UTF-8 : converter할 때 글자가 깨지는 문제가 발생할 수 있음 -> 다른 언어 파일마다 꼭 포함하기

스타일 : 주위 환경의 영향을 받음
요소 : block / inline / inline-block

box - 영역 - element 크기 - 영향을 미치는 면적

스타일 만들기

css란?

- CSS는 웹 페이지에 “디자인”이라는 시각적 가치를 부여하기 위한 언어입니다.
- HTML문서의 레이아웃과 스타일을 정의한다.
- W3C에서 표준안을 만들어 내고 있으며 1996년 CSS Level 1 이후, 현재 CSS Level 2.1에서 CSS Level 3까지 보급되어 있음.
- W3C 표준 : CSS Level 2 Revision 1 (CSS2.1), 모바일 : CSS3 (CSS2.1기술 포함됨)

Internal Style Sheet

html문서의 <head></head> 사이에 CSS를 선언하는 방식 (meta 뒤에, script 앞에)마찬가지로, CSS분리원칙에 위배됨)

```
<head>
        <style type="text/css">
                selector { 속성 : 값; }
        </style>
</head>
```

External Style Sheet:link사용(가장권장)

<link>요소로 외부 화일 사용하기

```
<head>
<link rel="stylesheet" type="text/css" href="css파일명" >
</head>
```

```
@charset "utf-8";

p{
    background: beige;
    color: gray;
}
p span{
    font-weight: bold;
    color: blue;
}
```

서식

```
선택자{속성    :    값;
       속성    :    값;
       속성    :    값;    }
```

- selector (선택자) : 스타일을 적용하는 대상
- property (속성) : 스타일의 종류
- value (값) : 속성이 가질 수 있는 값

```
body {color:gray;}
div {border:1px solid red;background-color:gray;}
```

선택자-1

기본선택자

선택자	기능설명
요소	**특정 요소 선택**
#id명	**특정id명을 가진 요소 선택**
.class명	**특정class명을 가진 요소 선택**
요소1 요소2	**요소1 안의 있는 모든 요소자손 요소2**
부모요소1>자식요소2	부모요소1 안에 있는 자식 요소2
형제요소1+형제요소2	형제요소1 옆에 있는 형제 요소2
형제요소1~형제요소2	요소1 의 모든 형제 요소2
선택자1, 선택자2	여러 선택자 한꺼번에 선택
*	모두 선택

타입선택자

요소명을 선택자로 사용

```
h1 {font-size:20px;}
ul {margin:0px;}
```

class 선택자 (요소명.class명 : 요소명 생략가능)

- 클래스 이름으로 선택하기, 문서내에서 여러 요소를 한 번에 선택할 수 있다.

```
div.note { color : blue ;}
.clr {clear:both;}
```

id 선택자 (요소명#id명 : 요소명 생략가능)아이디 이름으로 선택하기

```
ul#gnb {list-style-type : none ;}
#bodyContent {width:100%; height:500px;}
```

자손 ,종속선택자

- 특정 엘리먼트 하위의 엘리먼트를 지정할 때 사용
- 공백(space) 으로 구분함
- ID, CLASS등 여러 선택자와도 함께 사용 가능

```
div p {height:30px;}
div>p {color:blue;}
```

형제 이웃선택자

기준이 되는 요소 와 형제 또는 다음에 오는 요소 선택

```
h1+ul{background-color:red;}
h1~ul{background-color:red;}
```

그룹 선택자

- 선택자와 선택자를 콤마(,)로 구분하여 그룹화 하고 복수의 요소에 같은 스타일을 적용합니다.
- class 선택자나 id 선택자, 속성 선택자, 하위 선택자, 자식 선택자, 인접 선택자 등을 지정할 때도 콤마(,)로 구분하여 그룹화 할 수 있다.

```
#wrap, p.note, blockquote p, h1[title] { color : red ; }
```

Color

Keyword (색상명 , 17가지)

- black, silver, gray, white, maroon, red, purple, teal, fuchsia, green, lime, olive, yellow, navy, blue, aqua

```
color:lime;
background-color:maroon
```

셈플참조: https://en.wikipedia.org/wiki/Web_colors(구글에서 webcolor검색)

http://www.w3schools.com/colors/colors_picker.asp(구글에서 webcolor검색)

RGB값 (16진수, 6자리)

color : #ffff00 ;

RGB값 (16진수, 3자리)

color : #ff0 ;(#ffff00 =#ff0)

RGB값 (10진수)

color : rgb(255,0,0) ;

RGB값 (%)

color : rgb(100%,0,0) ;

주의할 점은 값이 '0' 이더라고 '%'를 생략해서는 안된다.

white	#ffffff	rgb(255,255,255)	rgb(100%,100%,100%)
black	#000000	rgb(0,0,0)	rgb(0%,0%,0%)

rgba(red,green,blue,alpha)

- red green blue alpha로 칼러 지정
- alpha:0~1(0.5=50%)

background-color:rgba(0,0,0,0.5)

hsl(hue,saturation,lightness)

- hue:색상환에서 0-360까지의 값으로 표현
- 채도:%로 나타냄(0:무채색~100%:순색)
- 밝기:%로 나타냄(0~100%)

background-color:hsl(0, 0%, 78%)

hsla(hue,sauration,lightness,alpha)

background-color:hsla(0, 0%, 78%.0.5)

단위

절대단위

pt: 포인트 1pt는 1/72 inch

cm, mm: 센티미터, 미리미터

pc: 파이카,1pc는 12pt

in: inch

상대단위

px : 디스플레이 해상도에 따라 화면에서 보이는 크기 변동

% : 부모 요소에 비해 얼마나 커야하는지 비율로 명시 기준의 크기를 100%로 잡는ㄴ다.

em : 부모 요소에 비해 얼마나 커야하는지 척도로 명시 기준의 크기를 1로 잡는다.

%와 em의 계산법

%: 자신의 크기/기준의 크기*100

320/450*100=71.111111111111%

em: 자신의 크기/기준의 크기

320/450=0.711111em

참고사이트:http://pxtoem.com

타이포그래피

font-family

- 글자의 모양을 설정한다.
- 하나 이상의 글꼴을 선언할 수 있으며 글꼴은 콤마(,)로 구분한다.
- 주의사항
  1. 한글글꼴은 “ " 반드시 필요하다. 한글 글꼴 뒤에는 꼭 글꼴의 영문이름을 추가한다.
  2. 영문글꼴이더라도 글꼴이름에 공백이 있는 경우 “ " 필요하다.(ex:PT mono, Malgum Gothic)
- 해당 글꼴이 없을 경우 제일 마지막에 대표 패밀리체가 렌더링되며, 만약에 폰트 페밀리를 지정하지 않으면, 사용자 컴퓨터에 설정된 브라우저 글꼴로 렌더링 된다.

형식:font-family:1순위글꼴패밀리명 , 2순위글꼴패밀리명 , 3순위글꼴패밀리명,…,대표패밀리명

```
font-family : "나눔고딕", "돋움", Dotum, sans-serif;
```

대표 패밀리란?

serif(인쇄물), snas-serif(온라인) cursive, fantasy, monospace

serif: 문자 끝부분에 장식용 가시와 고리모양고상하고 고전적인 느낌을 준다.
Times, Times New Roman, Georgia,바탕, 궁서

sans-serif: 고리모양이 없다.깔끔하고 읽기쉽다.
Verdana, Arial black, Arial,굴림, 돋움

Cursive(흘림체): 손으로 쓴것같은 폰트 (필기체)
Caflish script, Sanvito

Fantasy 화려한 장식
Impact

Monospace 일정한 폭의 문자 폰트로 이루어저 타자기로 타이핑한 것 같은 느낌
`Courier, Prestige`

font-size 웹 문서의 글자 크기를 지정한다.

- 방식:

키워드:xx-small, x-small,small, normal, large, x-large,xx-large
절대,상대크기:pt,px,%,em

font-size:키워드 또는 길이(상대, 절대)

```
font-size:12px;
font-size:1.5em;
```

```
font-size:150%
```

font-weight

- 글꼴의 굵기를 표현
- 방식:
  - 키워드:normal, bold, bolder, light, lighter
  - 숫자: 100(lighter)-400(normal)-600(bold)-900(bolder)

```
font-weight:키워드 , 또는 100-900
```

```
font-weight:normal;
font-weight:bold;
font-weight:900;
```

line-height

- 문단의 행간을 지정 할 때 사용한다.
- line-height는 font-size로 지정된 글자 크기를 포함하여 글자의 위쪽과 아래쪽 여백까지를 말한다.

- font-size보다 작게 지정할 경우에는 글자가 겹칠 수 있다.
- 보통은 120%로 신문등 글자가 많은 경우(신문)는 150%로 사용되는 경우가 많다.
- 수치는 짝수로 사용하는 것이 좋다.

```
line-height:normal,실수,길이,%,em,inherit
```

```
line-height:    normal;     기본값 = 1.2
line-height:    1.6;        normal의 1.6배
line-height:    16px;
line-height:    200%
```

font-sytle

- 글꼴의 스타일을 지정할 때 사용합니다.

```
font-style:normal ,italic,oblique;
```

```
font-style : normal ;
font-style : italic ;
font-style : oblique ;
```

font-variant

- 영문 글꼴의 소문자를 대문자로 변형할 때 사용합니다.
- 값 : normal | small-caps

```
font-variant:normal,small-caps;
```

font-variant : normal ;	if I had thought about it I might have been scared.
font-variant : small-caps ;	IF I HAD THOUGHT ABOUT IT I MIGHT HAVE BEEN SCARED.

font

- font-family, font-size, line-height, font-weight, font-style, font-variant 의 6가지 속성을 한번에 선언할 때 사용하는 대표 속성 이다.
- 선언**순서**를 지켜야 하고, **font-size와 font-family**는 꼭 선언해야 한다.

```
font:[ font-weight | font-style | font-variant ] | [ font-size ] | [ /line-height ] | [ font-family ] ;
```

font : bold italic small-caps 12px /1.6 “돋움”,Datum,sans-serif;

text-indent

- 문단의 첫 줄 들여쓰기를 적용(값 적용 가능값만큼 첫줄만 앞당겨짐)
- 블록요소에만 적용 가능 (인라인요소인 a, span태그에는 안먹힘)
- 음수도 가능하다. 하지만 음수인 경우에는 같은 값인 안의 여백이 필요하다.

```
text-indent:길이 | 퍼센트
```

text-indent:50px;
text-indent:2em;
p{text-indent:-50px; padding-left:50px;}

text-align

- 단락의 텍스트를 가로 기준으로 정렬할 때 사용 합니다.

```
text-align:left | center | right | justify
```

text-align : left;
text-align : center;
text-align : justify;

vertical-align

- 인라인 요소끼리의 세로 위치를 정렬할 때 사용된다.
- 인라인요소(이미지, 인풋, 스팬 등), 일부 블록요소(테이블의 th, td)에 사용가능
- 지정한 인라인요소 뒤에 오는 요소에 대한 세로정렬 가능

vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | 길이 | 퍼센트

vertical-align:top;
vertical-align:middle;

text-decoration

- 텍스트에 밑줄을 긋거나 취소선 등의 효과를 추가할 때 사용한다.

text-decoration:none | underline | overline | line-through

text-decoration : underline
text-decoration : line-through none;
text-decoration :overline;
text-decoration :none;

text-decoration: underline;
~~text-decoration: line-through;~~
text-decoration: overline;
text-decoration: none;

letter-spacing

문자 간격을 조절할 때 사용된다. 음수 양수 모두 사용가능.

letter-spacing:길이 | normal

letter-spacing:1em; (글자크기에 비례하여 조절)
letter-spacing:-3px;

word-spcing

단어와 단어 사이의 간격을 조절

word-spacing:길이 | normal

word-spacing:1em;
word-spacing:-3px;

text-transform

영문 대소문자를 변환할 때 사용 합니다.

text-transform : none | uppercase | lowercase | capitalize

text-transform : none;	This is a paragraph(none)
text-transform : capitalize;	This Is A Paragraph(capitalize)
text-transform : uppercase;	THIS IS A PARAGRAPH(UPPERCASE)
text-transform : lowercase;	this is a paragraph(lowercase)

white-space

공백 문자를 처리하는 방법
pre:입력한 형태 그대로 브라우저에 렌더링
nowrap:무조건 한 줄로 출력됨.

white-space : normal | nowrap | pre

리스트 꾸미기

list-style-type

목록을 생성할 때 기본적으로 제공되는 마커를 변경하고자 할 때 사용합니다.

list-style-type: none | disc | circle | square | decimal | decimal-leading-zero | upper-alpha | lower-alpha |upper-roman | lower-roman | upper-latin | lower-latin | lower-greek | armenian | georgian | inherit

list-style-type: none;
list-style-type: circle;

list-style-image

목록 마커 대신 이미지로 변경할 때 사용합니다.

list-style-image: url() | none | inherit

list-style-image : url(images/icon.gif);

block -> 세로 레이아웃
* 디폴트 스타일 : 크기, 여백 -> reset 사용
- 디폴트 가로 크기 : 부모 요소에 꽉참
~~blobk 안 : inline 가능 / block 가능 E) ul>li / dt > dt, dd~~
**예외 : hn과 p 안에는 block 요소가 들어갈 수 없음

inline : text 단어의 성격을 가짐
- 디폴트 가로 크기 : 콘텐츠(text, img)의 크기
-> 크기 제어를 할 수 없음

html : element + attribute + text : node

inline-block : img, video
- main = inline / sub = block
-> 크기 제어 가능 + 외부 여백(margin) 세로문제 <- 브라우저 모듈 에러)

Box Model

박스모델은 블록박스와 인라인 박스가 화면에 표시되는 영역과 다른 박스와 의 배치등에 관한 내용으로 주요한 속성은 크기, 테두리, 여백에 관한 것들이 있다.

안의 여백: padding

contents box와 border 사이의 여백

* border 없애기
border-width = 0
border-style = none
border = 0
border = none

width, height

Block 요소 또는 블록 성격을 가진 인라인 요소의 콘텐츠가 들어갈 가로와 세로의 길이이다.

width:길이 | 퍼센트 | auto

height:길이 | 퍼센트 | auto

default : 콘텐츠 크기

width:300px; height:500px;

border

* backgroud : border에서 border

- 콘텐츠의 외곽선을 그려줌
- 선의 두께와 선의 스타일, 그리고 칼라 값이 모두 같다면 한 번에 모두 지정할 수 있고,
- 만약에 서로 다르다면 속성에 방향을 같이 지정하여 정할 수 있다.
- border-width와 style이 지정되어야 기본적인 border가 완성된다.(이때는 color는 #000이 지정된다.)

1. border-width:외곽선의 굵기를 지정한다.
- 외곽선의 굵기를 지정한다.

2. border-style:외곽선의 모양을 지정할 수 있다.

auto 값 : 속성에 따라 기준이 달라짐, 브라우저 자동값 / 항상 default 값과 같지는 않음
E) width : auto -> 부모 요소에 꽉참
E) padding : auto -> 0
E) margin : 요소에 따라 다름

- 외곽선 종류:solid, dotted, dashed,double,groove,ridge,inset,outset,none,hidden

3. border-color:외곽선의 색상을 설정한다.

4. **Border : width style color**

- border의 단축형이다. 순서가 중요하다.

방향정하기 1

- 여러 속성에 방향을 표시하여 각 방향 마다 따로 값을 설정할 수 있다.
- 방향을 나타내는 명칭을 top, bottom, right, left이다.

속성명 1단어 -> 속성명 - 방향
속성명 2단어 -> 속성명1 - 방향 - 속성명2

1

border-width -> border-top-width
margin -> margin-left
border -> border-right
border-color -> border-top-color

border-radius에 사용

방향정하기 2 : 단축형 사용하기

- 하나의 속성에 방향을 달리 설정하는 방법
-

- 네 개가 모두 다른 값일 경우(top, right, bottom, left)
  margin:5px 0 15px 20px;
- left와 right의 값이 같을 경우(top, right과 left, bottom)
  margin:5px 10px 15px;
- left=right, top=bottom이 같을 경우는 2개로 단축한다.(top과 bottom, right과 left)
  margin:5pt 10pt;
- 네 방향이 모두 같을 경우 한 개로 단축한다.(모두 같음)
  margin:5pt;

E) 10 20 30 20 -> 10 20 30 으로 생략 가능
E) 10 20 10 30 -> 생략 불가능

---

padding

가스타일 : 0 -> 단위 생략 가능
*예외: color의 % 값은 0 생략 불가

- 콘텐츠의 안쪽 여백으로 요소의 border를 기준으로 안쪽에 생긴다.
- width와 height값에 영향을 준다.'

padding:길이 , % , auto , inherit

margin

- 블록 요소와 블록 오소 사이를 구분하는 빈 공간
- border를 기준으로 바깥쪽 여백이다.
- 투명으로 채워져 있으며, width와 height값에 영향을 주지 않는다.

margin:길이 , % , auto , inherit

margin-right: auto;
margin-left: auto;
전체 외부영역에서 좌우 반반으로

margin : auto; = margin : 0 auto;

margin 겹침 현상(margin collapsing) !!문제!!

두개 이상의 margin값이 세로 방향으로 만났을 때 하나의 margin으로 합쳐지는 현상.

-> 방향을 한 방향으로속만 계속
E) right 만 계속 / left 만 계속
E) top 만 계속 / bottom 만 계속
--> bottom으로 밑으로 내리는게 편함

div -> padding-top
p -> margin-top : 문제가 생기는 경우가 많음

*세로 : 안의 여백이 유리
* 가로 : 안의 여백, 밖의 여백

width와 height값 정하기

padding-left	padding-right	border-left	border-right	width
20px	50px	5px	45px	
50px	10px	15px	25px	

70px	25px	8px	12px	
35px	45px	10px	6px	

크기 기준 정하기

- width 속성과 height 속성은 글자를 감싸는 영역의 크기를 지정하는 속성이다. box-sizing 속성은 이러한 공식을 변경할 수 있는 CSS3 속성이다.

width, height 기본
box의 기준 변경

box-sizing 디폴트 = content-box;
-> border-box로 변경하여 기준 변경할 수 있음

블록요소와 인라인 요소

요소의 분류:

- 의미와 역할
- 성격

콘텐츠로 분리

HTML 문서에 의해 제공되는 콘텐츠는 크게 7개의 카테고리로 분류하고 있으며, 각 요소는 하나 이상의 카테고리에 포함될 수 있습니다(일부 요소는 해당 카테고리가 없다).

카테고리	설명	해당 요소
Metadata content	콘텐츠의 표현이나 동작 또는 다른 문서와의 관계를 설정하기 위한 콘텐츠	base, command, link, meta, noscript, script, title
Flow content	문서의 본문 영역에 나타나는 콘텐츠	a, abbr, address, area, article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, data, datalist, del, details, dfn, dialog, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hr, i, iframe, img, input, ins, kbd, keygen, label, link, main, map, mark, math, menu, meta, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style, sub, sup, svg, table, template, textarea, time, u, ul, var, video, wbr, text
Sectioning content	영역을 정의하기 위한 콘텐츠	article, aside, nav, section
Heading content	제목을 정의하기 위한 콘텐츠	h1, h2, h3, h4, h5, h6

Phrasing content	문단을 이루는 콘텐츠	a, abbr, area, audio, b, bdi, bdo, br, button, canvas, cite, code, data, datalist, del, dfn, em, embed, i, iframe, img, input, ins, kbd, keygen, label, link, map, mark, math, meta, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, template, textarea, time, u, var, video, wbr, text Embedded content 외부 리소스로 삽입되는 콘텐츠 audio, canvas, embed, iframe, img, math, object, svg, video
Interactive content	사용자와의 상호 작용을 위한 콘텐츠	a, audio, button, details, embed, iframe, img, input, keygen, label, object, select, textarea, video

Inline과 block로 분류하기

inline과 block요소의확인

같은 요소를 두번 이상 입력할 때 세로로 렌더링 되면 블록요소이고,같은 요소를 두번 이상 입력할 때 가로로 렌더링 되면 인라인요소이다.

설명

설명

<a>설명</a> <a>설명</a>

1. 블록 요소 (Block Elements)

성격:

- 줄을 바꿔 각각 **독립된 줄에 표시**
- 자신만의 크기와 여백을 가지고 있다.(내부여백, 외부여백)
- 인라인 요소와 텍스트 혹은 또 다른 블록 요소를 포함할수 있다.

예외: p와 hn요소는 inline요소만 포함할 수 있다

크기:

- default값으로 width : auto, height : auto값이다.
- width의 auto값은 부모요소에 맞춰지는 값이고
- height값은 콘텐츠에 맞추어 진다.

종류:

- h1~h6, p, div, ul, ol, li 등

2. 인라인 요소 (Inline Elements)

예외: a는 block 요소를 포함할 수 있다
-> 미리 display로 성격변화를 하는 것이 좋다

성격:

- 다른 인라인 엘리먼트와 같이 가로 줄에 표시
- 인라인 요소는 크기 개념이 적용이 안되고 세로 외부 여백이 적용 안되다.
- 또 다른 인라인 요소와 텍스트를 포함하지만 **블록 요소를 포함할 수는 없다.**
- 인라인 요소와 텍스트는 반드시 블록 레벨 요소에 포함시켜 나타내야 합니다.
- 즉 BODY 요소의 바로 하위에는 블록 레벨 요소만 자식 요소로 올 수 있고, 인라인 요소를 BODY 요소의 직접적인 하위 내용으로 하는 것은 옳지 않다.

크기:

- default값으로 width : auto, height : auto값이다.
- width의 auto값,height의 auto값 모두 콘텐츠에 맞추어 진다.

종류:

- a, strong, span, em, i, b, span 등

2. 인라인 블록 요소 (Inline-block Elements)

성격:

- 인라인 요소로 블록의 성격을 조금가지고 있다.
- 외부여백과 크기 조절이 가능하다.

크기:

- default값으로 width : auto, height : auto값이다.
- width의 auto값,height의 auto값 모두 콘텐츠에 맞추어 진다.

종류:

- img, input, select, texture 등

부모 요소와 자식 요소

- 요소는 경우에 따라 하나 이상의 다른 요소를 포함할 수 있다. 부모 요소(parent element)는 다른 요소를 포함하고 있는 요소이며, 자식 요소(child element)는 부모 요소에 포함된 요소를 말합니다.
- 위계적으로 보면 부모 요소는 상위 레벨의 요소, 자식 요소는 하위 레벨의 요소라고 볼 수 있습니다.
- HTML 문서에서 최상위 레벨의 요소를 루트 요소(root element)라고 부르는데 <html>요소가 이에 해당합니다.
- 루트 요소는 자식 요소만 가질 수 있으며, 모든 요소의 부모 요소가 됩니다.
- 포함 관계는 부모-자식 관계
- 병렬 관계는 형제 관계를 이루게 됩니다.
- 부모 요소에 적용된 속성은 자식 요소로 상속되며 자식 요소에 적용된 속성은 부모 요소로 역상속 되지는 않습니다.

브라우저에서 box 속성 확인하기

default값: 요소의 기본 속성
auto값: 자동으로 설정되는 값, 각 요소마다 자신의 값의 기준이 틀려진다.

위치지정하기

선택자 -> float, position
영역 -> flex, grid

float

어느 시점부터 float로 할지가 중요!

- 물 위에 요소가 떠 있는 듯 위치가 지정된다.
- 왼쪽과 오른쪽으로 밖에 움직이지 않는다.
- 주변 박스의 위치에 영향을 준다. 그래서 영향을 받는 요소에 문제점이 있을 때 해결하여야 한다.

float문제 해결

- float는 float로 막는다.
- float가 끝나는 시점에 다음에 오는 요소에 clear속성 넣기

float로 인해 부모요소에 일어나는 문제 해결

- float를 자식으로 가지고 있는 부모요소에 overflow속성 hidden넣기
- float를 자식으로 가지고 있는 부모요소에 height값 넣기

float : left | right | none

clear

플로트 된 요소 다음에 오는 요소에게 플로트의 영향에서 벗어나서 자신이 가진 속성에 따라 구현 되도록 해준다
블록 요소나 블록 요소 대체할 수 있는 요소에 지정해야 한다

clear:none | left | right | both

overflow

선택자의 크기와 위치를 확인할 수 있는 성격을 가지고 있다
자식이나 종속 요소의 크기와 위치까지도 감지할 수 있는 성격을 가지고 있다

- 부모 box요소가 자식의 크기를 감지하고
- 만약 자식(콘텐츠)요소이 부모요소보다 크기보다 더 클 경우 상황을 처리 하고자 할 때 사용 합니다.

overflow:visible | hidden | scroll | auto |overflow-x | overflow-y

overflow : visible; -> default 값
overflow : hidden;

scroll -> 상하 & 좌우
auto -> 둘 중에 알아서 함

<RESET>
1. 파일 이용 : 기본
2. CDN : 공개된 주소 링크
* RESET을 넣는 순서 중요! 제일 앞에!

작업 절차
1. 박스 크기 결정
2. 박스 위치 결정
3. 꾸미기 - 배경, 폰트 등등

font-weight는 부모에서 자식으로 상속이 안됨

position

- 좌표값으로 위치를 설정한다.(좌표점이 필요하다)
- 공중에 부양되어서 위치가 이동 되기 때문에 층이 생긴다.

static

- 요소이 본래 성격으로 자리 배치(블록 요소는 블록 요소성격에 맞게, 인라인 요소는 인라인 요소의 성격에 맞게 배치가 된다.)

relative absolute를 위해 빌려주는 용도로 많이 씀

- 선택요소의 자신을 기준으로 자리를 이동한다.
- 선택요소의 원래 있어야 할 곳의 자리에는 다른 요소가 들어 갈 수 없다.
- 자식요소 중 absolute요소가 있으면 선택요소를 기준으로 좌표점이 기준으로 위치를 잡는다.

absolute :

- 부모나 조상 요소중 position : relative, 또는 fixed가 있으면 그 요소를 기준으로 좌표값 이동을 한다. 없으면 body 기준

fixed :

- 스크린을 기준으로 좌표점 이동을 한다.

position : static | relative | absolute | fixed | inherit

좌표점 지정하기

- 가로의 좌표점을 left, right속성 중 하나를 설정하가.
- 세로의 좌표점은 top, bottom속성 중 하나를 설정한다.

선택이 안된 쪽은 auto 값이 들어감

left : 위치;

right: 위치;

bottom:위치 ;

top:위치;

position:absolute; top:150px; left:200px;

z-index

- **layer를 가진 요소** 박스가 겹쳐지는 순서를 지정할 때 사용 합니다.
- 수치가 작을수록 아래쪽에 있다는 것을 의미함.
- body는 기준이 0값이다.

z-index : auto | 정수(1, 2, 3 …); -999999 ~ 999999

만약에 순서를 정하지 않는 경우 **박스의 생성 순서에 따라 정수로 할당 된다.

위치지정하기3: display

display

요소가 가지고 있는 박스의 성격 변경할 때 사용 합니다.

고전 방식 기법

inline요소를 block요소로, block요소를 inline요소로 변경가능
none :요소의 존재를 없애줌. 스크린 리더 등에서 접근이 안 될 수 있다.
block: 인라인 요소를 블록 요소로 만들어 줌
inline : 블록요소를 인라인 요소로 만들어 줌
inline-block:inline요소처럼 배치되면서 block요소의 속성(width, height, margin,padding)들을 사용할 수 있는 속성

```
display : none | block | inline | inline-block | inherit
```

```
p{display:inline;}
```

Flexbox

Flexbox와 Grid 비교

- Flexbox와 Grid는 모두 이전 기술보다 더 진보 된 레이아웃을 만드는 데 사용할 수있는 새로운 레이아웃 모듈 입니다 . 유사한 구문 과 일부 동일한 속성 을 공유하지만 각각 고유 한 특성이 있습니다
  -

Flexbox	Grid layout
항목이 단일 축에 정렬	동시에 행과 열로 배열 할 수 있으므로 2 차원으로 간주

Flexbox

Flexbox는 요소 를 정렬 하고 크기 를 조정 하고보다 유연한 레이아웃을 만든다.

용어와 구조

flex-container: 기준이 되는 부모 요소

display, flex-flow,

justify-content, flex-direction, flex-wrap, align-content, align-items

flex-items:flex-container 내의 직접적인 자식 요소

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

flex-container

flex-items

flex-container 속성

display	Flex Container를 정의
flex-flow	flex-direction와 flex-wrap의 단축 속성
flex-direction	Flex Items의 주 축(main-axis)을 설정
flex-wrap	Flex Items의 여러 줄 묶음(줄 바꿈) 설정
justify-content	주 축(main-axis)의 정렬 방법을 설정
align-content	교차 축(cross-axis)의 정렬 방법을 설정(2줄 이상)
align-items	교차 축(cross-axis)에서 Items의 정렬 방법을 설정(1줄)

display:flex, inline-flex

display: flex;로 지정된 Flex Container는 Block 요소와 같은 성향(수직 쌓임)을 가지며,

display: inline-flex로 지정된 Flex Container는 Inline(Inline Block) 요소와 같은 성향(수평 쌓임)을 가집니다.

flex-flow: direction wrap

단축 속성으로 Flex Items의 주 축(main-axis)을 설정하고 Items의 여러 줄 묶음(줄 바꿈)도 설정합니다.

값	의미	기본값
flex-direction	Items의 주 축(main-axis)을 설정	row
flex-wrap	Items의 여러 줄 묶음(줄 바꿈) 설정	nowrap

flex-direction

Items의 주 축(main-axis)을 설정합니다.

값	의미	기본값
row	Itmes를 수평축(왼쪽에서 오른쪽으로)으로 표시	row
row-reverse	Items를 row의 반대 축으로 표시	
column	Items를 수직축(위에서 아래로)으로 표시	
column-reverse	Items를 column의 반대 축으로 표시	

***축(main-axis)과 교차 축(cross-axis)

값 `row`는 Items를 수평축으로 표시하므로 이때는 주 축이 수평이며 교차 축은 수직이 됩니다.

반대로 값 `column`은 Items를 수직축으로 표시하므로 주 축은 수직이며 교차 축은 수평이 됩니다.

즉, 방향(수평, 수직)에 따라 주 축과 교차 축이 달라집니다.

시작점(flex-start)과 끝점(flex-end)

시작점(flex-start)과 끝점(flex-end)이라는 개념도 있습니다.

이는 주 축이나 교차 축의 시작하는 지점과 끝나는 지점을 지칭합니다.

flex-start와 flex-end를 사용하는데 이는 방향에 맞는 그 시작점과 끝점을 의미합니다.

Flex-wrap

:Items의 여러 줄 묶음(줄 바꿈)을 설정합니다.

기본적으로 Items는 한 줄에서만 표시되고 줄 바꿈 되지 않습니다.

이는 지정된 크기(주 축에 따라 width나 height)를 무시하고 한 줄 안에서만 가변합니다.

Items를 줄 바꿈 하려면 값으로 wrap을 사용해야 합니다.

의미	설명	기본값
nowrap	모든 Itmes를 여러 줄로 묶지 않음(한 줄에 표시)	nowrap
wrap	Items를 여러 줄로 묶음	
wrap-reverse	wrap-reverse	

justify-content

주 축(main-axis)의 정렬 방법을 설정합니다.

값	의미	기본값
flex-start	Items를 시작점(flex-start)으로 정렬	flex-start
flex-end	Items를 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	

space-between	시작 Item은 시작점에, 마지막 Item은 끝점에 정렬되고 나머지 Items는 사이에 고르게 정렬됨	
space-around	Items를 균등한 여백을 포함하여 정렬	

align-content

교차 축(cross-axis)의 정렬 방법을 설정합니다.

주의할 점은 flex-wrap 속성을 통해 Items가 여러 줄(2줄 이상)이고 여백이 있을 경우만 사용할 수 있습니다.

****Items가 한 줄일 경우 align-items 속성을 사용하세요.

값	설명	기본값
stretch	Container의 교차 축을 채우기 위해 Items를 늘림	stretc
flex-start	Items를 시작점(flex-start)으로 정렬	
flex-end	Items를 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	

space-between	시작 Item은 시작점에, 마지막 Item은 끝점에 정렬되고 나머지	
space-around	Items를 균등한 여백을 포함하여 정렬	

값 stretch는 교차 축에 해당하는 너비(속성 width 혹은 height)가 값이 auto(기본값)일 경우 교차 축을 채우기 위해 자동으로 늘어납니다.

align-items

교차 축(cross-axis)에서 Items의 정렬 방법을 설정합니다.

Items가 한 줄일 경우 많이 사용합니다.

주의할 점은 Items가 flex-wrap을 통해 여러 줄(2줄 이상)일 경우에는 align-content 속성이 우선합니다.

따라서 align-items를 사용하려면 align-content 속성을 기본값(stretch)으로 설정해야 합니다.

값	의미	기본값
stretch	Container의 교차 축을 채우기 위해 Items를 늘림	stretch
flex-start	Items를 각 줄의 시작점(flex-start)으로 정렬	
flex-end	Items를 각 줄의 끝점(flex-end)으로 정렬	
center	Items를 가운데 정렬	
baseline	Items를 문자 기준선에 정렬	

Flex Items

속성	의미
order	Flex Item의 순서를 설정
flex	flex-grow, flex-shrink, flex-basis의 단축 속성
flex-grow	Flex Item의 증가 너비 비율을 설정
flex-shrink	Flex Item의 감소 너비 비율을 설정
flex-basis	Flex Item의 (공간 배분 전) 기본 너비 설정
align-self	교차 축(cross-axis)에서 Item의 정렬 방법을 설정

order

Item의 순서를 설정합니다.

Item에 숫자를 지정하고 숫자가 클수록 순서가 밀립니다.

음수가 허용됩니다.(HTML 구조와 상관없이 순서를 변경할 수 있기 때문에 유용합니다.)

값	의미	기본값
숫자	Item의 순서를 설정	0

flex-grow

Item의 증가 너비 비율을 설정합니다.

숫자가 크면 더 많은 너비를 가집니다.

Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없습니다.

의미	기본값	
숫자	Item의 증가 너비 비율을 설정	0

모든 Items의 총 증가 너비(flex-grow)에서 각 Item의 증가 너비의 비율 만큼 너비를 가질 수 있습니다.

예를 들어 Item이 3개이고 증가 너비가 각각 1, 2, 1이라면,

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 됩니다.

flex-shrink

Item이 감소하는 너비의 비율을 설정합니다.

숫자가 크면 더 많은 너비가 감소합니다.

Item이 가변 너비가 아니거나, 값이 0일 경우 효과가 없습니다.

값	의미	기본값
숫자	Item의 감소 너비 비율을 설정	1

감소 너비(flex-shrink)는 요소의 너비에 영향을 받기 때문에 계산하기 까다롭습니다.

영향을 받는다는 요소의 너비는 width, height, flex-basis 등으로 너비가 지정된 경우를 의미합니다.

Container의 너비가 줄어 Items의 너비에 영향을 미칠 경우, 영향을 미치기 시작한 지점부터 줄어든 거리 만큼 감소 너비 비율에 맞게 Item의 너비가 줄어듭니다.

flex-basis

Item의 (공간 배분 전) 기본 너비를 설정합니다.

값이 auto일 경우 width, height 등의 속성으로 Item의 너비를 설정할 수 있습니다.

하지만 단위 값이 주어질 경우 설정할 수 없습니다.

값	의미	기본값
auto	가변 Item과 같은 너비	auto
단위 px, em, cm 등 단위로 지정		

flex

Item의 너비(증가, 감소, 기본)를 설정하는 단축 속성입니다.

속성	설명	기본값
flex-grow	Item의 증가 너비 비율을 설정	0
flex-shrink	Item의 감소 너비 비율을 설정	1
flex-basis	Item의 (공간 배분 전) 기본 너비 설정	aut

flex: 증가너비 감소너비 기본너비;

```
.item {
flex: 1 1 20px;  /* 증가너비 감소너비 기본너비 */
flex: 1 1;  /* 증가너비 감소너비 */
flex: 1 20px;  /* 증가너비 기본너비 (단위를 사용하면 flex-basis가 적용됩니다) */
}
```

align-self

교차 축(cross-axis)에서 개별 Item의 정렬 방법을 설정합니다.

align-items는 Container 내 모든 Items의 정렬 방법을 설정합니다.

필요에 의해 일부 Item만 정렬 방법을 변경하려고 할 경우 align-self를 사용할 수 있습니다.

이 속성은 align-items 속성보다 우선합니다.

값	의미	기본값
auto	Container의 `align-items` 속성을 상속받음	auto
stretch	Container의 교차 축을 채우기 위해 Item을 늘림	
flex-start	Item을 각 줄의 시작점(flex-start)으로 정렬	
flex-end	Item을 각 줄의 끝점(flex-end)으로 정렬	
center	Item을 가운데 정렬	
baseline	Item을 문자 기준선에 정렬	

성격변화하기

display

요소가 가지고 있는 박스의 성격(블록 요소 또는 인라인 요소)을 변경할 때 사용 합니다.
inline요소를 block요소로, block요소를 inline요소로 변경가능
none :요소의 존재를 없애줌. 스크린 리더 등에서 접근이 안 될 수 있다.
block: 인라인 요소를 블록 요소로 만들어 줌
inline : 블록요소를 인라인 요소로 만들어 줌
inline-block:inline요소처럼 배치되면서 block요소의 속성(width, height, margin)들을 사용할 수 있는 속성

```
display : none | block | inline | inline-block | inherit
```

```
p{display:inline;}
```

visibility

요소를 표시하거나 감출 때 사용한다.
display:none -> 요소가 없는 것
visibility:hidden -> 보이지 않는 것

```
visibility: visible | hidden | inherit;
```

```
p{visibility:hidden;}
```

Color와 Background

background-color

요소의 배경 색상을 지정할 때 사용 합니다.
default값은 transparent(투명)이다.

background-color:색상명 | RGB 16진수 | RGB 10진수 | RGB 백분율 | inherit

background-color : red ;
background-color : #ff0000 ;
background-color : rgb(255,0,0) ;
background-color : rgb(100%,0%,0%) ;

background-image

-> default : repeat : repeat / position : 0 0 / attachment : scroll

요소의 배경 이미지를 지정할 때 사용 합니다.

background-image:none | uri(경로명) | inherit

background-image: none ;
background-image: url(images/bg.gif) ;

background-repeat

요소에 지정한 배경 이미지의 반복 여부를 변경하고자 할 때 사용 합니다.
repeat:x,y 축 모두 반복
repeat-x:x축만 반복
repeat-y:y축만 반복
no-repeat:반복 없어.

background-repeat:repeat | repeat-x | repeat-y | no-repeat | inherit

background-image: url(images/bg.gif) ;background-repeat:no-repeat;
background-image: url(images/bg.gif) ;background-repeat:repeat-y;

repeat-x repeat-y repeat no-repeat

기준 : background-position / background-clip / background-origin

background-position

요소에 지정한 배경 이미지의 위치를 변경하고자 할 때 사용 합니다.

background-position:가로 세로

	가로	세로
키워드	left, right	top, bottom
	center	center
길이	px	px
상대값	%, em	%, em

background-image: url(images/bg.gif) ;background-position:left top;
background-image: url(images/bg.gif) ;background-repeat:100px 200px;

background-position: x y ;

background-position: x y ; background-repeat:repeat-x;

background-attachment

요소에 지정된 배경 이미지를 스크롤되어도 항상 같은 위치에 고정시키고자 할 때 사용합니다.
scroll:스크롤 적용
fixed: 지정한 스크린 위치에 그대로 남음.

background-attachment:scroll | fixed | inherit

background-image: url(images/bg.gif) ;
background-attachment:fixed;

background

디폴트값은 생략가능 / 순서 중요

font 속성과 같이 배경과 관련된 다섯 가지 속성을 단축해서 한 번에 선언하고자 할 때 사용합니다.

background:color | image | repeat | position | attachment | inherit

background: yellow url(images/bg.gif) no-repeat left top fixed ;

background Image중첩사용

1. background-image
- 중첩사용이 가능하다.

background:url(img/coforwardQr2.png);
background:url(img/checkBox1.png) left top no-repeat,
url(img/checkbox2.png) right top no-repeat,
url(img/coforwardText.png);background-size

- IE8이하 사용 불가

background-color : 다중 X

background-image : url(1), url(2), url(3);
background-repeat : repeat1, repeat2, repeat3;
background-position : posx1 posy1, posx2 posy2, posx3 posy3;

background : url(1) repeat1 posx1 posy1, (위)
url(2) repeat2 posx2 posy2,
url(3) repeat3 posx3 posy3, (아래)
color

background-size : [x] [y];

auto 값 = 이미지값

[x]% [y]%	적용되는 요소의 크기에 비례하여 배경 이미지 적용	기준 : 선택자

[x]px [y]px	절대 크기로 배경 이미지 적용
cover	배경 이미지를 늘여 적용되는 요소 전체에 표시
contain	배경 이미지의 가로 세로 비율을 맞춰 요소에 표시할수 있는 최대 크기로 표시

가로를 기준으로 100%

```
#box_rel{background-size:40% 70%;}
#box_abs{background-size:50px 50px;}
#box_cover{background-size:cover;}
```

배경
- 기준 : padding-box
- 영역 : border-box

정사각형의 이미지를
준비하는 것이 좋음

background-origin -> 기준
: defalut = border-box
background-clip -> 영역
: default = padding-box

이 중 선택
1. content-box
2. padding-box
3. border-box

접두어 사용하기(vendor Prefix)

벤더 프리픽스는 웹 브라우저 공급 업체(마이크로소프트, 구글, 모질라, 애플, 오페라)에서 제공하는 실험적인 기능을 사용할 때 사용한다.
예를 들어 변환 속성은 CSS3 표준에 있지만 아직 완벽하게 제정된 상태가 아니다. 따라서 웹 브라우저 업체가 무턱대고 속성을 추가할 수 없다.
하지만 다른 웹 브라우저를 이기려면 새로운 기능을 모두 제공해야 하므로 벤더 프리픽스를 사용해 지원을 하는 것이다..
각 웹 브라우저마다 다음과 같은 벤더 프리픽스를 갖는다. firefox

사용 방법

기본형

-webkit-boxshadow
-moz-boxshadow
boxshadow

script 사용 -> 모바일에 랙이 걸림

prefix.js

gradient background-image: -벤더 프리픽스-종류-gradient()

- 아직 실험단계에 있는 대표적인 속성이다. 각 브라우저 마다 형식이 달리 표현되며, 요즘에 와서 조금 형식이 통일되기 시작했다.
- 절대로 중요한 부분에서는 사용하지 말아야 한다. 브라우저마다 사양이 많이 차이난다.

linear형식

background: linear-gradient(방향, 칼러 위치, 칼러 위치, 칼러 위치);

방향: to 방향, to 방향1 방향2, 방향각도 deg사용 default : 위에서 아래로

linear-gradient(0deg, skyblue, yellow);

linear-gradient(45deg, skyblue, yellow);

linear-gradient(90deg, skyblue, yellow);

linear-gradient(180deg, skyblue, yellow);

linear-gradient(skyblue, white);

linear-gradient(to right, skyblue, white);

linear-gradient(to right, skyblue 5%, green 20%, white);

background: -webkit-linear-gradient(45deg, red 10%,yellow 65%,#666 85%, blue 95%); /* For Safari 5.1 to 6.0 */

background: -moz-linear-gradient(45deg, red 10%,yellow 65%,#666 85%, blue 95%); /* For Firefox 3.6 to 15 */
background: linear-gradient(45deg, red 10%,yellow 65%,#666 85%, blue 95%); /* Standard syntax (must be last)

radial형식

background: radial-gradient(모양, 칼러 위치, 칼러 위치, 칼러 위치);

background: -webkit-radial-gradient(red 5%, green 15%, blue 60%); /* Safari 5.1 to 6.0 */
background: -moz-radial-gradient(red 5%, green 15%, blue 60%); /* For Firefox 3.6 to 15 */
background: radial-gradient(red 5%, green 15%, blue 60%); /* Standard syntax (must be last) */

default

모양: circle, ellipse

radial-gradient(green 5%, yellow 25%, white 50%)

radial-gradient(circle, green 5%, yellow 25%, white 50%)

모양 at 위치: background-position과 비슷하다.

radial-gradient(circle at 30%, green 5%, yellow 25%, white 50%)

radial-gradient(circle at left top, green 5%, yellow 25%, white 50%)

radial-gradient(circle at 60% bottom, green 5%, yellow 25%, white 50%)

참고 사이트 https://chrome.google.com/webstore/search/colorzilla
http://www.colorzilla.com/

FONT(@font-face)

- 브라우저 사용자의 컴퓨터에 폰트가 설치돼 있지 않아도 사용자의 컴퓨터등에서 내려 받을 수 있는 경로를 지정하여 해당 폰트를 사용할 수 있게 하는 방법

연결 -> @font-face{ } : 1폰트 1font face
사용 -> font-family;

cdn 사용 : 구글 제공 폰트 사용

- **사용 가능 글꼴 파일 형식**

truetype(ttf)
opentype(otf),
embedded-opentype(eot)
scalable-vector-graphic(svg, svgz)

사용법

1. 지정: @font-face{ font-family:'폰트이름';
   src : url('폰트경로') format('포맷이름');
   src : url('폰트경로') format('포맷이름'),
   url('폰트경로') format('포맷이름'); }

브라우저마다 사용가능한 폰트가 다르므로 여러 경로 연결

2. 사용: 선택자{fontfamily: '폰트이름', 대표페밀리체명;}

```
@font-face {
cont-family: 'ChunkFiveRegular';
src:        url('fonts/chunkfive.eot');
src:        url('fonts/chunkfive.eot') format('embedded-opentype'),
            url('fonts/chunkfive.woff') format('woff'),
            url('fonts/chunkfive.ttf') format('truetype'),
            url('fonts/chunkfive.svg') format('svg');        }
```

embedded-opentype
truetype

브라우저 \ 포맷	eot	woff	ttf / otf	svg
크롬(모두)				●
크롬 6+		●	●	●
파이어폭스 3.5			●	
파이어폭스 3.6+		●	●	
IE 5 – 8	●			
IE 9+	●	●	●	
오페라 10+			●	●
사파리 3.1+			●	●
iOS 4.2 미만				●
iOS 4.2+			●	●

국내 무료서체: 서울서체, 제주서체,skt 뫼비우스테, 아모레퍼스픽 아리따체
구글 폰트 사용하기: https://www.google.com/fonts/earlyaccess
폰트 컴버트 사이트: http://www.fontsquirrel.com Shadow

box-shadow

box-shadow:[dx] [dy] [bulr radius] [spread] [칼라] [set];

- 지정된 요소의 그림자를 넣어준다.
- 블록 요소 뿐 아니라 이미지등에도 사용할 수 있다.

dx	그림자의 가로방향 위치
dy	그림자의 세로방향 위치
blur	흐려짐 정도
spread	번짐 정도
set	그림자의 형태 outset : 요소 바깥쪽으로 그림자가 떨어짐 (기본값) inset : 요소 안쪽으로 그림자가 떨어짐

box-shadow:10px 20px 10px 5px #666666 inset;

text-shadow

text-shadow:[dx] [dy] [bulr] [color];

- 텍스트에 그림자를 지정함
- 속성

dx	그림자의 가로방향 위치
dy	그림자의 세로방향 위치
bulr	흐려짐 정도

text-shadow:3px 3px 5px #000;

**여러개의 그림자 사용하기:text-shadow:3px 3px 5px #000,3px 43px 3px #FC0;

border2

border-radius

- **Box Model 외각선을 부드럽게 해 준다.**

1. 기본형:

border-radius:x1 x2 x3 x4/y1 y2 y3 y4;

border-radius:20px 30px 5px 15px/45px 23px 34px 5px;

2. 각 방향 다르게 설정하기

border-top-left-radius
border-top-right-radius
border-bottom-left
border-bottom-right
border-top-left-radius:20px;
border-top-right-radius:20px;
border-bottom-left-radius:20px;
border-bottom-right-radius:20px;
border-radius: 50px 100px 200px 5px;

border-image테두리에 이미지 넣기

1. 속성:

속성	설명
border-image-slice	• 외각선에 지정될 이미지 경로
border-image-slice	테두리에 지정할 이미지가 요소 박스 상하좌우 테두리 영역에 보여질 크기를 지정
border-image-repeat	요소 박스에 지정된 이미지가 반복되는 스타일을 지정 repeat, round, stretch
border-image-width	각 방향별로 border 이미지 표시 영역의 길이를 지정
border-image-outset	테두리에 지정한 이미지가 border 영역에서 벗어나 요소 박스 외부에 표시되는 이미지 영역을 지정
border-image	요소 박스의 테두리 이미지 지정과 관련된 속성을 일괄 적용하는 대표 속성

변형과 움직임

2차원 변환	9	4.0	3.5	3.2	12.1.10.5
3차원 변환	10	12.0	16	4.0	23.0.15.0
애니메이션	10	4.0	5.0	4.0	12.1.12.0

Transform

- 요소의 형태를 변경함

transform function	설명	단위
translate(x,y)	요소의 위치를 X축으로 x만큼, Y축으로 y만큼 이동시킨다.	px, %, em 등
translateX(n)	요소의 위치를 X축으로 x만큼 이동시킨다.	px, %, em 등
translateY(n)	요소의 위치를 Y축으로 y만큼 이동시킨다.	px, %, em 등
scale(x,y)	요소의 크기를 X축으로 x배, Y축으로 y배 확대 또는 축소 시킨다.	0과 양수
scaleX(n)	요소의 크기를 X축으로 x배 확대 또는 축소 시킨다.	0과 양수
scaleY(n)	요소의 크기를 Y축으로 y배 확대 또는 축소 시킨다.	0과 양수
skew(x-angle,y-angle)	요소를 X축으로 x 각도만큼, Y축으로 y 각도만큼 기울인다.	+/- 각도 (deg)
skewX(x-angle)	요소를 X축으로 x 각도만큼 기울인다.	+/- 각도 (deg)
skewY(y-angle)	요소를 Y축으로 y 각도만큼 기울인다.	+/- 각도 (deg)
rotate(angle)	요소를 angle만큼 회전시킨다.	+/- 각도 (deg)

```
#sample_01{transform:속성명(값);}
#sample_01{transform:translate(50px, 30px);}
#sample_02{transform:scale(2);}
#sample_03{transform:rotate(45deg);}
#sample_04{transform:skew(30deg, 30deg);}
```

- **다중입력일 때는 공백으로 구분**

```
#sample_04{transform:skew(30deg, 30deg) rotate(45deg);}
```

- transform-origin

  transform의 기준점을 정해준다.

  transform-origin: 가로 세로

. 3D 트랜스폼 (3D Transform)

3D 트랜스폼은 프로퍼티값으로 변환함수(transform function)를 사용한다. 사용할 수 있는 변환 함수는 다음과 같다.

transform function	설명	단위
translate3d(x,y,z)	요소의 위치를 X축으로 x만큼, Y축으로 y만큼 Z축으로 z만큼 이동시킨다.	px, %, em 등
translateX(n)	요소의 위치를 X축으로 x만큼 이동시킨다.	px, %, em 등
translateY(n)	요소의 위치를 Y축으로 y만큼 이동시킨다.	px, %, em 등
translateZ(n)	요소의 위치를 Z축으로 z만큼 이동시킨다.	px, %, em 등
scale3d(x,y)	요소의 크기를 X축으로 x배, Y축으로 y배, Z축으로 z배 확대 또는 축소 시킨다.	0과 양수
scaleX(n)	요소의 크기를 X축으로 x배 확대 또는 축소 시킨다.	0과 양수
scaleY(n)	요소의 크기를 Y축으로 y배 확대 또는 축소 시킨다.	0과 양수
scaleZ(n)	요소의 크기를 Z축으로 z배 확대 또는 축소 시킨다.	0과 양수
rotate3d(x,y,z)	요소를 X축으로 x각도, Y축으로 y각도, Z축으로 z각도 회전시킨다.	+/- 각도 (deg)
rotateX(x)	요소를 X축으로 x각도 회전시킨다.	+/- 각도 (deg)
rotateY(y)	요소를 Y축으로 y각도 회전시킨다.	+/- 각도 (deg)
rotateZ(z)	요소를 Z축으로 z각도 회전시킨다.	+/- 각도 (deg)

transition

Transition

- **위치 속성** top, left, bottom, right
- **크기 속성** width, height
- **박스 속성** margin, padding
- **테두리 속성** border-width, border-radius, border-color
- **색상 속성** color, background-color
- **투명도 속성** opacity
- **변환 속성** transform

transition-delay	움직이기 이전의 기다리는 시간
transition-property	트랜지션할 속성(none, all)
transition-duration	트랜시션 시간
transition-timing-function	이펙트의 가속조절 (ease-in, ease-out,cubic-bezier, linear)

transition: property duration function delay

transition:background 3s ease;

- transtion 효과는 처음에는 적용안되고 마우스를 올리거나 다른이벤트에 다른 변형(transform) 을 줄때만 발생된다. 그리고 이벤트가 제거되면 처음 변형(transform)값으로 다시 transition 효과가 적용된다.
- 처음부터 transition 효과를 내려면 animation (에니메이션 @keyframes) 효과를 적용하여야 한다.

프로퍼티값	효과	그래프
ease	기본값. 느리게 시작하여 점점 빨라졌다가 느리지면서 종료한다.	
linear	시작부터 종료까지 등속 운동을 한다.	

ease-in	느리게 시작한 후 일정한 속도에 다다르면 그 상태로 등속 운동한다.	
ease-out	일정한 속도의 등속으로 시작해서 점점 느려지면서 종료한다.	
ease-in-out	ease와 비슷하게 느리게 시작하여 느리지면서 종료한다.	

animation (에니메이션)

@keyframes

- transtion 효과와 비슷하지만 transtion은 객체에 직접 적용하지만, animation은 별도의 이름을 가진 keyframe을 생성시켜 프레임 구간별로 다른 속성을 적용한 다음 적용 할 요소에 미리정의된 animation 이름과 여러가지 에니메이션속성을 적용한다.

```
@keyframes 에니메이션이름 {
    from {초기상태속성값}
    키프레임구간위치 % {해당구간 속성값}
    to { 종료상태 속성값}
```

animation

animation: name duration timing-function delay iteration-count direction fill-mode play-state

animation-name	적용할에니메이션(keyframes) 이름
animation-delay	실행전 지연시간
animation-duration	1회실행시간
animation-iteration-count	반복횟수 (infinite 무한반복)
animation-timing-function	가속도형태 (linear일정, ease중간빠르고시작끝은느리게, ease-in끝을빠르게, ease-out끝을느리게, ease-in-out시작끝을느리게)
animation-direction	반복형태 normal 다시처음부터, alternate 역방향재생
animation-fill-mode	애니메이션 미실행 시(종료 또는 대기) 요소의 스타일을 지정한다. (none, forwards, backwards, both)
animation-play-state	애니메이션 재생 상태(재생 또는 중지)를 지정한다.

- 주의) 에니메이션을 사용할때는 움직이기 전에 변화시켜줄 속성의 초기값이 설정되어있어야 한다.
- 처음부터 움직임을 진행할때는 animation 기능을 사용하는게 좋고 , 마우스반응으로 움직일때는 transform과 transtion 효과를 사용하는 것 이좋다.

선택자2

속성선택자

선택자[속성~=값]	속성 안의 값이 특정 값을 단어로 포함하는 태그를 선택
선택자[속성|=값]	속성 안의 값이 특정 값을 단어로 포함하는 태그를 선택
선택자[속성^=값]	속성 안의 값이 특정 값으로 시작하는 태그를 선택
선택자[속성$=값]	속성 안의 값이 특정 값으로 끝나는 태그를 선택
선택자[속성*=값]	속성 안의 값이 특정 값을 포함하는 태그를 선택
[속성 선택자][속성선택자]	

가상 선택자

:first-child	첫 번째에 위치하는 자손을 선택
:last-child	마지막에 위치하는 자손을 선택

:first-letter	첫 번째 글자를 선택
:first-line	첫 번째 줄을 선택

:before	선택자의 앞에 가상 공간을 만든다.
:after	마지막에 위치하는 자손을 선택

content: 원하는 값을 넣어 준다.

이미지: url()

텍스트: “ 내용 “

스크립트: “ “ 없이 넣기

하이퍼 링크태그 관련 선택자

a	하이퍼링크요소
a:link	하이퍼링크에 연결 경로가 있는 경우
a:visited	하이퍼링크 주소에 전에 방문한적 있는 경우 (쿠키에 등록 되어 있는 경우)
a:hover	하이퍼링크에 마우가가 롤 오버인 상태
a:active	하이퍼깅크의 활성 상태

:focus	사용하기 위해 포커스가 맞추어져 있는 상태

일반구조 선택자

:first-child	첫번째 자식요소 선택
:last-child	마지막 자식요소 선택
:nth-child(수열)	자식관계에서 앞에서 수열 번째 요소 선택
:nth-last-child((수열)	자식관계에서 뒤에서 수열 번째 요소 선택

수열

:nth-child(odd)
:nth-child(even)
:nth-child(숫자) :nth-child(3)
:nth-child(@n) :nth-child(3n)

:nth-child(@n+숫자) :nth-child(3n+2)

형태구조 선택자

:first-of-type	형제 관계 중에서 앞에서 첫 번째로 등장하는 요소 선택
:lasst-of-type	형제 관계 중에서 앞에서 마지막으로 등장하는 요소 선택
:nth-of-type(수열)	형제 관계 중에서 앞에서 수열 번째로 등장하는 요소 선택
:nth-last-of-type(수열)	형제 관계 중에서 뒤에서 수열 번째로 등장하는 요소 선택

반응문자선택자

::selection	사용자가 드래그한 글자를 선택

반응 선택자

:hover	마우스가 오버인 상태
:active	마우스가 클릭 한 상태

링크선택자

:link	하이퍼링크에 연결 경로가 있는 경우
:visited	하이퍼링크 주소에 전에 방문한적 있는 경우 (쿠키에 등록 되어 있는 경우)

상태선택자

:checked	체크 상태인 input
:focus	초점이 맞추어진 input
:enabled	사용가능한 input
:disabled	사용 불가능한 input

적용방식

. Inline Style Sheet

- 요소에 직접 스타일 적용
- 태그와 함께 CSS가 코딩되어 있어 CSS분리원칙에 위배됨
- 단, 디버깅용(오류찾기)으로만 사용

〈태그 style="속성:값;"〉

Internal Style Sheet

html문서의 <head></head> 사이에 CSS를 선언하는 방식 (meta 뒤에, script 앞에)마찬가지로, CSS분리원칙에 위배됨)

```
〈head〉
        〈style type="text/css"〉
                selector { 속성 : 값; }
        〈/style〉
〈/head〉
```

External Style Sheet:link사용(가장권장)

<link>요소로 외부 화일 사용하기

```
〈head〉
〈link rel="stylesheet" type="text/css" href="css파일명" /〉
〈/head〉
```

```
@charset "utf-8";

p{
   background: beige;
   color: gray;
}
p span{
   font-weight: bold;
   color: blue;
}
```

External Style Sheet:@import사용하기

여러개의 CSS파일 연결해서 사용

@import url (경로/파일명.css); 또는 @import "파일명.css"; 로 작성

```
〈style〉
   @import url(경로)
〈/style〉
```

```
@charset "utf-8";
@import url(sample.css);
```

반응형사이트만들기

I. 준비하기

viewport 설정하기

i. 모바일 웹 페이지는 화면에 대한 특별한 정보를 제공하려고 뷰포트 meta 태그를 사용한다.

ii. 뷰포트 meta 태그는 브라우저의 화면 설정과 관련된 정보를 제공한다

iii. 뷰포트를 사용하지 않으면 스마트폰의 웹 브라우저는 웹사이트를 자동으로 풀 브라우징해전체 화면으로 보여줍니다.

width	화면의 너비	width=240 width=device-width
height	화면의 높이	height=800 height=device-height
initial-scale	초기 확대 비율	initial-scale=1 initial-scale=2
user-scale	확대 및 축소의 기능 여부	user-scale=no
minimun-scale	최소 축소 비율	minimun-scale=1.0
maximun-scale	최대 축소 비율	maxximun-scale=2.0
target-densitydpi	DPI 지정	target-densitydpi=medium-dpi

*** 일부 모바일 브라우저에서는 *target-densitydpi* 속성을 지원하지 않으므 로 주의한다

<meta	name=" viewport"	속성명
content="	width=device-width,	화면의 넓이:기기의 넓이
	initial-scale=1.0,	초기화면 크기:1배
	maximum-scale=1.0,	최대 화면 크기:1배
	minimum-scale=1.0,	최소 화면크기:1배
	user-scalable=no,	사용자 화면 축소 확대기능: no
	target-densitydpi=medium-dpi	화면 dpi: 중간
" >		

```
<meta name="viewport" content="user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0, width=device-width, target-densitydpi=medium-dpi" >
```

II. 반응형 사이트 만들기

Fluid Grid(유동형 그리드:가변형 그리드)

설명추가

실제 사이즈 / 기준 사이즈 X 100= %

500

120 / 500 X 1000 = 24%

240

20

Flexible Images(가변이미지)

기존의 이미지의 width, height의 값을 지정하는 것에서 탈피하여 이미지를 싸고 있는 block의 가로 사이즈에 의해 이미지의 크기가 결정되는 방식

max-width	최대너비
min-width	최소너비
max-height	최대높이
min-height	최소높이
height	auto

```
img{max-width:100%:}
```

Media Query

웹 페이지가 여러 미디어 맞추어 , 그때 그때 환경에 맞는 스타일을 지정 할 때 사용된다.

<link href="경로" rel="stylesheet" media="미디어 종류">

미디어 종류

미디어 종류	설명
print	인쇄용
projection	프로젝트, 슬라이드쇼
screen	화면
tv	tv같은 형태
braille	시각장애인용 점자물
audio	음성브라우저 기기
all	모든 대상

미디어 쿼리 연산자

not, and, only,or(컴마)등의 연산자를 사용하여 복잡한 미디어 쿼리를 구성할 수 있습니다.

연산자	설명
and	and 연산자는 다중 media features를 묶어서 하나의 미디어 쿼리로 만들게 합니다. 각각의 결과가 참인 경우에만 미디어 쿼리가 실행될 것입니다. @media (min-width: 700px) and (orientation: landscape) { ... } @media tv and (min-width: 700px) and (orientation: landscape) { ... }
not	not 연산자는 미디어 쿼리 전체의 값을 반전시키는데 사용합니다. @media not all and (monochrome) { ... }
only	only 연산자는 쿼리 전체가 참일경우 실행되며, 특히 구버전의 브라우저가 해당 스타일을 적용하지 않게하는데 유용합니다.(not 이나 only 연산자를 사용할 경우 반드시 적확한 미디어 타입을 기재해야 합니다.)
or 또는 컴마	또한 콤마 또는 or를 사용하여 여러개의 미디어 쿼리를 연속으로 나열할 수 있습니다. 나열된 미디어 쿼리의 값중 하나라도 참일 경우, 전체 구문은 참이 됩니다. @media (min-width: 700px), handheld and (orientation: landscape) { ... }

미디어 쿼리 속성

width	출력 매체의 너비의 따라 정의
height	출력 매체의 높이의 따라 정의
device-width	출력 장치의 디스플레이 너비에 따라 정의
device-height	출력 장치의 디스플레이 높이에 따라 정의 @media screen and (max-device-height: 799px)

orientation	화면의 가로 세로 방향에 따라 미디어의 유형을 결정 landscape I portrait @media all and (orientation: portrait) { .}
aspect-ratio device-aspect-ratio	출력 장치 디스플레이의 가로, 세로 비율,또는 픽셀의 비율에 따라 미디어 유형을 정의 @media screen and (min-aspect-ratio: 1/1) { ... } @media screen and (device-aspect-ratio: 16/9), screen and (device-aspect-ratio: 16/10) { ... }
color	색상 표시가 가능한 매체의 색상 깊이 값으로 정의 @media all and (min-color: 4) { ... }
color-index	매체가 표시할 수 있는 색상의 개수로 정의 @media all and (min-color-index: 256) { ... }
monochrome	흑백 매체의 음영 깊이 값으로 정의 @media all and (min-monochrome: 8) { ... }
resolution	매체의 출력 해상도의 값으로 정의 @media print and (min-resolution: 300dpi) { ... }
scan	화면의 주사 방식에 따라 정의
grid	점자 출력기와 같은 물리적 격자 형태의 출력 형태의 유형을 정의

@media 해당미디어 종류명 연산자 (속성명: 값) 연산자 (속성명:값){

스타일 지정

}

```
@media screen and (max-width: 768px) and (min-width:500px){
        body {background-color: orange;}
}
```

새로운 단위

1. vh

부모 요소이 상대적 영향을 받는 단위 체계로 뷰포트에 영향을 받는다.

vh 요소는 높이 값의 *100*분의 *1*단위이다.

만약 브라우저 높이 값이 *700px*이면 *1vh*는 *7px*이 된다.

2. vw

*vh*와 비슷하게 뷰포트의 너비에 영향을 받는다.

만약 브라우저 높이 값이 *600px*이면 *1vw*는 *6px*이 된다.

3. vmin, vmax

뷰포트 너비값과 높이값 중 큰 것을 *vmax*로 작은 것을 *vmin*의 단위로 한다.

만약 브라우저의 크기가 너비 *1200px,* 높이 *800*이라면 *1vmin*은 *8px, 1vmax*는 *12px*이 된다.

너비 *680px,* 높이가 *920px*이라면 *vmin*은 *6.8px, vmax*는 *9.2*가 되게 된다.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile
						3.2-5.1							
6-8			4-19	3.1-5.1		[2,3] 6.1							
[1] 9	[2] 12-15	2-18	[2] 20-25	[2] 6	10-12.1	[3] 7.1		2.1-4.3					
[2] 10	16	19-61	26-68	6.1-11.1	15-54	8-11.2		4.4-4.4.4	7	12-12.1			[2] 10
[2] 11	17	62	69	12	55	11.4	all	67	10	46	69	62	[2] 11
	18	63-64	70-72	TP		12							

Rem

em과 rem은 폰트 크기를 기준으로 브라우저에 의해 픽셀 값으로 변환된다.

em - 지정된 요소의 폰트 크기를 기준으로 픽셀값으로 변환한다. 지정된 요소의 부모 요소의 폰트 크기에 따라 변화가 있음.

디폴트 - body : 16px

root em

rem - html 최상위 요소 폰트 기준으로 픽셀값으로 변환한다. 최상위 요소에 폰트를 지정하지 않으면 브라우저에 설정된 폰트 크기를 상속받는다.

폰트 크기를 지정하거나(px) 꼭 em 단위를 써야하는곳이 아니면 rem단위 사용을 권장한다.

rem 단위를 사용해야 하는 이유.

요소들의 상속 특성과 관계없이 일관된 크기를 설정해주는것뿐 아니라 px단위 대신 rem단위를 사용하면 사용자가 설정한 폰트의 크기에 따라 사이트에 배치된 모든 구성 요소가 적절하게 반응한다.

em단위를 사용하는 이유.

최상위 요소 외의 다른 요소의 폰트 크기에 따라 변하는 특성이있다. 특정한 상황에 따라 개별적으로 적용이 된다. (꼭 필요할 경우에만 사용하는것을 권장함.)