대학 주도형 아카데미 K-디지털 트레이닝 AI 데이터분석 풀스택 웹 개발자 양성과정

(Cascading Style Sheets) 김경민 pnumin@pusan.ac.kr









### CSS 적용

- HTML
   CSS
- JavaScript
- React facebook
- instagram

### 프론트엔드 FrontEnd

웹에서 사용자에게 시각적으로 보여지는 부분을 의미한다. 프론트엔드쪽에서는 주로 HTML, CSS, 자바스크립트가 쓰이며, 요즘은 페이스북에서 개발한 자바스크립트에서 파생된 싱글 페이지 애플리케이션 형태의 리엑트가 많이 사용된다.

#### HTLM



HTML(HyperText Markup Language)은 웹을 이루는 가장 기초적인 구성 요소로, 웹 콘텐츠의 의미와 구조를 정의할 때 사용

#### CSS



Cascading Style Sheets(CSS)는 HTML이나 XML(XML의 방언인 SVG, XHTML 포함)로 작성된 문서의 표시 방법을 기술하기 위한 스타일 시트 언어

### JavaScript



웹 페이지를 위한 스크립트 언어로 가벼운, 인터프리터 혹은 just-in-time 컴파일 프로그래밍 언어로, 일급 함수를 지원

### React



사용자 인터페이스를 만들기 위한 JavaScript 라이브러리

- Al 빅데이터본석 톱스틱서비스 5W 개발자 양성과정
   원개발, 자바/자료구교/Al 프로그램링, DS SOL, HOIOIE석, 텐서돔프립러닝, Al학습모델엔서비스프로젝트, 빅데이터분석엔서비스프로젝트
   부산대학교 과학기실연구종, 4명 401후 juris765@pursan.ac.kr 051-510-3764













### 프론트엔드 (FrontEnd)

웹에서 사용자에게 시각적으로 보여지는 부분을 의미한다. 프론트엔드쪽에서는 주로 HTML, CSS, 자바스크립트가 쓰이며, 요즘은 페이스북에서 개 발한 자바스크립트에서 파생된 싱글 페이지 애플리케이션 형태의 리액트가 많이 사용된다.

#### HTLM



HTML(HyperText Markup Language)은 웹을 이루는 가장 기초적인 구성 요소로, 웹 콘텐츠의 의미와 구조를 정의할 때

#### CSS



Cascading Style Sheets(CSS)는 HTML이나 XML(XML의 방언인 SVG, XHTML 포함)로 작성된 문서의 표시 방법을 기술하기 위한 스타일 시트 언어

#### JavaScript



웹 페이지를 위한 스크립트 언어로 가벼운, 인터프리터 혹은 just-in-time 컴파일 프로그래밍 언어로, 일급 함수를 지원

#### React



사용자 인터페이스를 만들기 위한 JavaScript 라이브러리









React JS

# **CSS(Cascading Style Sheets)**

- · HTML로 작성된 문서의 표시 방법을 기술하기 위한 스타일 시트
  - · 웹 페이지의 디자인, 레이아웃, 그리고 다양한 장치 및 화면 크기에 맞춘 변형을 정의하는 데 사용규칙 기반 언어
  - · 웹 페이지의 시각적 표현을 HTML 구조와 분리하여 관리
- ㆍ규칙 기반 언어
  - · 웹 페이지의 특정 요소 또는 요소 그룹에 적용할 스타일 그룹을 지정하는 규칙을 정의
  - · 규칙은 선택자(selector)와 선언(declaration)으로 구성
  - · 선택자는 어떤 요소에 스타일을 적용할지 결정하고, 선언은 적용할 스타일 속성 (property)과 그 값을 정의









## CSS 구성

### 선택자(selector)

- 스타일을 고칠 HTML 요소
- 고칠 스타일은 여러 개의 선언으로 지정가능
- 반드시 {}로 묶어야 함

### 선언(declaration)

- 콜론(:) 끝남으로 구분 된 CSS 속성 이름과 값이 포함
- 세미콜론(;)으로

```
선택자(Selector)
                    선언 (Declaration)
    section
        background-color : yellow
                          값 (Value)
        속성 (Property)
```

HTML









### HTML문서에 CSS3 스타일 시트 만들기

- · HTML 문서에 CSS3 스타일 시트 만드는 방법
  - · 인라인 스타일(Inline style) 유선순위 1

- · 태그의 style 속성으로 사용
- · 내부 스타일 시트(Internal style sheet) 우선순위 2

- · <style>·· </style>
- · 외부 스타일 시트(External style sheet) 우선순위 3









## HTML문서에 CSS3 스타일 시트 만들기

```
<!DOCTYPE html>
                                                                    본인소개
                                                                                             ×
<html>
<head>
                                                                               (i) 127.0.0.1:5500/css01.html
                                                                                                              Ø
 <meta charset="UTF-8">
 <title>본인소개</title>
 <style>
                                                                  CSS 적용방법
   main {
     background-color: Dlightcyan;
                                                                   .-▼1. <mark>인라인 스타일</mark>
 </style>
                                                                     2. 내부 스타일
 <link rel="stylesheet" href="./styles/css01.cs</pre>
                                                                    3. 외부 스타일
</head>
<body>
 <main>
   <h1>CSS 적용방법</h1>
   <01>
     <li!style="background-color: | cornsilk;"</pre>
     내부 스타일
     외부 스타일
   </main>
</body>
</html>
```







JavaScript React JS



- · CSS 선택자는 HTML 요소를 선택하고 스타일을 적용하는 데 사용
  - · 참고 ) https://htmlcheatsheet.com/css/

선택자종류	형식	내용
전체 선택자 (Universal Selector)	*	HTML 문서 내의 모든 요소를 선택 head 요소도 포함
요소 선택자 (Element Selector)	태그명	지정된 태그명을 가지는 요소를 선택
아이디 선택자 (ID Selector)	#아이디명	지정된 id 속성을 가진 요소를 선택
클래스 선택자 (Class Selector)	.클래스명 지정된 class 속성을 가진 요소를 선택	
속성 선택자 (Attribute Selector)	태그[속성] 태그[속성=값]	지정된 속성을 갖는 모든 요소를 선택









```
<!DOCTYPE html>
<html>
                                                                                                                  전체 선택자
                                                                                      color: ■ darkgray
<head>
 <meta charset="UTF-8">
 <title>본인소개</title>
 <link rel="stylesheet" href="./styles/css01.css">
                                                                                                                  태그 선택자
                                                                                      color: | black;
</head>
<body>
 <main>
                                                                                                                 클래스 선택자
   <a href="http://google.com" id="id1" class="c1" target=" blank"> = </a>
   <h1 class="c1">CSS 선택자(/h1)
                                                                                      background-color: Deeige;
   <a href="http://naver.com" id="id2"_class="c2">出の田(/本)
   <a href="http://daum.net" id="id3" class="c2"-target= top"</pre>
                                                                                                                 아이디 선택자
                                                                                    #id2 {
 </main>
                                                                                      background-color: Dlavender;
</body>
</html>
                                    본인소개
                                    ← → C ① 127.0.0.1:5500/css01.... Q ☆ ☆ □ k :
                                                                                  a[target] {
                                                                                                                  속성 선택자
                                                                                      color: blue;
                                   CSS 선택자
                                   구글 네이버 다음
                                                                                   a[target="_blank"] {
                                                                                     background-color: aqua;
                                                                                                                       vaScript React JS
```

- · 그룹화 선택자 (Grouping Selector)
  - ㆍ 동일한 스타일을 적용하는 선택자는 쉼표(,)로 구분
- · 복합 선택자 (Combinator Selector)
  - · 자식 선택자 (Child Combinator): 선택자 A > 선택자 B
  - · 후손 선택자 (Descendant Combinator): 선택자 A 선택자 B
  - · 일반 형제 선택자 (General Sibling Selector): 선택자 A ~ 선택자 B











- · 가상 클래스 셀렉터 (Pseudo-Class Selector)
  - · HTML 요소의 상태나 위치에 따라 특정 스타일을 적용
  - · 가상 클래스는 HTML 문서의 DOM(Document Object Model) 구조에서 실제로 존재하지 않는 상태를 나타내며, 요소가 특정 상태에 있을 때 스타일을 적용
  - 요소의 특정 상태에 따라 스타일을 정의할 때 사용
    - ·:link 셀렉터가 방문하지 않은 링크일 때
    - · :visited 셀렉터가 방문한 링크일 때
    - · :hover 셀렉터에 마우스가 올라와 있을 때
    - · :active 셀렉터가 클릭된 상태일 때
    - ·: focus 셀렉터에 포커스가 들어와 있을 때









```
<!DOCTYPE html>
<html>
<head>
 <meta charset="UTF-8">
 <title>메뉴</title>
 <link rel="stylesheet" href="./styles/css02.css">
</head>
<body>
 <nav>
   <a href="#">Home</a>
    <a href="#">Frontend</a>
    <a href="#">Backend</a>
    <a href="#">Database</a>
    <a href="#">AI</a>
   </nav>
</body>
</html>
```

```
text-decoration: none;
.nav_menu {
 list-style: none;
 padding: 0;
.nav menu > li{
 list-style: none;
 padding : 8px 12px;
                              후손 선택자
.nav_menu a:hover {
 background-color: ☐ lightsalmon;
```

HTML

CSS JavaScript React JS









### CSS 상속

- · 상속 (Inheritance)
  - · CSS에서 특정 스타일 속성이 부모 요소로부터 자식 요소에 자동으로 전달되는 것
  - ㆍ 상속을 활용하면 여러 요소에 일관된 스타일을 쉽게 적용
  - ㆍ 상속되는 속성
    - ・ 텍스트와 관련된 속성들이 주로 상속
    - color, font-family, font-size, line-height, text-align
  - ㆍ 상속되지 않는 속성
    - ・ 박스 모델과 관련된 속성들이나 배경 속성 등은 상속되지 않음으로 각 요소에서 개별적으로 지정
    - margin, padding, border, width, height
    - background, background-color









### CSS 선택자 우선순위

- · 여러 개의 CSS 규칙이 동일한 HTML 요소에 적용될 때, 어떤 규칙이 더 높은 우선순위를 가지는지를 결정
- · 선택자 우선순위 규칙
  - 1. 인라인 스타일 (Inline Styles) 가장 높은 우선순위
  - 2. ID 선택자 (ID Selector)
  - 3. 클래스 선택자, 속성 선택자, 가상 클래스 선택자
  - 4. 요소 선택자
  - 5. \*(모든 요소를 선택) 또는 관계형 선택자(+, >, ~ 등)는 가장 낮은 우선순위









# CSS 단위

유형	단위	설명	특징
절대 단위	рх	화면의 고정된 픽셀 크기를 기준으로 하는 단위	해상도에 따라 다르게 보일 수 있음, 반응형 디자인에 비효율적
상대 단위	%	부모 요소의 크기를 기준으로 비율을 설정	가변적인 디자인에 유용함
상대 단위	em	부모 요소의 font-size를 기준으로 크기 설정	부모 요소의 크기에 영향을 받음
상대 단위	rem	최상위 요소 (html)의 font-size를 기준으로 크기 설정	전체 사이트의 일관된 폰트 크기 조정 가능
상대 단위	vw	뷰포트(화면 너비)의 1%를 의미	반응형 웹에서 화면 크기에 맞춘 디자인 가능
상대 단위	vh	뷰포트(화면 높이)의 1%를 의미	전체 높이를 기준으로 크기를 조정할 때 유용

HTML

CSS JavaScript React JS

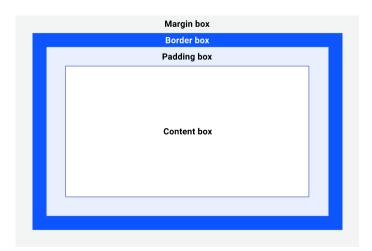






# 박스 모델(Box Model)

- · 모든 HTML 요소를 사각형 박스로 간주하고, 이 박스가 페이지에서 어떻게 공 간을 차지하는지를 설명하는 개념
- ・박스 모델 영역
  - · 콘텐츠 영역(content area)
    - · 글이나 이미지, 비디오 등 요소의 실제 내용을 포함
    - · 콘텐츠 너비(콘텐츠 박스 너비)와 콘텐츠 높이(콘텐츠 박스 높이)
  - · 안쪽 여백 영역(패딩 영역, padding area)
    - padding-top, padding-right, padding-bottom, padding-left
    - padding
  - · 테두리 영역(border area)
    - border-width
    - border
  - · 바깥 여백 영역(마진 영역, margin area)
    - margin-top, margin-right, margin-bottom, margin-left
    - margin











# 박스 모델(Box Model)

- · box-sizing 속성
  - ・ 박스 모델의 계산 방식을 변경
  - · content-box (기본값)
    - · 콘텐츠 영역의 크기는 width와 height 값으로 결정되어 패딩과 테두리는 이 크기에 추가로 더 해져 요소의 전체 크기를 결정
  - border-box
    - ㆍ 패딩과 테두리를 포함한 전체 요소의 크기가 width와 height 값으로 결정
    - · 레이아웃을 더 쉽게 관리할 수 있게 해주며, 요소의 전체 크기를 보다 직관적으로 설정















## 박스 모델(Box Model) 제어

### ・블록박스

- · 블록 레벨 요소는 사용 가능한 가로 공간을 전부 차지하며, 항상 새로운 줄에서 시작
- · width와 height 속성을 사용해 요소의 크기를 제어할 수 있으며, 패딩, 여백, 테두리 등의 스타일을 적용
- · \div\, \p\, \h1\, \ul\, \li\

### • 인라인 박스

- · 인라인 레벨 요소는 콘텐츠의 크기만큼의 공간만 차지하며, 같은 줄에 다른 인라인 요소들 과 함께 배치
- · width와 height 속성을 사용할 수 없으며, 요소의 크기는 콘텐츠에 따라 결정
- · 패딩, 여백, 테두리 등의 스타일은 적용되지만, 블록 요소와 다르게 다른 요소를 밀어내지 는 않음
- · <span>, <a>, <strong>, <em>









## 박스 모델(Box Model) 제어

- · display 속성
  - 요소가 블록 박스나 인라인 박스로 렌더링되는 방식을 제어
  - · 박스 모델의 동작 방식을 근본적으로 변경할 수 있어, 레이아웃을 유연하게 설계하는 데 유용
  - display: block;
    - · 요소를 블록 레벨 요소로 설정하여, 전체 가로 폭을 차지
  - display: inline;
    - ㆍ 요소를 인라인 레벨 요소로 설정하여, 콘텐츠의 크기만큼의 공간만 차지
  - display: inline-block;
    - · 블록 레벨 요소처럼 크기를 지정할 수 있으면서, 인라인 레벨 요소처럼 같은 줄에 나란히 배치





JavaScript Reac

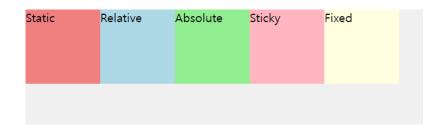






# 위치 지정(position)

- ㆍHTML 요소의 배치를 결정하는 데 사용
  - 요소가 페이지에서 어떻게 배치될지를 제어하며, 다른 요소들과 관계 속에서 위치 설정
- · 위치 지정 값
  - Postion 적용 전: static
    - 모든 HTML 요소의 기본 위치 지정 방식
    - top, right, bottom, left 값의 영향을 받지 않음



- relative (상대적 위치 지정)
  - 원래 위치를 기준으로 상대적으로 이동
  - right, bottom, left 값에 따라 요소를 조정
  - 다른 요소들은 이 위치 변경에 영향을 받지 않음













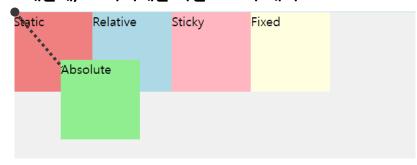




# 위치 지정(position)

### absolute (절대적 위치 지정)

- 조상 요소를 기준으로 위치를 지정
- 절대 위치 지정된 요소는 문서의 흐름에서 제거되기 때문에, 그 자리에는 다른 요소가 배치

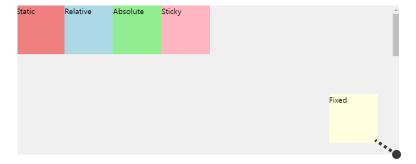


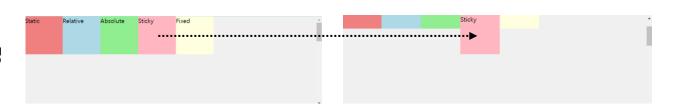
### sticky

- 일반적인 문서 흐름에 따라 배치되다가,스크롤 위치가 특정 임계점에 도달하면 그 위치에 고정
- 스크롤링할 때 특정 위치에 "붙어" 있도록 설정

### fixed (고정 위치 지정)

- 브라우저 창을 기준으로 고정된 위치에 배치
- 스크롤을 하더라도 요소는 항상 동일한 위치
- fixed 요소는 문서의 흐름에서 제거





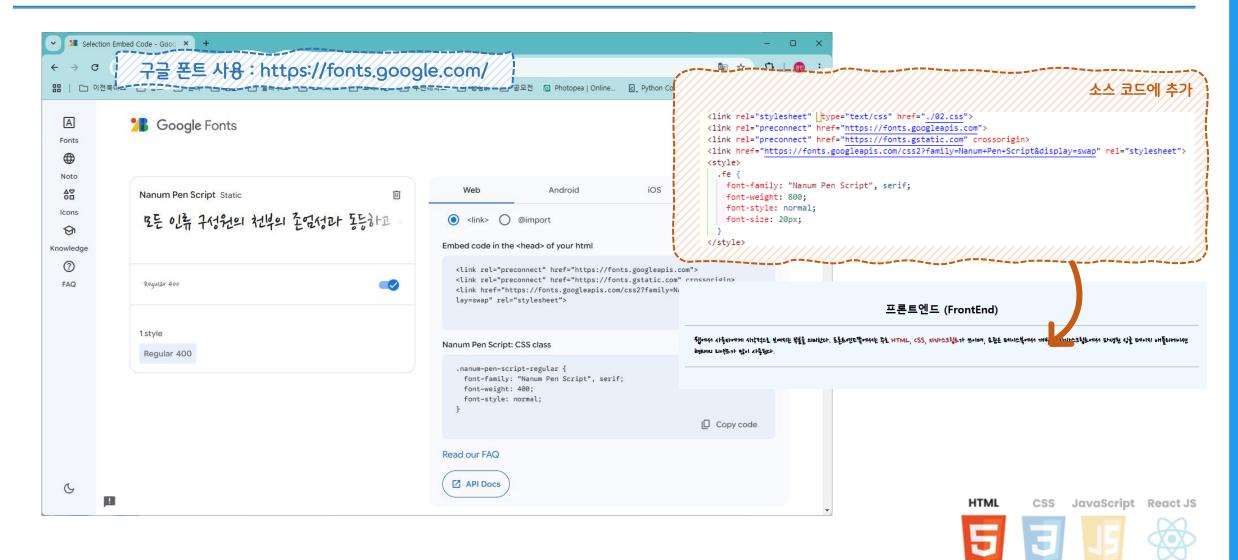




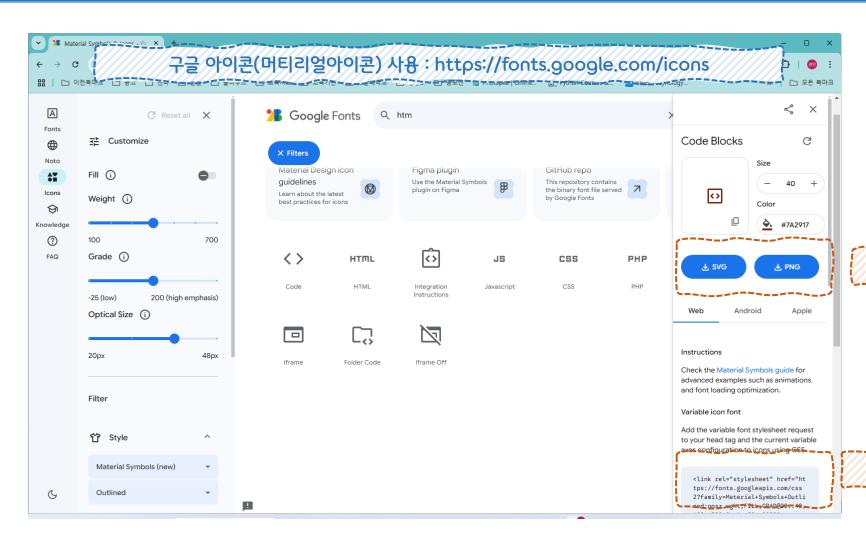




## 웹 폰트 사용



# 아이콘 폰트 사용 - 머티리얼 아이콘



이미지로 저장하여 사용

코드로 사용 HTML

CSS Java

JavaScript React JS

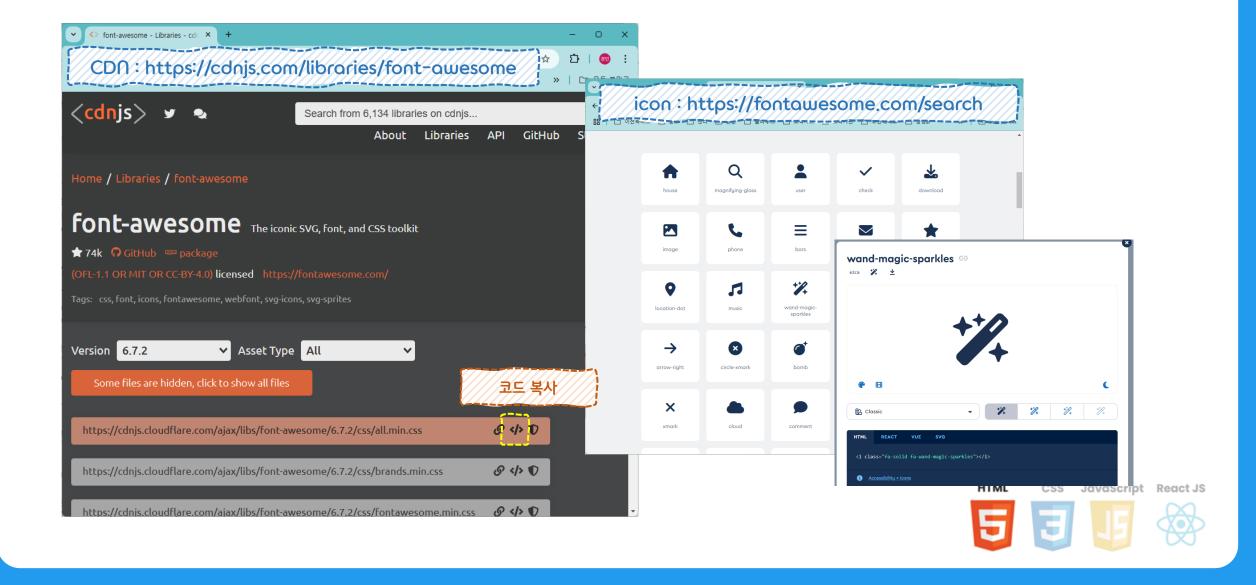








# 아이콘 폰트 사용 - Font Awesome



# Flexbox (Flexible Box Layout)

- · 복잡한 레이아웃을 보다 간단하게 구성할 수 있도록 돕는 CSS3 레이아웃 모듈
  - · 컨테이너(Container)와 그 안에 배치된 아이템(Items)들 간의 관계를 관리하여, 다양한 화면 크기와 장치에 맞는 유연한 레이아웃을 쉽게 만들 수 있게 해 줌
  - ㆍ 방향성, 정렬, 공간 분배 등의 문제가 있는 경우에 유용
  - · CSS 속성: display: flex;
- · Flexbox의 기본 개념
  - · 플렉스 컨테이너 (Flex Container)
    - · Flexbox 레이아웃이 적용된 요소로 이 컨테이너 내의 모든 자식 요소는 플렉스 아이템이 됨
  - · 플렉스 아이템 (Flex Items)
    - · 플렉스 컨테이너의 자식 요소들로, 컨테이너의 레이아웃 규칙에 따라 배치















# Flexbox (Flexible Box Layout)

### · 1차원으로 항목 그룹을 배치하도록 설계된 레이아웃

플렉스 컨테이너 (Flex Container)

플렉스 아이템 (Flex Items) 플렉스 아이템 (Flex Items) 플렉스 아이템 (Flex Items)



중심축

### flex-direction : 정렬방향을 지정

- row: 요소들을 텍스트의 방향과 동일하게 정렬
- row-reverse: 요소들을 텍스트의 반대 방향으로 정렬
- column: 요소들을 위에서 아래로 정렬
- column-reverse: 요소들을 아래에서 위로 정렬

### justify-content:

### 요소를 중심축으로 정렬

- flex-end: 요소들을 컨테이너의 오른쪽으로 정렬
- center: 요소들을 컨테이너의 가운데로 정렬
- space-between: 요소들 사이에 동일한 간격
- space-around: 요소들 주위에 동일한 간격

### align-items:

### 요소를 반대축으로 정렬

- flex-start: 요소들을 컨테이너의 꼭대기로 정렬
- flex-end: 요소들을 컨테이너의 바닥으로 정렬
- center: 요소들을 컨테이너의 세로선 상의 가운데로 정렬
- baseline: 요소들을 컨테이너의 시작 위치에 정렬
- stretch: 요소들을 컨테이너에 맞도록 늘림

act JS



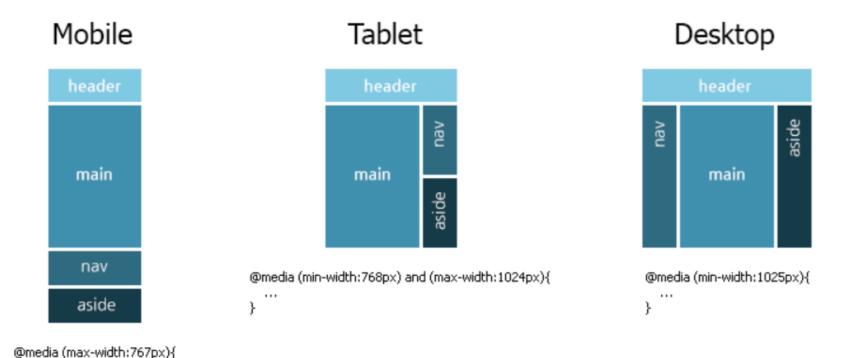






# 반응형 웹

- ・웹 페이지가 다양한 장치와 화면 크기에 맞게 자동으로 조정되도록 설계된 웹 디자인 접근 방식
  - ・ 하나의 웹사이트가 데스크톱, 태블릿, 스마트폰 등 여러 디바이스에서 최적의 사용자 경험을 제공











## 미디어 쿼리(media queries)

- · CSS모듈로 사이트에 접속하는 장치에 따라 특정한 CSS 스타일을 적용
  - ㆍ 웹 브라우저 화면 크기에 따라 사이트 레이아웃을 재배치

@media only screen and (min-width:400px) and (max-width:1000) {}

- only 미디어쿼리를 지원하지 않는 웹브라우저에서는 미디어쿼리를 무시하고 실행하지 않음
- not
   not 다음에 지정한
   미디어 유형 제외

- all 모든 미디어 유형에서 사용
- print
   인쇄장치에서 사용
- screen 컴퓨터나 스마트폰 화면에서 사용
- 기타
   tv, aural, braile, handeld,
   projection, tty, emmbossed

- · 웹 문서 가로 너비와 세로 높이 속성
- with, height
   웹 페이지의 가로 너비, 세로 높이
- min-width, min-height
   웹 페이지의 최소 너비, 최소 높이
- max-width, max-height
   웹 페이지의 최대 너비, 최소 높이
- 기타
   device-width, device-height
   min-device-width(height),
   max-device-width(height)

HIML

SS Javas

JavaScript React JS









### CSS3 Units

### ・절대단위

- · 디바이스의 해상도와 관계없이 고정된 크기를 나타내는 단위
- · px :디바이스 해상도(resolution)에 따라 상대적인 크기

### • 상대단위

- ㆍ 다른 요소나 환경에 따라 크기가 유동적으로 변하는 단위로 특히 반응형 웹 디자인에서 유 용하게 사용되며, 컨테이너나 상위 요소에 비례하여 크기를 지정
- · %: 부모요소의 백분률 단위의 상대 단위
  - 레이아웃에서 요소 간의 비율을 유지하면서 크기를 조정할 때 사용
- · em: 부모 요소의 글꼴 크기에 비례하는 단위
  - · 주로 텍스트 크기나 간격을 상대적으로 조정할 때 사용
- · rem : 최상위 요소(html)의 사이즈를 기준
  - ・ 전체 레이아웃이나 글꼴 크기를 통일성 있게 조정할 때 사용







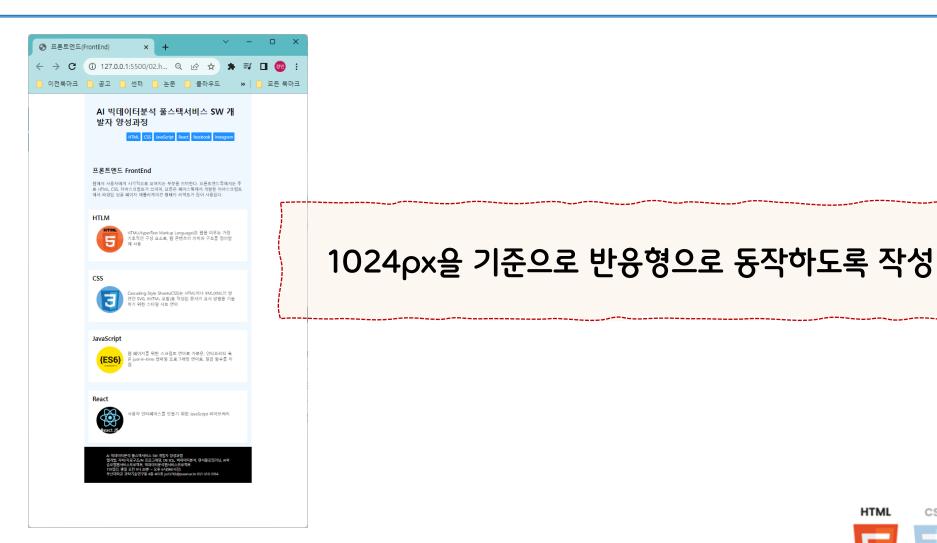








# 해결문제



HTML

JavaScript React JS







