Css 기본 문법 : 선택자 { 속성 : 값 ; }

선택자 : css를 적용할 대상

속성 : css의 종류 (color.. font-size..)

값 : css의 값 (blue , 50px..)

Margin 속성 : 요소 외부 여백

Css 선언 방식

내장 방식 : <style></style> 태그안에 내용으로 스타일을 작성하는 방식 (선호 x)

인라인 방식 : 요소에 직접 style속성을 주고 작성하는 방식 (선택자 없음)

별도로 선택자를 찾는 행위를 할 필요가 없음 (유지보수에 악영향을 미침) 선호 x

링크 방식 : <link/>로 외부 css 파일을 가져와서 연결하는 방식

Import 방식 @import 규칙으로 Css 파일 안에서 또 다른 css파일을 가져와 연결하는 방식

링크 방식을 제일 선호 !! ----------------------

CSS 선택자

-기본 선택자 , 복합 선택자 , 가상 클래스 , 가상 요소 , 속성

1. 기본 선택자
   1. 전체 선택자 \* : 모든 요소를 선택
   2. 태그 선택자 : 태그의 이름으로 요소를 선택 함
   3. 클래스 선택자 : HTML의 전역 class속성(중복 가능) 의 값으로 요소를 선택 한다. <div class = “css” /> >> .css {..};
   4. 아이디 선택자 : 클래스 선택자와 동일 한데 . 대신 #으로 요소를 선택하고 중복 허용 안된다.
2. 복합 선택자 (기본 선택자를 조합해서 사용 )
   1. 일치 선택자 : 선택자 두개를 동시에 만족하는 요소 선택

Ex ) span.fruit {} : 스팬태그 중에 클래스 값이 fruit 요소 선택

* 1. 자식 선택자 : 선택자 ABC의 자식요소 XYZ 선택

Ex ) ul > .orange : ul태그의 자식요소 중에 클래스 값이 orange요소 선택

2.3 하위(후손) 선택자 : 선택자 ABC의 하위 요소 XYZ 선택.띄워쓰기가 하위(후손)선택자의 기호 ex ) div .orange {} : div태그 안에서 클래스가 orange것을 모두 선택

* 1. 인접 형제 선택자 : 선택자 ABC의 다음 형제 요소 XYZ 하나를 선택

Ex ) ,oragne + li {}: 뒤에서 해석 하는게 편함 li 태그 들 중에서 클래스로 orange를 가지고 있는 요소 중 다음 li태그 요소가 선택됨

* 1. 일반 형제 선택자 : 선택자 ABC의 다음 형제 요소 XYZ 모두를 선택 ex ) ,oragne ~li {}

1. 선택자\_가상 클래스
   1. :hover (.클래스명:hover..)

Ex : ,box { width : 100px; , heigth : 100px; } /

.box:hover { width : 300px;} : 선택자에 마우스를 올리면 너비가 300Px로 변한다.

>> 선택된 요소에 마우스 커서가 올라가 있는 동안 선택됨

* 1. 선택자:active(마우스를 클릭 하는 동안 선택됨)
  2. 선택자:focus(선택자 요소가 포커스되면 선택)

Focus가 될 수 있는 요소는 HTML 대화형 콘텐츠가 해당된다.

대표적으로 input , button ,a , label , select 태그가 있다.

Tabindex= -1 을 추가해서 foucs가 가능 하게 할 수 있다.

* 1. 선택자:first-child : 선택자가 형제 요소 중 첫째 라면 선택

Ex) <div class = “fruits”>

<span>딸기</span>

<span>수박</span>

<div>망고</div>

</div>

.fruits span:first-child {} : 딸기가 선택됨

: 형제 요소 중에서 첫째 이면서 span 태그여야 하고 .fruits 선택자의 후손 이여야 한다.

.fruits div:first-child {} : 아무거도 선택안됨

* 1. 선택자:last-child : 형제 요소 중 막내 요소 선택됨
  2. 선택자:nth-child(n) : 선택자가 형제 요소 중 (n)번째 라면 선택

선택자:nth-child(2n) 이런식으로 가능 n은0부터 증가됨

* 1. 부정 선택자

.fruits \*:not(span) {} : 클래스명으로 fruits를 가지고 있는 요소의 후손 중 모두를 선택한다. (span 태그 선택자만 빼고)

위에 예시로 하면 망고만 선택된다.

1. 가상 요소 선택자
   1. 선택자 ::before >> 선택자 요소의 내부 앞에 내용을 삽입

1. 스타일 상속

부모 요소가 가진 스타일을 자식요소에게 전달 해준다.

상속 될 수 있는 요소 : font-style , font-size, font-weight,color, text-align .. 등

강제 상속 하는법 : 부모요소와 자식요소를 일치 시키려면

Ex ) 부모요소 : height : 200 px; 자식요소 : height : inherit;

라고 해두면 부묘오소에서 height 속성의 값을 바꿔도 자식요소에도 자동 적용 된다.

선택자 우선순위

: 같은 요소가 여러 선언의 대상이 된 경우, 어떤 선언의 css속성을 우선 적용 할지 결정하는 방법

* 점수가 높은 선언이 우선함
* 점수가 같으면 , 가장 마지막에 해석된 선언이 우선함!

!important가 있는 선택자 999999점 (1등)

ID 선택자 100점 (3등)

Class 선택자 10점

태그 선택자 1점

전체(모두)선택자 \* 점

인라인 선언 1000점 (2등)

Body(태그) – body태그는 모든 요소들의 부모이다. 하지만 상속은 우선순위에 점수 계산을 하지 않는다.

가상 클래스 선택자(:hover) : 클래스 선택자와 마찬가지로 10점

가상 요소 선택자 (::before,after): 태그 선택자와 동일 하게 1점

Css 속성 (propertis)

Css작업하기 전에 브라우저가 가지고 있는 고유의 css를 리셋 하자!

Width , height : 요소의 가로/세로 너비

Auto : 브라우저가 너비를 계산 (기본값) / 요소에 이미 들어있는 속성의 값

단위 : px , em ,vw 등 단위로 지정

인라인 요소 대표적으로 SPAN 태그 : 가로너비,세로너비가 최대한 줄어드려고함 (컨텐츠의 크기에 맞추려고한다.)

가로,세로 너비를 정할 수 없다. (글자를 취급하려고 만든 태그 이기 때문임)

블럭 요소 대표적으로 div태그 : 가로너비는 최대한 늘어나려고 한다.

블럭 요소의 세로너비는 요소가 가지고 있는 컨텐츠의 크기에 맞춰서 자동 설정 된다. (내가 특정 단위를 이용하여 값을 조정 할 수 있다.)

Max-width , max-height : 요소가 가질 수 있는 최대 세로/가로 너비를 제한 할 수 있따.( 기본값으로는 설정 하지 않은 상태이다.)

특정 단위로 지정 할 수 있다.

Min-width, min-height 도 마찬가지 이다. (기본값 0 : 0까지 줄어 들 수 있다.)

부모 요소의 가로너비가 300px 이고 자식 요소의 가로너비의 min-width 속성의 값을 400px로 주면 부모 요소의 너비보다 커질 수 있다.

Width , height 속성의 역할은 ? 요소의 가로너비 , 세로너비

Width , height 의 기본값 ? auto (브라우저가 알아서 지정)

Max width , height 의 기본값 ? None

Min width , height 의 기본값 ? 0

Ch 8 – 3강

Css – 단위

Px : 픽셀 ( 하나의 출력 할 수 있는 하나의 점)

% : 상대적 백분율 (ex 부모 – 자식 사이면 부모가 가진 크기에 비례하여)

Em : 요소의 글꼴 크기에 비례 (font-size)

font -size : 10px; 1em = 10;

vw : viewport의 사이즈(가로)에 비례 함

vh : viewport의 사이즈(세로)에 비례

ch 8 – 4강

Margin (외부 여백) : 요소의 외부 여백(공간)을 지정하는 단축 속성

음수 사용 가능 ,

Margin 기본값 : 0(외부 여백 없음)

Auto : 브라우저가 여백을 알아서 계산 ( 가로 (세로)너비가 있는 요소의 가운데 정렬을 위해 활용되기도 한다.)

Margin : 10px; (위,아래,좌,우 모두 10px씩 여백이 생긴다.)

Margin : 10px 20px ( 띄워쓰기해서 두개의 값을 넣으면 앞에값은 위아래 두번째는 좌우)

Margin 10px 20px 30px (위 ,좌우 , 아래)

Margin 10px 20px 30px 40px; (위 우 아래 좌)

Margin은 음수값을 적용 할 수 있다.

= 외부의 여백이 쪼그라든다.. (요소들이 겹쳐진다..)

Padding (요소의 내부 여백을 지정하는 단축 속성)

내부에 여백이 생기기 때문에(요소의 크기가 늘어날 수 있다.)

기본값 : 0(내부여백 없음)

Auto: 없음

% : 부모 요소의 가로너비에 비례해 지정

Margin과 동일하게 값 지정 가능 (1~4개)

-요소의 테두리선 (border)과 색상표현

Border : 요소의 테두리선을 지정하는 단축 속성

Border 속성을 사용하면 요소의 크기가 커진다.

개별 속성은 다음 순으로 입력하는걸 추천

Border-width : 선 두께

Border-style : 선 종류

Border-color : 선 색상

-border-width : 요소 테두리의 선 두께

기본값 : medium (중간 두께) - 비추천

단위 수치로 명시 하는 것을 추천 한다.

Border-width도 padding , margin 속성과 마찬가지로 방향 별로 지정 할 수 있다.

-border-style : 요소 테두리의 선 종류

기본값 : none(없음)

Solid(실선) , dashed(파선) 많이 사용 한다.

B0rderstyle도 각 방향별로 설정 가능

border-color : 요소 테두리선의 색상을 설정

기본값 : black

Transparent : 선의 색상을 투명하게 만든다.

두께는 만들어지는데 눈에는 안보임

마찬가지로 방향별로 색상 지정 가능

Css 에서 색상을 표현하는 방법

-색상 이름을 명시 하는 방법

-Hex 색상코드(#000 , #FFF 등등) – 16진수 색상 (추천)

-RGB 빛의 삼원색 =rgb (25,25,25)

-RGBA 빛의 삼원색 + 투명도 = rgba (0,0,0,0.5)

Box-sizing 계산 하기 (요소의 크기 계산 기준을 지정)

기본값 : content-box;

만약 우리가 지정한 가로,세로 너비가 있는데 거기에 Padding값이나 border를 주면 요소의 크기가 커진다.

하지만 기존에 있던 크기를 유지하면서 padding,border 속성을 추가 하고 싶으면 border-size : border-box; 속성을 이용 해야 한다.

Overflow(넘침흐름제어)

요소의 크기 이상으로 내용이 넘쳤을 때, 보여짐을 제어하는 단축 속성

예를 들어 부모 선택자의 가로너비가 200px 이고 자식 선택자의 가로너비가 300px 이면 실제로 화면상에는 자식 요소의 부모요소 영역에 넘쳐 나갈 것인데 이거를. 부모요소에서 제어 할 수 있다.

기본값 : visible (넘친 내용을 그대로 보여줌)

Hidden : 넘친 내용을 잘라냄

Auto : 넘친 내용이 있는 경우에만 잘라내고 스크롤바 생성

출력 특성(display)

요소의 화면 출력 (어떻게 보여짐) 특성

Display 각 요소에 이미 지정 되어 있는 값

Block (상자 요소 >> div ) , inline(글자 요소 >> span) , inline-block (글자 + 상자 요소) >> base는 글자요소 인데 가로,세로 값을 지정 가능 함 ( 글자요소는 가로세로 지정 불가)

기타..

따로 지정해서 사용하는 값

Flex (플레스 박스) – 1차원 레이아웃 – 수평정렬에 용이 (하나의 축만 사용)

Grid (그리드) – 2차원 레이아웃 – x,y축 둘다 사용

None – 화면에서 사라짐

스팬태그에 display:block 속성을 추가하면 가로,세로 너비를 가질 수 있다.

요소의 투명도(opacity)

기본값 1 (불투명)

0~1 사이의 값으로 지정 가능

글꼴

Font-style :글자의 기울기

기본값 :normal (기울기 없음)

Font-style : italic(이텔릭체)

Font-weight : 글자의 두께

기본값 : 400(normal)

700(bold) 두껍게

그 외에 100~900 까지 지정 가능

Font-size : 글자의 크기

기본 : 16px;

Px,em,rem등 여러 단위로 지정 가능

Line-height (한 줄의 높이)

숫자로 지정 : 요소의 글꼴 크기의 배수로 지정

만약 line-height: 1.4 로 별도의 단위 없이 지정하면 내 글꼴크기의 1.4배의 높이만큼 지정 하겠다 라는 뜻 (추천)

단위를 사용하여 지정 할 수 있다.

Font-family : 글꼴 지정

Font-family : “글꼴 1” , 글꼴 2 , 글꼴 계열;

글꼴 후보를 여러개 쓸 수 있다. 첫번째가 없으면 두번째 글꼴 적용..

마지막에는 항상 글꼴계열을 써줘야한다(필수)

글꼴을 “”로 감싸는 경우는 특수문자나, 띄워쓰기가 들어가있는 글꼴일 때는 “”로 감싸 주어야 한다.

Sans-serif(고딕체 계열)

Monospace(고정너비(가로폭이 동등)글꼴 계열)

문자

Color(글자의 색상)

기본값 : rgb(0,0,0) : 검정색

색상 표현의 방법으로 지정 가능

Text-align(문자의 정렬방식)

기본값 left(왼쪽정렬)

Right , center 로 오른쪽 가운데 정렬 가능

Text-indent : 문자 첫 줄 들여쓰기 (음수로 쓰면 짤림 – 내어쓰기)

배경과 관련된 css

Background-color : 가로,세로 너비를 지정한 경우 시각적으로 확인 하려고 색상을 부여함

Background-image : url(“경로”);

Background-size : 200px; (배경으로 삽입한 이미지의 크기를 지정)

하지만 내가 배경을 불러온 요소의 크기 보다 이미지의 크기를 작게 하면 바둑판식 배열 처럼 이미지가 여러개 삽입 된다.

그것을 방지 하기 위하여 background-repeat:no-repeat;

배경으로 불러온 이미지를 요소의 정ㄱ ㅏ운데에 배치

Background-position : center;

Background-repeat : repeat-x (수평으로만 반복)

Background-repeat : repeat-y (수직으로만 반복)

Background-position : center , top , buttom , left, right , 방향으로 지정 가능하고 단위로 지정 가능 (x축,y축 기준으로)

Background-position : top right; 이런식으로 가능

Background-size : 배경의 이미지 크기 지정

기본값 : auto(실제 이미지 크기)

단위를 사용하여 지정 가능

Cover: 비율을 유지 , 요소의 더 넓읍 너비에 맞춤 (가로 너비 , 세로 너비)

Contain : 비율율 유지 , 요소의 더 짧은 너비에 맞춤 (가로 너비 , 세로너비)

Background-attchment(요소의 배경 이미지 스크롤 특성)

기본값 : scroll (이미지가 요소를 따라서 같이 스크롤)

Fixed : 이미지가 뷰포트에 고정(스크롤x)

요소 배치 (1)

요소를 배치 할때는 기준을 먼저잡고 그 위치를 잡아 줘야한다!!!

Position (요소의 위치 지정 기준)

Position과 같이 사용하는 css 속성ㄴ들

Top bottom left right z-index : 모두 음수값을 사용 할 수있다.

Position 기본값 : static (기준없음)

Relative : 명시한 요소 자신을 기준

Absolute 위치상 부모 요소를 기준 (부모 요소 확인이 중요)

위치상 :

Fixed 뷰포트(브라우저)를 기준

기준이 마련 되면 실제 위치값을 선언 해줘야한다.

요소의 각 방향별 거리 지정 : Top bottom left right

기본값 : auto (브라우저가 계싼)

단위를 사용하여 수치 지정 가능

Position : relative (요소 자신을 기준을 배치!) – 잘안씀

Top : 30 px;

Left : 30px;

이렇게 속성을 주면 원래 있던 자리에서 위 , 왼쪽으로 30px 떨어진곳에 요소가 시각적으로 위치 하게 되고 기존에 있던 자리는 시각적으로만 비어있게 된다.(요소 자체는 그 자리에 그대로 있다.)

위치에 대한 기준만 부여되고 값을 부여 안하면 아무런 변화가 없다!!!!!

Position:absolute; (위치 상 부모 요소를 기준으로 배치!)

자식 요소들의 배치와는 무관하다 .. 붕 ~ 뜨면서 요소가 겹칠 수도 있다.

자식 1 자식 2 자식 3 이 있고 자식 2 에 position:absolute; 속성을 주면 자식 1 다음에 바로 자식 3번이 배치가 되고 자식 2번은 붕 ~ 뜨면서 겹친다.

진짜 부모 요소에 position-relative 속성을 안주면 위치 상 부모 요소가 뷰포트가 되어 버림

만약에 부모요소에 position:relative가 없고(position:static) 조상(상위)요소에 position:relative가 있으면 조상 요소가 위치 상 부모요소가 된다.

부모 , 상위 요소에 position값이 없으면 뷰포트(브라우저)가 위치 상 부모 요소가 된다.

Position:fixed (처음부토 뷰포트(브라우저)를 기준으로 배치)

요소 쌓임순서(statc order)

: 어떤 요소가 사용자와 더 가깝게 있는지(위에 쌓이는지) 결정

1 요소에 position 속성의 값이 있는 경우 위에 쌓임(기본값 Static 제외)

2 1번 조건이 둘다 있을 경우 , z-index 속성의 숫자 값이 높을 수록 위에 쌓임

3 1,2번 둘다 같은 경우 ,HTML의 다음 구조일 수록 위에 쌓임 (더 나중에 작성 되어 있을수록)

z-index (요소의 쌓임 정도를 지정)

기본값 : 0 or 부모 요소와 동일한 쌓임 정도

Position 속성의 값으로 absolute , fixed가 지정된 요소는 , display 속성이 block으로 자동으로 변경 된다!!!

Flex(플렉스) – 1차원의 레이아웃을 만드는 개념

<div> 대표적인 블럭요소 이고 블럭요소는 수직으로 쌓이는 성질을 가지고 있다.

이것을 수평으로 정렬 하려면 정렬 하려고 요소들의 부모에 display:flex; 을 추가 한다!

부모요소에 display:flex가 추가 되면 그 부모요소는 Flexcontainer가 되고 그 부모의 자식요소는 자연스럽게 Flexitems가 된다.

Flexcontainer에 부여하는 css속성과 Flexitems에 부여하는 css속성은 서로 다르다.

- display 속성 : 화면에 어떻게 보여질지 특성

Flex : 블록 요소와 같이 FlexContainer로 만든다.

Inline-flex : 컨테이너로 만드는데 인라인 요소 처럼 동작 한다.

Flexcontainer가 여러개면 일반적인 블럭요소 처럼 수직으로 쌓인다.

FlexContainer에 만든 후 거기에 사용하는 css속성

Flex-direction : 주 축을 설정 ( 수평 정렬 or 수직정렬)

기본값 : 행 축 : Row (좌 >> 우) - 수평

Row-reverse (행축 >> 우 에서 좌)

Column : 열축(위 에서 아래) – 수직

수평 정렬 한다고 가정하고!!

Flex-wrap : flexitems들의 묶음(줄 바꿈) 여부

기본값 nowrap (줄 바꿈 없음)

Wrap : 여러 줄로 묶음

Justify-content : 주 축의 정렬 방법 정의

기본값 : flex-start (Flex Items를 시작점으로 정렬)

Flex-end (Flex Items를 끝점으로 정렬)

Center (Flex items를 가운데 정렬)

Align-content 교차 축의 여러 줄 정렬 방법 (수평정렬이면 수직정렬에 대한 속성) – 기본적으로 줄바꿈 상태여야 한다(flexwrap 이 있어야함)

기본값 : stretch(Flex items 를 시작점으로 정렬)

flex-start : Flex Items를 시작점으로 정렬

Flex-end (Flex Items를 끝점으로 정렬)

Center (Flex items를 가운데 정렬)

Align-items(교차축의 한 줄 정렬 방법) – 활용도 높음

수평정렬이면 수직정렬에 대한 속성

기본값 : stretch(Flex items를 교차 축으로 늘림)

flex-start : Flex Items를 각 줄의 시작점으로 정렬

Flex-end (Flex Items를 각 줄의 끝점으로 정렬)

Center (Flex items를 각 줄의 가운데 정렬)

Flexitems에 부여하는 css 속성

Order : flexitems들이 정렬되는 순서

기본값 : 0 (순서없음)

숫자: 숫자가 작을수록 먼저

Flex-grow : flexitems의 증가 너비 비율

기본값 : 0 (증가비율 없음)

숫자 : 증가 비율

Flex-shrink (flexitems의 감소 너비 비율)

기본값 : 1 (플렉스 컨티에너의 너비에 따라 감소 비율 적용)

숫자 : 감소비율 ( 0을 입력하면 감소비율이 없다는 뜻..)

컨테이너의 너비에 맞춰서 아이템들이 줄어드는것을 방지 하기 위해 0을 입력하는 경우가 있다. ( 아이템들이 실제 너비 만큼 크기를 유지 하게 한다,)

Flex-basis ( flex-items의 공간 배분 전 기본 너비)

기본값 : auto (요소의 content 너비)

단위로 지정 (컨텐츠 영역이 가지는 너비)

Flexbasis : 0 으로 하면 각 아이템들에 flew-grow로 지정한 만큼 너비를 가진다.

전환 효과

Transition : 요소의 전환(시작과 끝)효과를 지정하는 단축 속성

Transition : 속성명 지속시간(필수 작성) 타이밍함수 대기시간;

<속성명>

transiton-property : 전환효과를 사용할 속성 이름을 지정

기본값 : all / 전환 효과를 사용할 속성명 명시

<지속시간>

transiton-duration : 전환효과를 지속하는 시간을 지정

기본값 : 0s / 시간 지정

여러개의 개별속성을 지정 가능

Transition :

width 1s ,

background-color 2s;

너비는 1초 , 배경 색상은 2초에 걸쳐 전환

Transiton-timing-function : 전환효과의 타이밍(easing) 함수 지정

기본값 (ease): 느리게 – 빠르게 – 느리게

Linear : 일정하게

Ease-in : 느리게 – 빠르게

Ease-out : 빠르게 – 느리게

Ease-in-out : 느리게 – 빠르게 – 느리게

transiton-delay : 전환 효과가 몇 초 뒤에 시작할지 대기시간 지정

기본값 : 0s

시간 s 초를 따로 지정 가능

변환 효과

Transform : 변환함수1 , 변환함수2 , 변환함수3...

Transform : 원근법 이동 크기 회전 기울임

변환함수(2D 변환 함수)

Px단위 사용

Translate(x,y) : 이동(x축 , y축) , translateX(x) : 이동(x축)

TranslateY(y) : 이동(y축)

Scale(x,y) : 크기(x축,y축)

Rotate(degree) : 회전(각도) : deg단위 사용

SkewX(x) : 기울임(x축) : deg단위 사용

skewY(y) : 기울임(y축) : deg단위 사용

3D 변환 함수

rotateX(x):회전(x축) : deg단위 사용

rotateY(y):회전(y축) : deg단위 사용

perspective(n) : 원근법(거리) : px단위 사용

원근법 함수는 제일먼저 작성해야 한다!

Perspective 속성 vs perspective 함수

속성 : perspective(px)

함수 : transform : perspective(px)

관찰대상

속성 : 관찰대상의 부모

함수 : 관찰 대상

Backface-visibility ( 3D변환으로 회전한 요소의 뒷면 숨김 여부)

기본값 : visible(뒷면 보임)

Hidden(뒷면 숨김)