

05

# Javascript와 HTML

# HTML에서 JS불러 사용하기

# 웹 브라우저의 특성

- 서버에서 Content-Type을 text/html로 전송한 경우 렌더링을 진행한다.
- 제일 윗줄부터 해석에 들어간다.
- html이 사용하는 css, js 등이 있다면 해당 파일도 다운로드(로딩)하여 사용한다.
  - css, js도 윗줄부터 로딩 및 해석이 된다.

# js 를 사용하는 두 가지 방법

- 같은 서버, 같은 폴더에 js 파일이 있는 경우

```
<script src="./index.js"></script>
```

- CDN (Content Delivery Network)- Js, Css 등을 전송해주는 서버 집단
- HTML은 내가 제공하는 서버에서 전송하고 Js, Css 등은 CDN에서 다운받음

```
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
```

# Document

- 웹 브라우저가 불러온 웹 페이지 클래스. 페이지 콘텐츠의 Root
  - [https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)
- Js에서 웹 페이지의 Content에 접근하는데 사용할 수 있다.
  - 변수 이름은 document로 제공된다.

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>Hello!!!!</title>
5    </head>
6    <body>
7      <script>
8        console.log(document.title);
9      </script>
10   </body>
11 </html>
```

# HTML의 구성 요소

[https://www.w3schools.com/html/html\\_elements.asp](https://www.w3schools.com/html/html_elements.asp)

- Element: 시작 태그 부터 종료 태그 까지의 항목. 중첩할 수 있다.

`<tagname>Content goes here...</tagname>`

- Attribute

- Element는 추가 정보를 제공하기 위해 Attribute를 가질 수 있다.
- name="value" 형태로 작성하며 시작 태그에 적는다.
- Attribute를 여러 개 사용할 경우 닫는 따옴표 뒤에는 공백 하나가 필요하다.

`<a href="https://www.w3schools.com">Visit W3Schools</a>`

# HTML의 구성 요소

[https://www.w3schools.com/html/html\\_id.asp](https://www.w3schools.com/html/html_id.asp)

- 공통 Attribute

- id: 해당 Element의 식별자. html 파일 안에서 유일하게 설정한다.

```
<h1 id="myHeader">My Header</h1>
```

- class: 주로 스타일 시트의 class와 연동하기 위해 사용하는 것으로 Element를 분류한다. 여러개의 class를 선언하는 것이 가능하다.

- 여러개를 적용할 경우 공백을 넣는다.

```
<h2 class="city main">London</h2>
```

```
<h2 class="city">Paris</h2>
```

```
<h2 class="city">Tokyo</h2>
```

# HTML의 구성 요소

[https://www.w3schools.com/tags/att\\_name.asp](https://www.w3schools.com/tags/att_name.asp)

- 공통 Attribute
  - name
    - js에서 특정 element를 지정
    - form에서 사용자가 입력한 데이터를 서버에서 받을 때 변수 이름으로 사용



# 주요 Element - h, p

[https://www.w3schools.com/html/html\\_headings.asp](https://www.w3schools.com/html/html_headings.asp)

- <hn> Headings: 제목을 출력하기 위한 태그. Hn 의 형태이며 n은 1~6을 사용한다.

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

- <p> Paragraphs: 문단. 새 줄로 시작한다.
  - 문장이 길면 브라우저의 사이즈에 따라 자동으로 줄이 바뀐다.

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

# 주요 Element - table

[https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)

- <table> Table
  - 행 <tr> 안에 열 <td>가 정의 되는 형태
  - 제목 행이나 제목 열의 경우 <th> 태그를 사용한다.
  - html5 버전에서는 <thead>로 제목 행을 <tbody> 로 본문 행, <tfoot>으로 마지막 부분을 나눈다.

# 주요 Element - table

[https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)

- <table> Table

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Title</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Hello</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Everyone</td>
    </tr>
  </tbody>
</table>
```

ID	Title
1	Hello
2	Everyone

실행 화면.  
테두리나 사이즈 등은  
css 또는 style로 상세하게 적용 가능

# 주요 Element - table

[https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)

- span : 병합
  - rowspan="num" 행을 병합
  - colspan="num" 열을 병합

Score는 열병합      Grade는 행병합

Score		Grade
1학기	2학기	
90	95	A
80	85	B

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <title>Hello!!!!</title>
    <style>
      table, th, tr, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
    </style>
  </head>
  <body>
    <table>
      <thead>
        <tr>
          <th colspan="2">Score</th>
          <th rowspan="2">Grade</th>
        </tr>
        <tr>
          <th>1학기</th>
          <th>2학기</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>90</td>
          <td>95</td>
          <td>A</td>
        </tr>
        <tr>
          <td>80</td>
          <td>85</td>
          <td>B</td>
        </tr>
      </tbody>
    </table>
    <script>
      console.log(document.title);
    </script>
  </body>
</html>
```

# 주요 Element - form

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

- 사용자의 입력을 받을 수 있음
- <form> 의 주요 속성
  - action: submit이 발생했을 때 할 일. 주로 데이터를 처리하는 서버 주소
  - target: response를 받았을 때 출력할 곳.
    - \_blank: 새 창. \_self: 현재 창(default) 등이 있다.
  - method: http request method. 생략할 경우 get이 된다. html form은 get/post method만 지원한다.

# 주요 Element - form

[https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

- <form> 태그 안에 입력받고자 하는 항목마다 <input>을 추가
  - 기본 값은 value 속성으로 지정 가능
  - <input type="text"> 가장 일반적인 한줄 문자열 입력
  - <input type="password"> 문자열이지만 화면에 \*로 표시한다.
  - <input type="button"> 버튼, <input type="submit"> form 제출
  - <input type="radio"> 라디오, <input type="checkbox"> 체크박스

# HTML에서 JS를 이용하면

- document 객체를 이용해 form의 입력 검증 가능
- JQuery, Axios 등의 외부 라이브러리를 사용하면 서버로 http request를 전송하고 결과를 받을 수 있음

# form에 접근하기

- forms 배열 또는 name 속성으로 접근할 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
  <head></head>
  <body>
    <form name="registerForm">
      <input type="text" name="name">
      <input type="text" name="phone">
    </form>
    <script>
      // forms 배열을 이용한 접근
      document.forms[0].elements[0].value="user";
      // name 속성을 이용한 접근
      document.registerForm.phone.value="010";
    </script>
  </body>
</html>
```



# form 에 접근하기

[https://www.w3schools.com/jsref/met\\_document\\_getelementbyid.asp](https://www.w3schools.com/jsref/met_document_getelementbyid.asp)

- element의 id, name, tag name
- document.getElementById('id') 등의 함수를 호출하여 변수에 저장하면 해당 항목에 접근할 때마다 document.forms[ ] 와 같이 접근하지 않아도 된다.

```
// id 로 찾기
```

```
const elementName = document.getElementById('name');
```

```
// name 으로 찾기. name은 중복 가능하므로 배열이 반환된다.
```

```
const elementPhone = document.getElementsByName('phone');
```

# form 의 입력 검증하기

- submit 버튼이 눌렸을 때 검증을 통과하면 사용하도록 할 수 있다.

```
<!DOCTYPE html>
<html lang="ko">
  <head></head>
  <body>
    <form name="registerForm" onsubmit="return check()">
      <input type="text" name="name">
      <input type="text" name="phone">
      <input type="submit" value="확인">
    </form>
    <script>
      function check(){
        if(document.registerForm.name.value.length==0){
          alert('이름을 입력하세요');
          // form 전송이 취소된다.
          return false;
        }
        // form이 전송된다.
        return true;
      }
    </script>
  </body>
</html>
```

함수가 true를 반환하면 동작.  
false인 경우 취소된다.

# form 의 입력 검증하기

- element를 받아서 검증하기

```
<!DOCTYPE html>
<html lang="ko">
  <head></head>
  <body>
    <form name="registerForm" onsubmit="return check()">
      <input type="text" name="name" id="name">
      <input type="text" name="phone" id="phone">
      <input type="submit" value="확인">
    </form>
    <script>
      function check(){
        // id 로 찾기
        const elementName = document.getElementById('name');
        // name 으로 찾기. name은 중복 가능하므로 배열이 반환된다.
        const elementPhone = document.getElementsByName('phone');
        if(elementName.value.length==0){
          alert('이름을 입력하세요');
          return false;
        }
        if(elementPhone[0].value.length==0){
          alert('전화번호를 입력하세요');
          return false;
        }
        return true;
      }
    </script>
  </body>
</html>
```

# 입력 검증 코드

## 글자 수

- 글자수 확인 => input.value.length

```
if(document.registerForm.name.value.length==0){  
    alert('이름을 입력하세요');  
    // form 전송이 취소된다.  
    return false;  
}
```

# 입력 검증 코드

## numbers

- 숫자만 입력하기

- input type="number" 로 하는 방법. 클릭 시 우측에 화살표가 보인다.

- <input maxlength="5"> 와 같이 속성을 이용해 입력 길이를 제한한다.

- isNaN 함수를 이용해 숫자로만 적혀 있는지 확인한다.

```
if(isNaN(birthday.value))
```

# 입력 검증 코드

## radio

- radio
  - 중복 선택을 허용하지 않는 인풋. name이 같아야 한다.
  - 중복 선택을 허용하지 않으므로 하나의 값이 return 되고 아무 것도 선택하지 않은 경우 undefined가 반환된다
  - document.querySelector('input[name="name"]:checked')
  - 체크된 최초 항목 하나를 반환

# 입력 검증 코드

## radio

```
<body>
  <form onsubmit="return check()">
    <input type="radio" name="dessert" value="fruit">과일
    <input type="radio" name="dessert" value="coffee">커피
    <input type="radio" name="dessert" value="tea">녹차
    <input type="radio" name="dessert" value="icecream">아이스크림
    <input type="submit" value="선택">
  </form>
  <script>
    function check(){
      const query = 'input[name="dessert"]:checked';
      const selected = document.querySelector(query);
      if(selected==null){
        alert("디저트를 선택해주세요");
        return false;
      }
      console.log(selected.value);
      return false; // console 메시지를 보기위해 false 함
    }
  </script>
</body>
```

# 입력 검증 코드

## checkbox

- 체크박스 : `querySelectorAll` - 조건을 만족하는 모든 항목을 반환
  - `document.querySelectorAll('input[name=name]');`
  - `document.querySelectorAll('input[name=name]:checked');`
  - `document.querySelectorAll('input[type=checkbox]');`
  - `document.querySelectorAll('input[type=checkbox][name=name]');`
- `document.getElementById('id').checked` 로 하나의 상태 확인



# 입력 검증 코드

## checkbox

```
<body>
  <form onsubmit="return check()">
    <input type="checkbox" name="fav" value="fruit">과일
    <input type="checkbox" name="fav" value="coffee">커피
    <input type="checkbox" name="fav" value="tea">녹차
    <input type="checkbox" name="fav" value="icecream">아이스크림
    <input type="submit" value="선택">
  </form>
  <script>
    function check(){
      // 모든 항목의 true/false 확인
      const query = 'input[type="checkbox"]';
      const checks = document.querySelectorAll(query);
      checks.forEach((i)=>{
        console.log(`All: ${i.value} - ${i.checked}`);
      });

      // 체크 된 항목만 검색
      const query2 = 'input[type="checkbox"]:checked';
      const trueChecked = document.querySelectorAll(query2);
      trueChecked.forEach((i)=>{
        console.log(`Checked: ${i.value} - ${i.checked}`);
      });

      // 특정 이름의 체크 된 항목만 검색
      const query3 = 'input[name="fav"]:checked';
      const favChecked = document.querySelectorAll(query3);
      favChecked.forEach((i)=>{
        console.log(`FAV Checked: ${i.value} - ${i.checked}`);
      });
      return false; // console 메시지를 보기위해 false 함
    }
  </script>
</body>
```

# 실습 1

# 새 폴더 만들고 html, js 파일 추가하기

- 새 폴더 만들고 그 폴더에 index.html 파일과 index.js 파일 추가하기

- index.html

```
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <title>Exercise</title>
5      <script src="./index.js"></script>
6      <style>
7        table, th, tr, td {
8          border: 1px solid ■black;
9          border-collapse: collapse;
10       }
11     </style>
12   </head>
13   <body>
14   </body>
15 </html>
```

# index.html 작업

- index.htmldp 다음 화면과 같은 입력 양식을 출력되도록 html 코드를 작성하라. 단 가장 아래 input은 submit 이다.
- 각 항목의 id, name은 다음과 같다.

아이디	<input type="text"/>	id="userid"
생년월일(YYMMDD)	<input type="text"/>	id="birthday"
비밀번호(최소4자)	<input type="text"/>	id="password"
비밀번호 확인	<input type="text"/>	id="password2"
<input type="submit" value="확인"/>		

# index.js 작업

- 입력에 대해 확인하고 alert창을 출력.

아이디	<input type="text"/>
생년월일(YYMMDD)	<input type="text"/>
비밀번호(최소4자)	<input type="text"/>
비밀번호 확인	<input type="text"/>
<input type="button" value="확인"/>	

id="userid",  
비어있을 경우 "아이디를 입력하세요" 출력

id="birthday", 6글자라야 하며 6글자 모두 숫자만 입력  
비어있거나 6글자가 아닐 때 "올바른 생년월일을 입력하세요."

id="password", 최소 4글자인지 확인.  
"비밀번호는 4글자이상 입력하세요."

id="password2",  
password와 같은 내용인지 확인 "비밀번호가 일치하지 않습니다."

# 실습 2

# html과 js 파일 만들기

- 앞의 예제와 마찬가지로 ex2.html 파일과 ex2.js 파일을 만든다.
- css 에서 border-collapse 속성을 삭제하면 테두리가 두줄로 나온다.

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <title>Exercise2</title>
    <script src="./ex2.js"></script>
    <style>
      table, th, tr, td {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

# form 작업

- [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp) 참조
- 성별은 둘 중 하나만 선택 가능
- 연령은 리스트 중 하나 선택 [https://www.w3schools.com/tags/tag\\_select.asp](https://www.w3schools.com/tags/tag_select.asp)
- 신청 항목은 중복 선택 가능

성별	<input type="radio"/> 남성 <input type="radio"/> 여성
연령	10대 미만 ▾
신청 항목	<input type="checkbox"/> 공책 <input type="checkbox"/> 연필 <input type="checkbox"/> 볼펜 <input type="checkbox"/> 자
<input type="button" value="확인"/>	

성별	<input type="radio"/> 남성 <input type="radio"/> 여성
연령	✓ 10대 미만
신청 항목	10대 <input type="checkbox"/> 연필 <input type="checkbox"/> 볼펜 <input type="checkbox"/> 자
확인	20대
	30대
	40대 이상



# form 검증

- 성별이 선택되지 않으면 “성별을 선택해주세요.”
- 신청 항목이 하나도 선택되지 않으면 “신청 항목을 선택해주세요.”
- 신청 항목이 선택되었다면 “n개 신청합니다.” 출력