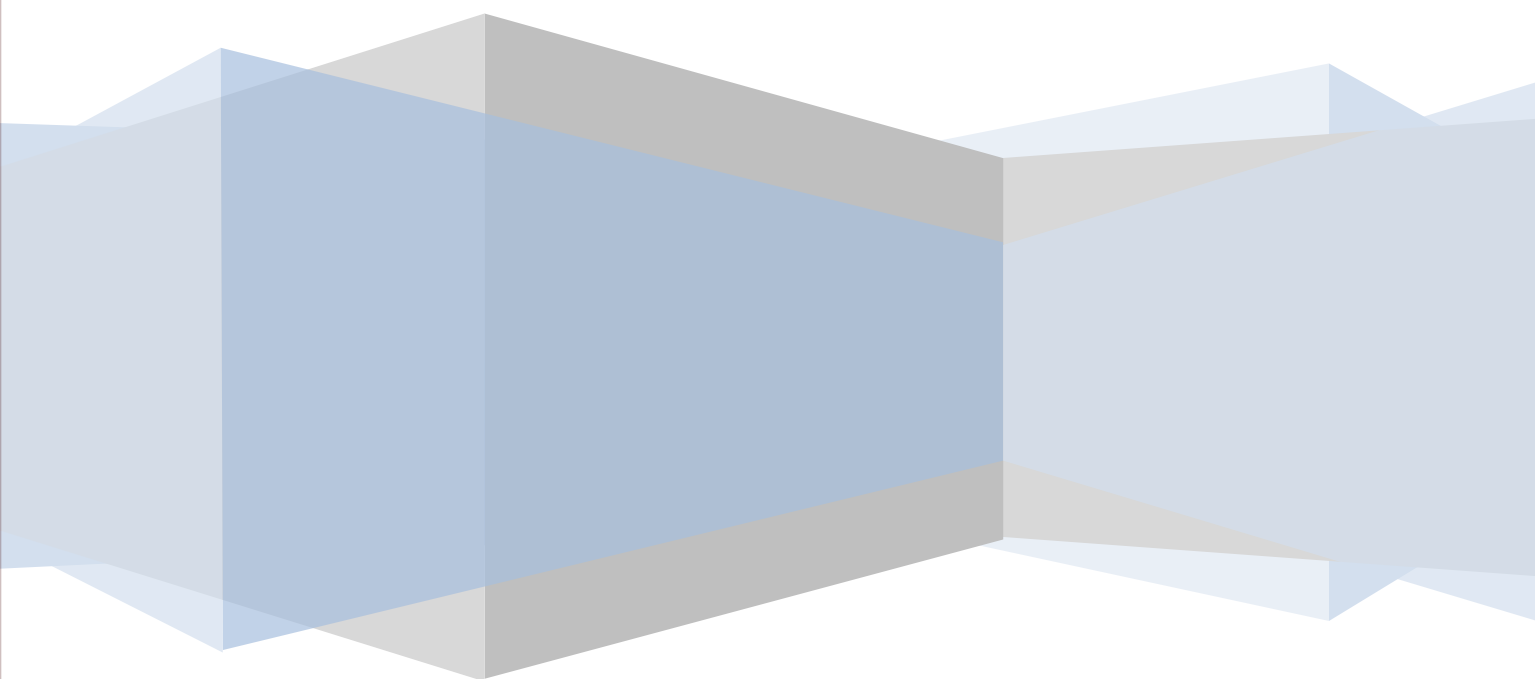


JAVA 개발자

양성과정

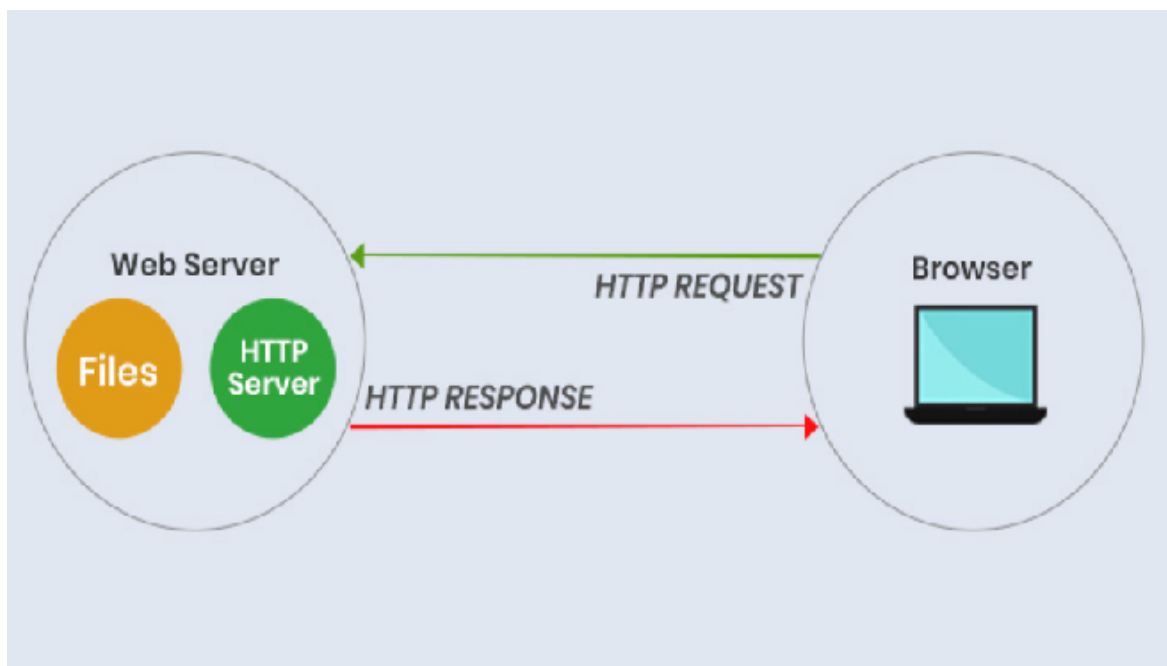
HTML & CSS



HTML 이란?

- Hyper Text Markup Language 이다.
- Hyper Text 란 문서를 서로 연결해주는 링크를 의미한다.
- 마크업 언어란 문서 내에서 요소 (이미지, 텍스트, 테이블) 를 정의하는 데 태그(tags) 를 사용하는 언어이다.
- HTML 의 기본 기능은 웹 브라우저에 보여줄 내용에 마크업하고 문서간에 링크하는 것이다.

● 웹 서버란?



- 웹 서버는 HTML 문서, images, CSS stylesheets, 그리고 JavaScript files 같은 **정적인 파일**을 저장한다.
- 웹 서버의 주된 기능은 웹 페이지를 클라이언트로 전달하는 것이다. 주로 그림 , CSS , 자바스크립트를 포함한 HTML 문서가 클라이언트(웹 브라우저)로 전달된다.

● HTML 문서의 기본 구조

// DOCTYPE : 현재 문서가 HTML5 로 작성된 웹 문서라는 것을 웹 브라우저에 알려준다.

```
<!DOCTYPE html>
```

```
<html lang='ko' >
```

// head : 현재 문서에 대한 정보를 웹 브라우저에 알려준다.

```
<head>
```

```
  <meta charset='UTF-8' >
```

```
  <title> HTML 기본 문서 </title>
```

```
</head>
```

```
<body>
```

```
  // 웹 브라우저에 내용을 표시한다.
```

```
</body>
```

```
</html>
```

웹 문서의 구조를 만드는 시멘틱 태그



● 시멘틱 태그를 사용하여 만든 웹 문서

```
<div = 'container'>
```

```
<header>
```

```
<nav> ... </nav>
```

```
</header>
```

```
<main>
```

```
<section> ... </section>
```

```
<section> ... </section>
```

```
</main>
```

```
<footer>
```

```
<section> ... </section>
```

```
</footer>
```

```
</div>
```

● 시멘틱 태그의 필요성

- 웹 브라우저가 HTML 소스코드만 보고도 문서의 구조를 쉽게 파악할 수 있다.
- 시각 장애인이 웹 사이트를 이용할 때 스크린 리더와 같은 보조기기에서 사이트의 구조를 제대로 이해함으로써 사용자에게 보다 더 정확한 내용을 전달할 수 있다. (웹 접근성 향상)
- 웹 사이트를 검색할 때 필요한 내용을 정확히 찾을 수 있다.

※ Web 접근성이란?

- 장애인이나 고령자들이 웹 사이트에서 제공하는 정보를 비장애인과 동등하게 접근하고 이용할 수 있도록 보장하는 것이다.

1. 헤더영역을 나타내는 <header> 태그

- <header> 태그는 헤더 영역을 의미한다. 사이트 전체의 헤더, 특정 영역의 헤더도 있다.
- 검색 창이나 사이트 메뉴를 삽입한다.

2. 네비게이션 영역을 나타내는 <nav> 태그

- <nav> 태그는 같은 웹 문서안에서 다른 위치로 연결(앵커)하거나 다른 웹문서로 연결하는 링크를 만든다.
- 웹 문서의 위치에 영향을 받지 않는다.

3. 핵심 콘텐츠를 담는 <main> 태그

- <main> 태그는 웹 문서에서 핵심이 되는 내용을 넣는다.
- <main> 태그는 웹 문서에서 한 번만 사용할 수 있다.
- 웹 문서마다 다르게 보여주는 내용으로 구성한다.

4. 독립적인 콘텐츠를 담는 <article> 태그

- <article> 태그는 해당 문서나 페이지 또는 사이트와는 완전히 독립적으로 구성할 수 있는 요소를 정의할 때 사용한다.
- 포럼(forum) 포스트
- 블로그 포스트
- 보도 기사
- 논평(comment) 등

5. 콘텐츠 영역을 나타내는 <section> 태그

- <section> 태그는 웹 문서에서 콘텐츠 영역을 나타낸다.
- HTML 문서의 독립적인 구획을 나타낸다.
- 논리적으로 관계 있는 문서 혹은 요소를 분리할 때 사용한다.
- <section> 요소는 보통 제목 요소(h1~h6)를 자식 요소로 포함하고 있는 경우가 많다.

ex]

HTML 소개

HTML 역사

...

6 . 사이드 바 영역을 나타내는 <aside> 태그

- 사이드바(sidebar)는 목차, 광고 배너, 인기/신규 게시물 목록, 연관/추천 상품 목록과 같이 보조적인 콘텐츠를 배치하기 위해서 사용한다.

7 . 푸터 영역을 나타내는 <footer> 태그

HTML 텍스트 요소

<h> 태그	제목(Heading)을 표시할 때 사용한다. <h1> ~ <h6>
<p> 태그	<p> 태그를 이용하여 문단을 표현한다. 태그(break line)를 사용하면 줄바꿈을 할 수 있다. <pre> 태그를 사용하면 텍스트 서식을 그대로 표현할 수 있다.
<blockquote>	길이가 긴 인용문은 <blockquote>태그를 사용하여 표현할 수 있다. <blockquote>태그는 이러한 인용 부분을 별도의 단락으로 구분하여 나타낸다.

목록(List) 만들기

· 여러 요소(항목)들을 일렬로 나열할때 사용한다.

1. 순서가 없는 리스트 (unorderd list)
2. 순서가 있는 리스트 (orderd list)
3. 정의 리스트 (definition list)

<p> 태그</p>	<ul style="list-style-type: none"> · 순서가 없는 리스트는 태그로 시작한다. · 각각의 리스트 항목은 태그로 시작한다. · <ul type='disc'> <ul style="list-style-type: none"> disc : ● square : ■ circle : ○ · CSS 의 list-style-type 속성을 사용하여 설정할 수 있다. <ul style="list-style-type: none"> <ul style = 'list-style-type: circle'>
<p> 태그</p>	<ul style="list-style-type: none"> · 순서가 있는 리스트는 태그로 시작한다. · 각각의 리스트 항목은 태그로 시작한다. · <ol type='1'> · CSS 의 list-style-type 속성을 사용하여 설정할 수 있다. <ul style="list-style-type: none"> <ol style = 'list-style-type: upper-alpha'> <ul style="list-style-type: none"> - decimal : 숫자 (기본설정) : 1 - upper-alpha : 영문 대문자 : A - lower-alpha : 영문 소문자 : a - upper-roman : 로마 숫자 대문자 : i - lower-roman : 로마 숫자 소문자 : I
<p><dl></p>	<ul style="list-style-type: none"> · 용어와 그에 대한 정의를 모아놓은 리스트이다. · <dt> 태그에는 용어의 제목이 들어가고, <dd> 태그에는 해당 용어에 대한 설명을 기술한다. <pre> <dl> <dt>Google Chrome</dt> <dd>Google Chrome is a freeware web browser developed by Google</dd> </dl> </pre>

표(Table) 만들기

<table> 태그	<ul style="list-style-type: none"> · <table> 태그를 사용하여 표를 작성한다. · CSS의 border 속성을 이용하여 테이블에 테두리를 표현할 수 있다. <pre> <style> table, th, td { border: 1 px solid black; border-collapse: collapse; } </style> </pre>
<tr> 태그	<tr> 태그는 테이블에서 행을 만들때 사용한다.
<td> 태그	<td> 태그는 테이블의 열을 각각의 셀로 나누어준다.
<th> 태그	<th> 태그는 각 열의 제목을 만들때 사용한다.
colspan 속성	colspan 속성을 사용하면 테이블의 열(column)을 합칠 수 있다. <td colspan = '병합하고자 하는 컬럼 수'> ... </td>
rowspan 속성	rowspan 속성을 사용하면 테이블의 행(row)을 합칠 수 있다. <td rowspan = '병합하고자 하는 행 수'> ... </td>
테이블 구조 관련 태그 : 웹 접근성	
<caption>태그	<caption>태그를 사용하면 테이블 상단에 제목이나 짧은 설명을 붙일 수 있다.
<thead> 태그	테이블에 제목을 나타낸다.
<tbody> 태그	테이블의 본문(body)을 나타낸다.
<tfoot> 태그	테이블에 대한 요약을 나타낸다.

```
<table>
  <caption>표에 대한 설명</caption>
  <thead>
    표의 제목
  </thead>
  <tbody>
    표의 내용
  </tbody>
  <tfoot>
    표에 대한 요약
  </tfoot>
</table>
```

하이퍼링크 삽입하기

- 하이퍼링크를 사용하면 다른 페이지나 다른 사이트로 연결한다.
- HTML에서는 `<a>` 태그로 하이퍼링크를 표현한다.

// href 속성은 'HTML 링크' 텍스트를 클릭하면 연결한 웹 문서나 사이트의 url 주소를 명시한다.

```
<a href = "링크할 주소" target='_self'> HTML 링크 </a>
```

target 속성값	설명
_blank	링크로 연결된 문서를 새창이나 새탭에서 엽니다.
_self	링크로 연결된 문서를 현재 프레임(frame)에서 엽니다. (기본설정)
_parent	링크로 연결된 문서를 부모 프레임(frame)에서 엽니다.
_top	링크로 연결된 문서를 현재 창의 가장 상위 프레임(frame)에서 엽니다.
프레임 이름	링크로 연결된 문서를 지정된 프레임(frame)에서 엽니다.

```
<body>
```

```
  <h1>a 태그의 target 속성값</h1>
```

```
  <h2><a href="/intro.html" target="_blank">blank</a></h2>
```

```
  <h2><a href="/intro.html" target="_self">self</a></h2>
```

```
  <h2><a href="/intro.html" target="_parent">parent</a></h2>
```

```
  <h2><a href="/intro.html" target="_top">top</a></h2>
```

```
  <h2><a href="/intro.html" target="myframe">myframe</a></h2>
```

```
  <iframe name="myframe" style="width:50%; height: 330px"></iframe>
```

```
</body>
```

iframe 요소

- iframe이란 inline frame의 약자이다.
- iframe 요소를 이용하면 해당 웹 페이지 안에 어떠한 제한 없이 또 다른 하나의 웹 페이지를 삽입할 수 있다.

// 외부 페이지의 콘텐츠를 임베드 할 수 있다.

```
<iframe src='삽입할 웹 페이지 주소' style = "width:100%; height:300px" >> </iframe>
```

//구글 지도를 웹 페이지에 임베드 하는 경우

<https://maps.google.com/?q=37.4982338,127.0320854&output=svembed>

이미지 삽입하기

· 웹에서 사용되는 대표적인 이미지 파일 형식은 다음과 같다.

1. GIF 이미지
2. JPEG / JPG 이미지
3. PNG 이미지

```
<img src = "이미지 파일 경로" alt = "대체 문자열" >
```

● 이미지 크기 / 테두리 설정

// HTML에서는 style 속성을 사용하여 이미지의 크기 및 테두리를 설정할 수 있다.

```
<style>
  img {
    width: 200px;
    height: 200px;
    border: 1px solid black;
  }
</style>

<img src='이미지 파일 경로' alt='대체 문자열' />
```

● 이미지에 링크(link) 설정

```
<a href="링크할 주소" target="_blank">
  
</a>
```


오디오와 비디오 삽입하기

● 비디오(video) 요소

```
<video style="width:576px; height:360px" controls autoplay loop>  
  // 웹 브라우저는 위쪽에서부터 순서대로 가장 먼저 인식되는 파일 형식과 파일 주소를 사용한다.  
  <source src="비디오 파일 경로" type="video/mp4">  
  <source src="비디오 파일 경로" type="video/ogg">  
  이 문장은 사용자의 웹 브라우저가 video 요소를 지원하지 않을 때 나타납니다.  
</video>
```

control 속성	재생, 정지 및 소리의 조절 등 비디오의 기본적인 동작을 조절할 수 있는 패널을 생성한다.
height, width 속성	웹 브라우저에 삽입되는 비디오 플레이어의 크기를 설정할 수 있다.
<source> 태그	웹 브라우저는 여러 개의 <source>태그 중 위쪽에서부터 순서대로 가장 먼저 인식되는 파일의 타입과 비디오 파일의 경로를 사용한다.
autoplay 속성	웹 페이지가 로드(load) 될 때 비디오를 자동으로 재생시켜 줄지 않을지를 설정한다.
loop 속성	loop 속성을 설정하면 비디오의 재생이 끝나도 계속적으로 반복해서 비디오를 재생한다.

● 오디오(audio) 요소

```
<audio controls autoplay>
```

//웹 브라우저는 위쪽에서부터 순서대로 가장 먼저 인식되는 파일 형식과 파일 주소를 사용한다.

```
<source src="오디오 파일 경로" type="audio/ogg">
```

```
<source src="오디오 파일 경로" type="audio/mp3">
```

이 문장은 사용자의 웹 브라우저가 video 요소를 지원하지 않을 때 나타납니다.

```
</audio>
```

※ HTML 지원하는 오디오 파일 형식

1. MP3 : Moving Picture Experts Group에 의해 개발되었으며, MPEG-1의 오디오 규격으로 개발된 손실 압축형 파일 형식이다.
2. WAV : IBM과 Microsoft에 의해 개발되었으며, 개인용 PC에서 오디오를 재생하기 위한 IBM과 Microsoft의 표준 오디오 파일 형식이다.
3. Ogg : Xiph 재단에 의해 개발되었으며, MP3의 대안으로 개발된 특허권으로 보호되지 않는 개방형 공개 멀티미디어 파일 형식이다.

폼 삽입하기

- 웹 문서에서는 <form> 태그를 사용하여 사용자로부터 데이터를 입력 받고, 사용자가 입력한 데이터를 서버로 보낼 때 사용한다.

```
<form action='요청을 처리할 서버 페이지 주소' method='get | post'>
  input 요소들 ...
</form>
```

method	<p>사용자로 부터 입력받은 데이터를 서버에 전달할 방식 GET 과 POST 방식으로 나눈다.</p> <ol style="list-style-type: none"> 1. GET 방식은 주소에 데이터(data)를 추가하여 전달하는 방식이다. 데이터가 주소 입력창에 그대로 나타나며, 전송할 수 있는 데이터의 크기 또한 제한적이다. 따라서 검색 엔진의 쿼리(query)와 같이 크기가 작고 중요도가 낮은 정보를 보낼 때 주로 사용한다. 2. POST 방식은 데이터(data)를 별도로 첨부하여 전달하는 방식입니다. 데이터가 외부에 드러나지 않으며, 전송할 수 있는 데이터의 크기 또한 제한이 없다. 따라서 보안성 및 활용성이 GET 방식보다 좋다.
action	입력받은 데이터를 처리할 서버 프로그램의 주소를 지정하다.
name	자바스크립트로 폼을 제어할 때 사용할 폼의 이름을 지정한다.

● 사용자 입력을 위한 input 태그

text	한 줄의 텍스트를 입력받을 수 있다.
password	비밀번호를 입력받을 수 있다. (별표나 원모양이 표시된다.)
search	검색어를 입력할 수 있는 텍스트 필드를 넣는다.
url	URL 주소를 입력할 수 있는 입력 필드를 넣는다. 이 입력 필드는 입력된 값이 유효한 URL 주소인지 여부를 자동으로 검증한다.
email	이메일 주소를 입력할 수 있는 입력 필드를 넣는다. 이 입력 필드는 폼 데이터를 제출(submit)하기 전 입력받은 값이 유효한 이메일 주소 인지를 자동으로 검증한다. 또한, multiple 속성을 명시하여 여러 개의 이메일 주소를 동시에 입력받을 수 있다. (콤마로 구분)
tel	전화번호를 입력할 수 있는 입력 필드를 넣는다.
checkbox	체크박스는 사용자가 여러 선택사항 중에서 하나 이상의 값을 선택할 수 있다.
radio	서로 연관된 옵션들의 집합을 보통 하나의 라디오 버튼 그룹으로 나타내며, 그룹 내의 라디오 버튼은 오직 하나만 선택할 수 있다.
number	숫자를 입력할 수 있는 입력 필드와 조절할 수 있는 스펀 박스를 넣는다. - max : 최대값 - min : 최소값 - step : 숫자들 사이의 간격 - value : 초기값
range	숫자를 조절할 수 있는 슬라이드 막대를 넣는다. - max : 최대값 - min : 최소값 - step : 숫자들 사이의 간격 - value : 초기값

date	자동으로 입력 내용을 검증하는 텍스트 필드를 통해 날짜를 직접 입력하거나, date picker 를 사용하여 날짜를 선택할 수 있다.
time	자동으로 입력 내용을 검증하는 텍스트 필드를 통해 시간을 직접 입력하거나, time picker 를 사용하여 시간을 선택할 수 있다.
datetime-local	사용자가 있는 지역을 기준으로 날짜와 시간을 넣는다.
submit	서버로 폼 데이터를 전송하는 제출 버튼(submit button)을 넣는다.
reset	폼(form) 요소의 값을 초기값으로 되돌리는 리셋 버튼을 넣는다.
image	제출 버튼(submit button)으로 사용될 이미지를 넣는다. <input type='image' src='이미지 경로'>
button	일반 버튼을 넣는다. 주로 자바스크립트에서 사용된다.
file	업로드할 파일을 선택할 수 있는 입력 필드와 “파일 선택” 버튼을 넣는다.
hidden	사용자에게는 보이지 않는 숨겨진 입력 필드를 넣는다.

<input type='email'>

```
<form action='요청을 처리할 페이지 주소' method='get'>
  수신인 : <input type="email" name="recipients"><br>
  참조인 : <input type="email" name="reference" multiple><br>
  <input type="submit">
</form>
```

<input type='checkbox'>

```
<form action = "요청을 처리할 페이지 주소" method = "get">
  여러분의 취미를 선택하세요<br>
  <input type="checkbox" name="hobby" value="game">게임 <br>
  <input type="checkbox" name="hobby" value="climb">등산 <br>
  <input type="checkbox" name="hobby" value="travel">여행 <br>
  <input type="checkbox" name="hobby" value="fishing">낚시<br>
  <input type="submit">
</form>
```

<input type='radio'>

```
<form action = "요청을 처리할 페이지 주소" method = "get">
  여러분의 나이대를 골라보세요. <br>
  <input type="radio" name="ages" value="teenage">10 대 <br>
  <input type="radio" name="ages" value="twenties">20 대 <br>
  <input type="radio" name="ages" value="thirties">30 대 <br>
  <input type="radio" name="ages" value="forties">40 대 이상 <br>
  <input type="submit">
</form>
```

<input type='file'>

```
<form action='요청을 처리할 서버 페이지 주소' method='post' enctype='multipart/form-data'>
  사진 : <input type="file" name="file1"> <br>
  <input type="submit">
</form>
```

● input 태그 속성

autofocus	웹 문서가 열리면 입력 필드안에 마우스 포인터를 표시한다.
placeholder	텍스트 필드에 힌트를 표시한다.
readonly	입력 필드를 읽기 전용으로 지정한다.
required	필수입력 필드를 지정한다.
size	텍스트 관련 필드에서 화면에 표시할 크기를 지정한다.

폼에서 사용하는 여러 가지 태그

<textarea> 태그

- 사용자가 여러 줄의 텍스트를 입력할 수 있는 텍스트 입력 영역을 정의할 때 사용한다.
- 텍스트 입력 영역의 크기는 <textarea> 요소의 cols 속성과 rows 속성으로 지정할 수 있으며, CSS에서 height 속성과 width 속성을 사용하면 더욱 손쉽게 지정할 수 있다.

```
<form action='요청을 처리할 서버 페이지 주소' method='get'>
  <textarea name="memo" cols="30" rows="5">메모를 입력하세요</textarea> <br>
  <input type="submit">
</form>
```

<select> 태그

- 옵션 메뉴를 제공하는 드롭다운 리스트(drop-down list)를 정의할 때 사용한다.
- <select> 요소 내부의 <option> 요소는 드롭다운 리스트(drop-down list)에서 사용되는 각각의 옵션을 정의한다.

```
<select name='coffee'>
  <option value="americano"> 아메리카노 </option>
  <option value="caffelatte"> 카페라테 </option>
  <option value="cafeaulait"> 카페오레 </option>    >
  <option value="espresso"> 에스프레소 </option>
</select>
```

HTML 드래그 앤 드롭 API

● DragEvent

- 드래그 이벤트는 사용자가 draggable 요소를 마우스로 선택하고, 마우스 포인터를 droppable 요소로 가져가 마우스 버튼을 땔때 발생한다. 드래그하는 도중에 많은 이벤트가 발생하고, 몇몇 이벤트는 여러번 발생하기도 한다.

이벤트	설명
dragover	드래그하는 요소가 드롭 영역을 지나갈 때 발생한다.
drop	드래그 하는 요소를 드롭 영역에 드롭 했을 때 발생한다.
dragenter	드래그하는 요소가 드롭 영역을 진입할 때 발생한다.
dragleave	드래그하는 요소가 드롭 영역에서 벗어났을 때 발생한다.

● 드롭 영역 정의하기

- 기본적으로는 브라우저는 HTML 요소에 뭔가를 드롭했을 때 아무 일도 일어나지 않는다. 특정 요소를 드롭 영역(droppable)으로 만들기 위해서는 해당 요소에 반드시 ondragover 와 ondrop 이벤트를 등록해야 한다.

● 드래그와 드롭 관련 인터페이스

- HTML 드래그와 드롭 관련 인터페이스는 `DragEvent`, `DataTransfer`, `DataTransferItem`, `DataTransferItemList` 이다.
- `DragEvent`의 `dataTransfer` 프로퍼티가 `DataTransfer`객체이다.
- `DataTransfer`객체는 드래그 데이터 (하나 이상의 아이템) 를 유지 관리한다.
- `DataTransfer`객체는 `DataTransferItem` 목록(`DataTransferItemList`)인 `items` 프로퍼티를 가지고 있다.
- 각 `DataTransferItem`은 하나의 드래그 아이템을 나타내고 각 아이템은 데이터의 종류 (`string` 혹은 `file`) 를 나타내는 `kind` 프로퍼티와 데이터 아이템의 종류 (MIME 종류) 를 나타내는 `type` 프로퍼티를 갖는다.

CSS Cascading Style Sheets

※ CSS를 사용하는 이유

- CSS는 웹 페이지의 스타일을 별도의 파일로 저장할 수 있어서 사이트의 전체 스타일을 손쉽게 제어할 수 있다.
- 웹 사이트의 스타일을 일관성 있게 유지할 수 있게 해주며, 그에 따른 유지 보수 또한 쉽다.
- 외부 스타일 시트는 보통 확장자를 .css 파일로 저장한다.

※ CSS 문법

```
선택자 { 속성명: 속성값; 속성명: 속성값; ... }
```

```
p {  
    color: blue;  
    text-align: center;  
}
```

- CSS의 문법은 선택자(Selector)와 선언부(Declaratives)로 구성된다.
- 선택자는 CSS를 적용하고자 하는 HTML 요소를 가리킨다.
- 선언부는 하나 이상의 선언들을 세미콜론(;)으로 구분하며, 중괄호를 사용하여 전체를 둘러싼다.
- 각 선언은 CSS 속성명(property)과 속성값(value)을 가지며 세미콜론(;)으로 끝난다.

CSS 스타일 적용

· HTML 문서에 CSS 스타일을 적용할 때에는 다음과 같이 세 가지 방법을 사용한다.

1. 인라인 스타일(Inline style)
2. 내부 스타일 시트(Internal style sheet)
3. 외부 스타일 시트(External style sheet)

● 인라인 스타일(Inline style)

· HTML 요소 내부에 style 속성을 사용하여 CSS 스타일을 적용하는 방법이다.

이러한 인라인 스타일은 해당 요소에만 스타일을 적용할 수 있다.

```
<body>
  <h2 style='color:green; text-decoration:underline;' >
    인라인 스타일을 이용하여 스타일을 적용하였습니다.
  </h2>
</body>
```

● 내부 스타일 시트(Internal style sheet)

· HTML 문서 내의 <head>태그에 <style>태그를 사용하여 CSS 스타일을 적용한다.

이러한 내부 스타일 시트는 해당 HTML 문서에만 스타일을 적용할 수 있다.

```
<head>
  <style>
    body {
      background-color: lightyellow;
    }
  </style>
</head>
```

● 외부 스타일 시트(External style sheet)

· 외부에 작성된 CSS 파일은 .css 확장자를 사용한다.

· 외부 스타일 시트를 사용하면 웹 사이트 전체의 스타일을 하나의 파일에서 변경할 수 있다.

· 스타일을 적용할 웹 페이지의 <head>태그에 <link>태그를 사용하여 외부 스타일 시트를 포함 해야만 스타일이 적용된다.

```
<head>
  <link rel='stylesheet' href='외부 스타일 시트 파일 경로'>
  <style> ... </style>
</head>
```

스타일 적용의 우선순위

1. !important
2. 인라인 스타일 (HTML 요소 내부에 위치함)
3. 내부 / 외부 스타일 시트 (HTML 문서의 head 요소 내부에 위치함)
 - 내부 스타일 시트와 외부 스타일 시트는 가장 마지막에 적용된 스타일 시트가 적용된다.
4. 웹 브라우저 기본 스타일

```
<style>
h1 { color: blue; !important; }
</style>
```

스타일 상속

- 부모 요소에서 설정된 일부 CSS 속성 값은 자식 요소에 의해 상속되며, 일부는 그렇지 않는 경우도 있다.

CSS 기본 선택자(selector)

· CSS에서는 스타일을 적용할 요소를 선택하기 위해서 선택자(selector)를 사용한다.

1. 전체 선택자
2. HTML 요소 선택자
3. 아이디(id) 선택자
4. 클래스(class) 선택자
5. 그룹(group) 선택자

전체 선택자

· CSS를 적용할 대상으로 HTML 문서 내부의 모든 요소를 선택한다.

```
<style>  
  * { color: blue; }  
</style>
```

HTML 요소 선택자

· CSS를 적용할 대상으로 HTML 요소의 이름을 사용하여 선택한다.

```
<style>  
  h1 { color: blue; text-decoration: underline; }  
</style>
```

아이디(id) 선택자

- 아이디 선택자는 CSS를 적용할 대상으로 특정 요소를 선택할 때 사용한다.

```
<style>
  #heading { color: blue; text-decoration: line-through; }
</style>
...
<h3 id='heading'></h3>
```

클래스(class) 선택자

- 클래스 선택자는 특정 집단의 여러 요소를 한 번에 선택할 때 사용한다. 이러한 특정 집단을 클래스(class)라고 하며, 같은 클래스 이름을 가지는 요소들을 모두 선택해 준다.

```
<style>
  .heading { color: blue; }
</style>
...
<h2 class='heading'></h2>
<h3 class='heading'></h3>
```

그룹 선택자

- 그룹 선택자는 위에서 언급한 여러 선택자를 같이 사용하고자 할 때 사용한다.
- 그룹 선택자는 여러 선택자를 쉼표(,)로 구분하여 연결한다
- 그룹 선택자는 코드를 중복해서 작성하지 않도록 하여 코드를 간결하게 만들어준다.

```
<style >
  h2 { color: navy; }
  h2, h3 { text-align: center; }
  h2, h3, p { background-color: lightgray; }
</style >
```

화면 레이아웃과 관련된 스타일

● position 속성

· position 속성은 엘리먼트(요소)를 페이지에 어떻게 위치할 지를 지정하는 속성이다.

1. static
2. relative
3. absolute
4. position

static

· position을 아무것도 지정하지 않으면 사용되는 기본 값이다. position 속성을 static 으로 지정하면 위치가 지정된 것이 아니라 그냥 HTML 문서에 따라 위치가 정해졌다고 보면 된다.

```
div.static { position: static; }
```

relative

· position 을 relative 로 지정하면 해당 요소가 기본적으로 표시된 위치를 기준으로 새로운 위치를 지정 할 수 있다. 이 말을 다른 말로 하면 position 을 relative 로 지정하더라도 top 이나 left, right, bottom 등으로 새로운 위치를 지정하지 않는 이상 기본적으로 표시된 위치와 다르지 않음을 이야기 하는 것이다.

```
div.relative {  
  position: relative;  
  top: -100px;  
  left: -100px;  
  bottom: -100px;  
  right: -100px;  
}
```

absolute

· position 을 absolute 로 지정하는 것은 fixed 와 비슷하지만 다르다. fixed 는 사용자의 웹 브라우저 기준을 기준으로 위치를 고정시키는 것이지만 absolute 는 해당 요소의 위치가 지정된 부모 요소를 기준으로해서 fixed 시키는 것과 같다고 생각하면 된다.

위치가 지정된 부모요소라는 것은 결국 position 속성이 static 이 아닌 다른 값으로 지정 되어야 함을 의미한다.

```
div.parent { position: relative; }  
  
div.child {  
  position: absolute;  
  top: 0px;  
  left: 0px;  
  bottom: 50px;  
  right: 50px;  
}
```

fixed

· position 을 fixed 로 지정하면 해당 요소의 위치를 사용자의 웹 브라우저를 기준으로 설정할 수 있다. 스크롤을 내려도 항상 같은 곳에 위치하는 상단바, 하단바, 광고 등이 이 속성을 이용하는 것이다. 이를 위해서는 top, bottom, left, right 등으로 위치를 지정하고 width 와 height 로 요소의 크기를 지정할 수 있다.

```
// 스크롤을 내려도 하단에 항상 고정되어 있는 하단바를 만드는 예제이다.
```

```
div.footer {  
    position: fixed;  
    bottom: 0;  
    left: 0;  
    width: 100%;  
    height: 50px;  
    background-color: blue;  
}
```

● display 속성

- 웹 페이지의 레이아웃을 결정하는 속성이다.
- 모든 HTML 요소는 display 속성값을 갖고 있으며 대부분 block 아니면 inline 이다.

1. block
2. inline
3. inline-block
4. none
5. flex

block 요소

- display 속성값이 블록(block)인 요소는 언제나 줄바꿈한다. 왜냐하면 width가 100%이므로 해당 라인의 모든 너비를 차지하기 때문이다.
- width, height 속성을 지정할 수 있다.

```
<p style='border:3px; solid:red;'>  
    p요소는 display 속성값이 블록인 요소입니다.  
</p>
```

※ <div>요소

- <div>요소는 다른 HTML 요소들을 하나로 묶는 데 자주 사용되는 대표적인 블록(block) 요소이다.

inline 요소

- display 속성값이 인라인(inline)인 요소는 줄바꿈이 되지 않는다.
왜냐하면 인라인 요소의 너비(width)는 HTML 요소의 내용(content) 만큼 차지한다.
- width와 height를 지정할 수 없다.
- , <a>, 요소는 display 속성값이 인라인(inline)인 대표적인 요소이다.

inline-block 요소

- display 속성이 inline-block 으로 지정된 요소는 기본적으로 인라인 요소처럼 전후 줄바꿈 없이 한 줄에 다른 요소들과 나란히 배치된다. 하지만 인라인 요소에서 불가능하던 width 와 height 속성 지정 및 margin 과 padding 속성의 상하 간격 지정이 가능해진다.

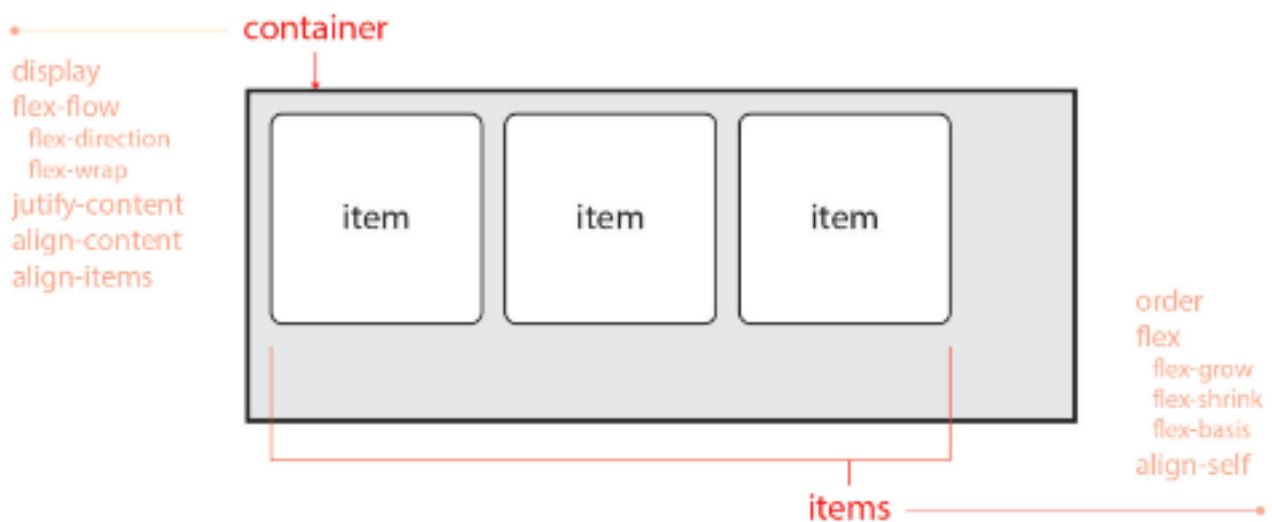
none 요소

- display 속성이 none인 요소는 화면에 표시되지 않는다. visibility 속성과 다르게 none인 요소는 화면에 해당 요소에 대한 공간을 차지 않는다.

● CSS3 Flexible Box

- Flex Box는 요소의 크기가 불분명하거나 동적인 경우에도, 각 요소를 정렬할 수 있는 효율적인 방법을 제공한다.
- Flex Box는 Flex Container, Items 으로 구성된다.
- Flex Container는 Items를 감싸는 부모 요소이며, 각 Item을 정렬하기 위해선 필수이다.
- Flex Container와 Items에 적용하는 속성이 구분되어 있다.

Flex Container에는 display, flex-flow, justify-content 등의 속성을 사용할 수 있으며, Items에는 order, flex, align-self 등의 속성을 사용할 수 있다.



Flex Container

속 성	설 명
display (flex)	flex container 를 정의한다.
flex-flow	flex-direction 와 flex-wrap 의 단축 속성이다.
flex-direction	Items 의 주 축(main-axis)을 설정
flex-wrap	Items 의 줄바꿈을 설정
justify-content	주 축(main-axis)의 정렬 방법을 설정
align-content	교차 축(cross-axis)의 정렬 방법을 설정(2 줄 이상)
align-items	교차 축(cross-axis)에서 Items 의 정렬 방법을 설정(1 줄)

1. flex-direction

· Items의 주축(main-axis)을 설정한다.

속성 값	설 명
row (기본값)	items 를 수평축(왼쪽에서 오른쪽으로)으로 표시
row-reverse	items 를 row 의 반대축으로 표시
column	items 를 수직축(위에서 아래로)으로 표시
column-reverse	items 를 column 의 반대축으로 표시
justify-content	주 축(main-axis)의 정렬 방법을 설정
align-content	교차 축(cross-axis)의 정렬 방법을 설정(2 줄 이상)
align-items	교차 축(cross-axis)에서 Items 의 정렬 방법을 설정(1 줄)

● 주축(main-axis)과 교차축(cross-axis)

- `flex-direction: row` 인 경우 `items`를 수평축으로 표시하므로 이때는 주 축이 수평이며 교차 축은 수직이 된다.
- `flex-direction: column` 인 경우 `items`를 수직축으로 표시하므로 주 축은 수직이며 교차 축은 수평이 된다. 즉, 방향(수평, 수직)에 따라 주 축과 교차 축이 달라진다.

● 시작점(flex-start)과 끝점(flex-end)

- 주 축이나 교차 축의 시작하는 지점과 끝나는 지점을 지칭한다.
역시 방향에 따라 시작점과 끝점이 달라진다.

2. flex-wrap

- `Items` 의 줄 바꿈을 설정한다.

속성 값	설 명
nowrap	모든 <code>items</code> 를 한줄에 표시한다. (기본값)
wrap	<code>items</code> 를 여러줄에 표시한다.
wrap-reverse	<code>items</code> 를 <code>wrap</code> 의 역 방향으로 여러줄에 표시한다.

3. justify-content

속성 값	설 명
flex-start	items 를 시작점(flex-start)으로 정렬한다.
flex-end	items 를 끝점(flex-end)으로 정렬한다.
center	items 를 가운데 정렬한다.
space-between	시작 item 은 시작점에 마지막 item 은 끝점에 정렬되고 나머지 items 는 사이에 고르게 정렬된다.
space-around	items 를 균등한 여백을 포함하여 정렬한다.

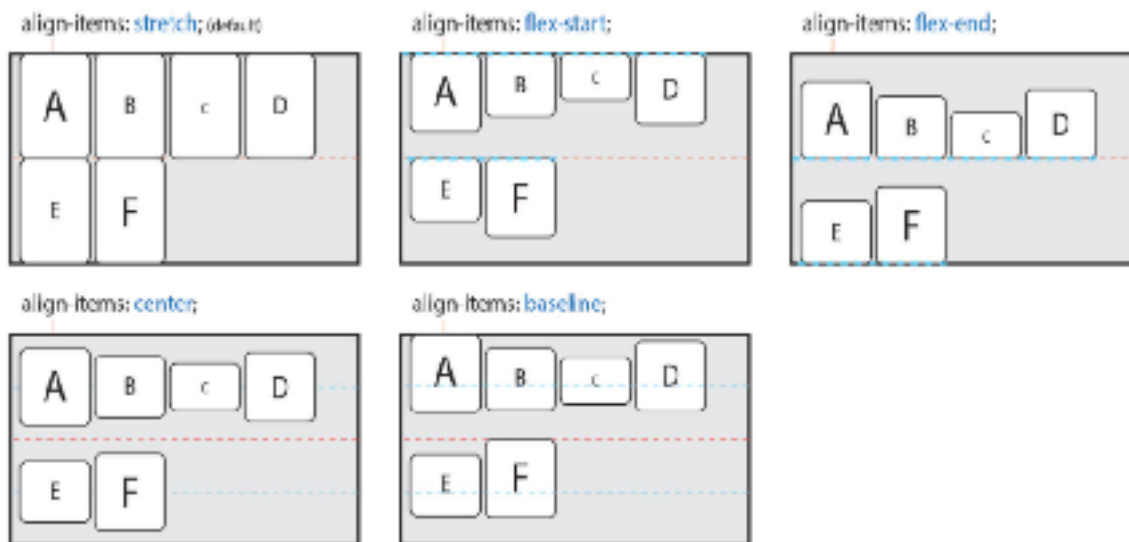
4. align-content

- 교차 축(cross-axis)의 정렬 방법을 설정한다.
- 주의할 점은 flex-wrap 속성을 통해 Items 가 여러 줄(2 줄 이상)이고 여백이 있을 경우만 사용할 수 있다.

속성 값	설 명
stretch (기본값)	flex container 의 교차축을 채우기 위해 items 를 늘린다.
flex-start	items 를 시작점(flex-end)으로 정렬한다.
flex-end	items 를 끝점(flex-end)으로 정렬한다.
center	items 를 가운데 정렬한다.
space-between	시작 item 은 시작점에 마지막 item 은 끝점에 정렬되고 나머지 items 는 사이에 고르게 정렬된다.
space-around	items 를 균등한 여백을 포함하여 정렬한다.

5. align-items

- 교차 축(cross-axis)에서 items의 정렬 방법을 설정한다.
 - items가 한 줄일 경우 많이 사용한다.
 - Items가 flex-wrap 을 통해 여러 줄(2줄 이상)일 경우에는 align-content 속성이 우선한다.
- 따라서, align-items 를 사용하려면 align-content 속성을 기본값(stretch)으로 설정해야 한다.



Flex Items

속성 값	설 명
order (0)	item 의 순서를 설정한다.
flex	flex-grow, flex-shrink, flex-basis 의 단축 속성
flex-grow (0)	flex item 의 증가 너비 비율을 설정한다.
flex-shrink (1)	flex item 의 감소 너비 비율을 설정한다.
flex-basis (auto)	flex item 의 기본 너비 설정한다.
align-self	교차축(cross-axis)에서 item 의 정렬 방법을 설정한다.

1. order

- item의 순서를 설정한다.
- item에 숫자를 지정하고 숫자가 클수록 순서가 밀린다.

2. flex-grow

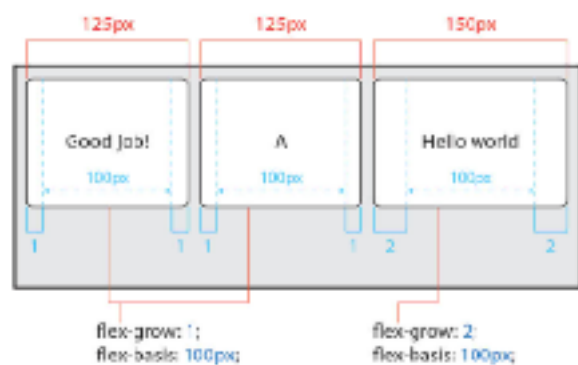
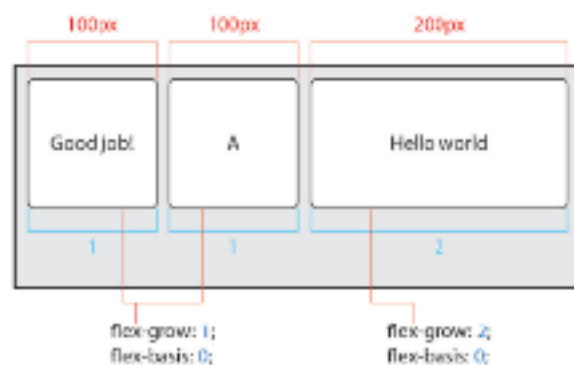
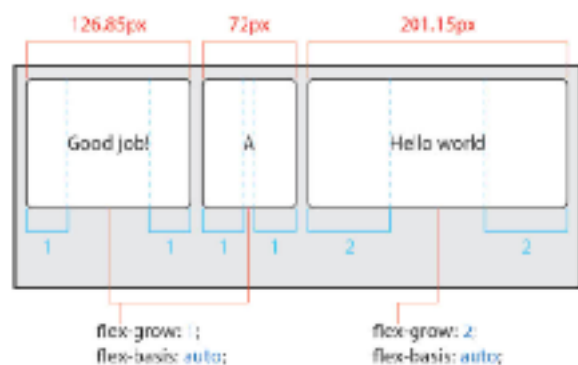
- Item의 증가 너비 비율을 설정한다.
- 숫자가 크면 더 많은 너비를 갖는다.
- Item이 가변 너비가 아니거나, 값이 0 일 경우 효과가 없다.
- 예를 들어 Item이 3개이고 증가 너비가 각각 1, 2, 1 이라면,
첫 번째 Item은 총 너비의 25%(1/4)을,
두 번째 Item은 총 너비의 50%(2/4)를,
세 번째 Item은 총 너비의 25%(1/4)을 가지게 된다.

3. flex-shrink

- Item이 감소하는 너비의 비율을 설정한다.
- 숫자가 크면 더 많은 너비가 감소한다
- Item이 가변 너비가 아니거나, 값이 0 일 경우 효과가 없다.

4. flex-basis

- Item의 (공간 배분 전) 기본 너비를 설정한다.
 - 값이 auto 일 경우 width, height 등의 속성으로 Item의 너비를 설정할 수 있다.
하지만 단위 값이 주어질 경우 설정할 수 없다.
- ※ flex 속성에서 설명한 것 같이 단축 속성 내에서 flex-basis 를 생략하면 값이 0 이 된다.



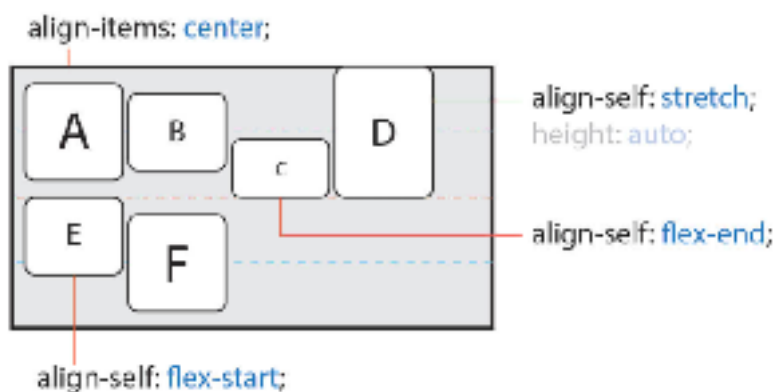
5. align-self

- 교차 축(cross-axis)에서 개별 Item의 정렬 방법을 설정한다.
- align-items는 Container 내 모든 Items의 정렬 방법을 설정한다.

필요에 의해 일부 Item만 정렬 방법을 변경하려고 할 경우 align-self 를 사용할 수 있다.

이 속성은 align-items 속성보다 우선한다.

속성 값	설 명
auto (기본값)	Container 의 align-items 속성을 상속 받음
stretch	flex container 의 교차축을 채우기 위해 items 를 늘린다.
flex-start	items 를 각 줄의 시작점(flex-start)으로 정렬한다.
flex-end	items 를 각 줄의 끝점(flex-end)으로 정렬한다.
center	items 를 가운데 정렬한다.
baseline	items 를 문자 기준선에 정렬



● float 속성

· float 라는 단어는 원래 ‘뜨다’ 라는 의미이며, 원래 웹 페이지에서 이미지를 어떻게 띄워서 텍스트와 함께 배치할 것인가에 대한 용도로 등장했다.

1. left

2. right

3. none

```
<style>
  .content > img { float: left; }
</style>
```

· left와 right를 통해 float 속성을 지정하는 경우 display 속성은 무시된다.

(단, none은 제외) 또한 이후 요소에 clear 속성이 있으면 페이지 흐름이 달라진다.

레이아웃에서의 float

· 요즘에는 레이아웃용으로 많이 사용하고 있다.

· <div> 태그를 float 속성 및 width, height 속성 등을 이용하여 레이아웃을 배치하는 예제

clear

- float는 오른쪽, 왼쪽으로 HTML 요소를 띄워서 정렬하는 속성이다.
- 이미지에 float 속성을 적용하면 이후에 오는 글자나 다른 요소들은 그 이미지 주위로 둘러싼 형태가 된다.
- 이미지 주위로 글자나 요소가 따라 붙지 않게 하려면 clear 속성을 사용해야 한다.

1. clear: none 는 기초값으로 clear를 설정하지 않은 것과 같다.
2. clear: left - 왼쪽을 취소
3. clear: right - 오른쪽을 취소
4. clear: both - 오른쪽 왼쪽을 취소

텍스트를 표현하는 스타일

● Font (글꼴)

font	모든 font 속성을 이용한 스타일을 한 줄에 설정할 수 있다. <code>#text4 { font: italic bold 20px serif; }</code>
font-family	텍스트의 글꼴을 설정한다. 1. generic family 비슷한 모양을 가지는 글꼴 집합 ("Serif", "Monospace" 등) 2. font family 특정 글꼴 집합 ("Times", "Courier" 등) <code>.serif {font-family: "Times New Roman", Times, serif; }</code>
font-style	주로 이탤릭체를 사용하기 위해 사용된다. <code>.italic { font-style: italic; }</code>
font-variant	텍스트에 포함된 영문자 중 소문자만을 작은 대문자(small-caps) 로 변경한다. <code>.smallCaps { font-variant: small-caps; }</code>
font-weight	텍스트를 얼마나 두껍게 표현할지를 설정한다.
font-size	텍스트의 크기를 설정한다. 1. 절대 크기 : px, pt 2. 상대 크기 : em, rem : 반응형 웹 폰트 단위로 사용한다.

1. rem 단위

HTML 태그의 폰트 크기를 기준으로 배수에 해당하는 크기를 가지게 하기 위해서 사용하는 단위이다.

```
<p style='font-size: 2em;'>
```

```
// 16px (HTML 태그의 폰트 크기) * 2(em) = 32px
```

HTML 태그의 폰트 크기를 변경한 경우

```
html { font-size: 20px; }
```

```
<p style='font-size: 2em;'>
```

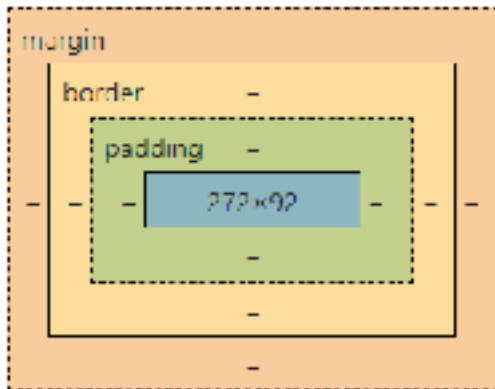
```
// 20px (HTML 태그의 폰트 크기) * 2(em) = 40px
```

2. em 단위

부모 태그의 폰트 크기를 기준으로 배수에 해당하는 크기를 가지게 하기 위해서 사용하는 단위이다.

박스 모델 (box model)

- 모든 HTML 요소는 박스(box) 모양으로 구성되며, 이것을 박스 모델(box model)이라고 한다.
- 박스 모델은 HTML 요소를 패딩(padding), 테두리(border), 마진(margin), 그리고 내용(content)으로 구분한다.



1. 내용(content) : 텍스트나 이미지가 들어있는 박스의 실질적인 내용 부분이다.
2. 패딩(padding) : 내용과 테두리(border) 사이의 간격이다.
3. 테두리(border) : 내용과 패딩 주변을 감싸는 테두리이다.
4. 마진(margin) : 테두리와 이웃하는 요소 사이의 간격이다.

● height와 width 속성의 이해

· 모든 웹 브라우저에서 HTML 요소들을 표현하려면 박스 모델이 어떻게 동작하는지 정확히 알아야 한다.

· CSS에서 height와 width 속성으로 설정된 HTML 요소의 높이과 너비는 내용(Content)의 크기만 포함되고 패딩(padding), 테두리(border), 마진(margin)의 크기는 포함되지 않는다.

· HTML 요소의 전체 너비(width)를 계산하는 공식은 다음과 같다.

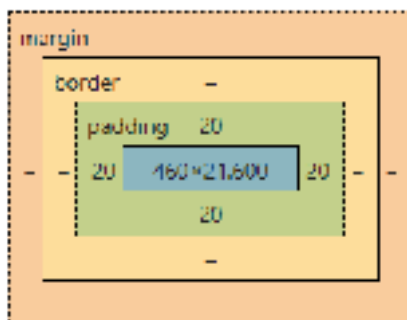
$$\text{width} + \text{left padding} + \text{right padding} + \text{left border} + \text{right border} + \text{left margin} + \text{right margin}$$

· HTML 요소의 전체 높이(height)를 계산하는 공식은 다음과 같다.

$$\text{height} + \text{top padding} + \text{bottom padding} + \text{top border} + \text{bottom border} + \text{top margin} + \text{bottom margin}$$


```
.box {
  /* box-sizing: border-box; */
  width: 500px;
  padding: 20px;
}
```

- 위 스타일에서 사용자가 기대하는 것은 넓이가 500px인 요소이지만 실제로는 패딩 간격이 적용되어 넓이가 540px인 요소가 표시된다.
- **box-sizing** 속성을 이용하면 패딩, 보더 등의 속성값을 요소의 넓이에 반영하지 않는다. 따라서 위의 코드 결과 넓이가 540px인 요소가 아니라 500px인 요소가 표시된다. 물론 padding 속성의 값은 넓이가 500px 인 요소 내부에 적용된다.



// width(460px) + left padding(20px) + right padding(20px) = 500px

● 테두리(border)

- border 속성은 내용, 패딩 영역을 둘러싸는 테두리의 스타일을 설정한다.
- border-style
- border-width
- border-color
- border

border-style 속성

- 테두리(border)를 다양한 모양으로 설정할 수 있다.

```
border-top-style: dotted;  
border-right-style: dashed;  
border-bottom-style: solid;  
border-left-style: ridge;
```

dotted : 테두리를 점선으로 설정함.

dashed : 테두리를 약간 긴 점선으로 설정함.

solid : 테두리를 실선으로 설정함.

double : 테두리를 이중 실선으로 설정함.

groove : 테두리를 3차원인 입체적인 선으로 설정하며, border-color 속성값에 영향을 받음.

ridge : 테두리를 3차원인 능선효과가 있는 선으로 설정하며, border-color 속성값에 영향을 받음.

inset : 테두리를 3차원인 내지로 끼운 선으로 설정하며, border-color 속성값에 영향을 받음.

outset : 테두리를 3차원인 외지로 끼운 선으로 설정하며, border-color 속성값에 영향을 받음.

none : 테두리를 없앴.

hidden : 테두리가 존재하기는 하지만 표현되지는 않음.

border-width 속성

· border-width 속성은 테두리(border)의 두께를 설정한다.

border-color 속성

· border-color 속성은 테두리(border)의 색상을 설정합니다.

테두리(border)의 개별 설정

· CSS를 사용하면 테두리의 위쪽, 오른쪽, 아래쪽, 왼쪽 부분에 대하여 개별적으로 스타일을 적용할 수 있다.

```
// 4개의 border-style 속성값을 가질 때는 top, right, bottom, left 순으로 설정한다.
```

```
border-style: dotted dashed solid double;
```

```
// 3개의 border-style 속성값을 가질 때는 top, right와 left, bottom 순으로 설정한다.
```

```
border-style: dotted dashed solid;
```

```
// 2개의 border-style 속성값을 가질 때는 top과 bottom, right와 left 순으로 설정한다.
```

```
border-style: dotted dashed;
```

```
// 1개의 border-style 속성값을 가질 때는 모든 테두리의 스타일을 같게 설정한다.
```

```
border-style: dotted ;
```

테두리 축약 표현(border shorthand)

```
<style>
```

```
  .box { border: 3px solid blue; }
```

```
</style>
```

● Margin

· margin 속성은 테두리(border)와 이웃하는 요소 사이의 간격의 크기를 설정한다.

이러한 마진 영역은 패딩 영역과는 달리 background-color 속성으로 설정하는 배경색의 영향을 받지 않는다.

마진(margin) 속성은 다음과 같다.

1. margin-top
2. margin-right
3. margin-bottom
4. margin-left

```
margin-top: 10px;  
margin-right: 5px;  
margin-bottom: 20px;  
margin-left: 30px;
```

마진 축약 표현(margin shorthand)

```
margin: 20px 50px 30px 50px; // top, right, bottom, left
```

※ margin 속성값에 auto를 사용하는 이유

· margin 속성값을 auto로 설정하면, 웹 브라우저가 수평 방향 마진(margin) 값을 자동으로 설정한다. 즉, 해당 HTML 요소의 왼쪽과 오른쪽 마진을 자동으로 설정하게 된다.

그 결과 해당 요소는 그 요소를 포함하고 있는 부모(parent) 요소의 정중앙에 위치하게 된다.