**<a> 태그의 속성값 ‘target’**

\_self : default값. 현재 창에서 열기

\_blank : 새 창으로 열기

\_parent : 가장 근접한 부모의 브라우저에서 열기

( 창 A에서 창 B를 열었고, 창 B에서 \_parent로 창을 열면 창 A에서 열리게 됨 )

\_top : 여러 중첩된 브라우저가 존재할 때 최상위 브라우저에서 열기

**‘em’과 ‘rem’**

# ‘em’을 사용해 폰트 크기 바꾸는 법

em은 기준이 되는 폰트 크기가 있어야 하고, ‘직계 부모’를 기준으로 작성됨

폰트 크기에 통일성을 부여하므로 현업에선 더러 사용함

Ex) section { font-size: 16px; }

* 1em을 특별한 경우가 아닐 시 16px로 지정함

이후 하위 태그에 1em, 3em 혹은 0.8em 이런식으로 작성 가능

# ‘rem’을 사용해 폰트 크기 바꾸는 법

‘rem’의 ‘r’은 ‘root’라는 의미로 뿌리, 근본이라는 뜻

따라서 ‘rem’을 사용할 때는 html의 폰트 크기를 꼭 부여해줘야 함

Ex) html { font-size: 16px; }

* 해당 문서의 1rem을 16px로 지정함

이후 하위 태그에 1em, 3em 혹은 0.8em 이런식으로 작성 가능

**폰트 관련 css 속성**

\* letter-spacing: 자간 설정. 음수도 가능함 ( px, em, rem 등으로 표기 )

\* border: 1px solid #000;

= border-width: 1px; / border-style: solid; / border-color: #000

풀어서 쓰는 것이 유지보수 및 자바스크립트 접근에 좋음

\* text-align: 블록 속성에만 적용이 가능함, span과 같은 인라인 속성에는 적용되지 않음

**background-image 관련 css 속성**

\* background-image: url( ); 로 이미지 삽입 가능

\* background-position: left, center, top / px / % 으로 표기 가능

\* background-size: cover / contain

# 불러오는 역할을 하는 속성

scr: html에서 사용

href: 통로만 연결해주는 역할

url: css에서 사용

**flex 관련 css 속성**

\* display: inline-flex; 자기 자신은 인라인 속성이 되고 자식요소를 flex 환경으로 설정

\* flex-direction의 값

row: default값, 기본축 가로, 진행방향 좌 > 우

row-reverse: 기본축 가로, 진행방향 우 > 좌

column: 기본축 세로, 진행방향 위 > 아래

column-reverse: 기본축 세로, 진행방향 아래 > 위

\* flex-wrap: 줄바꿈 여부에 관여

\* justify-content의 값

flex-start: default값, 자식요소를 주축의 시작지점에서부터 정렬함

flex-end: 자식요소를 주축의 종료지점에 배치함.

세부 컨텐츠의 순서 및 정렬에는 관여하지 않고, 한 덩어리의 컨텐츠를 어디에 놓을지만 결정함

\* flex: 1;

= flex-grow: 1; / flex-shrink: 1; / flex-basis: 0%;

flex-basis의 속성이 너비, 높이값을 무시하는 것이므로 flex에 숫자값을 부여하면 width값을 무시하고 전체 페이지의

가로 너비에서 설정값만큼 나눠가짐

**css 선택자 종류와 예시**

전체 선택자: ‘ **\*** ’

아이디 선택자: ‘ **#** ‘

클래스 선택자: ‘ **.** ’ ( 태그의 종류나 위치가 달라도 같은 클래스가 부여됐을 시 동시에 호출 )

이중 조건 선택자: ‘ .txt.text ‘ ( 두가지 클래스를 동시에 만족하는 요소를 호출 )

자손 선택자: ‘ .div .child\_div ‘

태그 선택자(요소 선택자): ‘ div ’ ( 해당 페이지 안에 있는 모든 div요소를 호출, 사용에 유의해야함 )

직계 자식 선택자: ‘ **>** ‘ ‘ .txt > p ‘ ( 단순한 태그 선택자로는 한계가 있을 시 직계 자식 선택자 ‘ > ’ 사용으로 직계 자식 호출이 가능 )

복수 선택자: ‘ **,** ‘ ‘ .p, .s ‘ ( and의 의미 )

상태 선택자: ‘ **:** ‘ ‘ input:checked ‘ ( input 태그가 checked 상태일 때를 의미함 )

인접 형제 선택자: ‘ **+** ‘ ( 어떤 요소가 특정 상황이나 조건일 때 바로 뒤에 있는 형제 요소를 호출하는 것으로 바로 다음에 오는 요소만 적용 가능함. 중간에 다른 요소가 끼어있으면 안됨 )

일반 형제 선택자: ‘ **~** ‘ ‘ h3 ~ ul ‘ ( 상속받은 요소에 적용, 예시에서 h3에는 적용되지 않음 )

가상 선택자: ‘ :before ’,’ ‘ :after ‘

속성 선택자: ‘ **[ ]** ‘ ‘ input[type=’email’] ‘ ( ‘ a[target] ‘ < a태그 중 target 속성이 존재하는 요소만을 선택 )

\*속성 선택자 종류

h2[class^=’main’] : main으로 시작하는 클래스를 가진 요소를 호출

h2[class$=’\_txt2’] : \_txt2로 끝나는 클래스를 가진 요소를 호출

h2[class~=’com’] : com이라는 클래스 명을 앞뒤 여백(띄어쓰기)을 가지고 완벽히 포함하는 요소를 호출

h2[class\*=’txt’] : txt라는 클래스 명을 포함하는 모든 요소를 호출 ( 띄어쓰기 상관없이 )

**css 선택자의 우선순위**

기본적인 규칙 : 나중에 쓴 내용이 이전의 내용을 덮어씀.

1. 속성의 값 뒤에 !important 를 붙이면 무조건 최우선이 되지만, 절대 사용해서는 안됨 ( 나머지의 우선순위가 다 꼬이게 되어 적용이 어려움. 따라서 정말 최종적으로 사용할 수 밖에 없을 때만 사용 )
2. html에 style 속성으로 css를 적용하면 (최)우선이 됨, 그러나 html 문서의 길이가 너무 길어지고, html과 css에 스타일을 나눠서 적용하면 협업과 유지보수가 불편해지므로 되도록 지양함
3. id로 지정한 속성
4. class로 지정한 속성
5. 태그 이름으로 지정한 속성
6. 상위 객체로부터 상속된 속성이 적용

=> !important > html style > id > class > 태그 > 상속된 속성

**마진 역상속**

부모 블록요소에서 첫번째 자식 블록요소에 마진 탑이 설정될 경우 그 값이 부모 요소로 역상속되는 현상을 말함

따라서 패딩값을 주는 것이 좋음

**img 태그의 css 속성**

\*object-fit의 값

fill: default값, 이미지의 비율을 무시하고 부여된 너비값으로 채움

contain: 이미지의 비율을 유지하면서 ‘가로 너비’를 채워서 삽입함.

따라서 상하 여백이 생길 수 있음, 이미지의 좌우에 중요한 컨텐츠가 있고 위아래는 중요하지 않거나 의미없는 여백일 때 이 방법이 더 좋을 수 있음

cover: 이미지의 비율을 유지하면서 ‘세로 너비’를 채워서 삽입함.

따라서 좌우 여백이 생길 수 있음

**transform 관련 css 속성**

※ transform은 한 번에 여러 효과를 적용시킬 수 있으나, 중첩 시 작성한 순서에 따라서 결과가 달라짐.

그러므로 원하는 효과를 얻기 위해서는 적용시키는 순서에 주의해야함

\*translate: 이동 효과, 방향에 양수 혹은 음수가 적용됨

\*scale: scale은 축소 또는 확대하는 것으로 1보다 큰 값이면 확대, 1보다 작은 값이면 축소되며 값은 하나만 적음

scale은 모든 방향으로 적용, scaleX는 x축으로만 적용, scaleY는 y축으로만 적용

\*rotate: 회전 효과이며 단위는 deg로 사용

\*skew: 왜곡 효과, 해당 도형이나 개체를 비틀어서 기울이는 형태로 만듦. skew값을 90deg를 줄 경우 개체가 아예 사라진 것처럼 보임 ( skew 자체를 잘 안쓰지만 특정 상황에서 더러 사용함 )

\*perspective: 원근 효과의 값을 조절하는 것으로 700~1600px 정도의 값을 추천함

\*transform-origin: 기준점을 잡는 것으로 left center 등의 방향으로 설정 가능

**transition 관련 css 속성**

※ transition은 모션이나 애니메이션 속도를 조절하는 것으로, 부여한 속성이 ‘즉시’ 결과값으로 나타나는 것이 아닌, ‘일정 시간에 걸쳐 자연스럽게’ 나타나도록 하는 것을 뜻함

\*transition-duration: 지속시간을 의미하며 단위는 s를 사용( ms도 가능하지만 사용하지 않음 )

\*transition-property: 적용시킬 대상을 의미하며 보통 all이라고 적고, 필요에 따라 특정 대상을 적음

\*transition-timing-function: 가속도를 의미함

linear: 처음부터 끝까지 일정한 가속도 ( 자주 사용 )

ease: 천천히 시작되어서 빨라지다가 마지막에 다시 느려짐

ease-in: 천천히 시작해서 노멀하게 마침

ease-out: 천천히 마침

ease-in-out: 천천히 시작해서 천천히 마치는 것

\*transition-delay: 지연시간을 의미함