JavaScript & DOM

Part-II

DOM

- DOM: Document Object Model
- DOM represents a document as a family tree

What is DOM?

- A programming interface for HTML or XML
- DOM represents the document as nodes and objects
  - Nodes and objects can be created or changed or removed
- 3 different parts
  - Core DOM
  - XML DOM
  - HTML DOM

The Levels of DOM

- Level 0: Supports an interface DOM and refers to what existed before the standard
- Level 1: allows Navigation of DOM, content manipulation
- Level 2: Support namespace, filtered views and event
- Level 3: has many specifications:
    - Core
    - Load and Save
    - XPath
    - Views and formatting
    - Requirements
    - Validation

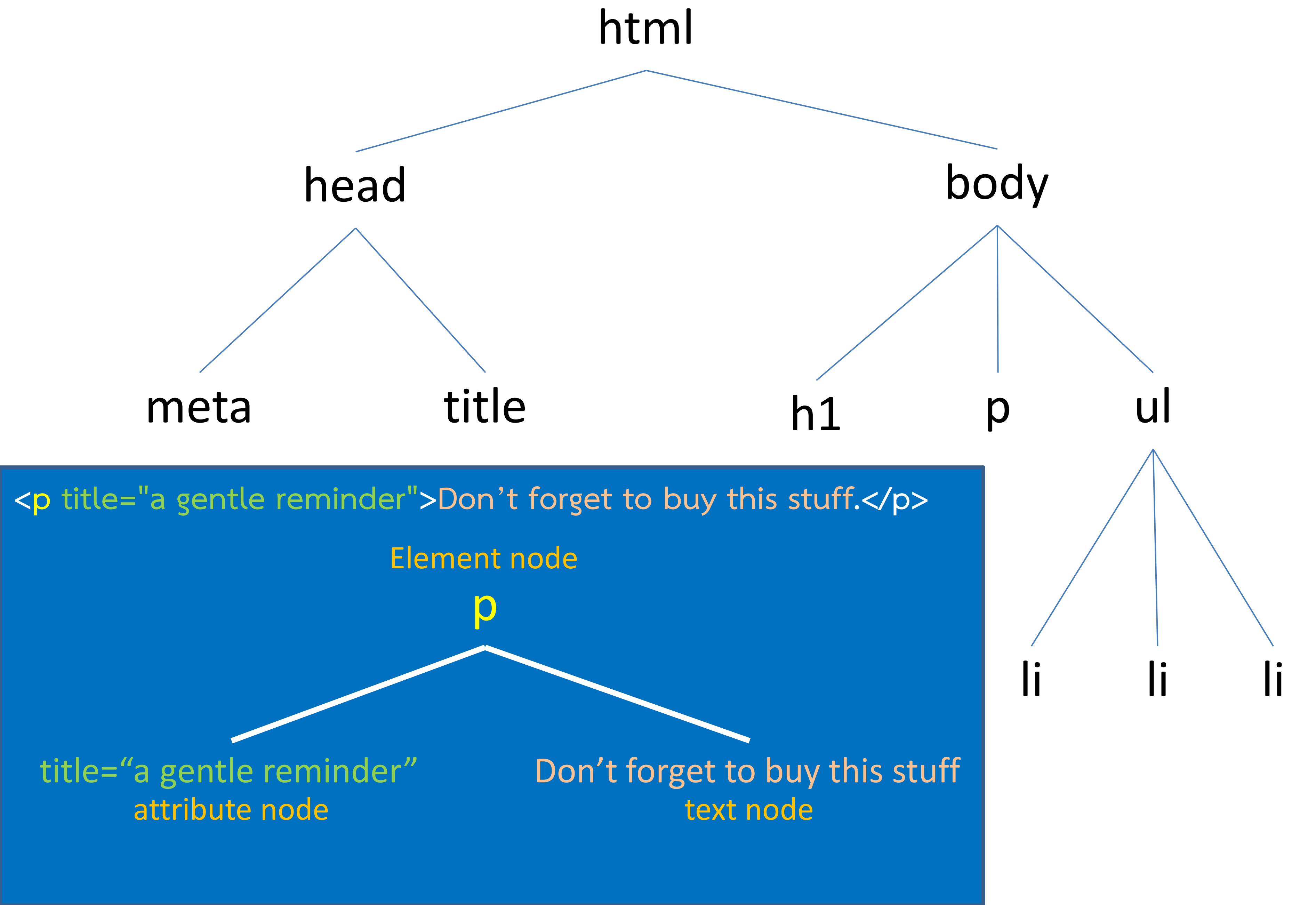
(further reading: https://www.c-sharpcorner.com/UploadFile/79037b/let%E2%80%99s-understand-the-levels-of-dom-document-object-model/)

Fundamental Data Types

- Document object
  - Represent web page
- Node : basic object within document
  - Element : type of node
    - NodeList : Array of elements
  - Attribute : type of node
  - Text
  - Comment
  - etc.

Example

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">
<head>
    <meta http-equiv="content-type"
content="text/html; charset=utf-8" />
    <title>Shopping list</title>
</head>
<body>
    <h1>What to buy</h1>
    <p title="a gentle reminder">Don’t forget to buy
this stuff.</p>
    <ul id="purchases">
      <li>A tin of beans</li>
      <li>Cheese</li>
      <li>Milk</li>
    </ul>
</body>
</html>
```



html

head

body

meta

title

h1

p

ul

li

li

li

```
<p title="a gentle reminder">Don’t forget to buy this stuff.</p>
```

Element node

p

title=“a gentle reminder”
attribute node

Don’t forget to buy this stuff
text node

DOM accessing

- As of now, every web browser uses DOM as a gateway to the document for scripting language (e.g. JavaScript)

Working with Document Object

- Properties of Document Object
  - Image property
  - Form property
  - Etc.
- Method of Document Object
  - Selecting elements
  - Traversing elements
  - Manipulating elements
  - Attributes
  - Manipulating Styles

Selecting elements

- Method of Document Object
  - getElementById
  - getElementByName
  - getElementsByClassName
  - getElementsByTagName
  - querySelector

The getElementById() method

- Allow access to an element by the value of its id attribute

Example

```
<div id="some_text">This is some text.</div>
```

```
var text_element =
          document.getElementById("some_text");
```

The getElementByName() method

- Allow access to an element by the value of its name attribute

Example

```
<input type=“radio” name=“color” value=“violet”>
```

```
var text_element =
     document.getElementByName(“color");
```

**if there are elements with the name specified,
text_element var. will be array filled with matched elements.

The getElementsByClassName() method
The getElementsByTagName() method

- Get an array filled with all the elements in the document that have specified class/tag name

querySelector

- Select element(s) by using CSS selector
  - querySelector() : select only first element that matched
  - querySelectorAll : select all elements that matched
- Example:

```
Let items=document.querySelector('*');
Let items=document.querySelectorAll('*');
Let items=document.querySelector('div');
Let items=document.querySelectorAll('div');
Let items=document.querySelector('div p');
Let items=document.querySelectorAll('div p');
```

Traversing elements

- Get parent node
  - node.parentNode //the returned parent node is read-only
- Get child nodes
  - node.firstChild
  - node.firstElementChild
  - node.lastChild
  - node.lastElementChild
  - node.childNode
  - node.children
- Get sibling
  - node.nextElementSibling
  - node.previousElementSibling

Example

- HTML

```
<div>
     <p id=“firstchild”>test
          <span>Hello</span>
     </p>
     <a href=“www.kmitl.ac.th”>Click</a>
     <img src=“picture.png”></img>
</div>
```

- DOM

```
let n = document.querySelector(‘#firstchild’);
let parent = n.parentNode
let childs = n.children
let next = n.nextElementSibling
```

Manipulating elements

- createElement
- appendChild
- createTextNode
- textContent
- innerHTML
- removedChild

Document node creation

```
<body>
    <div id="div1" title="All about me!">
        This page is about me, me, and... me!
    </div>
</body>
```

```
var me_div = document.getElementById("div1");
var inner_div = document.createElement("div");//<div><div>
var inner_div_text = document.createTextNode("More...");
inner_div.appendChild(inner_div_text);//<div>More…</div>
me_div.appendChild(inner_div);
```

Result

```
<body>
<div id="div1" title="All about me!">
This page is about me, me, and... me!
<div>
More...
</div>
</div>
</body>
```

getAttribute() method

- Get the values of attribute of an element

Example

```
<img id=”i1" src="images/pr.gif” alt="my image" />
```

```
var i_id = document.getElementById(“i1”);
var i_src = i_id.getAttribute(“src”);
```

setAttribute() method

- Set the values of attribute of an element

Example

```
<img id=”i1" src="images/pr.gif” alt="my image" />
```

```
var i_id = document.getElementById(“i1”);
i_id.setAttribute(“src”, “images/xx.jpg”);
```

Manipulating Style

- Style property**
- getComputedStyle
- className
- classList
- width & height

Example

```
let item = getElementById(‘i1’);
item.style.color = ‘blue’;
item.style.background-color = ‘yellow’;
```

Events

- Normally used in combination with functions
- Examples of DOM event:
  - abort: loading of a media is aborted
  - blur: element loses focus
  - change: content of element has changed
  - click: mouse clicks on element
  - focus: element gets focus
  - etc.

(further reading: https://www.w3schools.com/jsref/dom_obj_event.asp)

Example

```
<div>
     <button id=“id01”>Click here</button>
</div>

var d = document.getElementById(“id01”);
d.setAttribute(“onclick”, “()=>{return
false;}
Or.
d.adEventListener(‘click’, ()=>{return
false;)
```

Navigator Object

- Contains information about the browser
- Examples:
  - appName: name of the browser
  - appVersion: version of the browser
  - cookieEnabled: cookies are enabled or not
  - etc.

(further reading: https://www.w3schools.com/jsref/obj_screen.asp
)

Further reading

- reference:

https://www.javascripttutorial.net/javascript-dom/

Js_event_01.js

```
function test(v) {
window.alert(v.getAttribute(“type”));
}
```

```
<body>
<form>

<input type="button" value="Click Me!" onclick=“test(this);"
/>
</form>
<script type="text/javascript" src="js_event_01.js"></script>
</body>
```