#### **CSS** ―――

둘 다 <head> **요기 CSS 요기** </head> 사이에 넣음!

css값은 끝난 뒤에 **;** 가 필수! 빼먹을 경우 그 아래 값이 전부 인식XXX

시작할 때 넣을 것: @charset "utf-8";

폰트설정:

@import url('<https://fonts.googleapis.com/css2?family=Nanum+Gothic&display=swap>');

| 내부 CSS | <!DOCTYPE html>  <html lang="ko">  <head>  <meta charset="UTF-8">  <title>Document</title>  **<style>**  /\*여기가 이제 css존\*/  **</style>**  </head>  <body>    </body>  </html> |
| --- | --- |
| 외부 CSS | <!DOCTYPE html>  <html lang="ko">  <head>  <meta charset="UTF-8">  <title>Document</title>  **<link rel="stylesheet"**  **type="text/css"**  **href="style3.css**(외부 css 제목)**">**  </head>  <body>    </body>  </html> |

#### 

#### 

#### **웹 브라우저별** 호환

- css 웹 브라우저별 접두사

CSS3는 표준안이 아직 확정되지 않은 상태이기 때문에 최신 웹 브라우저들은 CSS3 속성을 실험적으로 제공하고 있음. 이를 위해 속성이나 속성 값 앞에 웹 브라우저별로 접두사(vendor prefix)를 제공하고 있으며 이 접두사의 경우 웹 브라우저별로 다르기 때문에 하나의 속성을 선언하기 위해서는 여러 번의 동일한 선언을 지정해야 함.

| 파이어폭스 -moz- | 크롭, 사파리 -webkit- | 오페라 -o- | 익스플로러 -ms- |
| --- | --- | --- | --- |

#### **색상** 적용

[RGBA 형식 및 HSLA 형식의 사용 예]

참고 링크: http://www.colorpicker.com/

RGBA형식 --> color:rgba(255, 127, 45, 0.5)

color:rgba(red, greeen, blue, 투명도)

참고 링크: http://www.workwithcolor.com/hsl-color-picker-01.htm

HSLA형식 --> color:hsla(0, 0%, 100%, 0.5)

color:hsla(색상, 채도, 명도, 투명도)

[그라디언트]

참고 링크: http://www.colorzilla.com/gradient-editor/

background:-webkit-linear-gradient(각도deg, blue 20%, red 70%)}

#### **선택자** 종류1

보통은style 내에서 아래에 쓰인 값이 강하게 적용됨.

| **style** | **body** | **비고** |
| --- | --- | --- |
| 선택자A>선택자B; | 자식선택자  tr>th | 선택자A의 자식에 위치하는 선택자 B 선택  tr의 첫번째 th에 해당 |
| 인접 형제 선택자 | +  h1 + p {color: blue;} | h1 요소 다음에 오는 p 요소들 중  첫번째 p요소에만 파란색 글씨 적용 |
| 형제 선택자 | ~  h1 ~ p {color: blue;} | h1 요소 다음에 오는 모든 형제 p요소에 밑줄 적용 |
| [속성] | [required] | 해당 속성이 있는 요소  input[type="search"]{  border: 2px solid #329fff;  } |
| [속성 = 값] | [target = \_blank] | 지정한 속성 값이 있는 요소 |
| [속성 ~= 값] | [class ~= button] | 해당 값이 포함되어 있는 요소를 찾아 적용 |
| [속성 |= 값] | [title |= us] | 특정 값이 포함된 속성을 가진 요소를 찾아 적용  하이픈으로 연결해 한 단어 값을 이루는 요소 |
| [속성 ^= 값] | [title ^= eng] | 지정한 속성 값으로 시작하는 요소 |
| [속성 $= 값] | [href $= xls] | 지정한 속성값으로 끝나는 요소 |
| [속성 \*= 값] | [href \*= w3] | 지정한 속성값의 일부가 일치하는 요소 |

#### **선택자** 종류2

보통은style 내에서 아래에 쓰인 값이 강하게 적용됨.

| **style** | **body** | **비고** |
| --- | --- | --- |
| #aaa{내용} | id="aaa" | 하나만 지정. |
| .bbb{내용} | 클래스 선택자  class="bbb" | 그룹. 여러개 지정 가능.  스타일에서 이름앞에 "."으로 표시한다.  내가 원하는 곳에 특정 이름을 부여해서 스타일 적용함  리고 원하는 요소가 똑같이 적용가능  지정하고자 하는 태그에 class="선택자명" 삽입함  첫 글자는 알파벳으로 시작해야 합니다. (a~z 또는 A~Z)  두 번째부터는 알파벳과 숫자 그리고 '-' 또는 '\_'가 사용될 수 있습니다. 대소문자를 구분. |
| \*{내용} | 모든 태그  전체 선택자 | 모든 태그에 내용이 적용됨.  전체 기본값으로 세팅할때 사용 |
| p{내용} | 모든 p 단락 | 모든 p 단락에 내용이 적용됨.  \*{color: red;} 폰트만 적용인데…  바로 위 [\*{border: 색, 선타입, 두께} ]  에서 선 색 적용 안시키니까 선 까지 칠해버리더라.. |
|  |  |  |
| - | <div></div> | 레이아웃 나눔/박스(블럭)구성 <div id="aaa">어서오세요</div> |
| - | <span/</span> | 인라인 요소  <div>와 <p>태그와 함께 웹페이지의 일부분에 스타일을 적용시키기 위해 사용. span 태그안에 아무런 컨텐츠가 없다면 해당 부분은 변화가 없지만 span태그 내부에 객체가 들어가면 그 객체의 크기만큼 공간이 할당 됨. |
| 태그 선택자 |  | 현재 페이지에 있는 특정태그명으로 선택을 해서 스타일을 지정. 태그에게 선택자를 지정하지 않고 스타일에서 지정된 태그명으로 선택이 됨 |

#### **선택자** 종류3

| **특수코드값** | **비고1** | **비고2** |
| --- | --- | --- |
| 선택자A>선택자B; | 자식선택자  tr>th | 선택자A의 자식에 위치하는 선택자 B 선택  tr의 첫번째 th에 해당 |
| 선택자A 선택자B; | 후손선택자 | 선택자A의 후손에 위치하는 선택자 B 선택 |
| :nth-child(수열) |  | 앞에서 수열번쨰 있는 자손 선택 |
| #a>div:nth-child(2) div:nth-child(3)  { background-color: firebrick } | id=a 가 있는 곳의 자손 div의 2번째 자손의  div의 3번쨰 자손에 { } 를 적용한다 | |
| 선택자A+선택자B; | 동위선택자 |  |
| <div id="a">1  <div="b">2  <div>3  <div>4</div>  </div>  <div>5</div>  </div>  <div>6  <div>7</div>  <div>8.1</div>  <div>8.2</div>  <div>8.3</div>  <div>8.4</div>  <div>8.5</div>  </div>  <div>9</div>  <div>10</div>  </div> | #a>div{내용}**=** 2/6/9/10 (자손)  #a div{내용}**=** 2/3/4/5/6/7/8… 등등 아래의 전부 해당 (후손)  #b+div{내용}**=** 6 (동급 중 바로 아래 있는 애)  #a>div:nth-child(3){내용} **=** 9  #a>div:nth-child(2) div:nth-child(3){내용} **=** 8.2 | |
| hover | hover는 항상이 아니라  마우스 올라갔을대만  이라고 조건걸어준거라고 생각 | #id1:hover(띄어쓰기:내부의모든요소탐색)class1을 선택 / 이게되는거야  id1안의 class1인거고 |
|  |  |  |
|  |  |  |

#### **가상클래스** 종류

a태그에 사용된다

link > visited > hover > active

| **특수코드값** | **비고1** | **비고2** |
| --- | --- | --- |
| :link |  | 방문하지 않은 링크에 스타일 적용 |
| :visited |  | 방문한 링크에 스타일 적용 |
| :active |  | 웹 요소를 활성화했을 때의 스타일 적용 |
| :hover |  | 웹 요소에 마우스커서를 올려놓을 때의 스타일 적용 |
| :focus |  | 웹 요소에 초점이 맞춰졌을 때의 스타일 적용 |

#### **특수코드** 추가값

| **특수코드값** | **비고1** | **비고2** |
| --- | --- | --- |
| background-color: 색상; | 배경색 | background-color: rgba(255,0,0,0.4);  /\*rgba 투명값 마지막에 들어간 숫자가 0~1까지 조절가능\*/ |
| \*{border: **색, 선타입, 두께**} | \*{border-color: red;}  \*{border-style: solid;}  \*{border-width: 2px;} | 좌측에서 비워두면 기본값 지정됨 오류 날 수 있으니 채우는걸 추천 |
| margin  중앙정렬할때 행복해진대  애니메이션 안먹을수잇대 | 사회적 거리,, 바깥 뽷  {margin: 20px 50px;} | [상 - 하 - 좌 - 우] [상하 - 좌우]  [상 - 좌우 - 하] [상하좌우]  margin-top, margin-right, margin-bottom, margin-left 하나하나 마진 적용할때  \*\*\* 두 개 요소의 마진값이 중복이 될 때 큰 값이 적용됨 |
| padding | 패딩,, 내부 뽀핳  {padding: 30px 10px 20px;} | [상 - 하 - 좌 - 우] [상하 - 좌우]  [상 - 좌우 - 하] [상하좌우]  \*\*\* 패딩은 안쪽 여백이므로 값이 요소의 가로값, 세로값에서 벗어나면 요소의 크기가 변화됨 |
| border-radius  border-radius:[length]; | {border-radius: 50px;} | 모퉁이 둥글게 효과  Box Model 외각선의 곡률 지정  개별 지정 가능 |
| {**word-break**: break-all;} | 요소에 width나 height를 지정해야 이 속성값들이 적용된다. | 줄바꿈 속성  normal 일반적 규칙에 따름  break-all: 강제 줄바꿈 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### **display** 속성

요소의 종류를 선택하는 속성

inline 요소를 block 요소로 보이게 만들 수도 있고,

block 요소를 inline 처럼 보이게 만들 수도 있습니다.

| **inline 요소** :줄 속에 끼워 넣는 요소 (예 :<span>, <b>, <a>, <img>태그)  **block 요소** : 해당 요소가 속한 줄은 가로 화면 전체100%를 차지하며 앞뒤로 줄 바꿈이 됨  (예 : <p>, <div>, <ol>, <ul>, <table> 등) |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| display: inline  가로배치O. 크기조절X | 기본값으로, 요소를 inline 요소처럼 표시합니다.  따라서 앞뒤로 줄바꿈 되지 않습니다.  (span태그, b태그, I태그) |
| display: block  가로배치X. 크기조절 가능 | 요소를 block 요소처럼 표시합니다.  따라서 요소 앞 뒤로 줄바꿈 됩니다  (가로길이가 기본적으로 100%) |
| display: none  보이지 않음 | 박스가 생성되지 않습니다.  따라서 공간을 차지하지도 않습니다. |
| display: inline-block  가로배치O. 크기조절 O | 요소는 inline인데 내부는 block 처럼 표시함.  즉, 박스 모양이 inline 처럼 옆으로 늘어섬. |

div= header / aside / bside / footer

#### **inline** 속성

요소의 종류를 선택하는 속성

img, a. span, strong

주로 내용부분 즉 화면에 표시한 콘텐츠들

한줄에 여러개가 정렬, 자신만 크기를 인식하여 표시함

width, height를 적용하여 사용할 수 없다.

#### **block** 속성

div, p, ul, li, h1~h6

기본적으로 세로 정렬

한줄에는 하나 차지. 즉 가로폭을 자신이 100%인식함

width, height를 적용하여 사용가능

display: 화면 보여주는 방식

display:none; 화면에서 숨김

display:block; 블록레벨 요소로 만듬 (화면에 보여줌)

display:inline: 인라인 레벨로 만듬 (ul>li의 float: left 효과를 이걸로 대체 가능하더라)

display: inline-block 인라인 레벨에서 가로값과 세로값이 적용됨

#### **Font** 속성

참조: <https://aboooks.tistory.com/144>

| p (font: italic small-caps bold 17px arial, "Times New Roman", sans;}  \* 순서: **font-style** font-variant font-weight font-size **font-family** |
| --- |

| <head>  <meta charset="UTF-8">  <title>Document</title>  <link rel="stylesheet"  type="text/css"  href="style5.css">  **<link href="https://fonts.googleapis.com/css2?family=Manrope:wght@200&display=swap" rel="stylesheet">**  </head> |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| font-style | normal: 기본 값  Italic: 이탤릭체  oblique: 기울어짐 꼴 (인위적으로 비튼 기울임) |
| font-variant | 소문자를 대문자로 만드는 속성  normal: 브라우저 기본 값  small-caps: 소문자로 대문자로 만듬. 일반 대문자 크기보단 작다! |
| font-weight | normal: 기본값  bold: 진하게 표시  100~900: 숫자별로 글꼴 굵기 지정 (400=normal)  bolder: 상위 요소에서 상속된 굵기보다 더 진하게 (상속X)  lighter: 상위 요소에서 상속된 굵기보다 더 흐리게 (상속X)  \* font-family에서 가능한 것 중 가장 가까운 bolder나 lighter이다.  글꼴에 400과 700밖에 없을 경우 700보다 bolder한 경우라도 700표시 |
| font-family | family-name: 글꼴 이름 (arial, verdana, 나눔고딕, 궁서, 굴림 등)  하나 이상의 글꼴 열거시 쉼표 사용  글꼴 이름에 공란이 들어갈 경우 인용부호('') 사용 |
| generic family: 모양이 비슷한 글꼴 그룹 |

#### **font-size** 종류

7가지 단위 - rem, vh, vw, vmin, vmax, ex, ch

| 글꼴 크기 지정  px: 고정된 크기 단위. 크기 조정 안됨  pt: 고정된 크기 단위 (주로 인쇄 매체) 크기 조정 안됨  em: 웹 문서에서 사용. 크기 조정 가능. 1em이 12pt 16px 100%와 동일  %: em과 비슷하게 상대적 크기 조절 가능 |
| --- |

| **특수코드값** | **설명** | **비고** |
| --- | --- | --- |
| vh | 뷰 폰트 높이의 1%  500px 너비의 뷰폰트에서 1vw의 값은 5px  브라우저를 늘려서 윈도우 너비가 850px가 될 때 1vw = 8.5px가 된다. | IE9부터 지원하는 속성. (Can I Use : vw)  이들 단위는 뷰포트 영역 전체를 차지하게 하거나, 그 일부분만 차지하게 하는 데 유용하게 쓸 수 있다. 당장 100vw, 100vh만 써도 너비와 높이를 꽉꽉 채운다.  calc()와 결합하면 좋은 시너지를 낼 수 있다. 헤더영역 높이가 50px이고 컨텐츠 영역 높이를 '헤더를 제외한 나머지 높이 전체'로 만들고 싶다면, height: calc(100vh - 50px);을 지정해주면 된다.  height: 100%로 지정해도 100vh처럼 전체 높이를 차지할 수 있지 않나요? 라 생각할 수 있지만 %는 부모 요소를 기준으로 계산한다는 점에서 차이가 있다. |
| vw | 뷰 폰트 너비의 1%  500px 너비의 뷰폰트에서 1vw의 값은 5px  브라우저를 늘려서 윈도우 너비가 850px가 될 때 1vw = 8.5px가 된다. |
| vmin | vw와 vh 중 더 작은 것을 적용  vmin은 IE9부터 사용 | 이러면 vw나 vh랑 뭐가 다른가 싶지만, 잘 사용하면 '언제든 화면에 보이는 요소'와 '언제든 화면을 꽉 채우는 요소'를 쉽게 만들 수 있다.  vmin은 둘 중 더 작은 값을 택하기 때문에, 작아진 높이값에 반응해서 잘리지 않고 화면에 표시. 즉, 너비/높이값 중 무엇이 바뀌던 상관없이 화면에 잘 나타낼 수 있습니다. |
| vmax | vw와 vh 중 더 큰 것을 적용  IE에서 지원되지 않으며 Edge부터 사용 |
| ex | 동적인 단위  '1em = 부모의 폰트 크기' | 대부분의 브라우저에서 폰트 크기의 디폴트 값은 16px  따라서 별도로 스타일링을 해주지 않았다면 1em = 16px. 2em이라면 32px  만약 부모의 폰트 크기가 20px이라면요?  1em = 20px이 되고, 5em = 100px으로 커진다.  즉, 부모의 폰트 크기를 키우면 자식 요소의 크기도 덩달아 커지게끔 할 수 있다 |
| rem | 최상위 폰트를 기준으로 한 단위 | 1rem=html의 폰트크기  예) html의 폰트 크기가 12px일 경우 1rem=12px |

#### **text** 관련 스타일

| **특수코드값** | **비고** |
| --- | --- |
| line-height (=행간) | "normal, 숫자, 숫자em, %": 줄간격  /\*비율(1:100%), 비율은 글자 기준으로 계산됨\*/ |
| text-indent | "숫자em, 퍼센트, 숫자" : 들여쓰기, 음수 사용시 내어쓰기  /\*부모를 기준으로 100%\*/ |
| text-align | "left, right, underline, overline, line-through" : 텍스트나 이미지 수평정렬  \*\*\* 자신의 자식요소를 정렬할때 사용한다.대신 조건이 있음.  자신의 가로값이 선언되어 있어야 한다. |
| text-decoration | "none, underline, overline, line-through" : 줄 긋기, 유전되지 않음 |
| letter-spacing | "normal, 숫자em, 숫자" : 글자간격 |
| word-spacing | 단어와 단어 사이 여백 설정 |
| text-transform | "capitalize, uppercase, lowercase, none"  : 영문자 변환 첫글자만 대문자 소문자 |
| vertical-align | "baseline, sub, sup, middle, text-top - top, text-bottom, bottom"  : 텍스트나 이미지 수직정렬 // 인라인요소 |
| text-shadow:  [dx] [dy] [bulr] [color];  (box-shadow도 존재함) | 텍스트에 그림자를 지정  dx : 본체와 그림자의 가로방향 거리  dy : 본체와 그림자의 세로방향 거리  bulr : 그림자의 흐려짐 정도  color : 그림자의 색상 |

**border**  속성

**border-style : 테두리의 스타일 지정**

| <div></div> | html 내부에 다이렉트로 선언하는 방법. (강한 압력. 좋지 않음) |
| --- | --- |
| div {width: 50px;  height: 50px;  border: 1px solid black;  border-left-color: transparent;  border-right-color: transparent;  border-top-style: dotted;  } | <div class="border-styles">  <p style="border-style: solid"> solid </p>  <p style="border-style: dotted"> dotted </p>  <p style="border-style: dashed"> dashed </p>  <p style="border-style: double"> double </p>  <p style="border-style: groove"> groove </p>  <p style="border-style: ridge"> ridge </p>  <p style="border-style: inset"> inset </p>  <p style="border-style: outset"> outset </p>  </div> |

| **특수코드값** | **비고** |
| --- | --- |
| none | 테두리 없음 |
| solid | 실선 |
| dotted | 도트(점선) |
| dashed |  |
| double | 두 줄선 |
| groove | 홈이라는 뜻으로 움푹 파인 효과를 가진 테두리를 지정 |
| ridge | 산등성이처럼 솟은 부분이란 뜻 |
| inset | top(상) + left(좌) 테두리가 진하게 표시되어 안으로 들어간 느낌을 지정 |
| outset | right(우) + bottom(하) 테두리가 진하게 표시되어 밖으로 나온 느낌을 지정 |

**border-width : 두께 지정**

| **특수코드값** | **비고** |
| --- | --- |
| medium - default | 중간 테두리(3px) |
| thick | 굵은 테두리로 지정(5px) |
| px 지정 | 가장 많이 사용 |

**border: 축약속성**

| 보더가 생기는 위치는 박스가 있으면 박스 바깥쪽에 생성  보더의 일부 테두리를 없애려면  1) 지정된 값을 0px로  2) 지정된 값 아래에 border-top:0px; 처럼 재설정  3) border-collapse : collapse - 테두리를 하나로 합치기 지정  (border-spacing이나 빈 셀 속성은 무시)  border-spacing : px - 테두리선과 요소의 간격을 지정.  table border와 td border 사이의 간격. |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| border-top | 상 |
| border-left | 좌 |
| border-right | 우 |
| border-bottom | 하 |

**border-color : 색상 지정(1px)**

| 컬러 지정은 상하좌우 각각 지정 가능 (top right bottom left)  border-color:red green blue pink; (순서:top right bottom left) |
| --- |

#### **Visibility** 속성

| 태그의 가시성을 결정. 기본 값은 inherit |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| visible | 보임 |
| hidden | 숨김 (자신의 영역은 계속 차지) |
| collapse | 겹치도록 지정  (테이블의 행과 열 요소만 지정할 수 있으며, 그 외 요소의 지정하면 hidden으로 해석)  모양을 유지하고 없애듬. 히든과 같은 속성  (콜스페이스 크기는 유지되는데 보더 속성에 의해서 약간 짜푸된다. 의미를 모르겠는 속성. 차라리 히든 쓰는게 낫지 않을까???? 잘 안씀!!!) |
| inherit | 부모 요소의 값을 상속 |

#### **background** 속성

background: url(images/5.jpg) no-repeat 0 0;

/\*위 세가지 속성을 한줄로 표현. [주소, 반복여부, 위치]\*/

| **특수코드값** | **비고** |
| --- | --- |
| background-image: url(); | 링크 내 이미지 불러와 배경으로 지정 |
| background-repeat: "repeat, repeat-x, repeat-y, no-repeat" | 배경이미지 반복 여부 |
| background-size: cover; | width height 사이즈 맞게 조절,  사이즈 딱 맞게 만듬. |
| background-size: contain; | width height 사이즈 맞게 조절,  비율조절해서 사이즈 딱맞게 |
| background-attachment "scroll, fixed" | 화면이 스크롤 될 때 이동여부 |
| background-position: "top, bottom, left, center, right, 가로%, 세로%" | 이미지 반복 여부 가로p 세로p (음수사용가능) |

#### **position** 속성

| **\*, html, body { margin:0; padding:0;}**  기본적으로 html에는 margin이랑 padding이 약간 들어있다  기본값이 0이 아니기 때문. 그래서 위의 것을 기입하고 시작하는 게 좋음. |
| --- |

| 위치값이 **top left right bottom** 기준  absolute랑 fixed 는 블록 특성인 100% 가로길이 이거 없앰!!! 그래서 크기 쪼꼬매짐 |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| static  position: static; | 초기값으로 위치를 지정하지 않을 때와 같다  앞에 설정된 position을 무시할 때 사용할 수 있음  top, bottom, left, right 속성값이 적용되지 않음  {position: static; top: 20px; left: 30px;}  /\*static 은 기본값. 위치 정할 수 없음. 상속됐다? 안에 있어서 쏙 들어감\*/ |
| relative  position: relative; | 위치 계산을 할 때 static의 원래 위치부터 계산한다  위(top), 아래(bottom), 좌(left), 우(right)의 위치를 같이 설정할 수도 있음  {position: relative; top: 20px; left: 30px;}  /\*<원래있던 위치>를 기준으로 좌표 설정.\*/  absolute는... 상속받아야하는 애라 위에 무조건 relatice가 있어야한다!!  /\*안에 요소 중 포지션 선언했다면 기준 설정\*/ |
| absolute;  position: absolute; | relative와 달리 문서의 원래 위치와 상관없이 위치를 지정할 수 있다  하지만 가장 가까운 상위 요소를 기준으로(단, static은 제외) 위치가 결정됨  상위 요소가 없으면 위치는 html를 기준으로 설정  {position: absolute; top: 20px; right: 30px;}  /\*절대값. 무조건 저 위치. 창 크기 수정해도 그 위치 고정\*/ |
| fixed;  position: absolute; | 브라우저 화면의 상대 위치. (상위 요소에 영향을 받지 않음)  따라서 화면이 바뀌더라도 고정된 위치를 설정할 수 있다  {position: fixed; top: 20px; right: 30px;}  /\*상대값. 스크롤 상관없이 항상 화면 그곳에 존재. absolute가 사라져도 얜 있음\*/ |

#### **overflow** 속성

| 요소의 박스에 내용이 더 길 때 어떻게 보일지 선택하는 속성 |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| visible | 기본값으로 내용이 더 길어도 그대로 보인다.  (내용이 흘러넘침) |
| hidden | 내용이 흘러넘치면 자른다. 자른 부분은 보이지 않음. |
| scroll | 내용이 넘치지 않아도 항상 스크롤바가 보임. |
| auto | 내용이 잘릴 때만 스크롤바가 보임. |

#### **z-index** 속성

| 어느 객체가 앞으로 나오고, 뒤에 나올지 배치 순서를 결정하는 속성  z-index는 position (relative, absolute, fixed)속성이 적용된 요소에서만 작동함 |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| auto | 기본값으로 z-index를 지정하지 않은 것과 같음 |
| number | 배치 순서를 결정, 숫자가 낮을수록 뒤에 배치되며 숫자가 높을수록 앞에 나옴  -1 처럼 음수도 가능 |
| initial | 기본값으로 설정함 |

#### **transform** 속성

| transform:요소의 형태를 변경함 |
| --- |

| **특수코드값** | **비고** |
| --- | --- |
| translate([dx],[dy]) | 지정한 크기 만큼 이동 시킴 |
| scale([ds]) | 지정한 배율 만큼 확대함(1은 현재 선언된 원래 크기) |
| rotate([deg]) | 지정한 각도만큼 회전 시킴 |
| skew([degx],[degy]) | 지정한 경사로 기울림 |

#### **float** 속성

| css에서 정렬하기 위해 사용되는 속성 (※절대 위치 요소는 float 속성을 무시)  문서에 사진과 그림이 있을 때, 그림을 왼쪽이나 오른쪽으로 띄워서 정렬하거나 각 객체를 오른쪽이나 왼쪽으로 정렬하여 전체 문서를 배치(layout)할 때도 사용할 수 있음  포지션이 뭉개질 수 있음: 방지책 4가지 clearfix (버그해결법)  2~4번은 잘 사용하지 않는다. (각각 단점이 있음) 그런 게 있구나 알고 1번만 사용!   | 1. 가상 요소(선택자) ::after 사용(권장)  2. overflow속성 사용  3. 빈 엘리먼트로 clear속성 적용  - clear: float 배치 설정을 무시하고 새롭게 배치를 하는 속성  - clear: left;  - clear: right;  4. float로 대응 | | --- | |
| --- | --- |

| **특수코드값** | **비고** |
| --- | --- |
| none | 띄우지 않음 (기본값) |
| left | 왼쪽에 띄움 |
| right | 오른쪽에 띄움 |
| initial | 기본값으로 설정함 |
| inherit | 부모 요소로부터 상속함 |

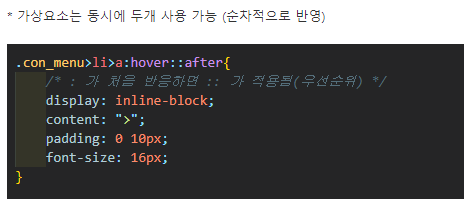
#### **가상 선택자** 속성

가상 클래스(class)와 가상 요소(element)를 편의상 함께 칭할 때 사용, 정식 명칭은 아님.

폼(form, input...) 태그와 이미지 태그(img)는 가상선택자가 적용되지 않는다.

| 가상 클래스(Pseudo-class) | 요소에 직접적으로 클래스를 부여하지는 않았지만, 요소의 상태에 따라서 클래스를 적용한 것처럼 효과를 다르게 줄 수있다. 가장 많이 쓰는 예시는 a태그의 효과를 바꾸기 위해 a:hover, a:link 이고, 이때 a뒤에 쓰인 :hover, :link를 가상 클래스라고 한다. |
| --- | --- |
| 가상 요소(Pseudo-elements) | 가상의 요소를 만들고 내용을 넣어 출력하겠다는 것이다. 보통 ::after, ::before을 자주 사용한다. |

CSS2.1에서는 가상 클래스와 가상 요소 모두 싱글콜론(:)을 사용했지만, CSS3 부터는 둘을 구분하기 위해 가상 클래스는 싱글콜론(:), 가상 요소는 더블콜론(::)을 사용한다.



#### **clearfix** 속성

float 속성을 적용한 요소의 부모요소에 ::after를 사용

예시로 float 속성 적용된 상위 부모에게 다음과 같은 class를 적용해주면 된다.

.clearfix::after{

display: block; content: ''; clear: both;

}

태그 뒤에 내용이 빈(content: '') 테이블을(display: table) 만들고, float: left, float: right를 초기화 시키겠다는 뜻이다. 이렇게 작성을 해주면 뒤에 다른 내용을 작성을 하면 위로 딸려올라가지 않고 다음에 위치하게 된다.

여기서 display: table 대신 display: block을 쓰는 것도 가능하고, clear: left를 쓰면 float: left를, clear: right를 쓰면 float: right를 초기화 한다는 뜻인데 **항상 clear: both를 사용**해도 상관이 없다.

#### **flex** 속성

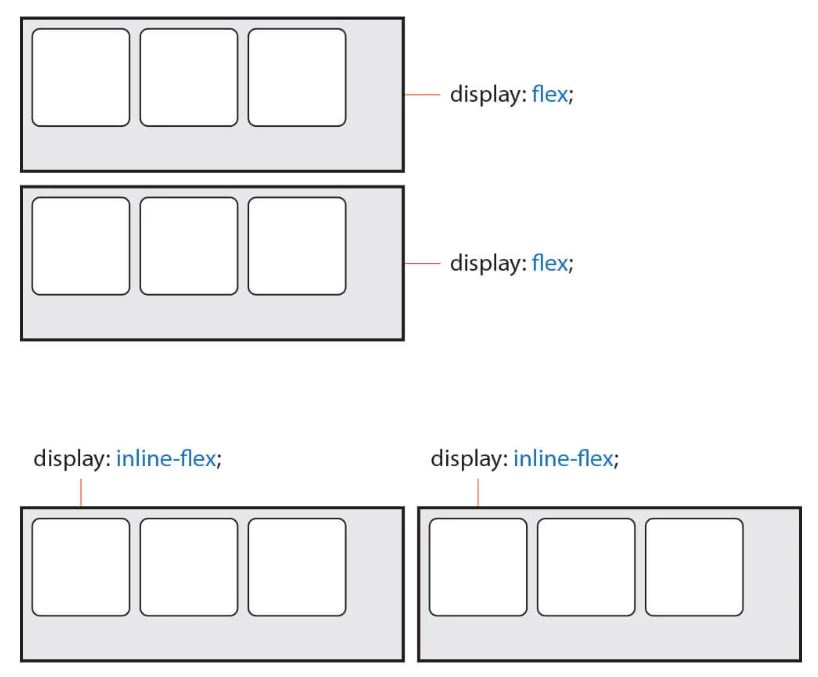
| 플렉서블 레이아웃(flex)  대부분의 웹사이트는 수직 구성이며 "위-아래"로 스크롤 하여 사용됩니다. 레이아웃을 구성할 때 가장 많이 사용하는 요소들이 대부분 블록개념으로 표시되며 이는 뷰에 수직으로 쌓이기 때문에 수직 구성은 상대적으로 쉽게 구성할 수 있습니다. 하지만 수평 구성의 경우는 속성이 명확하지 않기 때문에 <table>, float, inline-block으로 구성했습니다.  따라서 쉽게 수평을 구성할 수 있는 flex(플렉서블 레이아웃)이 차선택으로 만들어졌습니다. |
| --- |

| Flex는 수평이 될 요소들의 Container(box-container)에 display: flex;를 적용. 세부 속성이 필요하지 않은 경우도 많기 때문에 상당히 쉽고 빠르게 수평 요소를 구성할 수 있음.  flex의 구성요소: Container, Items  **Container:** Items를 감싸는 부모 요소. 각 Item을 정렬하기 위해 필수적.  ※ Container와 Items에 적용하는 속성이 구분됨  **Container:** display, flex-flow, justify-content 등의 속성 사용  **Items:** order, flex, align-self 등의 속성 사용 |
| --- |

| **속성** | **의미** |
| --- | --- |
| flex | Block 특성의 Flex Container를 정의  지정된 Flex Container는 Block 요소와 같은 성향(**수직 쌓임**) |
| inline-flex | Inline 특성의 Flex Container를 정의  지정된 Flex Container는 Inline(Inline Block) 요소와 같은 성향(**수평 쌓임**). |

여기서 말하는 수직과 수평 쌓임은 Items가 아니라 Container라는 것에 주의.

두 값의 차이는 내부에 Items에는 영향을 주지 않음



#### **Flex container** 속성

| **속성** | **의미** |
| --- | --- |
| display | Flex Container를 정의 |
| flex-flow | flex-direction(주축 설정, 기본값 row)  flex-wrap(줄바꿈설정, 기본값 nowrap)의 단축 속성 |
| flex-direction | Flex Items의 주 축(main-axis)을 설정 |
| flex-wrap | Flex Items의 여러 줄 묶음(줄 바꿈) 설정   | nowrap | 모든 items를 여러줄로 묶지 않음 (한줄로 표시, 기본값) | | --- | --- | | wrap | items을 여러줄로 묶음 | | wrap-reverse | items를 wrap의 역방향으로 여러줄 묶음 | |
| justify-content | 주 축(main-axis)의 정렬 방법을 설정   | flex-start | items를 시작점으로 정렬 | | --- | --- | | flex-end | items를 끝점으로 정렬 | | center | items를 가운데 정렬 | | space-between | 시작 item은 시작점, 마지막 item은 끝점에 정렬,  나머지 item은 사이에 고르게 정렬 | | space-around | items를 균등한 여백을 포함하여 정렬 | |
| align-content | 교차 축(cross-axis)의 정렬 방법을 설정(2줄 이상이고 여백이 있을때만 적용됨)  ※ items가 한 줄일 경우 align-items 속성을 사용해야함   | strectch | containger의 교차축을 채우기 위해 items를 늘림  (width 혹은 height 값이 auto일 경우 변경됨) | | --- | --- | | flex-start | items를 시작점으로 정렬 | | flex-end | items를 끝점으로 정렬 | | center | items를 가운데 정렬 | | space-between | 시작 item은 시작점, 마지막 item은 끝점에 정렬,  나머지 item은 사이에 고르게 정렬 | | space-around | items를 균등한 여백을 포함하여 정렬 | |
| align-items | 교차 축(cross-axis)에서 Items의 정렬 방법을 설정(1줄)  ※ items가 두 줄 이상일 경우 align-content 속성이 우선됨 |

| flex-wrap  부모요소적용 | 플렉스 라인에 더 이상의 여유가 없을 때, 플렉스 요소의 위치를 다음줄로 넘길지 여부를 설정   | nowrap | 기본 설정, 플렉스 요소가 다음 줄로 넘어가지 않습니다. 대신 플렉스 요소의 너비를 줄여서 한 줄에 모두 배치 | | --- | --- | | wrap | 플렉스 요소가 여유 공간이 없으면 다음 줄로 넘겨서 배치 | | wrap-reverse | 플렉스 요소가 여유 공간이 없으면 다음 줄로 넘어가서 배치됩니다. 단, 아래쪽이 아닌 위쪽으로 넘어감 | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| justify-content  부모요소적용 | 플렉스 요소의 수평 방향 정렬 방식을 설정   | flex-start | 기본 설정, 플렉스 요소는 플렉스 컨테이너의 앞쪽에서부터 배치 | | --- | --- | | flex-end | 플렉스 요소는 플렉스 컨테이너의 뒤쪽에서 부터 배치 | | center | 플렉스 요소는 플렉스 컨테이너의 가운데서부터 배치 | | space-between | 플레스 요소는 요소들 사이에만 여유 공간을 두고 배치 | | space-around | 플렉스 요소는 앞, 뒤 그리고 요소들 사이에 모두 여유 공간을 두고 배치 | | space-evenly | 아이템들의 사이와 양끝에 균일한 간격을 생성 | |
| align-items  부모요소적용 | 플렉스 요소의 수직 방향 정렬 방식을 설정   | stretch | 기본 설정, 플렉스 요소의 높이가 플렉스 컨테이너의 높이와 같게 변경된 뒤 연이어 배치 | | --- | --- | | flex-start | 플렉스 요소는 플렉스 컨테이너의 위쪽에 배치 | | flex-end | 플렉스 요소는 플렉스 컨테이너의 아래쪽에 배치 | | center | 플렉스 요소는 플렉스 컨테이너의 가운데 배치 | | baseline | 플렉스 요소는 플렉스 컨테이너의 기준선에 배치 | |
| align-self  자식요소적용 | 플렉스 컨테이너의 align-items 속성보다 우선 적용됩니다.  이 속성을 사용하면 플렉스 요소마다 서로 다른 align 속성값을 설정할 수 있습니다. |

order

플렉스 컨테이너 안에 있는 플렉스 요소들의 순서를 설정합니다.

flex-grow

플렉스 컨테이너 아이템들이 차지할 너비들에 대한 증가형 숫자를 지정합니다.

flex-shrink

플렉스 컨테이너 아이템들이 차지할 너비들에 대한 감소형 숫자를 지정합니다.

flex-basis

플렉스 컨테이너 아이템의 길이를 지정합니다.

\* flex : flex-grow flex-shrink flex-basis 을 한번에 설정

flex: 2 2 0

#### **Flex items** 속성

| **속성** | **의미** |
| --- | --- |
| order | flex item의 순서 설정 (음수허용)  숫자가 클 수록 순서가 밀림. HTML구조와 상관없이 순서 변경 가능. |
| flex | flex-grow, flex-shrink, flex-basis (Item의 너비(증가, 감소, 기본))의 단축 속성   | flex-grow | item의 증가 너비 비율을 설정 | 0 | | --- | --- | --- | | flex-shrink | item의 감소 너비 비율을 설정 | 1 | | flex-basis | item의 (공간 배분전) 기본 너비 설정 | auto |   flex-grow를 제외한 개별 속성은 생략 가능. flex: 1;은 flex-grow: 1;과 같다.  flex-basis의 기본값은 auto이나, 단축 속성인 flex에서 그 **값을 생략할 경우 0이 적용**.  다시 정리하면 flex: 1; 혹은 flex: 1 1;은 flex: 1 1 0;이 됨. (주의!) |
| flex-grow | Flex Item의 증가 너비 비율을 설정. 숫자가 크면 더 많은 너비 증가.  Item이 가변 너비가 아니거나, 값이 0일 경우 효과 X (기본값 0) |
| flex-shrink | Flex Item의 감소 너비 비율을 설정. 숫자가 크면 더 많은 너비 감소.  Item이 가변 너비가 아니거나, 값이 0일 경우 효과 X (기본값 1) |
| flex-basis | Flex Item의 (공간 배분 전) 기본 너비 설정 |
| align-self | 교차 축(cross-axis)에서 Container 내 모든 Items의 정렬 방법을 설정  ※ 일부 Item만 정렬 방법을 변경하려고 할 경우 align-self를 사용 가능.  이 속성은 align-items 속성보다 우선됨   | auto | Container의 align-items 속성을 상속받음 | | --- | --- | | strectch | containger의 교차축을 채우기 위해 items를 늘림  (width 혹은 height 값이 auto일 경우 변경됨) | | flex-start | items를 시작점으로 정렬 | | flex-end | items를 끝점으로 정렬 | | center | items를 가운데 정렬 | | baseline | item을 문자 기준선에 정렬 | |

#### **Font Awesome** 속성

아이콘으로 사용되는 웹폰트 종류 중 하나 (가장 잘 알려져있음)

**※아이콘 폰트:** 이미지가 백터 그래픽 형태로 되어 있어서 크기를 늘리든 줄이든 상관없이 선명한 화질을 보장받을 수 있고 색상도 일반 웹폰트처럼 CSS에서 color 속성으로 변경을 할 수가 있다. 다만 아이콘이 폰트형식이기 때문에 단색으로 밖에는 설정할 수 없음.

| **설치방법** | |
| --- | --- |
| CDN | 특별히 아이콘을 커스터마이징 할 일이 없을 때 사용.  웹사이트의 <head> 태그영역 안에 아래의 코드를 삽입 |
| 다운로드 | 약간의 커스터마이징을 하거나  오프라인상태에서도 사용하기 위해서는 아이콘 폰트를 다운로드 받아서 설치   | 1. 해당 URL(https://fontawesome.com/start)에서 DOWNLOAD 버튼 클릭  2. 팝업창에서 가장 아래에 있는  No thanks, just download Font Awesome 4를 클릭.  초록색 버튼의 Font Awesome Pro는 유료 버전.  3. 다운로드한 아이콘 폰트를 웹사이트의 적절한 위치에 저장을 합니다.  4. 웹사이트의 <head> 태그영역 안에  font-awesome.min.css 파일 위치를 아래와 같이 링크. | | --- | |

| **사용방법** | |
| --- | --- |
| <i>태그 | <i>태그에 class명으로 아이콘의 종류를 선택 |
| class | fa는 font awesome의 앞글자로 Font Awesome을 사용한다는 선언이고 그 뒤에는 fa-다음에 나오는 글자는 아이콘명을 의미한다. 아이콘은 Font Awesome 페이지에서 아이콘명을 찾아서 적용 |
| 크기 | fa-lg, fa-2x, fa-3x, fa-4x, fa-5x CLASS를 적용 |
| 테두리  위치 | fa-border CLASS로 아이콘의 테두리를 표시 할 수 있고  fa-pull-left / fa-pull-right CLASS로 아이콘을 좌측 / 우측으로 붙일 수 있다. |
| 회전  (애니메이션) | fa-spin CLASS를 적용하여 아이콘에게 움직이는 회전 애니메이션을 적용 |
| 회전(고정)  상하좌우반전 | fa-rotate-90, fa-rotate-180, fa-rotate-270,  fa-flip-horizontal, fa-flip-vertical CLASS를 적용 |

#### **미디어쿼리(유동형 패턴)** 속성

반응형 웹페이지를 작성할 때 유용한 구문

| @media media-type and (media-feature-rule) /\*여기 공백 꼭 들어가야함\*/ {  /\* CSS rules go here \*/  ) |
| --- |

| **사용방법** | |
| --- | --- |
| css 내부에 삽입 | <style>  @media (max-width: 700px) {  .container{  margin:0px; padding: 0px;  }  }  </style> |
| 링크로 연결 | <link red="stylesheet" media="(max-width: 700px)" href="test.css" /> |

| **지정 가능한 미디어 유형**  \* 미디어 유형은 선택사항입니다. 미디어 쿼리에 미디어 유형을 표시하지 않으면 미디어 쿼리는 기본값으로 모든 미디어 유형에 대한 것으로 해석됩니다 | | |
| --- | --- | --- |
| 너비와 높이 | min-width와 max-width, width 등 | @media screen and (width: 600px) {  body {  color: red;  }  } |
| 방향성  (휴대폰 or 태블릿 등) | 세로모드  :Portrait(포트레이트) 모드  가로모드  : Landscape(랜드스케이프) 모드 | @media (orientation: landscape) {  body {  color: rebeccapurple;  }  } |

| **media feature(미디어 특징)** | |
| --- | --- |
| width  화면 영역 너비 | 예 1: @media (min-width: 20em) {...}  \* 모든 장치에서 최소 너비 20em 이상이면 적용    예2 : <link rel="stylesheet" media="screen and (min-width: 500px) and (max-width: 800px)  href="test.css" />  \* 스크린 장치 최소 화면이 500px보다 크고 800px보다 작을 때 스타일 적용 |
| device-aspect-ratio  출력 장치의 가로세로 비 | 예: @media screen and (device-aspect-ratio: 16/9) { ... }  \* 장치 가로 세로 비가 16:9일 때 적용. |

| **연산자** | |
| --- | --- |
| 기본(all) | @media (min-width: 700px) {background-color: yellow;}  = 「@media all and (min-width: 700px) {...}」  모든 유형의 장치이며 최소너비 700px 일때 스타일을 적용  \*미디어 타입이 생략되어 있지만, 미디어 타입의 기본값은 all |
| and | @media (min-width: 700px) and (orientation: portrait) { ... }  모든 유형의 장치이며 최소 너비 700px 이상이며,  방향이 세로 모드일 때만 적용  @media print and (min-width: 700px) and (orientation: landscape) { ... }  프린트 장치이며, 최소 너비가 700px이상이며,  방향이 가로일 때 적용함.  \*새로운 미디어 특징들을 추가할 때마다 and 연산자를 사용 |
| 쉼표 | @media (min-width: 700px), print and (orientation: landscape) { ... }  모든 장치에서 최소너비 700px이상일 때 적용하거나,  프린트 장치에서는 가로 방향일 때만 적용하겠다는 뜻.  \*쉼표는 각각 개별 미디어 쿼리로 인식 |
| not | @media not all and (color) { ... }  not은 전체 미디어 쿼리를 수식.  즉 not은 all and (color)를 포함하여 부정함.  @media not (all and (color)) { ... } 와 같은 말  모든 색상 장치에서 이 스타일을 적용하지 않겠다는 뜻.  \* 쉼표로 분리하여 사용할 때, 미디어 구문은 개별 미디어 쿼리로 인식하므로 not은 쉼표 이후에 영향을 미치지 않는다. |

#### **transition** 속성

트랜지션. 지속시간을 부여해서 css 속성이 변하는 속도를 조절하는 것

속성의 변화를 **부드럽게** 표현할 수 있다. (ex. 너비가 100px 도형이 마우스 오버할 경우 200px로 늘어나거나, 14px였던 글씨가 20px로 확대되는 등) 보통 트렌지션은 요소에 마우스 오버했을 때 변화가 나타난다.

모든 속성의 기본값은 0.

{ transition: transition-property, transition-duration, transition-timing-function, transition-delay; }

| transition-delay: 애니메이션의 지연시간을 지정함  transition-duration : 애니메이션이 실행시간을 지정함  transition-property : 어떤 속성을 변형할지 지정함  transition-timing-function : 애니메이션이 속도 형태를 지정함  linear:시작에서 끝까지 똑같은 속도로 진행  ease:처음에는 천천히 시작하고 점점빨라지다가 마지막엔 천천히 끝남  ease-in:시작은 느리게  ease-out:느리게 끝냄  ease-in-out:느리게 시작하고 느리게 끝냄  cubic-bezier:직접베이직 함수를 정의해서 사용  n에서 사용할 수 있는 값은 0~1사이입니다 |
| --- |

| **속성명** | **설명** |
| --- | --- |
| transition-property | CSS 속성. 움직일 수 있는 속성은 어지간해선 다 가능.   | none | 아무속성도 트렌지션 효과도 나타나지 않음 | | --- | --- | | all | 기본값. 모든 속성에 트랜지션 효과 나타남 | | property | 트랜지션 효과를 적용하고 싶은 css 속성  하나 이상의 속성 나열시 **쉼표로 분리** |   ▼ 예시   | div{ **width: 100px;**  background-color:yellow;  transition-property:width 3s; /\* transition-duration 속성은 항상 지정\*/ }  div:hover{  **width: 200px**;  } | | --- | |
| transition-duration | 트랜지션 실행 기간 (\*해당 속성은 필수 지정)  해당 설정을 지정하지 않을 경우 기본값 0s 적용되어 효과가 나타나지 X  ▼ 예시   | div{ width:100px;  font-size:14px;  **transition-property: width, font-size;**  transition-duration:**3s, 5s;** /\*width에 3초, font-size에 5초가 부여됨\* /  }  div:hover{ width: 200px;  font-size:20px;  } | | --- | |
| transition-timing-function | 시기 선택. 트랜지션이 실행되는 동안 속도 변화 설정   | ease | 기본값. 천천히 시작해서 빠르게 진행하고 천천히 끝남. 약강약. ( 0.25, 0. 1, 0.25, 1 ) | | --- | --- | | linear | 처음과 끝이 동일한 속도로 진행 ( 0, 0, 1, 1 ) | | ease-in | 천천히 시작 ( 0.42, 0, 1, 1 ) | | ease-out | 천천히 끝 ( 0, 0, 0.58, 1 ) | | ease-in-out | 천천히 시작해서 천천히 끝남  ( 0.42 , 0, 0.58 , 1 ) | | cubic-bezier | 직접 값을 지정. 0~1까지 (n, n, n, n)  \* <http://www.roblaplaca.com/examples/bezierBuilder/#> | |
| transition-delay | 지연. 언제 시작할지 지정. n초 뒤 시작 |

#### **키프레임(@keyframes)** 속성

CSS 애니메이션에서 구간별로 어떤 스타일을 적용시킬지 정하는 문법

Animation 속성과 자주 쓰임

| **속성명** | **설명** |
| --- | --- |
| animation-name | 사용자가 직접 지정한 이름, @keyframes 가 적용될 애니메이션의 이름 |
| 스테이지 | from - to 로 0~100% 의 구간 |
| CSS 스타일 | 각 스테이지(구간)에 적용시킬 스타일 |

▼ 애니메이션 명이 fadeOut 인 애니메이션에 0% 일 때 스타일을 opacity 를 1로 지정하고, 100% 일 때는 0 으로 지정하는 키프레임. 아래는 동일 내용이다.

| **스테이지 %** | **스테이지 from to** |
| --- | --- |
| @keyframes fadeOut {  0% {  opacity: 1;  }  100% {  opacity: 0;  }  } | @keyframes fadeOut {  from {  opacity: 1;  }  to {  opacity: 0;  }  } |

#### **Animation** 속성

keyframes 애니메이션의 키프레임을 지정함

form : 애니메이션의 시작 프레임를 설정함

to : 애니메이션의 시작 프래임를 설정함중간의 키프레임을 %단위로 지정할수 있음

| **속성명** | **설명** |
| --- | --- |
| animation-name | @keyframes 이름 (아래 예제에서는 fadeOut 을 사용) |
| animation-duratuion | 타임 프레임의 길이, 키프레임이 동작하는 시간을 설정할 때 사용 |
| animation-timing  -function | 애니메이션 속도 조절 / 그래프  ( linear / ease / ease-in / ease-out / ease-in-out / cubic-bezier ) |
| animation-delay | 애니메이션을 시작하기 전 지연시간 설정 |
| animation-iteration  -count | 반복 횟수 지정 |
| animation-direction | 반복 방향 설정   | normal | 정해진 순서로 진행 | | --- | --- | | reverse | 반대 순서로 진행 | | alternate | 정해진 순서로 갔다가 반대 순서로 진행 | | alternate-reverse | 반대 순서로 갔다가 정해진 순서로 진행 | |
| animation-fill-mode | | none | **시작:** 프레임(from)에 설정한 스타일을 적용하지 않고 대기  **종료:** 애니메이션 실행 전 상태로 애니메이션 요소와 프로퍼티 값을 되돌리고 종료 | | --- | --- | | forwards | **대기:** 시작프레임에 설정한 스타일을 적용하지 않고 대기  **종료:** 종료프레임(to)에 적용한 스타일을 적용하고 종료 | | backwards | **대기:** 시작프레임(from)에 설정한 스타일을 적용하고 대기  **종료:** 애니메이션 실행 전 상태로 애니메이션 요소의 프로퍼티 값을 되돌리고 종료 | | both | **대기:** 시작프레임(from)에 설정한 스타일을 적용하고 대기  **종료:** 종료프레임(to)에 설정한 스타일을 적용하고 종료 | |

▼ class 명이 memo 인 HTML 태그는 1초의 지연시간 후 4초 동안 사라졌다가 4초 동안 나타났다가를 반복하는, 즉 무한으로 깜빡이는 효과를 만든다. 좌측을 우측처럼 class 명이 memo 인 HTML 태그는 1초의 지연시간 후 4초 동안 사라졌다가 4초 동안 나타났다가를 반복하는, 즉 무한으로 깜빡이는 효과를 만들 수 있다.

| .memo {  animation-name: fadeOut;  animation-duration: 4s;  animation-delay: 1s;  animation-iteration-count: infinite;  animation-timing-function: linear;  animation-direction: alternate;  } | .memo {  animation: fadeOut 4s 1s infinite linear alternate;  } |
| --- | --- |

\* rotate 에 대해서도 추가하기 (돌아오라 인터넷인터넷)

animation-play-state:paused; << 중지하게 만듬

#stop:checked ~ #move{

animation-play-state:paused;

}

#move{

position: relative;

width: 50px; height: 50px;

background-color: orange;

animation: m1 3s linear infinite alternate;

}

@keyframes m1 {

from{left: 0px;}

to{left: 450px;}

}

#### **input(폼문)** 속성

- 그룹선택자

input[type="text"]:focus{

background-color: aquamarine;

border: 3px dotted blue;

}

input[type="radio"],[type="checkbox"]{

width: 100%;

}

속성조건 [] 로 지정해야함. 동시 지정도 가능.

| **특수코드값** | **비고** |
| --- | --- |
| focus | 클릭해서 활성화 -커서 깜빡임- 될 때 나타나는 효과 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |