웹퍼블리싱 강의

2022.8

강 현 준

human@human.or.kr

강의 목차

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

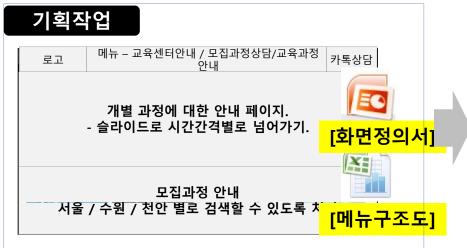
1장. HTML

1.1. 웹퍼블리싱이란?

- 웹퍼블리싱이란? 기획자의 의도대로 실제 웹사이트를 구축하는 과정을 의미함



웹퍼블리싱의 진행과정









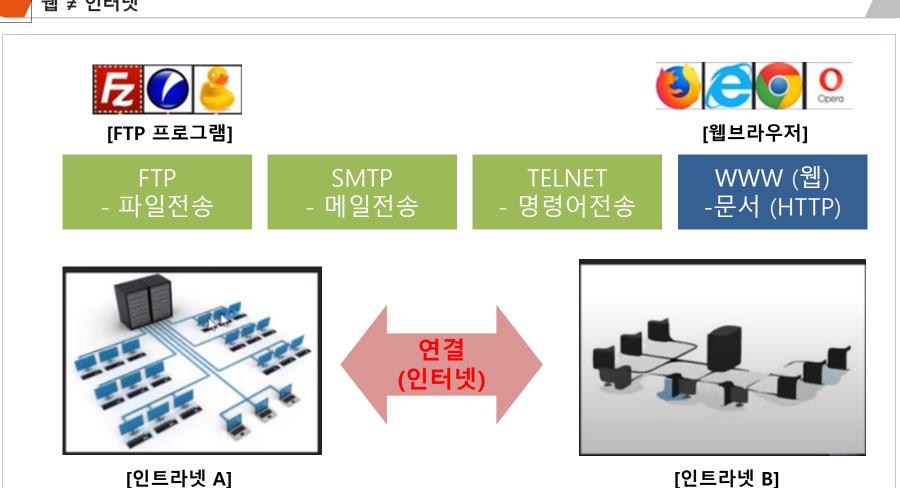


1.2. 웹과 인터넷의 관계

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



웹 ≠ 인터넷



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 브라우져에서 표시 가능함.



Markup 언어란 문서의 콘텐츠를 구분하기 위해 Tag를 사용함





1.5. HTML 문서의 기본구조

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



HTML 문서의 기본구조

```
<html>
  <head>
        화면의 설정과 제목
 </head>
 <body>
     화면에 표현할 내용
  </body>
</html>
```



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

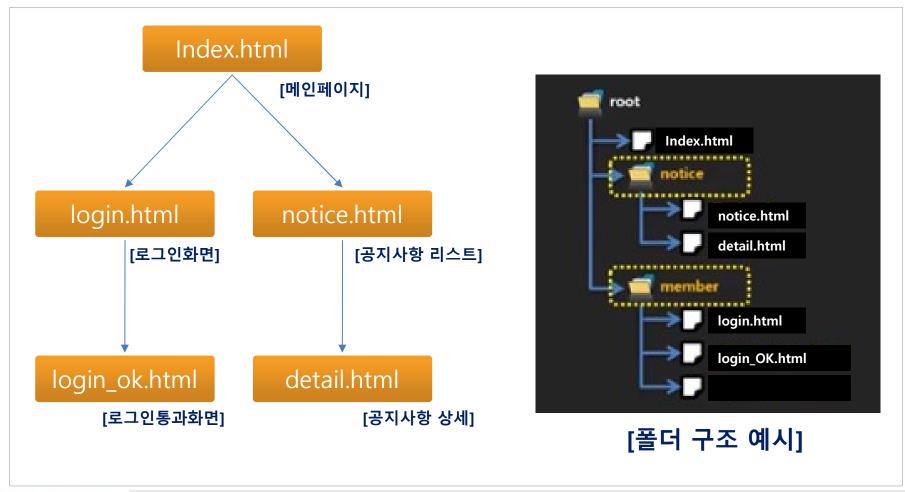


1.7.1. 실습 - 1

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



실습



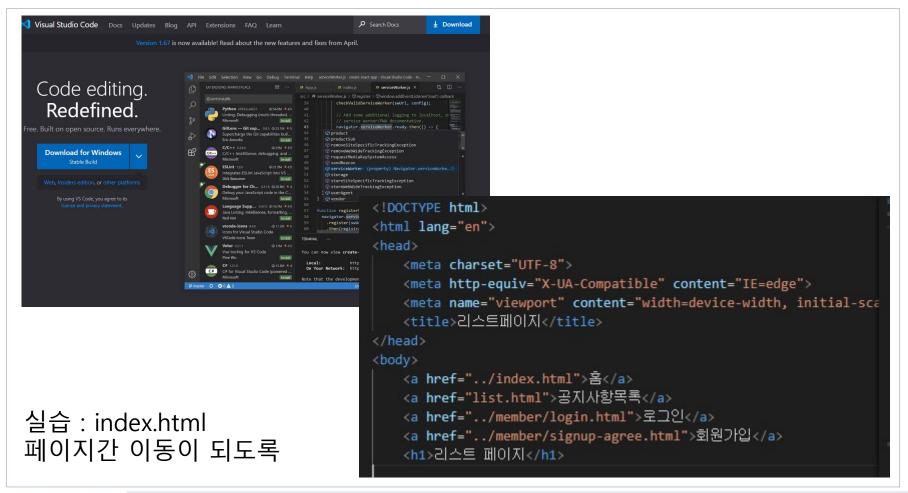


1.7.2. 실습 - 1

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



상대경로 기반으로 각 페이지별로 이동이 되는 확인.



1.8.1. 웹문서 작성 1단계 – 콘텐츠 작성하기

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



콘텐츠 예시 확인 방법



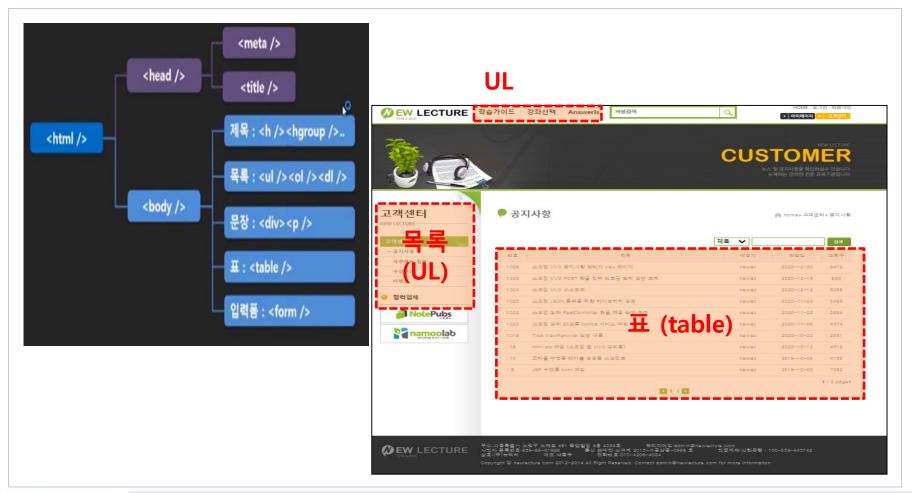


1.8.2. 웹문서 작성 2단계 – 콘텐츠 블록의 종류

- 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



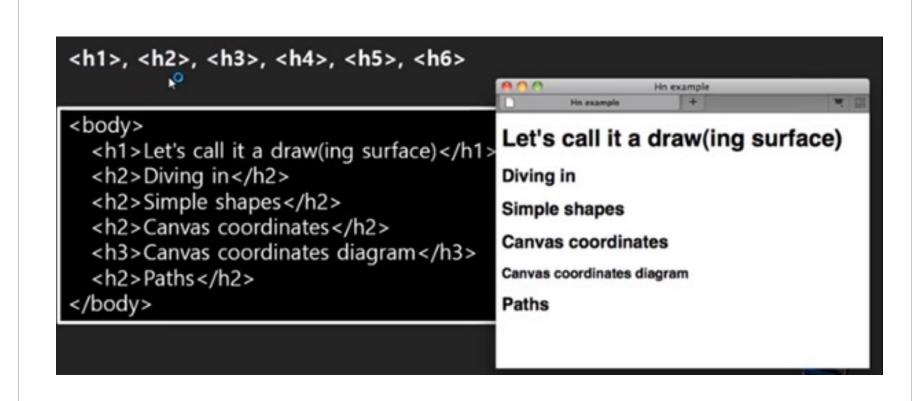


1.8.3. 웹문서 작성 2단계 - 콘텐츠 블록의 종류 (제목)

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



제목 Tag



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



5 휴먼교육센터

1.8.5. 웹문서 작성 2단계 – 콘텐츠 블록의 종류 (입력폼)

동영상. 오디오. 이미지

https://inpa.tistory.com/entry/HTML-%F0%9F%93%9A-%EB%AF%B8%EB%94%94%EC%96%B4imgaudiovideo-%EA%B4%80%EB%A0%A8-%ED%83%9C%EA%B7%B8-%EC%A0%95%EB%A6%AC

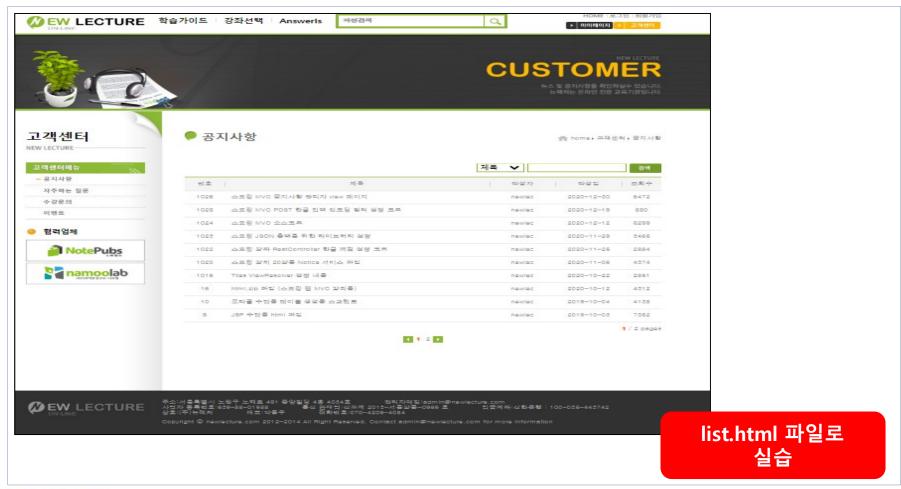
1.8.6. 웹문서 작성 2단계 – 콘텐츠 작성 실습

- 콘텐츠 작성 실습

_



콘텐츠 작성 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

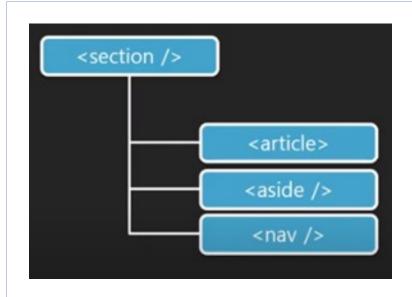




1.8.8. 웹문서 작성 3단계 - outline 설정하기 (section 활용)

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





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

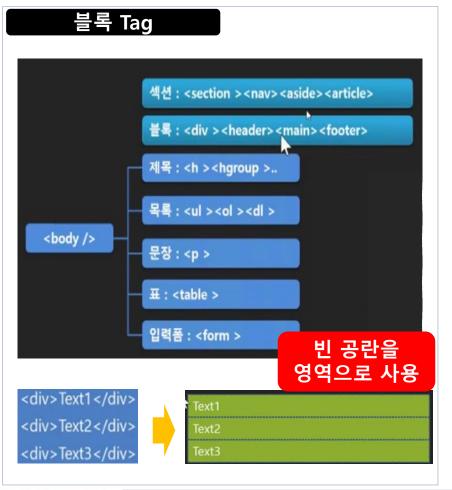
> 실제 완성은 CSS 학습이후 Holy grail Layout

1.9. 블록 Tag / 인라인 Tag

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



블록 Tag / 인라인 Tag







2장. CSS

2.1. CSS 개요

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



CSS 개요



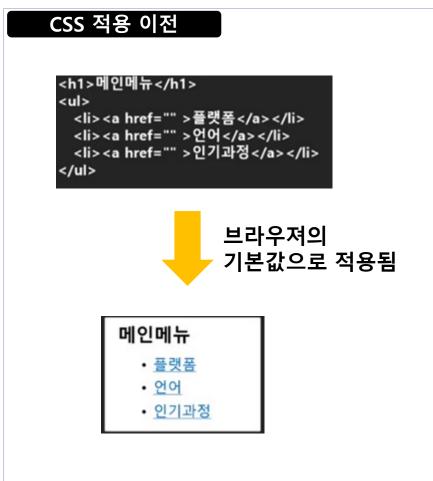
⑤ 휴먼교육센터

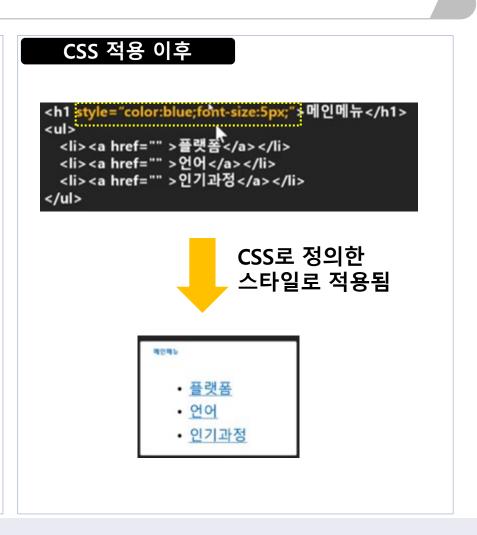
2.2. CSS를 적용하지 않았다면

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



CSS 적용 사례





2.3. CSS 속성들

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



CSS 속성

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

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







H1이 속한 모든 Tag에 적용됨 속성 집중관리 가능



[웹 개발자]



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 가장낮음 tag 기준 tag명 높음 class 기준 .class명 id 기준 #id명 가장높음 [Style CSS] tag 선택자 background-color: red; 클래스 선택자 background-color: green; background-color: yellow ID 선택자 [HTML 문서] <H1 id="human1" class="human2">휴먼교육센터</H1>

```
우선순위 실습
      background-color: green;
      background-color: vellow;
                       휴먼교육센터
head>
 <H1 id="human1" class="human2">휴먼교육센터</H1>
      h1{
          background-color:  red;
       .human2{
          background-color: green;
          background-color: _yellow;
   </style>
</head>
   <H1 class="human2">휴먼교육센터</H1>
```

2.5.2. 선택자 (Selector) - Basic Selectors

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



선택자 – Basic Selectors - 2



```
우선순위 실습
    background-color: ■red;
 .human2{
    background-color: green;
h1[id]{
    background-color: blue;
   background-color: ■red;
   background-color:  green;
   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





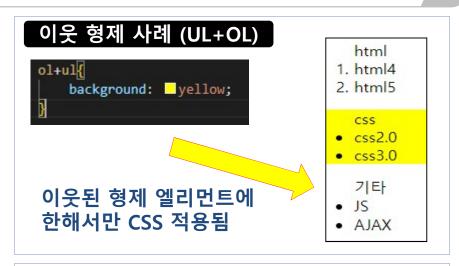
2.5.4. 선택자 (Selector) - Combinators

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



선택자- Combinators - 2

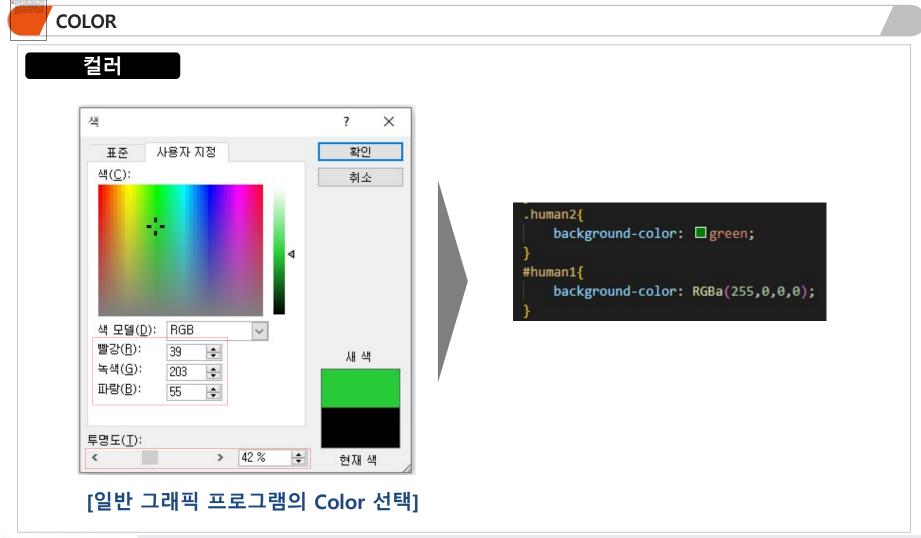






2.6. Color

- 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



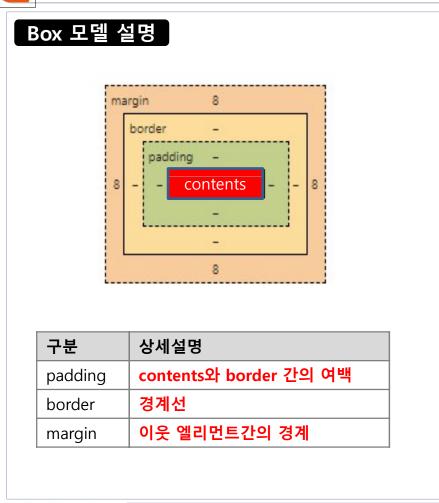
⑤ 휴먼교육센터

2.7.1. BOX 모델

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



BOX 모델





2.7.2. BOX 모델 주의사항

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





<div id="small">휴먼교육센터1</div>
<div id="large">휴먼교육센터2</div>

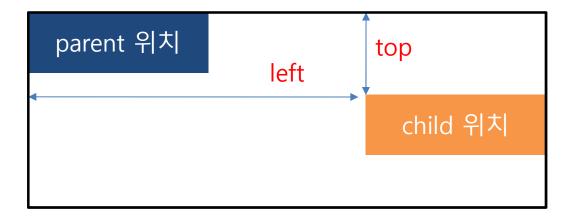
2.8.1. Position 속성

- 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:statid
         left : 100px;
                          미적용됨
         top:100px;
   </style>
   <div>휴먼교육센터1</div>
  <div id="parent"> parent
      <div id="me">child</div>
                                휴먼교육센터1
                                parent
                                 child
```

```
Position - relative
      div {
         border: 1px solid ■gray;
        margin: 10px;
         position:relative
        left : 100px;
        top:100px;
<body>
   <div>휴먼교육센터1</div>
  <div id="parent"> parent
      <div id="me">child</div>
                               휴먼교육센터1
                               parent
                                         Top: 100px
                                 left : 100px
                위치 이동함
                                             child
```

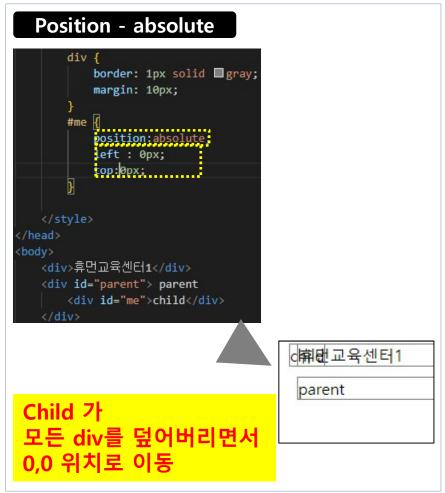


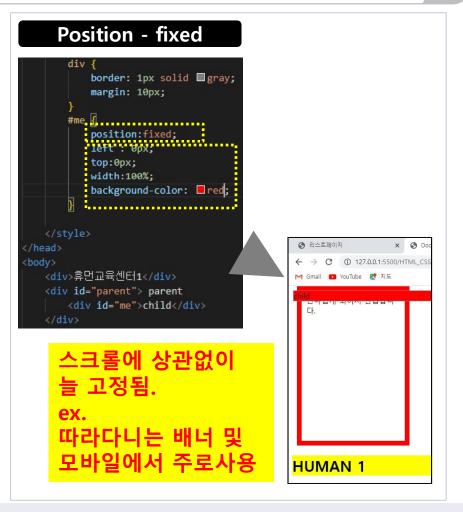
2.8.3. Position 속성 – 2 (absolute / fixed)

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



Position 속성 -2 (absolute / fixed)

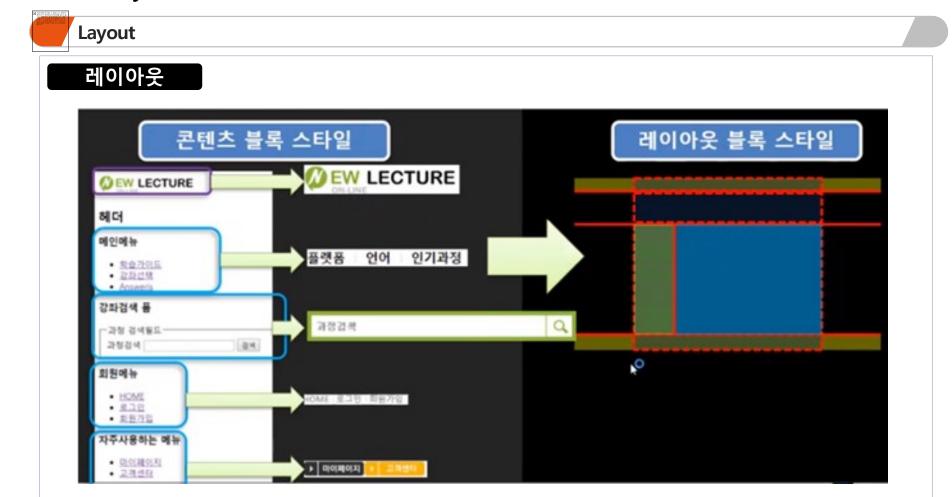






2.9.1. Layout

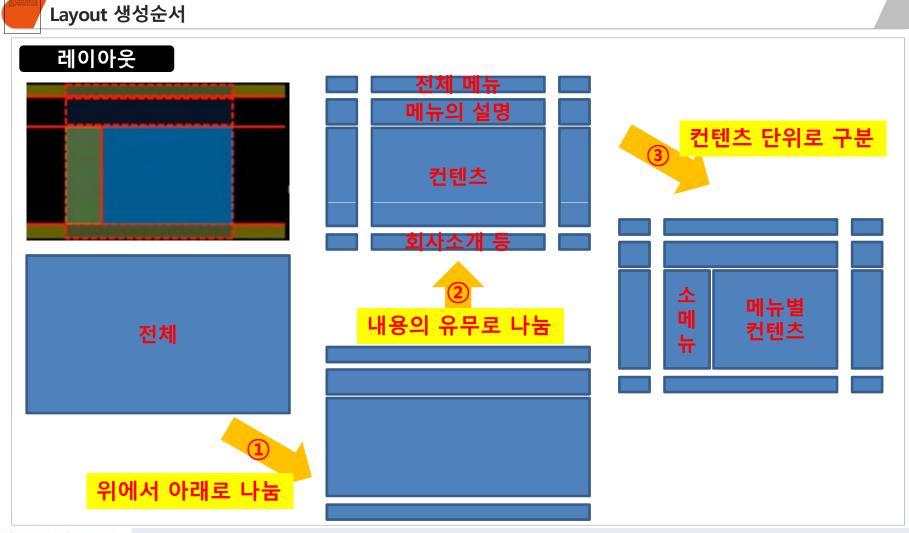
- Layout은 HTML 문서를 2차원 평면에 블록 단위로 배치하는 것을 의미함
- CSS의 Layout은 가로에서 세로로 표현할 수 있으며 격자형 방향은 중첩해서 만듦.





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



Flex

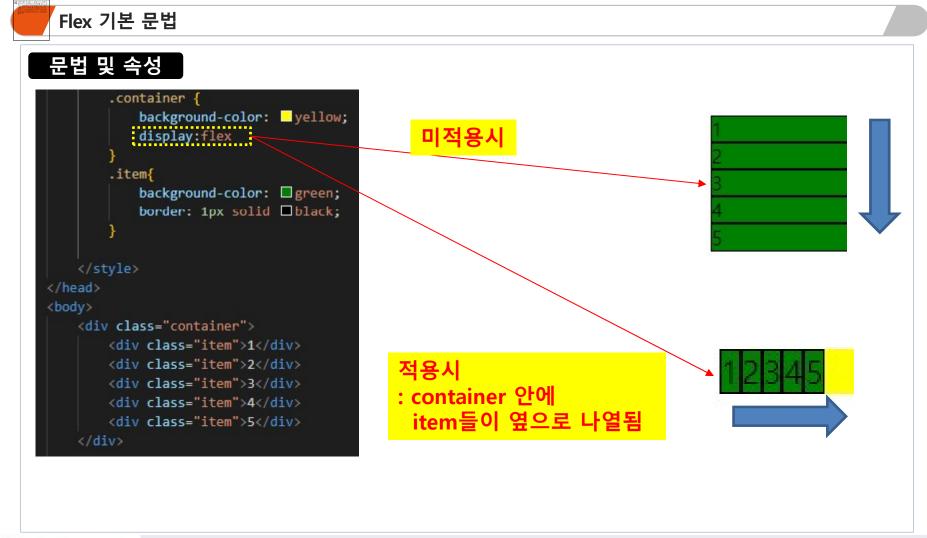
문법 및 속성

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

container	item
display	order
flex-direction	flex-grow
flex-wrap	flex-shrink
flex-flow	flex-basis
justify-content	flex
align-items	align-self
align-content	

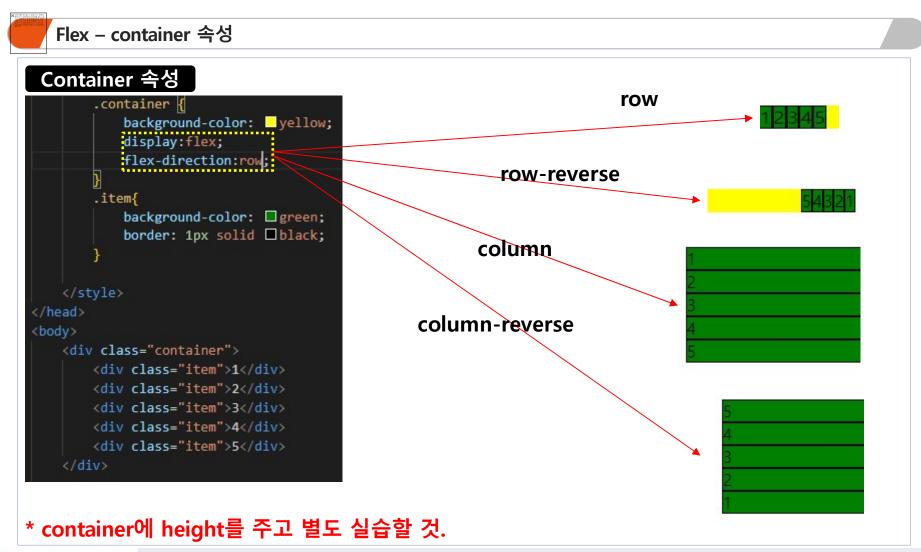
2.10.2 Flex 기본 문법

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

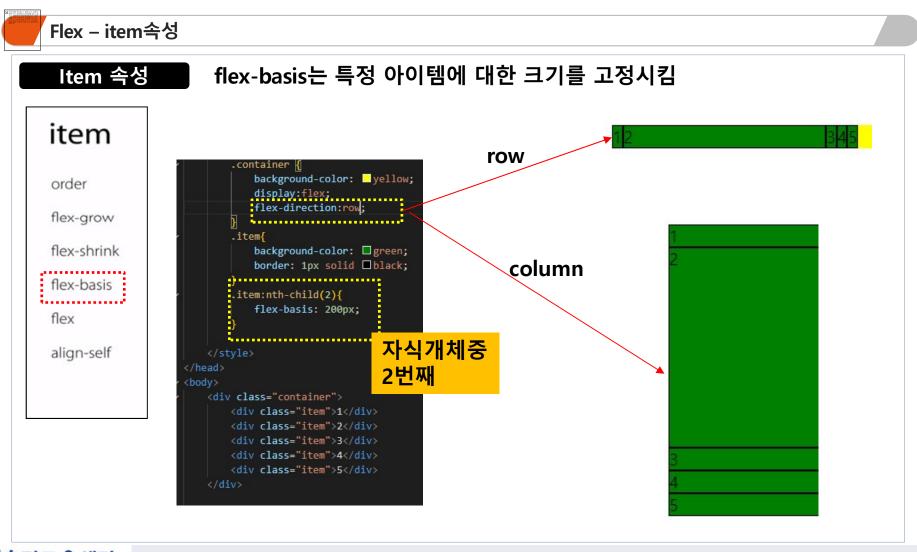




- Flex의 Container 속성에 따라 다양한 Layout 이 생성됨



- Flex의 item 속성에 따라 다양한 Layout 이 생성됨



- Flex의 item 속성에 따라 다양한 Layout 이 생성됨



- 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>
   <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

Flex 박스 속성 https://codepen.io/enxaneta/pen/adLPwv

- Flex 도입에 따라 holy grail layout 구현이 쉬워짐

Holy Grail Layout Main 실습 : holy grail layout.html