DOM & DOM API

Rule the UI

Agenda

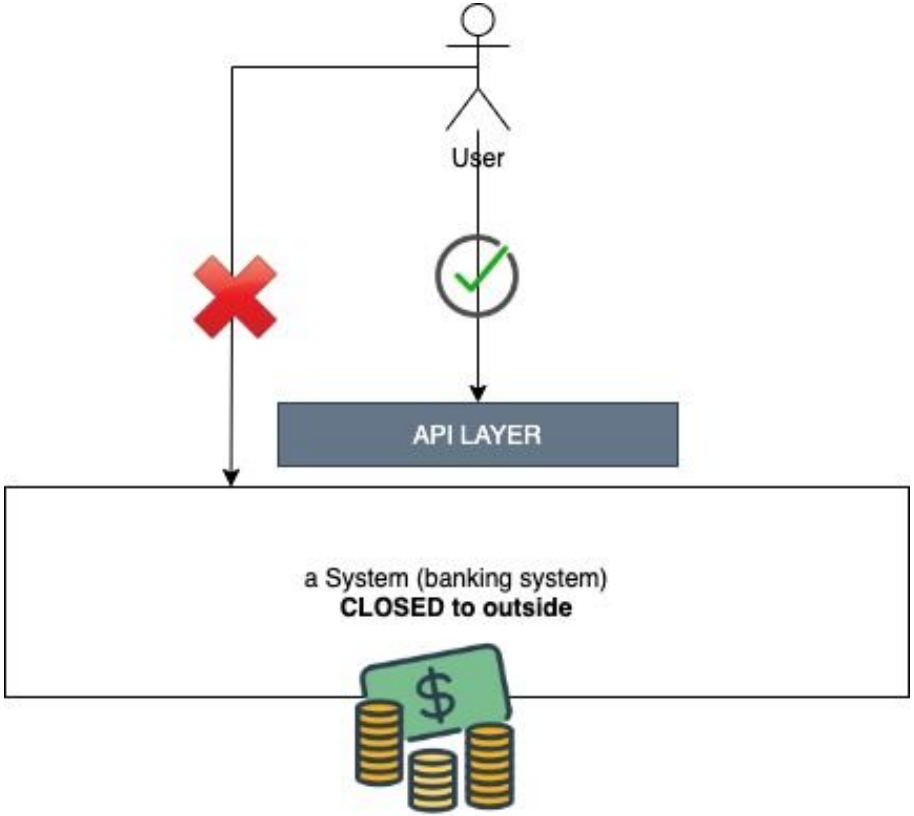
- What is DOM?
- A Tree? Why?
- The Rendering Process
- What is the DOM API?
- DOM Methods and Attributes
- What is BOM?
- What is SPA?
- Examples

Learn Objectives

- You understand what an API is.
- You know the need of the dom how to access it (dom API)
- You know the most important dom operations
- You know how to interact with user’s actions (event handlers/listeners)
- You know what the SPA is.
- You know the bom and bom api

What's an API?

- A provided way of accessing to sandboxed/blackboxed system
- An API has operations (methods, functions), objects
- An API is allowed to change the internal state of a system (bank system)
- Widely known usages
  - ReST API
  - SOAP API
  - Web API
  - Standard API
  - Domain API
  - X, Y, Z ... API

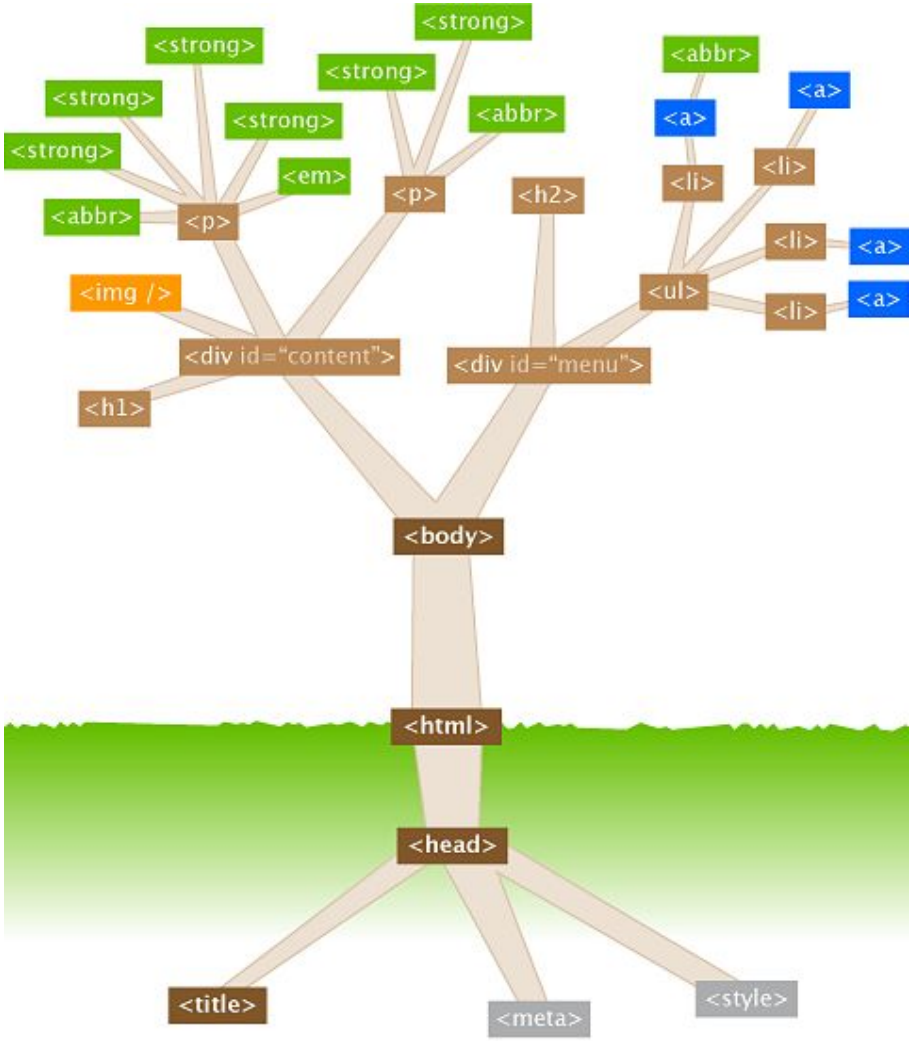


How an API enables the access to a closed system.

What is DOM?

- DOM stands for “Document Object Model”
- It is a memory representation of the downloaded html document
- It represents the whole html structure including CSS, Javascript and other web content providers.
- It is actually a tree structure from the software engineering point of view.

DOM (DOM Tree)



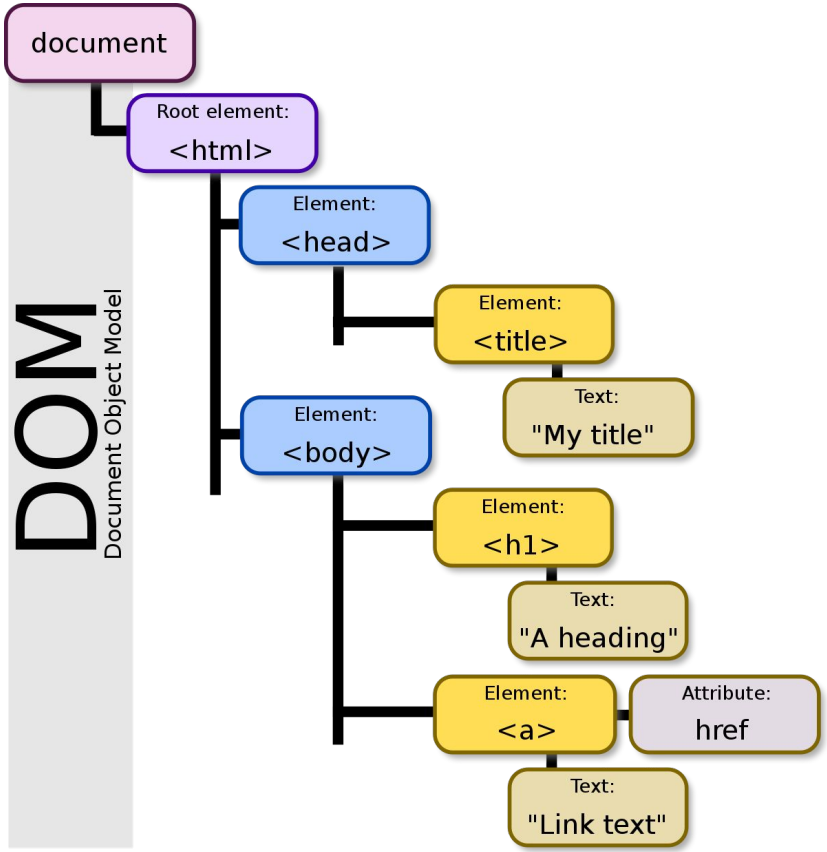
<abbr>
<p>
<abbr>
<p>
<h2>
<abbr>
<a>
<a>

<a>

<a>
<img />

<h1>

<body>
<html>
<head>
<title>
<meta>
<style>

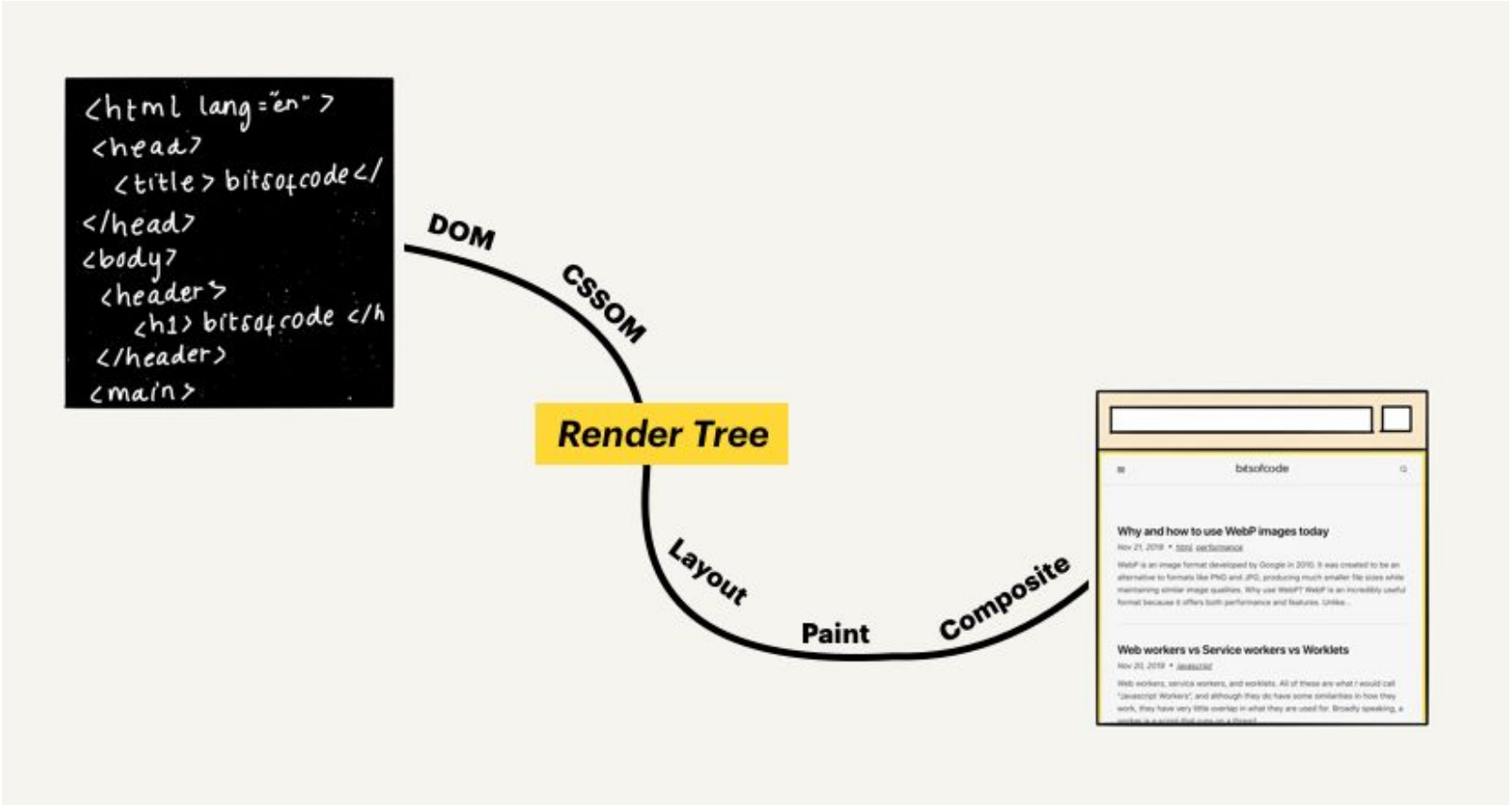


Source: wikipedia

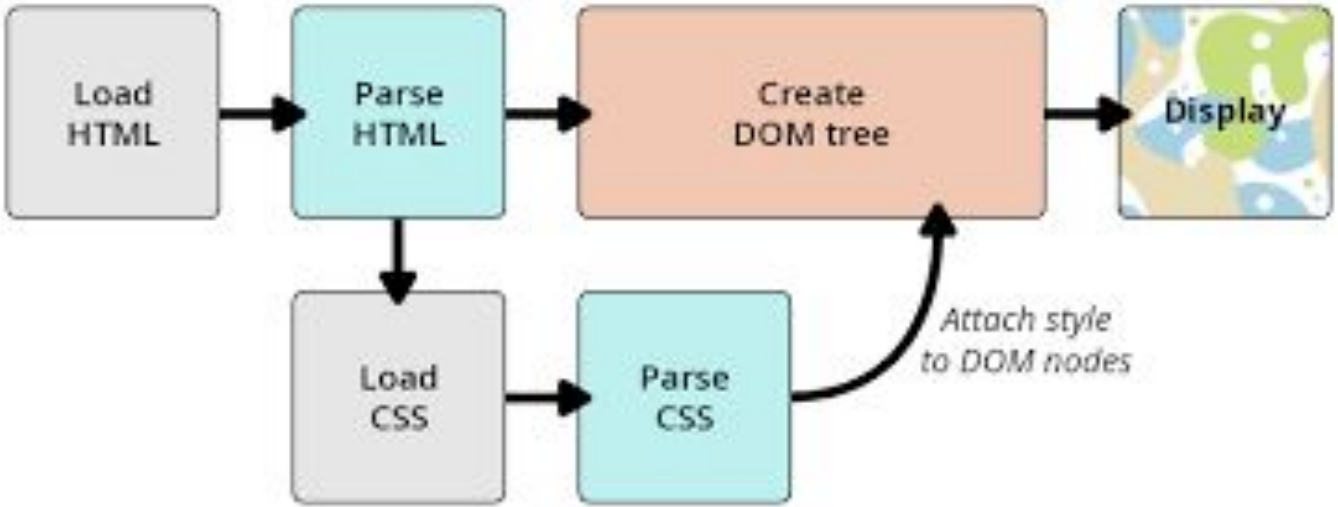
(https://en.wikipedia.org/wiki/Document_Object_Model#/media/File:DOM-model.svg)

Rendering Process

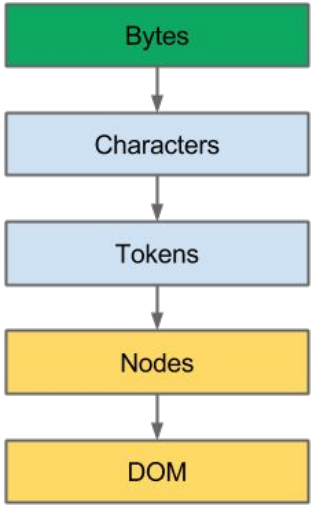
- Rendering process is the process beginning from the first byte hitting the browser and finally display the page to the user.
- It contains HTML&CSS parsing, creating memory model (DOM), applying JS, layouting, etc…
- The process simply creates an image based on the html document downloaded from the server.



The rendering process (source: https://bitsofco.de/what-exactly-is-the-dom/)



General Overview of the rendering process (source:)

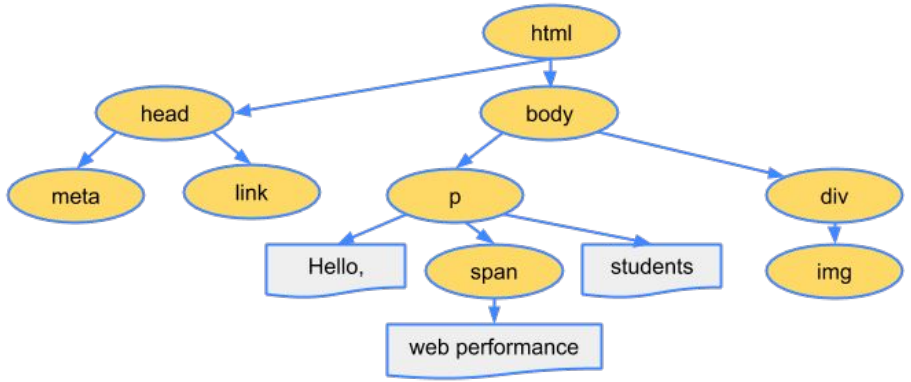


3C 62 6F 64 79 3E 48 65 6C 6C 6F 2C 20 3C 73 70 61 6E 3E 77 6F 72 6C 64 21 3C 2F 73 70 61 6E 3E 3C 2F 62 6F 64 79 3E

<html><head>...</head><body><p>Hello web performance...

StartTag: html | **StartTag:** head | ... | **EndTag:** head | **StartTag:** body | **StartTag:** p | Hello | ...

html | head | meta | body | p | Hello




```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <link href="style.css" rel="stylesheet">
    <title>Critical Path</title>
  </head>
  <body>
    <p>Hello <span>web performance</span> students!</p>
    <div><img src="awesome-photo.jpg"></div>
  </body>
</html>
```

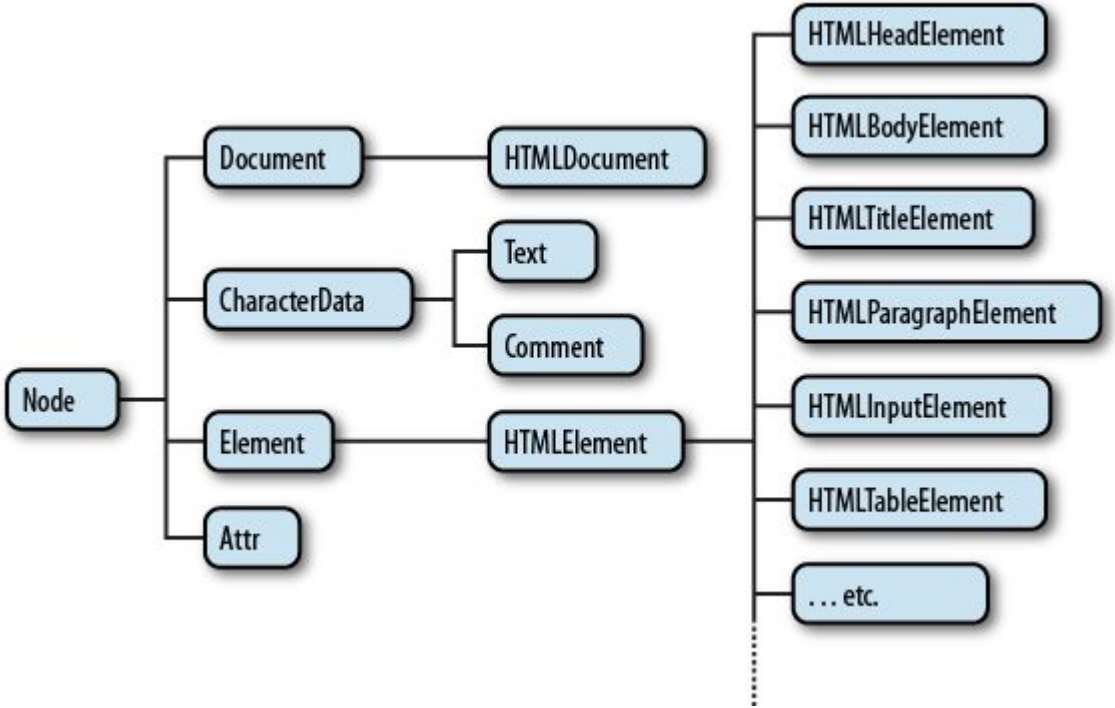
Actual happenings in the transport layer (source: https://developers.google.com/web/fundamentals/performance/critical-rendering-path/constructing-the-object-model)

DOM Element Types

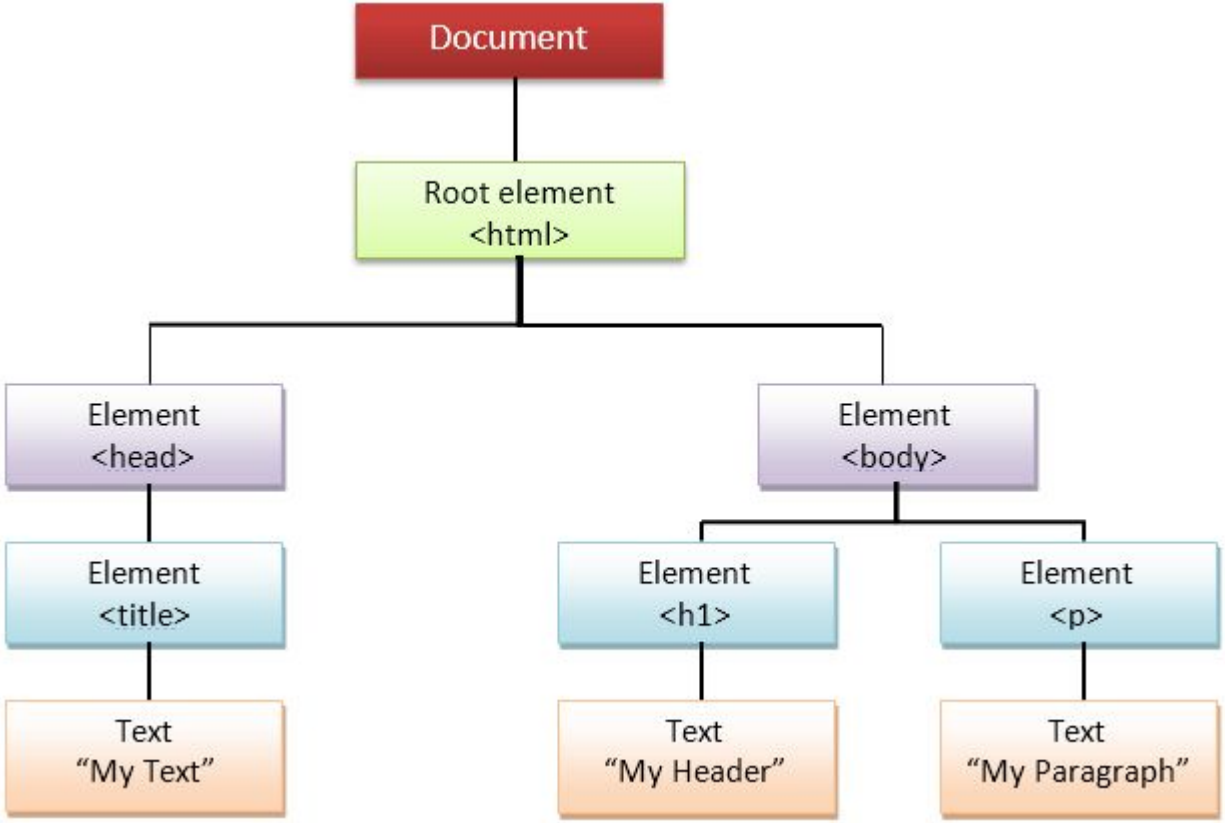
The DOM structure is almost equivalent to any real tree. It has also its own internal structures to represent the html document.

It has

- a root
- body
- branches
- leafs
- others



DOM Elements or Node Types (source: https://web.stanford.edu/class/cs98si/slides/the-document-object-model.html)



An example to DOM Types (source: https://seleniumsubbu.blogspot.com/2015/12/how-to-use-dom-and-events-in-javascript.html)

DOM API

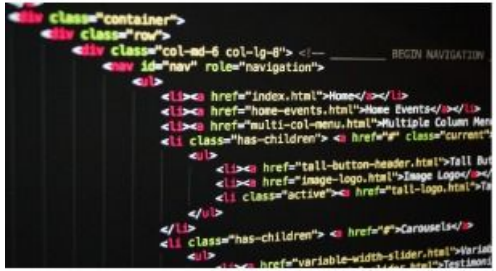
DOM API

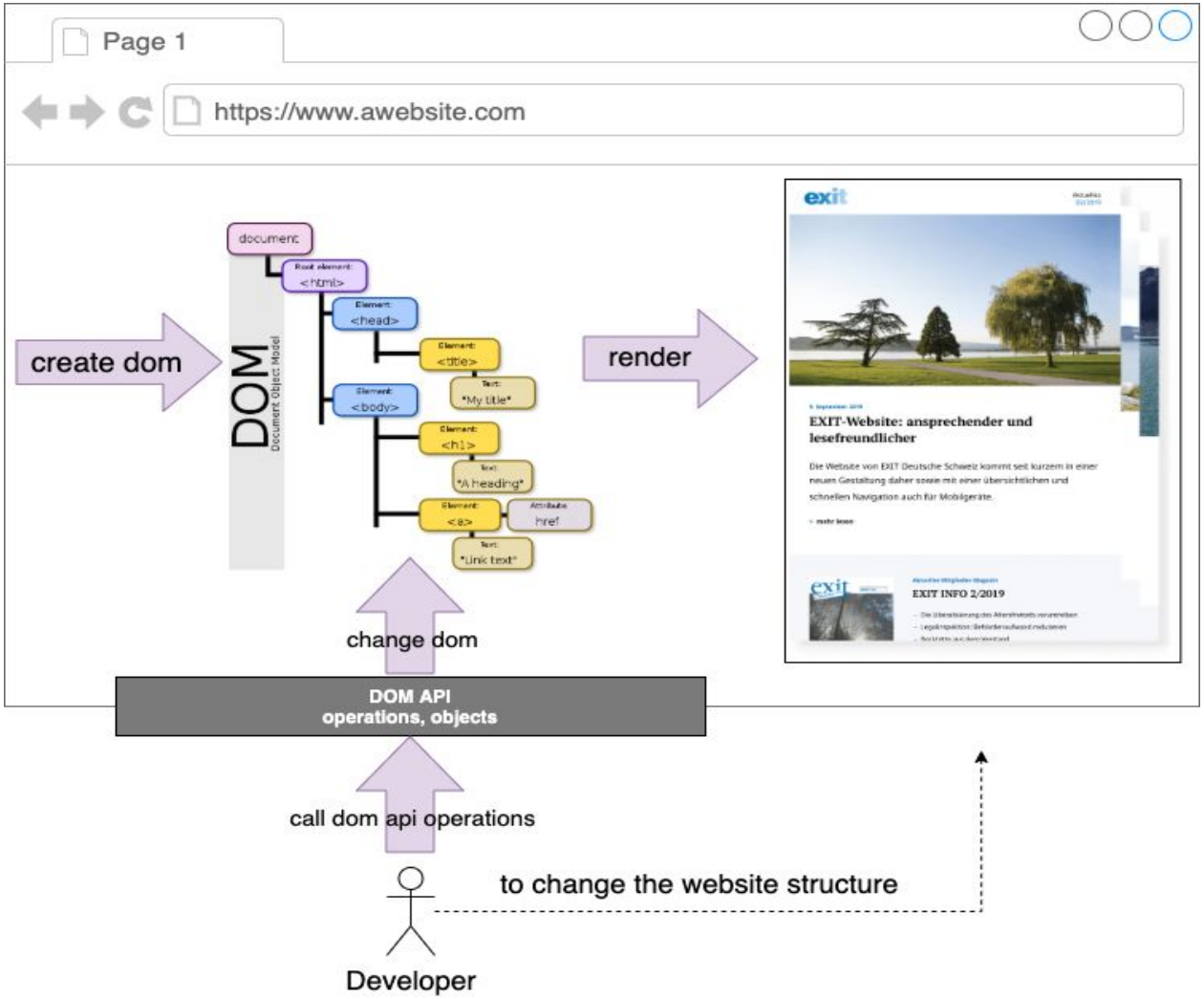
- Is the DOM no more accessible for any changes after rendering process?
- Is there any way to access the dom?
- Is there any benefit to be able to manipulate the dom from outside?

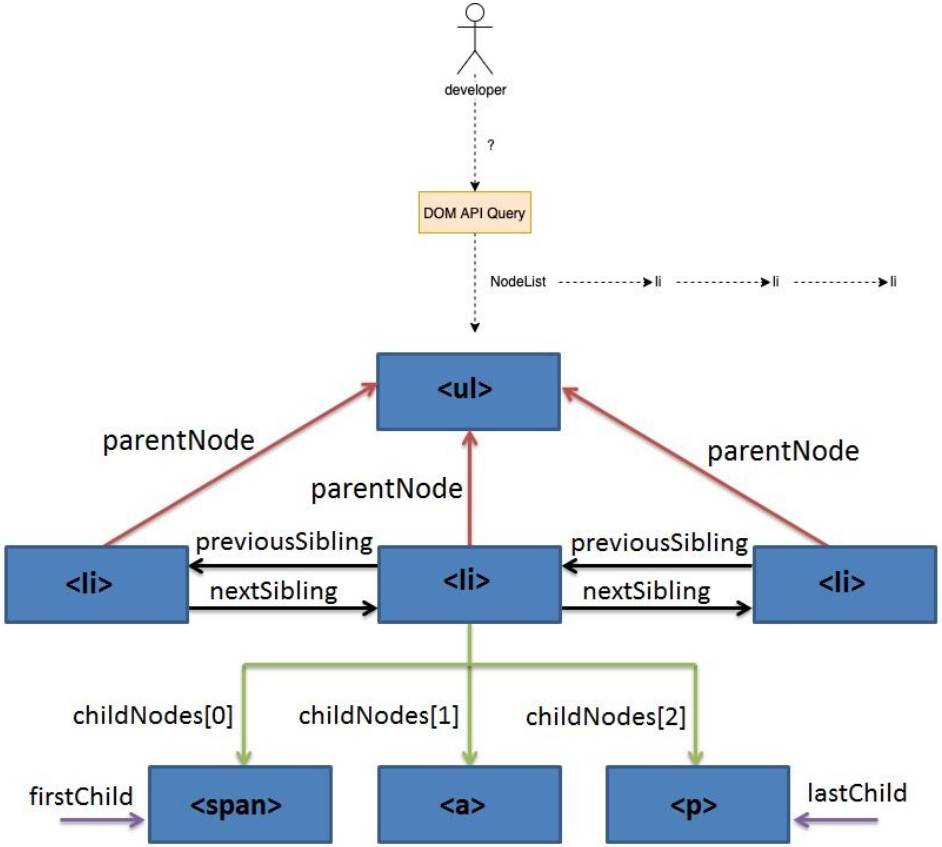
The DOM API is the provided way to access and manipulate the DOM structure form the outside. It can be seen a layer between the DOM and the programmer. It provides answers to programmers queries.

It enables

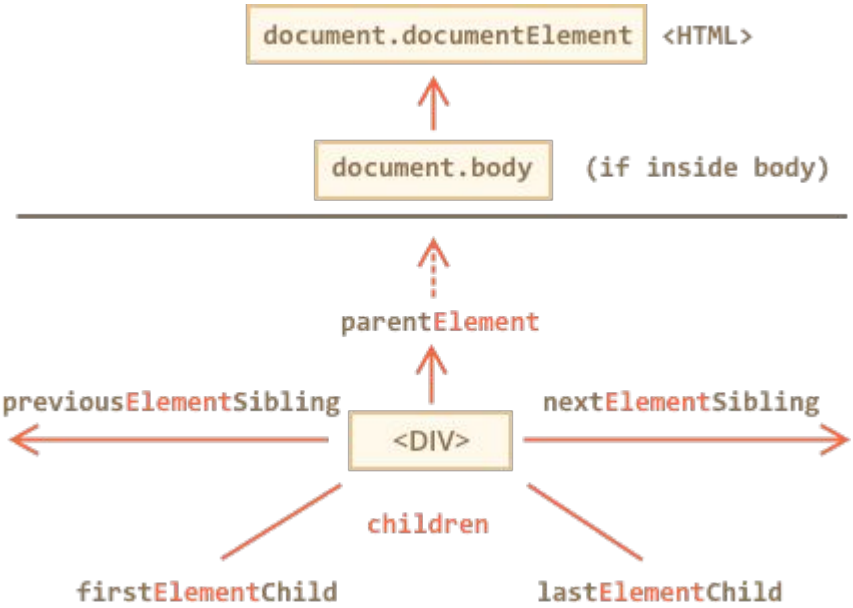
- To query some specific nodes
- To change the attributes
- To change the existing structure
- To listen events (user actions)
- ...


load in browser
request code
https://www.awebsite.com
Web sServer in internet
Page 1
https://www.awebsite.com
create dom
DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute:
href
Text:
"Link text"
render
exit
EXIT-Website: ansprechender und lesefreundlicher
Die Website von EXIT Deutsche Schweiz kommt seit kurzem in einer neuen Gestaltung daher sowie mit einer übersichtlichen und schnellen Navigation auch für Mobilgeräte.


Page 1
https://www.awebsite.com
create dom
DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute
href
Text:
"Link text"
render
exit
EXIT-Website: ansprechender und lesefreundlicher
Die Website von EXIT Deutsche Schweiz kommt seit kurzem in einer neuen Gestaltung daher sowie mit einer übersichtlichen und schnellen Navigation auch für Mobilgeräte.
EXIT INFO 2/2019
change dom
DOM API
operations, objects
call dom api operations
to change the website structure
Developer



Node relationships (source: https://www.qualitestgroup.com/resources/knowledge-center/how-to-guide/traverse-dom/)



Some DOM objects

Categories of the DOM API Methods

Traversing

- **document**.getElementById(id)
- **document**.getElementsByTagName(name)
- **document**.getElementsByClassName(name)
- **document**.querySelector(“CSS Selector”)
- **document**.querySelectorAll(“CSS Selector”)

Manipulating

- **document**.createElement(name)
- parentNode.appendChild(node)
- parentNode.removeChild(node);
- parentNode.replaceChild(newnode, oldnode)

Best known DOM API members

- node.childNodes
- node.firstChild
- node.lastChild
- node.parentNode
- node.nextSibling
- node.previousSibling

- **element.innerHTML**
- element.style.left
- element.setAttribute()
- element.getAttribute()
- **element.addEventListener()**
- **element.removeEventListener()**

Query the DOM

```
let matches = document.querySelectorAll("p");

let matches = document.querySelectorAll("div.note, div.alert");

var container = document.querySelector("#test");

var matches = container.querySelectorAll("div.highlighted > p");

var highlightedItems = userList.querySelectorAll(".highlighted");
highlightedItems.forEach(function(userItem) { deleteUser(userItem); });
```

Query the DOM

```
const fakeImages = document.querySelectorAll(".fake-image");

for (var i = 0; i < fakeImages.length; i++) {
  console.log('fakeImage: ', fakeImages[i]);
}

for (const fakeImage of fakeImages) {
  console.log('fakeImage: ', fakeImage);
}

fakeImages.forEach(fakeImage => {
  console.log('fakeImage: ', fakeImage);
});

Array.from(fakeImages).forEach(fakeImage => {
  console.log('fakeImage: ', fakeImage);
});
```

DOM Events

DOM Events

There happens too many events as the user interacts with the page. Some of them are

- Loading the page itself
- Clicking on an element
- Hovering over an element
- Double Click
- Focusing on an element
- …

This events can be caught through some event handlers (functions). To bind the DOM/BOM events with the handlers (JS Functions), we use the “addEventListner” method.

On Click

```
<p id="Absatz"></p>
<button id="Eingabe">Los!</button>

<script>

 var elem = document.getElementById('Eingabe');
  elem.addEventListener('click', erzeugeZeitStempel);

  function erzeugeZeitStempel() {
    var Zeitstempeltext = document.createTextNode(document.lastModified);
    var TextZuvor = document.createTextNode('Datum des letzten Updates: ');
    document.getElementById('Absatz').appendChild(TextZuvor);
    document.getElementById('Absatz').appendChild(Zeitstempeltext);
  }
</script>
```

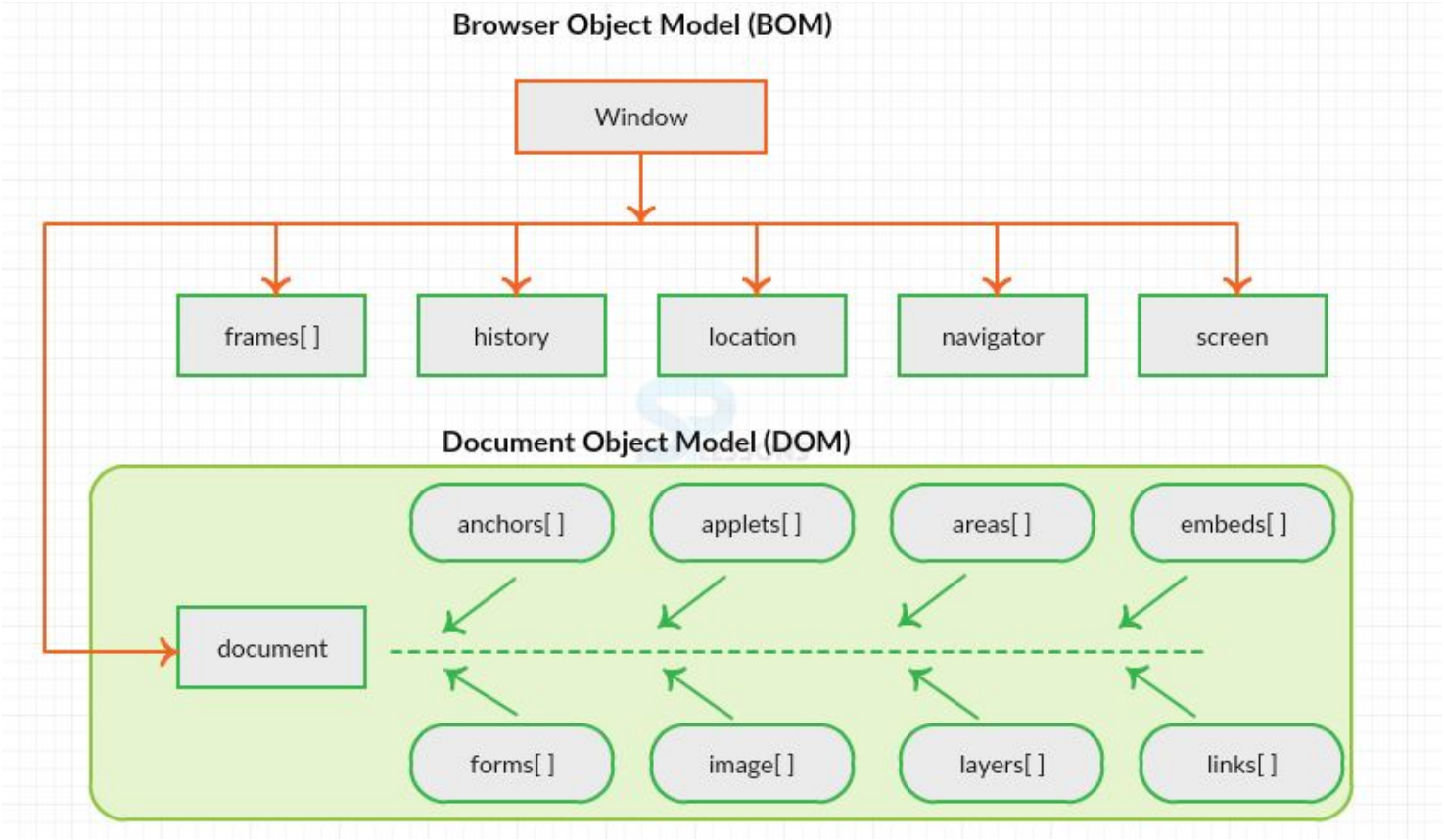
On Mouseover

```
let elem = document.getElementById('Eingabe');
let ausgabe = document.getElementById('Ausgabe');
elem.addEventListener('mouseover', mouseOver);
elem.addEventListener('mouseout', mouseOut);

function mouseOver() {
  ausgabe.innerHTML = 'Ich bin dynamisch!';
  elem.innerHTML = 'Drüber!';
}

function mouseOut() {
  ausgabe.innerHTML = ' ';
   elem.innerHTML = 'Wieder weg!';
}
```

BOM API

What is BOM?

The BOM (Browser Object Model) consists of the objects navigator, history, screen, location and document which are children of window. In the document node is the DOM (Document Object Model), the document object model, which represents the contents of the page. You can manipulate it using javascript.



Relation between DOM and BOM (source: https://www.splessons.com/lesson/javascript-bom/)

Questions?