**<1주차(float)>**

**### text1.html**

- block은 한 칸을 다 차지하기 때문에 width 값을 줄 필요가 없다.

- left는 왼쪽, right는 오른쪽!!

- auto는 쭉 미는 것. margin-left: auto는 왼쪽(자신)에서 오른쪽으로 쭉- 민다.

- margin: 0 auto는 왼쪽 오른쪽 미는 힘이 같기 때문에 중앙정렬된다.

- >는 직계 선택자. 친자만을 선택한다. 손자말고!!

- calc는 자체 계산기. +, -는 앞뒤를 꼭 띄어야 한다.

- nth-child와 nth-of-type의 차이(<https://blog.naver.com/jj601400/222771502720>)

* nth-child: **부모 요소**의 ‘모든’ 자식 요소 중 n번째(같은 부모 요소를 가진 모든 자식 요소를 순서대로 센다.)
* nth-of-type: **부모 태그**의 ‘특정’ 자식 태그 중 n번째(같은 부모 요소를 가졌더라도 특정한 자식요소만 순서대로 센다.)

- ★float 해제법 세가지★

* 1. 부모에 overflow:hidden
* 2. 마지막 태그에 clear:both 두 번째 건 방법만 알아두고 영영 쓸일없다.
* 3. 가상요소를 활용해서 clear:both를 준다.

- alt + shift로 라인복사

- 선택자 작성할 때, 라인복사로 자식 요소 모두를 한 번에 작성한 뒤에 내용 적을 것!!

미리 작성해두면 다시 안 써도 되니 편하다~!

급하다고 앞에 요소 절대로!!! 날려먹지 말기. 광범위하게 선택하면 곂칠 수 있음

- m10을 주고 마지막 요소에서 m을 빼는 방법이 번거로움.. 확장성이 없다. 다음, 다다음에 다르게 처리하는 법 배울 것

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

**### text2.html**

- alt 누른 채 마우스로 특정 부분 선택

- ctl + d로 모양이 같은 부분을 자신의 아래 방향으로 하나씩 선택

- text가 없으면 영역이 안 보이니까 꼭 넣어주자!

- pof(position: fixed)

* 요소가 inline-block이 돼서, 글씨처럼 움직이고 크기 조절이 가능해진다.
* 기준은 창, 보고있는 화면이다! html이 아니라. html은 크기가 정해져 있다.
* 좌표를 꼭 줘야한다.(필수) 좌표가 없는데 부모에 크기가 생기면 그 영향을 받아서 움직이기 때문.

- m10-10-10-10 = m10 상좌우하(시계방향)

- m0-auto 상하 좌우

- m10-10-10 상 좌우 하

- margin을 body보다는 다른 태그에게 주는 것이 훨씬 좋다.

- CSS점수

태그: 1

클래스: 10

가상선택: 10

아이디: 100

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

**### text3.html**

- 검색엔진최적화(seo):

- 웹접근성: 장애인 비장애인 차별없이 동일한 서비스를 제공받아야 한다.

-> 따라서 태그의 사용 순서가 중요하다!

- h1은 홈페이지에서 딱 하나 나온다. 우리나라에선 주로 logo. 홈페이지에서 가장 중요한 타이틀.

- h1에 a를 추가. 메인페이지부터 들어가서 링크를 봐야하니까.

- +는 형제만들어 줌. .sns+.login -> <div class=”sns”>@@@</div><div class=”login”>###</div>

- 모든 영역은 section으로 만든다!! main visual 영역, slide 영역, event 영역, board 영역….

- section은 같은 범주에 속하는 논리적 집합체

- h2는 영역의 제목.

- h2 꼭 넣어줘야 함. h2 숨기는 방법은 나중에 배운다!

- div : 묶는 영역 디자인태그

span : 텍스트 디자인태그

- a를 제외하고, 웬만하면 inline으로 block이나 inline-block을 감싸지 않는다! 억지임

- lorem3은 세단어를 출력

- /는 최상위 디렉토리(서버환경에서만 작동함), ./는 현재 디렉토리, ../는 현재 디렉토리의 상위 디렉토리

* 절대경로: 최상위 디렉토리부터 해당 파일까지 경유한 모든 경로를 전부 기입
* 상대경로: 현재 파일이 존재하는 디렉토리를 기준으로 해당 파일까지의 위치를 작성

- position: relative vs. position: absolute

* por: 기준점
* poa: 기준점 내에서 절대위치

- poa l50p t50p 왼쪽상단 모서리가 기준이라서 정중앙이 아님.

* (tftl) transform: translate(-50%, -50%): 내 너비, 높이의 50%만큼 이동하여 정중앙에 위치하게 된다.

- 로고의 모든 부분에서 클릭이 가능하려면,

* header h1 a에다가 db h100p를 준다.

- 실무에서는 반드시 CSS파일을 따로 만들어서 링크를 건다. 연습할 때는 어디에 넣던 상관없지만!

- 선택자 두 개를 붙여서 쓰는 이유 두가지

* 1. 점수를 올릴 수 있다.
* 2. 중복을 방지한다.

- 앞에 여러 선택자가 배열되어 있어서 더 이상 곂치지 않을 것 같으면 간략하게 써도 된다.

- pof, poa, float, df를 주면 요소가 inline-block화 된다.

- .box1 .pagination span {w50 h50 bg#fff}

* dib를 주면 줄바꿈(엔터값)을 인식하기 때문에 요소 간에 여백이 생긴다.
* 이럴 땐 float를 주어서 dib로 만들면 여백이 사라진다. flex를 배우면 부모에 flex를 주면 끝이다.

- padding은 border 안쪽 영역으로, 나를 포함함. 전체 크기에 영향O.

- (tac) text-align: center / (fwb) font-weight: bold / (fz50) font-size: 50px

- margin은 border 바깥 영역으로, 전체 크기에 영향X.

- li에 margin을 주지않고 a에 줄 것임.

* a는 inline으로, 원래 inline은 block이나 inline-block을 감쌀 수 없지만, 예외로 a만 마크업적으로 가능한데 CSS로는 말이 안 된다.
* 크기를 지정하기 위해서 a에 db를 준다.

- ul을 중앙으로 잡아버리면 안 된다.. 껍데기인 wrap을 추가로 만들어서 중앙정렬한다.

- ★내가 완전 헷갈리는 부분★ (box-sizing: content-box일 때)

* 부모 wrap의 width가 800px이어서 자식 ul, li의 width가 최대 800px로 잡힌다.(크기 한계치가 정해짐. 상속X)

-> 자식에게 border, padding을 아무리 많이 줘도 부모 width 내로 잡힌다.

**- margin** (<https://velog.io/@songjy377/CSS-Negative-Magin>)

- 요소의 시작점을 바꾼다.

- negative margin을 주면 시작점이 원래 위치보다 앞으로 옮겨진다.

* 1. width가 정해진 경우
  + static 요소인 경우에,
    - negative mr, mb는 해당 요소가 위쪽/왼쪽으로 이동한다. 해당 위치에 다른 요소가 있다면 그 다른 요소를 덮는다.
    - negative mr, mb는 뒷 요소의 시작점과 맞물리기 때문에 뒤에 요소가 영향을 받아서 움직인다. 해당 요소는 그대로 있고, 해당 요소 다음에 오는 요소가 있다면 그 모든 요소를 위/오른쪽으로 끌어당긴다.
* 2. widht가 부모의 크기로 정해진 경우
  + negative ml mr로 width를 확장, negative ml mr로 height를 확장할 수 있다.
* 3. float 요소의 경우
  + float 요소는 뒤에 오는 float 적용되지 않은 요소는, float 요소를 인식하지 못 하고 자신의 영역을 확보한다. 또 float 요소의 영역을 덮지 않는다.
    - 이를 이용해서 float 반대 방향으로 음수 마진을 줘서 컨텐츠를 끌어당길 수 있다.
* li에 p, m을 주지 않았다. 순수한 content다.
* width는 content의 크기이다. border-box일 때 width는 border까지이다. (content+padding+border)
* padding은 border 내부 영역, margin은 border 바깥 영역.
* .box2 ul li에다가 (p) %를 주었기 때문에, .box2 ul에 mr-10을 주면 li 크기가 유동적으로 변한다. (px이면 변화X)

- object-fit

* 요소의 크기에 맞게 img와 video 크기를 조절한다.
* object-fit: cover는 요소의 가로세로 수치에 최대한 맞춰서 크기가 조절되고, 비율은 고정된다. 개체 크기에 맞게 잘리며, 가득 채울 때 까지 확대된다.

- 일반형제 선택자: A ~ B A와 B가 같은 계층에 있을 때 A 바로 뒤에 있는 모든 B를 선택자로 지정한다.

- 인접형제 선택자: A + B A와 B가 같은 계층에 있을 때 A 바로 뒤에 있는 B를 선택자로 지정한다.

- 박스 위아래 간격을 띄울 때, 원래는 mb or mt를 전체에 줬다가 마지막 요소에서 빼야했는데, .box ul li+li {mt10}을 주면 첫 번째 li를 제외하고 스타일이 적용되어 원하는 모양이 바로 나온다. 코드를 줄일 수 있어서 더 좋은 방법이다!!!

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

**### reset.css의 기본**

- (lisn) list-style: none (tdn) text-decoration: none

- \* {font: inherit; color: inherit;} 상속. 브라우저 기본값 상속받기

- a태그에 color: #000주지 않기. 나중에 상속을 받지 못하기 때문. 상속X? 값 변경되면 선택해서 값 다시 줘야함..

- img {max-width: 100%;} 최대 한계값.

- width와 max-width의 차이 (img)

* width: 100%: 요소를 감싸고 있는 부모의 크기에 맞춰 100%로 유지
* width값 지정X: 창 크기를 조절해도 이미지 크기에 영향X
* max-width: 100%: 요소의 기본크기를 초과하여 조절되지 않음. but, 그 기본 크기가 부모보다 큰 경우 부모의 크기로 재설정 됨.

- inline-block 여백 (img)

* 항상 영역 사이, 영역 아래에 여백이 약간 있다.
* 이를 없애기 위해서 db 또는 ★쌍으로 자주 쓰임★ dib & vat display: inline-block & vertical-align: top을 준다. 3주차 때 많이 쓰임!

- ★행간, 자간 조절하기 꼭 알아야!★

* line-height는 글씨 크기의 배수이다. line-height 브라우저 기본값은 16px. 즉, lh1의 크기는 16px.
* 행간: line-height, 자간: letter-spacing

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

**### test4.html**

- logo는 header가 기준이 아니다. header에 gnb도 포함되어 있는데 검은색인 box1 영역에서 중심이어야 하기 때문.

- gnb는 중요도가 header급이라서 앞에 다른 선택자 없이 gnb라고 바로 쓴다.

- a에 width, height를 주는 것이 h1에 주는 것보다 더 좋다!!

* h1은 위치, a는 크기를 맡아서 역할을 구분할 수 있다.
* a에 바로 크기를 주면 h1에는 크기를 줄 필요가 없어져서 코드를 줄일 수 있다.

- float 사용할 때 부모에 크기가 있다면, overflow: hidden을 주지 않아도 된다. 해제가 필요없다.

- nav는 block임.

- ★gnb 중앙정렬 꼭 알아야!★

* .gnb ul li는 이미 왼쪽 정렬이 되었기 때문에 중앙으로 못 간다. 이때는 부모인 .gnb ul을 이용한다.
* w과 m0-a를 줄 수도 있지만, 컨텐츠 양이 늘어나면 줄변경이 된다. 크기가 정해져 있어 유동성이 없다.
* poa도 w가 있어야 해서 위와 같은 이유로 사용 불가.
* dib를 주면 컨텐츠 크기만큼 크기가 잡힌다.
* inline-block은 글씨 취급. 따라서 부모에게 tac를 주면 중앙정렬 된다. 여백은 vat를 줘서 없앤다.
* flex를 사용한다면 (df)display: flex justicy-content: center를 주면 된다.

- box-sizing: content-box, box-sizing: border-box

* content-box: width는 content 크기다. border, padding이 content 바깥에 들어간다.
* border-box: width 안쪽으로 border, padding이 들어간다. 정확하게 등분을 나눠서 배치하고 그 안쪽으로 p와 b를 줄 때 주로 사용한다.

- 자식이 아무리 커도 부모의 크기를 벗어나 넘칠 수는 없다!!!

- 레이아웃 구성 시 가장 먼저 해야하는 것!

* 넌 왼쪽, 넌 오른쪽.. 이런 식으로 영역 선언을 해서 틀을 잡아둬야 한다. 각각의 li에 스타일 주면 안 된다.

- li로 틀을 만들고 a라는 실세가 있을 때는 a에 mr을 주고 ul에서 옆으로 너비를 확장했는데, li만 있으면 방법이 달라진다. 대부분의 경우에는 a가 있다.

-> w33.33p 말고, w c (계산식)를 써야 해서 번거롭다. 특정 태그에 mr을 없애고 mt을 주는 코드를 추가해야 해서 귀찮다.. (float 환경에서나 나올 수 있는 소스)

-> a에 mr를 주는 게 훨씬 더 좋은 소스이다.

-> 이 시안은 img 너비가 px이고, 특정 레이아웃을 만들기 위해 작성한 선택자 있어서, 값 변화에 자동으로 대응하지 못 한다.

- n+4: 4 이상, -n+3: 3 이하

- flex는 float와 달리 레이아웃만을 위해 만들어진 태그이다.

* flex는 stretch 속성으로 요소가 높이를 공유할 수 있다.
* float는 레이아웃을 위해 만들어진 게 아니라서 이런 기능이 없다.

- 앞서 배운 건 레이아웃을 효율적으로 만드는 최고의 방법.

- ★float 활용의 끝!★(고급기법) good 소스

* oh는 block에서만 적용된다.
* a는 inline이기 때문에 db를 준다. a를 보는 순간 바로 block으로 만들자!!
* p는 block이기 때문에 꽉 차있어서 w를 줄 필요없이 h만 주면 된다.
* p에 oh를 주면, img에 가려졌던 공간이 옆으로 빠져나온다. w를 따로 안 줘도 돼서 편하다!!
* p에 flr 주고, p와 img에 w c()로 값을 줘도 똑같이 되지 않느냐? but, 값 변화 시 p, img 둘다 스타일 변경해야 해서 번거롭다. 지금 방법은 img의 w만 변경하면 대응한다.

-> 변화에 한 쪽만 대응하는 소스가 좋은 것.

- overflow: hidden

* 요소의 컨텐츠가 너무 커서 지정된 영역에 맞지 않을 때, 넘치는 부분(overflow)을 자르고 숨긴다.
* 특정 높이가 지정된 block에서만 적용된다.

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

**### test5.html**

- main에게 w800, m0-a 주고 싶은데 중간에 w100p인 배너가 있어서 불가능.

-> box2를 제외한 나머지 main의 자식을 묶어서 공통 클래스를 할당한다. ex) inner에게 w를 줄 수 있다.

- ctl + shift + p -> ‘wrap’ 작성 -> ‘Emmet: 약어로 래핑’ 톱니바퀴 선택 -> 연필 누르기 -> 원하는 키 조합 만들기

- alt 누른 채로 원하는 영역 드래그 -> ctl + shift + w(내가 만든 키 조합) 누르기 -> 약어로 원하는 선택자 작성

- 하나하나 만들면 컨트롤이 힘들 때는 wrap으로 감싸준다.

- 나중에 common CSS라고 배울 것. .inner {w800 m0-a}를 줘서 header와 banner 빼고 중앙을 잡는다.

- float만 주면 a에서도 w h값을 줄 수 있다. float 들어가면 inline-block이 되니까!

- inline-block으로 바꿔주는 속성!

- poa position: absolute

- pof position: fixed

- float float: left, float: right

- df display: flex

- border: 양쪽에 bor1을 주면 사이는 2px이 된다. bor를 곂쳐서 1px로 만들기 위해 왼쪽 요소(prev)에 m-1을 준다.

- ~~오른쪽 요소(next)는 전체 위치를 잡아주고(지탱하고) 있으니 건들지 말자.~~ negative mb mr은 자신이 아닌 인접 요소를 움직이게 한다.

- 강조하려는 문장에는 strong, 일반 내용에는 p를 씀. inline을 block으로 만들려면 CSS에 db를 준다. 마크업은 최대한 깔끔하게 작성해야 한다!!!

- 말줄임

: Inline-block 또는 block일 때 작동(즉, w가 있을 때)

white-space: nowrap; 가로 한 줄로 만든다.

overflow: hidden; 자른다.

text-overflow: ellipsis; …을 만든다.

: 여러줄

overflow: hidden;

text-overflow: ellipsis;

display: -webkit-box;

-webkit-line-clamp: 2; (SASS에서 나온다. mixin. 정말 간편해서 요즘 SASS가 필수임)

-webkit-box-orient: vertical;

-----------------------------------------------------------------------------------------------------------------------------------**### 공부방법**

1. CSS를 분리한 마크업 완성파일 필요

2. 시안 보고 마크업 - 중간에 생각 안 나는 건 과감히 버리기

3. 완성파일과 비교해서 수정하고 오답노트 작성 - 스스로 피드백!!! 엄청 중요

4. 시안 보고 스타일 꾸미기

5. 완성파일과 비교해서 수정하고 오답노트 작성

6. 오답노트와 시안만 보고 코딩

7. 타임아웃. (12분 내로 완성해야 완벽히 이해한 것.) - 0에서 끝까지 스스로 만들 수 있으면 그제야 도전!

- 단축키 최대한 활용하기

- 할당량 채우는 게 최우선이고 남는 시간 생기면 정리하기.

- notion, velog, git에다 정리하기

* 제일 중요한 건 velog 관리 - 1주차부터 시작해야!
* ★공부 흔적★을 남겨야 한다.
* 배운 거, 따로 궁금한 거, 프로젝트 진행한 거… 다른분들 거 참조하기
* 남들이 써놓은 거 내 입맛대로 조합해서 써두면 된다.

- git은 세가지만 알면 된다.

* pull : 소스 최신화
* push : 밀어넣기 업로드 서버에
* commit : 코멘트 달기 업로드X
* 소스 제어 -> Guthub에 게시