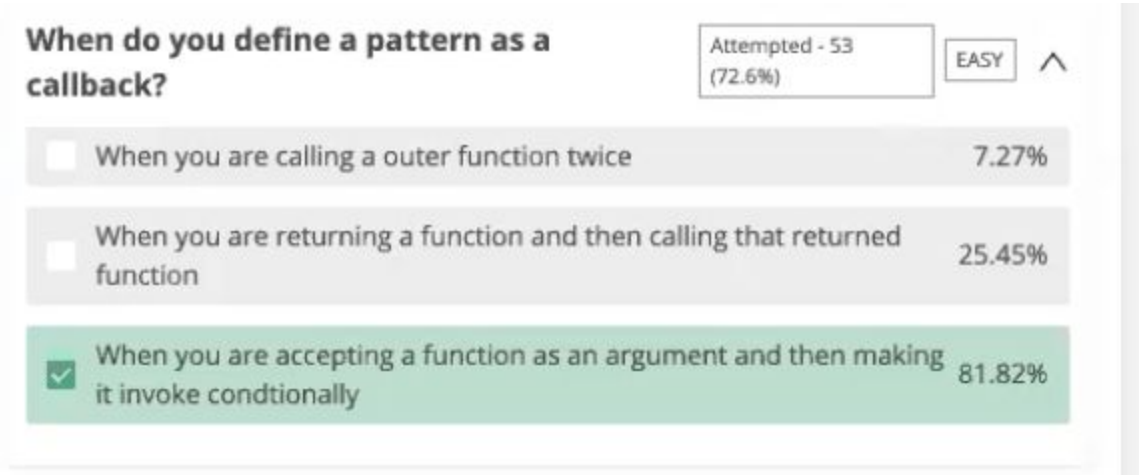
```
var h2Element = document.querySelector("h2");
function clickCallbackFunction(event) {
  console.log(event);
}
h2Element.addEventListener("click", clickCallbackFunction);
```

Event Target:

```
var h2Element = document.querySelector("h2");
function clickCallbackFunction(event) {
  console.log(event.target);
}
h2Element.addEventListener("click", clickCallbackFunction);
```

```
var h2Element = document.querySelector("h2");
function clickCallbackFunction(event) {
  console.log("I am being clicked");
}
h2Element.addEventListener("click", clickCallbackFunction);
```

MCQ 3:


When do you define a pattern as a callback?
Attempted - 53 (72.6%)
EASY
When you are calling a outer function twice
7.27%
When you are returning a function and then calling that returned function
25.45%
When you are accepting a function as an argument and then making it invoke condtionally
81.82%

MCQ 4:


Are events instantaneous?
EASY
Yes
63.08%
No
41.54%