

웹퍼블리싱 강의

2022. 6

강 현 준

human@human.or.kr

강의 목차

1. HTML
2. CSS
3. JAVA SCRIPT
4. JQuery
5. AJAX

1장. HTML

1.1. 웹퍼블리싱이란?

- 웹퍼블리싱이란? 기획자의 의도대로 실제 웹사이트를 구축하는 과정을 의미함

웹퍼블리싱의 진행과정

기획작업

로고	메뉴 - 교육센터안내 / 모집과정상담/교육과정 안내	카톡상담
개별 과정에 대한 안내 페이지. - 슬라이드로 시간간격별로 넘어가기.		
모집과정 안내 서울 / 수원 / 천안 별로 검색할 수 있도록		

[화면정의서]

[메뉴구조도]

상세디자인

[이미지파일]



완성



웹퍼블리싱



1.2. 웹과 인터넷의 관계

- 서로 떨어져있는 인트라넷간의 연결을 하는 것이 인터넷
- 인터넷을 통해서 문서 전송이 수월해짐. 이로써 웹의 활성화가 진행됨.

웹 ≠ 인터넷



[FTP 프로그램]

FTP
- 파일전송

SMTP
- 메일전송

TELNET
- 명령어전송

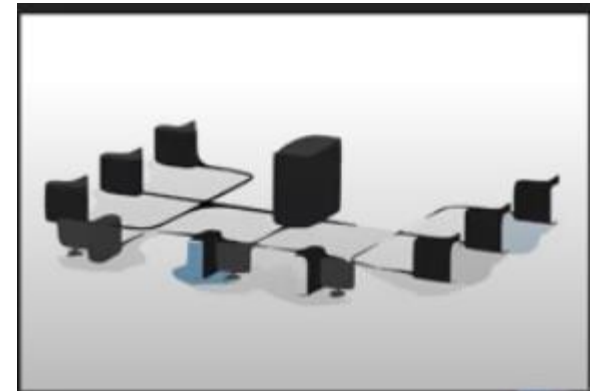


[웹브라우저]

WWW (웹)
- 문서 (HTTP)



[인트라넷 A]

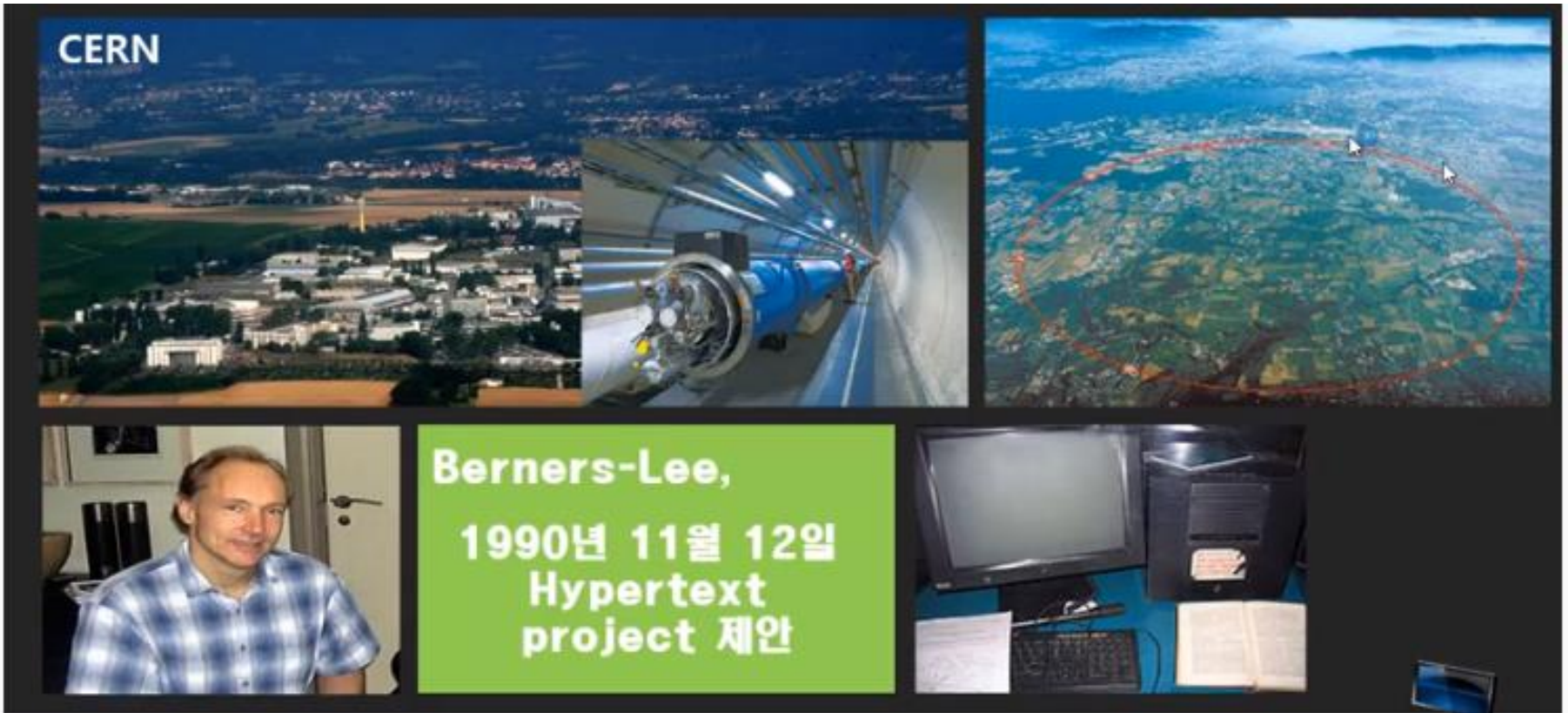


[인트라넷 B]

1.3. 웹의 탄생배경

- CERN이란 동네에서 입자가속관련 물리학자들이 모여살고 있음
- 연구결과를 서로 쉽게 공유하기 위해 Hyper Text Project 제안함.

HyperText Project 제안



- * HyperText : 연결구조까지 포함된 단위 텍스트. 클릭만으로 다른문서들을 손쉽게 찾아볼 수 있음
- * HTML (Hypertext Markup Language): 하이퍼텍스트를 Markup하는 프로그램 언어

1.4. Markup 언어란?

- 이미지, 목록, 문장 등이 포함된 문서를 구조화하는 언어
- 문서특징에 맞게 TAG를 사용하여 브라우저에서 표시 가능함.

Markup 언어란 문서의 콘텐츠를 구분하기 위해 Tag를 사용함

모집과정 및 신청접수

입학상담

온라인상담

모집과정

휴먼교육센터는 전문 재취업훈련 전문교육기관입니다.

이미지



목록

[모집중]자바(JAVA) 프로그래밍
전선 개발자 양성

기간 : 22년 6월 15일 개강
시간 : 09:00 ~ 17:50
인원 : 30
장소 : 2교육실
교육비 : 5,731,560(전액지원)
문의 : 041-561-1122

응용소프트웨어 개발자
전력산업직종 과정

리콜럼의 국가기간

입학상담

온라인상담

문장

디지털 기술을 기반으로 다양한 기기의 융합, 콘텐츠의 융합을 통해 새로운 형태의 제품이나 융합서비스를 창출하기 위하여 기기, 콘텐츠의 기획, 설계, 제작, 운용 및 시험을 수행하는 기능, 기술을 수행할 수 있는 4차산업혁명에 핵심적인 과정입니다.

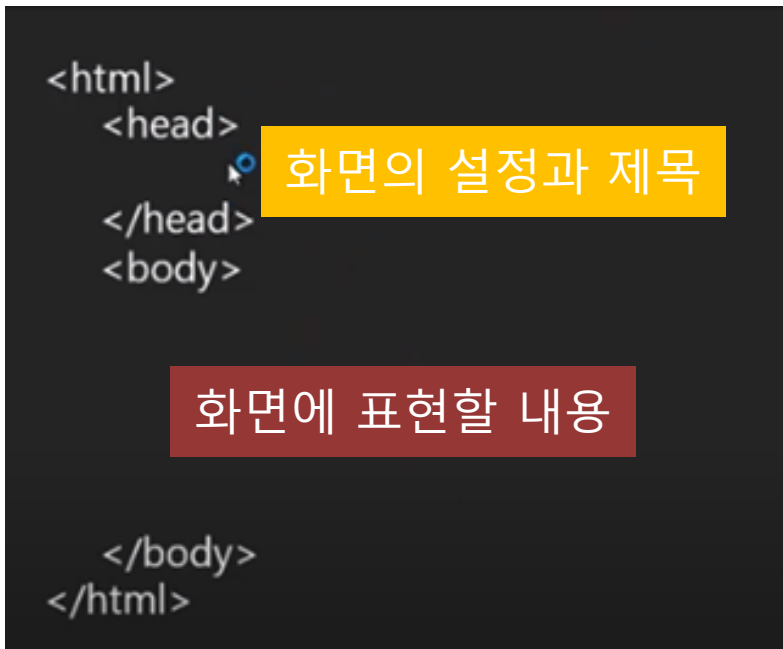
TAG 집합

h	p	ul	ol	dl	li
table	tr	td	addr	scope	br
font	strong	b	u	i	s
center	body	html	head	h1	div

1.5. HTML 문서의 기본구조

- <HTML>로 정의하고, <head> ➔ <body> 순으로 정의함.
- Tag의 시작과 끝을 정의하고, 이 기준에 의해 브라우저에서는 렌더링하여 화면에 표시됨

HTML 문서의 기본구조



1.6. HTML 문서의 활용

- 요즘은 PC외에도 다양한 기기들이 있으며, 이들을 지원함에 있어 한계를 극복하고 있음
- HTML5 시대에 와서 반응형 웹 기반으로 웹퍼블리싱이 이뤄지고 있음

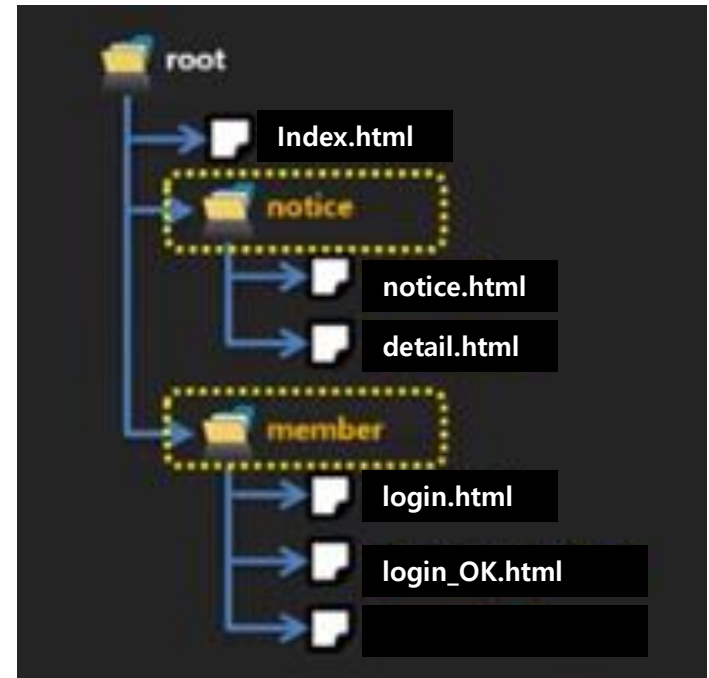
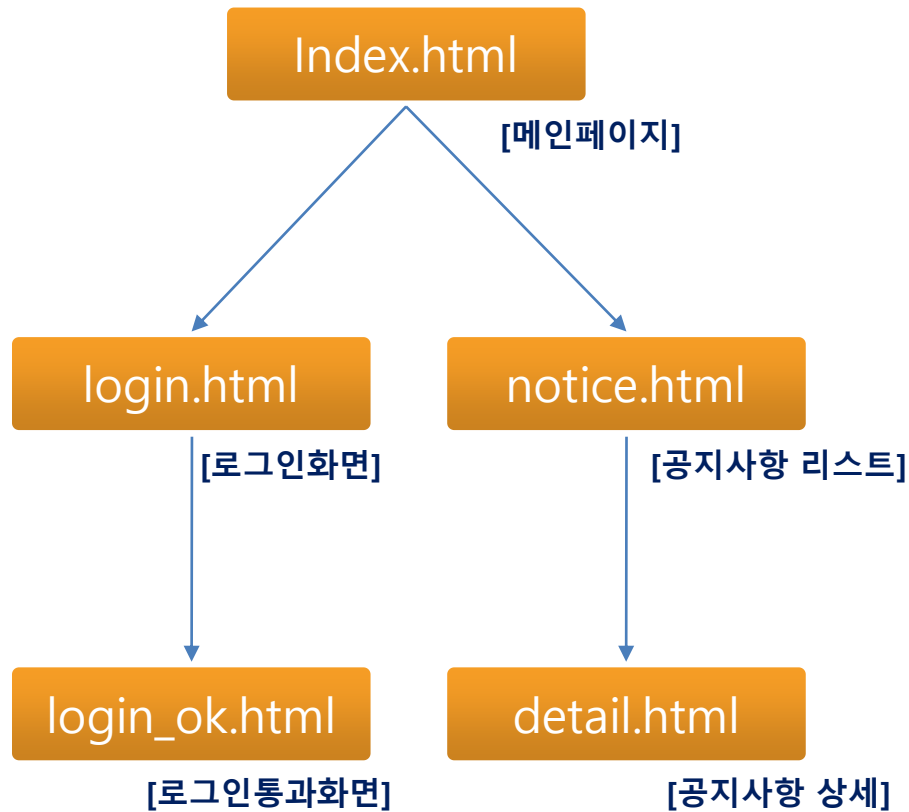
HTML 문서의 다양한 기기에서의 활용



1.7.1. 실습 - 1

- 메인페이지, 로그인페이지, 공지사항의 페이지 실습
- 폴더구조 및 페이지 파일은 아래와 같이 구조화 함.

실습

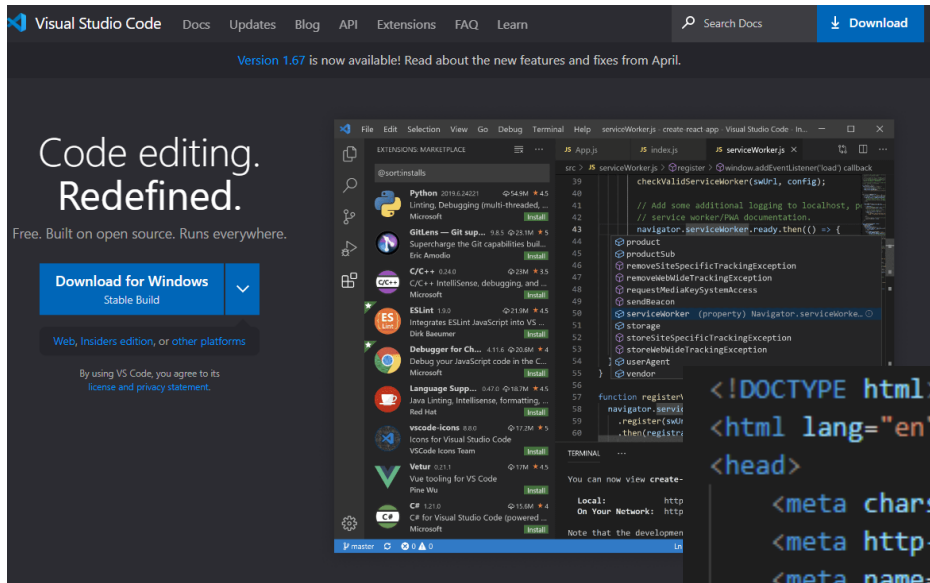


[폴더 구조 예시]

1.7.2. 실습 - 1

- 비주얼스튜디오코드 설치 및 라이브서버 확장팩 설치
- 상대경로와 절대경로에 대한 설명 (/ , ./ , ../)

상대경로 기반으로 각 페이지별로 이동이 되는 확인.



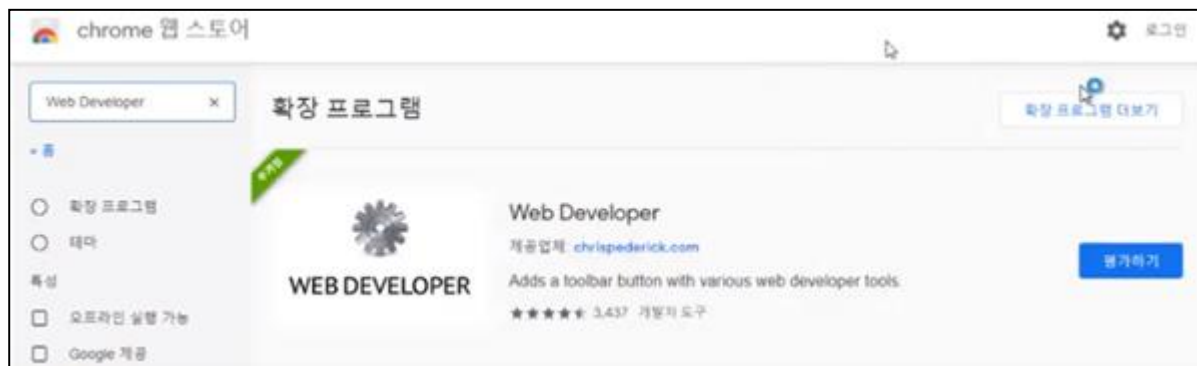
실습 : index.html
페이지간 이동이 되도록

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>리스트페이지</title>
</head>
<body>
  <a href="../index.html">홈</a>
  <a href="list.html">공지사항목록</a>
  <a href="../member/login.html">로그인</a>
  <a href="../member/signup-agree.html">회원가입</a>
  <h1>리스트 페이지</h1>
```

1.8.1. 웹문서 작성 1단계 - 콘텐츠 작성하기

- 웹문서를 작성하는 것은 우선적으로 콘텐츠를 작성하는 것임.
- 콘텐츠란 웹문서에 포함할 주요 주제임.

콘텐츠 예시 확인 방법



CSS 적용

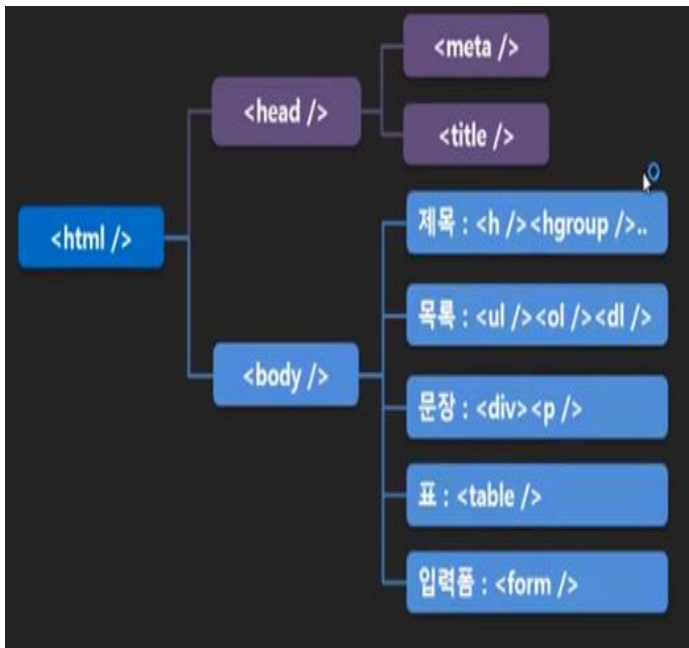


CSS 미적용
- 콘텐츠만 확인

1.8.2. 웹문서 작성 2단계 - 콘텐츠 블록의 종류

- HTML 문서에서 콘텐츠 문서에 대한 종류와 관련 TAG의 표준
- 제목, 목록, 문장, 표, 입력폼 등이 있으며, 해당 문서에는 사용되는 표준이 있다

문서별 사용되는 표준 TAG



UL

표 (table)

번호	제목	작성자	작성일	조회수
1028	소프트웨어 MVC 공지사항 관리자 view 페이지	newlec	2020-12-09	8472
1026	소프트웨어 MVC POST 한글 입력 컨트롤러 및 view 설정 코드	newlec	2020-12-18	880
1024	소프트웨어 MVC 소스코드	newlec	2020-12-12	6299
1023	소프트웨어 JSON 출력을 위한 라이브러리 설정	newlec	2020-11-29	2466
1022	소프트웨어 MVC RestController 한글 개입 및 코드	newlec	2020-11-26	2384
1020	소프트웨어 MVC 20강 Notice 서비스 페이지	newlec	2020-11-06	4274
1016	Twitter ViewResolver 설정 내용	newlec	2020-10-22	2361
10	html.zip 파일 (소프트웨어 MVC 강좌용)	newlec	2020-10-12	4212
10	모바일 수험생 페이지를 설정용 소스코드	newlec	2019-10-04	4108
9	JSP 수험생 html 파일	newlec	2019-10-03	7382

1.8.3. 웹문서 작성 2단계 - 콘텐츠 블록의 종류 (제목)

- 제목TAG : 6개로 구분되는 H1~H6으로써 제목의 Hierachy를 설정가능
- <Section> Tag를 통해서 동일 Section에서는 H1에서 시작하는 것이 좋음

제목 Tag

The diagram illustrates the relationship between HTML heading tags and their visual output in a web browser. On the left, a code editor shows the following HTML structure:

```
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

<body>
  <h1>Let's call it a draw(ing surface)</h1>
  <h2>Diving in</h2>
  <h2>Simple shapes</h2>
  <h2>Canvas coordinates</h2>
  <h3>Canvas coordinates diagram</h3>
  <h2>Paths</h2>
</body>
```

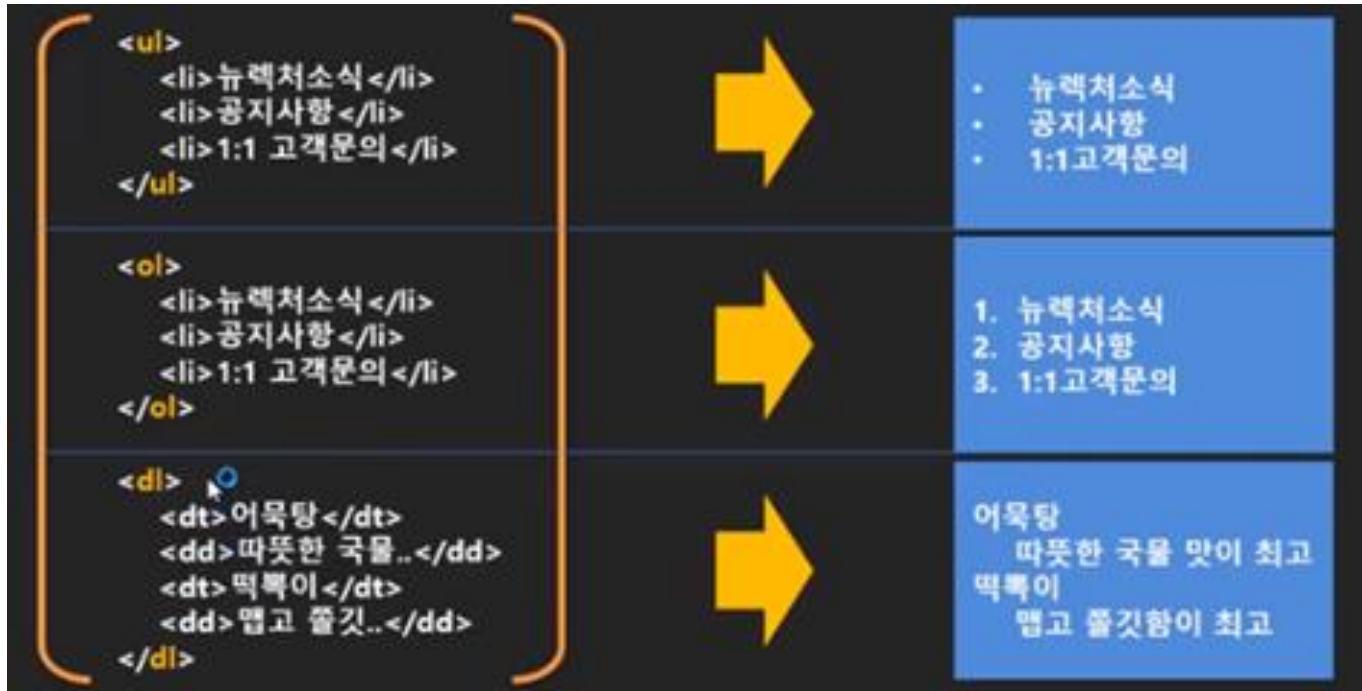
On the right, a browser window titled "Hn example" displays the rendered output, where the tags are converted into visually distinct heading levels:

- Let's call it a draw(ing surface) (H1)
- Diving in (H2)
- Simple shapes (H2)
- Canvas coordinates (H2)
- Canvas coordinates diagram (H3)
- Paths (H2)

1.8.4. 웹문서 작성 2단계 - 콘텐츠 블록의 종류 (목록)

- 목록 TAG : UL, OL, DL 3가지로 구분되며, 목적에 따라 맞추어서 사용

목록 Tag



- * UL : Unordered List (순서가 없는 것에 적합)
- * OL : Ordered List (순서가 있는 것에 적합)
- * DL : Definition List (데이터 정의하는 것에 적합)
- * LI : List Item

1.8.5. 웹문서 작성 2단계 - 콘텐츠 블록의 종류 (입력폼)

- 입력폼 Tag : 여러가지 Input Type의 요소로 구성되어 있음
- 아래외에도 다양한 Input Type이 있음 (ex. Radio button)



입력폼 Tag

The image shows a Naver login form with several annotations in red boxes identifying different HTML input types:

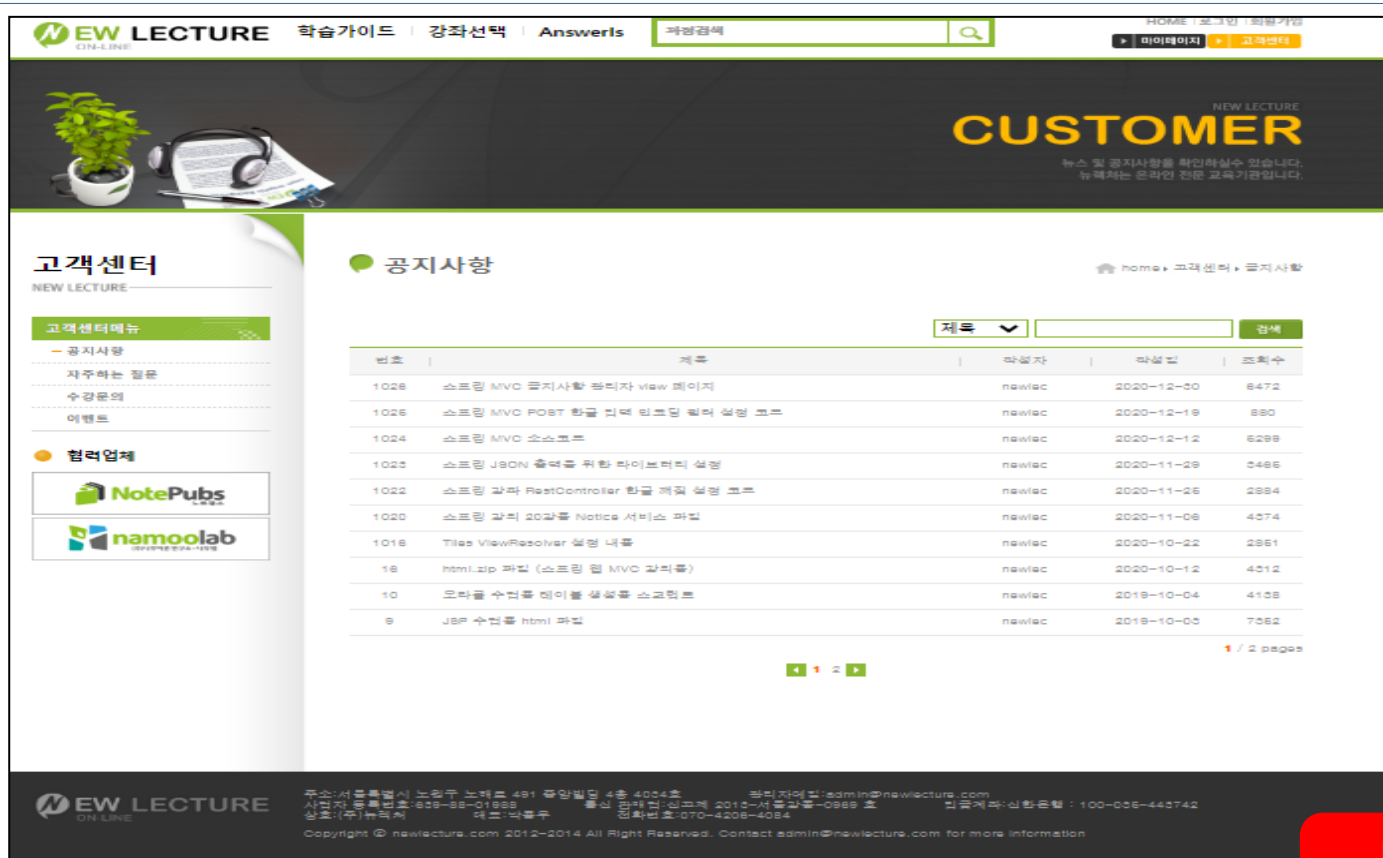
- text**: Points to the '아이디' (ID) input field.
- password**: Points to the '비밀번호' (Password) input field.
- checkbox**: Points to the '로그인 상태 유지' (Keep login state) checkbox.
- toggle**: Points to the 'IP보안' (IP Security) toggle switch.
- submit**: Points to the '로그인' (Login) button.
- label**: Points to the '비밀번호 찾기' (Find password) link.

Other visible elements include the Naver logo, a QR code login option, and links for '아이디 찾기' (Find ID) and '회원가입' (Sign up).

1.8.6. 웹문서 작성 2단계 - 콘텐츠 작성 실습

- 콘텐츠 작성 실습
- 뉴렉처의 공지사항을 기반으로 실습 (검증된 강의자료 기반으로 설명)

콘텐츠 작성 실습



NEW LECTURE 학습가이드 | 강좌선택 | Answers |

HOME | 로그인 | 회원가입
마이페이지 | 고객센터

CUSTOMER

뉴스 및 공지사항을 확인하실 수 있습니다.
뉴렉처는 온라인 전문 교육기관입니다.

고객센터

NEW LECTURE

고객센터메뉴

- 공지사항
- 자주하는 질문
- 수강문의
- 이벤트

협력업체

- NotePubs
- namoolab

공지사항

home > 강좌선택 > 공지사항

제목

번호	제목	작성자	작성일	조회수
1026	스프링 MVC 공지사항 형식자 view 페이지	newlec	2020-12-00	6472
1025	스프링 MVC POST 한글 입력 된보일 필자 설명 보트	newlec	2020-12-19	880
1024	스프링 MVC 소스코드	newlec	2020-12-12	6299
1023	스프링 JSON 출력물 위한 라이브러리 설명	newlec	2020-11-29	5486
1022	스프링 광자 RestController 한글 객체 설명 보트	newlec	2020-11-26	2884
1020	스프링 광자 20광자 Notice 서비스 파일	newlec	2020-11-06	4574
1016	Tiles ViewResolver 설명 내용	newlec	2020-10-22	2861
18	html.zip 파일 (스프링 웹 MVC 광자물)	newlec	2020-10-12	4512
10	트러블 수업을 레이블 설정물 소코르보	newlec	2019-10-04	4158
9	JSP 수업물 html 파일	newlec	2019-10-05	7562

1 / 2 pages

NEW LECTURE ON LINE

주소: 서울특별시 노원구 노원로 491 중앙빌딩 4층 4024호 | 관리자이메일: admin@newlecture.com
사업자 등록번호: 839-88-01988 | 통신판매업: 2015-서울강남-0969 호 | 대표이사: 신한용 | 100-036-443742
상호(주)뉴렉처 | 대표: 박용우 | 전화번호: 070-4208-4084 | 팩스: 070-4208-4084
Copyright © newlecture.com 2012-2014 All Right Reserved. Contact admin@newlecture.com for more information

list.html 파일로
실습

1.8.7. 웹문서 작성 3단계 - outline 설정하기

- 영역 구분 : header, main, footer, 그리고 aside 로 구분함
- 각 영역별로 주요한 특징이 있음.



영역구분의 표준안

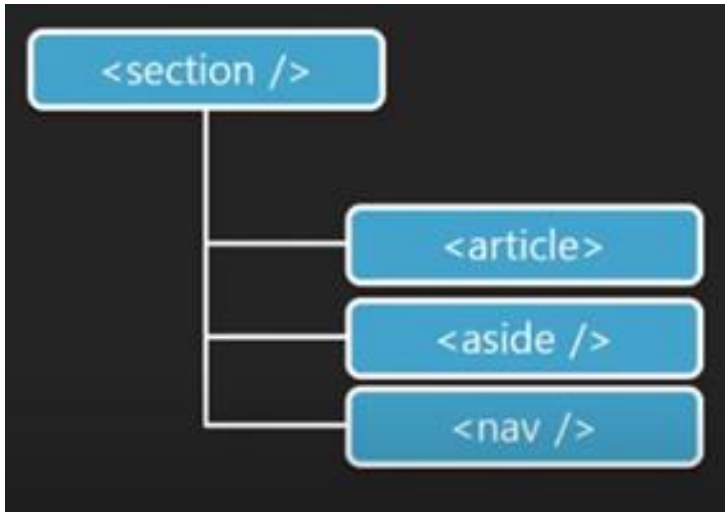


1.8.8. 웹문서 작성 3단계 - outline 설정하기 (section 활용)

- HTML5로 오면서 독립적인 부분을 설정할 때 section을 주로 사용함
- Section별 제목 tag는 H1에서부터 시작하는 것이 관례



Section Tag



포스트, 보도기사, 논평 등에 주로 사용
배너광고 및 기타링크 등의 사이드바
Navigation : HyperText 가 있는 부분

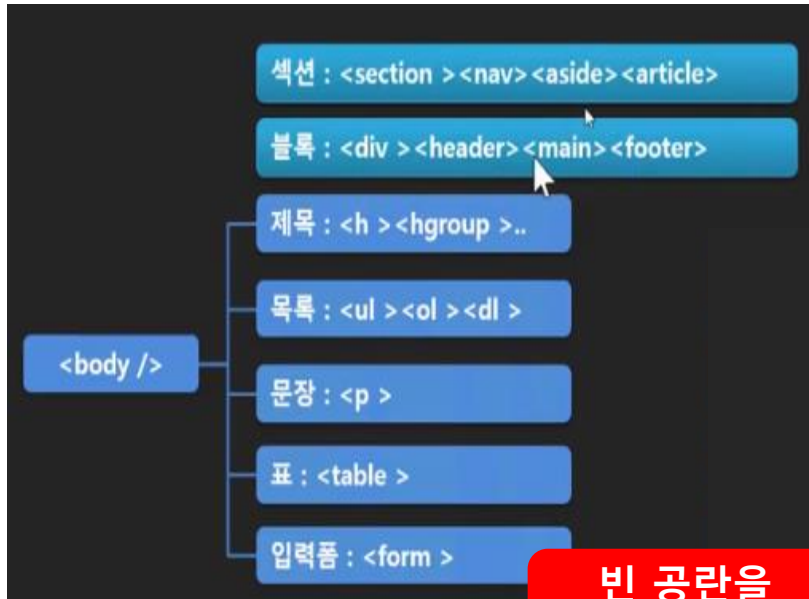
실제 완성은
CSS 학습이후
Holy grail Layout

1.9. 블록 Tag / 인라인 Tag

- 블록 Tag는 한 줄 전체를 블록화하여 영역을 선택함
- 인라인 Tag는 블록 Tag안에 본인의 영역만큼만 선택되어 짐.

블록 Tag / 인라인 Tag

블록 Tag



빈 공간을
영역으로 사용

```
<div>Text1</div>
<div>Text2</div>
<div>Text3</div>
```



Text1
Text2
Text3

인라인 Tag

Inline tags (examples shown in red dashed boxes):

- a, abbr, acronym, b, bdo, big, 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

```
<div><a>열공</a><b>화이팅</b><u>응원합니다.</u></div>
<div>Text2</div>
<div>Text3</div>
```

inline 태그들

본인 크기만큼만
영역설정.

열공 화이팅 응원합니다.
Text2
Text3

2장. CSS

2.1. CSS 개요

- CSS : Cascading Style Sheet
- HTML 문서에 스타일을 입히는 작업

CSS 개요

문서

뉴렉처 온라인

머리말

메인메뉴

- 학습가이드
- 능력과정
- 강좌선택

강좌검색 폼

검색할도

과정검색

검색

회원메뉴

- 홈
- 로그인
- 회원가입

고객센터메뉴

- 마이페이지
- 고객센터

고객센터

CSS 적용



2.2. CSS를 적용하지 않았다면

- CSS 적용 이전 : HTML 문서가 가지고 있는 기본값에 의해 Style 적용
- CSS 적용 이후 : 사용자가 CSS에 정의된 대로 Style이 적용됨.

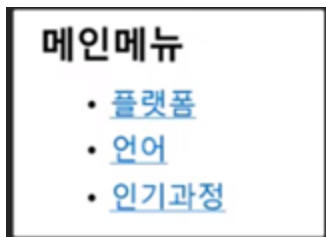
CSS 적용 사례

CSS 적용 이전

```
<h1>메인메뉴</h1>
<ul>
  <li><a href="" >플랫폼</a></li>
  <li><a href="" >언어</a></li>
  <li><a href="" >인기과정</a></li>
</ul>
```



브라우저의
기본값으로 적용됨

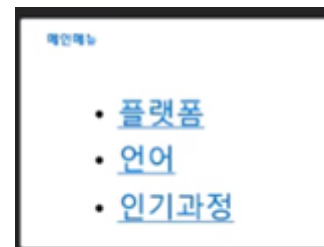


CSS 적용 이후

```
<h1 style="color:blue;font-size:5px;">메인메뉴</h1>
<ul>
  <li><a href="" >플랫폼</a></li>
  <li><a href="" >언어</a></li>
  <li><a href="" >인기과정</a></li>
</ul>
```



CSS로 정의한
스타일로 적용됨



2.3. CSS 속성들

- CSS는 Tag별로 속성을 구체화함으로써 스타일을 정의할 수 있음
- developer.mozilla.org 사이트의 Reference 메뉴에서 속성들을 참조 가능함.

CSS 속성

display	margin	text-indent	height
visibility	margin-left	text-align	min-height
	margin-right		max-height
float	margin-top	color	
clear	margin-bottom	font	
		font-family	/* caption-side */
position	border	font-size	/* clip */
z-index	border-left	font-style	/* content */
overflow	border-left-color	font-variant	/* empty-cells */
cursor	border-left-width	font-weight	/* outline */
	border-left-style	text-decoration	/* outline-color */
left	border-right	text-transform	/* outline-style */
right	border-right-color		/* outline-width */
width	border-right-width	vertical-align	/* quotes */
min-width	border-right-style	line-height	/* orphans */
max-width	border-top	white-space	/* page-break-inside */
top	border-top-color	word-spacing	/* widows */
bottom	border-top-width	letter-spacing	
height	border-top-style		

2.4. CSS 속성 적용 3가지 방법

- CSS 적용하는 3가지 방법이 있음 (개별 Tag에 Style 직접 적용, Style Tag 사용, CSS 파일 Link)
- 우선순위 : 적용하는 위치에 따라 다름. (개별 Tag는 바로 적용되므로 가장 우선함.)

CSS 적용 방법

개별 Tag에 적용

우선순위 : ①

```
<h1 style="color:blue;font-size:5px;">메인메뉴</h1>
<ul>
  <li><a href="">플랫폼</a></li>
  <li><a href="">언어</a></li>
  <li><a href="">인기과정</a></li>
</ul>
```

현재 H1 Tag에만 적용됨.

Style Tag에 사용

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

```
<h1 style="color:blue;"></h1>
<h1 style="color:blue;"></h1>
<h1 style="color:blue;"></h1>
```

H1이 속한
모든 Tag에 적용됨
➔
속성 집중관리 가능

CSS 파일 Link



[CSS 작업자]

```
body, input, select, textarea {
  font-family: "Raleway", Arial, Helvetica, sans-serif;
  font-weight: 400;
  font-size: 1rem;
  line-height: 1.65; }

a {
  -moz-transition: color 0.2s ease-in-out;
  -webkit-transition: color 0.2s ease-in-out;
  -ms-transition: color 0.2s ease-in-out;
  transition: color 0.2s ease-in-out;
  text-decoration: underline; }
a:hover {
  text-decoration: none; }

strong, b {
  font-weight: 600; }

em, i {
  font-style: italic; }
```

[Style.css 파일]



[웹 개발자]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>리스팅페이지</title>
  <!-- 없을: tag / ; class / #: id에 사용 -->
  <link href = "/"css/reset.css" type = "text/css" rel="stylesheet">
  <link href = "../css/style.css" type = "text/css" rel="stylesheet">
</head>
<body>
  <header id="header">
    <div class="contents-box">
      <h1><a href = "http://www.human.or.kr">책목 : 휴먼교육센터</a></h1>
      <section>
        <h1 style = "color: red;font-weight: bold;"> header </h1>
        <nav>
```

link

2.5.1. 선택자 (Selector) – Basic Selectors

- 선택자 : HTML내의 Tag를 선택하는 것 (Tag = Element)
- 선택자는 Basic Selector와 Combinator로 구분됨

선택자 – Basic Selectors - 1

Basic Selectors

구분	Selector 기호	우선순위
tag 기준	tag명	가장낮음
class 기준	.class명	높음
id 기준	#id명	가장높음

[Style CSS]

```
<style>
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
</style>
```

tag 선택자

클래스 선택자

ID 선택자

[HTML 문서]

```
<H1 id="human1" class="human2">휴먼교육센터</H1>
```

우선순위 실습

```
<style>
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
</style>
</head>
<body>
<H1 id="human1" class="human2">휴먼교육센터</H1>
</body>
```

휴먼교육센터

```
<style>
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
</style>
</head>
<body>
<H1 class="human2">휴먼교육센터</H1>
</body>
```

휴먼교육센터

2.5.2. 선택자 (Selector) – Basic Selectors

- 선택자 : HTML내의 Tag를 선택하는 것 (Tag = Element)
- 선택자는 Basic Selector와 Combinator로 구분됨

선택자 – Basic Selectors - 2

속성 선택자

구분	Selector 기호
속성값 기준	[속성 = 값]

우선순위 :

ID > 속성[id] > 속성[class] > class > tag

```
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
h1[id]{
  background-color: blue;
}
```

휴먼교육센터

<H1 id="human1" class="human2">휴먼교육센터</H1>

우선순위 실습

```
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
h1[id]{
  background-color: blue;
}
```

휴먼교육센터

```
h1{
  background-color: red;
}
.human2{
  background-color: green;
}
#human1{
  background-color: yellow;
}
h1[class]{
  background-color: blue;
}
```

휴먼교육센터

2.5.3. 선택자 (Selector) – Combinators

- 선택자 : HTML내의 Tag를 선택하는 것 (Tag = Element)
- 선택자는 Basic Selector와 Combinator로 구분됨

선택자- Combinators - 1

Combinators

구분	Selector	상세설명
자손	ul li	ul 하위 모든 Tag 중 li 선택
자식	ul > li	ul 하위 Tag 중 li 선택
형제	ul + li	ul 형제 중 다음 li 선택
형제	ul ~ li	ul 모든 형제 중 li 선택

CSS 적용사례

```
<ol>
  <ul>html
    <li>html4</li>
    <li>html5</li>
  </ul>
  <ul>
    css
    <li>css2.0</li>
    <li>css3.0</li>
  </ul>
</ol>
```

[자손]

```
ol li{
  background: yellow;
}
```

[자식]

```
ol>li{
  background: yellow;
}
```

자식 / 자손 사례

```
html
  ○ html4
  ○ html5
css
  ○ css2.0
  ○ css3.0
```

하위 모든 Elements에 대해 적용

li는 ul의 자식이므로
이 기준에서는 미적용

```
html
  ○ html4
  ○ html5
css
  ○ css2.0
  ○ css3.0
```

2.5.4. 선택자 (Selector) – Combinators

- 선택자 : HTML내의 Tag를 선택하는 것 (Tag = Element)
- 선택자는 Basic Selector와 Combinator로 구분됨

선택자- Combinators - 2

HTML 문서

```
<ol>html
  <li>html4</li>
  <li>html5</li>
</ol>
```

[OL : order list]

```
<ul>css
  <li>css2.0</li>
  <li>css3.0</li>
</ul>
```

[UL : unorder list]

```
<ul>기타
  <li>JS</li>
  <li>AJAX</li>
</ul>
```

[UL : unorder list]

이웃 형제 사례 (UL+OL)

```
ol+ul{
  background: yellow;
}
```

이웃된 형제 엘리먼트에
한해서만 CSS 적용됨

html
1. html4
2. html5

css
• css2.0
• css3.0

기타
• JS
• AJAX

전체 형제사례 (UL~OL)

```
ol~ul{
  background: yellow;
}
```

전체의 형제 엘리먼트에
한해서만 CSS 적용됨

html
1. html4
2. html5

css
• css2.0
• css3.0

기타
• JS
• AJAX

2.6. Color

- 색상 : CSS의 색상은 빛의 삼원색의 구조를 따르며, 투명도를 표현하는 Alpha가 있음.
- 표현방법 : RGB (255,255,255) → 흰색 / RGBA(255,255,255,0) → 투명도 100%를 의미

COLOR

컬러



```
.human2{  
    background-color: green;  
}  
#human1{  
    background-color: RGBA(255,0,0,0);  
}
```

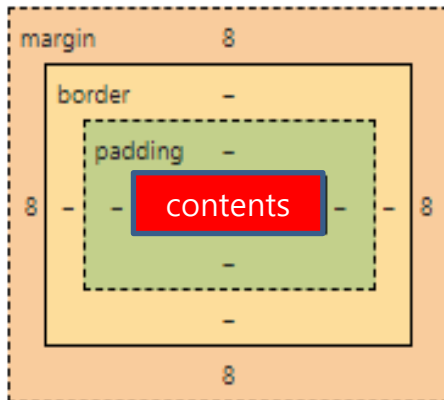
[일반 그래픽 프로그램의 Color 선택]

2.7.1. BOX 모델

- BOX : 엘리먼트는 거의 대부분이 BOX의 형태로 영역을 차지하고 있음.
- Contents가 차지하는 위치의 표현방법

BOX 모델

Box 모델 설명



구분	상세설명
padding	contents와 border 간의 여백
border	경계선
margin	이웃 엘리먼트간의 경계

Box 모델 실습

```
p{  
  border-width: 10px;  
  border-style: solid;  
  border-color: red;  
  /* border:10px solid red; */  
  width : 200px;  
  height : 300px;  
  padding : 10px;  
  margin : 10px;  
}
```



2.7.2. BOX 모델 주의사항

- Box 모델의 주요 주의사항
- 인라인 태그 / 박스-sizing / 마진겹침

BOX 모델 주요 주의사항

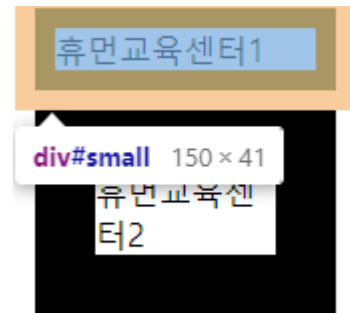
인라인 태그

```
a {  
  width : 10px;  
}  
  
</style>  
</head>  
<body>  
<a>휴먼교육센터1 </a>  
<a>휴먼교육센터2 </a>
```

인라인 Tag는
width 값 미적용.
본인 길이만큼 적용됨

margin 겹침

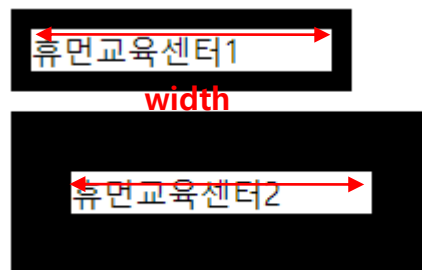
이웃한 margin 중 큰 것으로
적용됨.



박스 sizing

```
div {  
  margin : 10px;  
  width : 150px;  
}  
#small {  
  border : 10px solid black;  
}  
#large {  
  border : 30px solid black;  
}  
  
</style>  
</head>  
<body>  
<div id="small">휴먼교육센터1</div>  
<div id="large">휴먼교육센터2</div>
```

div width는
contents의 크기에 적용됨

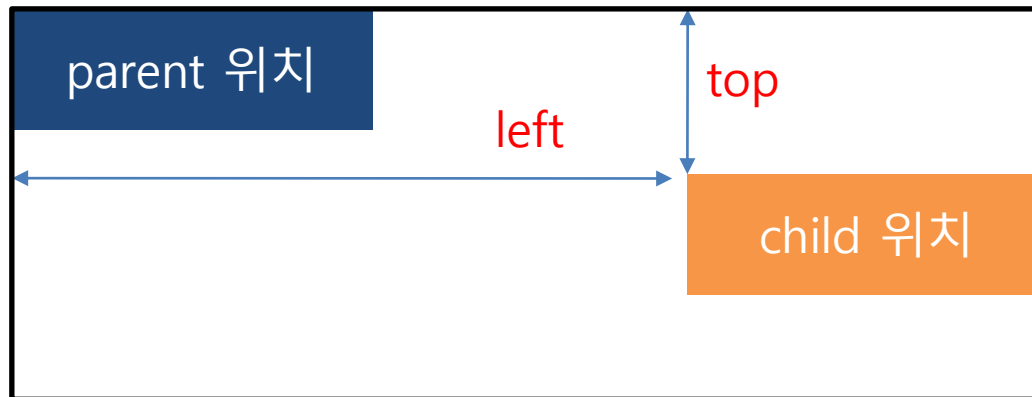


2.8.1. Position 속성

- Position 속성은 엘리먼트의 위치를 지정하는 것으로 layout 생성에 중요한 특성이 됨
- Position 속성값 (4가지) : static(기본값) / relative / absolute / fixed

Position 속성

Position 사용법



구분	상세설명
static	기본값. 위치의 통제 불가 (left, top 등의 속성 사용불가)
relative	부모의 위치에서 상대적인 left, right, top, bottom의 위치 기준으로 이동함. (단, left가 right 보다 우선하며, top이 bottom보다 우선함. 동시 사용불가)
absolute	부모의 위치와 상관없이 static이 아닌 부모의 위치를 기준으로하여 절대 위치 체크함
fixed	절대위치 처리가 되며, absolute와 다르게 scroll 상황에서도 본인의 위치 고수함

2.8.2. Position 속성 - 1 (static/relative)

- Static 속성은 기본값이며, left, top 등의 위치 미적용됨.
- Relative 속성은 부모위치에서 위치 이동함

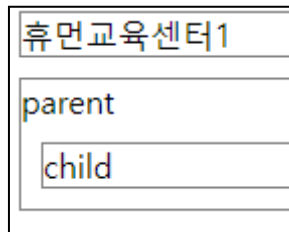
Position 속성 -1 (static/relative)

Position - static

```
div {  
  border: 1px solid gray;  
  margin: 10px;  
  position: static;  
}  
#me {  
  left: 100px;  
  top: 100px;  
}
```

미적용됨

```
</style>  
</head>  
<body>  
  <div>휴먼교육센터1</div>  
  <div id="parent"> parent  
    <div id="me">child</div>  
  </div>
```

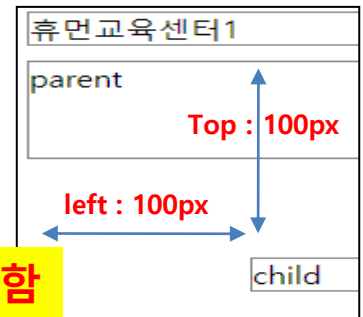


Position - relative

```
div {  
  border: 1px solid gray;  
  margin: 10px;  
  position: relative;  
}  
#me {  
  left: 100px;  
  top: 100px;  
}
```

적용됨

```
</style>  
</head>  
<body>  
  <div>휴먼교육센터1</div>  
  <div id="parent"> parent  
    <div id="me">child</div>  
  </div>
```



위치 이동함

2.8.3. Position 속성 - 2 (absolute / fixed)

- absolute 속성은 static인 부모의 위치까지 찾아서 엘리먼트를 이동함
- fixed 속성은 현재의 browser 위치에 늘 고정됨. (스크롤 상황에서도 고정됨)

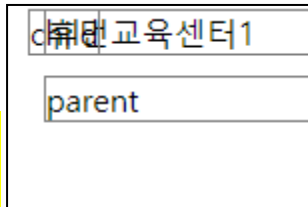
Position 속성 -2 (absolute / fixed)

Position - absolute

```
div {  
  border: 1px solid gray;  
  margin: 10px;  
}  
#me {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
}
```

```
</style>  
</head>  
<body>  
  <div>휴먼교육센터1</div>  
  <div id="parent"> parent  
    <div id="me">child</div>  
  </div>
```

Child 가
모든 div를 덮어버리면서
0,0 위치로 이동

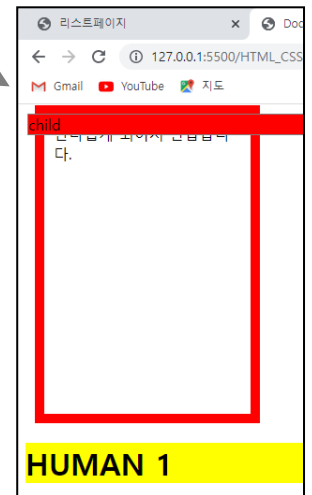


Position - fixed

```
div {  
  border: 1px solid gray;  
  margin: 10px;  
}  
#me {  
  position: fixed;  
  left: 0px;  
  top: 0px;  
  width: 100%;  
  background-color: red;  
}
```

```
</style>  
</head>  
<body>  
  <div>휴먼교육센터1</div>  
  <div id="parent"> parent  
    <div id="me">child</div>  
  </div>
```

스크롤에 상관없이
늘 고정됨.
ex.
따라다니는 배너 및
모바일에서 주로 사용

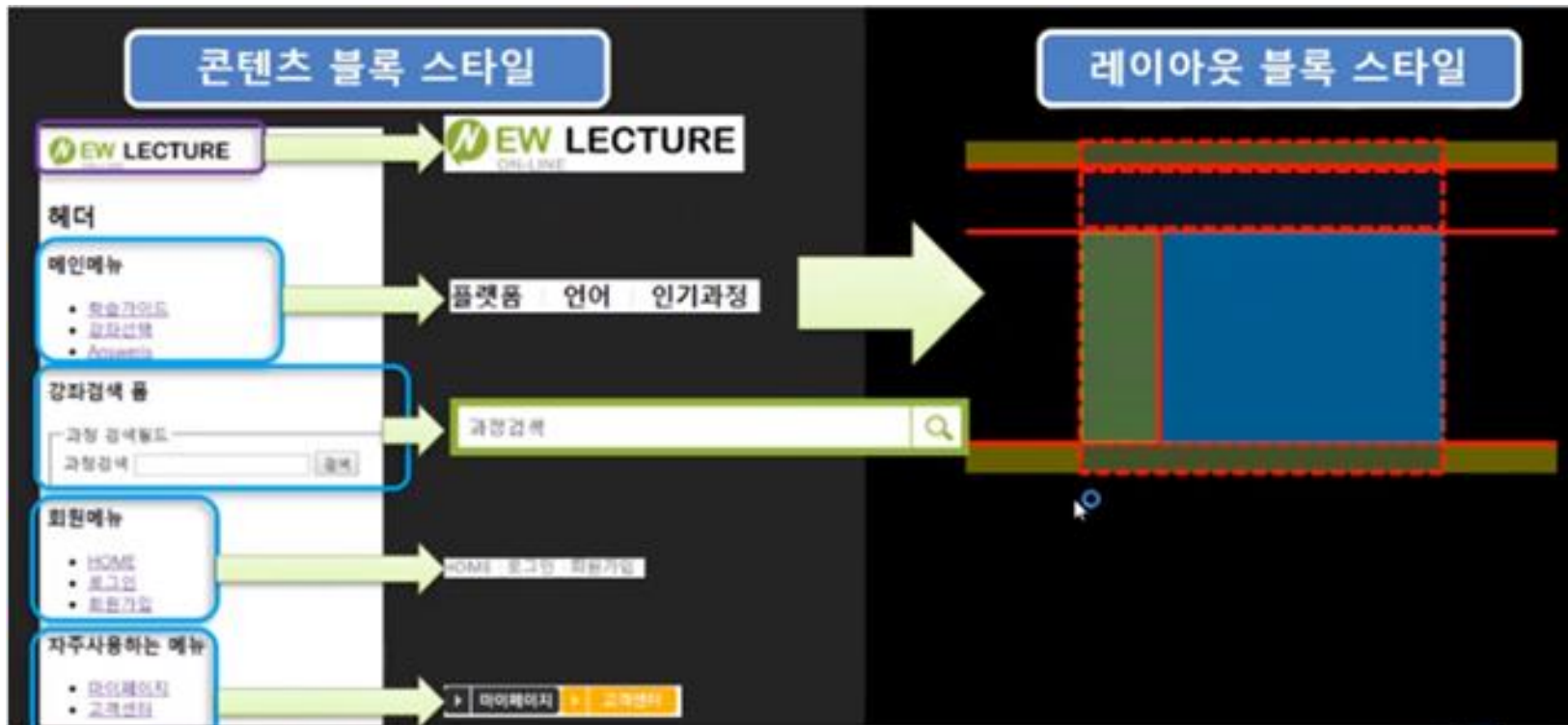


2.9.1. Layout

- Layout은 HTML 문서를 2차원 평면에 블록 단위로 배치하는 것을 의미함
- CSS의 Layout은 가로에서 세로로 표현할 수 있으며 격자형 방향은 중첩해서 만들.

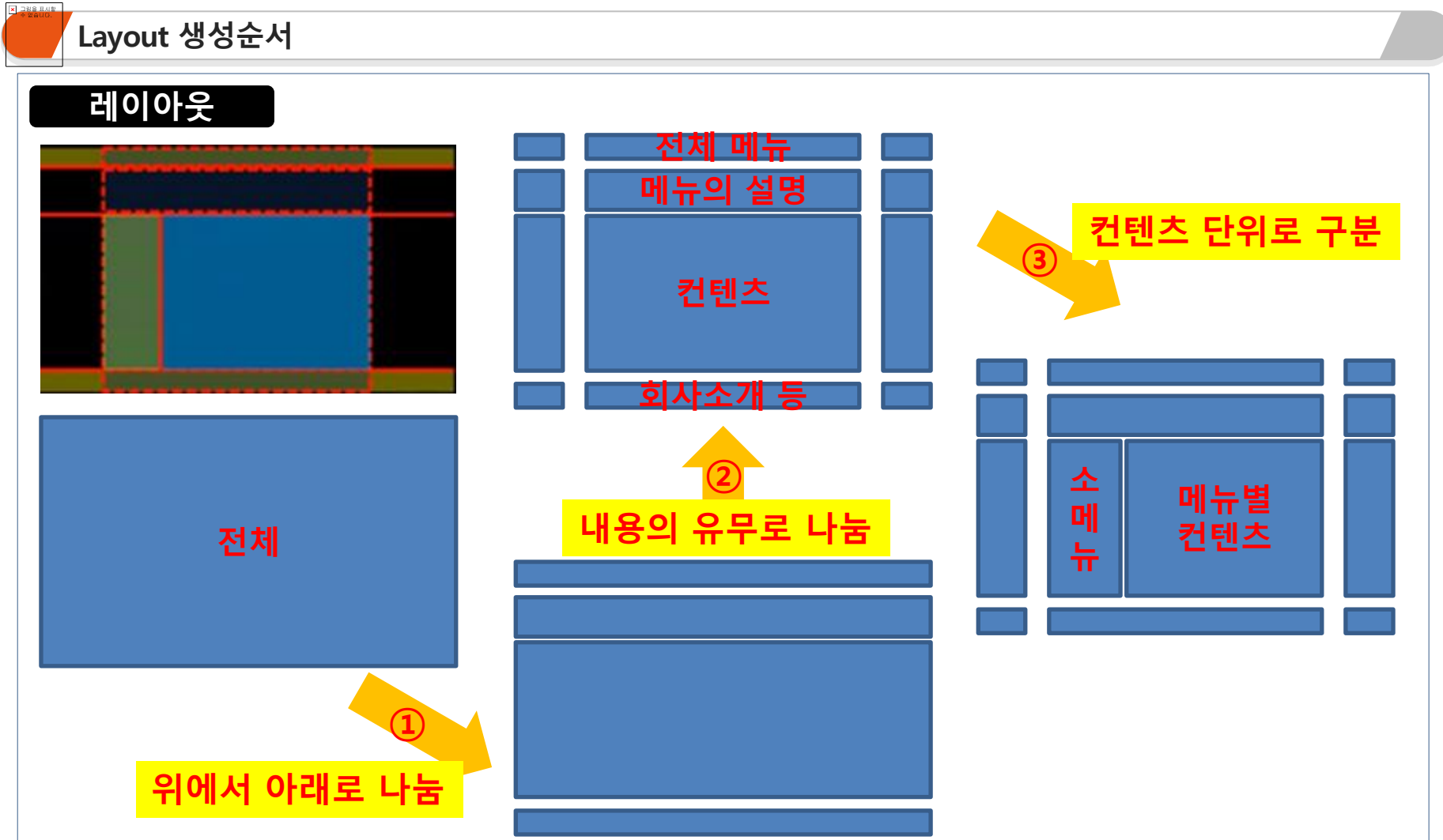
Layout

레이아웃



2.9.2. Layout 생성 순서

- CSS의 Layout은 가로에서 세로로 표현할 수 있으며 격자형 방향은 중첩해서 만듦.
- Block Tag 기반으로 hierachy를 구성하면서 layout 생성함.



2.10.1. Flex

- Flex는 layout을 생성하는데 주로 사용함. (container는 행, item은 행 내의 세부layout)
- container와 item 간의 정의할 수 있는 속성이 다르며, 조합에 따라서 다양한 layout이 생성됨



문법 및 속성

```
<container>
  <item></item>
  <item></item>
</container>
```

container

display
flex-direction
flex-wrap
flex-flow
justify-content
align-items
align-content

item

order
flex-grow
flex-shrink
flex-basis
flex
align-self

2.10.2 Flex 기본 문법

- Flex는 layout을 생성하는데 주로 사용함.
- Flex 적용시 container 안에 item들이 옆으로 나열됨.

Flex 기본 문법

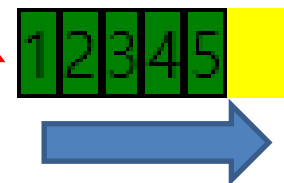
문법 및 속성

```
.container {  
  background-color: yellow;  
  display: flex;  
}  
.item {  
  background-color: green;  
  border: 1px solid black;  
}  
  
</style>  
</head>  
<body>  
  <div class="container">  
    <div class="item">1</div>  
    <div class="item">2</div>  
    <div class="item">3</div>  
    <div class="item">4</div>  
    <div class="item">5</div>  
  </div>
```

미적용시



적용시
: container 안에
item들이 옆으로 나열됨



2.10.3. Flex – container 속성 실습

Flex 박스 속성

<https://codepen.io/enxaneta/pen/adLPwv>

- Flex의 Container 속성에 따라 다양한 Layout 이 생성됨

Flex – container 속성

Container 속성

```
.container {  
  background-color: yellow;  
  display: flex;  
  flex-direction: row;  
}  
  
.item {  
  background-color: green;  
  border: 1px solid black;  
}
```

```
</style>  
</head>  
<body>  
  <div class="container">  
    <div class="item">1</div>  
    <div class="item">2</div>  
    <div class="item">3</div>  
    <div class="item">4</div>  
    <div class="item">5</div>  
  </div>
```

row

1 2 3 4 5

row-reverse

5 4 3 2 1

column

1
2
3
4
5

column-reverse

5
4
3
2
1

* container에 height를 주고 별도 실습할 것.

2.10.4. Flex – item 속성 실습 (1)

Flex 박스 속성

<https://codepen.io/enxaneta/pen/adLPwv>

- Flex의 item 속성에 따라 다양한 Layout 이 생성됨

Flex – item속성

Item 속성

item

order

flex-grow

flex-shrink

flex-basis

flex

align-self

flex-basis는 특정 아이템에 대한 크기를 고정시킴

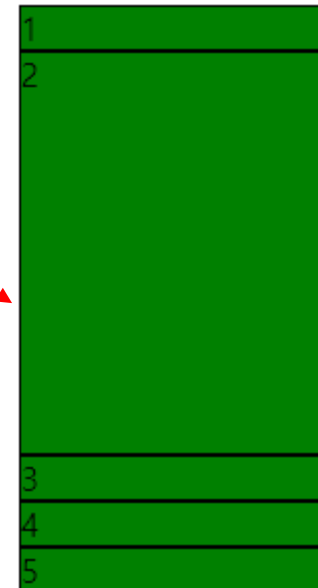
```
.container {  
  background-color: yellow;  
  display: flex;  
  flex-direction: row;  
}  
.item {  
  background-color: green;  
  border: 1px solid black;  
}  
.item:nth-child(2) {  
  flex-basis: 200px;  
}
```

</style>
</head>
<body>
 <div class="container">
 <div class="item">1</div>
 <div class="item">2</div>
 <div class="item">3</div>
 <div class="item">4</div>
 <div class="item">5</div>
 </div>

자식개체중
2번째

row

column



2.10.4. Flex – item 속성 실습 (2)

Flex 박스 속성

<https://codepen.io/enxaneta/pen/adLPwv>

- Flex의 item 속성에 따라 다양한 Layout 이 생성됨

Flex – item속성

Item 속성

item

order

flex-grow

flex-shrink

flex-basis

flex

align-self

flex-grow는 특정 아이템에 대한 크기를 다른 것과 비율로 조정함

```
.container {
  background-color: yellow;
  display: flex;
  flex-direction: row;
}
.item {
  background-color: green;
  border: 1px solid black;
  flex-grow: 1;
}
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
  </div>
```

Flex-grow : 0

1 2 3 4 5

Flex-grow : 1

→ 여백을 채워서 균등 분배함

1 2 3 4 5

만약 2번째 아이템의
flex-basis : 200px 또는
Flex-grow : 2이면(혹은 더 늘리면)

1 2 3 4 5

2.10.4. Flex – item 속성 실습 (3)

Flex 박스 속성 실습

<https://codepen.io/enxaneta/pen/adLPwv>

- Flex의 item 속성에 따라 다양한 Layout 이 생성됨

Flex – item속성

Item 속성

flex-shrink는 여백이 부족할 경우 특정 아이템에 대한 크기를 배수로 잠식

item

order

flex-grow

flex-shrink

flex-basis

flex

align-self

```
.container {
  background-color: yellow;
  display: flex;
  flex-direction: row;
}
.item {
  background-color: green;
  border: 1px solid black;
  flex-grow: 1;
}
.item:nth-child(1) {
  flex-basis: 150px;
  flex-shrink: 1;
}
.item:nth-child(2) {
  flex-basis: 150px;
  flex-shrink: 2;
}
```

```
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
  </div>
```

화면의 크기를
좌우로 늘리고 줄일 때
배수단위로 늘어나고 줄어듦

2.11. Holy Grail Layout

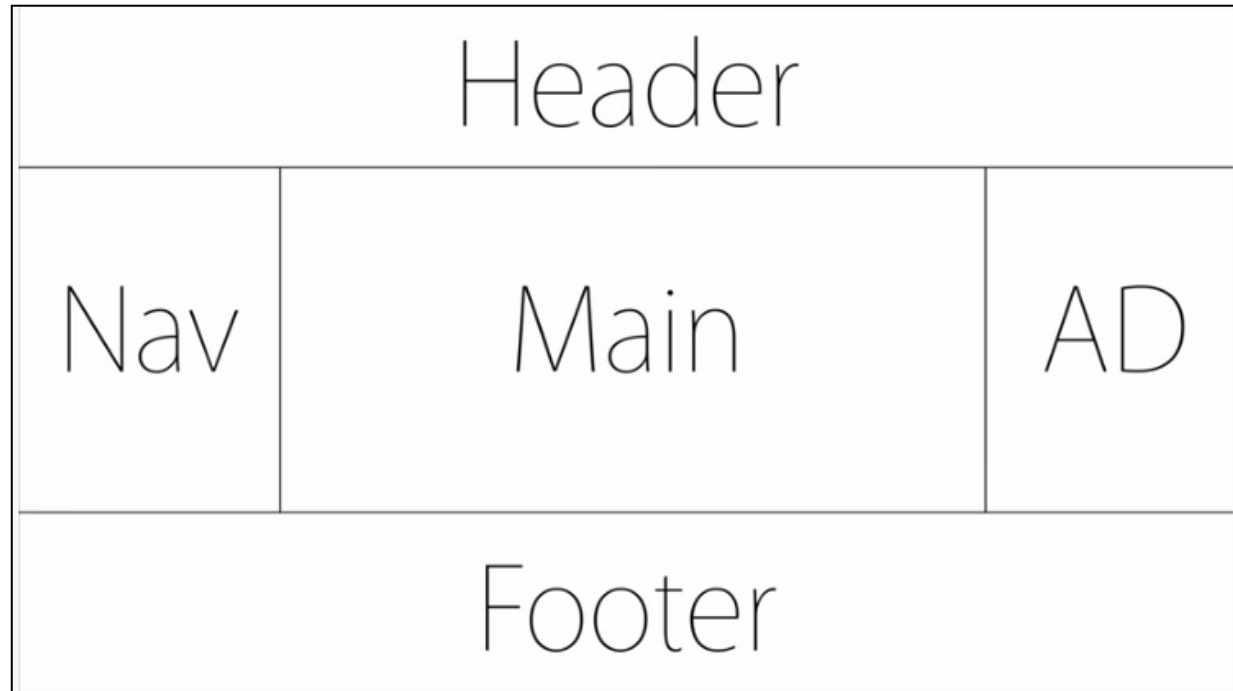
Flex 박스 속성

<https://codepen.io/enxaneta/pen/adLPwv>

- Flex 도입에 따라 holy grail layout 구현이 쉬워짐



Holy Grail Layout



실습 : holy grail layout.html