<공부방법>

1. 결과물을 보고 일단 따라 만들어보려고 시도해본다.
2. 막히는 부분이 생기면 완성된 코드를 본다.
3. 코드를 봐도 알 수 없는 부분들은 강의를 다시 본다.
4. 강의를 봐도 알 수 없는 부분들은 구글에 검색해서 찾는다.
5. 알게 된 것들을 여기 메모해 둔다(그 상황만을 해결하는 방법보다는 일반화된 해결 방법론을 메모하는 것이 좋다). 잘못 메모해둔 부분은 그때그때 고친다.
6. 결과물을 다 따라 만들었다면, 막히는 부분이 없을 때까지 1~5를 반복한다(이때 메모해둔 것들을 참고해도 된다).
7. 나중에 공부가 부족하거나 빠진 부분이 있다고 판단되면, 결과물의 일부분만 따라해본 것은 아닌지 체크해본다. 만약 그렇다면 결과물에 다른 부분의 코드를 쭉 읽어보면서 모르는 부분을 체크하고 찾아본다.
8. 그래도 공부가 부족하면 모든 부분을 따라 만든다(1~6).

<코드를 깔끔하게 정리하는 법>

* css파일들은 css폴더에 묶어서 보관. 이때 html과 연결되는 css파일은 styles.css로 통일한 후, 그 안에서 components, screens, 기타의 영역으로 구분지어 import해야 함. 이를 위해 css파일 내부에 component, screens 파일을 각각 생성해야 함. Components 영역에는 여러 화면에 걸쳐 반복적으로 등장하는 구성 요소들에 대한 디자인 정보를 담고, screens 영역에는 한 화면 단위의 디자인 정보들을 담음. (reset.css도 잊지 말기!!!)

Ex) ->styles.css

@import “reset.css”;

@import "variables.css";

(…)

/\*components\*/

@import "components/status-bar.css";

(…)

/\*screens\*/

@import "screens/login.css";

(…)

* -와 \_\_의 차이 : -는 긴 이름 지을 때 스페이스 바 대신 사용. \_\_는 특정 요소의 하위 요소로 들어감을 의미함.

Ex) status-bar, status-bar\_\_icon

* 공통되는 속성이 있을 수 있음을 고려하여 같은 클래스명을 부여하자.
* <div *class*="status-bar">
* <div *class*="status-bar\_\_left">
* </div>
* <div *class*="status-bar\_\_middle">
* </div>
* <div *class*="status-bar\_\_right">
* </div>
* </div>

여기서 left, middle, right로 따로 이름을 부여하면 공통적인 속성을 일일이 부여해야 하므로 비효율적이다.

* <div *class*="status-bar">
* <div *class*="status-bar\_\_column">
* </div>
* <div *class*="status-bar\_\_column ">
* </div>
* <div *class*="status-bar\_\_column ">
* </div>
* </div>

이렇게 통일한 후 first child, nth child같은 걸 이용해서 추가적인 부분들을 지정하도록 하자.

* 전체 설정 : body처럼 가장 부모 태그에 설정해주면 됨. 그렇게 해도 하위 요소들까지 다 적용됨. 그렇게 하면 하위 요소들에 일일이 속성을 부여하지 않아도 됨. 하위 요소에서 설정을 바꾸고 싶다면 그 요소를 클래스로 지정해서 속성을 바꿔주면 바뀜.

Ex)

Body{

    color : var(--kokoa-dark-gery); */\*\*/*

}

<html-css 연결하는법>

Head 태그 내부에

<link rel="stylesheet" href="css/styles.css">

(css/styles.css는 html에 연결할 css파일의 주소와 이름. 기본 위치는 html파일이 존재하는 폴더임. (폴더->css->styles.css로 들어간다는 의미))

<reset.css>

Css에 여러 태그들에는 기본 margin이 설정되어 있는데, 이걸 처음에 다 제거해 주어야 한다. 이때 reset.css라는 개념을 사용한다. 구글에 reset css라고 쳐서 파일 내용을 찾은 후, css폴더 내부에 reset.css파일을 만들어 거기에 복붙하면 된다. Styles.css에 넣어주자.

https://cssdeck.com/blog/scripts/eric-meyer-reset-css/

<아이콘 넣는 법>

Fontawesome 사이트 들어가서 이메일 등록하고, code kit(자바스크립트 파일임)를 받아서 html파일의 body맨 밑에 복붙해서 집어넣은 다음 찾고싶은 아이콘 검색해서 코드 복사->붙여넣기하면 됨.

* 크기조절(ex)

<i class="fa fa-camera fa-lg"></i> <!-- 33% 크기 -->

<i class="fa fa-camera fa-2x"></i> <!-- 2배 크기 -->

<i class="fa fa-camera fa-3x"></i> <!-- 3배 크기 -->

<i class="fa fa-camera fa-4x"></i> <!-- 4배 크기 -->

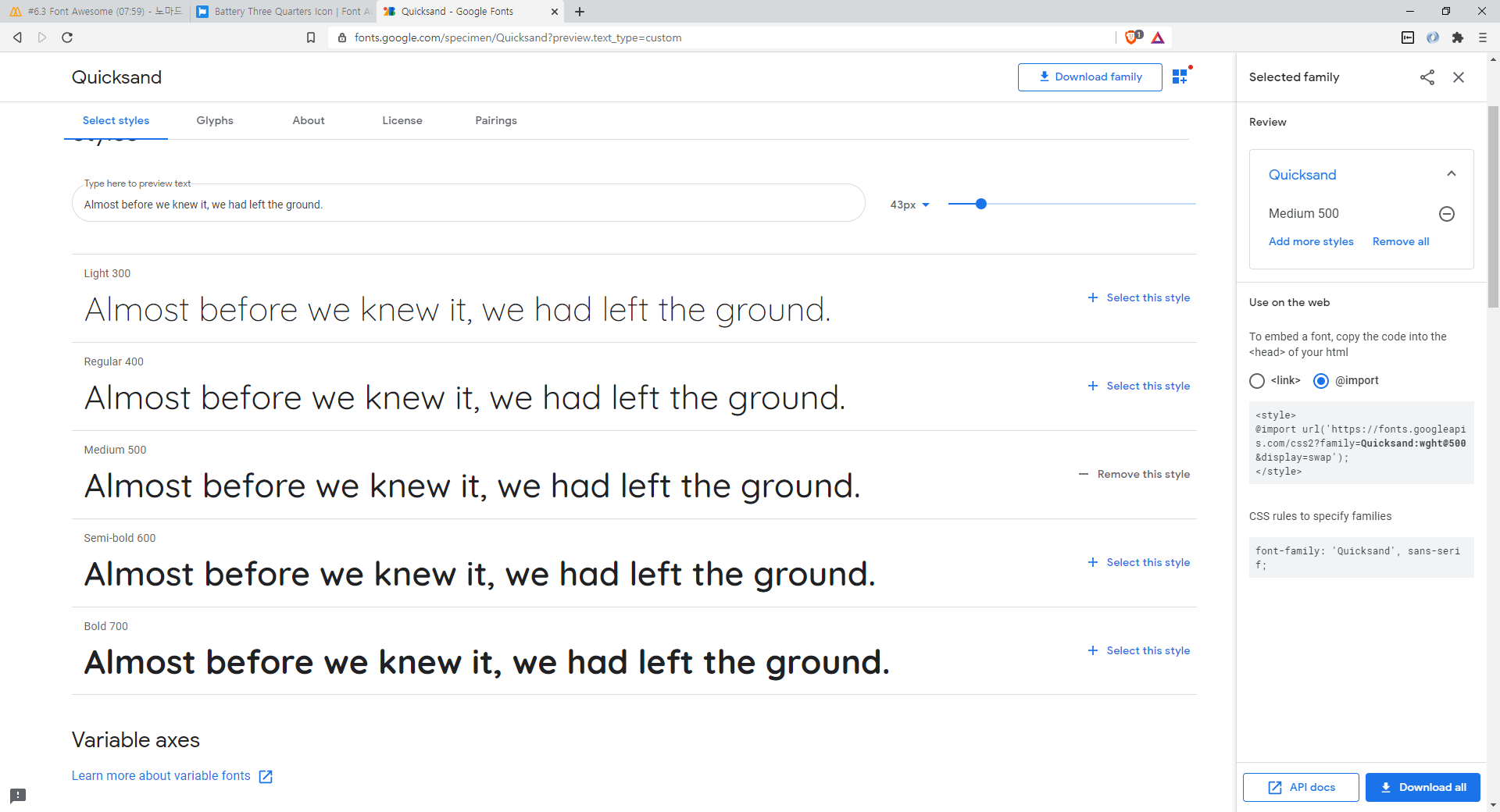
<i class="fa fa-camera fa-5x"></i> <!-- 5배 크기 -->

이것보다 더 세밀하게 하고 싶다면 font-size : ~~~px로 조절하면 됨.

(아이콘이랑 글씨랑 거의 비슷한 것 같음)

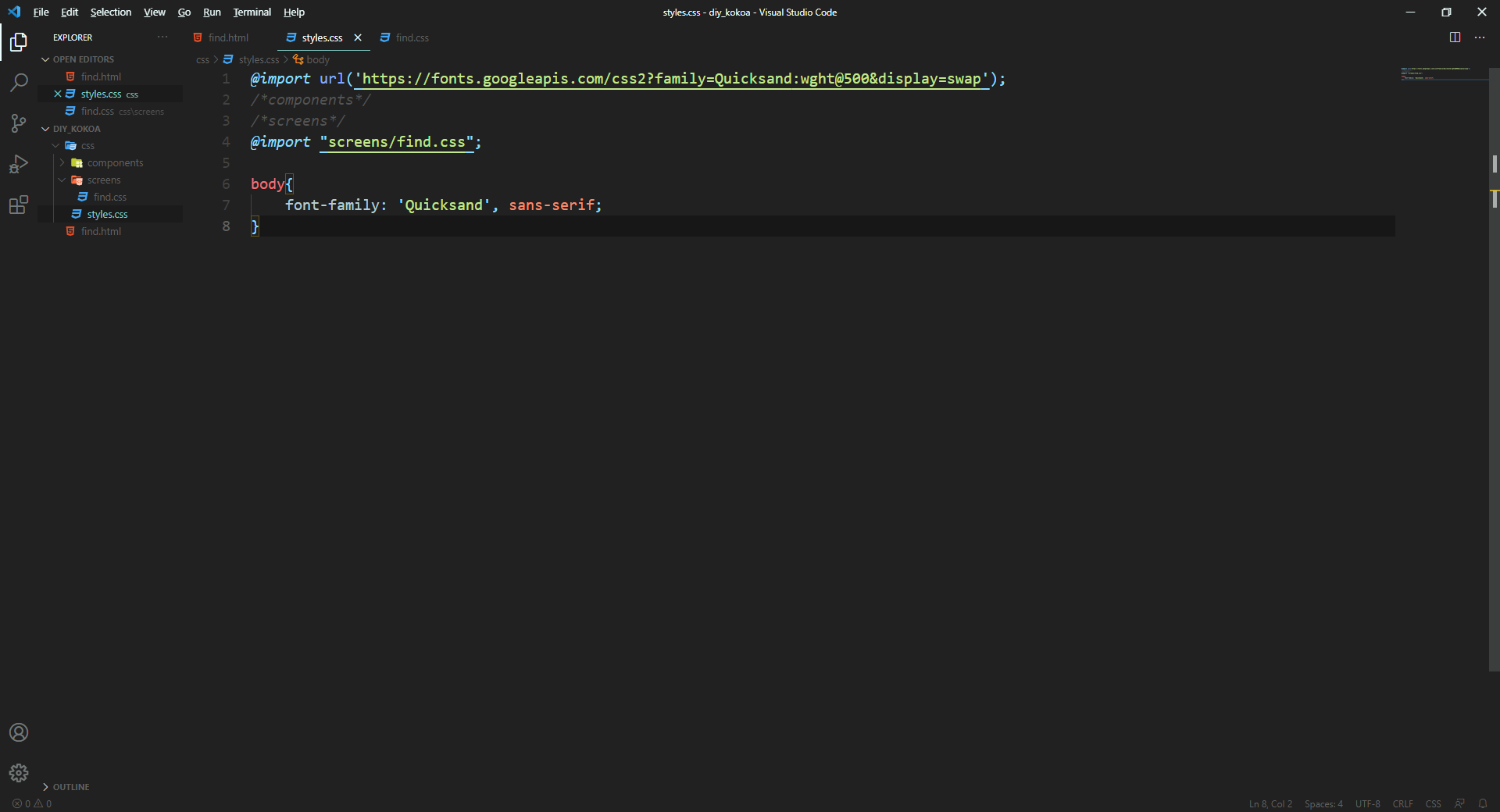
<글꼴 설정하는 법>

구글에 google fonts라고 입력해서 사이트 들어간 다음 사용하고 싶은 글꼴 클릭->



Weight 골라서 select this style 누르고 <link> 대신 @import 누르고 css(전체 글꼴이라면 styles.css이런식)파일 위쪽에 <style>태그 내부에 있는 @import~~~부분 복붙, 글꼴 적용하고 싶은 곳에 font-family~~이 부분 복붙.

Ex)



<다음 그림과 같이 배치하기(status-bar)(css hack)>

    <div *class*="status-bar">

        <div *class*="status-bar\_\_column">

            <span>No Service</span>

            <i *class*="fas fa-wifi"></i>

        </div>

        <div *class*="status-bar\_\_column">

            <span>18:43</span>

        </div>

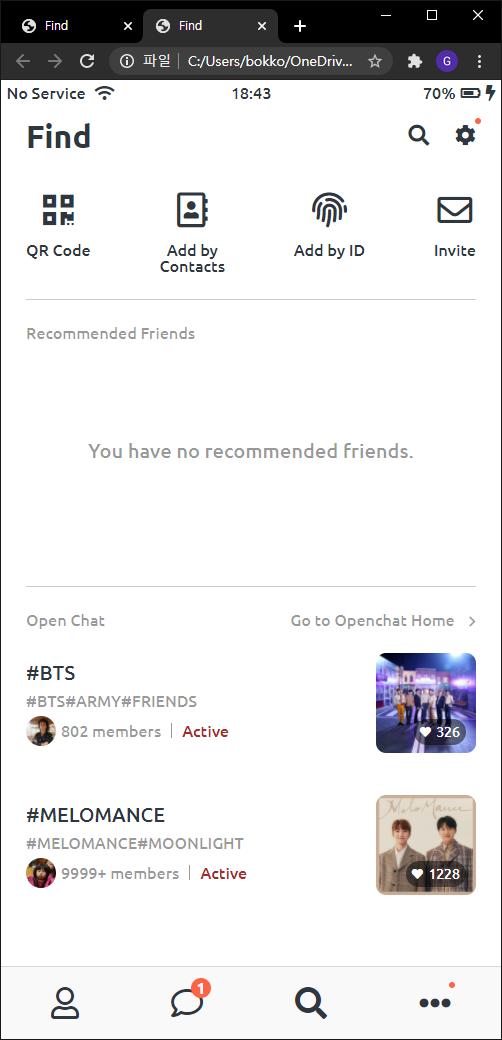
        <div *class*="status-bar\_\_column">

            <span>70%</span>

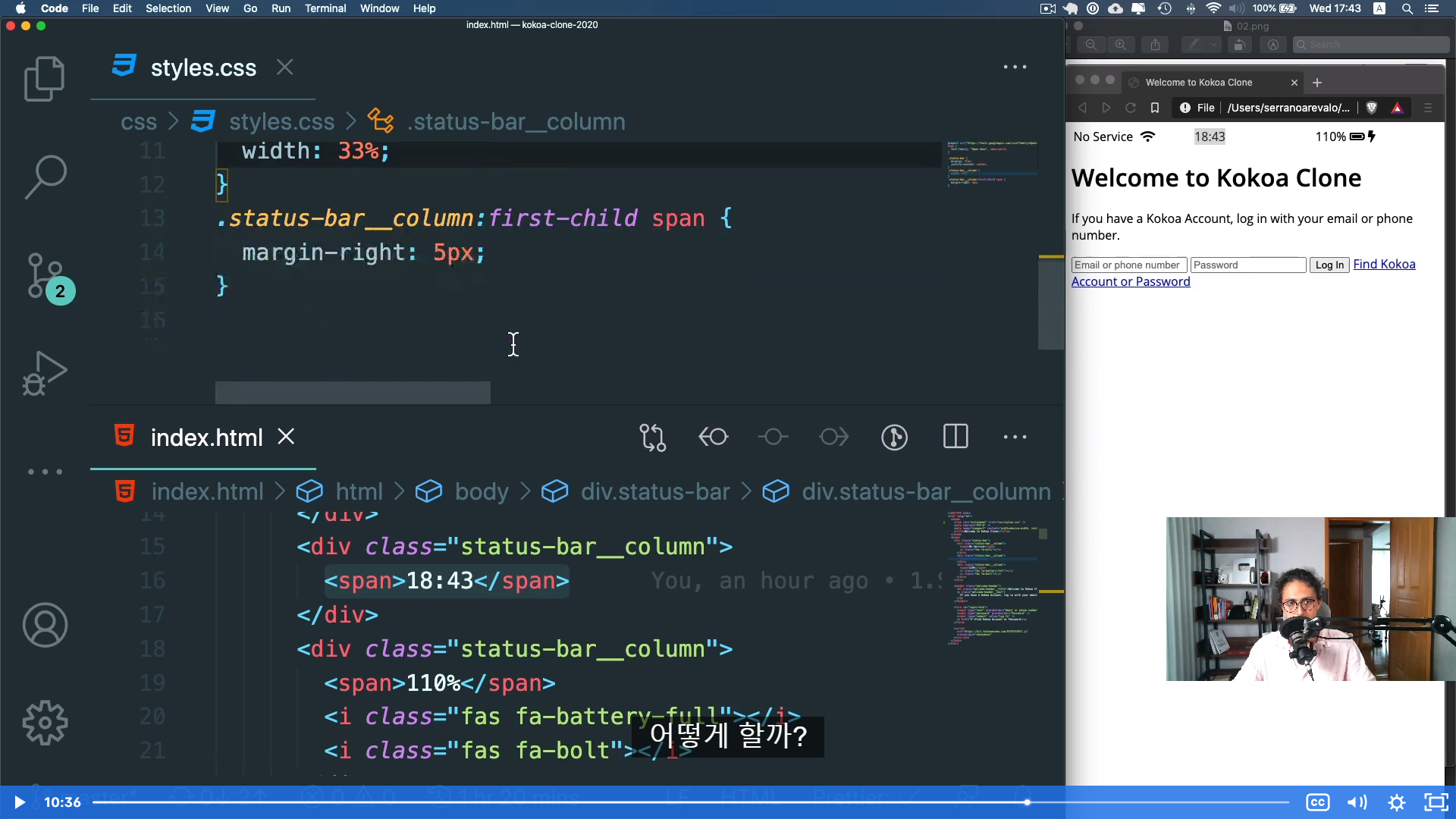
            <i *class*="fas fa-battery-three quarters"></i>

        </div>

    </div>



1. column들의 세로 배치를 가로로 바꿔야 한다. Display :flex; (배치 기준, 즉 배치 ‘축’이 세로에서 가로로 바뀜.)
2. column들을 축을 기준으로 가운데로 모은다. (justify-content: center;)
3. column들에게 33%의 width를 부여한다. (width: 33%;) 그럼 요 상태가 된다.



1. 왼쪽 column은 건들 필요가 없다.
2. 가운데 column 내부의 span을 가운데로 이동시킨다. 그러려면 축을 가로로 바꾼 다음, 축을 기준으로 내부 요소를 가운데로 옮겨야 한다. (display: flex; justify-content: center;)
3. 오른쪽 column 역시 축을 가로로 바꾸고, 축을 기준으로 내부 요소를 맨 끝으로 옮긴다. (display: flex; justify-content: flex-end;)
4. Column 내부 요소들의 높이가 달라졌다면, 축의 수직 방향 기준 가운데로 모은다. (align-items: center;)
5. Column 내부 요소들간의 간격을 조절하면 끝. (margin-right: 5px;)

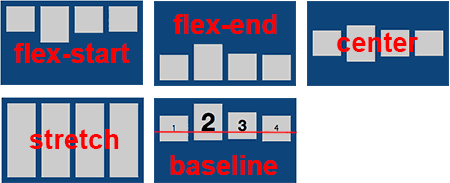
<justify-content>

Div의 축을 기준으로 내부 요소들을 정렬한다. Default는 세로축이며, Display :flex;를 하면 가로축으로 바뀐다. (다음 그림은 display flex를 한 상태이다.)



<align-items>

Div 축의 수직 방향을 기준으로 내부 요소들을 정렬한다. Default는 세로축이며, Display :flex;를 하면 가로축으로 바뀐다. (다음 그림은 display flex를 한 상태이다.)



<css-클래스 내부의 클래스, 태그 등을 지정하는 방법>

* 그냥 외부 클래스 옆에 내부 클래스명, 태그명을 써서 지정하면 된다.

Ex) .status-bar .status-bar\_\_column span{

}이런식.

<단축키>

div.upperclass>div.lowerclass\*4>a 치고 엔터누르면

<div class="upperclass">

<div class="lowerclass"><a href=""></a></div>

<div class="lowerclass"><a href=""></a></div>

<div class="lowerclass"><a href=""></a></div>

<div class="lowerclass"><a href=""></a></div>

</div>

이렇게 자동완성됨.

>: 내부로 들어간다는 의미.

. : 클래스명을 지어준다는 의미.

<전역 css 변수 사용하는 방법>

1. Css 폴더 내부에 variables.css파일을 생성한다. (styles.css에 import하고)
2. variables.css파일 내부에 다음과 같이 입력하여 전역 변수를 선언한다.

:root{

--전역변수이름 : 전역변수내용; ex) --horizontal-space : 25px;

}

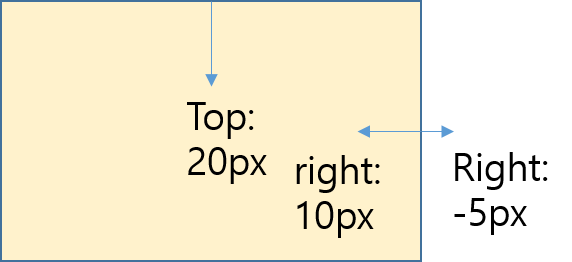
1. 다음과 같이 변수를 사용한다.

Var(전역변수이름)

ex) 25px 대신 var(--horizontal-space)

<relative, absolute, top | bottom | left | right>

* Position :relative -> 원래 있는 위치를 기준으로, top | bottom | left | right를 조정한 만큼 이동함.
* position: absolute -> 가장 가까운 Position :relative 인 div를 기준으로, top | bottom | left | right를 조정한 만큼 이동함. Absolute를 사용하려면 부모 div가 position : relative가 되어야 함.



<flex 축 방향 바꾸기>

display: flex;

flex-direction: column;

이렇게 하면 바뀐다. i – span 처럼 기본적으로 가로로 배열되어 있는 애들의 방향을 세로로 강제할 때 쓴다.

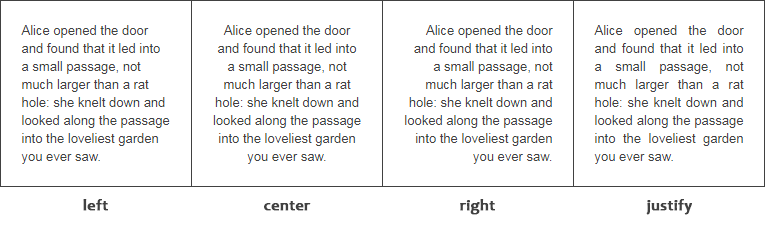
<테두리(border)설정하는 법>

Border(-top,bottom,left,right) : border두께 border스타일 border색깔;

Ex) border-bottom: 1px solid var(--kokoa-light-gery);

<글씨 정렬 조정>

Text-align으로 조정함. Ex) text-align: center;



<inline / block>

- box는 옆에 다른 box가 올 수 없다.

- 다른 요소가 옆에 올 수 없는 걸 'block', 다른 요소가 올 수 있는 걸 'inline'이라고 한다.

- 고로 box 대신에 block으로 부른다.

- 대부분의 요소는 block인데, inline인 것은 몇 없다. (span, a, image 등)

Inline과 block을 서로 바꿀 수 있다. (display속성이라고 함)

(display: block; 이나 display: inline; 이렇게 바꾼다.)

어떤 요소가 inline 이면, 높이와 너비를 가질 수 없다.

Block은 높이와 너비가 있다.

block의 특징 margin, padding, border

(block만이 margin, padding을 가질 수 있다.)

<이미지 html에 넣는 법>

외부에서 가져올 경우 : 사이트에서 이미지 주소를 복사한 후 img src에 넣으면 됨.

이미지 파일이 있을 경우 : images 폴더를 따로 생성한 다음 이미지파일을 그 안에 집어넣고, 이미지 주소를 img src에 넣으면 됨.

<divider>

글씨와 글씨를 구분하는 칸 같은 것. | 처럼 생겼고, Div로 한땀한땀 만들면 된다.

<a에 기본 부여된 파랑,보라 색상 없애는법>

a 태그지정하고 color : inherit; 해주면 된다.

<a(anchor)가 면적을 가지는 방법>

<a>태그 안에 i나 span, div같은 눈에 보이는 무언가를 집어넣어주면 된다.

<inherit>

CSS 상속 (inherit)

자식 요소에서 특별히 속성을 지정하지 않은 경우, 부모 요소의 속성을 물려받는 걸 의미. 자식 요소에 별도로 속성 지정 안해도 상속 되지만, 자식 요소 속성에 inherit 적어 상속을 명시화 해줄 수 있음. Ex)

a{

color: inherit;

}

<cursor>

마우스(커서)를 갖다 대면 마우스의 모양이 바뀐다. Pointer 등등 여러가지가 있다.

Ex) .button{

cursor:pointer;

}

<form과 input>

https://developer.mozilla.org/ko/docs/Web/HTML/Element/form

https://developer.mozilla.org/ko/docs/Web/HTML/Element/input

사용자로부터 입력을 받는 창. 다양한 속성을 가지고 있다. 아직 왜인지는 모르지만 input은 꼭 form 안에 넣어주자. => input은 결국 사용자로부터 입력을 받기 위해 존재하고, 그러기 위해서는 submit을 꼭 해야 하는데, form이 없으면 submit이 불가능하다!!

Ex)

<form class="name-ask">

<input type="text" placeholder="What is your name?">

</form>

속성 : (여기 없는 것들은 공식 문서나 인터넷을 참고하자)

* type -> 가장 중요한 속성. Type에 따라 다양한 형식을 사용자로부터 입력받을 수 있다. (ex) color, file, text, password, submit,…(Submit -> submit 버튼을 누르면 form을 어디론가 보내 줌.)
* placeholder -> 입력하기 전에 써있는 글 내용. 커서를 올리거나 무언가 입력하면 없어진다. 주로 어떤 내용을 써야 하는지 안내하는 역할을 한다.
* Min, maxlength -> submit을 하기 위한 최소, 최대 문자열 길이.
* Required -> submit을 하려면 required 속성이 있는 input은 무조건 텍스트가 있어야 함.