자바스크립트선택자

자바스크립트에서 DOM 요소를 선택하는 방법에는 여러 가지가 있습니다. 선택자 메서드를 사용하여 HTML 문서 내의 요소를 접근하고, 다양한 방식으로 조작할 수 있습니다. 여기서 주요 선택자와 그 예제를 소개하겠습니다.

1. getElementById()

특정 ID를 가진 요소를 선택합니다.

한 개의 요소만 선택 가능하며, 해당 ID는 문서 내에서 유일해야 합니다.

예제:

html

코드 복사

```
<div id="myDiv">Hello, World!</div>


<script>
    const element = document.getElementById("myDiv");
    console.log(element.textContent); // "Hello, World!" 출력
</script>
```

2. getElementsByClassName()

특정 클래스를 가진 모든 요소를 선택합니다.

HTMLCollection이라는 유사 배열을 반환합니다.

예제:

html

```
<div class="myClass">First Div</div>
<div class="myClass">Second Div</div>


<script>
```

```
    const elements = document.getElementsByClassName("myClass");
    console.log(elements[0].textContent); // "First Div" 출력
    console.log(elements[1].textContent); // "Second Div" 출력
</script>
```

3. getElementsByTagName()

특정 태그를 가진 모든 요소를 선택합니다.

HTMLCollection을 반환합니다.

예제:

html

```
<p>Paragraph 1</p>
<p>Paragraph 2</p>

<script>
    const paragraphs = document.getElementsByTagName("p");
    console.log(paragraphs.length); // 2 출력
</script>
```

4. querySelector()

CSS 선택자를 사용하여 첫 번째로 일치하는 요소 하나를 선택합니다.

클래스, ID, 태그 등을 조합한 CSS 선택자를 사용할 수 있습니다.

예제:

html

```
<div class="myDiv">First Div</div>
<div class="myDiv">Second Div</div>

<script>
```

```
    const element = document.querySelector(".myDiv"); // 첫 번째 myDiv 요소 선택
    console.log(element.textContent); // "First Div" 출력
</script>
```

5. querySelectorAll()

CSS 선택자를 사용하여 일치하는 모든 요소를 선택합니다.

NodeList라는 유사 배열을 반환하며, 이를 사용해 각 요소에 접근할 수 있습니다.

예제:

html

```
<div class="myDiv">First Div</div>
<div class="myDiv">Second Div</div>

<script>
    const elements = document.querySelectorAll(".myDiv"); // 모든 .myDiv 요소 선택
    elements.forEach(function(el) {
        console.log(el.textContent);
    });
    // "First Div"와 "Second Div"가 차례로 출력됨
</script>
```

6. getAttribute() 및 setAttribute()

선택한 요소의 속성 값을 읽거나 설정합니다.

예제 (속성 가져오기):

html

```
<img id="myImage" src="image.jpg" alt="Sample Image">
<script>
    const img = document.getElementById("myImage");
```

```
    const srcValue = img.getAttribute("src");
    console.log(srcValue); // "image.jpg" 출력
</script>
```

예제 (속성 설정):

```
html
<img id="myImage" src="image.jpg" alt="Sample Image">

<script>
    const img = document.getElementById("myImage");
    img.setAttribute("src", "newImage.jpg");
    console.log(img.src); // "newImage.jpg" 출력
</script>
```

7. parentNode, children, nextElementSibling, previousElementSibling

요소 간의 관계를 선택할 때 사용합니다.

예제 (부모 요소 선택):

```
html
코드 복사
<div id="parent">
    <p id="child">Hello!</p>
</div>

<script>
    const child = document.getElementById("child");
    const parent = child.parentNode;
    console.log(parent.id); // "parent" 출력
```

</script>

예제 (자식 요소 선택):

html

<ul id="myList">

    Item 1

    Item 2

    Item 3

<script>

    const list = document.getElementById("myList");

    const children = list.children;

    console.log(children[0].textContent); // "Item 1" 출력

</script>

8. closest()

선택한 요소에서부터 가장 가까운 상위 요소를 찾습니다. 주어진 CSS 선택자에 일치하는 요소를 찾아냅니다.

예제:

html

    Some text

<script>

    const para = document.getElementById("myPara");

```
    const closestDiv = para.closest(".container");

    console.log(closestDiv.className); // "container" 출력
</script>
```

정리

자바스크립트의 DOM 선택자를 이용해 페이지 요소를 효율적으로 선택하고 조작할 수 있습니다. **getElementById**는 ID로 선택할 때, querySelector 및 **querySelectorAll**은 복잡한 CSS 선택자를 사용할 수 있어 매우 강력합니다. 이러한 선택자 메서드를 상황에 맞게 사용하여 HTML 문서의 요소를 동적으로 처리할 수 있습니다.