

ch02_2.Responsive Web Design

디지털마케팅 프로그래밍

by Wooyoung Choi, KMU Digital Marketing MBA, 2025

Responsive vs Adaptive

Fixed-width Design

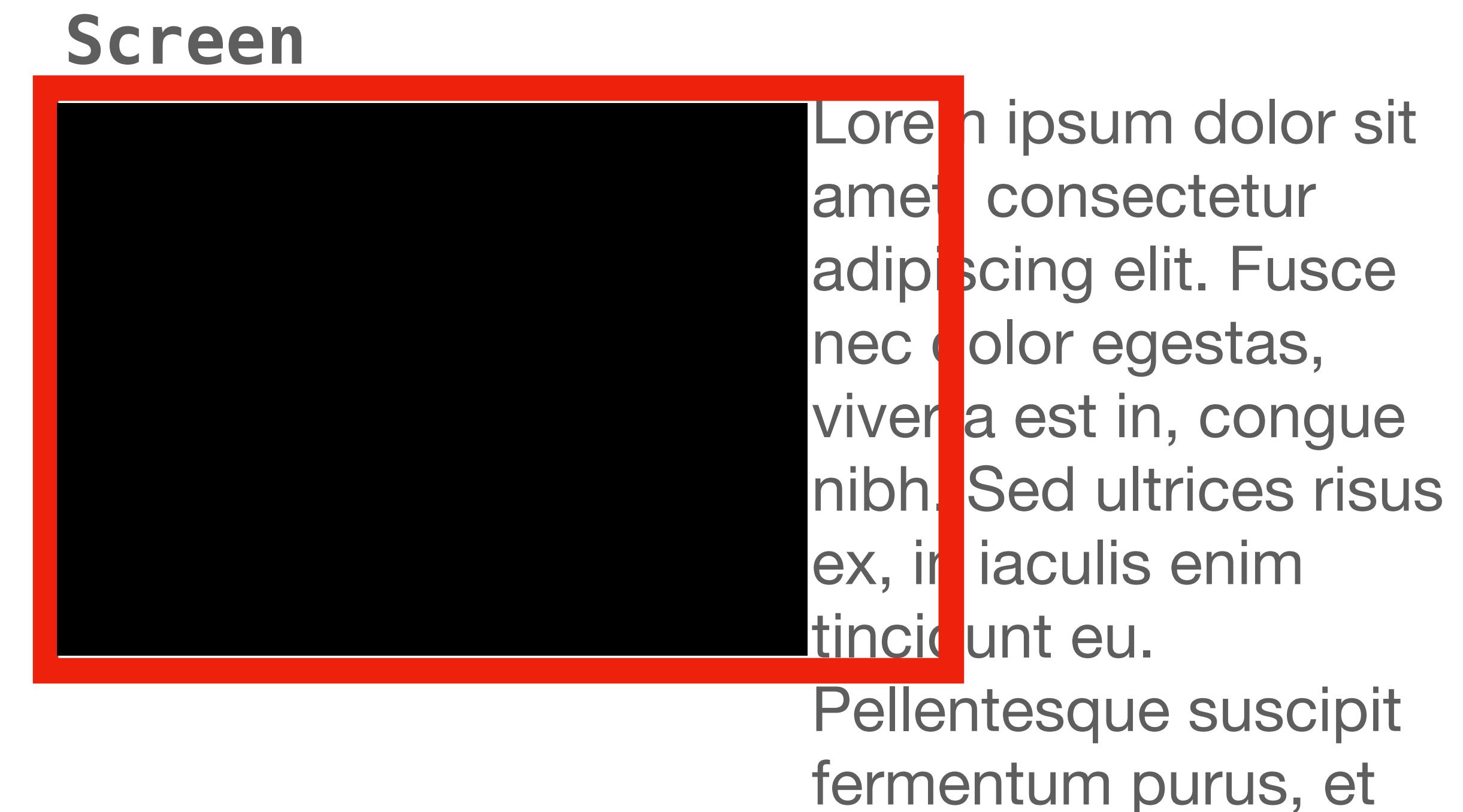
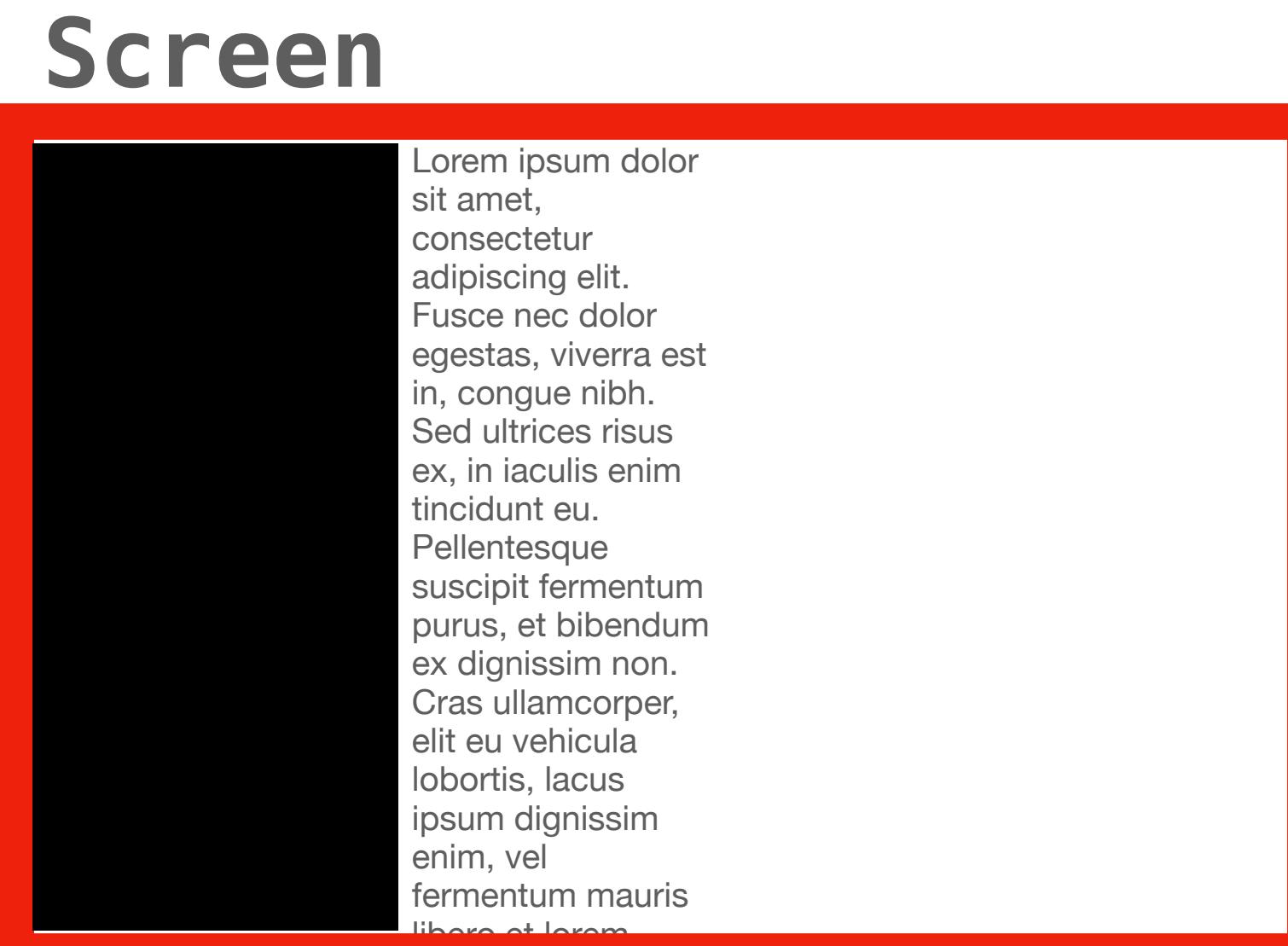
- 고정된 폭으로 웹페이지를 표현하는 방식(1990s~)
- 사용자의 기기 해상도와 관련없이 페이지가 정의한 폭으로 페이지를 표현
- 정의된 폭보다 작은 해상도의 기기에서는 페이지가 제대로 표출되지 않음

Screen

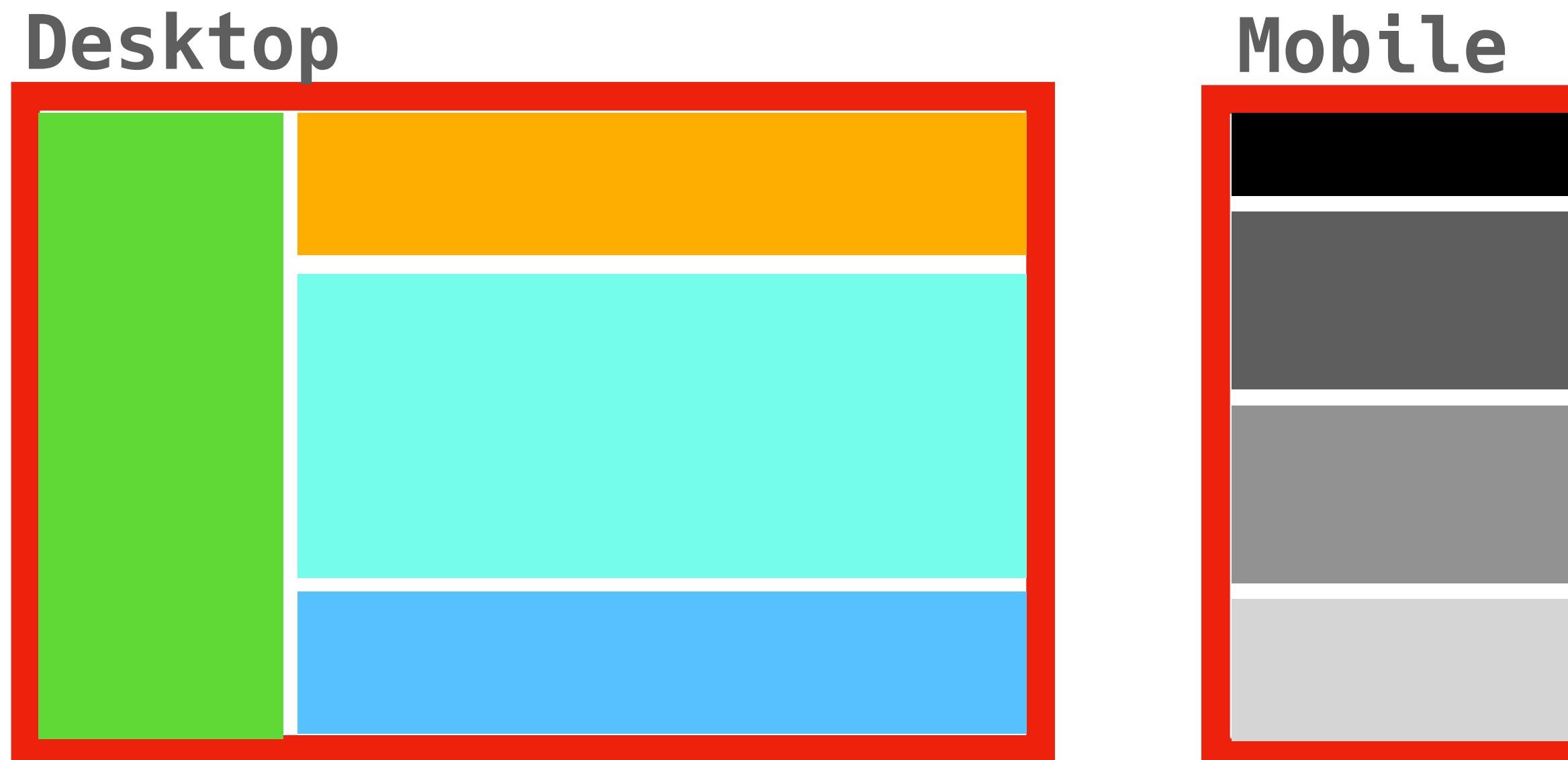
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce nec dolor egestas, viverra est in, congue nibh. Sed ultrices risus ex, in iaculis enim tincidunt. Pellentesque suscipit fermentum purus, et bibendum ex dignissim non. Cras ullamcorper, elit eu vehicula lobortis, lacus ipsum dignissim enim, vel fermentum mauris libero et lorem. Fusce eu lorem non ipsum posuere imperdiet nec ac velit. Morbi pretium pulvinar risus. Mauris tempus augue leo, quis pharetra diam sollicitudin et. Vestibulum elementum fringilla quam quis ullamcorper. Ut ullamcorper ultricies convallis. Aenean viverra mattis nisl, quis lacinia dui dapibus vitae. Praesent et condimentum ante, eu convallis purus. Praesent purus sapien, tempor et pharetra in, sagittis vel odio. Nunc vitae dolor lobortis, gravida dui eu, consectetur mauris. Maecenas eget odio augue. Pellentesque habitant morbi tristique semper, netus et malesuada fames ac turpis egestas.

Liquid Layouts

- min-width, max-width 등을 활용하여 가변적인 레이아웃을 표현
- 낭비되는 영역이 너무 많음 or 레이아웃이 제대로 표현되지 않을 수 있음
- user-agent를 활용하여 디바이스 크기별 다른 리소스 제공



Adaptive Web

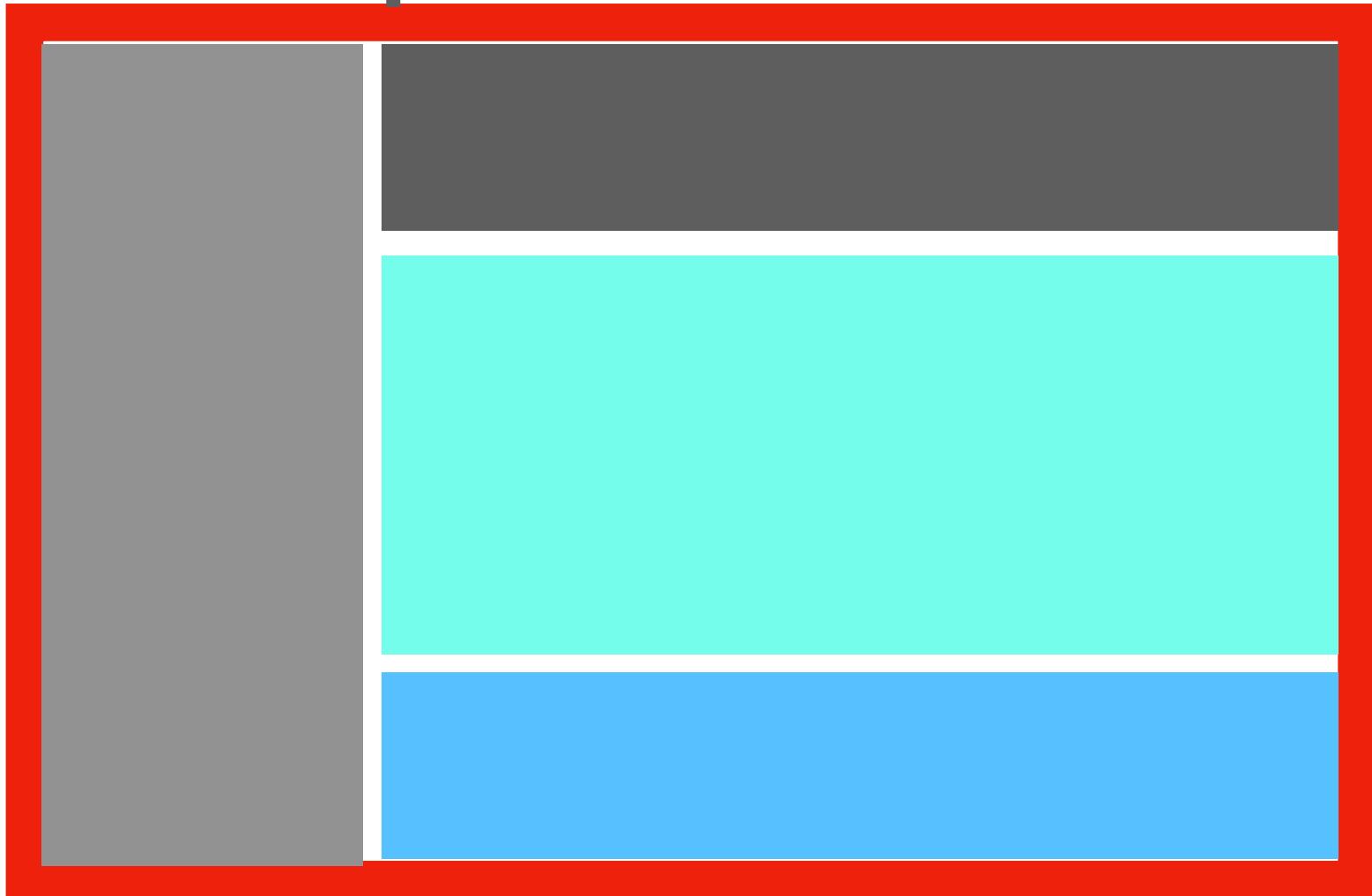


Adaptive Web

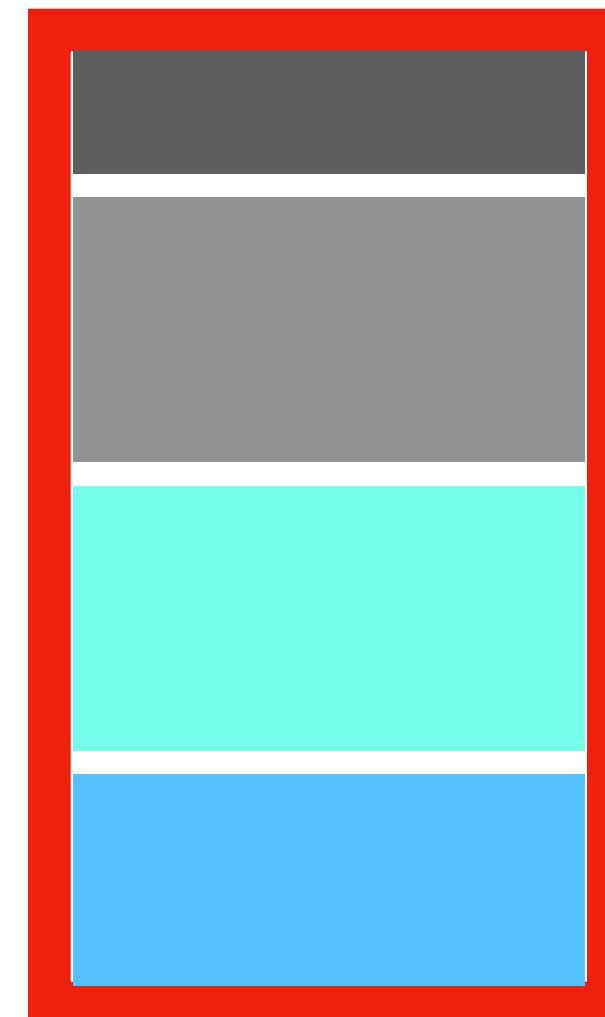
- 사용자가 접속한 디바이스에 따라 다른 페이지를 대응하여 보여주는 방식
- ‘Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement’
 - by Aaron Gustofson(2011)
- ex) <https://www.naver.com/> <https://m.naver.com/>
- 모든 디바이스에 대응하기 위한 코드를 받을 필요가 없어 초기 로딩속도가 빠름
- 기기 별 리소스가 별도로 존재하여 개발 및 유지보수 비용 증가
- 반응형에 비해 SEO에 불리
 - 기기별 일관성이 떨어지는 데이터
 - 너무 많은 URL

Responsive Web

Desktop



Mobile



Responsive Web

- 웹의 해상도, 레이아웃 등이 디바이스에 따라 반응하여 유동적으로 변환되는 방식
- A List Apart by Ethan Marcotte(2010)
 - Fluid grid, Fluid media, Media queries
- 하나의 리소스를 사용자의 디바이스 크기에 따라 다르게 표현
- Modern Responsive web
 - Media Feature: 사용자 설정에 접근
 - Container Query: component 자체 반응형 정보를 소유
 - picture: 기기 비율별 미디어 리소스의 전개방법 다양화

tailwindCSS

Framework

- 원하는 기능 구현에 집중하여 개발할 수 있도록 일정한 형태와 기능을 갖추고 있는 골격, 뼈대
- 개발을 위해 필요한 뼈대를 제공하고 이러한 뼈대 위에서 사용자가 코드를 작성하는 형태
- 사용자는 프레임워크가 정해준 방식으로 클래스와 메소드를 구현
- 문제 해결을 위해 상호협력하는 클래스와 인터페이스의 집합

Library

- 단순 활용 가능한 도구들의 집합
- 프로그램이 사용하는 비휘발성 자원의 모임
- 사용자가 제어흐름에 주도성을 가짐

tailwindCSS

- 오픈소스 CSS 프레임워크
- Bootstrap(by X)과 다르게 미리 정의된 클래스와 요소가 존재하지 않음
- 미리 세팅된 유ти리티 클래스를 활용하여 HTML 코드 내에서 스타일링 가능
- 편리함과 빠른개발, 일관된 디자인 가능
- 단, HTML과 CSS가 혼재

tailwindCSS 적용하는 법

CLI, PostCSS, CDN 등 다양한 방식 적용 가능

Play CDN

```
<head>  
  <script src="https://cdn.tailwindcss.com"></script>  
</head>  
  
<body>  
  <h1 class="text-3xl font-bold underline">Hello</h1>  
</body>
```

header

```
<header class="bg-gray-800 text-white py-4">  
  <div class="container mx-auto flex justify-between">  
    <h1 class="text-2xl font-bold">TailwindCSS</h1>  
  
    <nav>  
      <ul class="flex space-x-4">  
        <li><a href="#" class="hover:bg-gray-700">Home</a></li>  
        <li><a href="#" class="hover:bg-gray-700">Blog</a></li>  
        <li><a href="#" class="hover:bg-gray-700">About</a></li>  
      </ul>  
    </nav>  
  </div>  
</header>
```

header

```
<header class="bg-gray-800 text-white  
py-4">  
  
  <div class="container mx-auto flex  
justify-between">  
  
    <h1 class="text-2xl font-  
bold">TailwindCSS</h1>  
  
    <nav>  
  
      <ul class="flex space-x-4">  
  
        <li><a href="#"  
class="hover:bg-gray-700">Home</a></li>  
  
        <li><a href="#"  
class="hover:bg-gray-700">Blog</a></li>  
  
        <li><a href="#"  
class="hover:bg-gray-700">About</a></  
li>  
  
      </ul>  
  
    </nav>  
  
  </div>  
</header>
```

bg-gray-800: 배경색 설정

text-white: 글꼴 색 설정

py-4: 내부간격(vertical) 4px

container: max-width 설정

mx-auto: 외부간격(horizontal) 자동, center

flex: flex 적용

justify-between: 아이템 배치 방법 justify

text-2xl: 글꼴 크기 설정

font-bold: 글꼴 굵기 설정

space-x-4: 요소 간 간격

main(1)

```
<main class="container mx-auto py-8">

  <section>
    <h2 class="text-2xl font-bold pb-4">Lorem ipsum</h2>

    <p class="text-gray-600">Lorem ipsum dolor sit amet,
    consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua. Ut enim ad
    minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat. Duis aute irure
    dolor in reprehenderit in voluptate velit esse cillum
    dolore eu fugiat nulla pariatur. Excepteur sint occaecat
    cupidatat non proident, sunt in culpa qui officia deserunt
    mollit anim id est laborum.</p>

  </section>
```

pb-4: padding bottom 4px

main(2)

```
<section class="flex flex-wrap justify-between mt-4">  
  <article class="w-full md:w-1/2 p-4 bg-gray-200 rounded-md">  
    <h3 class="text-lg font-bold pb-2">섹션 1</h3>  
    <p class="text-gray-600">섹션 1 내용을 작성합니다.</p>  
  </article>  
  <article class="w-full md:w-1/2 p-4 bg-gray-200 rounded-md">  
    <h3 class="text-lg font-bold pb-2">섹션 2</h3>  
    <p class="text-gray-600">섹션 2 내용을 작성합니다.</p>  
  </article>  
</section>  
</main>
```

main(2)

```
<section class="flex flex-wrap justify-between mt-4">  
    <article class="w-full md:w-1/2 p-4 bg-gray-200 rounded-md">  
        <h3 class="text-lg font-bold pb-2">섹션 1</h3>  
        <p class="text-gray-600">섹션 1 내용을 작성합니다.</p>  
    </article>  
    <article class="w-full md:w-1/2 p-4 bg-gray-200 rounded-md">  
        <h3 class="text-lg font-bold pb-2">섹션 2</h3>  
        <p class="text-gray-600">섹션 2 내용을 작성합니다.</p>  
    </article>  
</section>  
</main>
```

flex-wrap: 너비가 충분하지 않을 경우 다음줄에 나열

mt-4: 외부간격 (top) 4px

w-full: width:100%

md:w-1/2: breakpoint(md)에서 width:50%

p-4: padding: 4px;

rounded-md: 모서리 모양

footer

```
<footer class="bg-gray-800 text-white py-4">  
  <div class="container mx-auto text-center">  
    <p>&copy; 2024 tailwind</p> text-center: 텍스트 배열(가운데)  
  </div>  
</footer>
```