

## 2021 / SASS(SCSS)

- 전처리기(preprocessor), 자료형, \$변수, @mixin/@include, @function/@return, @extend

!Important Honey Tips

※ Node16 = sass@6

※ CSS(=작업이 고도화할수록 불편함), 전처리기 사용(Less, Sass(SCSS), Stylus)

Sass(SCSS)

CSS확장언어, 다른 전처리기보다 먼저 나옴. 성숙도 높고 커뮤니티에서 문제해결.

Sass =(CSS preprocessor, 전처리기를 사용해 컴파일러로)=> CSS로 전환

Node.js로 설치 npm sass -v로 버전확인	터미널	npm init 후 버전에 맞는 npm install node-sass@버전 node16이상 => npm install node-sass@6
	버전 오류	npm uninstall node-sass npm uninstall -g node-sass

※ 주석 Comment:  
(js와 같음)

//	컴파일 안됨
/* */	컴파일 됨, (sass는 각 줄 앞에 *를 붙이고 라인 맞춰야 함)

※ 데이터종류,형:

Numbers	단위 다르면 calc, 나누기 단독x: \$변수/2, (100/2), 10+90/2
Strings	첫 번째 피연산자 “”에 따라 따옴표 처리, +사용
Color	rgb도 연산 가능 / 알파값은=> opacity(); transparentize();
Booleans	and(&&), or(  ), not(!). @if not (조건문){ ~~ }
Nulls	속성값이 null이면 컴파일x
Lists	coma나 빈칸으로 목록
Maps()	()필수

※ 중첩, 네스팅 Nesting: ul{ display: block; li { a { ~~; } a:hover{ ~~; } } } 등지등지

※ 부모참조 Ampersand: li { &:nth-child(2n){ ~~; } &.active { ~~; } }

※ #{&변수}; 코드 어디서든 변수 값 사용

변수 선언 variable assignment	\$변수명: 속성값;	\$color-main: #ff00ff; .box { color: \$color-main; }	변수 앞에 항상 \$붙임 스코프 지역 내에서 사용
변수 재할당 Reassignment	\$변수1: \$변수2	\$color-main: \$ocean;	변수에 변수 지정
변수 전역설정 !global flag	\$변수명: 속성값 !global;	\$ocean: #2196f3 !global;	전역변수 설정(기본)
변수 초기값 !default flag	\$변수명: 속성값 !default;	\$ocean: #2196f3 !default;	할당되지 않은 변수의 초기값 외부라이브러리 변수 중복 덮어쓰기

문법		
opacity(color, alpha); transparentize();	color: opacity(\$color, .3)	30% 불투명 증가 // 0.5 + 0.3 = 0.8
	color: transparentize(\$color, .3)	20% 투명 증가 // 0.5 - 0.2 = 0.3
@mixin 변수명(\$매개, \$매개) @include 변수명(\$매개, \$매개)  매개변수 Parameters 전달인수 Arguments	@mixin size(\$w, \$h){ width: \$w; height: \$h; }	인수에 기본값 default value설정 (인수전달 없으면 기본값 사용)
	@mixin line(\$width: 1px, \$color: #000) .box { @include size(100px, 100px) }	가변인수 variable arguments @mixin 믹스인이름(\$매개변수 ... ) 개수 지정x
@function 함수명(인수) { @return 값 }	@function plus(\$A, \$B) { @return \$A + \$B }	함수, return값을 반환, 스코프 필수
@extend 선택자	.btn { @extend .box; color: #ddd }	선택자의 스타일을 불러와서 확장

# 2021 / SASS(SCSS)

## - 문법, 내장함수

문법		
@if, @else if, @else	괄호없이 사용, 스코프{}	@if \$num>10 and \$num<20 { ~~ } @else { ~~ }
@if(조건, 표현식1, 표현식2)	3항 conditional ternary operator	
@for // through, to	nth-child()에는 through사용, from(시작) ~ to / through(종료)	@for \$i from 1 through 3 { .box:nth-child(#{ \$i }) { width: 20px * \$i }}
@each	(=for in) list와 map데이터를 반복	@each \$변수 in 데이터 { // 반복내용 }
@while	@while \$i > 0 { .item-#{ \$i } { width: 10px * \$i }, \$i : \$i - 1; }	
@import // @export	@import 'font.scss'(모듈화 sass 불리와 css로 컴파일, .scss확장자는 생략가능) css @import와 달리 http호출 필요x 속도저하x, _mixin.scss, _header.scss	

내장함수(Built-in Functions)		
Color: 색상  RGB HSL Opacity	mix(\$color1, \$color2) : 두 개의 색 섞음 lighten(\$color, \$amount) : 밝게 darken(\$color, \$amount) : 어둡게 saturate(\$color, \$amount) : 채도 desaturate(\$color, \$amount) : 채도 낮추기 grayscale(\$color) : 무채색	invert(\$color) : 색반전 rgba(\$color, \$alpha) : 투명도 opacify(\$color, \$amount) : 더 불투명하게 transparentize(\$color, \$amount) : 더 투명하게 fade-in(\$color, \$amount) : 더 불투명하게 fade-out(\$color, \$amount) : 더 투명하게
String: 문자	unquote(\$string) : 따옴표 제거 quote(\$string) : 따옴표 추가 str-insert(\$string, \$insert, \$index) : 문자의 index째에 특정문자를 삽입 str-index(\$string, \$substring) : 문자에서 특정문자의 첫 index 반환	str-slice(\$string, \$start-at, [\$end-at]) : 특정 문자를 start부터 end까지 추출 (세 번째 인수가 없으면 마지막까지 추출) to-upper-case(\$string) : 문자를 대문자로 변환 to-lower-case(\$string) : 문자를 소문자로 변환
Number: 숫자	percentage(\$number) : 백분율로 변환(단위무시) round(\$number) : 정수로 반올림 ceil(\$number) : 정수로 올림 floor(\$number) : 정수로 내림	abs(\$number) : 숫자 절대 값, 실수 min(\$numbers...) : 숫자 중 최소값 반환 max(\$numbers...) : 숫자 중 최대값 반환 random() : 0부터 1 사이의 난수 반환
List: List 갱신x	length(\$list) : List의 length반환 nth(\$list, \$n) : List에서 n번째 값 반환 set-nth(\$list, \$n, \$value) : List n번째 값 수정	join(\$list1, \$list2, [\$separator]) : List를 결합 zip(\$lists...) : List를 다차원 List로 결합 index(\$list, \$value) : List에서 특정 값 index 반환
Map: Map 갱신x	map-get(\$map, \$key) : Map에서 특정 key의 value를 반환 map-merge(\$map1, \$map2) : Map을 병합해 새로운 Map 만들기 map-keys(\$map) : Map에서 모든 key를 List로 반환 map-values(\$map) : Map에서 모든 value를 List로 반환	
Introspection: 관리	variable-exists(name) : 변수가 현재 범위에 존재하는지 여부를 반환(인수는 \$없이 변수의 이름만 사용) unit(\$number) : 숫자의 단위를 반환 unitless(\$number) : 숫자에 단위가 있는지 여부를 반환 comparable(\$number1, \$number2) : 두 개의 숫자가 연산 가능한지 여부를 반환	