Part 2
CSS로 웹 페이지 꾸미기

길벗

코딩 자율학습

HTML+CSS+자바스크립트

4장
웹 스타일링을 위한 CSS 기초 배우기

4.1 CSS 문법 살펴보기

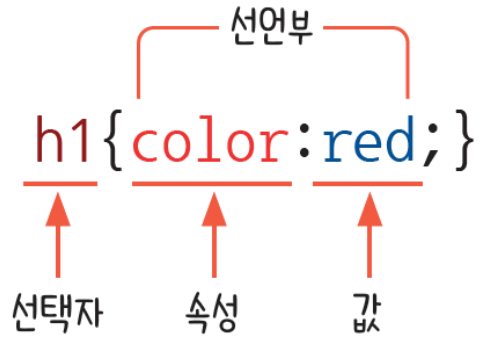
4.2 CSS 적용하기

4.1 CSS 문법 살펴보기

1. 문법 형식

- 선택자 : CSS 속성을 적용할 대상(태그, 요소)을 지정하는 영역
- 선언부 : 선택자로 지정된 대상(태그, 요소)에 적용할 CSS 속성과 값을 적는 영역

그림 4-2 CSS 문법 형식



2. 주석

- 코드에 설명을 남기는 것

```
형식 /* 주석 내용 */
```

4.2 CSS 적용하기

1. 내부 스타일 시트 사용하기

- HTML 문서 내부에 style 태그로 CSS 코드를 작성

형식
```
<style>
    /* CSS 코드 */
</style>
```

2. 외부 스타일 시트 사용하기

- 별도의 CSS 파일을 만들어 HTML 문서와 link 태그로 연결

형식 `<link rel="stylesheet" href="css 파일 경로">`

3. 인라인 스타일 사용하기

- 모든 태그에서 사용할 수 있는 style 속성을 사용

형식 `<태그 style="CSS 코드"></태그>`

5장
CSS 선택자 다루기

5.1 기본 선택자 사용하기

5.2 조합 선택자 사용하기

5.3 가상 요소 선택자 사용하기

5.4 가상 클래스 선택자 사용하기

5.5 다양한 선택자 조합하기

5.1 기본 선택자 사용하기

1. 전체 선택자

- 모든 태그를 선택자로 지정

  형식 `*{/* CSS 코드 */}`

2. 태그 선택자

- 태그명으로 선택자 지정

  형식 `태그명{/* CSS 코드 */}`

3. 아이디 선택자

- id 속성값으로 선택자 지정

  형식 `#id속성값{/* CSS 코드 */}`

5.1 기본 선택자 사용하기

4. 클래스 선택자

- class 속성값으로 선택자 지정

  형식 `.class속성값{/* CSS 코드 */}`

5. 기본 속성 선택자

- HTML 태그에서 사용할 수 있는 속성과 값으로 선택자 지정

  형식 `[속성]{/* CSS 코드 */}`

  `[속성=값]{/* CSS 코드 */}`

5.1 기본 선택자 사용하기

6. 문자열 속성 선택자

- 태그의 속성값이 특정한 문자열과 일치하는 요소를 선택자로 지정
  - [속성~=문자열] 속성값에 문자열이 포함되어 있으면 선택(단어 기준)
  - [속성|=문자열] 속성값이 문자열과 같거나 문자열-(하이픈)으로 시작하면 선택
  - [속성^=문자열] 속성값이 문자열로 시작하면 선택
  - [속성$=문자열] 속성값이 문자열로 끝나면 선택
  - [속성*=문자열] 속성값에 문자열이 포함되면 선택(전체 값 기준)

5.2 조합 선택자 사용하기

1. 그룹 선택자

- 여러 선택자를 그룹으로 묶어 지정
- 구분자 : ,

형식 선택자1, 선택자2,… 선택자n{/* CSS 코드 */}

2. 자식 선택자

- 선택자 범위를 자식 관계로 제한
- 구분자 : >

형식 부모 선택자 > 자식 선택자{/* CSS 코드 */}

5.2 조합 선택자 사용하기

3. 하위 선택자

- 선택자 범위를 자식 및 자손 관계로 제한
- 구분자 : 공백

형식 선택자1 선택자2 선택자3 ...{/* CSS 코드 */}

4. 인접 형제 선택자

- 특정 태그와 가장 인접한 형제 관계 태그를 선택자로 지정
- 구분자 : +

형식 이전 선택자 + 대상 선택자{/* CSS 코드 */}

5.2 조합 선택자 사용하기

5. 일반 형제 선택자

- 특정 태그와 형제 관계에 있는 모든 태그를 선택자로 지정
- 구분자 : ~

형식 이전 선택자 ~ 대상 선택자{/* CSS 코드 */}

5.3 가상 요소 선택자 사용하기

- 실제로 존재하는 요소는 아니지만, 존재한다고 가정하고 선택하는 방법
- 가상 요소 선택자 앞에 ::(콜론 2개)를 붙임

  형식 기준 선택자::가상 요소 선택자{/* CSS 코드 */}

- ::before – 기준 선택자 요소의 앞 선택
- ::after - 기준 선택자 요소의 뒤 선택

5.4 가상 클래스 선택자 사용하기

- 요소의 상태를 이용해 선택자를 지정하는 방법
- 가상 클래스 선택자 앞에 : 기호를 붙임

  형식 기준 요소:가상 클래스 선택자{/* CSS 코드 */}

1. 링크 가상 클래스 선택자

- :link - 링크를 한 번도 방문한 적 없는 상태
- :visited - 링크를 한 번 이상 방문한 적이 있는 상태

5.4 가상 클래스 선택자 사용하기

2. 동적 가상 클래스 선택자

- :hover - 마우스를 올린 상태
- :active - 마우스로 클릭한 상태

3. 입력 요소 가상 클래스 선택자

- :focus - 입력 요소에 커서가 활성화된 상태
- :checked - 체크박스 요소에 체크한 상태
- :disabled - 상호작용 요소가 비활성화된 상태
- :enabled - 상호작용 요소가 활성화된 상태

5.4 가상 클래스 선택자 사용하기

4. 구조적 가상 클래스 선택자

종류	설명
:first-child	첫 번째 자식 태그
:last-child	마지막 자식 태그
:nth-child(n)	n번째 자식 태그
:nth-last-child(n)	끝에서 n번째 자식 태그
:nth-of-type(n)	n번째 특정 자식 태그
:nth-last-of-type(n)	끝에서 n번째 특정 자식 태그
:first-of-type	부모의 첫 번째 특정 자식 태그
:last-of-type	부모의 마지막 특정 자식 태그

5.5 다양한 선택자 조합하기

- 선택자는 조합해서 사용할 수 있음

예

```
div.box{} /* class 속성값이 box인 div 태그 */

section#main /* id 속성값이 main인 section 태그 */

#main.box{} /* id 속성값이 main이고, class 속성값이 box인 요소 */

div:hover button{} /* div 태그에 마우스를 올린(hover) 상태일 때, 해당 div 태그 하위에 있는 button 태그 선택 */

div:hover > button{} /* div 태그에 마우스를 올린(hover) 상태일 때, 해당 div 태그와 자식 관계에 있는 button 태그 선택 */
```

6장
CSS 필수 속성 다루기

6.1 CSS의 특징 살펴보기

1. 기본 스타일 시트

- 웹 브라우저에 기본으로 내장된 스타일 시트

2. 적용 우선순위와 개별성

- 단계적 적용 : 같은 태그에 여러 스타일이 적용되더라도 단계적으로 적용되어 결국 마지막에 영향을 주는 하나의 스타일만 적용
- 적용 우선순위 : 같은 태그에 스타일 속성이 중복으로 작성됐을 때, 어느 속성을 적용할지 결정하는 기준
- 개별성 규칙의 점수에 따라 계산

3. 상속

- 부모 요소의 속성을 자식 요소가 물려받는 것

6.1 CSS의 특징 살펴보기

4. 단위

- 절대 단위
  - px : 모니터 화면을 구성하는 사각형 1개의 크기
- 상대 단위
  - % : 해당 속성의 상위 요소 속성값에 상대적인 크기
  - em : 부모 요소의 텍스트 크기에 상대적인 크기
  - rem : html 태그의 텍스트 크기에 상대적인 크기
  - vw : 뷰포트의 너비를 기준으로 상대적인 크기
  - vh : 뷰포트의 높이를 기준으로 상대적인 크기

6.1 CSS의 특징 살펴보기

5. 색상 표기법

- 키워드 표기법 : 색상의 영문명을 속성값으로 사용
- RGB 표기법 : Red, Green, Blue, alpha 값을 0~255로 표기
- HEX 표기법 : Red, Green, Blue에 해당하는 값을 각각 16진수로 변환해 00~ff로 표기

그림 6-7 RGB 색상 표기법의 rgb와 rgba 형식

```
rgb(red, green, blue)
rgba(red, green, blue, alpha)
```

그림 6-9 HEX 표기법

```
#RRGGBB
```

6.2 텍스트 속성으로 텍스트 꾸미기

1. font-family 속성

- 글꼴 지정

  형식 `font-family:<글꼴1>, <글꼴2>, ...<글꼴 유형>;`

- 글꼴 유형
  - serif : 삐침이 있는 명조 계열의 글꼴
  - sans-serif : 삐침이 없고 굵기가 일정한 고딕 계열의 글꼴
  - monospace : 텍스트 폭과 간격이 일정한 글꼴
  - fantasy : 화려한 글꼴
  - cursive : 손으로 쓴 것 같은 필기체 계열의 글꼴

6.2 텍스트 속성으로 텍스트 꾸미기

2. font-size 속성

- 텍스트 크기 지정

  형식 `font-size:<크기>;`

3. font-weight 속성

- 텍스트 굵기 지정

  형식 `font-weight:<숫자 표기법>|<키워드 표기법>;`

- 숫자 표기법 : 100 단위로 굵기를 표기
- 키워드 표기법 : lighter, normal, bold, bolder의 키워드로 표기

6.2 텍스트 속성으로 텍스트 꾸미기

4. font-style 속성

- 글꼴 스타일 지정

  형식 `font-style:<속성값>;`

- 속성값
  - normal : 기본 형태로 표시
  - italic : 이탤릭체로 표시
  - oblique : 기울임꼴로 표시

6.2 텍스트 속성으로 텍스트 꾸미기

5. font-variant 속성

- 영문 소문자를 크기가 작은 대문자로 변경

  형식 `font-variant:<속성값>;`

- 속성값
  - normal : 텍스트를 변환하지 않음
  - small-caps : 텍스트를 크기가 작은 대문자로 변환

6. color 속성

- 텍스트 색상 지정

  형식 `color:<속성값>;`

6.2 텍스트 속성으로 텍스트 꾸미기

7. text-align 속성

- 텍스트 정렬을 지정

  형식 `text-align:<속성값>;`

- 속성값
  - left : 텍스트를 왼쪽 정렬
  - center : 텍스트를 중앙 정렬
  - right : 텍스트를 오른쪽 정렬
  - justify : 텍스트를 양쪽 정렬

6.2 텍스트 속성으로 텍스트 꾸미기

8. text-decoration 속성

- 텍스트 꾸밈을 지정(텍스트에 선을 긋는 것)

  형식 `text-decoration:<속성값>;`

- 속성값
  - none : 텍스트 장식을 모두 지움
  - line-through : 텍스트 중간을 관통하는 선
  - overline : 텍스트 위에 선
  - Underline : 텍스트 아래에 선

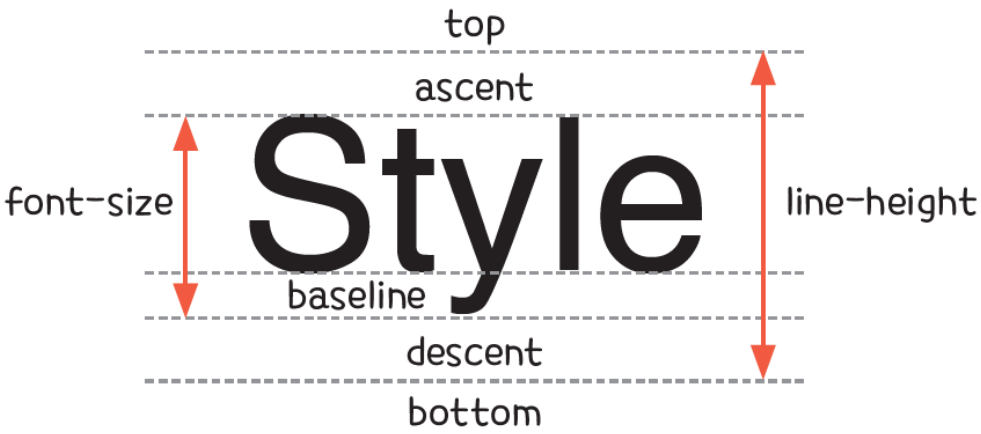
6.2 텍스트 속성으로 텍스트 꾸미기

9. letter-spacing 속성

- 자간을 지정(normal 또는 크기)

형식 `letter-spacing:normal|<크기>;`

그림 6-15 텍스트 상세



6.2 텍스트 속성으로 텍스트 꾸미기

10. line-height 속성

- 행간을 지정(normal이나 숫자, 퍼센트, 크기)

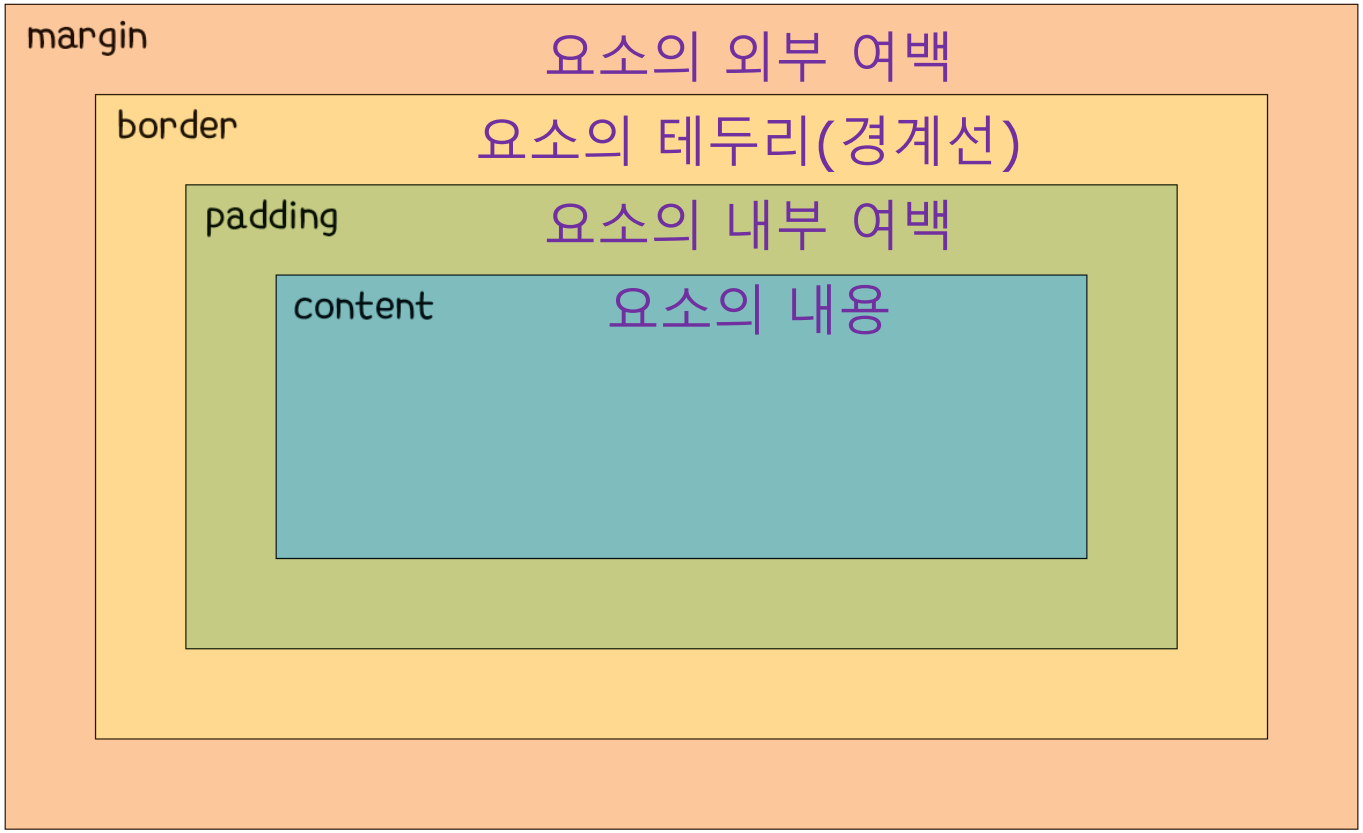
  형식 `line-height:normal|<속성값>;`

- 속성값
  - normal : 웹 브라우저에서 정한 기본값을 적용
  - 숫자 : 현재 font-size 값에 입력한 숫자를 곱한 값을 적용
  - 퍼센트 : 현재 font-size 값에 입력한 비율을 곱한 값을 적용
  - 크기 : 입력한 크기를 적용

6.3 박스 모델을 구성하는 속성 다루기

- 박스 모델 : 모든 태그가 사각형 모양으로 구성됐다는 개념

그림 6-17 박스 모델



6.3 박스 모델을 구성하는 속성 다루기

1. margin 영역

형식 margin-top:<크기>;

margin-right:<크기>;

margin-bottom:<크기>;

margin-left:<크기>;

margin:<margin-top> <margin-right> <margin-bottom> <margin-left>;

margin:<margin-top> <margin-right> <margin-bottom>;

margin:<margin-top & margin-bottom> <margin-right & margin-left>;

margin:<margin-top & margin-right & margin-bottom & margin-left>;

- margin 겹침 현상 : 인접한 margin 값이 둘 중 더 큰 값으로 통일되는 것

6.3 박스 모델을 구성하는 속성 다루기

2. border 영역

형식 `border:<border-width> <border-style> <color>;`

- border-width 속성 : 테두리 굵기를 지정

  형식 `border-width:<크기>;`

- border-style 속성 : 테두리 모양을 지정

  형식 `border-style:<속성값>;`

6.3 박스 모델을 구성하는 속성 다루기

3. padding 영역

형식 padding-top:<크기>;

padding-right:<크기>;

padding-bottom:<크기>;

padding-left:<크기>;

padding:[padding-top] [padding-right] [padding-bottom] [padding-left];

padding:[padding-top] [padding-right] [padding-bottom];

padding:[padding-top & padding-bottom] [padding-right & padding-left];

padding:[padding-top & padding-right & padding-bottom & padding-left];

6.3 박스 모델을 구성하는 속성 다루기

4. content 영역

- width 속성 : content 영역의 너비

  형식 `width:<크기>;`

- height 속성 : content 영역의 높이

  형식 `height:<크기>;`

- box-sizing 속성 : width, height 속성의 적용 기준을 지정

  형식 `box-sizing:<속성값>;`

  - content-box : width, height 속성 적용 범위를 content 영역으로 제한
  - border-box : width, height 속성 적용 범위를 border 영역으로 제한

6.3 박스 모델을 구성하는 속성 다루기

5. 박스 모델의 성격과 display 속성

- 박스 모델의 성격
  - 블록 성격 : 요소의 너비가 콘텐츠 유무와 상관없이 항상 가로 한 줄을 다 차지하는 것
  - 인라인 성격 : 요소의 너비를 콘텐츠 크기만큼만 차지하는 것
  - 인라인 블록 성격 : 요소의 너비가 콘텐츠의 크기만큼만 차지하지만, 그 외의 성격은 블록 성격을 가지는 복합적인 것
- display 속성 : HTML 태그가 기본으로 가지고 있는 박스 모델의 성격을 변경

6.4 배경 속성으로 요소의 배경 설정하기

1. background-color 속성

- 배경색 지정

  형식 `background-color:<색상값>;`

2. background-image 속성

- 배경에 이미지 삽입

  형식 `background-image:url("이미지 경로");`

6.4 배경 속성으로 요소의 배경 설정하기

3. background-repeat 속성

- 배경 이미지의 반복 여부를 지정

  형식 `background-repeat:<속성값>;`

- 속성값
  - no-repeat : 이미지를 반복하지 않음
  - repeat-x : 이미지를 가로 방향으로 반복
  - repeat-y : 이미지를 세로 방향으로 반복
  - repeat : 이미지를 가로와 세로 방향으로 반복
  - round : 이미지를 반복하되 이미지가 요소에 맞도록 크기 자동 조절
  - space : 이미지가 잘리지 않도록 반복

6.4 배경 속성으로 요소의 배경 설정하기

4. background-size 속성

- 배경 이미지의 크기 지정

  형식 `background-size:auto|cover|contain|<너비 높이>;`

- 속성값
  - auto : 이미지 크기 유지
  - cover : 이미지의 가로 세로 비율을 유지하면서 크기를 확대하거나 축소해 요소의 배경에 꽉 채우기
  - contain : 이미지의 가로 세로 비율을 유지하면서 이미지가 배경 요소 안에 들어가도록 크기를 확대하거나 축소
  - <너비 높이> : 이미지 크기를 직접 지정

6.4 배경 속성으로 요소의 배경 설정하기

5. background-position 속성

- 배경 이미지의 위치를 지정

  형식 `background-position:<x 위치> <y 위치>;`

- 속성값
  - left, center, right : x축(가로) 방향의 위치를 지정
  - top, center, bottom : y축(세로) 방향의 위치를 지정
  - px, rem, em, % : 위치를 직접 지정

6.4 배경 속성으로 요소의 배경 설정하기

6. background-attachment 속성

- 배경 이미지를 스크롤할 때의 모습을 결정

  형식 `background-attachment:<속성값>;`

- 속성값
  - local : 삽입된 이미지가 요소와 웹 브라우저에서 모두 스크롤
  - scroll : 삽입된 이미지가 요소에서는 고정되지만, 웹 브라우저에서는 스크롤
  - fixed :삽입된 이미지가 요소와 웹 브라우저에서 모두 고정

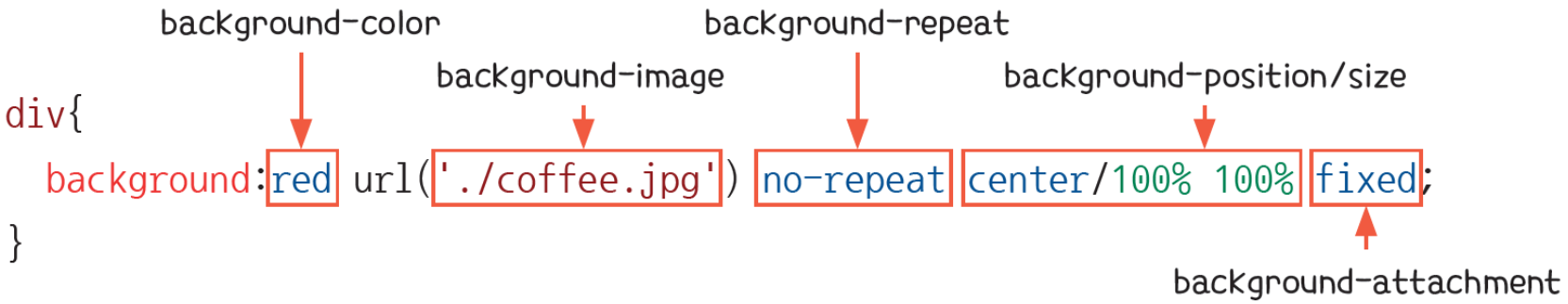
6.4 배경 속성으로 요소의 배경 설정하기

7. background 속성으로 한 번에 지정하기

- 모든 배경 속성을 한 번에 사용할 수 있는 단축 속성

형식 background:<color 속성값> <image 속성값> <repeat 속성값>

06/04/background.html



6.5 위치 속성으로 HTML 요소 배치하기

1. position 속성

형식 `position:<속성값>;`

- 속성값
  - static : 요소를 기본 흐름에 따라 배치
  - relative : 요소를 기본 흐름에 따라 배치, 좌표 속성 사용 가능
  - absolute : 요소를 기본 흐름에서 벗어나 절대적인 좌표 위치에 따라 배치
  - fixed : 요소를 기본 흐름에서 벗어나 절대적인 좌표 위치에 따라 배치(단, 스크롤해도 해당 위치에 고정)
  - sticky : 요소를 static 값처럼 기본 흐름에 따라 배치, 지정한 좌표의 임계점에 이르면 fixed 값처럼 화면에 고정

6.5 위치 속성으로 HTML 요소 배치하기

- z-index 속성 : position 속성으로 배치한 요소의 z축 위치를 결정

형식 z-index:<정숫값>;

예
```
.red-box{
    background-color:red;
    position:relative;
    z-index:10;
}
```

그림 6-54 실행결과

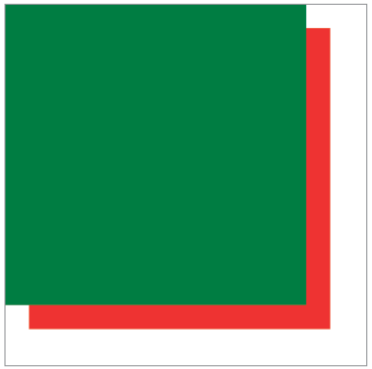
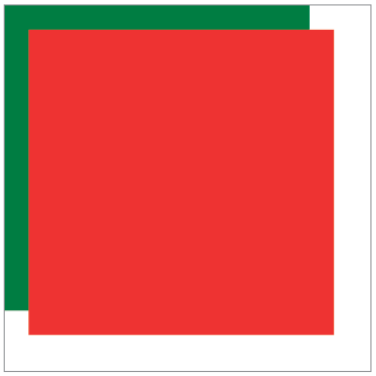


그림 6-55 실행결과



6.5 위치 속성으로 HTML 요소 배치하기

2. float 속성

형식 `float:<속성값>;`

- 속성값
  - none : float 속성을 적용하지 않음
  - left : 대상 요소를 공중에 띄워 왼쪽에 배치하면서 다음에 오는 요소를 주변에 자연스럽게 배치
  - right : 대상 요소를 공중에 띄워 오른쪽에 배치하면서 다음에 오는 요소를 주변에 자연스럽게 배치
- float 속성 사용 시 문제점 : float 속성은 적용 대상의 원래 위치를 보장하지 않으므로 주의!

6.5 위치 속성으로 HTML 요소 배치하기

3. clear 속성

형식 `clear:<속성값>;`

- 속성값
  - left : float 속성의 left 값을 해제
  - right : float 속성의 right 값을 해제
  - both : float 속성의 left와 right 값을 모두 해제

6.6 전환 효과 속성 적용하기

1. 전환이란

- 어떤 속성의 값이 다른 값으로 변경되는 것

2. transition-property 속성

- 전환 효과의 대상 속성명을 값으로 지정

  형식 `transition-property:<속성값>;`

- 속성값
  - none 전환 효과 속성을 지정하지 않음
  - all 모든 속성을 전환 효과 대상으로 지정

6.6 전환 효과 속성 적용하기

3. transition-duration 속성

- 전환 효과가 진행되는 시간을 지정

  형식 `transition-duration:<시간>;`

- 어떤 요소에 전환 효과를 지정하려면 반드시 transition-property 속성과 transition-duration 속성을 함께 사용해야 함

4. transition-delay 속성

- 전환 효과가 지연되는 시간을 지정

  형식 `transition-delay:<시간>;`

6.6 전환 효과 속성 적용하기

5. transition-timing-function 속성

- 전환 효과 속도를 지정

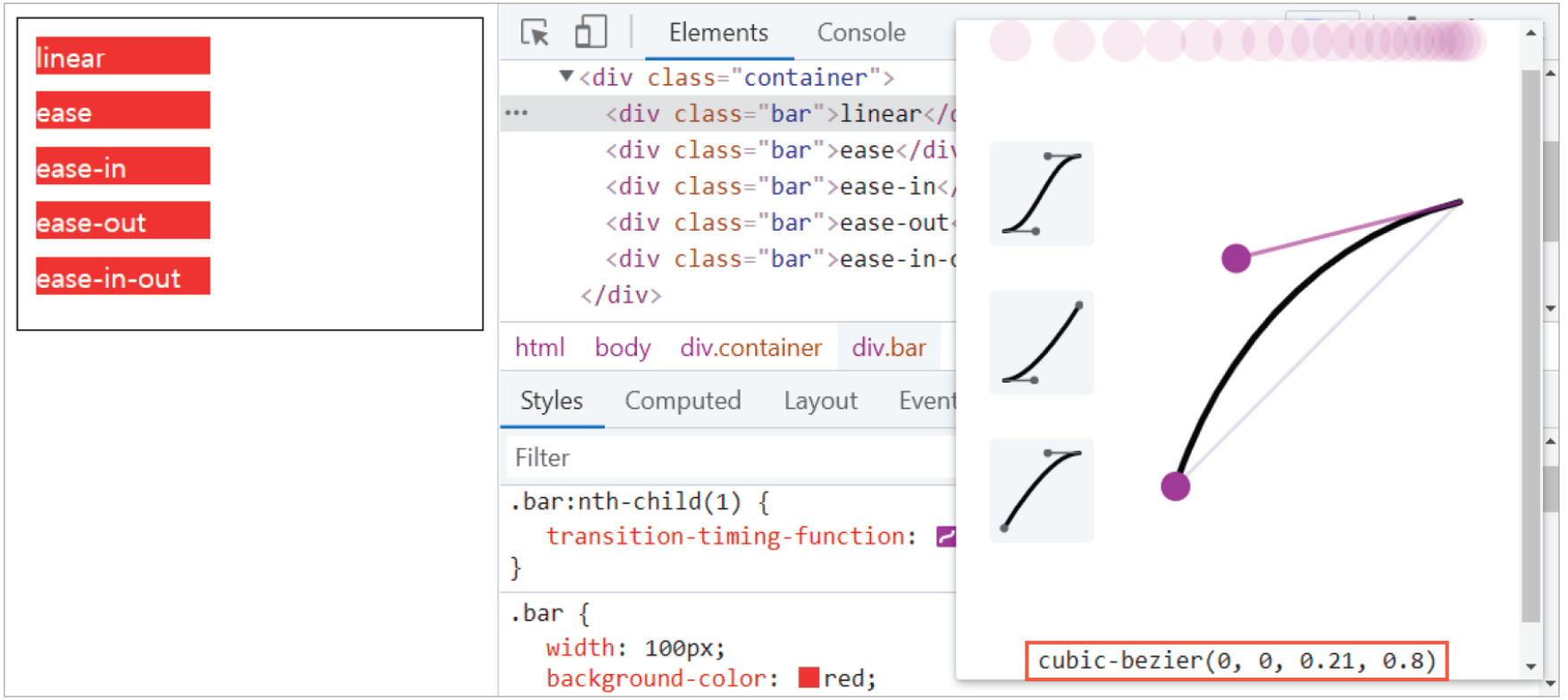
  형식 `transition-timing-function:<속성값>;`

- 속성값
  - linear : 처음 속도와 마지막 속도가 일정
  - ease : 처음에는 속도가 점점 빨라지다가 중간부터 점점 느려짐
  - ease-in : 처음에는 속도가 느리지만 완료될 때까지 점점 빨라짐
  - ease-out : 처음에는 속도가 빠르지만 완료될 때까지 점점 느려짐

6.6 전환 효과 속성 적용하기

- cubic-bezier(p1, p2, p3, p4) : 사용자가 정의한 속도로 진행

그림 6-73 곡선 조정 후



6.6 전환 효과 속성 적용하기

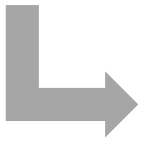
6. transition 속성으로 한 번에 지정하기

- 모든 전환 효과 속성을 한 번에 지정할 수 있는 단축 속성

형식 transition:<property>, <duration>, <timing-function>, <delay>`.

예

```
transition-property:width;
transition-duration:1s;
transition-timing-function:ease-in;
transition-delay:1s;
```



```
transition:width, 1s, ease-in, 1s;
```

6.7 애니메이션 속성으로 전환 효과 제어하기

1. 키 프레임 정의하기

- @keyframes 애니메이션의 전체 속성 정의

형식 `@keyframes <키 프레임명>{`

```
    0%{/* CSS 코드 */}
    n%{/* CSS 코드 */}
    100%{/* CSS 코드 */}
}
```

2. animation-name 속성

- 애니메이션을 지정할 키 프레임명 지정

형식 `animation-name:<키 프레임명>;`

6.7 애니메이션 속성으로 전환 효과 제어하기

3. animation-duration 속성

- 애니메이션의 지속 시간을 지정

  형식 `animation-duration:<지속 시간>;`

- 애니메이션은 키 프레임, animation-name 속성, animation-duration 속성 3가지 중 하나라도 빠지면 전환 효과는 적용되지 않음

4. animation-delay 속성

- 애니메이션의 지연 시간을 지정

  형식 `animation-delay:<지연 시간>;`

6.7 애니메이션 속성으로 전환 효과 제어하기

5. animation-fill-mode 속성

- 애니메이션 실행 전과 종료 후의 상태 지정

속성값	상태	설명
none	실행 전	시작 시점(0%, from)의 스타일을 적용하지 않고 대기
	실행 후	실행되기 전의 스타일 적용 상태로
forwards	실행 전	시작 시점(0%, from)의 스타일을 적용하지 않고 대기
	실행 후	키 프레임에 정의된 종료 시점(100%, to)의 스타일을 적용하고 대기
backwards	실행 전	키 프레임에 정의된 시작 시점(0%, from)의 스타일을 적용하고 대기
	실행 후	실행되기 전의 스타일 적용 상태로
both	실행 전	키 프레임에 정의된 시작 시점(0%, from)의 스타일을 적용하고 대기
	실행 후	키 프레임에 정의된 종료 시점(100%, to)의 스타일을 적용하고 대기

6.7 애니메이션 속성으로 전환 효과 제어하기

6. animation-iterator-count 속성

- 애니메이션의 반복 횟수 지정

  형식 `animation-iteration-count:<횟수>;`

7. animation-play-state 속성

- 애니메이션의 진행/정지 상태 정의

  형식 `animation-play-state:<속성값>;`

- 속성값
  - paused : 애니메이션의 실행을 일시 정지
  - running : 애니메이션을 실행

6.7 애니메이션 속성으로 전환 효과 제어하기

8. animation-direction 속성

- 애니메이션의 진행 방향을 지정

  형식 `animation-direction:<속성값>;`

- 속성값
  - normal : 애니메이션의 진행 방향을 키 프레임에 정의된 시간 순서대로 진행(to → from)
  - reverse : 애니메이션의 진행 방향을 키 프레임에 정의된 시간 순서의 역으로 진행(from → to)
  - alternate : 애니메이션이 1회 이상 실행될 경우 홀수 번째는 normal로, 짝수 번째는 reverse로 진행
  - alternate-reverse : 애니메이션이 1회 이상 실행될 경우 홀수 번째는 reverse로, 짝수 번째는 normal로 진행

6.7 애니메이션 속성으로 전환 효과 제어하기

9. animation 속성으로 한 번에 지정하기

- 모든 애니메이션 관련 속성 지정

형식 animation:<name> <duration> <timing-function> <delay>
<iteration-count> <direction> <fill-mode> <play-state>;

예 animation:bgchange 5s 3 ease-in;

6.8 변형 효과 적용하기

1. transform 속성

- 요소에 특정 변형 효과를 지정

  형식 `transform:<함수>;`

- 2차원 좌표 이동하기
  - translate(x, y) : 요소를 현재 위치에서 x(x축)와 y(y축)만큼 이동
  - translateX(n) : 요소를 현재 위치에서 n만큼 x축으로 이동
  - translateY(n) : 요소를 현재 위치에서 n만큼 y축으로 이동

6.8 변형 효과 적용하기

- 2차원 확대 또는 축소하기
  - scale(x, y) : 요소를 x(x축)와 y(y축)만큼 확대 또는 축소
  - scaleX(n) : 요소를 n만큼 x축으로 확대 또는 축소
  - scaleY(n) : 요소를 n만큼 y축으로 확대 또는 축소
- 2차원 기울이기
  - skew(xdeg, ydeg) : 요소를 x축과 y축으로 xdeg, ydeg만큼 기울임
  - skewX(deg) : 요소를 deg만큼 x축 방향으로 기울임
  - skewY(deg) : 요소를 주어진 deg만큼 y축 방향으로 기울임
- 2차원 회전하기
  - rotate(deg) : 요소를 deg만큼 회전

6.8 변형 효과 적용하기

2. transform-origin 속성

- 변형 효과의 기준점을 변경할 때

  형식 `transform-origin:<x축 위치> <y축 위치>;`

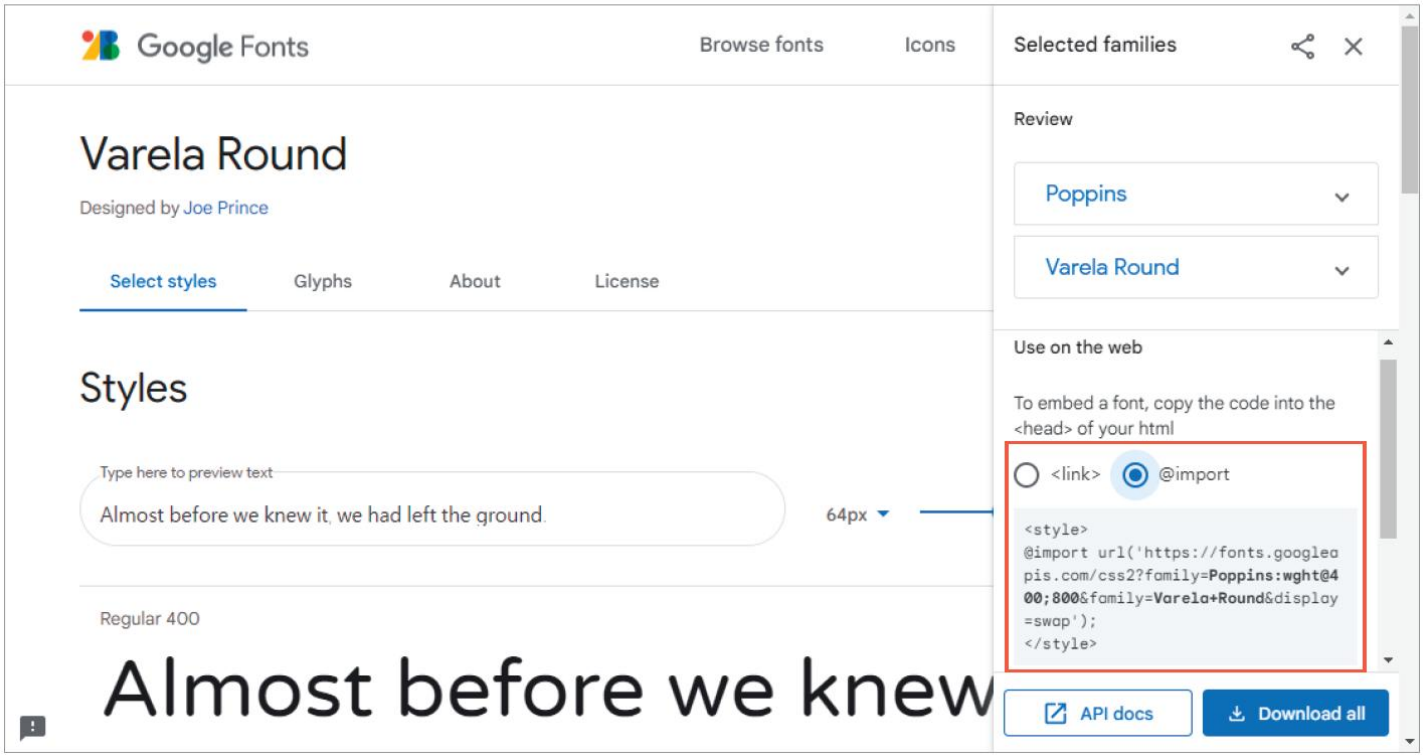
- 속성값
  - x축 기준 위치 : left 왼쪽 / center 중앙 / right 오른쪽
  - y축 기준 위치 : top 위쪽 / center 중앙 / bottom 아래쪽

6.9 웹 폰트와 아이콘 폰트 사용하기

1. 구글 폰트 적용하기

- 구글 웹 폰트 https://fonts.google.com

그림 6-87 글꼴 사용

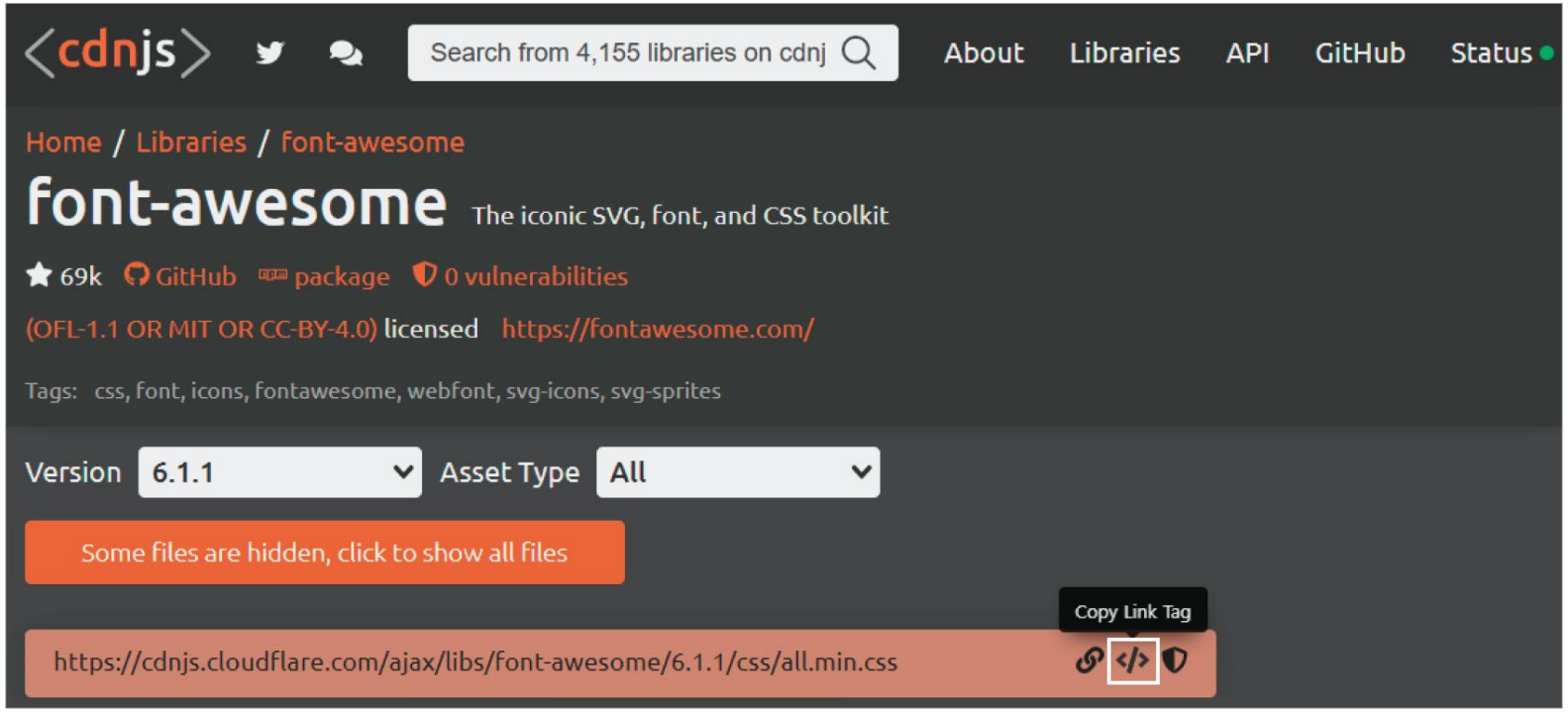


6.9 웹 폰트와 아이콘 폰트 사용하기

2. 아이콘 폰트 적용하기

- Font Awesome 라이브러리
  https://cdnjs.com/libraries/fontawesome

그림 6-89 CDNJS에서 제공하는 Font Awesome 라이브러리



7장
효과적인 레이아웃을 위한 CSS 속성 다루기

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

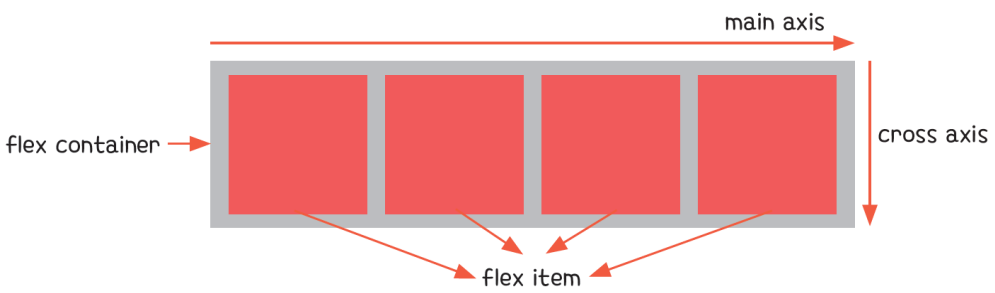
7.3 반응형 웹을 위한 미디어 쿼리 사용하기

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

1. 플렉스 박스 레이아웃 살펴보기

- 구성 요소
  - 주축(main axis) : 플렉스 박스의 진행 방향과 수평한 축
  - 교차축(cross axis) : 주축과 수직하는 축
  - 플렉스 컨테이너(flex container) : display 속성값으로 flex나 inline-flex가 적용된 요소
  - 플렉스 아이템(flex item) : 플렉스 컨테이너와 자식 관계를 이루는 태그 요소

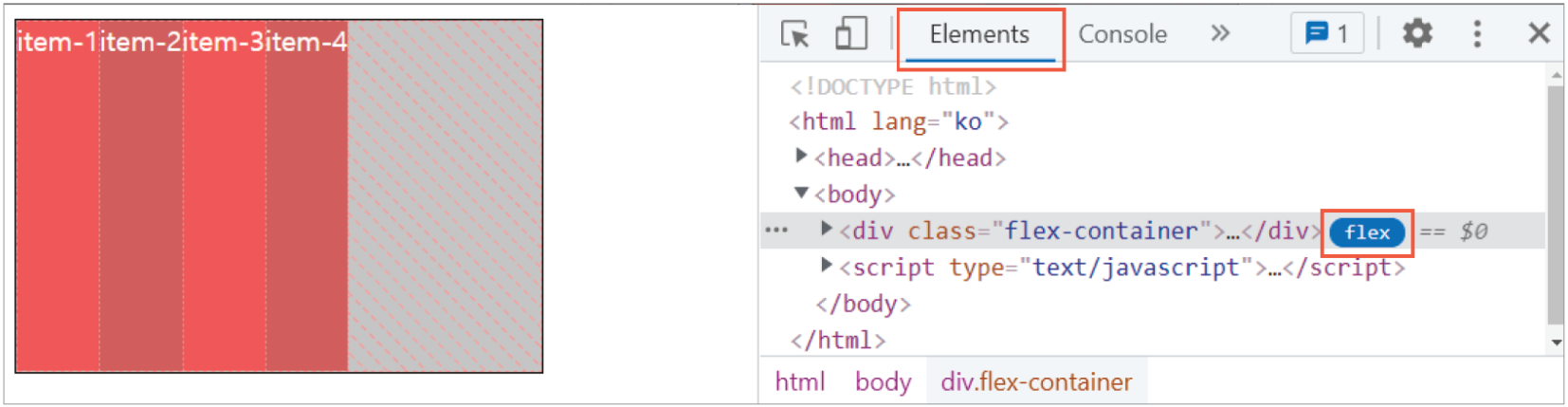
그림 7-1 플렉스 박스 레이아웃의 구성 요소



7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

- 레이아웃 확인 방법
  - 개발자 도구의 Elements 탭 > flex 아이콘 클릭

그림 7-2 크롬 브라우저에서 플렉스 박스 레이아웃 확인



7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

2. 플렉스 박스 레이아웃의 기본 속성

- display 속성 : flex, inline-flex 값을 지정하면 해당 요소가 플렉스 컨테이너로 설정

  형식 `display:flex; /* inline-flex */`

- flex-direction 속성 : 플렉스 박스 레이아웃의 주축 방향을 지정

  형식 `flex-direction:<속성값>;`

  - row : 주축 방향을 왼쪽에서 오른쪽으로 지정
  - row-reverse : 주축 방향을 오른쪽에서 왼쪽으로 지정
  - column : 주축 방향을 위쪽에서 아래쪽으로 지정
  - column-reverse : 주축 방향을 아래쪽에서 위쪽으로 지정

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

- flex-wrap 속성 : 플렉스 아이템의 자동 줄 바꿈 여부를 지정

  형식 `flex-wrap:<속성값>;`

  - nowrap : 플렉스 아이템이 플렉스 컨테이너를 벗어나도 무시
  - wrap : 플렉스 아이템이 플렉스 컨테이너를 벗어나면 줄 바꿈
  - wrap-reverse : 플렉스 아이템이 플렉스 컨테이너를 벗어나면 wrap의 역방향으로 줄 바꿈

- flex-flow 속성 : flex-direction과 fled-wrap 속성을 한 번에 사용할 수 있는 단축 속성

  형식 `flex-flow:<flex-direction> <flex-wrap>;`

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

3. 플렉스 박스 레이아웃의 정렬 속성

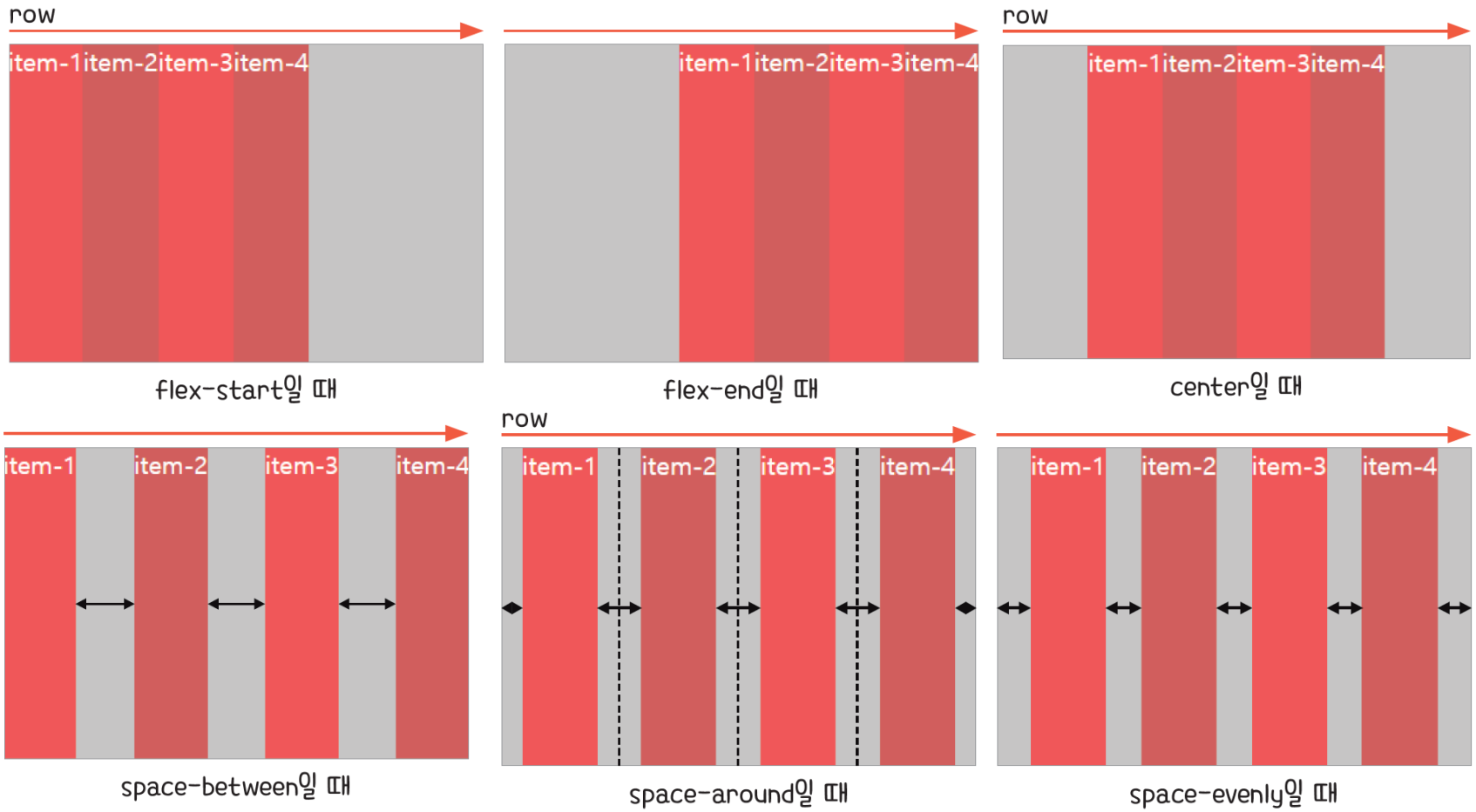
- justify-content 속성 :플렉스 아이템을 모두 주축 방향으로 정렬

  형식 `justify-content:<속성값>;`

  - flex-start : 주축 방향의 시작을 기준으로 정렬
  - flex-end : 주축 방향의 끝을 기준으로 정렬
  - center : 주축 방향의 중앙에 정렬
  - space-between : 플렉스 아이템 사이의 간격이 균일하도록 정렬
  - space-around : 플렉스 아이템의 둘레(around)가 균일하도록 정렬
  - space-evenly : 플렉스 아이템 사이와 양끝의 간격이 균일하도

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

그림 7-8 주축의 방향이 row일 때 justify-content 속성값에 따른 정렬 결과



7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

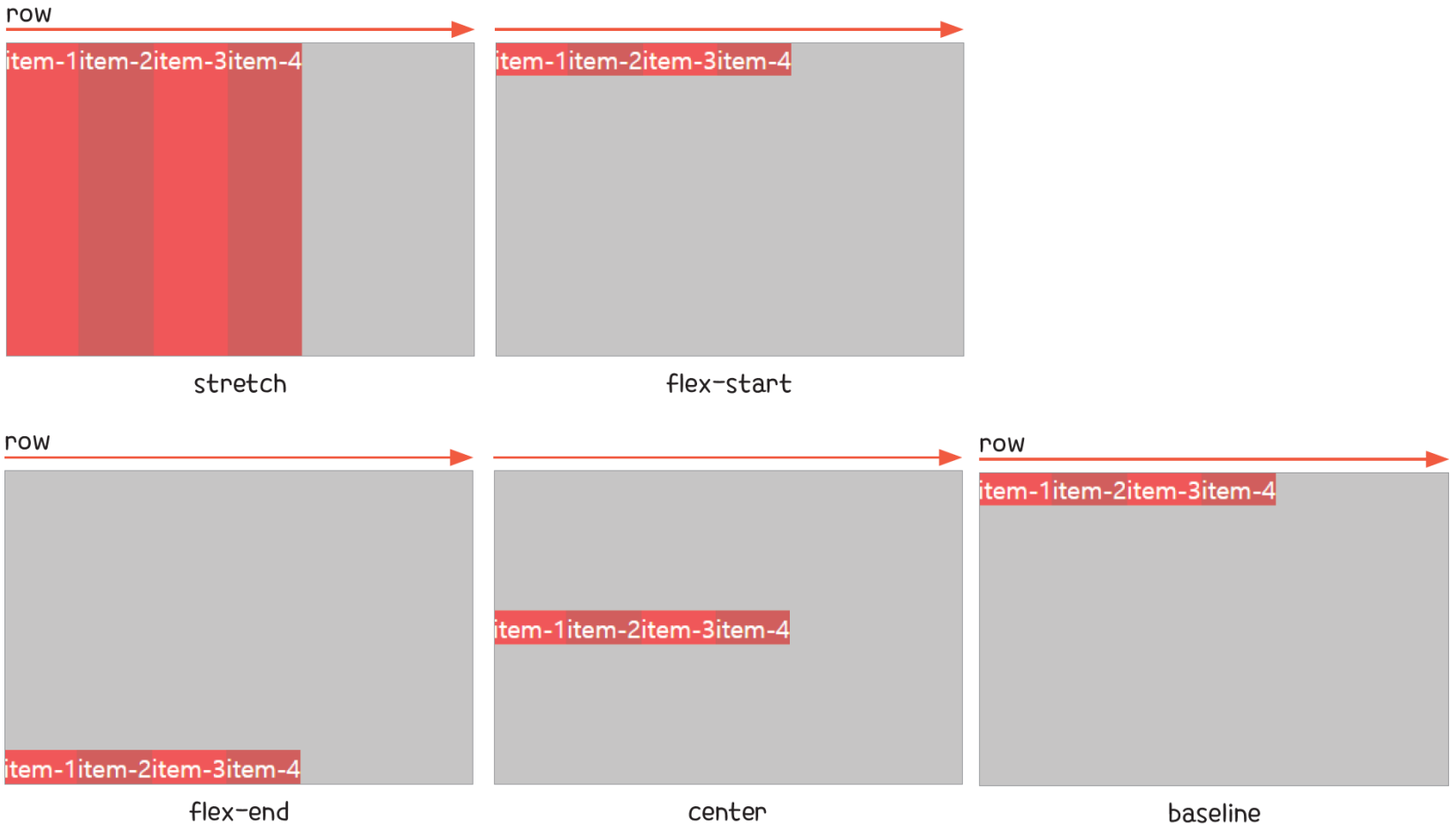
- align-items 속성 : 플렉스 아이템을 모두 교차축 방향으로 정렬

  형식 `align-items:<속성값>;`

  - stretch : 교차축 방향으로 플렉스 아이템의 너비나 높이가 늘어남
  - flex-start : 교차축 방향의 시작을 기준으로 정렬
  - flex-end : 교차축 방향의 끝을 기준으로 정렬
  - center : 교차축 방향의 중앙을 기준으로 정렬
  - baseline : 플렉스 아이템의 baseline을 기준으로 정렬

7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

그림 7-9 주축의 방향이 row일 때, align-items 속성값에 따른 정렬 결과



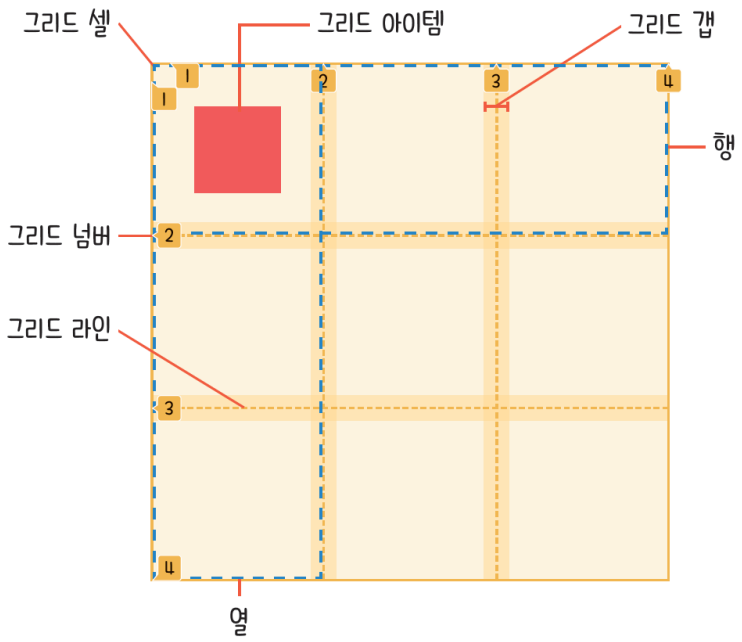
7.1 플렉스 박스 레이아웃으로 1차원 레이아웃 설계하기

- align-content 속성 : 플렉스 아이템이 두 줄 이상일 때 교차축 방향으로 정렬
- align-self 속성 : 각각의 플렉스 아이템을 교차축 방향으로 정렬, align-items 속성으로 플렉스 아이템을 한 번에 정렬하지 않고 각각 정렬하고 싶을 때 사용

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

1. 그리드 레이아웃 살펴보기

그림 7-10 그리드 레이아웃 구성 요소

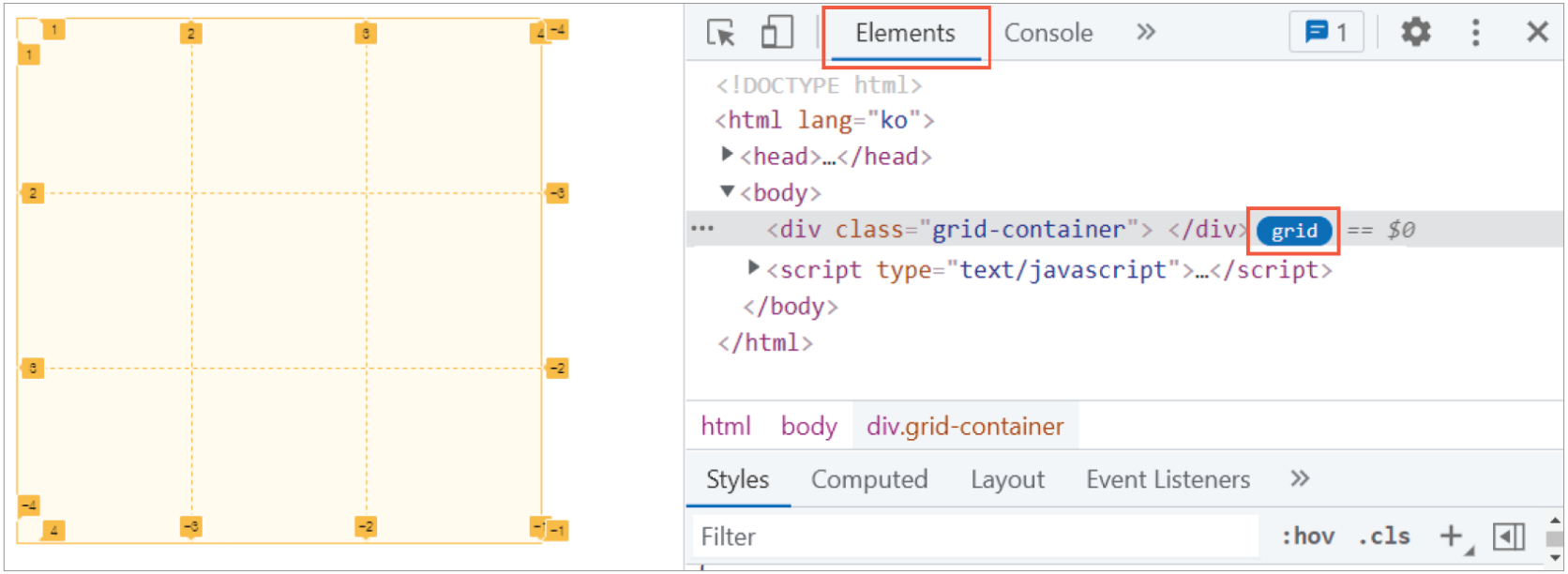


- 행(row) : 그리드 레이아웃의 가로줄
- 열(column) : 그리드 레이아웃의 세
- 그리드 셀(grid cell) : 행과 열이 만나 이루어지는 하나의 공간
- 그리드 갭(grid gap) : 그리드 셀과 그리드 셀 사이의 간격
- 그리드 아이템(grid item) : 그리드 안에 포함되는 콘텐츠
- 그리드 라인(grid line) : 그리드 행과 열을 그리는 선
- 그리드 넘버(grid number) : 그리드 라인에 붙는 번호
- 그리드 컨테이너(grid container) : 그리드 아이템을 묶는 부모 요소

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

- 레이아웃 확인 방법
  - 개발자 도구의 Elements 탭 > grid 아이콘 클릭

그림 7-11 크롬 브라우저에서 그리드 레이아웃 확인



7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

2. 그리드 레이아웃의 기본 속성

- display 속성 : 속성값을 grid, inline-grid로 지정하면 그리드 레이아웃을 만들 수 있음

  형식 `display:grid; /* inline-grid */`

- grid-template-columns와 grid-template-rows 속성 : 그리드 레이아웃의 행과 열을 지정

  형식 `grid-template-columns:<1열값> <2열값> ...;`

  `grid-template-rows:<1행값> <2행값> ...;`

- row-gap과 column-gap 속성 : 그리드 셀과 셀 사이의 간격을 지정

  형식 `row-gap:<크기>;`

  `column-gap:<크기>;`

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

3. 그리드 레이아웃의 정렬 속성

- align-items 속성 : 그리드 아이템 전체를 셀의 세로 방향으로 정렬
  - stretch : 그리드 아이템이 그리드 셀을 꽉 채우도록 크기를 늘림
  - start : 그리드 아이템을 그리드 셀의 맨 위에 배치
  - center : 그리드 아이템을 그리드 셀의 세로 방향 중간에 배치
  - end : 그리드 아이템을 그리드 셀의 맨 아래에 배치
- align-self 속성 : 각각의 그리드 아이템을 셀의 세로 방향으로 정렬

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

- justify-items 속성 : 그리드 아이템 전체를 셀의 가로 방향으로 정렬
  - stretch : 그리드 아이템을 그리드 셀이 꽉 차도록 늘림
  - start : 그리드 아이템을 그리드 셀의 왼쪽 끝에 배치
  - center : 그리드 아이템을 그리드 셀의 가로 방향 중간에 배치
  - end : 그리드 아이템을 그리드 셀의 오른쪽 끝에 배치
- justify-self 속성 : 각각의 그리드 아이템을 셀의 가로 방향으로 정렬

7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

- place-item 속성 : align-items와 justify-items 속성을 한 번에 사용할 수 있는 단축 속성

  형식 place-items:<align-items> <justify-items>;

- place-self 속성 : align-self와 justify-self 속성을 한 번에 사용할 수 있는 단축 속성

  형식 place-self:<align-self> <justify-self>;

  예

```
place-items:center end; /* align-items:center, justify-items:end */
place-self:center end; /* align-self:center, justify-self:end */
```

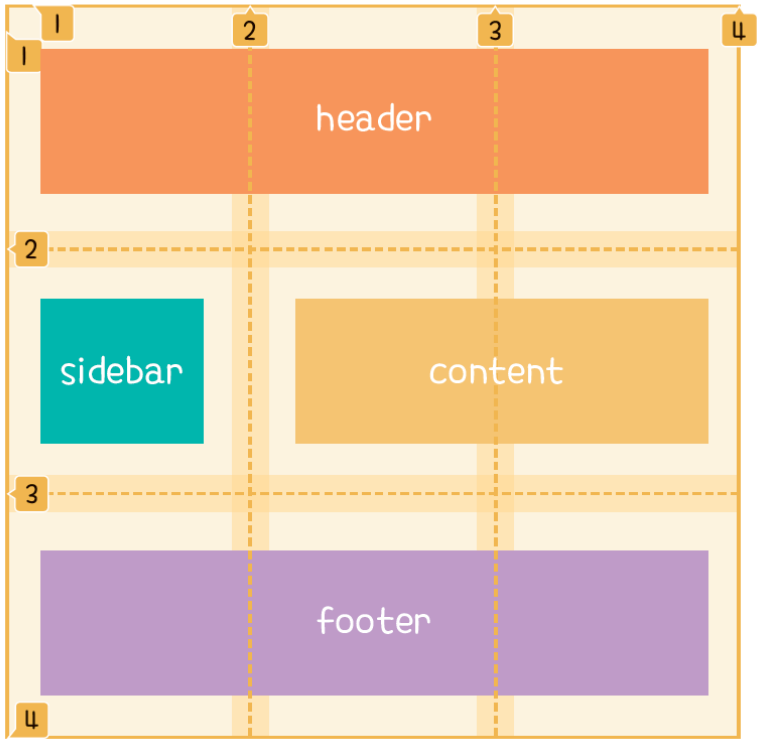
7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

4. 그리드 레이아웃의 배치 속성

- grid-template-areas 속성 : 그리드 레이아웃에서 행과 열을 이름으로 지정
- grid-area 속성 : 그리드 아이템에 이름을 지정

형식 `grid-area:<행과 열 이름>;`

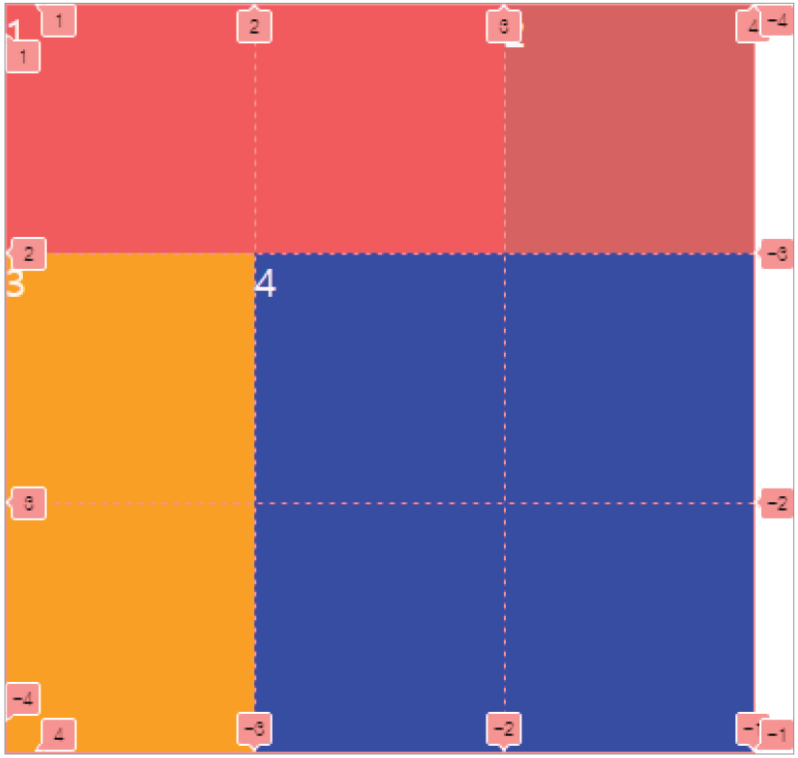
그림 7-18 그리드 레이아웃 예시



7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

- grid-column-start, grid-column-end 속성 : 그리드 레이아웃에서 열의 시작 번호와 끝 번호를 지정
- grid-row-start, grid-row-end 속성 : 그리드 레이아웃에서 행의 시작과 끝 번호를 지정
- 그리드 라인 : 그리드 컨테이너를 구성하는 행과 열을 그리는 선
- 그리드 넘버 : 그리드 라인에 있는 고유한 번호

그림 7-19 그리드 넘버



7.2 그리드 레이아웃으로 2차원 레이아웃 설계하기

- grid-column 속성 : grid-column-start와 grid-column-end 속성을 한 번에 사용할 수 있는 단축 속성
- grid-row 속성 : grid-row-start와 grid-row-end 속성을 한 번에 사용할 수 있는 단축 속성

형식 grid-column:<start> <end>;

grid-row:<start> <end>;

형식 grid-column:<start>/span <열 개수>;

grid-row:<start>/span <행 개수>;

7.3 반응형 웹을 위한 미디어 쿼리 사용하기

1. 미디어 쿼리란

- 사이트에 접속하는 미디어 타입과 특징, 해상도에 따라 다른 스타일 속성을 적용할 수 있게 하는 기술

그림 7-20 모질라 사이트의 반응형 웹

7.3 반응형 웹을 위한 미디어 쿼리 사용하기

2. 뷰포트 알아보기

- 뷰포트 : 웹 페이지가 접속한 기기에서 보이는 실제 영역의 크기

그림 7-22 해상도가 작은 기기에서 보이는 화면

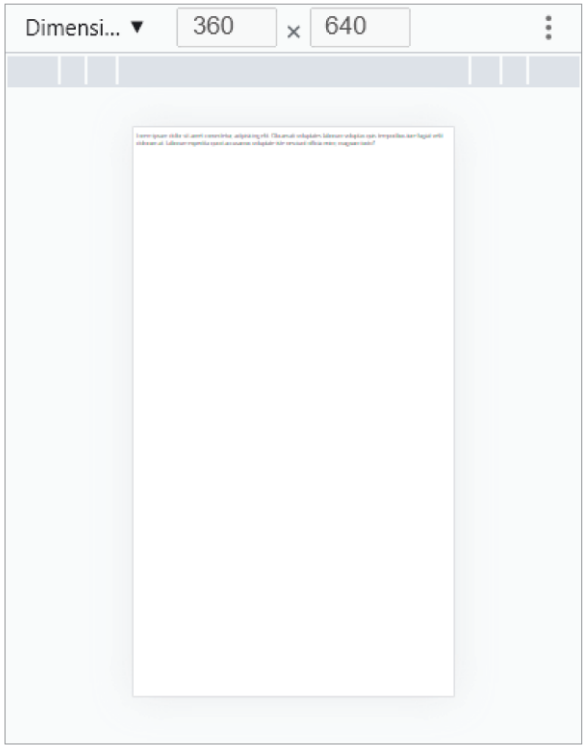


그림 7-23 뷰포트 적용 후 화면

7.3 반응형 웹을 위한 미디어 쿼리 사용하기

3. 미디어 쿼리의 기본 문법

형식

```
@media <not|only> <media type> and (<media feature>)
    <and|or|not> (<media feature>){
      /* CSS 코드; */
    }
```

- not/only
  - not : 뒤에 오는 모든 조건을 부정
  - only : 미디어 쿼리를 지원하는 기기만

7.3 반응형 웹을 위한 미디어 쿼리 사용하기

- media type : 미디어 쿼리가 적용될 미디어 타입
  - all : 모든 기기
  - print : 인쇄 장치(예: 프린터기)
  - screen : 컴퓨터 화면 장치 또는 스마트 기기
  - speech : 스크린 리더기 같은 보조 프로그램으로 웹 페이지를 소리 내어 읽어 주는 장치

7.3 반응형 웹을 위한 미디어 쿼리 사용하기

- media feature : 미디어 쿼리가 적용될 미디어 조건
  - min-width <화면 너비> : 미디어 쿼리가 적용될 최소 너비
  - max-width <화면 너비> : 미디어 쿼리가 적용될 최대 너비
  - orientation portrait : 세로 모드, 뷰포트의 세로 높이가 가로 너비보다 큰 경우
  - orientation landscape : 가로 모드, 뷰포트의 가로 너비가 세로 높이보다 큰 경우