HTML/CSS

반응형 웹 및 미디어 쿼리

김태민

저번에 우리는

CSS3와 현대적 발전 (2000년대 말~2010년대)

주요 기능:

- 선택자 모듈: 속성 선택자([attr]), 가상 요소(::before, ::after)
- 레이아웃: Flexbox, Grid, 반응형 디자인을 위한 미디어 쿼리
- 시각효과/애니메이션 : border-radius, box-shadow, gradient, transform, @keyframes, transition

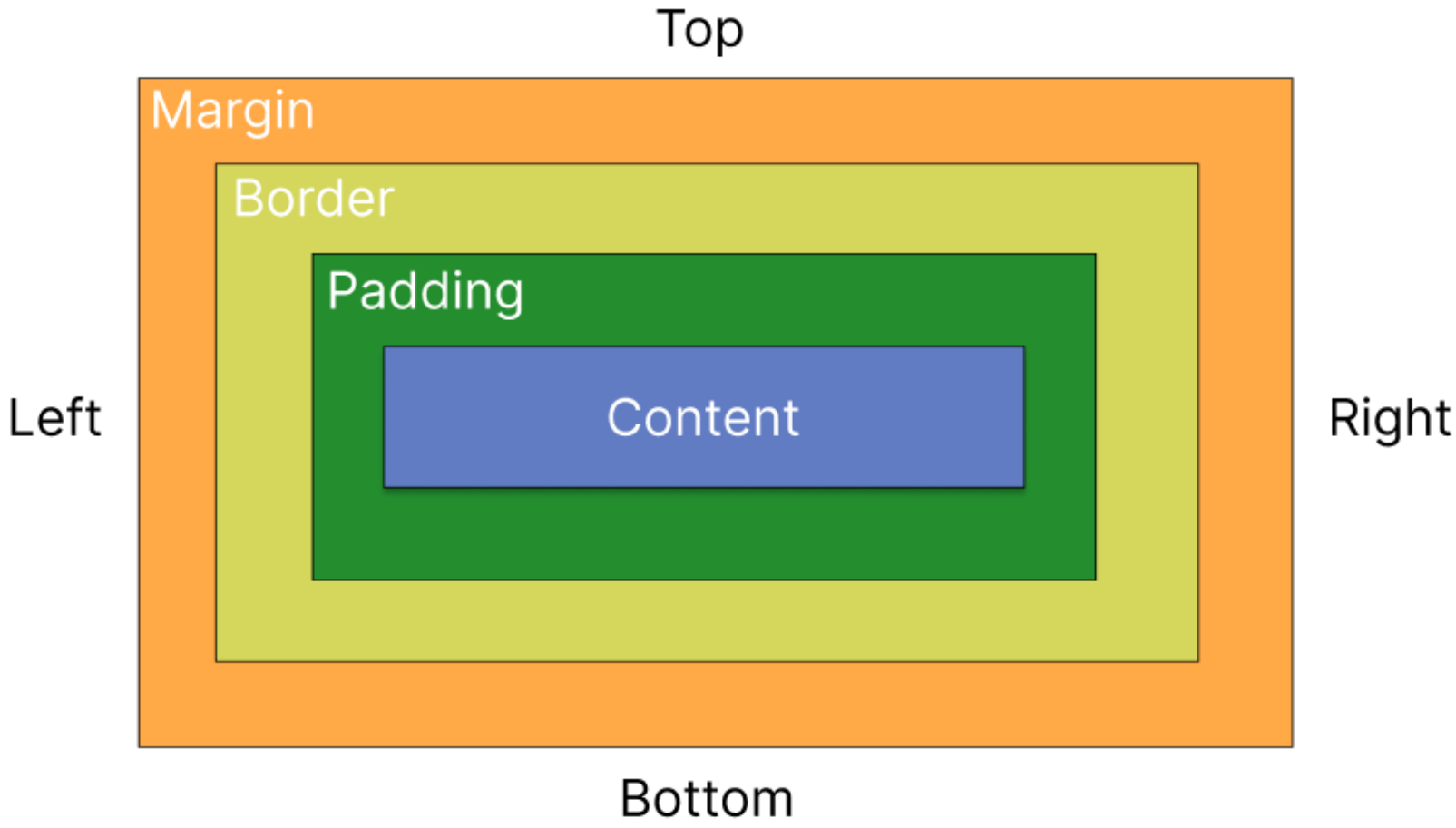
영향:

- 모바일 웹 급성장(스마트폰 보급)으로 반응형 디자인 필수
- 플래시 사용 감소, HTML5+CSS3 중심의 웹 개발 표준화

표준화:

- 브라우저 호환성 대폭 개선(크롬, 사파리, 엣지)

박스모델 개념



Block: 정적인 구조, 콘텐츠 블록 단위로 쌓을 때 적합
Flex: 동적인 배치, 요소 간 간격/정렬 조정이 필요한 경우 유리

Block vs Flex

항목	**Block**	**Flex**
기본 동작	새 줄에서 시작, 부모 너비 100% 차지	자식 요소를 유연하게 배치 (행/열)
레이아웃	수직 쌓임, 고정된 블록 형태	1차원 레이아웃, 정렬 및 크기 조정 가능
자식 요소 제어	자식 요소의 배치 직접 제어 불가	justify-content, align-items로 정렬
주요 속성	width, height, margin, padding	flex-direction, flex-wrap, gap
유연성	고정된 구조에 적합	반응형, 동적 레이아웃에 적합

Position 속성

CSS의 position 속성은 HTML 요소의 위치를 제어하며, 요소가 문서 흐름에서 어떻게 배치되는지 결정

역할: 요소를 원하는 위치에 정밀하게 배치하거나, 스크롤 및 뷰포트에 따라 동작을 조정

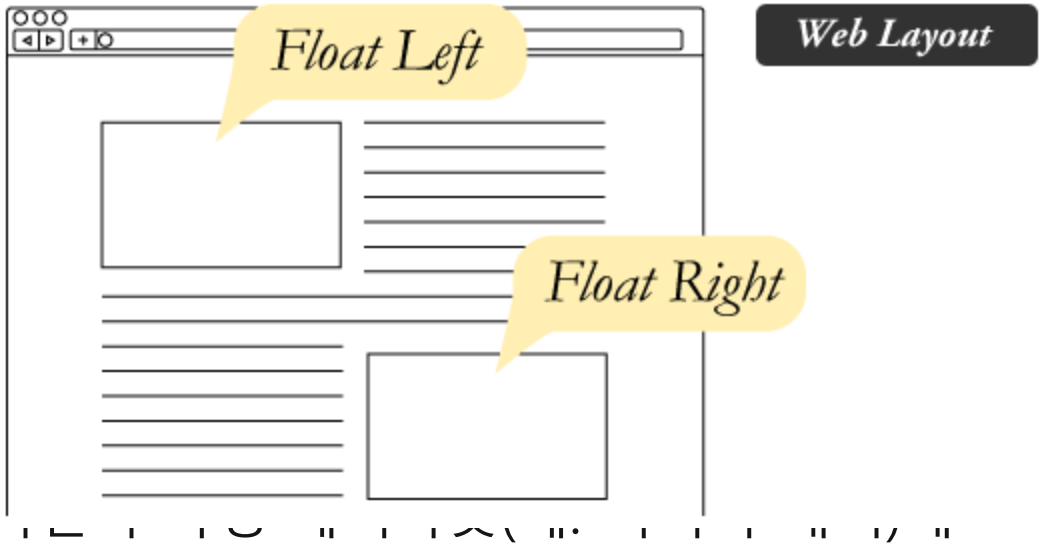
주요 값: static, relative, absolute, fixed, sticky

참고: position 속성은 top, right, bottom, left 속성과 함께 사용해 위치 조정

Float와 Clear

Float:

- CSS 속성으로, 요소를 문서 흐름에서 띄워 왼쪽 또는 오른쪽으로 배치, 주변 콘텐츠가 이를 감싸도록 함
- 이미지와 텍스트를 나란히 배치하거나, 간단한 레이C



Clear:

- Float된 요소의 영향을 제거하여 정상적인 문서 흐름
- Float로 인해 어긋난 레이아웃을 정리
- 현대에는 Flexbox와 Grid가 주로 사용되지만, float는 서 유용

실습 환경

실습환경

OS : Window / Mac

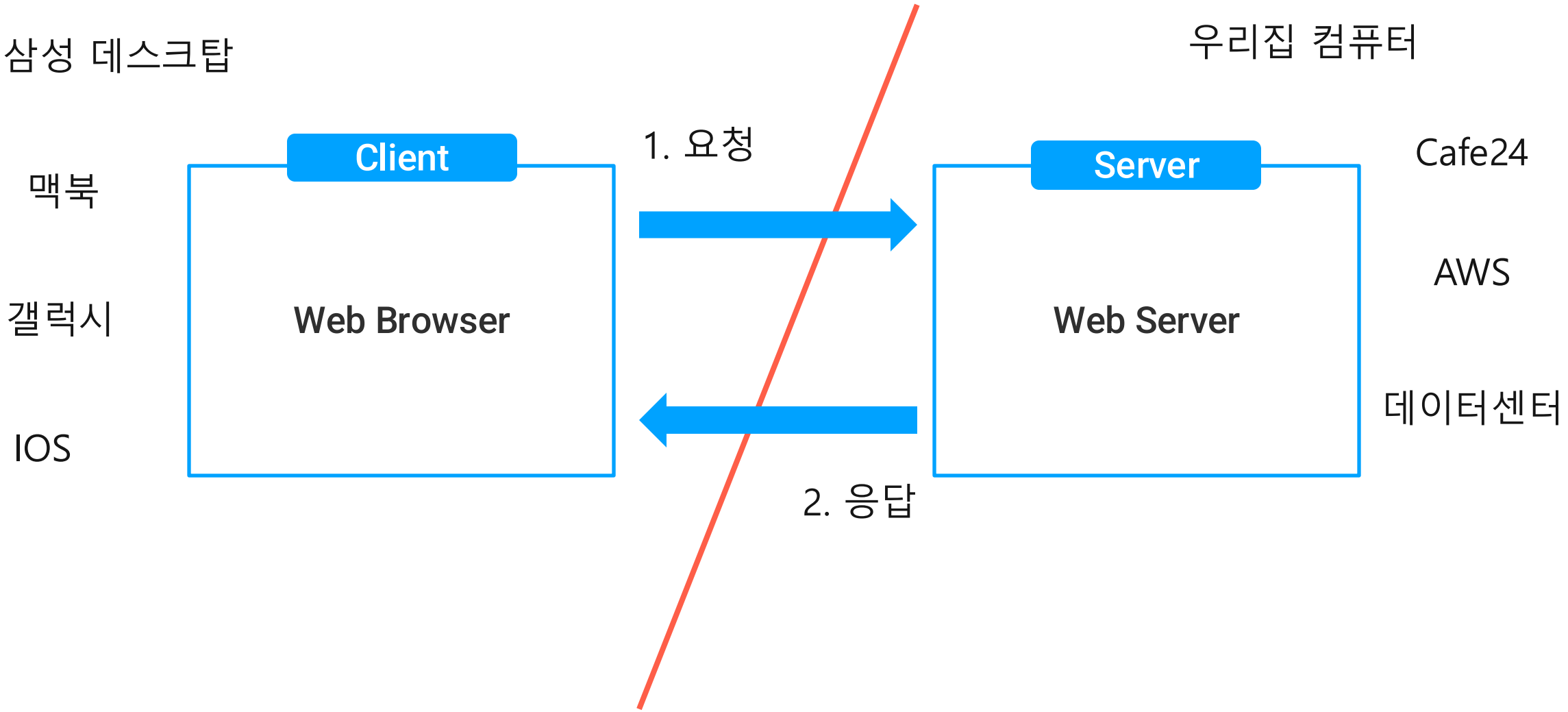
브라우저 : Chrome

에디터 : VS Code https://code.visualstudio.com/

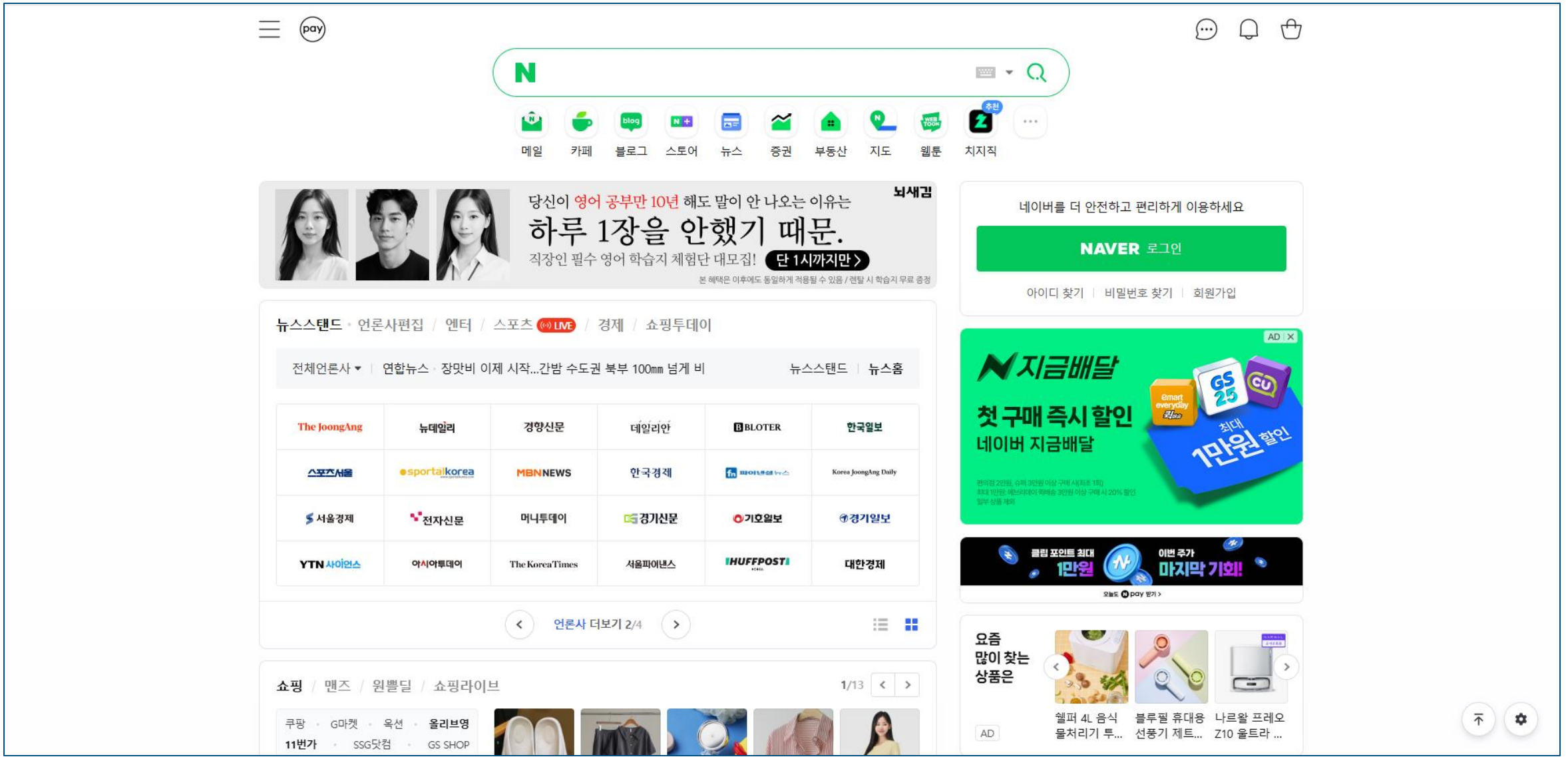
VS Code 익스텐션 : ESLint, Prettier, HTML CSS Support, HTML to CSS autocompletion, Auto Rename Tag, Auto Close Tag, htmltagwrap

반응형 웹 디자인

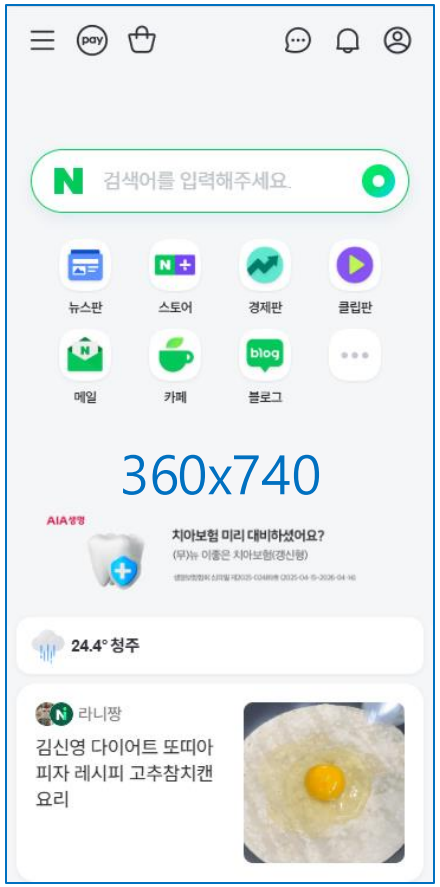
반응형 웹 디자인


삼성 데스크탑
맥북
갤럭시
IOS
Client
Web Browser
1. 요청
2. 응답
우리집 컴퓨터
Server
Web Server
Cafe24
AWS
데이터센터

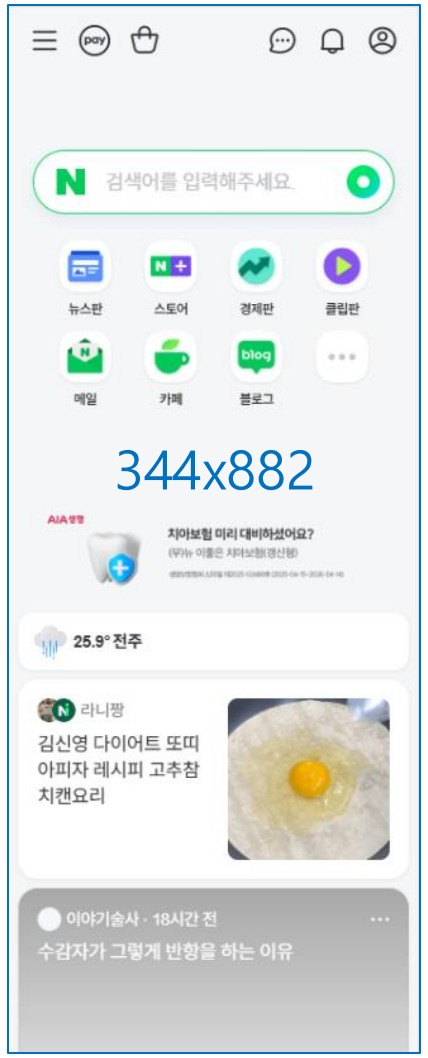
반응형 웹 디자인


pay
N
메일
카페
블로그
스토어
뉴스
증권
부동산
지도
웹툰
치지직
추천
뇌새김
당신이 영어 공부만 10년 해도 말이 안 나오는 이유는
하루 1장을 안했기 때문.
직장인 필수 영어 학습지 체험단 대모집!
단 1시까지만 >
본 혜택은 이후에도 동일하게 적용될 수 있음 / 렌탈 시 학습지 무료 증정
네이버를 더 안전하고 편리하게 이용하세요
NAVER 로그인
아이디 찾기 | 비밀번호 찾기 | 회원가입
뉴스스탠드 · 언론사편집 / 엔터 / 스포츠 LIVE / 경제 / 쇼핑투데이
전체언론사 | 연합뉴스 장맛비 이제 시작...간밤 수도권 북부 100㎜ 넘게 비
뉴스스탠드 | 뉴스홈
The JoongAng
뉴데일리
경향신문
데일리안
BLOTER
한국일보
스포츠서울
sportalkorea
MBNNEWS
한국경제
Korea JoongAng Daily
서울경제
전자신문
머니투데이
경기신문
기호일보
경기일보
YTN 사이언스
아시아투데이
The Korea Times
서울파이낸스
HUFFPOST
대한경제
언론사 더보기 2/4
쇼핑 / 맨즈 / 원쁠딜 / 쇼핑라이브
1/13
쿠팡 · G마켓 · 옥션 · 올리브영
11번가 · SSG닷컴 · GS SHOP
AD
지금배달
첫 구매 즉시 할인
네이버 지금배달
최대 1만원 할인
클립 포인트 최대 1만원
이번 주가 마지막 기회!
요즘 많이 찾는 상품은
AD
웰퍼 4L 음식 물처리기 투...
블루필 휴대용 선풍기 제트...
나르왈 프레오 Z10 울트라 ...

반응형 웹 디자인


검색어를 입력해주세요.
뉴스판
스토어
경제판
클립판
메일
카페
블로그
360x740
치아보험 미리 대비하셨어요?
24.4° 청주
라니짱
김신영 다이어트 또띠아피자 레시피 고추참치캔요리


검색어를 입력해주세요.
뉴스판
스토어
경제판
클립판
메일
카페
블로그
412x914
치아보험 미리 대비하셨어요?
25.9° 전주
라니짱
김신영 다이어트 또띠아피자 레시피 고추참치캔요리
이야기술사 · 18시간 전
수감자가 그렇게 반항을 하는 이유


검색어를 입력해주세요.
뉴스판
스토어
경제판
클립판
메일
카페
블로그
344x882
치아보험 미리 대비하셨어요?
25.9° 전주
라니짱
김신영 다이어트 또띠아피자 레시피 고추참치캔요리
이야기술사 · 18시간 전
수감자가 그렇게 반항을 하는 이유


검색어를 입력해주세요.
뉴스판
스토어
경제판
클립판
메일
카페
블로그
853x1280
치아보험 미리 대비하셨어요?
24.3° 대전
요즘 인기 콘텐츠
수감자가 그렇게 반항을 하는 이유
에이미는 수감자에게
라니짱 · 1일 전
김신영 다이어트 또띠아피자 레시피 고추참치캔요리
SK매직 AD
SK매직 정수기렌탈시 최대 12개월반값!

반응형 웹 디자인

다양한 기기(스마트폰, 태블릿, 데스크톱)와 화면 크기에 맞춰
웹 페이지의 레이아웃, 콘텐츠, 기능을 최적화하는 설계 접근법

핵심 구성 요소:

- 유연한 레이아웃: 상대 단위(%, vw, vh, rem, em) 사용
- 미디어 쿼리: 화면 크기나 기기 특성에 따라 스타일 조정
- 반응형 이미지: 기기 해상도에 맞는 이미지 제공

모든 사용자에게 일관되고 최적화된 경험 제공

반응형 웹 디자인 필요성

다양한 기기 환경:

- 2025년 기준, 글로벌 인터넷 트래픽의 70% 이상이 모바일 기기에서 발생
- 스마트폰, 태블릿, 스마트워치, 데스크톱 등 화면 크기와 해상도 다양화
- 단일 고정 레이아웃으로는 사용자 요구 충족 불가

사용자 경험(UX) 개선:

- 기기에 맞는 직관적이고 편리한 인터페이스 제공
- 작은 화면에서 과도한 스크롤/줌 제거, 큰 화면에서 공간 낭비 방지

반응형 웹 디자인 필요성

검색엔진 최적화(SEO):

- 구글은 모바일 친화적 웹사이트를 검색 순위 상위에 우선 배치
- 단일 URL로 모든 기기 지원, SEO 관리 효율성 증가

비즈니스 및 비용 효율성:

- 별도의 모바일 앱/사이트 개발 대신 단일 웹사이트로 모든 기기 대응
- 유지보수 및 업데이트 비용 절감

미래 대비:

- 새로운 기기(폴더블폰, AR/VR 기기) 등장에도 유연한 대응 가능

뷰포트와 해상도

디바이스	해상도	스크린 크기	뷰포트
iPhone 13 mini	1080x2340	5.4"	360x780
iPhone 13 Pro	1170x2352	5.4"	390x844
iPhone 13 Pro Max	1284x2778	6.7"	428x926
Huawei Mate 20 Lite	1080x2340	6.3"	360x780
Huawei Mate 20 Pro	1440x3120	6.39"	360x780
Galaxy Note 20	1080.2400	6.7"	412x915
Galaxy Note 20 Ultra	1440x3088	6.9"	412x883

뷰포트 Viewport

사용자가 웹 페이지를 볼 수 있는 브라우저의 가시 영역

- 뷰포트는 기기 화면 크기와 브라우저 창 크기에 따라 달라짐
- 반응형 웹 디자인에서 핵심적인 개념
- 웹 페이지의 레이아웃과 콘텐츠가 기기에 맞게 표시되도록 제어
- 모바일 기기에서 확대/축소 및 스크롤 동작 관리

해상도

화면의 픽셀 밀도를 나타내는 지표, 일반적으로 DPI(Dots Per Inch) 또는 PPI(Pixels Per Inch)로 표현

- CSS 픽셀: 논리적 픽셀, CSS 스타일링에 사용 (예: width: 100px)
- 기기 픽셀: 물리적 픽셀, 실제 화면의 픽셀 수 (예: Retina 디스플레이는 2x/3x 밀도)
- 고해상도 디스플레이(예: Retina, 4K)에서 선명한 이미지/텍스트 제공 필수
- 반응형 웹에서 적절한 이미지 크기와 스타일 제공
- DPR(Device Pixel Ratio):
  - 기기 픽셀과 CSS 픽셀의 비율 (예: Retina 디스플레이 DPR=2)
  - 예: DPR=2인 기기에서 100px(CSS)는 200x200 물리적 픽셀로 렌더링

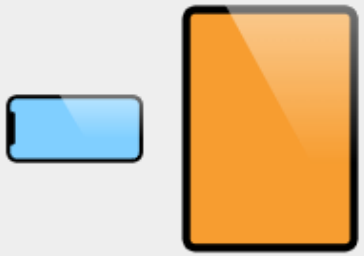
뷰포트와 해상도

mediaquri.es 에서는 320px(스마트폰), 768px(태블릿), 1024px(넷북), 1600px(데스크탑)로 경계치를 정하고 있습니다

Viewport breakpoint

0~480

스마트폰



481~768

스마트폰(가로모드, 큰 사이즈)
& 테블릿 PC

769~1279

테블릿 PC(가로모드, 큰 사이즈)
& 노트북
& 작은 해상도의 모니터

1280+

일반 해상도(13인치/가로 1600)
이상의 모니터

미디어 쿼리

미디어 쿼리

CSS의 @media 규칙을 사용해 화면 크기, 기기 특성, 해상도 등 특정 조건에 따라 스타일을 적용하는 기술
반응형 웹 디자인의 핵심으로, 다양한 기기(모바일, 태블릿, 데스크톱)와 환경에 맞는 스타일링 제공

목적:

- 사용자 경험(UX) 최적화: 기기에 맞는 레이아웃 제공
- 접근성 강화: 다양한 화면 크기와 환경 지원
- 성능 개선: 조건에 따라 최적화된 리소스 로드

미디어 쿼리

기본 문법

```
@media [미디어 유형] [and/or/not] (미디어 조건) {
    /* 스타일 규칙 */
}
```

구성 요소:

- 미디어 유형: all, screen, print, speech 등
- all: 모든 기기 (기본값)
- screen: 화면 표시 기기 (모니터, 스마트폰 등)
- print: 인쇄용 스타일

미디어 조건:

- min-width: 화면 너비가 지정 값 이상
- max-width: 화면 너비가 지정 값 이하
- orientation: portrait(세로), landscape(가로)
- min-resolution: 해상도 조건

논리 연산자: and, or, not으로 조건 결합

미디어 쿼리

기본 문법

```
@media [미디어 유형] [and/or/not] (미디어 조건) {
    /* 스타일 규칙 */
}
```

예시

```
@media screen and (min-width: 768px) {
    .container {
        width: 50%;
        font-size: 1.2rem;
    }
}
```

미디어 쿼리 - 화면크기

767px

1024px

```
@media (max-width: 767px) {
    .box {
        width: 33.33%;
    }
}
```

```
@media (min-width: 768px) and (max-width: 1023px){
    .box {
        width: 33.33%;
    }
}
```

```
@media (min-width: 1024px) {
    .box {
        width: 33.33%;
    }
}
```

max-width:
화면 너비가 특정 값 이하일 때

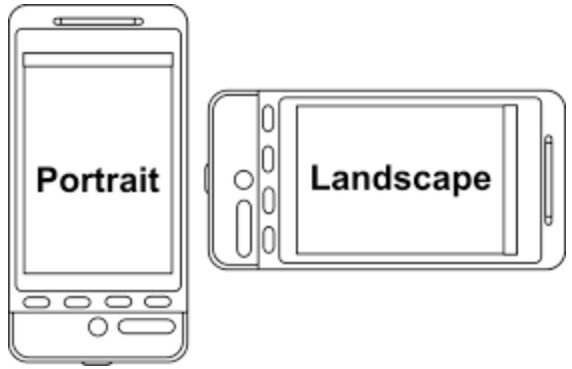
범위 지정

min-width:
화면 너비가 특정 값 이상일 때

미디어 쿼리 – 기기방향

Orientation: 화면의 가로/세로모드

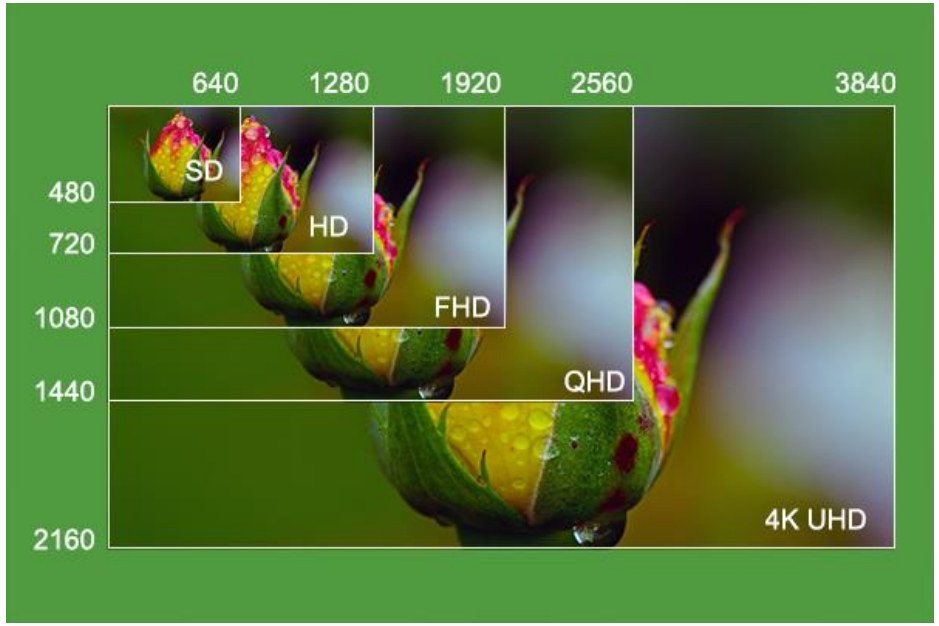
```
@media (orientation: landscape) {
    .container {
        flex-direction: row;
    }
}
```



미디어 쿼리 – 해상도

min-resolution, max-resolution: 고해상도 디스플레이 지원

```
@media (min-resolution: 2dppx) {
    .image {
        background-image: url('high-res.jpg');
    }
}
```



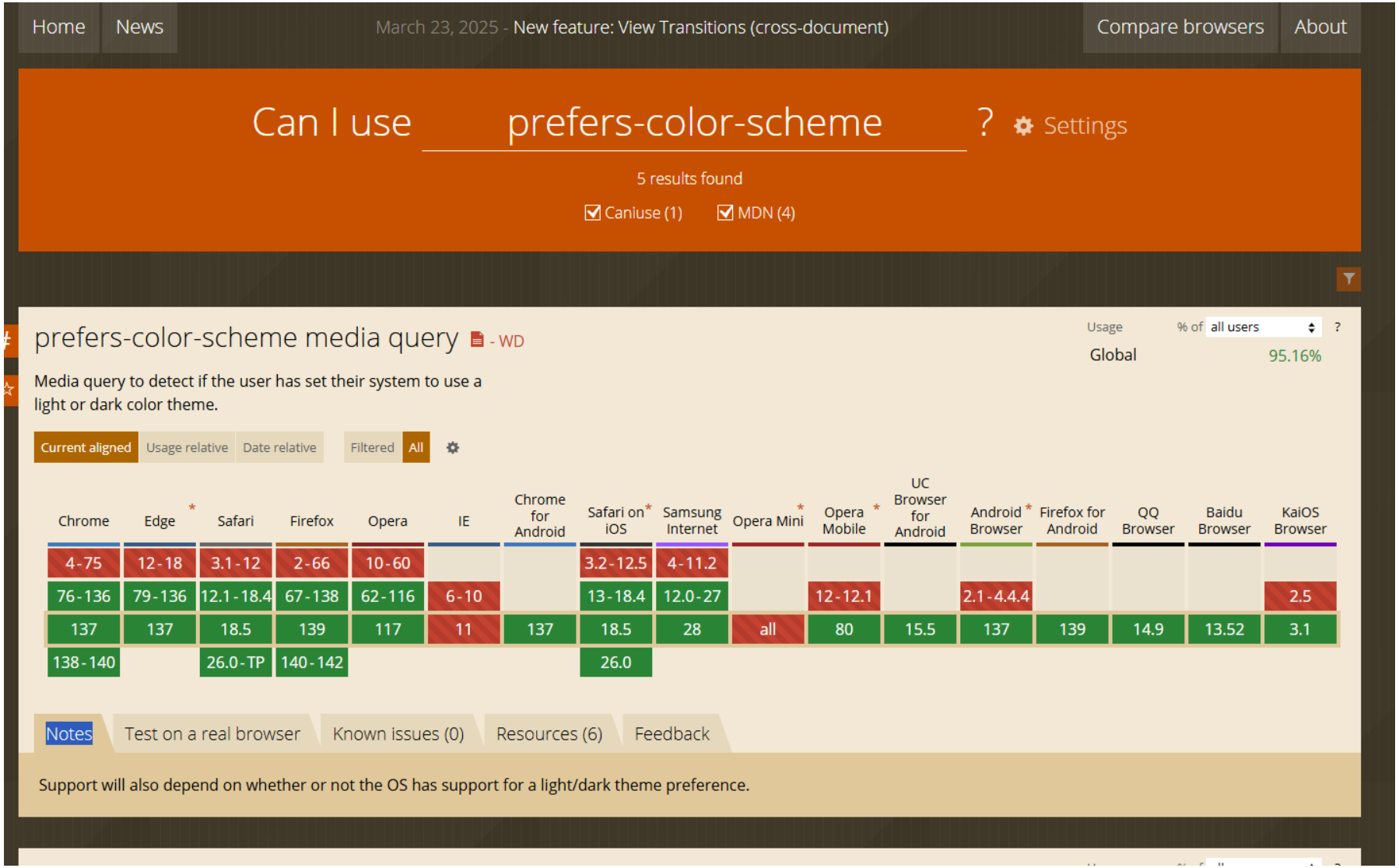
미디어 쿼리 – 다크/라이트 모드

prefers-color-scheme: 다크/라이트 모드 지원

```
@media (prefers-color-scheme: dark) {
    body {
        background-color: #333;
        color: #fff;
    }
}
```

미디어 쿼리

Can I use...?



미디어 쿼리

day05/media

index.html

```html
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width" />
    <title>미디어 쿼리 예시</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <header>
      <h1>반응형 웹사이트</h1>
    </header>
    <main>
      <div class="card">카드 1</div>
      <div class="card">카드 2</div>
      <div class="card">카드 3</div>
    </main>
  </body>
</html>
```

styles.css

```css
body {
  margin: 0;
  font-family: Arial, sans-serif;
}
header {
  text-align: center;
  padding: 32px;
  background-color: lightblue;
}
main {
  padding: 16px;
  display: flex;
  flex-wrap: wrap;
  gap: 16px;
}
.card {
  width: 100%;
  padding: 16px;
  margin-bottom: 16px;
  background-color: lightcoral;
  text-align: center;
  box-sizing: border-box;
}
```

styles.css

```css
/* 태블릿: 768px 이상 */
@media screen and (min-width: 768px) {
  .card {
    width: calc(50% - 8px);
  }
}
/* 데스크톱: 1024px 이상 */
@media screen and (min-width: 1024px) {
  .card {
    width: calc(33.33% - 20px);
  }
}
/* 고해상도 디스플레이 */
@media (min-resolution: 2dppx) {
  .card {
    background-image: url("high-res-bg.jpg");
    background-size: cover;
  }
}
```

모바일퍼스트 vs 데스크톱퍼스트

상대 단위

CSS에서 요소의 크기나 간격을 상대적인 기준(부모, 뷰포트, 폰트 크기 등)에 따라 설정하는 단위

역할: 반응형 웹 디자인에서 유연하고 동적인 레이아웃 구현, 다양한 기기와 화면 크기에 적응

주요 상대 단위: %, vw, vh, em, rem

장점:

- 고정 단위(px)와 달리, 화면 크기나 폰트 크기 변화에 따라 자동 조정
- 접근성과 유지보수성 향상

상대 단위 - %

부모 요소의 크기(너비, 높이 등)에 비례한 단위

특징:

- 주로 width, height, margin, padding에 사용
- 반응형 레이아웃에서 컨테이너 내 요소 배치에 유용

활용 사례:

- 부모 컨테이너에 맞춘 유연한 너비 설정
- 그리드 시스템에서 비율 기반 레이아웃

```
.parent {
  width: 500px;
  height: 200px;
  background-color: lightgray;
}
.child {
  width: 50%; /* 부모 너비의 50% = 250px */
  height: 50%; /* 부모 높이의 50% = 100px */
  background-color: lightcoral;
}
```

상대 단위 – vw(Viewport Width)

뷰포트 너비의 1% (1vw = 뷰포트 너비의 1/100)

특징:

- 뷰포트 크기에 따라 동적으로 조정
- 스크롤바 포함 여부는 브라우저마다 다를 수 있음
- 고정 단위(px)와 달리 화면 크기에 비례

활용 사례:

- 반응형 타이포그래피(폰트 크기 조정)

```
.hero {
  width: 100vw; /* 뷰포트 너비 전체 */
  height: 50vw; /* 뷰포트 너비의 50% */
  background-color: lightblue;
}
```

상대 단위 – vh(Viewport Height)

뷰포트 높이의 1% (1vh = 뷰포트 높이의 1/100)

특징:

- 뷰포트 높이에 따라 동적으로 조정
- 주소창/네비게이션 바 포함 여부에 따라 동작 상이

활용 사례:

- 풀스크린 섹션(예: 랜딩 페이지)
- 세로 중앙 정렬 레이아웃

```
.section {
  height: 100vh; /* 뷰포트 높이 전체 */
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: lightgreen;
}
```

상대 단위 – em

현재 요소의 폰트 크기(font-size)에 비례한 단위

특징:

- 부모 요소의 font-size를 상속받거나, 자체 font-size 기준
- 복합적인 상속 구조에서 크기 계산 복잡할 수 있음

활용 사례:

- 텍스트 크기와 관련된 상대적 스타일링
- 버튼, 여백 등 폰트 크기에 비례한 디자인

```
.parent {
  font-size: 16px;
}
.child {
  font-size: 1.5em; /* 16px * 1.5 = 24px */
  padding: 0.5em; /* 24px * 0.5 = 12px */
  background-color: lightcoral;
}
```

상대 단위 – rem

루트 요소(html)의 폰트 크기(font-size)에 비례한 단위

특징:

- 모든 요소가 동일한 기준(html의 font-size)을 참조
- 상속 구조와 무관, 일관된 크기 관리 가능

활용 사례:

- 전역적으로 일관된 스타일링(텍스트, 여백 등)
- 반응형 타이포그래피와 레이아웃

```
html {
  font-size: 16px;
}
.box {
  font-size: 1.5rem; /* 16px * 1.5 = 24px */
  margin: 1rem; /* 16px * 1 = 16px */
  background-color: lightblue;
}
```

모바일 퍼스트와 데스크톱 퍼스트

모바일 퍼스트:

- 기본 스타일을 모바일(작은 화면)에 맞춰 작성
- 미디어 쿼리(min-width)를 사용해 더 큰 화면에 스타일을 추가

데스크톱 퍼스트:

- 기본 스타일을 데스크톱(큰 화면)에 맞춰 작성
- 미디어 쿼리(max-width)를 사용해 작은 화면에 스타일을 조정

모바일 퍼스트

- 기본 스타일은 모바일(작은 화면)에 최적화
- 미디어 쿼리(min-width)로 점진적으로 태블릿, 데스크톱 스타일 추가
- 간결한 스타일부터 시작, 복잡도 점차 증가

적합 상황:

- 모바일 트래픽 비율이 높은 웹사이트
- 새로운 프로젝트 또는 경량화된 웹 앱
- 모바일 중심 사이트(전자상거래, 소셜 미디어)

데스크톱 퍼스트

- 기본 스타일은 데스크톱(큰 화면)에 최적화
- 미디어 쿼리(max-width)로 작은 화면 스타일 조정
- 복잡한 레이아웃부터 시작, 점차 단순화

적합 상황:

- 데스크톱 사용자 비율이 높은 웹사이트(예: 기업용 소프트웨어)
- 복잡한 UI/데이터 시각화 중심 프로젝트
- 데스크톱 중심 사이트(기업용, 관리자 대시보드)

미디어 쿼리

styles-mobile-first.css

styles-mobile-first.css

day05 > meida > styles-mobile-first.css > {} @media

```css
body {
  margin: 0;
  font-family: Arial, sans-serif;
}
header {
  text-align: center;
  padding: 1rem;
  background-color: lightblue;
}
main {
  padding: 1rem;
}
.card {
  width: 100%;
  padding: 1rem;
  margin-bottom: 1rem;
  background-color: lightcoral;
  text-align: center;
  box-sizing: border-box;
}
```

styles-mobile-first.css

day05 > meida > styles-mobile-first.css > ...

```css
/* 태블릿 */
@media (min-width: 768px) {
  main {
    display: flex;
    flex-wrap: wrap;
    gap: 1rem;
  }
  .card {
    width: calc(50% - 0.5rem);
  }
}
/* 데스크톱 */
@media (min-width: 1024px) {
  .card {
    width: calc(33.33% - 0.67rem);
  }
}
```

styles-desktop-first.css

styles-desktop-first.css

day05 > meida > styles-desktop-first.css > ...

```css
body {
  margin: 0;
  font-family: Arial, sans-serif;
}
header {
  text-align: center;
  padding: 2rem;
  background-color: lightblue;
}
main {
  padding: 1rem;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 1rem;
}
.card {
  padding: 1rem;
  background-color: lightcoral;
  text-align: center;
  box-sizing: border-box;
}
```

styles-desktop-first.css

day05 > meida > styles-desktop-first.css > .card

```css
/* 태블릿 */
@media (max-width: 1023px) {
  main {
    grid-template-columns: repeat(2, 1fr);
  }
}
/* 모바일 */
@media (max-width: 767px) {
  main {
    grid-template-columns: 1fr;
  }
  .card {
    margin-bottom: 1rem;
  }
}
```

오늘 우리는

반응형 웹 디자인

다양한 기기(스마트폰, 태블릿, 데스크톱)와 화면 크기에 맞춰
웹 페이지의 레이아웃, 콘텐츠, 기능을 최적화하는 설계 접근법

핵심 구성 요소:

- 유연한 레이아웃: 상대 단위(%, vw, vh, rem, em) 사용
- 미디어 쿼리: 화면 크기나 기기 특성에 따라 스타일 조정
- 반응형 이미지: 기기 해상도에 맞는 이미지 제공

모든 사용자에게 일관되고 최적화된 경험 제공

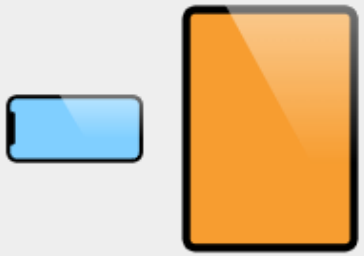
뷰포트와 해상도

mediaquri.es 에서는 320px(스마트폰), 768px(태블릿), 1024px(넷북), 1600px(데스크탑)로 경계치를 정하고 있습니다

Viewport breakpoint

0~480

스마트폰



481~768

스마트폰(가로모드, 큰 사이즈)
& 테블릿 PC

769~1279

테블릿 PC(가로모드, 큰 사이즈)
& 노트북
& 작은 해상도의 모니터

1280+

일반 해상도(13인치/가로 1600)
이상의 모니터

미디어 쿼리

CSS의 @media 규칙을 사용해 화면 크기, 기기 특성, 해상도 등 특정 조건에 따라 스타일을 적용하는 기술
반응형 웹 디자인의 핵심으로, 다양한 기기(모바일, 태블릿, 데스크톱)와 환경에 맞는 스타일링 제공

목적:

- 사용자 경험(UX) 최적화: 기기에 맞는 레이아웃 제공
- 접근성 강화: 다양한 화면 크기와 환경 지원
- 성능 개선: 조건에 따라 최적화된 리소스 로드

상대 단위

CSS에서 요소의 크기나 간격을 상대적인 기준(부모, 뷰포트, 폰트 크기 등)에 따라 설정하는 단위

역할: 반응형 웹 디자인에서 유연하고 동적인 레이아웃 구현, 다양한 기기와 화면 크기에 적응

주요 상대 단위: %, vw, vh, em, rem

장점:

- 고정 단위(px)와 달리, 화면 크기나 폰트 크기 변화에 따라 자동 조정
- 접근성과 유지보수성 향상

모바일 퍼스트와 데스크톱 퍼스트

모바일 퍼스트:

- 기본 스타일을 모바일(작은 화면)에 맞춰 작성
- 미디어 쿼리(min-width)를 사용해 더 큰 화면에 스타일을 추가

데스크톱 퍼스트:

- 기본 스타일을 데스크톱(큰 화면)에 맞춰 작성
- 미디어 쿼리(max-width)를 사용해 작은 화면에 스타일을 조정

감사합니다