**2주차(float, grid, SCSS)**

- html은 scss는 쓸 수 없다. scss는 css를 간편하게 사용하기 위한 도구일뿐이다.

- 나중에 css가 업그레이드가 되면, css도 scss처럼 쓰는 문법도 있다.

- map 파일 필요없이 #style.css 파일만 필요하면, 설치한 ‘Live Sass Compiler’의 확장설정 -> Generate Map의 편집에 들어가서 저장을 한 뒤, 생성된 savePath: null을 savePath: ~/../css/로 바꿔준다.

- 어떻게 컴파일 했어요? 물어보면, VS코드에 있는 확장도구로 컴파일 했습니다. 라고 말하면 됨. 이외에도 컴파일 방법이 다양함.

- less, sass, scss, stylelus 등 다양한 도구가 있으며, 그중 scss가 가장 유명하다.

- 유지보수를 편하게 하기 위해 파일을 폴더별로 분할한다.

- 포트폴리오는 Sass를 써서 만들거라서 올려놓으신 링크 3개 꼭 읽어봐야함!!

- 파일명에 ‘\_’를 쓰면 컴파일이 안 된다.

- 컴파일은 style.scss만 된다! 이 파일 안에 모든 파일을 @import해서 한꺼번에 컴파일 하는 것임.

- style.scss는 반복해서 만들 필요없이 처음 한 번만 셋팅해두면 된다. 새로운 게 생기면 추가만 하면 된다.

- font는 구글폰트에서 가져오기(@import로). typo에 @import 붙여넣고, var에 폰트 이름 붙여넣는다.

- var에서 font-family: ‘~’, ~;를 $font-kr, $font-en 이런식으로 변수화 할 수 있다(변수 세팅). 이름 쓰고 싶을 때 간단히 변수만 쓰면 된다!!

- font-family는 국문이 기반이 되어야 한다. 영문이 기반이 되면, 국문 안에 있는 영문도 영문이 되어버린다…!

- 그래서, 국문을 기반으로 하고 필요한 부분에 가서 선택적으로 영문을 줄 것임!

- @mixin에 재활용 할 소스를 적는다. 원하는 선택자 내의 맨 위에 “@include 믹스인 이름”으로 @mixin을 호출한다.

- vertical: 수직, 세로 / horizontal: 수평, 가로

- 클래스를 붙여서 쓰고 싶으면(.btn.prev 처럼) scss에서 클래스 앞에 &를 붙인다.

- 클래스 bullet bullet-active가 있을 때, scss의 .bullet 안에다 &-active를 줘서 bullet-active를 선택할 수 있다.

- .box2, .box3은 중앙을 잡아두면 좋으니까 .inner로 묶는다. - 두 부분을 반응형 처리하기 위함이다.

- .inner는 공통 클래스라서 common.scss에다가 쓴다.

- max-width / min-width - 반응형 처리! 가능해짐

- 부모에게 gtr 안 주고, 자식에게 h를 줘서 레이아웃을 만드는 것도 한 방법임!

- -webkit-box webkit은 엔진임. webkit으로 만든 애들은 webkit 소스를 붙여줘야 한다.

- @mixin ellipsis($line:n){ 몇 줄 쓸 건지 변수에 담는 것임.

@if($line== 1){ 조건이 참이면 @if 중괄호를 실행, 거짓이면 @else 중괄호를 실행

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

@else{

overflow: hidden;

text-overflow: ellipsis;

display:-webkit-box;

-webkit-line-clamp: $line;

-webkit-box-orient: vertical;

}

}

- 한줄만 쓰고 싶으면? main.scss에 @include ellipsis;, mixin.scss에 디폴트값 $line:1을 준다. 굳이 1을 안 써도 됨.

- 처음 만들 때만 오래 걸리지, 나중에 재사용 할 수 있기 때문에 코드 짜기가 상당히 쉬워진다.

/\*반응형 화면 크기\*/

$mobile: 767px; // 767 ~ **[320]** 최대 ~ 최소

$tablet: 1023px; // 1023 ~ **[768]**

$desktop: ???; // ??? ~ **[1024]** <- **프로젝트마다 다르다!!** 요소가 공존이 안 되는, 너무 타이트한 부분이 생기면 한 번 잡아주는 것.

/\*반응형, 브라우저 크기가 767px 이하일때\*/

@mixin mobile{

@media (max-width: $mobile){

@content;

}

}

/\*반응형, 브라우저 크기가 768이상, 1023px 이하일때\*/

@mixin tablet{

@media (max-width: $tablet){

@content;

}

}

/\*반응형, 브라우저 크기가 1024px 이상일때\*/

@mixin desktop{

@media (max-width: $desktop){

@content;

}

}

- 나중에 css로 반응형 만드는 프로젝트가 나올 것!(노다가 해야 됨) 그때 되면 아 Sass가 엄청 편했구나..를 알게 될 것임

- 개발자 모드: ctl + shift + I

- mixin.scss의 @content 덕분에 따로 써주지 않아도 style.css에 max-width: 1023px값이 들어간다.

<mixin.scss>

$tablet: 1023px;

@mixin tablet{

@media (max-width: $tablet){

@content;

}

}

<main.scss>

.bottom{

@include tablet{

}

}

<style.css>

.box2 .bottom{

@media (max-width: 1023px){

}

}

- SCSS와 CSS 반응형 할 때 차이점.

**- Sass는 공부를 꾸준히 스스로 하면 더 좋다!! 올려주신 Sass 링크에 모든 문법이 다 써져 있음! 보면 엄청 도움될 것.**

**-------------------> 여기까지가 기본 지식. 포폴, 반응형 프로젝트 할 때 좀 더 자세히 나올 것임!!**