

Тема 1. Основи JavaScript

Тема: DOM

*Морозов Андрій Васильович,
к.т.н, доц.,
декан факультету інформаційно-комп'ютерних
технологій ЖДТУ*

2. Навігація по DOM-структурі

- Отримання DOM-об'єкта за ідентифікатором:
`document.getElementById("ідентифікатор")`
- Отримання DOM-елементів, що мають заданий атрибут name:
`document.getElementsByName("ім'я")`
- Пошук DOM-об'єктів за тегом:
`elem.getElementsByTagName("тег")`
- Пошук DOM-елементів за класом:
`elem.getElementsByClassName("клас")`
- Пошук DOM-елементів за CSS-селектором (усі, що знайдені):
`elem.querySelectorAll("CSS-селектор")`
- Пошук DOM-елементу за CSS-селектором (першого):
`elem.querySelector("CSS-селектор")`

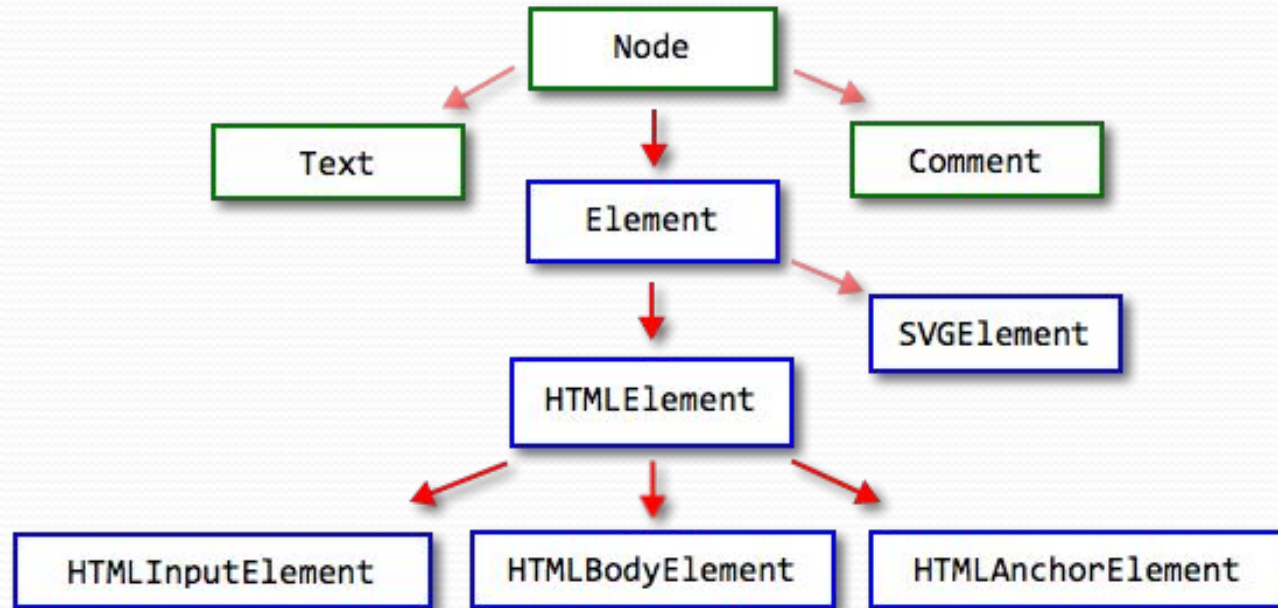
Повертаються DOM-елементи за виключенням текстових

тег <html>	<code>document.documentElement</code>
тег <body>	<code>document.body</code>
колекція нащадків	<code>elem.children</code>
батьківський елемент	<code>elem.parentNode</code>
перший нащадок	<code>elem.firstChild</code>
останній нащадок	<code>elem.lastElementChild</code>
правий сусід	<code>elem.nextElementSibling</code>
лівий сусід	<code>elem.previousElementSibling</code>

Повертаються усі DOM-елементи, включаючи текстові

колекція нащадків	<code>elem.childNodes</code>
перший нащадок	<code>elem.firstChild</code>
останній нащадок	<code>elem.lastChild</code>
правий сусід	<code>elem.nextSibling</code>
лівий сусід	<code>elem.previousSibling</code>

DOM-вузли є об'єктами різних класів:



```
console.log(document.body instanceof HTMLBodyElement); // true
console.log(document.body instanceof HTMLElement); // true
console.log(document.body instanceof Element); // true
console.log(document.body instanceof Node); // true
```

У полі `elem.nodeType` кожного DOM-вузла зберігається тип вузла, який може мати одне із значень:

Значення	Константа	Пояснення
1	<code>ELEMENT_NODE</code>	Тег
2	<code>ATTRIBUTE_NODE</code>	Атрибут тега
3	<code>TEXT_NODE</code>	Текстовий вузол
8	<code>COMMENT_NODE</code>	Вузол, що відповідає за коментар
9	<code>DOCUMENT_NODE</code>	для об'єкту <code>document</code>

```
var childNodes = elem.childNodes;
for (var i = 0; i < childNodes.length; i++) {
    // знайти DOM-елементи, які відповідаються тегам
    if (childNodes[i].nodeType !== Node.ELEMENT_NODE)
    {
        ...
    }
}
```

Властивості DOM-вузлів (можна лише зчитувати):

- `tagName` – є тільки у тегів
(`elem. nodeType == Node.ELEMENT_NODE`)
- `nodeName` – для тегів містить `tagName`, а для усіх інших елементів містить рядок з типом вузла;
- `textContent` – вміст тегу у текстовому вигляді
(конкатенація усіх текстових вузлів)

Властивості DOM-вузлів (можна зчитувати і записувати):

- `innerHTML` – містить вміст тегу (лише для тегів - рядок з HTML-кодом);
- `data` – вміст DOM-елементів (для усіх DOM-елементів окрім тегів, застосовують для, наприклад текстового вузла, або коментаря)

Отримання класів тегу:

- `className` – значення атрибуту `class` (рядок);
- `classList` – класи у вигляді колекції;
 - `elem.classList.contains("class")` – повертає `true/false`, в залежності від того, чи є у елемента клас `class`.
 - `elem.classList.add/remove("class")` – додає/видаляє клас `class`
 - `elem.classList.toggle("class")` – якщо класу `class` немає, додає його, якщо є – видаляє.

Методи для роботи з атрибутами тегу:

<code>elem.hasAttribute(name)</code>	перевіряє наявність атрибуту
<code>elem.getAttribute(name)</code>	повертає значення атрибуту
<code>elem.setAttribute(name, value)</code>	встановлює значення атрибуту
<code>elem.removeAttribute(name)</code>	видаляє атрибут