■css창 폰트관련 코드 정리■

========

font-style:italic; 글씨스타일

\*normal(기본) / italic(이탤릭)

font-weight:bold; 글씨굵기

\*normal(기본) / lighter(얇게) / bold(두껍게) / 100~900 수치가능

400이 기본값

font-size:70px; / 3.5rem; / 글씨크기

\*inherit; => 상속받게하겠다.

line-height: 2; 글씨행간

\*기본 120%(=1.2) / % 또는 소수점 또는 px입력가능

font-family: 'Noto Sans Kr', sans serif; 글씨체변경

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

========

위 다섯개를 한 줄로 표현가능(순서중요!!)

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

color:blue;

background:darkblue;

opacity:0.2;

border:1px solid red;

text-decoration:underline; 문단꾸밈

\*none(기본) / underline(밑줄) / linethrough(취소)

text-align:center; 문단정렬

\*left(기본) / right / center / justigy(양쪽정렬)

text-indent:50px; 첫줄들여쓰기

\*px단위

text-transform:capitalize; 영소문자/대문자

\*uppercase(대문자로) / lowercase(소문자로) / capitalize(앞글자만대문자)

letter-spacing; 자간설정

\*rem

word-spacing:20px; 단어와 단어사이의 간격

\*px

word-break:keep-all; 단어쪼개기

\*keep-all(쪼개지않기) / break-all(쪼개기)

white-space:nowrap; 강제로 한줄나열(줄바꿈불가)

\*normal(기본) / nowrap(한줄)

rem; => html에 적용된 글씨크기

em; => 부모태그에 적용된 글씨크기

■폰트등록방법■

<<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■ - 순서를 이용

nth-of-type(숫자) : 순서지정

last-of-type : 마지막 순서 지정

first-of-type : 첫번째 순서 지정

■display태그의 형태■

display:block; 부모태그만큼의 너비(박스)

display:inline; 라인(줄바꿈불가)

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

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

예)img, video, input, select, textarea

■박스의 크기■

box-sizing:content-box; 테두리안쪽만(기본)

box-sizing:border-box; 테두리바깥쪽까지

■박스의 모델■

border: 테두리

\*border:1px solid red;

\*border-left:... (right, top, bottom 지정가능)

margin: 바깥여백

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

★★★★★태그의 맨 위와 맨 아래는 마진을 쓰지 않는다. 밀어내야하기 때문에 패딩을 쓴다.

 첫번째 자식태그한테 margin-top을 만들어서 여백을 주는 것 보다 부모태그한테 padding-top을 지정해주는것이 좋다. 그 이유는 자식태그는 content-box안에 놓여있고 content-box안에서 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

padding: 안쪽여백

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

\*\*\*\*\*

상하,좌우 값이 같을 경우

TOP/BOTTOM RIGHT/LEFT;

50px 100px;

좌우값만 같을경우

TOP RIGHT/BOTTOM LEFT;

50px 200px 100px;

상하좌우 다를경우

TOP RIGHT BOTTOM LEFT;

50px 100px 150px 200px;

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

■overflow속성■ (흘러넘친거 처리할때)

overflow:hidden; 안보이게

overflow:scroll; 스크롤바

overflow:auto; 자동스크롤바

overflow:visible; 기본값, 그대로 보여주기

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

■background속성■

background-color:transparent;(기본) 색상

background-image:none;(기본) 이미지

\*background-image:url("경로")

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

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

background-repeat:repeat;(기본) 반복속성제어

\*background-repeat:no-repeat; 반복하지않는다.

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

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

background-position:x y;(기본) 좌표제어

\*background-position-x:100px;

\*background-position-y:100px;

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

background-size:(x) (y);

\*px, %, auto

- background-size:contain; => 최대한 가득 메꾸는(비율유지)

- background-size:cover; => 빈틈없이 가득 메꾸는(잘려나감)

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

-, + 띄어쓰기 필수 / 나누기 곱하기는 띄어쓰기 안해도 됨

색상 background-color: pink;

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

반복 background-repeat: no-repeat;

x축 background-position-x:100px;

y축 background-position-y:100px;

통합 background-position: 100px 100px;

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

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%

- (콤마)를 사용해서 배경이미지 2개이상 넣어줄 수 있다.

background-blend-mode:hue; 레이어효과

\*exclusion; 등등

background-attachment:fixed; 배경고정

\*기본값은 scroll;

■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

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

/\* 1.RESET \*/

\*{padding:0; margin:0;}

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

b, strong{font-weight:normal;}

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

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

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

■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 속성■ (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

♣- 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 속성■ - li 태그만 적용가능

list-style:none;

\*disc; circle; square; upper-alpha; lower-alpha; upper-roman; lower-roman;

■border-radius■

border-radius:20px 40px 60px 80px; (시계방향순으로)

\*%단위도 가능, 20px; - 사방을 20픽셀곡률로 지정

■그림자■

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)

transition-property:width; 넓이

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

transition-duration:2s; 지속시간

transition-timing-function:ease; 가속도스타일

\*linear 일정한 속도

transition-delay:0s; 지연시간

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

■object-fit■ - video/img 크기조절할 때 사용

object-fit:cover; 이미지잘려도신경안씀

object-fit:contain; 이미지비율유지

object-position: 0% 0% 커버쓸 때 이미지 좌표조정할 때 사용

■position■

position:relative; 기준영역을 잡아주는 용도

position:absolute; 가장 가까운 상위포지션 속성 기준으로 배치

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

Position:fixed; 화면고정

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.

Transform:translateX( ); x축이동

Transform:translateY( ); y축이동

* 한 줄로 Transform:translate(0, 0);

Transform:translateZ( ); 앞뒤로 이동

Perspective:

Perspective-origin:

Transform-origin:

Transform:scaleX( );

Transform:scaleY(2);

Transform:scale(1.5);

Transform:rotateX(360deg);

Transform:rotateY(180deg);

Transform:rotate(180deg);

Transform:skewX(30deg);

Transform:skewY(30deg);

■slider■

기차역 (관통하는 ul영역) 영역

<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);