

AI 활용 빅데이터분석

풀스택웹서비스 SW 개발자 양성과정

CSS (Cascading Style Sheets)



• CSS (Cascading Style Sheets)

–HTML로 작성된 문서의 표시 방법을 기술하기 위한 스타일 시트

- 다양한 장치 및 화면 크기에 대한 디스플레이의 디자인, 레이아웃 및 변형을 포함하여 웹 페이지의 스타일을 정의

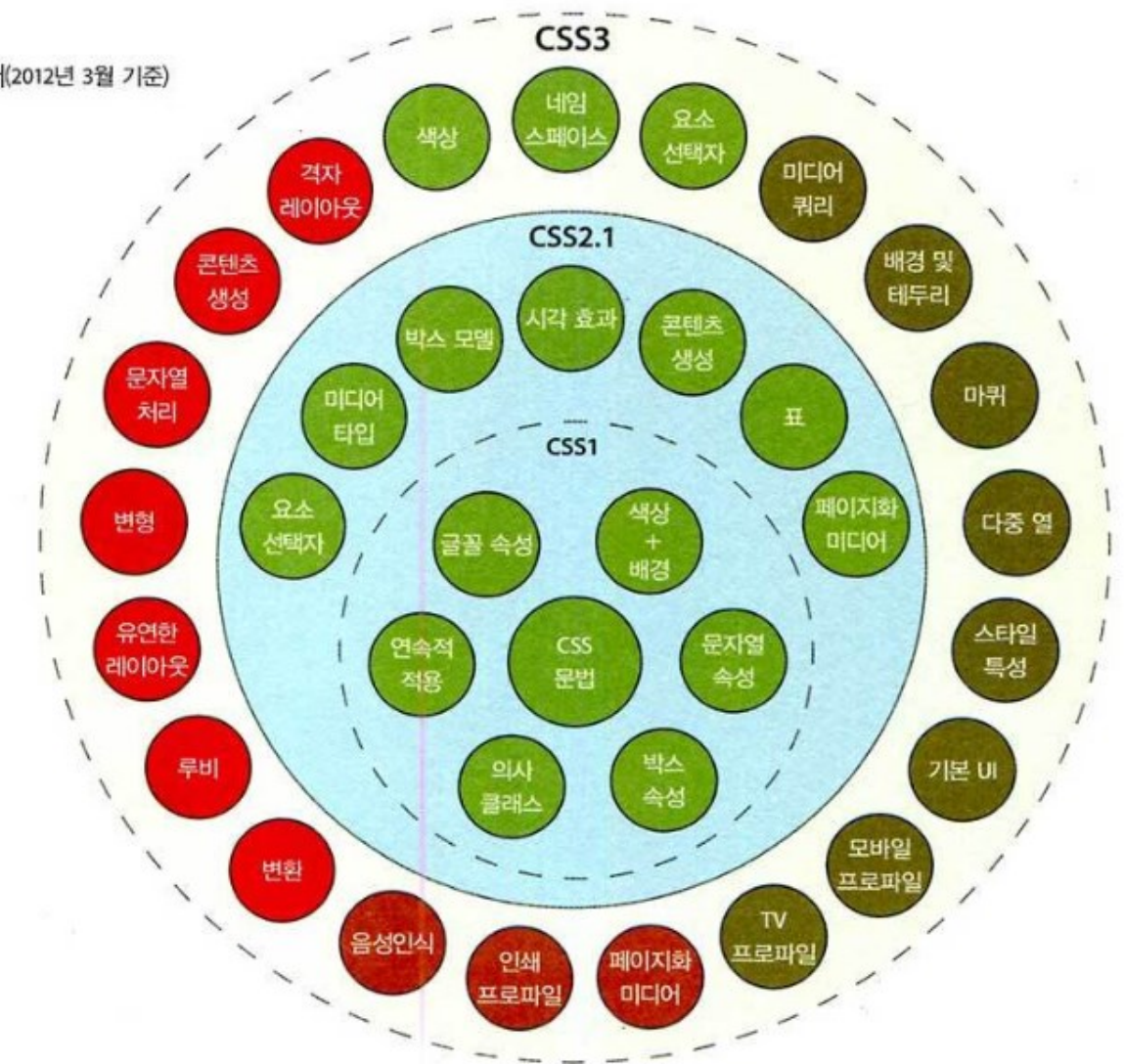
–규칙 기반 언어

- 웹 페이지의 특정 요소 또는 요소 그룹에 적용할 스타일 그룹을 지정하는 규칙을 정의
- **HTML** 문서에 있는 요소들에 선택적으로 스타일을 적용

CSS3

분류 & 상태(2012년 3월 기준)

- W3C 권고안
- 권고 후보
- 최종 검토 중
- 작업 초안



<https://webdir.tistory.com/338>

CSS 구성

선택자(Selector)

선언 (Declaration)

```
section {  
  background-color : yellow ;  
}
```

속성 (Property)

값 (Value)

• 선택자(selector)

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

• 선언(declaration)

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

HTML문서에 CSS3 스타일 시트 만들기

- HTML 문서에 CSS3 스타일 시트 만드는 방법

- 인라인 스타일 (Inline style) **우선순위 1**

- 태그의 **style** 속성으로 사용

- 내부 스타일 시트 (Internal style sheet) **우선순위 2**

- **<style>... </style>**

- 외부 스타일 시트 (External style sheet) **우선순위 3**

- **<link rel="stylesheet" type="text/css" href="03.css">**

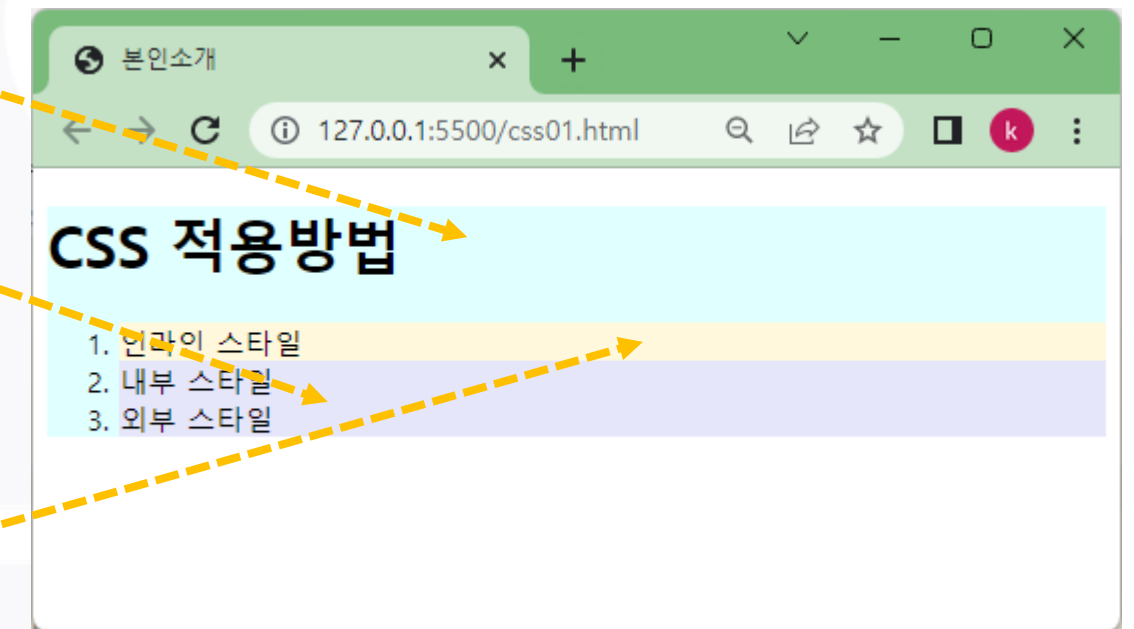
HTML문서에 CSS3 스타일 시트 만들기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>본인소개</title>
  <style>
    main {
      background-color: lightcyan;
    }
  </style>
  <link rel="stylesheet" href="./styles/css01.css">
</head>
<body>
  <main>
    <h1>CSS 적용방법</h1>
    <ol>
      <li style="background-color: cornsilk;">인라인 스타일</li>
      <li>내부 스타일</li>
      <li>외부 스타일</li>
    </ol>
  </main>
</body>
</html>
```

내부스타일 (우선순위 2)

외부스타일 (우선순위 3)

인라인스타일 (우선순위 1)

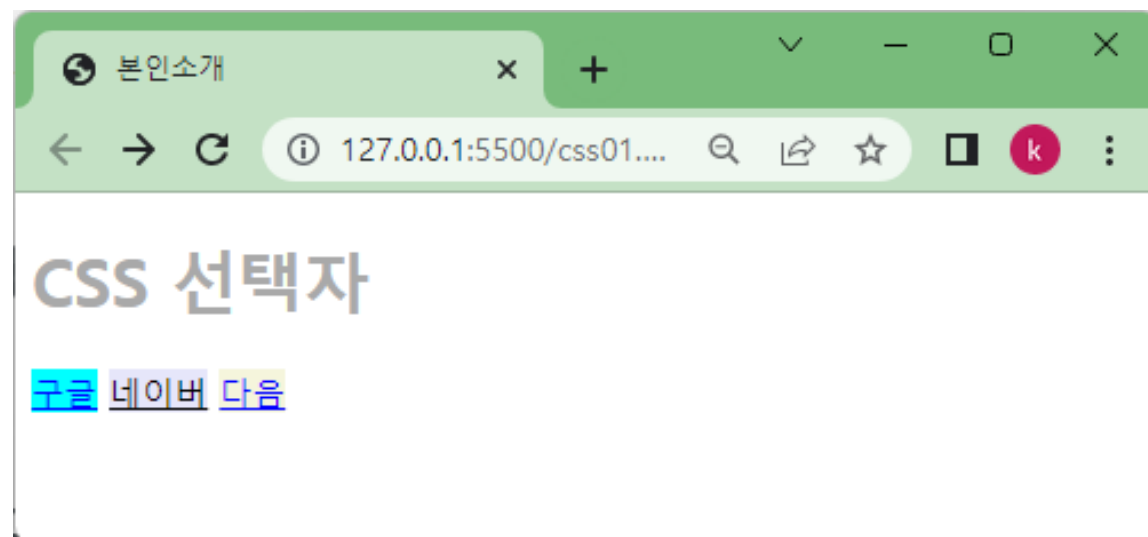


CSS 선택자

선택자종류	형식	내용
전체 선택자 (Universal Selector)	*	HTML 문서 내의 모든 요소를 선택 head 요소도 포함
태그 선택자 (Type 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

CSS 선택자

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>본인소개</title>
  <link rel="stylesheet" href="./styles/css01.css">
</head>
<body>
  <main>
    <h1 class="c1">CSS 선택자</h1>
    <a href="http://google.com" id="id1" class="c1" target="_blank">구글</a>
    <a href="http://naver.com" id="id2" class="c2">네이버</a>
    <a href="http://daum.net" id="id3" class="c2" target="_top">다음</a>
  </main>
</body>
</html>
```



전체 선택자

```
* {
  color: darkgray;
}
```

태그 선택자

```
a {
  color: black;
}
```

클래스 선택자

```
c2 {
  background-color: beige;
}
```

클래스 선택자

```
#id2 {
  background-color: lavender;
}
```

속성 선택자

```
a[target] {
  color: blue;
}
```

```
a[target="_blank"] {
  background-color: aqua;
}
```

CSS 선택자

- 셀렉터 그룹화

- 동일한 스타일을 적용하는 선택자는 쉼표(,)로 구분

- 복합 선택자 (Combinator)

- 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)

- 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

CSS 선택자

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

태그선택자

```
a {
  text-decoration: none;
}
```

클래스 선택자

```
nav_menu {
  list-style: none;
  padding : 0;
}
```

자식 선택자

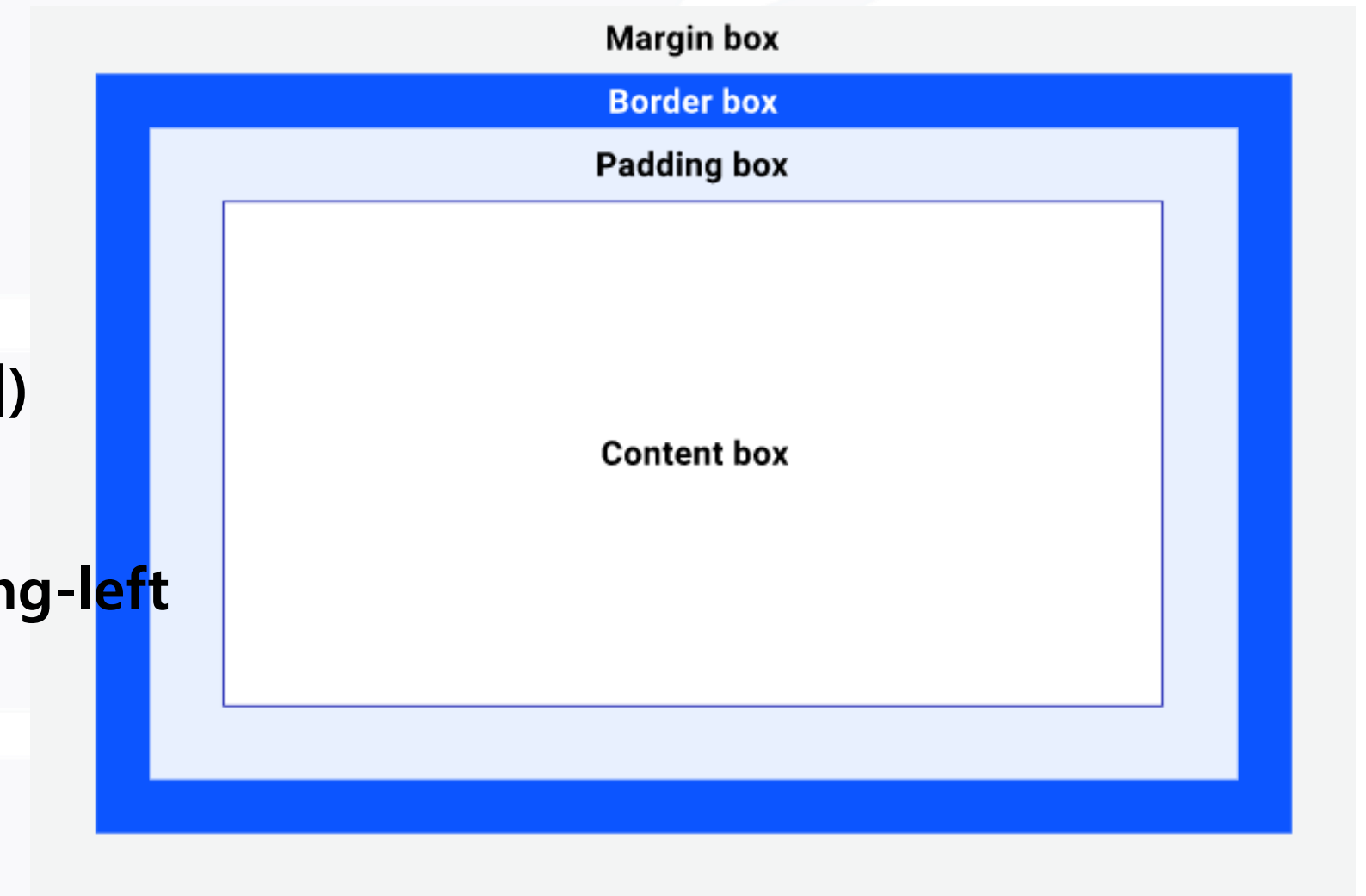
```
nav_menu > li {
  list-style: none;
  padding : 8px 12px;
}
```

후손 선택자

```
nav_menu a:hover {
  background-color: lightsalmon;
}
```

박스 모델(Box Model)

- CSS가 표시하는 모든 것은 박스
- 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) 제어

- 블록박스

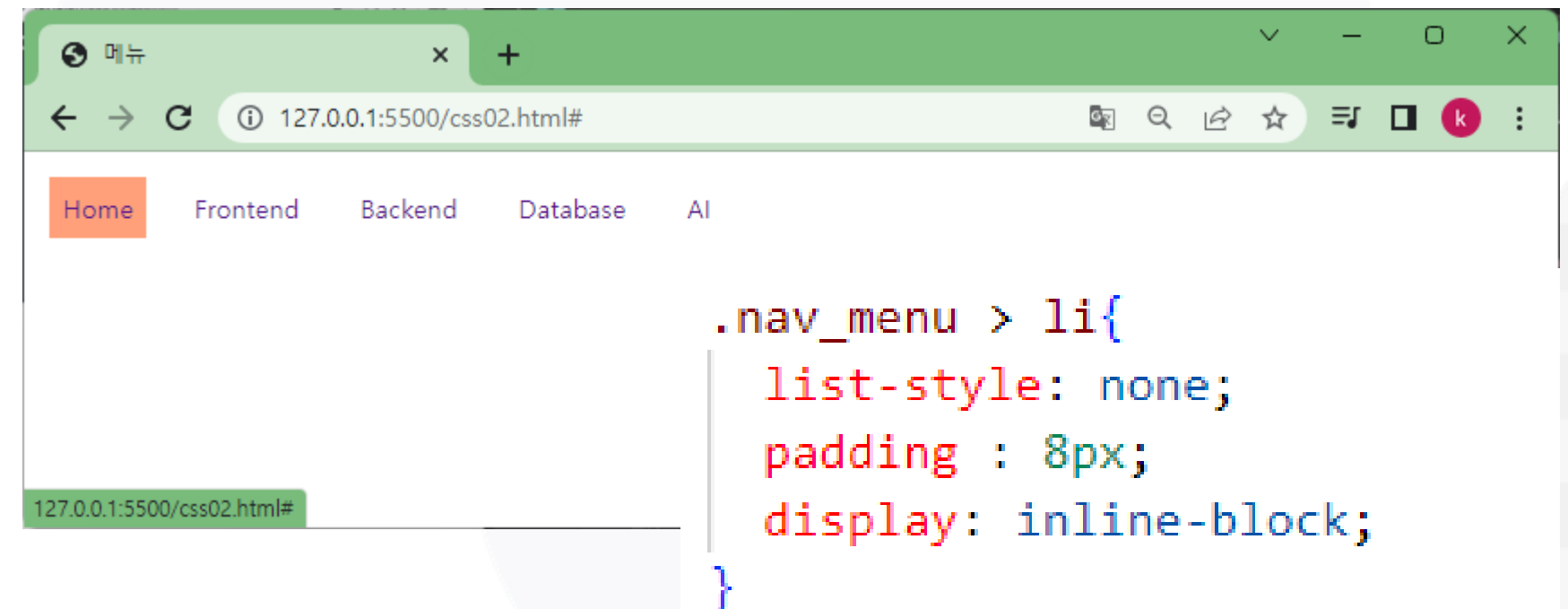
- 사용 가능한 공간을 양 옆으로 100% 사용
- width와 height속성을 사용해 스타일 제어
- 패딩과 여백, 테두리 스타일 제어하며 다른 요소들은 밀려남

- 인라인박스

- 컨텐츠 박스만큼의 크기만 가짐
- width와 height 속성을 사용할 수 없음

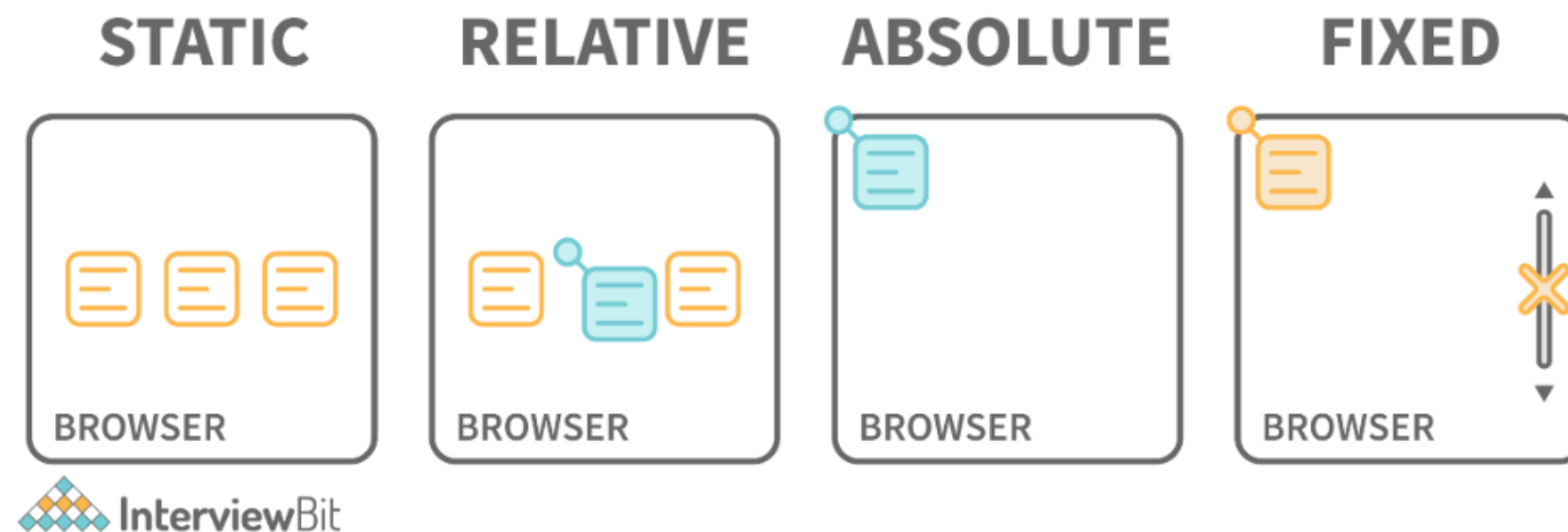
- 박스 모델 제어 : display

- display : inline
- display : block
- display : inline-block



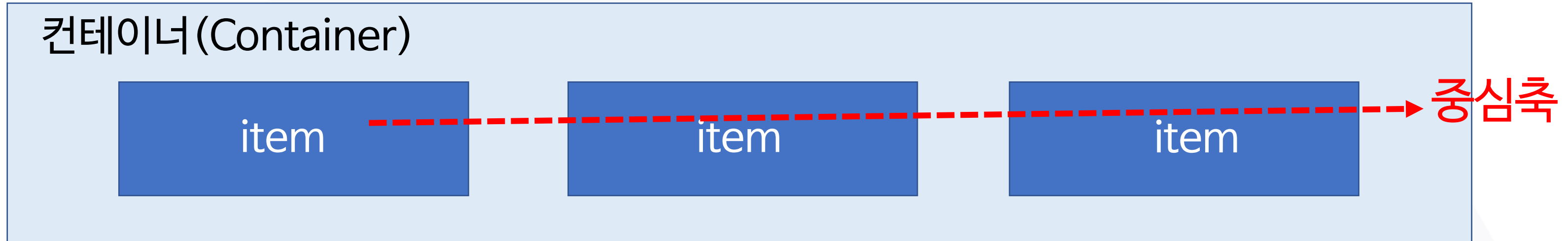
위치 지정(position)

- 요소에 사용되는 위치 지정 방법의 유형을 지정
- 속성값
 - **static** : 기본값 , HTML에 정의된 순서대로 표시
 - **relative** : 원래 있어야 하는 위치에서 상대적을 top,right,bottom,left 값을 기반으로 자체적으로 옮김
 - **absolute** : 적용된 요소의 상위 요소에서 top,right,bottom,left 값을 기반으로 위치를 옮김
 - **fixed** : 박스를 벗어나 웹페이지를 기준으로 top,right,bottom,left 값을 기반으로 위치를 옮김
 - **sticky** : 원래있어야 하는 자리에서 있으면서 스크롤링되어라도 그위치에 있음



플렉스 박스(Flexbox)

• 1차원으로 항목 그룹을 배치하도록 설계된 레이아웃



• 컨테이너에 적용되는 속성

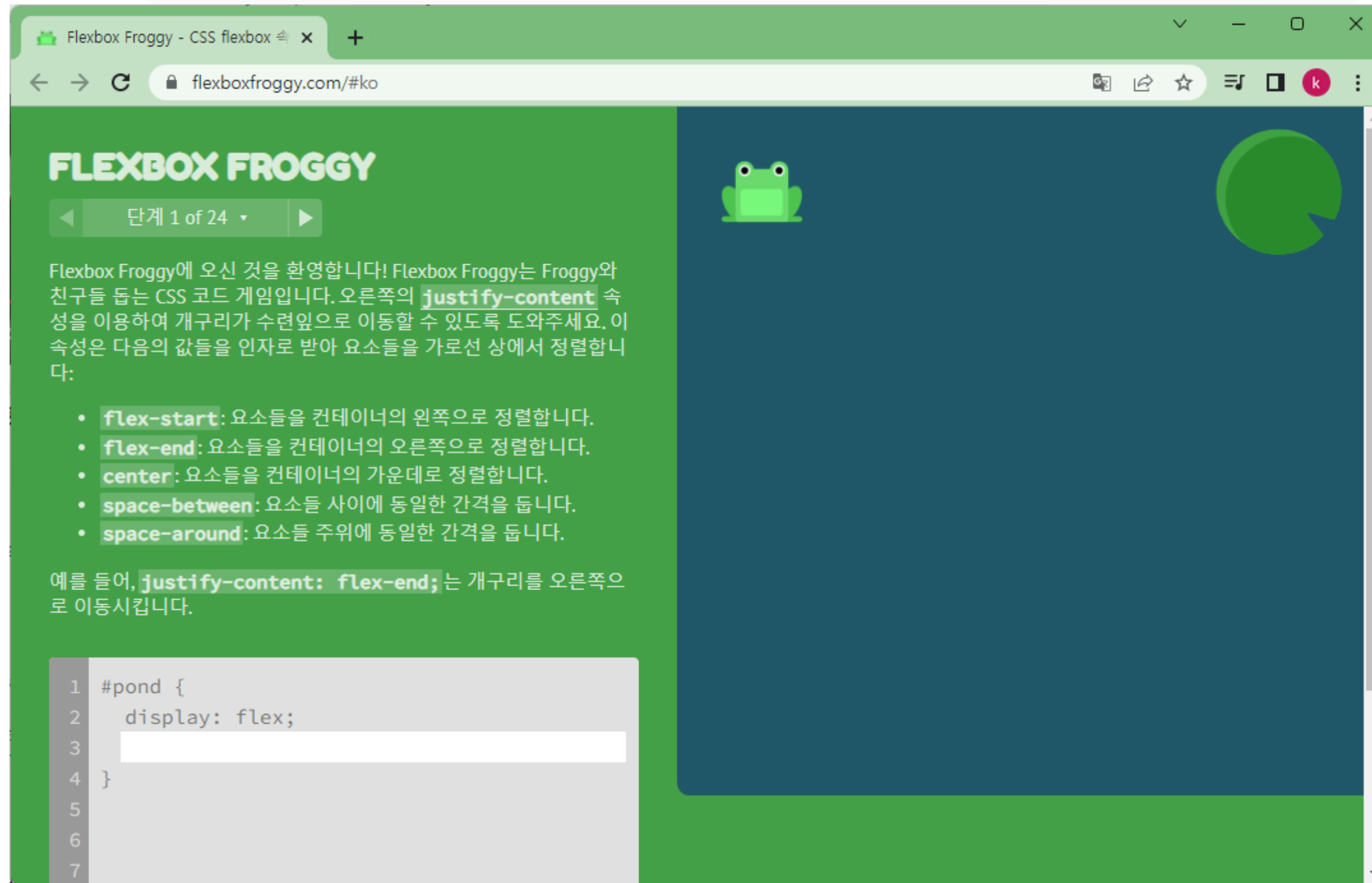
- `display: flex`
- `flex-direction: row` (중심축설정)
- `flex-wrap: nowrap` (item들이 무조건 한줄에)
- `flex-flow: row nowrap`
- `justify-content: center` (중심축에서 배치)
- `align-items: center` (반대축에서 배치)
- `align-content: center` (반대축에서 배치)

• 아이টে에 적용되는 속성

- `order` (순서변경)
- `flex-grow`: 숫자 (컨테이너를 채움)
- `flex-shrink`: 숫자 (컨테이너가 작아질때 크기)
- `flex-basis`: 비율 (컨테이너 비율에 따라 크기지정)
- `align-self: center` (컨테이너에서 벗어나서 지정)

플렉스 박스(Flexbox)

- <https://flexboxfroggy.com/#ko>



플렉스 박스(Flexbox) 연습

- **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
- **flex-wrap : 요소들을 한줄 또는 여러줄에 걸쳐 정렬**
 - nowrap: 모든 요소들을 한 줄에 정렬
 - wrap: 요소들을 여러 줄에 걸쳐 정렬
 - wrap-reverse: 요소들을 여러 줄에 걸쳐 반대로 정렬

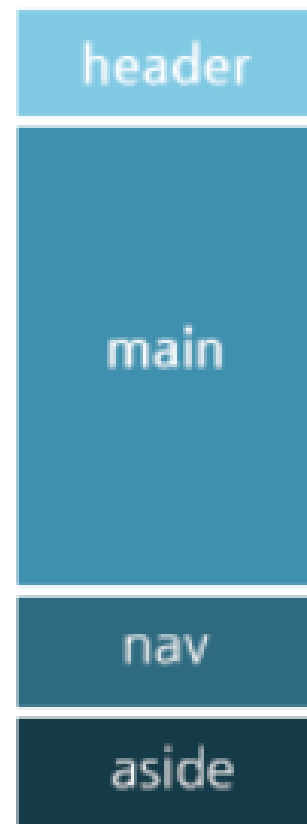
플렉스 박스(Flexbox) 연습

- **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

반응형 웹

- 사이트가 보여지는 장치의 요구와 기능에 대응하도록 사이트를 만드는 방법

Mobile



```
@media (max-width: 767px){  
  ...  
}
```

Tablet



```
@media (min-width: 768px) and (max-width: 1024px){  
  ...  
}
```

Desktop



```
@media (min-width: 1025px){  
  ...  
}
```

뷰포트 설정

- 메타 뷰포트 태그

- 페이지의 크기와 배율을 제어하는 방법에 대한 정보를 브라우저에 제공

- width=device-width

- 장치 독립적인 픽셀로 화면 너비를 일치

- initial-scale=1

- 브라우저가 장치 방향에 관계없이 **CSS** 픽셀과 장치 독립적인 픽셀 사이에 **1:1** 관계를 유지하도록 지시하고 페이지에서 전체 가로 너비를 활용

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <meta name="viewport" content="width=device-width, initial-scale=1">
    ...
  </head>
  ...
```


CSS3 Units

- 절대단위

- px :디바이스 해상도 (resolution)에 따라 상대적인 크기

- 상대단위

- %: 부모요소의 백분율 단위의 상대 단위
- em: 배수 단위로 상대 단위
 - 폰트 사이즈 설정이나 콘텐츠를 포함하는 컨테이너의 크기 설정에 사용하면 상대적인 설정이 가능하여 편리
- rem : 최상위 요소(html)의 사이즈를 기준

미디어 쿼리 중단점(break point)

- 미디어 쿼리를 작성할 때 화면 크기에 따라 서로 다른 CSS를 적용할 분기점

- 스마트폰 : 320 x 480

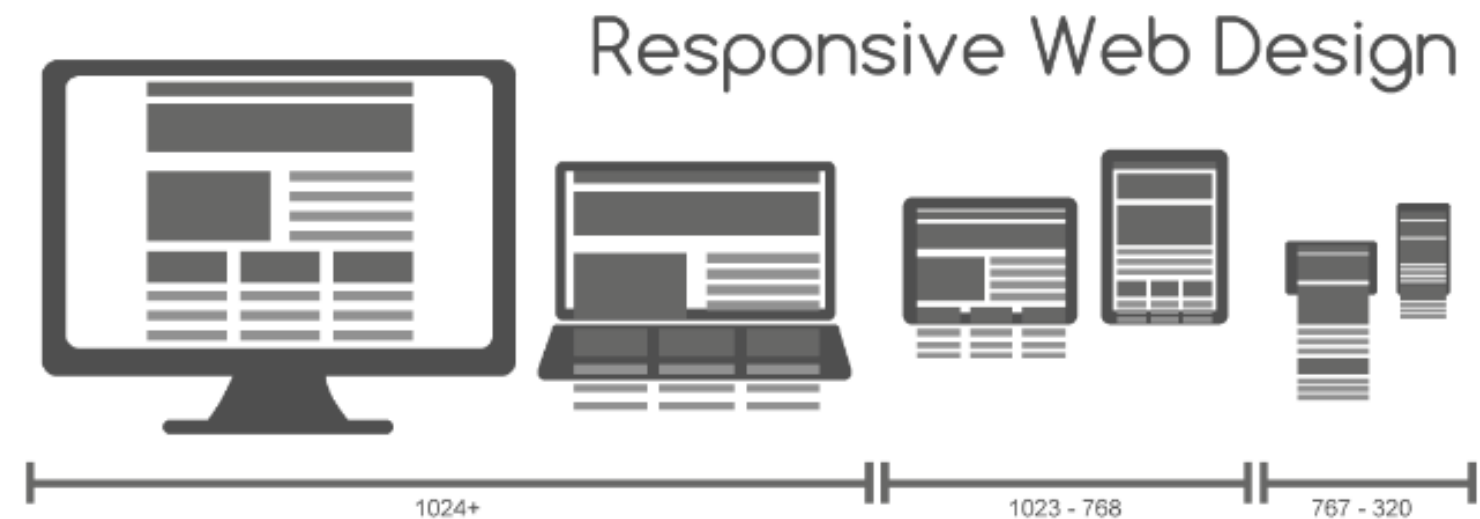
- 태블릿 : 768 x 1024

- 데스크톱: 1024 이상

- 참고 : <https://yesviz.com/devices.php>



스마트폰용인 320px에 맞추어 웹사이트를 제작하더라도
스마트폰용 모바일 브라우저는 웹킷(webkit)을 기반으로
기본 뷰포트 너비는 980px로 표시됨



CSS3 Units

- Viewport 단위

- viewport를 기준으로 한 상대적 사이즈
- vw : viewport 너비의 1/100(1%)
- vh : viewport 높이의 1/100(1%)
- vmin : viewport 너비 또는 높이 중 작은 쪽의 1/100
- vmax : viewport 너비 또는 높이 중 큰 쪽의 1/100

미디어 쿼리(media queries)

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

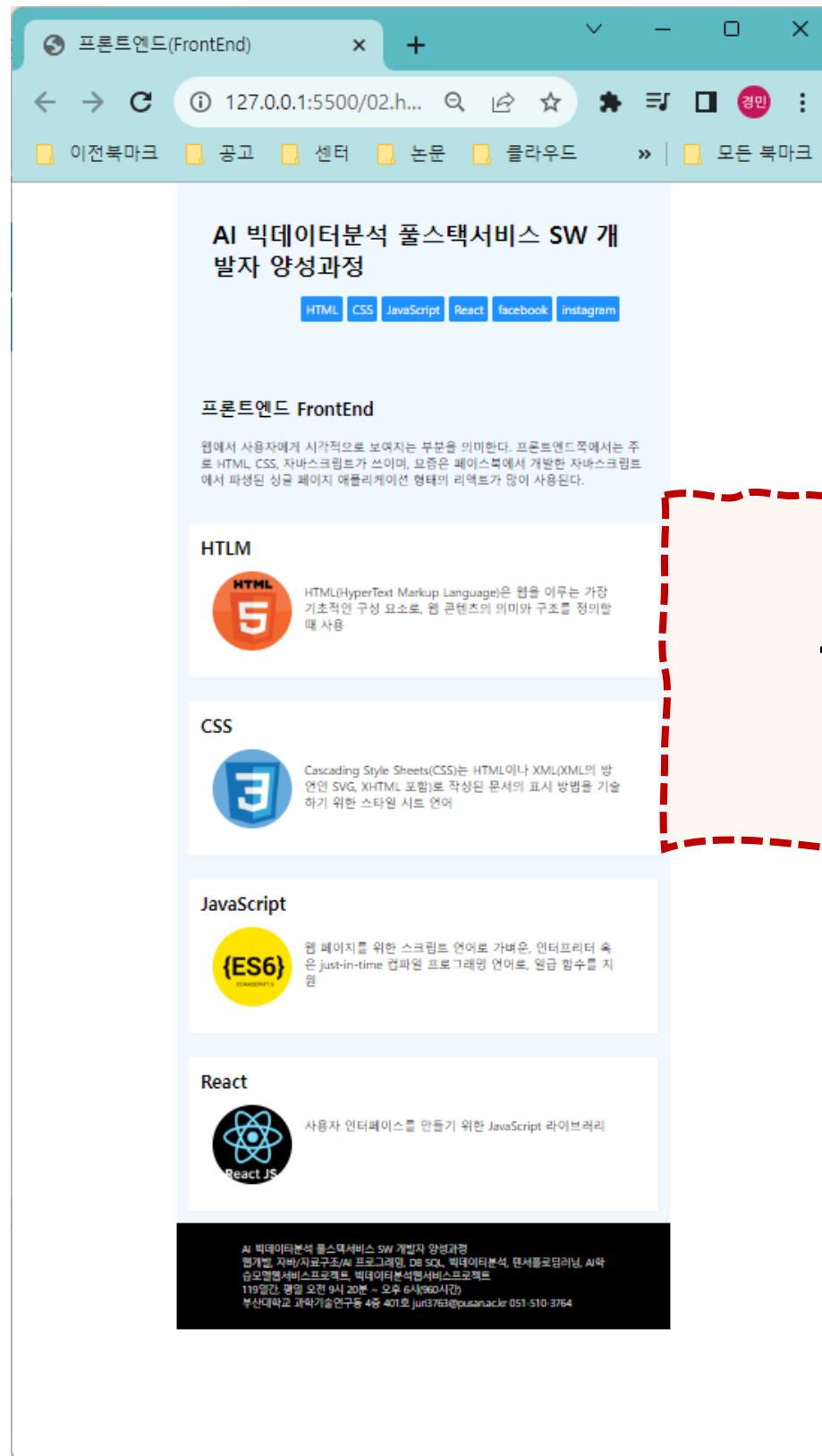
@media **only screen** and (min-width:400px) and (max-width:1000) {}

- 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, braille, handheld,
projection, tty, embossed

- 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)

실습문제



1024px을 기준으로 반응형으로 동작하도록 작성