**font**

**글씨스타일 font-style** : normal(기본) / italic(이탤릭)

**글씨 굵기 font-weight** : normal(기본) / lighter(얇게) / bold(두껍게) / 100~900 수치가능 (400이 기본값)

**글씨 크기 font-size** : inherit(상속) / 00px / 0.0rem

**rem; => html에 적용된 글씨크기**

**em; => 부모태그에 적용된 글씨크기**

**글씨 행간 line-height** : 2 \*기본 120%(=1.2) / % 또는 소수점 또는 px입력가능

**글씨체 적용 font-family**: 'Noto Sans Kr', sans serif

\*대안폰트 serif(명조) / san-serif(고딕) / cursive(손글씨)

위 다섯개를 한 줄로 표현가능(순서중요!!)

font: italic bold 70px / 2 'Times New Roman', Times, serif;

---------------------------------------------------------------------------

**밑줄긋기 text-decoration** : none(기본) / underline(밑줄) / linethrough(취소)

**문단정렬 text-align** : left(기본) / center(중앙) / right(오른쪽) / justify(양쪽정렬)

**vertical-align(거의안씀)** : baseline(기본) / top(위) / middle(중앙) / bottom(아래)

**첫줄들여쓰기 text-indent** : 00px

**영소문자/대문자 text-transform** : capitalize(앞글자만대문자) / uppercase(대문자) / lowercase(소문자)

**자간설정 letter-spacing** : 00px

**단어간격 word-spacing** : 00px

**단어쪼개기 word-break** : keep-all(쪼개지않기) / break-all(쪼개기)

**강제로 한줄나열 white-space** : normal(기본) / nowrap(한줄-줄바꿈불가)

**font 등록방법**

<<웹폰트가 아닌경우>>

1. 폰트를 다운받고, 폰트폴더를 만들어 파일을 넣어준다.

2. CSS 파일에 폰트를 등록한다.

@font-face {

font-family: 폰트이름지정; (한글과 띄어쓰기를 넣으려면 ""안에 넣어준다.)

src: url("폰트파일경로"); }

3. 사용할 곳에 font-family : 속성을 넣어준다.

<<웹폰트인 경우>>

1. 웹상의 폰트코드를 복사하고, html헤드 안에 넣어준다.

**관계선택자**

띄어쓰기 : ~안에

> : ~안 자식 중에서

+ : 다음형제 하나만

~ : 다음형제 여러개

**[] : 속성+속성값**

**순번선택자**

nth-of-type(숫자) : 순서지정

last-of-type : 마지막 순서 지정

first-of-type : 첫번째 순서 지정

**display :**

block 부모태그만큼의 너비(박스)

inline 라인(줄바꿈불가), 스타일적용 불가(너비, 높이 등)

\*inline 여부를 확인하기 위해서는 너비를 조정해보고, 조정이 안된다면 inline이다.

inline-block; 나란히 배치 가능, 사이즈 지정 가능(width, height)

예)img, video, input, select, textarea

**box-sizing**

content-box (기본 - 테두리안쪽만) / border-box (테두리바깥쪽까지)

**margin/padding**

**바깥 여백 margin** : 30px; (left, right, top, bottom 지정가능)

★★★★★**태그의 맨 위와 맨 아래는** 마진을 쓰지 않는다. 밀어내야하기 때문에 **패딩**을 쓴다. (**부모태그**한테)

그 이유는 자식태그는 content-box안에 놓여있고 content-box 안에서 바깥쪽으로 부모태그의 padding 그리고 border를 밀어내는 것이 자식태그의 margin이기 때문이다.

만약에 부모태그에 padding과 border가 없다면 자식태그가 밀어내는 여백이 부모태그 바깥으로 흘러버리게 되어 부모태그의 높이가 정상적으로 유지되지 않는다.

margin : 0 auto; ★★★★★ 부모태그의 공간 안에서 태그 자체를 중앙에 배치할 때 사용하는 표현

**안쪽 여백 padding** : 30px; (left, right, top, bottom 지정가능)

상하, 좌우;

상, 우하, 좌

상, 우, 하, 좌(시계방향)

**overflow :**

hidden(안보이게) / scroll(스크롤바) / auto(자동스크롤바) / visible(기본값, 그대로 보여주기)

**background**

**배경색상** background-color : transparent(기본)

**배경이미지** background-image : none(기본) / url("경로")

\*background-image : linear-gradient(45deg,red,blue); 선형그라데이션

\*background-image : radial-gradient(red,blue); 원형그라데이션

\*background : linear-gradient(0deg, black, transparent)

**배경반복** background-repeat : repeat(기본) / no-repeat(노반복)

\*background-repeat-x : repeat; 가로사이즈만 반복

\*background-repeat-y : repeat; 세로사이즈만 반복

**좌표제어** background-position : x y(기본) 좌표제어

\*background-position-x : 100px

\*background-position-y : 100px

\*(left/right/top/bottom/center)

**크기제어** background-size:(x) (y);

\*px, %, auto

- background-size : contain;=> 최대한 가득(비율유지) / cover (빈틈없이가득(잘려나감)

배경속성 한줄로 정리 가능 : 색상, 이미지, 반복, x축, y축

=> background: pink url(../img/sudal.jpg) no-repeat 100px 100px;

**레이어효과** background-blend-mode : hue / exclusion; 등등

**배경고정** background-attachment : scroll(기본) / fixed(고정)

**calc**

calc(100% - 100px) calc(100% - 50px);

-, + 띄어쓰기 필수 / 나누기 곱하기는 띄어쓰기 안해도 됨

**- (콤마)를 사용해서 배경이미지 2개이상 넣어줄 수 있다.**

background-image: url(../img/sudal.jpg), url(../img/sudal.jpg)

background-repeat: no-repeat, no-repeat;

background-position: 0 0, 100% 100%;

=> background: url(../img/sudal.jpg) no-repeat 0 0, url(../img/sudal.jpg) no-repeat 100% 100%

**CSS작성 순서**

1. RESET

\*{padding:0; margin:0}

h1,h2,h3,h4,h5,h6{font-size:inherit; font-weight:normal;}

a{text-decoration:none; color:inherit;}

strong,b{font-weight:normal;}

i,em,address{font-style:normal;}

2. COMMON STYLE(상속스타일)

html{}

@font-face{}

3. LAYOUT (크기, 배치)

4. DETAIL

**CSS 가상클래스(Virtual-Class)**

:link{} 방문한적 없는 링크

:visited{} 방문한적 있는 링크

:active{} 마우스클릭상태

:hover{} 마우스커서올린상태

:before{content:""} 선택한 태그의 가상의 첫번째 자식태그 만듦

:after{content:""} 선택한 태그의 가상의 마지막 자식태그를 만듦

==============

**float** (블록태그들을 나란히배치)

float : left; right;

clear : both; left; right; float를 해제할 때 맨 마지막 자식태그에 사용

.클래스명:after {

content: "";

display: block;

clear: both;

}

=====>>> 거의 공식과 같음

float를 사용하여 공중에 띄우게 되면 암묵적으로 display 형태는 inline-block으로 변한다.

=================

■flex container속성■

**반드시 부모-자식 관계에서 사용**

**사용하면 포함하는 컨텐츠의 너비만큼으로 바뀐다. (그래서 text-align 못 쓴다)**

**display : flex** 선택한 태그를 flex-container로 지정한다.

**flex-direction :** 메인축의 방향변경

row 기본값 / row-reverse 우-좌 / column 위-아래 / column-reverse 아래-위

**justify-content:** ★메인축★을 따라서 아이템을 정렬

flex-start 기본값 - 메인축의 시작지점

flex-end - 메인축의 마지막지점

center - 메인축의 중앙지점

space-between - 아이템을 양끝에 딱 붙이고 사이여백을 자동으로 똑같이 넣어줌

space-arround - 아이템을 양끝에 붙이지 않고 아이템마다 좌우 바깥여백을 똑같이 설정해서 정렬하는 방법

**align-itmes:** ★수직축★을 따라서 아이템을 정렬하는 방법

stretch 기본값(높이가 설정된만큼 늘어남) flex-start flex-end center

**align-content:** ★수직축★에 기준하여 ★전체 flex-item contents들을 정렬★하는 방식(flex-wrap:wrap; 일 때 유효)

**flex-wrap:** 아이템들의 줄바꿈설정, nowrap 기본값 - 허용x / wrap 허용

=================

■flex item 속성■ (처음부턴 안써도 무방함. width로 대체 가능)

**Order :** 아이템들의 배치순서정함 - order값이 작은것부터 먼저 배치됨.

**align-self :** 선택한 flex-item을 수직축 기준으로 정렬할 수 있다.

flex-start; flex-end; center; stretch;

**flex-basis :** 가로사이즈의 기준을 정할 때(기본점유크기)(가능한~)

auto; px; %; 숫자;

**flex-shrink :** flex 컨테이너의 너비가 줄어들어서 아이템들이 설정된 flex-basis의 값보다 더 작아져야할 때, 줄어드는 비율(nowrap상태일 때 유효함)

**flex-grow :** 플렉스 컨테이너의 공간이 여유가 있어서 남았을 때, 남은 공간을 아이템들이 나눠갖는 비율

**flex: 1;** (이 한 문장이 아래 3줄을 축약한 것)

/\* flex-basis: 0;

flex-grow: 1;

flex-shrink: 1; \*/

}

**list-style :** none / disc / circle / square / upper-alpha / lower-alpha / upper-roman / lower-roman

- li 태그만 적용가능

**border-radius :** 20px 40px 60px 80px; (시계방향순으로 - 위 오른쪽 아래 왼쪽)

\*%단위도 가능, 20px; - 사방을 20픽셀곡률로 지정

**shadow**

**box-shadow** : 0px 0px 0px 색상;

**text-shadow :** 0px 0px 0px 색상;

\* x좌표 y좌표 확산정도 색상

\* 콤마를 사용해서 여러개의 그림자를 넣을 수 있다.

**transition :** 전환■

(변경하기 전 스타일(부모)에 미리 설정)

(1)넓이 transition-property : width; 넓이

\*height; background; border-radius; all; 등

(2)지속시간 transition-duration : 2s; 지속시간

(3)가속도스타일 transition-timing-function : ease; 가속도스타일

\*linear 일정한 속도

(4)지연시간 transition-delay : 0s; 지연시간

\*위 4개를 한 줄로 합치면 => **width 2s ease 0s;**

**object-fit :** cover(이미지잘려도신경안씀) / contain(이미지비율유지)

- video/img 크기조절할 때 사용

**object-position :** 0% 0% cover한 후 이미지 좌표조정할 때 사용

**position : 이 기능을 사용하면 인라인블록태그가 된다.**

relative; 기준영역을 잡아주는 용도 / absolute; 가장 가까운 상위포지션 속성 기준으로 배치

fixed; 화면고정

\*\*\*반드시 top, bottom, right, left 값을 함께 써주어야 함

: z-index : 1; 숫자값으로 포지션으로 배치된 태그의 위아래 정돈 (기본값 : 0)

\*\*\* 형제끼리 비교하는게 가장 좋다.

**transform :**

변형 – 애니메이션 만들 때 사용

\*\*\* 박스형태에만 적용 가능하다.

\*\*\* 값은 px 또는 %로 쓸 수 있다.

\*\*\* ‘%’ 는 선택된 태그의 너비나 높이를 기준했을 때 비율이다.

translateX( ) x축이동 / translateY( ) y축이동 - 한 줄로 Transform:translate(0, 0)

translateZ( ) 앞뒤로 이동

**perspective: px 수치입력 (입체감을 만들어주는 속성 – rotate 효과 주기 위해 필수 – 부모태그에 적용)**

perspective-origin:

transform-origin: 0% center; 변형을 할 때 기준좌표를 중앙에서 맨 위쪽 중앙을 기준으로 바꾸어줄 때

scaleX( ) scaleY(2) scale(1.5)

rotateX(360deg) rotateY(180deg) rotate(180deg);

skewX(30deg); skewY(30deg);

**slider**

기차역 (관통하는 ul영역) 영역

전체구역은 sliderwrap이라고 많이 함

Li는 slide라고 많이 함

<html>코드공식

<div class=”station”> => width:1270px 기준이면

<ul class=train> => width:300%; (슬라이드가 3개일 경우), height:100%;

<li></li> => width:33.3%, height:100%;

</ul>

</div>

<css>코드공식

기차역에 overflow:hidden;

슬라이드를 2개 3개씩 한번에 보여주고 싶을 때는, 기차의 너비를 보이길 원하는 슬라이드의 개수로 나누면 됨.

관통하는ul에 transform:translateX(0px);

--------------------------

**cursor :** pointer ; 마우스를 올렸을 때 커서모양으로 바꾸어 줌

**pointer-events :** auto; 기본값. 마우스에 반응한다. / none; 마우스에 반응하지 않는다.

**■반응형 css■ - media-query**

**PC 디자인** @media (min-width:1024px) { 적용될 스타일들 }

**탭 디자인** @media (min-width:768px) and (max-width:1023px) { 적용될 스타일들 }

**모바일 디자인** @media (max-width:767px) { 적용될 스타일들 }

**vh, vw**

vh = viewport height / 100vh : 전체화면 기준

vw = viewport width / 100vw : 전체화면 기준

현재 실행중인 스크린 크기에 맞춰 상대적 크기를 반환하겠다

계산식 : 원하는 픽셀크기 / 전체 넓이 x 100 = vw. vh 계산됨.

예를들어,

현재 스크린 크기가 height=1000px, width=800px 이라면

1vh = 10px / 1vw = 8px이 될 것이고,

Height:50vh;

Width:25vw;

로 설정한다면

Height – 500px, width – 200px 이런식으로 적용이 된다.

**Img{max-width : 100%}**

**animation (인라인태그는 적용불가)**

**(1)애니메이션을 등록 - @keyframes 등록할애니메이션명{ 0%{}, 100%{} }**

@keyframes naraRotate {

  0% {transform: rotate(0deg);}

  25%{border-radius: 50%;}

  50% {background: grey;}

  100% {transform: rotate(360deg);}

}

**(2)태그에 애니메이션을 적용**

**animation-name :** etc

**animation-duration :** 5s

**animation-timing-function :** linear

**animation-delay :** 0s

**animation-iteration-count :** infinite (기본값은 1(안써도 됨))

**animation-direction :** normal 또는 1(기본값,안써도 됨) / alternate(흐름연속) / reverse / **forwards(스타일유지)**

🡺 한 줄로 표현 가능 **animation : etc 5s linear 0s infinite alternate;**

**animation-play-state :** paused; 마우스를 올렸을 때 애니메이션이 일시정지 됨

**이미지의 크기를 부모 크기만큼으로 제한하기**

**max-width:100%;**