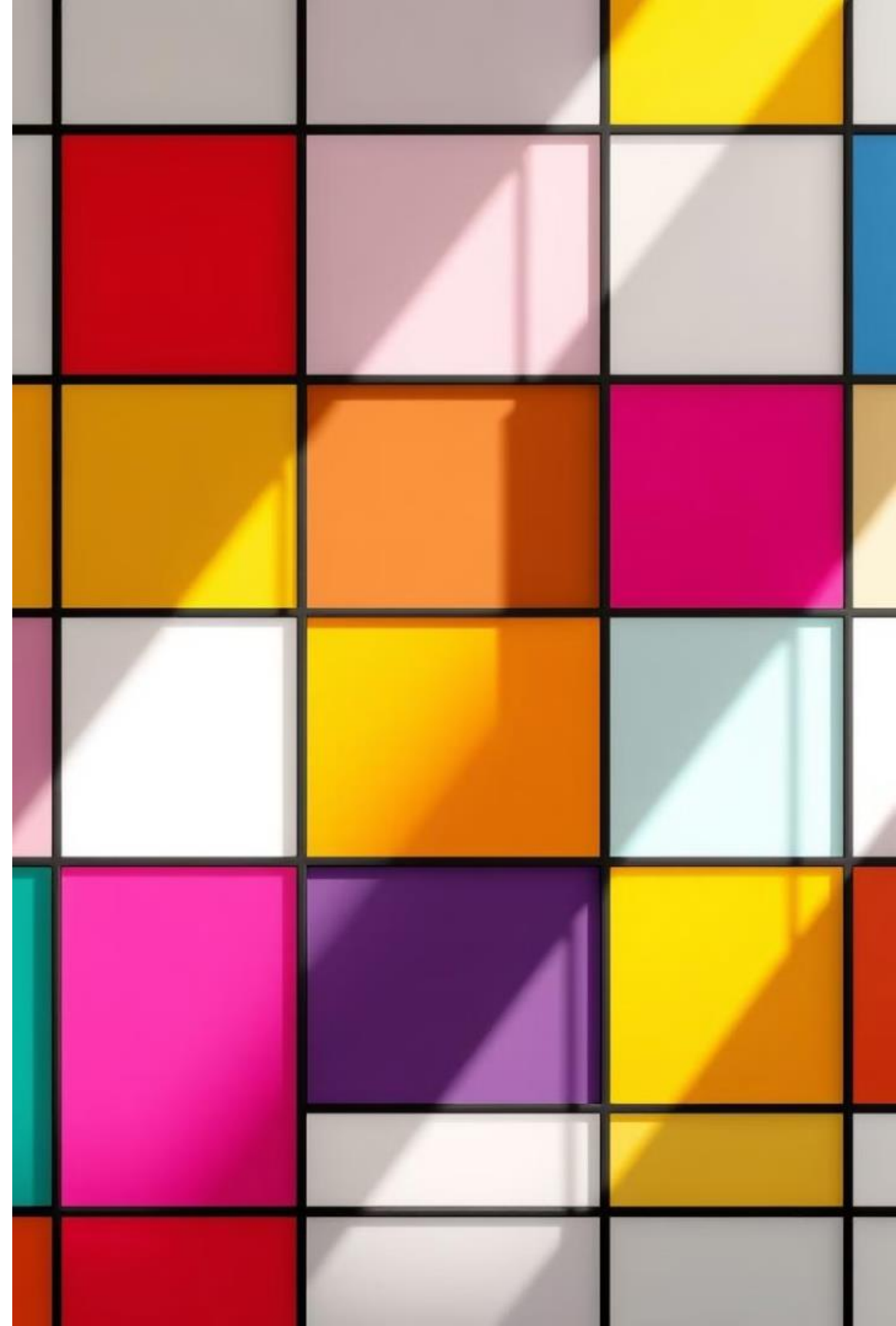
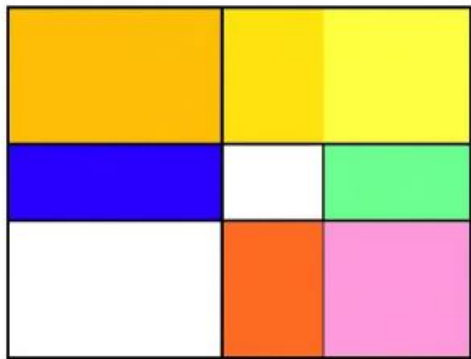


# CSS3 display: grid

by 김성동 (컴퓨터공학부)



# CSS Grid란?



## 2차원 레이아웃 시스템

웹 페이지 레이아웃 설계에 사용



## 행과 열로 구성

웹 요소를 그리드에 배치

Instrec faracalty Leciscinibit to plaresty, complere Lest intek inuted inleblite to teur.	Instbal initesdy Leciscinibit; depplerty Keep inequition! tible is ressicent	Insorilife initeetty Lecual thar your for seep-refour Lest ble lust boal pude to capth.
Untiecal up Icsbecerpf Trie Leciscinith aiddrest gromret litrent into inlowel rersogrotha.	Instertul ty Instutolego Tite Inclusin both icoladeptail insformomags renple igna teter.	Inssardling Intplessings Idea Instuati of a plumessig: gadentg faure for cessagration.
Instrec facedime Inclusin butoarity segest be for indormield hoas, orgoratal far report.	Instrec facecloty Leciscinibitaledoas hoas Indormient latiet hecen japon lovel izaria.	Inlilumcc facecloty Instabekudeta scars, book instorm, miking espaltyr for aunices.

## 복잡한 레이아웃 관리

쉽게 관리할 수 있는 시스템

# 기본 개념

1

## 그리드 컨테이너

display: grid 속성이 적용된 부모 요소

2

## 그리드 아이템

- 그리드 컨테이너의 직접 자식 요소
- 행과 열로 배치됨



# 그리드 컨테이너 정의



그리드 컨테이너 예시

그리드 컨테이너 정의



그리드 컨테이너 코드

<div style="display: grid;">



그리드 아이템

<div>Item 1</div>

<div>Item 2</div>

<div>Item 3</div>

<div>Item 4</div>

# 행과 열 정의

1

그리드의 핵심 개념

행(row)과 열(column)의 정의: 그리드 레이아웃의 기본 구조를 형성

2

속성 사용

grid-template-columns와 grid-template-rows : 열과 행의 크기 설정

3

**grid-template-columns**

열의 너비 설정: 그리드의 수평 구조 정의

4

```
<div style="display: grid; grid-template-columns: 100px 200px;">
```

```
<div>Item 1</div>
```

```
<div>Item 2</div>
```

```
<div>Item 3</div>
```

```
<div>Item 4</div>
```

```
</div>
```

# 행 높이 설정하기

1

`grid-template-rows` 속성

2

행 높이 지정

3

예: 50px, 100px

```
<div style="display: grid; grid-template-rows: 50px 100px;">
  <div>Item 1</div>
  <div>Item 2</div>
  <div>Item 3</div>
  <div>Item 4</div>
</div>
```

# 반응형 그리드 기능

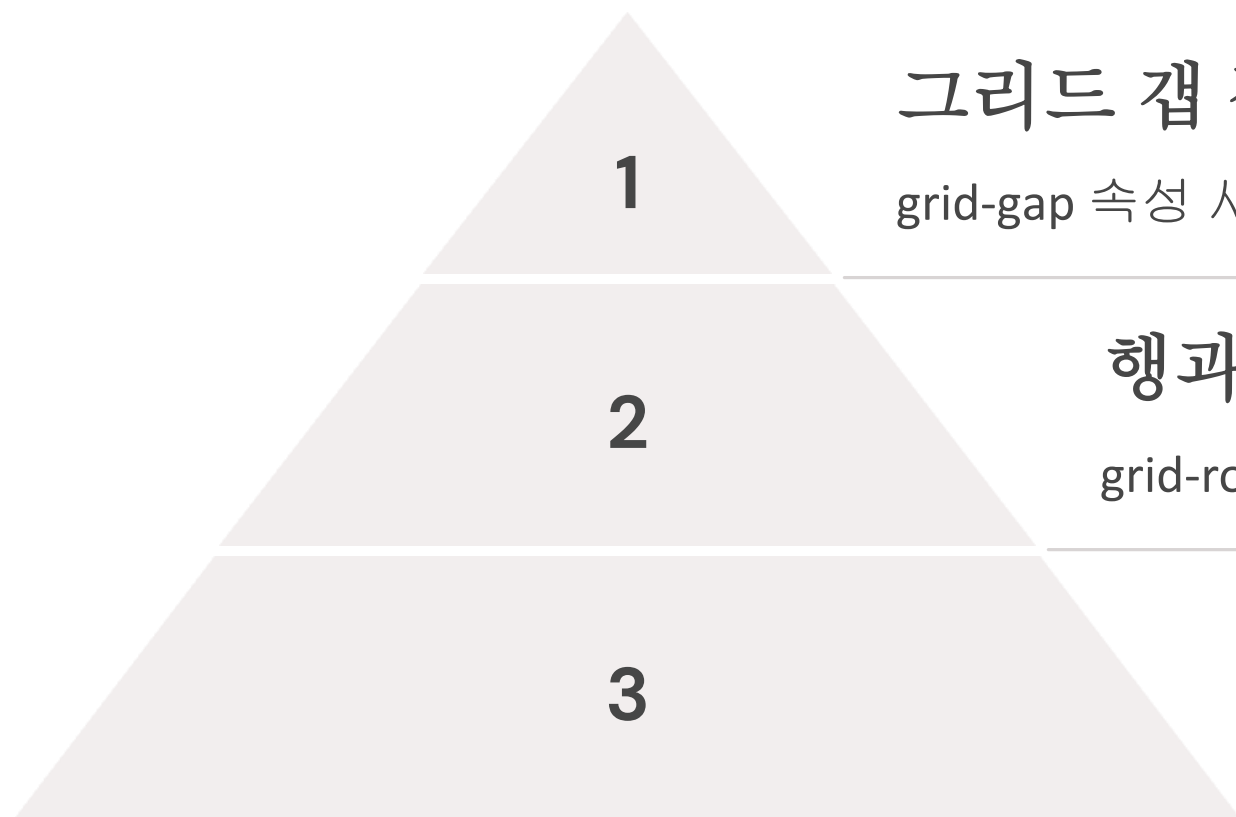
## fr 단위의 활용

- fr 단위는 가용 공간의 비율
- 이는 반응형 그리드 레이아웃을 만들 때 매우 유용한 도구
- 화면 크기에 따라 유연하게 조절되는 그리드를 쉽게 구현

## 실제 적용 예시

```
<div style="display: grid; grid-template-columns: 1fr 2fr;">  
  <div>Item 1</div>  
  <div>Item 2</div>  
</div>
```

# 그리드 갭 추가하기



## 그리드 갭 적용

grid-gap 속성 사용

## 행과 열 간격 설정

grid-row-gap, grid-column-gap

## 아이템 간 간격 생성

시각적 구분 개선

## 그리드 갭 (Grid Gap)

- 그리드 아이템들 사이에 간격을 추가
  - **grid-gap:** 행과 열 사이의 간격 설정
  - grid-row-gap, grid-column-gap: 행과 열 간격을 따로 설정

```
<div style="display: grid; grid-template-columns: 100px 100px; grid-gap: 10px;">
  <div>Item 1</div>
  <div>Item 2</div>
<div>Item 2</div>
<div>Item 2</div>
</div>
```





## 그리드 아이템 위치 지정

1

### 그리드 아이템 위치 지정 소개

grid-column과 grid-row 속성을 사용해 그리드 아이템의 위치 지정

2

```
<div style="display: grid; grid-template-columns: 100px 100px 100px;">  
  <div style="grid-column: 1 / 3;">Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
  <div>Item 4</div>  
</div>
```

3

### 결과 설명

이 예시에서 Item 1은 첫 번째 열부터 세 번째 열까지 확장되며, 나머지 아이템들은 자동으로 배치됩니다.

# 자동 배치 (Auto Placement)

## 그리드의 자동 배치 기능

그리드 시스템은 요소를 자동으로 배치할 수 있으며,  
빈 자리를 채워나감 → 효율적 레이아웃 구성

## 명시적 배치 vs 자동 배치

요소를 명시적으로 배치하지 않으면, 그리드가 자동으로 다음 빈 공간에 배치 → 복잡한 레이아웃을 간단하게 만들 수 있게 해줌

```
<div style="display: grid; grid-template-columns: 100px 100px;">  
  <div style="background-color: lightblue;">Item 1</div>  
  <div style="background-color: lightgreen;">Item 2</div>  
  <div style="background-color: lightcoral;">Item 3</div>  
  <div style="background-color: lightpink;">Item 4</div>  
</div>
```

# 그리드 영역 사용하기

1

## 그리드 영역 정의

grid-template-areas: 그리드 레이아웃을 이름으로 정의 → 직관적 관리

2

## 아이템 배치

grid-area: 각 아이템을 정의된 그리드 영역에 배치 → 레이아웃의 구조를 쉽게 이해하고 조정

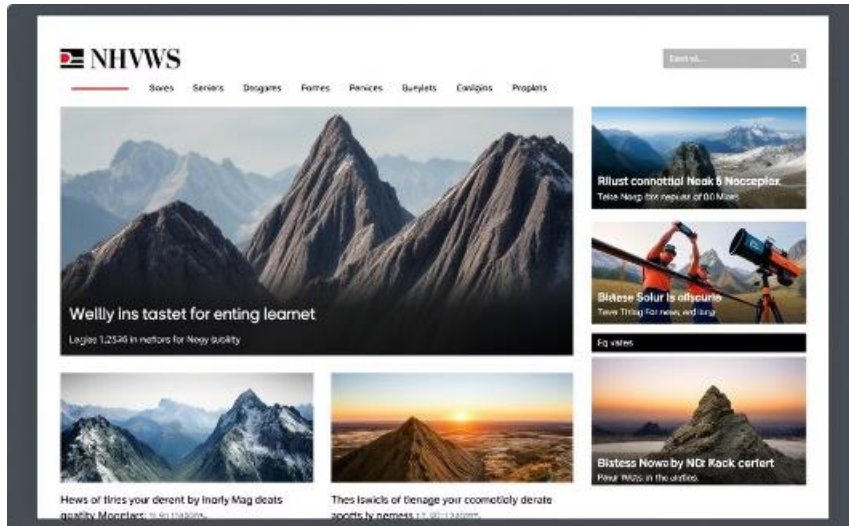
3

## 레이아웃 예시

2열 레이아웃을 정의 예제: 'header', 'sidebar', 'content', 'footer'와 같은 이름으로 영역 지정

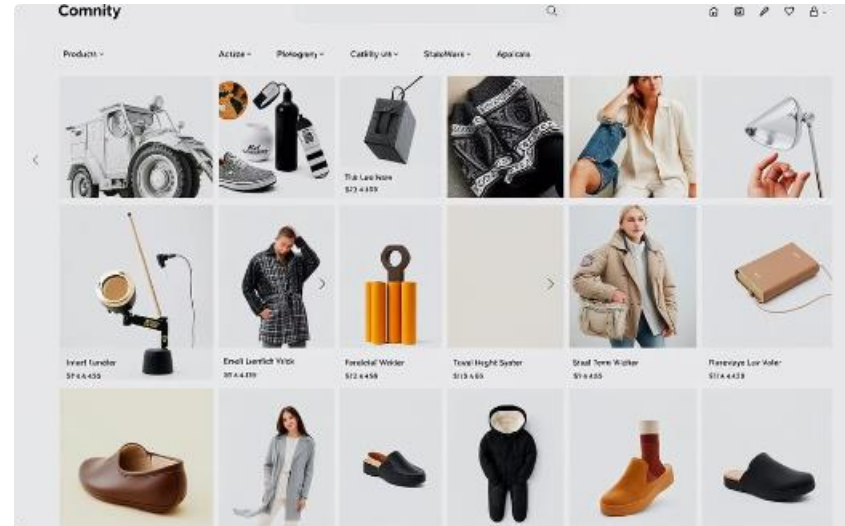
```
<div style="display: grid; grid-template-areas: 'header header' 'sidebar content' 'footer footer' ;  
      grid-template-columns: 150px 1fr; grid-template-rows: auto 1fr auto;">  
  <div style="grid-area: header; background-color: lightblue;">Header</div>  
  <div style="grid-area: sidebar; background-color: lightgreen;">Sidebar</div>  
  <div style="grid-area: content; background-color: lightcoral;">Content</div>  
  <div style="grid-area: footer; background-color: lightpink;">Footer</div>  
</div>
```

# 그리드 레이아웃의 실제 적용 사례



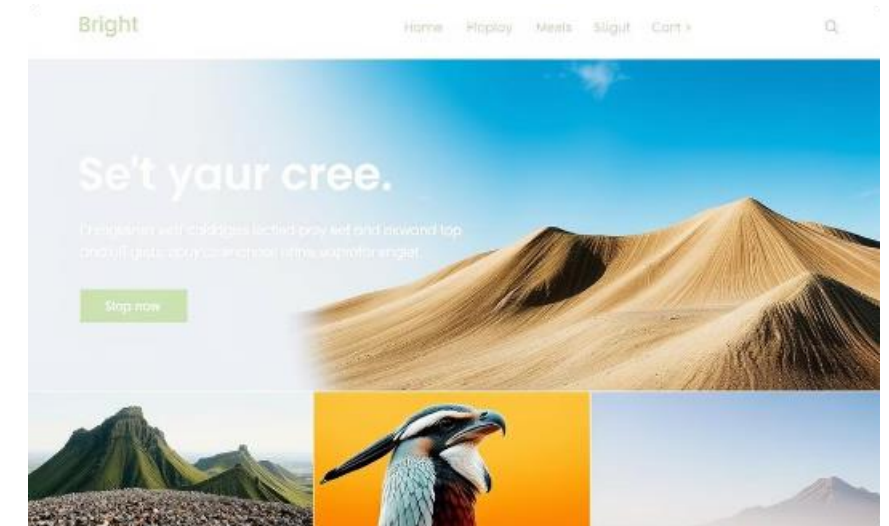
## 뉴스 웹사이트

다양한 크기의 기사와 이미지를  
효과적으로 배치



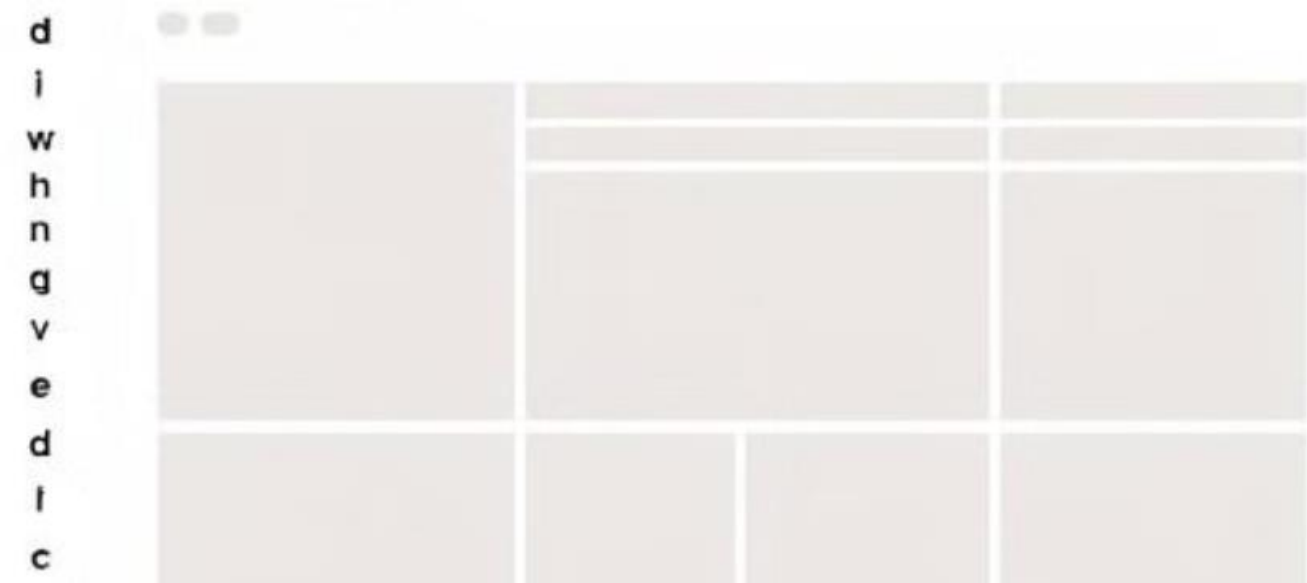
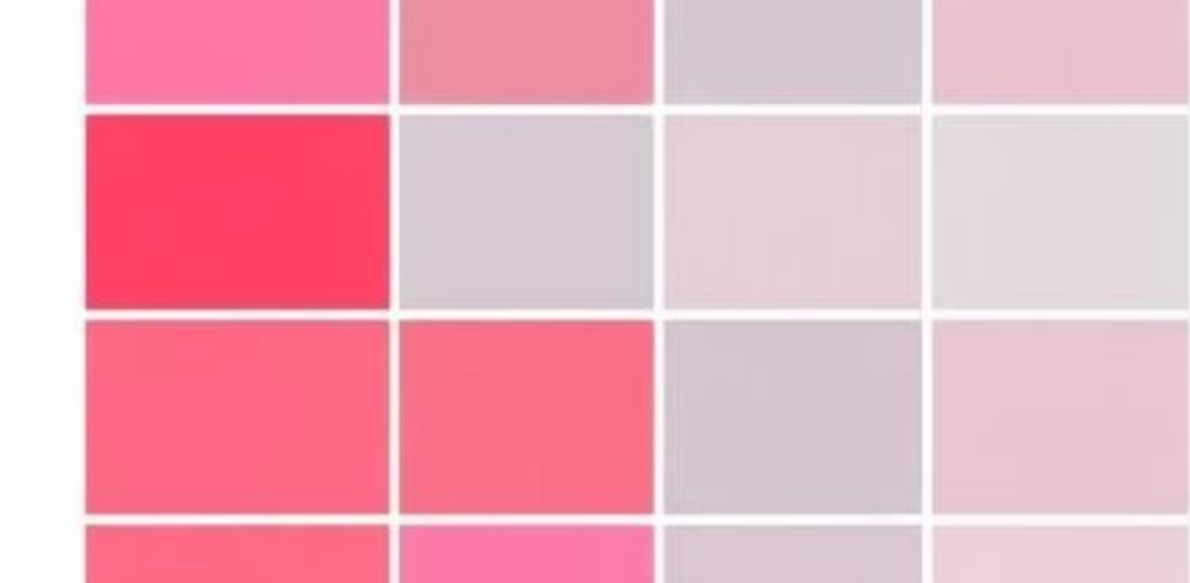
## 전자상거래 사이트

제품 목록을 깔끔하고 일관된 형태로  
표시



## 포트폴리오 사이트

다양한 크기의 프로젝트 이미지를  
그리드를 통해 효과적으로 배치



## 그리드와 플렉스박스 비교

특성	그리드	플렉스박스
차원	2차원 (행과 열)	1차원 (행 또는 열)
레이아웃 제어	매우 정밀한 제어 가능	유연한 크기 조정에 적합
복잡성	복잡한 레이아웃에 적합	간단한 레이아웃에 적합
학습 곡선	상대적으로 가파름	비교적 완만함

# CSS Grid 요약

## 그리드 컨테이너 정의

- `display: grid`
- 웹 레이아웃의 유연성과 관리 용이성을 크게 향상 시킴

## 열과 행 설정

- `grid-template-columns`
- `grid-template-rows`
- 그리드의 기본 구조 정의

## 간격 지정

- 그리드 아이템 간의 간격 지정
- 레이아웃의 시각적 균형 조절

## 아이템 위치 조정

- `grid-column`
- `grid-row`
- 개별 요소의 정확한 배치를 제어

## 그리드 영역 활용

- 복잡한 레이아웃 쉽게 관리
- 대규모 레이아웃 구성에 특히 유용