

UI

0.들여가기 전에..

웹의 아버지

팀 버너스 리

팀 버너스 리 경(sir. Tim Berners-Lee). 영국의 컴퓨터 과학자
WWW, URL, HTTP, HTML 설계 등.

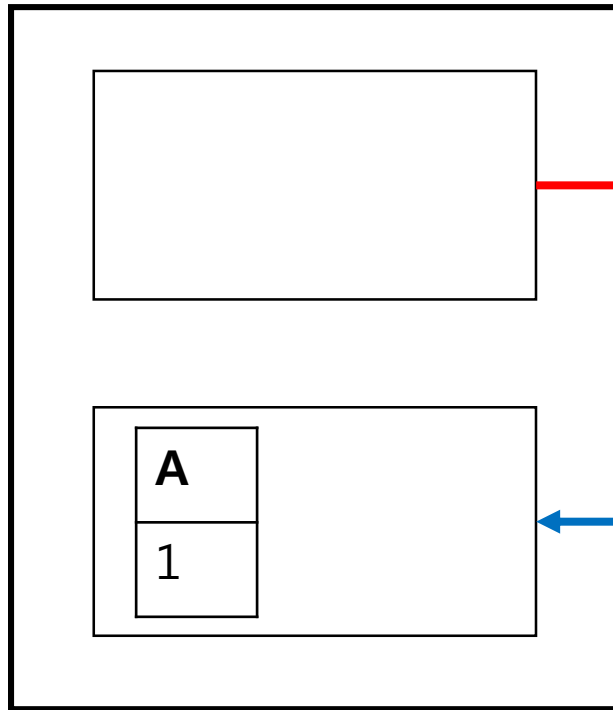
[강연 링크](#)



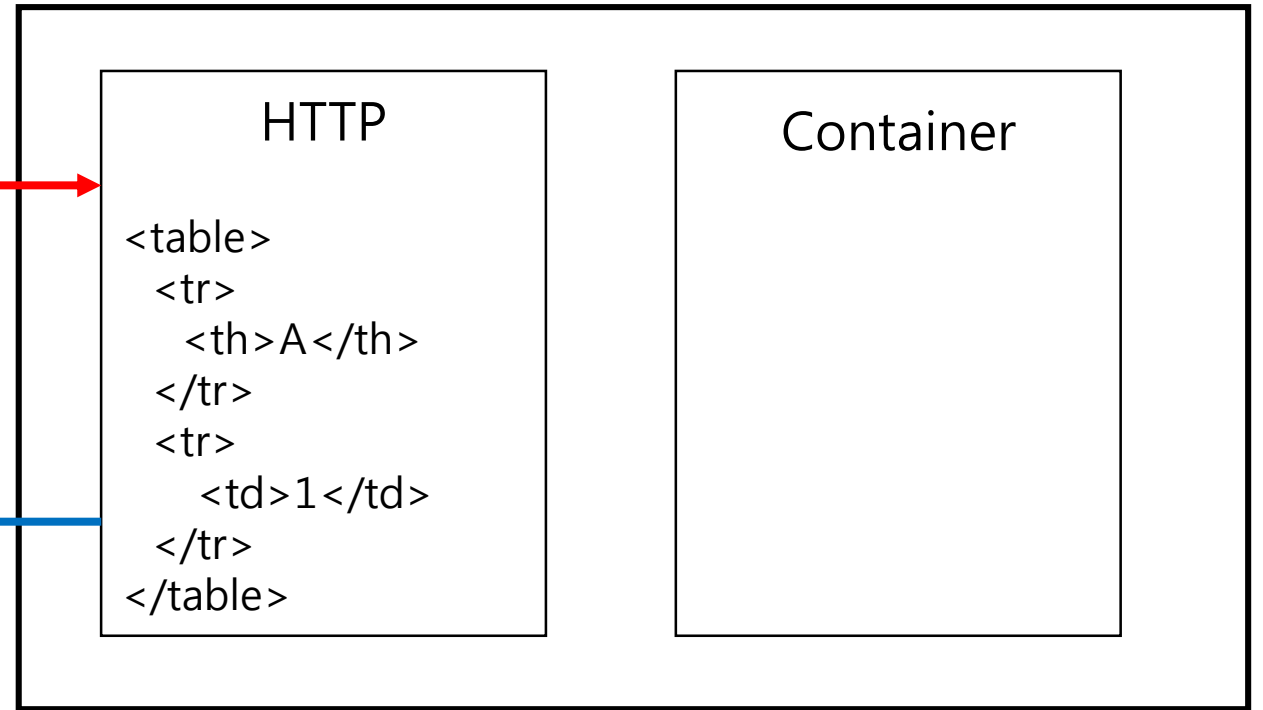
0.들어가기 전에..

기본구조

클라이언트



서버



request

response

0.들어가기 전에..

기본구조

Client가 Server에 요청(request) 하면

HTTP(통신 규약)에 의해

HTML 문서를, 정해진 모양으로 응답(response) 한다.

* Client가 보는 화면(view)은 서버가 보내준 모양대로 나타나게 된다.

1.HTML

구조

HTML : HyperText Markup Language

- 온라인 상의 문서(page)를 만들기 위한 구조화 된 언어.

```
<!DOCTYPE html>
```

문서의 타입을 정의

```
<html>
```

이 문서는 html 이다.

```
<head>
```

문서를 정의

```
<title>Hello, World</title>
```

```
</head>
```

```
<body>
```

문서의 내용

```
<p>Hello, Wolrd! Hello, Web!</p>
```

```
</body>
```

```
</html>
```

1.HTML

구조

HTML 요소(element, tag)의 구성

<tag> </tag>

<div> </div>

<tag 속성 = '값'> </tag>

<div align='center'> </div>

<tag 속성 = '값'>문자</tag>

<p style='color:red;'>Hello, World!</p>

1.HTML

head

문서를 정의(<head>)하는 기본 태그

- <meta> : 문서의 기본 정보 등을 정의
- <title> : 문서의 제목을 정의
- <script> : javascript 등을 정의
- <style> : css 등을 정의
- <link> : 외부 문서를 연결할 때 정의

1.HTML

DTD

DTD(Document Type Definition) 선언

strict : 선언된 html 버전의 문법과 구조를 정확하게 사용

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

transitional : 선언된 html 버전 이외의 문법과 구조를 허용

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

frameset : transitional + frame 지원 (사실상 transitional과 동일)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```


1.HTML

DTD

DTD 해석해보자.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

이 문서는 **html**로 작성하고, **국제공용문서**이며,
W3C기관에 문서형식에 의해 HTML4.01을 Frameset방식의 영어로 출력하고,
참조할 DTD문서는 "<http://www.w3.org/TR/html4/frameset.dtd>" 이다.

1.HTML

문법

1. 요소를 정확하게 매칭
2. 요소, 속성 이름은 소문자
3. 요소는 항상 닫아야 한다 (일부 단일 태그 존재 ex)
4. 특수문자를 쓸 때는 Entity Name(code) 으로 사용

1.HTML

문법

블록요소, 인라인 요소

유 형	특 징
블록 요소	줄 바꿈 블록 요소 안에 텍스트와 인라인 요소 포함 가능 블록 요소 안에 블록 요소 포함 가능 (일부 불가)
인라인 요소	줄 바꿈 x 인라인 요소 안에 텍스트와 인라인 요소 포함 가능 인라인 요소 안에 블록 요소 포함 불가

1.HTML

문법

블록 요소

address	article	aside	audio	blockquote	canvas	dd	div	dl	fieldset	figcaption
figure	footer	form	h1	h2	h3	h4	h5	h6	header	hgroup
hr	noscript	ol	output	p	pre	section	table	tfoot	ul	video

인라인 요소

a	abbr	acronym	b	bdo	br	button	cite	code	dfn
em	i	img	input	kbd	label	map	object	q	samp
small	script	select	span	strong	sub	sup	textarea	tt	var

1.HTML

문법

제목태그 : <h1> ~ <h6> (총 6개)

- 문서 내부의 컨텐츠 제목을 정의
- 순서에 맞게 작성
- css로 임의의 크기 지정 가능
- h1은 한 문서에 한번만 지정하는 것을 권장

1.HTML

문법

단락(paragraph)

<p>

- 블록 요소 (다른 블록 요소 포함 불가)

- 문단 간격 정의 태그. 개행요소. 빈 요소(종료 태그 X)

1.HTML

문법

주소

<address>

- 연락처 정보 나타내는 요소
- 블록요소 (다른 블록요소 포함 불가)
단, DTD가 'transitional' 인 경우 포함 가능

1.HTML

문법

구분선 (Horizontal line)

`<hr>`

- 블록요소이지만, 빈 요소이기에 종료 태그 없다.
- css 대신 내용 구분할 때 주로 사용

1.HTML

문법

인용문(Quotation)

<blockquote>

- 다른 사이트에서 인용한 일부를 지정
- 블록요소인 p요소 사용
- cite 속성 : 인용문의 출처, cite 요소 : 작품의 제목

<q>

- 짧은 인용문 지정
- 텍스트나 인라인 요소 포함

1.HTML

문법

텍스트 표현 요소

 : 텍스트 진하게

 : 텍스트 진하게

 : 텍스트 기울임

<i> : 텍스트 기울임

<small> : 작은 텍스트 표시, 코멘트 등

<sub> : 아래 첨자

<sup> : 윗첨자

<ins> : 내용 추가

 : 내용 삭제

1.HTML

문법

이미지

- 주요 속성

alt : 이미지를 설명하는 내용 지정. 이미지가 보이지 않을 경우, 이미지 설명 텍스트

width : 이미지의 가로 크기 지정

height : 이미지의 세로 크기 지정

src : 이미지 경로 지정

1.HTML

문법

링크

<a>

- 텍스트나 이미지 등을 통해 다른 페이지로 정보 연결
- 인라인 요소(인라인 요소와 텍스트를 포함)
- 주요 속성

href : 링크의 주소 지정

title : 링크의 설명 표시

target : 링크된 문서를 어떤 창에서 오픈할 것인지 지정

1.HTML

문법

기본 목록 구성

- 주로 메뉴 작성시 사용

 : 비 순차적 목록 나열

 : 순차적 목록 나열

 : 목록의 내용

1.HTML

문법

테이블

<table>

- 문서에 표를 작성할 때 사용
- 주요 속성

rowspan : 수직방향으로 셀 병합

colspan : 수평방향으로 셀 병합

1.HTML

문법

테이블 주요 구성 태그

<tr> : 행 정의. th와 td만 자식요소로 포함

<th> : 제목 셀 정의

<td> : 데이터 셀 정의

<thead> : 테이블의 머리글

<tbody> : 테이블의 본문

<tfoot> : 테이블의 바닥글

<caption> : 테이블 제목

<col> : 테이블 열 정의

<colgroup> : 테이블의 열 그룹화

1.HTML

문법

<form>

- 폼의 최상위 요소. 폼을 구성할 때 정의
- 주요 속성 및 태그

action : 폼 서식에 작성한 값들을 처리할 서버 프로그램 주소

method : 작성된 값들의 전송 방식 설정 (get, post)

<fieldset> : 여러 폼 요소들의 그룹화 범위 지정

<legend> : 그룹화 제목 표시

<input> : 폼 안에 기본적인 컨트롤 생성

<textarea> : 여러줄로 된 텍스트 필드 생성.

<select>, <option> : 선택 목록 정의, 목록 표시

<button> : <input type="button">과 기능적으로 같지만, 보다 유연하게 표현 가능

1.HTML

문법

`<input type="">` 요소의 type 값

text : 일반 텍스트

password : 비밀번호 (*로 표시)

checkbox : 체크박스 (중복 가능)

radio : 라디오버튼 (중복 불가)

submit : 전송 버튼

reset : 폼 안의 입력요소 값 초기화

button : 기본 버튼

image : 이미지 버튼

file : 파일 선택

hidden : 화면에 표시되지 않음 (기능은 정상 작동)

1.HTML

문법

`<input type="">` 요소의 속성

name : 컨트롤의 이름 설정 (서버와의 데이터처리 작업 시, 반드시 필요)

value : 컨트롤의 값

size : 컨트롤의 가로 크기

maxlength : 최대 입력 문자수 지정

checked : 체크박스, 라디오버튼의 초기 선택 지정

disabled : 포커스, 선택, 변경 불가

readonly : 입력내용 변경 불가, 데이터 전송 가능

1.HTML

문법

<textarea> 속성

- name, disabled, readonly : input 요소 속성과 동일
- rows : 표시 줄 수 지정 (브라우저에 보이는 줄 수 의미)
- cols : 표시 폭 지정 (브라우저의 설정에 따라 차이가 있을 수 있음)

1.HTML

문법

grouping

- 여러 요소를 그룹화. css 적용시 많이 사용

<div> : 블록 요소

 : 인라인 요소

1.HTML

html5

새로운 구조

<hgroup> : 제목과 부제목을 묶어주는 요소. 최상위 제목만 outline 된다.

<section> : 의미가 같은 내용들을 그룹화

*<h>, <p>, <section>, <article>을 묶을 때나, 독립된 영역을 만들 때도 사용

<header> : 문서의 머릿말

<footer> : 문서의 꼬릿말

<nav> : 메인 네비게이션

<article> : 문서에서 독립적인 콘텐츠

<aside> : 사이드바 (메인 콘텐츠와 분리된, 독립적 요소)

1.HTML

html5

<header>

<nav>

</nav>

</header>

<section>

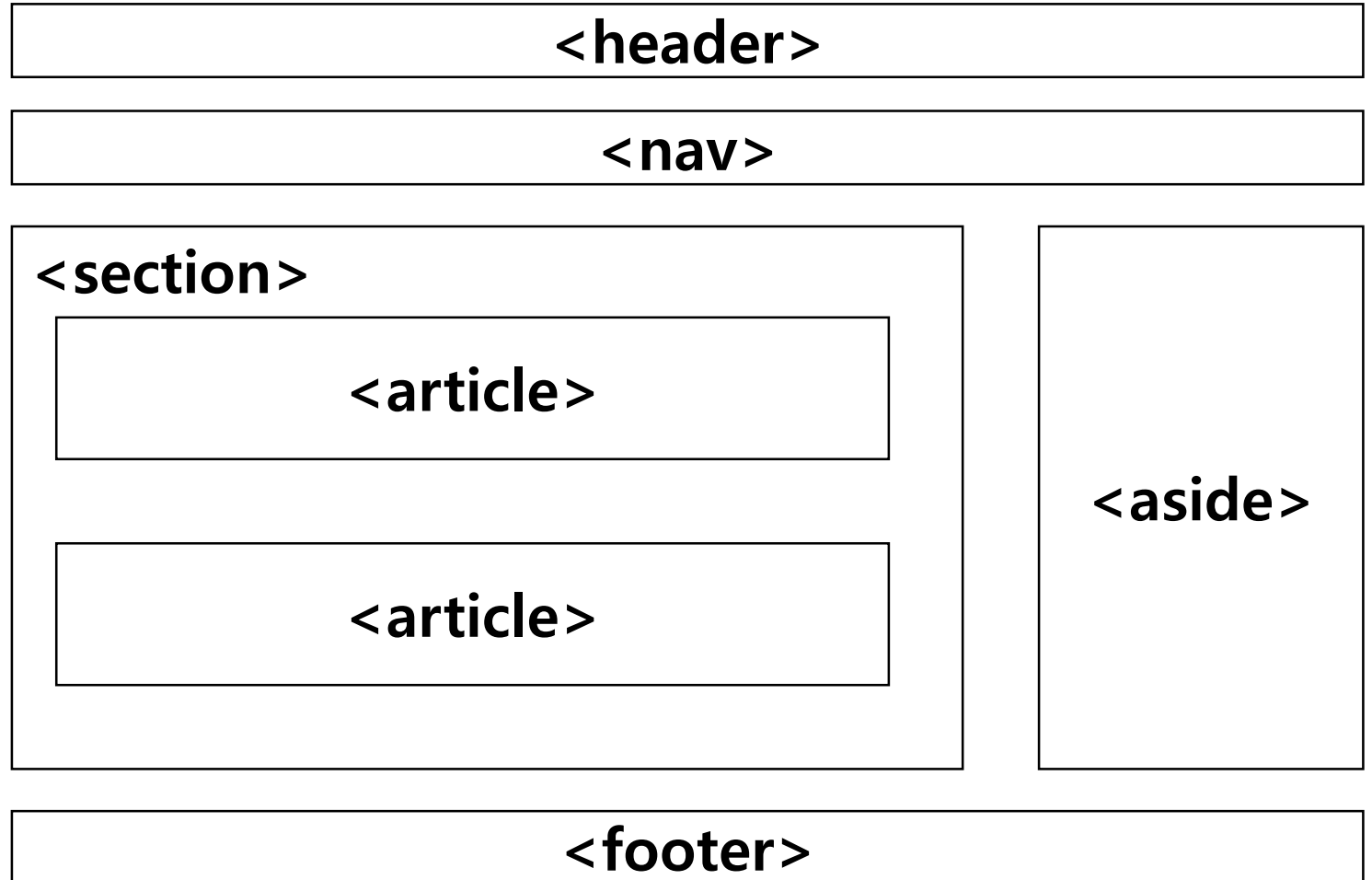
<article> </article>

<article> </article>

</section>

<aside> </aside>

<footer> </footer>



1.HTML

html5

추가된 요소

<figure> : 삽화, 다이어그램, 사진 코드목록 등에 설명을 달아줌

<mark> : 부분 강조

<time> : 날짜와 시간을 기계가 이해할 수 있는 형태로 (date type으로)

<details> : 접기/펼치기 기능 (summary : details의 제목)

<progress> : 다운로드 진행상황

<meter> : 현재 진행 상황

1.HTML

html5

<form> 에 추가된 속성

<input type=""> : search, email, url, tel, number, range, date, datetime, datetime-local, month, week, time, color

required : 유효성 검사

placeholder : 입력 형식 힌트

autocomplete : 자동완성

list : 자동완성과 비슷한 기능 (미리 만들어놓은 리스트 보여줌)

autofocus : 커서 자동이동

pattern : 정규 표현식 방법으로 유효성 검사

multiple : 다중 파일 선택

2.CSS

css?

CSS (Cascading Style Sheets)

-문서의 스타일을 지정해주는 언

* 문서(document) : markup language (html, xml, ...)

```
<li> <font color="red">HTML</font> </li>
```

```
<li> <font color="red">CSS</font> </li>
```

 : 정보 태그

 : 스타일(시각적인 효과) 태그

=> 정보와 디자인이 혼합되어 있어서, 보기에 좋지 않다.

2.CSS

표현방법

인라인 : 태그 안에서

<태그 style="속성='값';">

내부 : head 안에서

<style>...</style>

외부 : head 안에서 .css 파일 연결

<link href="파일.css" rel="stylesheet" type="text/css">

*순서 : 절차식 (마지막에 선언된 스타일이 우선순위를 가진다.)

2.CSS

선택자

selector : 스타일을 적용시킬 범위를 선택하기 위한 표현식

1. 타입 선택자
2. id 선택자
3. class 선택자
4. 전체 선택자
5. 자식 선택자
6. 인접 선택자
7. 속성 선택자
8. 가상 클래스 선택자
9. 종속 선택자
10. 하위 선택자
11. 그룹 선택자

2.CSS

선택자

타입 선택자

- 태그 이름을 지정하여 선언
- 동일한 속성에 대한 여러 요소를 콤마(,) 로 구분하여 선언 가능

```
<style type="text/css">
```

```
h1{font-size:10px;}
```

```
h1,h2,h3{color:blue;}
```

```
ul{margin:0px;}
```

```
</style>
```

2.CSS

선택자

id 선택자

- 요소에 id를 지정하고, style에서 지정된 id값으로 사용
- #으로 구분한다.
- '유일한 하나' 인 경우에 사용하는 것이 좋다.

```
<style type="text/css">
```

```
  #header{margin:0px;}
```

```
</style>
```

```
...
```

```
<div id="header"> </div>
```

2.CSS

선택자

class 선택자

- 요소에 class를 지정하고, style에서 지정된 class 값으로 사용
- .(dot)으로 구분한다.
- 한번에 여러 개를 바꿀 때 사용하는 것이 좋다.

```
<style type="text/css">
```

```
  .content{font-size:10px;}
```

```
</style>
```

```
<div class="content">1</div>
```

```
<div class="content">2</div>
```

2.CSS

선택자

전체 선택자

- *를 사용한다. (자주 사용하지 않음)

```
<style type="text/css">  
  *{font-size:10px;}  
</style>
```

2.CSS

선택자

자식 선택자

- 부모/자식 관계에서 자식 요소를 지정한다.

```
<style type="text/css">
```

```
  div > p {font-size:10px;color:blue;}
```

```
  p > span {color:red;}
```

```
</style>
```

...

```
<div> <p>자식선택자</p> </div>
```

```
<p> <span>나도 자식인가</span> </p>
```


2.CSS

선택자

인접 선택자

- 지정한 요소 다음에 나오는 요소를 선택한다.

```
<style type="text/css">
```

```
  h1 + span {color:red;}
```

```
</style>
```

...

```
<h1>인접<span>했나요?</span></h1>
```

```
<span>인접했나요!</span>
```

2.CSS

선택자

속성 선택자

- 속성이 정의된 태그만 선택

```
<style type="text/css">
```

```
  h1[class] {color:blue;}
```

```
</style>
```

...

```
<h1 class="abc">속성(class)있음</h1>
```

```
<h1>속성(class) 없음</h1>
```

2.CSS

선택자

속성 선택자

1. [attribute] : attribute라는 속성을 가지고 있는 요소들을 선택
2. [attribute=value] : 속성값과 일치하는 요소들을 선택
3. [attribute~=value] : 속성값과 일치하는 요소들을 선택, 공백으로 구분하여 포함된 속성값도 선택
4. [attribute |= value] : 속성값으로 시작하는 요소들을 선택,
하이픈(-)으로 구분하여 포함된 속성값도 선택
5. [attribute ^= value] : 속성값으로 시작하는 요소들을 선택
6. [attribute \$= value] : 속성값으로 끝나는 요소들을 선택
7. [attribute *= value] : 속성값을 포함하고 있으면 선택 (위치 상관 없음)

2.CSS

선택자

가상 클래스 선택자

- 요소의 상태에 따라 선택자 지정
- 상태가 html 문서에 표현되는 것은 아님
- css에서만 구별

```
<style type="text/css">
```

```
  a:link{color:blue;text-decoration:none;} /*방문하지 않은 링크*/
```

```
  a:visited{color:red;text-decoration:none;} /*방문했던 링크*/
```

```
  a:hover,a:active{color:yellow;text-decoration:underline;} /*마우스 오버, 클릭 시*/
```

```
</style>
```

2.CSS

선택자

종속 선택자

- 타입, id, class 선택자가 결합된 형태

```
<style type="text/css">  
  p.sel a {font-size:10px;}  
  div #mul {color:red;}  
</style>
```

2.CSS

선택자

하위 선택자

- 특정 요소 하위의 요소를 지정할 때 사용
- 공백으로 작성

```
<style type="text/css">  
  h1 span {font-size:10px;}  
</style>
```

...

```
<h1> <span>abc</span>abc</h1>
```

2.CSS

선택자

그룹 선택자

- 여러 요소에 각각 같은 속성 적용
- ,(콤마)로 구분

```
<style type="text/css">
```

```
  p, li, a {margin:0px; color:blue;}
```

```
</style>
```

```
<p>abc</p>
```

```
<ul> <li>1</li> <li>2</li> <li>3</li> </ul>
```

```
<a href="#">abc</a>
```

2.CSS

선택자

선택자 우선순위

- 스타일 선언은 위에서 아래로 순차적으로 실행되면서
마지막에 선언된 스타일이 우선순위를 갖는다.

```
.test{color:red;}
```

```
.test{color:blue;}
```

```
.test{color:yellow;}
```

-> class test의 최종 폰트 색은 yellow;

2.CSS

선택자

선택자 우선순위

*특정 요소에 중복 선언할 경우

```
.test02{color:red;}
```

```
p{color:blue;}
```

```
#test01{color:yellow;}
```

...

```
<p id="test01" class="test02" style="color:red;">우선순위</p>
```

inline > id > class > type

2.CSS

선택자

선택자 우선순위

*강제 우선순위 선언

```
.test02{color:red;}
```

```
p{color:blue; !important;}
```

```
#test01{color:yellow;}
```

...

```
<p id="test01" class="test02" style="color:red;">우선순위</p>
```

!important > inline > id > class > type

2.CSS

서체

font

- 글자와 관련된 스타일을 지정할 수 있는 속성

font-family : 글꼴 지정

font-size : 크기 지정

font-weight : 굵기 지정

font-style : 글씨체 지정

font-variant : 소문자->대문자 변환 (일반 대문자보다 작다)

font-height : 줄 간격 지정

font : 위 속성을 한 줄에 모두 지정(font-size/font-family 생략 불가)

2.CSS

문단

paragraph

- 문단을 중심으로 정렬, 들여쓰기 등을 지정

text-align : 글자의 정렬 기준 지정 (justify : 양쪽정렬)

vertical-align : 이미지나 폼 요소를 세로 정렬

text-indent : 들여쓰기

text-transform : 대소문자 변경 (capitalize : 첫 글자만 대문자)

text-decoration : 밑줄 등(밑줄, 가운데 줄, 윗줄, 밑줄 제거)

letter-spacing : 글자와 글자간의 간격

2.CSS

배경

background

background-color : 배경 색 지정

background-repeat : 배경 이미지 지정

background-position : 배경 이미지 반복 지정

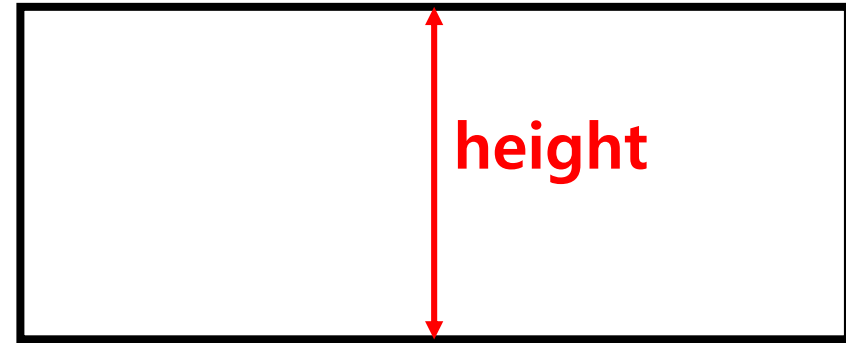
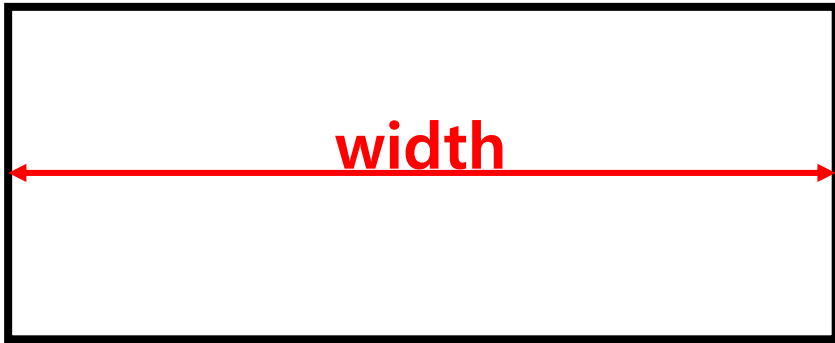
background-attachment : 배경 이미지의 고정 여부

background : 관련 속성을 한 줄로 표기

2.CSS

상자

width, height : px, %, em 등으로 수치 표기

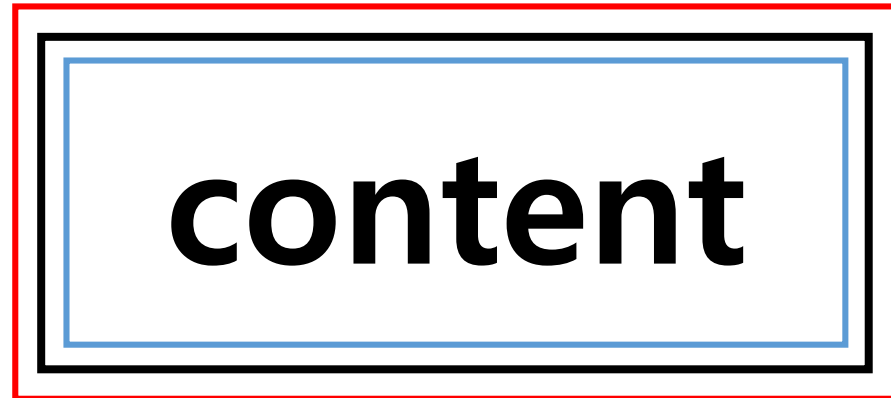


2.CSS

상자

margin(바깥여백), border(테두리), padding(안쪽여백)

margin
boarder
padding



2.CSS

상자

margin 병합

- 세로 방향 마진을 지정한 두 개의 서로 다른 요소가 수직 방향으로 접해있을 때 두 요소 사이의 margin 간격은 큰 쪽으로 병합

Margin: 20px;

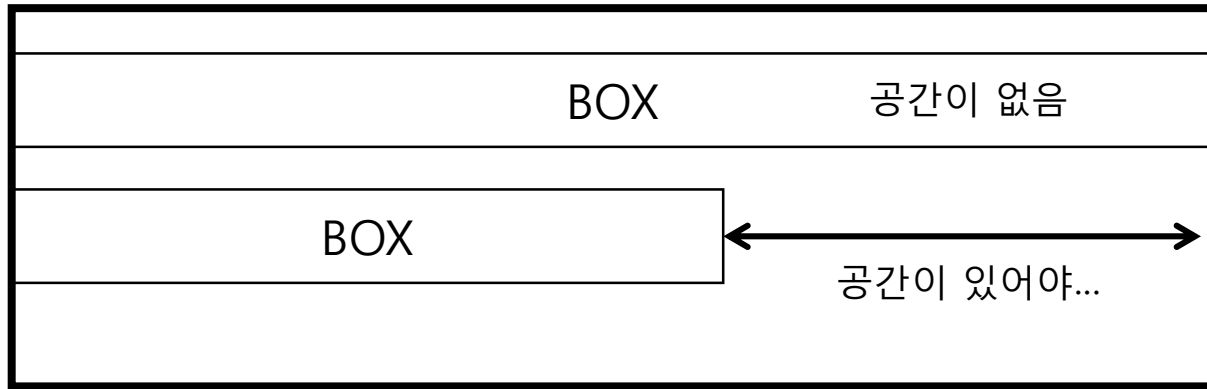


2.CSS

상자

margin을 이용한 중앙 정렬

- 박스의 좌우 속성값을 auto로 지정해서 중앙에 위치.
(정렬하려는 박스의 width를 반드시 지정)



2.CSS

상자

layout : 블록들의 위치를 나열

- float : 요소를 블록형태로, 특정 방향으로 정렬
- clear : float을 해제할 때 사용
(이전 요소의 float값을 상속받아, 같은 방향으로 정렬되는 현상 방지)
- display : 요소의 성격을 바꿔주는 속성 (인라인 요소 <-> 블록 요소)
- overflow : 박스 안의 내용이 박스보다 클 경우, 넘치는 부분을 처리
(부모 사이즈가 지정되어 있어야 사용 가능)
- position : 블록박스의 위치를 지정하는 속성 (겹치게 배열 가능)

2.CSS

css3

<border>

border-radius : 박스의 테두리 선을 둥글게

box-shadow : 박스의 그림자 지정

border-image : 특정 이미지를 테두리의 배경으로 지정

<background>

gradient : 여러 색을 원형이나 선형으로 칠해주는 효과

background-size : 배경 이미지 크기 조절

background-origin : 배경 이미지 테두리 맞춤

multi-background : 배경 이미지 여러 개 지정

background-clip : origin과 비슷

2.CSS

css3

<text>

text-shadow : 텍스트에 그림자 지정

word-wrap : 자동 줄바꿈

font-face : 웹폰트 설정

<2d transform>

rotate : 회전

scale : 확대/축소

translate : 이동

skew : 찌그러뜨리기

2.CSS

css3

<transitions>

background, color, height, width, transformation 등의 속성들에 애니메이션 적용

<animation>

박스의 크기, 위치, 색상 등의 값을 여러 단계로 나눠 실행. (애니메이션 효과)

<다단 편집>

multi column 지원

<사용자 환경>

box-sizing : border가 박스 크기에 포함/미포함

resize : 박스의 크기 조절

outline : border를 감싸는 테두리

0.챕터

소제목

내용