**기존 요소의 외부에 새로운 콘텐츠 추가 메소드**

|  |  |
| --- | --- |
| **메소드** | **설명** |
| **.before()** | **선택한 요소의 바로 앞에 새로운 요소나 콘텐츠를 추가함.** |
| **.after()** | **선택한 요소의 바로 뒤에 새로운 요소나 콘텐츠를 추가함.** |

**기존 요소의 내부에 새로운 콘텐츠 추가 메소드**

|  |  |
| --- | --- |
| **메소드** | **설명** |
| **.append()** | **선택한 요소의 마지막에 새로운 요소나 콘텐츠를 추가함.** |
| **.prepend()** | **선택한 요소의 처음에 새로운 요소나 콘텐츠를 추가함.** |
| **.appendTo()** | **선택한 요소를 해당 요소의 마지막에 삽입함.** |
| **.prependTo()** | **선택한 요소를 해당 요소의 처음에 삽입함.** |
| **.html()** | **해당 요소의 HTML 콘텐츠를 반환하거나 설정함.** |
| **.text()** | **해당 요소의 텍스트 콘텐츠를 반환하거나 설정함.** |

**<input>요소 선택자**

|  |  |
| --- | --- |
| **선택자** | **설명** |
| **:button** | **type 속성값이 "button"인 요소를 모두 선택함.** |
| **:checkbox** | **type 속성값이 "checkbox"인 요소를 모두 선택함.** |
| **:file** | **type 속성값이 "file"인 요소를 모두 선택함.** |
| **:image** | **type 속성값이 "image"인 요소를 모두 선택함.** |
| **:password** | **type 속성값이 "password"인 요소를 모두 선택함.** |
| **:radio** | **type 속성값이 "radio"인 요소를 모두 선택함.** |
| **:reset** | **type 속성값이 "reset"인 요소를 모두 선택함.** |
| **:submit** | **type 속성값이 "submit"인 요소를 모두 선택함.** |
| **:text** | **type 속성값이 "text"인 요소를 모두 선택함.** |
| **:input** | **<input>, <textarea>, <select>, <button>요소를 모두 선택함.** |
| **:checked** | **type 속성값이 "checkbox" 또는 "radio"인 요소 중에서 체크되어 있는 요소를 모두 선택함.** |
| **:selected** | **<option>요소 중에서 선택된 요소를 모두 선택함.** |
| **:focus** | **현재 포커스가 가지고 있는 요소를 선택함.** |
| **:disabled** | **비활성화되어있는 요소를 모두 선택함.** |
| **:enabled** | **활성화되어있는 요소를 모두 선택함.** |

**필터링 선택자**

|  |  |
| --- | --- |
| **선택자** | **설명** |
| :eq(n) | 선택한 요소 중에서 인덱스가 n인 요소를 선택함. |
| :gt(n) | 선택한 요소 중에서 인덱스가 n보다 큰 요소를 모두 선택함. |
| :lt(n) | 선택한 요소 중에서 인덱스가 n보다 작은 요소를 모두 선택함. |
| :even | 선택한 요소 중에서 인덱스가 짝수인 요소를 모두 선택함. |
| :odd | 선택한 요소 중에서 인덱스가 홀수인 요소를 모두 선택함. |
| :first | 선택한 요소 중에서 첫 번째 요소를 선택함. |
| :last | 선택한 요소 중에서 마지막 요소를 선택함. |
| :animated | 선택한 요소 중에서 애니메이션 효과가 실행 중인 요소를 모두 선택함. |
| :header | 선택한 요소 중에서 h1부터 h6까지의 요소를 모두 선택함. |
| :lang(언어) | 선택한 요소 중에서 지정한 언어의 요소를 모두 선택함. |
| :not(선택자) | 선택한 요소 중에서 지정한 선택자와 일치하지 않는 요소를 모두 선택함. |
| :root | 선택한 요소 중에서 최상위 루트 요소를 선택함. |
| :target | 선택한 요소 중에서 웹 페이지 URI의 fragment 식별자와 일치하는 요소를 모두 선택함. |
| :contains(텍스트) | 선택한 요소 중에서 지정한 텍스트를 포함하는 요소를 모두 선택함. |
| :has(선택자) | 선택한 요소 중에서 지정한 선택자와 일치하는 자손 요소를 갖는 요소를 모두 선택함. |
| :empty | 선택한 요소 중에서 자식 요소를 가지고 있지 않은 요소를 모두 선택함. |
| :parent | 선택한 요소 중에서 자식 요소를 가지고 있는 요소를 모두 선택함. |

**제이쿼리(jQuery)란?**

제이쿼리(jQuery)는 오픈 소스 기반의 자바스크립트 라이브러리입니다.

제이쿼리는 여러분의 웹 사이트에 자바스크립트를 더욱 손쉽게 활용할 수 있게 해줍니다.

또한, 제이쿼리를 사용하면 짧고 단순한 코드로도 웹 페이지에 다양한 효과나 연출을 적용할 수 있습니다.

이러한 제이쿼리는 오늘날 가장 인기 있는 자바스크립트 라이브러리 중 하나입니다.

**제이쿼리의 역사**

제이쿼리는 2006년 미국의 존 레식(John Resig)이 뉴욕시 바캠프(Barcamp)에서 처음으로 소개하였습니다.

현재는 jQuery Team이라는 개발자 그룹이 jQuery Foundation을 통해 개발과 유지 보수를 담당하고 있습니다.

**제이쿼리의 장점**

제이쿼리가 많이 사용되는 이유는 다음과 같습니다.

1. 제이쿼리는 주요 웹 브라우저의 구버전을 포함한 대부분의 브라우저에서 지원됩니다.

2. HTML DOM을 손쉽게 조작할 수 있으며, CSS 스타일도 간단히 적용할 수 있습니다.

3. 애니메이션 효과나 대화형 처리를 간단하게 적용해 줍니다.

4. 같은 동작을 하는 프로그램을 더욱 짧은 코드로 구현할 수 있습니다.

5. 다양한 플러그인과 참고할 수 있는 문서가 많이 존재합니다.

6. 오픈 라이선스를 적용하여 누구나 자유롭게 사용할 수 있습니다.

[**Chart.js**](https://www.chartjs.org/docs/latest/) ⇒ 무료 자바스크립트 차트 라이브러리

[AIoT 빅데이터 분석 산업솔루션 개발 부트캠프 2 커리큘럼](https://docs.google.com/spreadsheets/d/1RReZGc9Po22A3un3qB_6G69zc7iKR-kzkE5UFVdN2Ro/edit?usp=sharing)

**HTML 태그 속성 접근**

1. 일반 속성:

모든 HTML 요소에는 다양한 속성이 있으며, 이를 JavaScript를 통해 접근할 수 있습니다.

|  |
| --- |
| <a id="myLink" href="https://example.com" target="\_blank">링크</a> |

|  |
| --- |
| const link = document.getElementById('myLink'); console.log(link.href); // '[https://example.com](https://example.com/)' console.log(link.target); // '\_blank' |

2. 입력 요소:

|  |
| --- |
| <input id="myInput" type="text" value="기본값" placeholder="입력하세요"> |

|  |
| --- |
| const input = document.getElementById('myInput'); console.log(input.value); // '기본값' console.log(input.placeholder); // '입력하세요' |

3. 스타일 속성:

CSS 스타일 속성도 JavaScript를 통해 접근할 수 있습니다.

|  |
| --- |
| <div id="myDiv" style="color: red; width: 200px;">안녕하세요</div> |

|  |
| --- |
| const div = document.getElementById('myDiv'); console.log(div.style.color); // 'red' div.style.color = 'blue'; // 색상 변경 |

**addEventListener 메서드의 이벤트 타입**

|  |
| --- |
| 마우스 이벤트   * click: 요소를 클릭할 때 발생 * dblclick: 요소를 더블 클릭할 때 발생 * mouseover: 마우스가 요소 위로 올라갈 때 발생 * mouseout: 마우스가 요소에서 나갈 때 발생   키보드 이벤트   * keydown: 키를 누를 때 발생 * keyup: 키를 뗄 때 발생   폼 이벤트   * submit: 폼이 제출될 때 발생 * change: 폼 요소의 값이 변경될 때 발생 * input: 입력 값이 변경될 때 발생   문서/윈도우 이벤트   * load: 페이지나 이미지가 로드될 때 발생 * resize: 창 크기가 변경될 때 발생 * scroll: 사용자가 스크롤할 때 발생 |

**웹 요소 접근**

|  |
| --- |
| * innerText * 화면에 표시되는 텍스트만 포함 * 스타일 적용된 텍스트만 반환 * textContent * 모든 텍스트를 포함 * HTML 태그는 제외되며, 모든 텍스트 반환 * innerHTML * 요소의 HTML 내용을 포함 * HTML 태그와 내용을 모두 반환하며 HTML 구조 유지 |

**querySelector와 querySelectorAll**

DOM에서 요소를 선택하는 데 사용되는 메서드로, CSS 선택자를 사용하여 HTML 요소를 쉽게 찾을 수 있게 해준다.

|  |
| --- |
| **querySelector**  주어진 CSS 선택자와 일치하는 첫 번째 요소를 반환합니다.  const element = document.querySelector('.class-name');   * 선택된 요소가 없으면 null을 반환합니다. * CSS 선택자를 사용할 수 있어 매우 유연합니다. * 단일 요소만 반환하므로, 여러 요소를 선택할 필요가 있을 때는 다른 방법을 사용해야 합니다.   **querySelectorAll**  주어진 CSS 선택자와 일치하는 모든 요소를 반환합니다.  const elements = document.querySelectorAll('.class-name');   * 반환값은 NodeList입니다. * NodeList는 배열과 유사한 객체로, 여러 요소를 포함합니다. * 선택된 요소가 없으면 빈 NodeList를 반환합니다. * NodeList는 forEach 메서드를 사용할 수 있어 반복 처리에 용이합니다. |

**논리 OR 연산자를 이용한 표현식**

|  |
| --- |
| let b = 0; // falsy let c = 5;  let a = b || c; // a는 5가 됨 console.log(a); // 5  b = 10; // truthy a = b || c; // a는 10이 됨 console.log(a); // 10 |

**classList.add와 className 의 차이**

|  |
| --- |
| **classList.add('highlight'):** highlight 클래스를 현재 클래스 목록에 추가합니다. 이미 highlight 클래스가 존재하는 경우 중복하여 추가하지 않습니다. 여러 클래스를 추가할 수 있습니다   * myDiv.classList.add('class1', 'class2');   **className = "highlight":** className 속성에 "highlight" 문자열을 직접 할당합니다. 이 경우, 기존의 모든 클래스명이 제거되고 highlight 클래스만 남습니다. 즉, 요소가 가지고 있던 모든 기존 클래스를 잃게 됩니다. |

**classList 메서드**

|  |
| --- |
| **add(className)**: 기능: highlight 클래스를 myDiv에 추가합니다. 버튼 클릭 시 해당 클래스가 추가되고 알림이 표시됩니다.  **remove(className)**: 기능: highlight 클래스를 myDiv에서 제거합니다. 버튼 클릭 시 해당 클래스가 제거되고 알림이 표시됩니다.  **toggle(className)**: 기능: hidden 클래스를 myDiv에 추가하거나 제거합니다. 버튼 클릭 시 클래스가 토글되고 그에 대한 알림이 표시됩니다.  **contains(className)**: 기능: myDiv에 highlight 클래스가 있는지 확인합니다. 버튼 클릭 시 클래스가 존재하는지 여부에 따라 알림이 표시됩니다. |

**localStorage**

|  |
| --- |
| localStorage는 웹 스토리지 API의 일부로, 웹 애플리케이션이 브라우저에 데이터를 저장할 수 있도록 해줍니다. 이 데이터는 브라우저를 닫거나 새로 고침해도 유지됩니다.   * 데이터 저장 * localStorage.setItem(key, value) * 데이터 조회 * localStorage.getItem(key) * 데이터 삭제 * localStorage.removeItem(key) * 모든 데이터 삭제 * localStorage.clear() |

**Location 객체**

|  |
| --- |
| 1. location.href 현재 페이지의 URL을 가져오거나 새로운 URL로 변경하는 데 사용됩니다.  - 읽기: 현재 페이지의 URL을 반환합니다. - 쓰기: 다른 URL로 페이지를 이동합니다.  2. location.reload() 현재 페이지를 새로 고침합니다.  기본적으로 캐시된 페이지를 새로 고침하지만, 매개변수로 true를 전달하면 서버에서 새로운 페이지를 요청합니다.  3. location.replace()  현재 페이지를 새로운 URL로 변경하지만, 브라우저의 세션 기록에 현재 페이지를 남기지 않습니다. 즉, 사용자가 "뒤로" 버튼을 눌러도 이전 페이지로 돌아갈 수 없습니다. |

**window 객체의 주요 속성과 메서드**

|  |  |  |
| --- | --- | --- |
| 구분 | | 설명 |
| 속성 | innerWidth | 웹 브라우저 화면의 너비를 px(픽셀) 단위로 나타냅니다. |
| innerHeight | 웹 브라우저 화면의 높이를 px 단위로 나타냅니다. |
| outerWidth | 웹 브라우저 창의 너비를 px 단위로 나타냅니다. |
| outerHeight | 웹 브라우저 창의 높이를 px 단위로 나타냅니다. |
| screenY | 웹 브라우저 위쪽 면과 모니터의 간격을 px 단위로 나타냅니다. |
| screenX | 웹 브라우저 왼쪽 면과 모니터의 간격을 px 단위로 나타냅니다. |
| scrollX | 웹 브라우저의 수평 스크롤 위치를 px 단위로 나타냅니다. |
| scrollY | 웹 브라우저의 수직 스크롤 위치를 px 단위로 나타냅니다. |
| 메서드 | alert() | 알림창을 표시합니다. |
| confirm() | 확인창을 표시합니다. |
| prompt() | 입력창을 표시합니다. |
| open() | 새로운 웹 브라우저 창을 엽니다. |
| close() | 웹 브라우저 창을 닫습니다. |
| setTimeout() | 일정 시간(ms) 뒤에 콜백 함수를 한 번만 실행합니다. |
| setInterval() | 일정 시간(ms)마다 콜백 함수를 반복적으로 실행합니다. |
| clearInterval | setInterval() 메서드로 반복 실행되는 함수를 중지합니다. |
| scrollTo() | 웹 브라우저의 스크롤을 특정 위치만큼 이동합니다. |
| scrollBy() | 웹 브라우저의 스크롤을 현재 위치에서 상대적 위치로 이동합니다. |

**브라우저 객체 모델의 종류**

|  |  |
| --- | --- |
| 종류 | 설명 |
| window | 웹 브라우저가 열릴 때마다 생성되는 최상위 관리 객체 |
| document | 웹 브라우저에 표시되는 HTML 문서 정보가 포함된 객체 |
| location | 웹 브라우저에 현재 표시된 페이지에 대한 URL 정보가 포함된 객체 |
| history | 웹 브라우저에 저장된 방문 기록이 포함된 객체 |
| navigator | 웹 브라우저 정보가 포함된 객체 |
| screen | 웹 브라우저의 화면 정보가 포함된 객체 |

**인기언어 순위 (참고)**



**참여기업**

[프로젝트 참여 기업정보](https://docs.google.com/document/d/11dJCU9lYRNIv8eAT_WSRAPg15yHSS5IsdrJ2XleLRy8/edit?usp=sharing) (2024.10 기준)

**Math 객체의 주요 메서드**

|  |  |
| --- | --- |
| 종류 | 설명 |
| Math.floor() | 주어진 숫자와 같거나 작은 정수 중에서 가장 큰 수를 반환합니다(내림). |
| Math.ceil() | 주어진 숫자와 같거나 큰 정수 중에서 가장 작은 수를 반환합니다(올림). |
| Math.round() | 주어진 숫자를 반올림한 수와 가장 가까운 정수를 반환합니다(반올림). |
| Math.random() | 0 이상 1 미만의 난수를 반환합니다. |

**Date 객체의 메서드 (**new Date()**)**

|  |  |
| --- | --- |
| 종류 | 설명 |
| getFullYear()  setFullYear() | 연도를 4자리 숫자로 표시합니다. |
| getMonth()  setMonth() | 월을 0부터 11까지의 숫자로 표시합니다(1월 → 0, 12월 → 11). |
| getDate()  setDate() | 일을 1부터 31까지의 숫자로 표시합니다. |
| getDay() | 요일을 0부터 6까지의 숫자로 표시합니다(일요일 → 0, 토요일 → 6). |
| getTime()/setTime() | 1970년 1월 1일 12:00 이후의 시간을 밀리초(1/1000초) 단위로 표시합니다. |
| getHours()  setHours() | 시를 0부터 23까지의 숫자로 표시합니다. |
| getMinutes()  setMinutes() | 분을 0부터 59까지의 숫자로 표시합니다. |
| getSeconds()  setSeconds() | 초를 0부터 59까지의 숫자로 표시합니다. |
| getMilliseconds()  setMilliseconds() | 밀리초를 0부터 999까지의 숫자로 표시합니다. |

**Array 객체의 주요 속성과 메서드**

|  |  |  |
| --- | --- | --- |
| 구분 | | 설명 |
| 속성 | length | 배열의 요소 개수를 반환합니다. |
| 파괴적 메서드 | push() | 배열의 맨 뒤에 데이터를 추가합니다. |
| pop() | 배열의 맨 뒤에서 데이터를 추출합니다. |
| unshift() | 배열의 맨 앞에 데이터를 추가합니다. |
| shift() | 배열의 맨 앞에서 데이터를 추출합니다. |
| sort() | 배열의 요소를 정렬합니다. |
| reverse() | 배열의 요소를 역순으로 정렬합니다. |
| 비파괴적 메서드 | forEach() | 배열의 요소를 하나씩 순회하면서 요소마다 콜백(callback) 함수를 호출합니다. |
| filter() | 배열의 요소를 하나씩 순회하면서 요소마다 콜백 함수를 호출해 true를 반환하는 요소만 추출합니다. 추출한 요소로 새로운 배열을 만들고 만들어진 배열을 반환합니다. |
| 비파괴적 메서드 | find() | 배열의 요소를 탐색하면서 주어진 판별 함수를 만족하는 첫 번째 값을 반환합니다. |
| findIndex() | 값 대신 인덱스 숫자를 반환한다는 것만 빼면 find() 메서드와 같습니다. |
| includes() | 배열에 특정 값이 포함되어 있는지 확인해서 포함됐으면 true, 아니면 false를 반환합니다. |
| join() | 배열의 모든 요소를 주어진 구분자로 합칩니다. |

**string 객체의 주요 속성과 메서드**

|  |  |  |
| --- | --- | --- |
| 구분 | | 설명 |
| 속성 | length | 문자열의 길이를 반환합니다. |
| 메서드 | includes() | 메서드의 매개변수에 인자로 전달되는 문자열이 대상 문자열에 포함되어 있으면 true, 아니면 false를 반환합니다. |
| replace() | 대상 문자열에서 메서드의 매개변수에 인자로 전달되는 문자열과 일치하는 한 부분을 찾아서 다른 데이터로 변경한 새로운 문자열을 반환합니다. |
| replaceAll() | 대상 문자열에서 메서드의 매개변수에 인자로 전달되는 문자열과 일치하는 모든 부분을 찾아서 다른 데이터로 변경한 새로운 문자열을 반환합니다. |
| split() | 메서드의 매개변수에 인자로 전달되는 구분자를 이용해 대상 문자열을 여러 개의 문자열로 분리하고, 분리한 문자열을 새로운 배열로 반환합니다. |
| toUpperCase() | 대상 문자열을 대문자로 변경해 반환합니다. |
| trim() | 대상 문자열의 앞뒤 공백을 제거한 값을 반환합니다. |
| indexOf() | 대상 문자열과 일치하는 첫 번째 문자의 인덱스를 반환합니다. |

**자바스크립트 개발 서비스 예제**(개인 포트폴리오용 서비스 주제: 배열 또는 객체 활용)

* **전자상거래 플랫폼**: 상품 목록, 장바구니 관리, 주문 처리.
* **개인 금융 관리 앱**: 수입/지출 기록, 예산 관리, 금융 목표 설정.
* **프로젝트 관리 도구**: 작업 목록, 팀원 관리, 진척도 추적.
* **소셜 미디어 플랫폼**: 사용자 프로필, 게시물 관리, 댓글 기능.
* **온라인 교육 플랫폼**: 강의 목록, 수업 진행 관리, 퀴즈 기능.
* **뉴스 애그리게이터**: 기사 수집, 카테고리 분류, 북마크 기능.
* **레시피 공유 사이트**: 레시피 목록, 재료 검색, 사용자 리뷰.
* **여행 계획 앱**: 여행 일정 관리, 장소 추천, 예산 계산.
* **건강 관리 앱**: 운동 기록, 식단 관리, 건강 목표 설정.
* **이벤트 관리 플랫폼**: 행사 생성, 참석자 관리, 일정 알림.
* **블로그 플랫폼**: 게시물 작성, 태그 관리, 댓글 기능.
* **채팅 애플리케이션**: 사용자 관리, 메시지 전송, 알림 기능.
* **투표 시스템**: 설문 생성, 응답 수집, 결과 분석.
* **포트폴리오 웹사이트**: 프로젝트 목록, 기술 스택, 연락처 양식.
* **유튜브 클론**: 동영상 업로드, 댓글 시스템, 추천 알고리즘.
* **부동산 관리 플랫폼**: 매물 등록, 검색 기능, 문의 관리.
* **기부 플랫폼**: 프로젝트 생성, 기부 관리, 후원자 관리.
* **가계부 앱**: 수입/지출 기록, 카테고리 분석, 보고서 생성.
* **날씨 정보 앱**: 지역별 날씨 조회, 알림 기능, 예보 그래프.
* **쇼핑 목록 관리 앱**: 품목 추가/삭제, 카테고리 분류, 공유 기능.
* **스케줄 관리 앱**: 일정 추가, 알림 기능, 반복 일정 설정.
* **할 일 목록 앱**: 작업 추가, 마감일 설정, 우선순위 지정.
* **커뮤니티 포럼**: 주제별 게시판, 사용자 관리, 신고 기능.
* **버전 관리 시스템**: 파일 버전 추적, 변경 이력 관리, 롤백 기능.
* **QR 코드 생성기**: URL 입력, QR 코드 생성, 다운로드 기능.
* **채용 플랫폼**: 구인 공고 작성, 지원자 관리, 면접 일정 조율.
* **비디오 회의 플랫폼**: 회의 생성, 참가자 초대, 녹화 기능.
* **디지털 달력**: 이벤트 추가, 반복 일정 관리, 공유 기능.
* **음악 스트리밍 서비스**: 플레이리스트 관리, 추천 기능, 사용자 리뷰.
* **사진 공유 앱**: 사진 업로드, 앨범 관리, 댓글 및 좋아요 기능.

**수업자료**

<https://drive.google.com/drive/folders/1LGUxG1upp12jsvtN4DUK5EtbliwCeTJe?usp=sharing>

**자바스크립트 배열 메소드**

**1. push()**

배열의 끝에 요소를 추가합니다.

|  |
| --- |
| let fruits = ['apple', 'banana']; fruits.push('cherry'); // ['apple', 'banana', 'cherry'] console.log(fruits); |

**2. pop()**

배열의 마지막 요소를 제거하고 반환합니다.

|  |
| --- |
| **let** numbers = [1, 2, 3]; **let** lastNumber = numbers.**pop**(); // lastNumber는 3 console.log(numbers); // [1, 2] |

**3. shift()**

배열의 첫 번째 요소를 제거하고 반환합니다.

|  |
| --- |
| let colors = ['red', 'green', 'blue']; let firstColor = colors.shift(); // firstColor는 'red' console.log(colors); // ['green', 'blue'] |

**4. unshift()**

배열의 시작 부분에 요소를 추가합니다.

|  |
| --- |
| let animals = ['dog', 'cat']; animals.unshift('rabbit'); // ['rabbit', 'dog', 'cat'] console.log(animals); |

**5. splice()**

배열의 특정 위치에서 요소를 추가하거나 제거합니다.

|  |
| --- |
| let numbers = [1, 2, 3, 4, 5]; numbers.splice(2, 1, 10); // 인덱스 2에서 1개 요소를 제거하고 10을 추가 console.log(numbers); // [1, 2, 10, 4, 5] |

**6. slice()**

배열의 특정 부분을 잘라내어 새로운 배열을 반환합니다.

|  |
| --- |
| **let** fruits = ['apple', 'banana', 'cherry', 'date']; **let** citrus = fruits.slice(1, 3); // 인덱스 1부터 3까지 잘라냄 console.log(citrus); // ['banana', 'cherry'] |

**7. forEach()**

배열의 각 요소에 대해 제공된 함수를 실행합니다.

|  |
| --- |
| **let** numbers = [1, 2, 3]; numbers.forEach(num => {  console.log(num \* 2); // 2, 4, 6 출력 }); |

|  |
| --- |
| **let** numbers = [1, 2, 3];  numbers.forEach(**function**(num) {  console.log(num \* 2); // 2, 4, 6 출력  }); |

**8. map()**

각 요소에 대해 주어진 함수를 호출하고, 그 결과로 새로운 배열을 생성합니다.

|  |
| --- |
| **let** numbers = [1, 2, 3]; **let** doubled = numbers.map(num => num \* 2); // [2, 4, 6] console.log(doubled); |

**9. filter()**

주어진 조건을 만족하는 요소들로 새로운 배열을 생성합니다.

|  |
| --- |
| **let** numbers = [1, 2, 3, 4, 5]; **let** evenNumbers = numbers.filter(num => num % 2 === 0); // [2, 4] console.log(evenNumbers); |

**10. reduce()**

배열을 단일 값으로 축약하는 데 사용됩니다.

|  |
| --- |
| **let** numbers = [1, 2, 3, 4]; **let** sum = numbers.reduce((acc, num) => acc + num, 0); // 10 console.log(sum); |

**자바스크립트 Array**

자바스크립트 Array는 자바스크립트에서 배열을 생성하고 조작하는 데 사용되는 내장 객체입니다. 배열은 여러 개의 데이터를 순차적으로 저장할 수 있는 데이터 구조로, 각 데이터는 인덱스를 통해 접근할 수 있습니다.

**주요 특징**

* 동적 크기: 자바스크립트의 배열은 크기가 고정되어 있지 않으며, 필요에 따라 요소를 추가하거나 삭제할 수 있습니다.
* 다양한 데이터 유형: 배열은 숫자, 문자열, 객체 등 다양한 데이터 유형을 혼합하여 저장할 수 있습니다.
* 0부터 시작하는 인덱스: 배열의 인덱스는 0부터 시작합니다. 첫 번째 요소는 인덱스 0에 위치합니다.

**배열 생성 방법**

리터럴 표기법

|  |
| --- |
| const fruits = ['apple', 'banana', 'cherry']; |

Array 생성자:

|  |
| --- |
| const numbers = new Array(1, 2, 3, 4, 5); |

**주요 메서드**

* push(): 배열의 끝에 요소를 추가합니다.
* pop(): 배열의 마지막 요소를 제거하고 반환합니다.
* shift(): 배열의 첫 번째 요소를 제거하고 반환합니다.
* unshift(): 배열의 시작 부분에 요소를 추가합니다.
* splice(): 배열의 특정 위치에서 요소를 추가하거나 제거합니다.
* slice(): 배열의 특정 부분을 잘라내어 새로운 배열을 반환합니다.
* forEach(): 배열의 각 요소에 대해 제공된 함수를 실행합니다.
* map(): 각 요소에 대해 주어진 함수를 호출하고, 그 결과로 새로운 배열을 생성합니다.
* filter(): 주어진 조건을 만족하는 요소들로 새로운 배열을 생성합니다.
* reduce(): 배열을 단일 값으로 축약하는 데 사용됩니다.

**예시**

|  |
| --- |
| **const** numbers = [1, 2, 3, 4, 5];  // 배열에 요소 추가 numbers.push(6); // [1, 2, 3, 4, 5, 6]  // 배열에서 요소 제거 **const** last = numbers.pop(); // last는 6, numbers는 [1, 2, 3, 4, 5]  // 배열을 순회 numbers.forEach(num => {  console.log(num); // 1, 2, 3, 4, 5 출력 });  // 평균 계산 **const** average = numbers.reduce((acc, num) => acc + num, 0) / numbers.length; console.log(average); // 3 출력 |

Array 객체는 자바스크립트에서 데이터를 다루는 데 매우 유용하며, 다양한 메서드를 통해 효율적으로 데이터를 처리할 수 있게 해줍니다.

**CSS 셀렉터**

|  |
| --- |
| * 태그: p { margin:0} * 클래스: .classname { margin:0} * 아이디: #uniqueid { margin:0} * 그룹화 셀렉터 h1, h2, h3 { margin: 0; } * 자식 셀렉터 div p {margin:0} div > li {margin} div > .classname {margin} * 형제 셀렉터 h1 + p { margin-top: 20px; } - 인접한 형제 h1 ~ p { color: green; } - 먼형제 * 속성 셀렉터 input[type="text"] { border: 1px solid black; } * 가상클래스 셀렉터 a:hover { text-decoration: underline; } li:first-child { font-weight: bold; } * 가상요소 셀렉터 div::before { content: "▶ "; } div::after { content: " ♥♥♥"; } div::before {  content: url('path/to/your/image.png');  display: inline-block;  margin-right: 5px; } |

**OT**

* 학습자료 추천
* [https://www.w3schools.com](https://www.w3schools.com/)
* 추천 자격증
* SQLD
* 경영정보시각화능력
* 무료 학점은행
* KMOOC
* 특허청: 지식재산학과
* 기상청: 대기과학과
* 기술교육대학교: 컴퓨터공학과, 기계공학과, 메트로닉스학과
* 무료 및 저렴 웹호스팅
* 카페24: 500원 부터 시작
* 닷홈: 무료
* 컴퓨터 프로그램 언어 특성
* 데이터 유형
* 배열: 리스트, 튜플, 딕셔너리
* 제어
* 반복
* 입출력
* 함수
* 클래스
* 파일
* 데이터베이스
* AI풀스택
* 기초역량: 파이썬, MySQL, 아두이노
* 웹개발: HTML5, CSS3, JavaScript, jQuery, Flask
* 빅데이터 분석: 데이터분석/시각화, 머신러닝/딥러닝, 자연어처리, 이미지인식
* 서비스 인프라 구축: Linux, 클라우드, 가상화, Git, Docker
* 디스코드
* <https://discord.gg/5VKKJxAq>
* AIOT 빅데이터 과정 사전 설문
* [설문지](https://docs.google.com/forms/d/13NMlbnF8Ea7qhS9-9AGOMmZ7I3hsCkM9S2y6AwXhtHU/edit)
* 공유문서 공유를 위한 이메일 정보

강사: 이요섭 ( [protizen@gmail.com](mailto:protizen@gmail.com) )

|  |  |  |
| --- | --- | --- |
| **순번** | **이름** | **이메일** |
| 1 | 이지엽 | olzlduq1@gmail.com |
| 2 | 김혜민 | tulipsfairy@gmail.com |
| 3 | 허지원 | dnjsdl1008@gmail.com |
| 4 | 안기부 | cowasdiy1234@naver.com |
| 5 | 황수용 | ghkdtndyd122@naver.com |
| 6 | 황병천 | hbc1347@naver.com |
| 7 | 박선정 | psj0691@naver.com |
| 8 | 임건우 | kw10171@gmail.com |
| 9 | 임정은 | wjddms3416@naver.com |
| 10 | 임승준 | atlasjun@gmail.com |
| 11 | 손유승 | youseung5153@naver.com |
| 12 | 진예찬 | jinstartup@gmail.com |
| 13 | 김동휘 | kimccmpiano@gmail.com |
| 14 | 신준혁 | junhyuk000@naver.com |
| 15 | 정유진 | uzzin822@gmail.com |
| 16 | 주정은 | joojeun1026@naver.com |
| 17 | 손유빈 | dbqls6637@gmail.com |
| 18 |  |  |
| 19 |  |  |
| 20 |  |  |
| 21 |  |  |
| 22 |  |  |
| 23 |  |  |
| 24 |  |  |
| 25 |  |  |