Information Technologies for Industrial Engineers

เทคโนโลยีสารสนเทศสำหรับวิศวกรอุตสาหการ

Document Object Model (DOM)

What is the DOM?

- The Document Object Model (DOM) is a programming interface for web documents.
- It represents the page so that programs can change the document structure, style, and content.
- The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

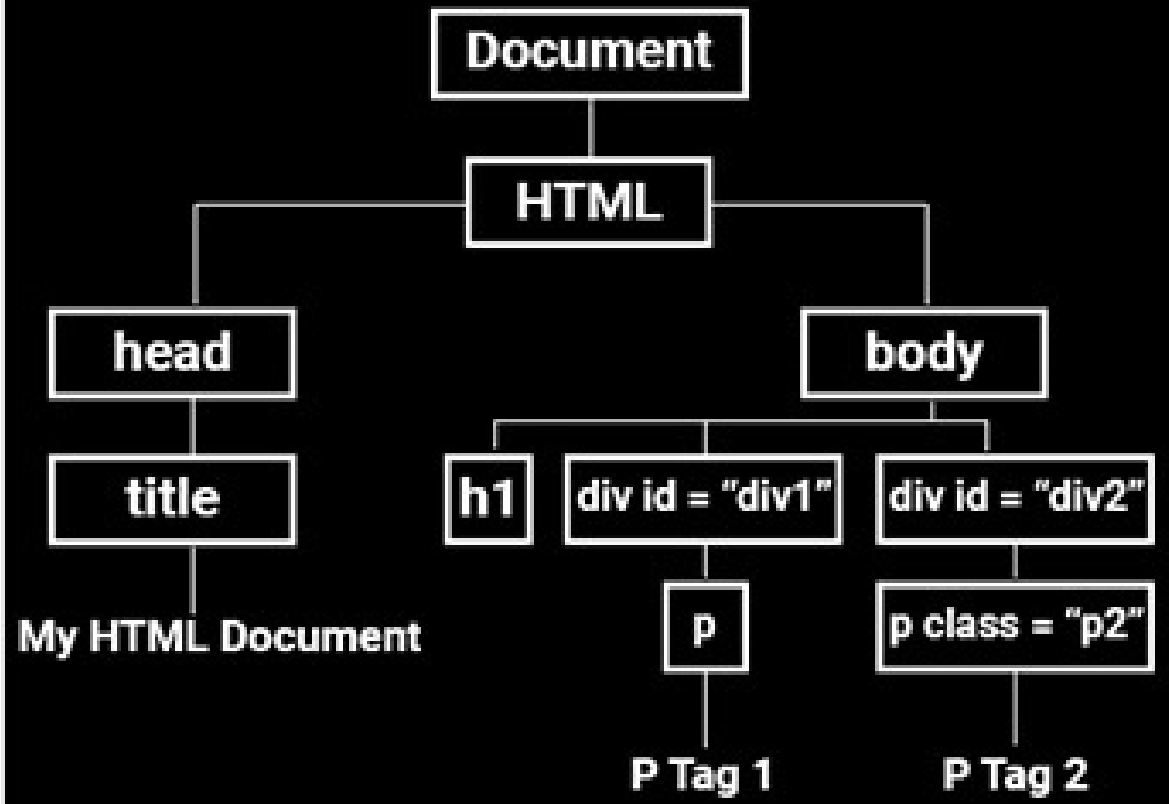
What is Document Object Model ?

HTML Document

index.html

```
1  <html>
2      <head>
3          <title>My HTML Document</title>
4      </head>
5
6      <body>
7          <h1>Heading</h1>
8          <div id="div1">
9              <p>P Tag 1</p>
10         </div>
11         <div id="div2">
12             <p class="p2">P Tag 2</p>
13         </div>
14     </body>
15 </html>
```

Document Object Model (DOM)



Data type

- `Document` : represents any web page loaded in the browser
- `Node` : represents an object located within a document.
  - `Element` : represents an element in HTML.
  - `TextNode`: specifies text in an element.
  - `Attr`: speicifies attributes of an element.
- `NodeList` : A nodeList is an array of elements.

DOM Navigation

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
  </head>
  <body>
    <h1>Headings</h1>
    <div id="div1">
      div1 text
      <p>p text</p>
    </div>
    <div id="div2">div2 text</div>
  </body>
</html>
```

```
// document object
console.log(document.childNodes);

// html
const html = document.childNodes[1];
console.log(html);

// head, body, text nodex
console.log(html.childNodes);
const head = html.childNodes[0];
const body = html.childNodes[2];
console.log(head);
console.log(body);
```

```
// #div1
console.log(body.childNodes);
const div1 = body.childNodes[3];
console.log(div1);

// We can change the text of #div1
div1.childNodes[0].textContent = "Changed";

// Note that this is different from
// div1.textContent = 'Changed';
```

Introduction to events

Event

- Events are *actions* that happen in the system you are programming, which the system tells you about.
  - So your code can react to them.
- For example, if the user clicks a button on a webpage, you might want to react to that action by displaying an information box.

Event type

- The user selects a certain element or hovers the cursor over a certain element.
- The user chooses a key on the keyboard.
- The user resizes or closes the browser window.
- A web page finishes loading.
- A form is submitted.
- A video is played, paused, or finishes.
- An error occurs.

Event type

- Event reference
- `Element` -> `click event`

Event handler

- To react to an event, you attach an event handler to it.
- This is a block of code that runs when the event fires.
- Event handlers are sometimes called *event listeners*.

Example

```
<button>Click Me</button>
```

Example

```
const btn = document.querySelector("button");

function random(number) {
  return Math.floor(Math.random() * (number + 1));
}

function clickHandler() {
  const rndCol = `rgb(${random(255)}, ${random(255)}, ${random(255)})`;
  document.body.style.backgroundColor = rndCol;
}

btn.addEventListener("click", clickHandler);
```

Adding multiple listeners

(Add to the above code)

```
function doubleClickHandler() {
  alert("Reset to white");
  document.body.style.backgroundColor = "white";
}

btn.addEventListener("dblclick", doubleClickHandler);
```

Remove listeners

```
btn.removeEventListener("dblclick", doubleClickHandler);
```

Event object

- Sometimes, inside an event handler function, you'll see a parameter specified with a name such as `event`, `evt`, or `e`.
- This is called the **event object**, and it is automatically passed to event handlers to provide extra features and information.

Example

HTML

```
<div style="width: 16rem; height: 16rem; border: 1px solid gray"></div>
```

Example

JavaScript

```
const div = document.querySelector("div");

div.addEventListener("mousemove", onMouseMove);

function onMouseMove(e) {
  div.innerText = `${e.offsetX}, ${e.offsetY}`;
  div.style.backgroundColor = `rgb(${e.offsetX}, ${e.offsetY}, ${
    (e.offsetX + e.offsetY) / 2
  })`;
}
```

Event target

`event.target`

- The element that caused the event.
- Useful when you want to reuse the event handler.

```
<head>
  <style>
    body {
      display: flex;
      flex-wrap: wrap;
      gap: 1rem;
    }
    div {
      width: 8rem;
      height: 8rem;
      border: 1px solid gray;
      display: flex;
      justify-content: center;
      align-items: center;
    }
  </style>
</head>
```

```
<body>
  <div id="div1">Box1</div>
  <div id="div2">Box2</div>
  <div id="div3">Box3</div>
  <div id="div4">Box4</div>
  <div id="div5">Box5</div>
  <div id="div6">Box6</div>
</body>
```

```
const divs = document.querySelectorAll("div");

divs.forEach((div) => {
  div.addEventListener("mousemove", handler);
});

function handler(e) {
  e.target.innerText = `${e.offsetX}, ${e.offsetY}`;
  e.target.style.color = "white";
  e.target.style.backgroundColor = `rgb(${e.offsetX}, ${e.offsetY}, ${
    (e.offsetX + e.offsetY) / 2
  })`;
}
```

Form

index.html

```
<h1>Enter Your Name</h1>
<div>
  <label for="id-fname">First name:</label>
  <input type="text" id="id-fname" name="fname" />
</div>
```

script.js

```
const fname = document.getElementById("id-fname");
const h1 = document.querySelector("h1");

fname.addEventListener("input", (e) => {
  const text = e.target.value;
  if (text) {
    h1.textContent = e.target.value;
  } else {
    h1.textContent = "Enter Your Name";
  }
});
```

Super Secret

`index.html`

```
<h1>My Super Secret</h1>
<label for="id-pass">Password:</label>
<input type="password" id="id-pass" name="pass" />
```

script.js

```
const pass = document.getElementById("id-pass");
const h1 = document.querySelector("h1");

pass.addEventListener("input", (e) => {
  const text = e.target.value;
  if (text === "1234") {
    h1.textContent = "You know my password, great job.";
  } else if (text === "") {
    h1.textContent = "My Super Secret";
  } else {
    h1.textContent = "Wrong Password";
  }
});
```