14. 반응형 웹이란

14-1 모바일 기기와 웹 디자인

14-2 가변 그리드 레이아웃

14-3 가변 요소

모바일 기기와 웹 디자인

반응형 웹 디자인

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 표시하는 방법
- 다양한 화면 크기의 모바일 기기들이 계속 쏟아져 나오는데 그 때마다 그 크기에 맞춘 사이트를 별도로 제작하는 것은 비효율적

  → 화면 크기에 '반응'해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것이 바로 반응형 웹 디자인

반응형 웹 디자인의 장단점

장점

- 모든 스마트 기기에서 접속 가능
- 가로 모드에 맞춘 레이아웃 변경 가능
- 사이트 유지,관리 용이

단점

반응형 웹 기술이 최신 웹 표준인 CSS3의 일부

→ 최신 모던 웹 브라우저에서만 지원됨

모바일 기기와 웹 디자인

뷰포트(viewport)

- 뷰포트 : 실제 내용이 표시되는 영역
- PC 화면과 모바일 화면의 픽셀 표시 방법이 다르기 때문에 모바일 화면에서 의도한대로 표시되지 않음
  → 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시

뷰포트 지정하기

- <head> 태그 안에서 <meta> 태그를 이용해 뷰포트 지정

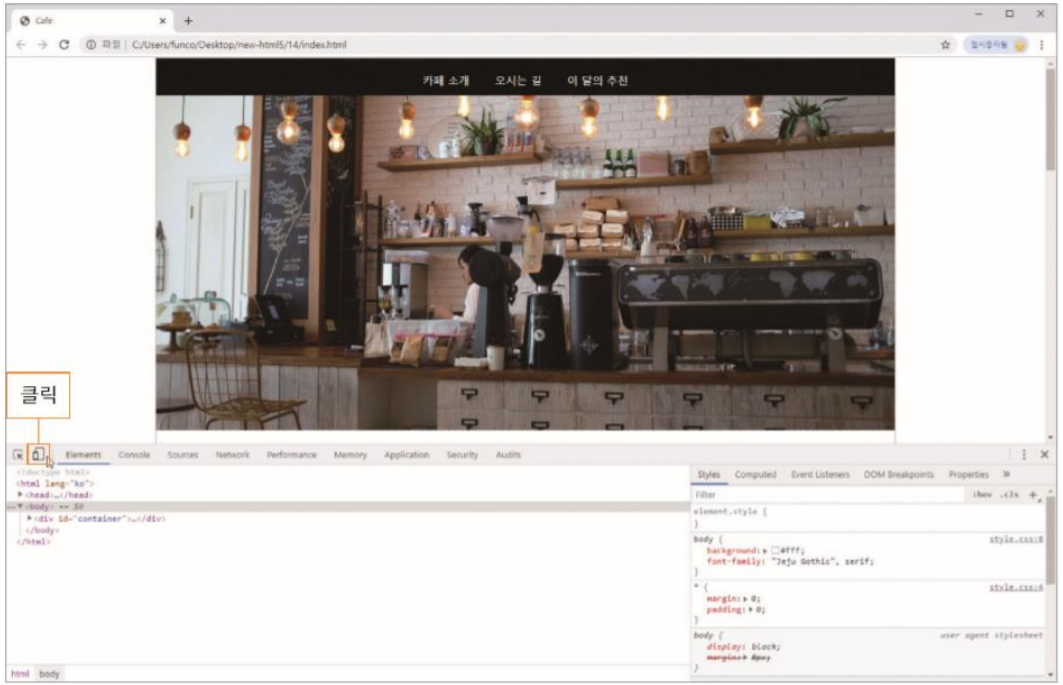
기본형 `<meta name="viewport" content="<속성1=값>, <속성2=값2>, ... ">`

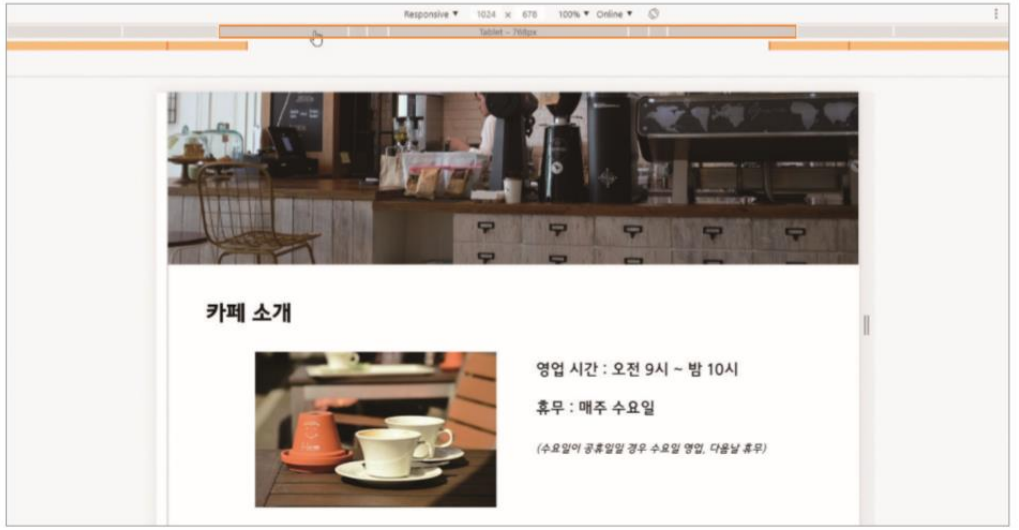
속성	설명	사용 가능한 값	기본 값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본 값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본 값
user-scalable	확대/축소 가능 여부	yes 또는 no	yes
initial-scale	초기 확대/축소 값	1~10	1
minimum-scale	최소 확대/축소 값	0~10	0.25
maximum-scale	최대 확대/축소 값	0~10	1.6

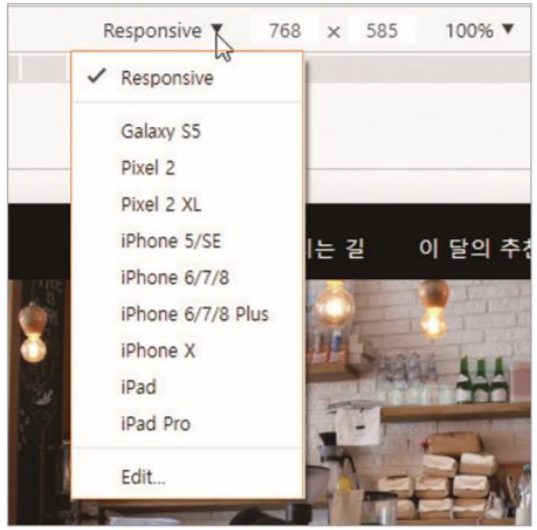
- 일반적인 사용법 : 뷰포트의 너비를 스마트폰 화면 너비에 맞추고 초기 화면 배율을 1로 지정

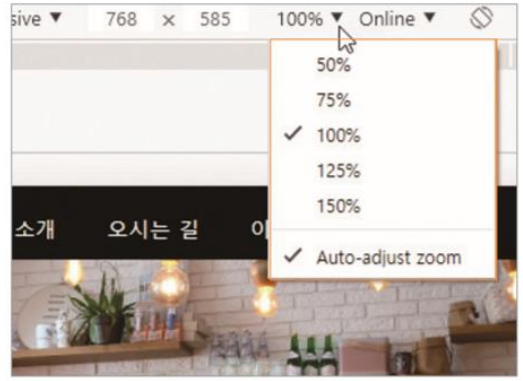
```
<meta name=”viewport” content=”width=device-width, initial-scale=1”>
```

[실습] 크롬의 디바이스 모드 활용하기







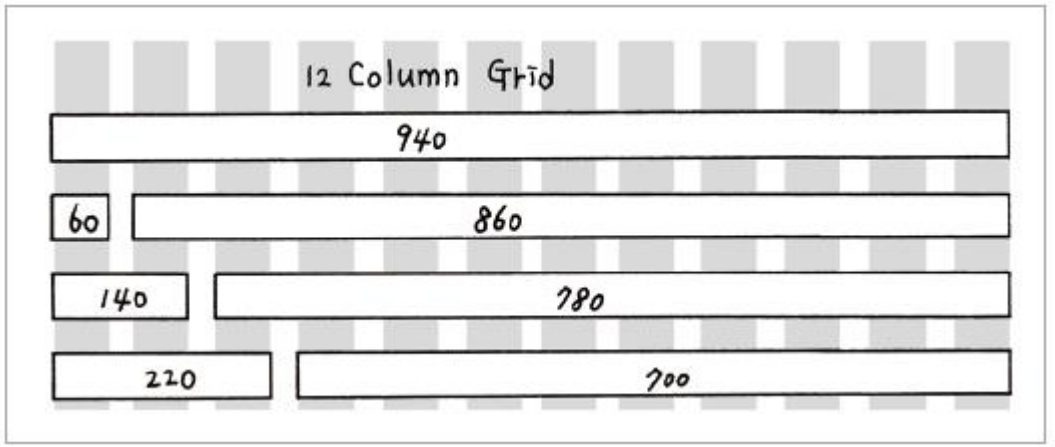


가변 그리드 레이아웃

그리드 시스템(grid system)

- 화면을 여러 개의 칼럼(column)으로 나누어, 필요할 때마다 칼럼들을 묶어 배치하는 방법
- 화면 너비 값에 따라 '960 그리드 시스템', '1200 그리드 시스템' 등으로 나뉨
- 칼럼 개수에 따라 '12 칼럼 그리드 시스템', '16 칼럼 그리드 시스템', '24 칼럼 그리드 시스템' 등으로 나뉨
- 주로 960 픽셀 12 칼럼의 그리드 시스템 사용
- 고정 그리드 : 화면 너비를 일정하게 고정하고 레이아웃 만듦.
- 가변 그리드 : 화면 너비를 % 같은 가변 값으로 지정.
- 가변 그리드 레이아웃을 사용할 경우, 너비 값이 줄어들면 실제 콘텐츠를 확인하기 불편하므로 가능하면 간결한 디자인을 사용하는 것이 좋음.

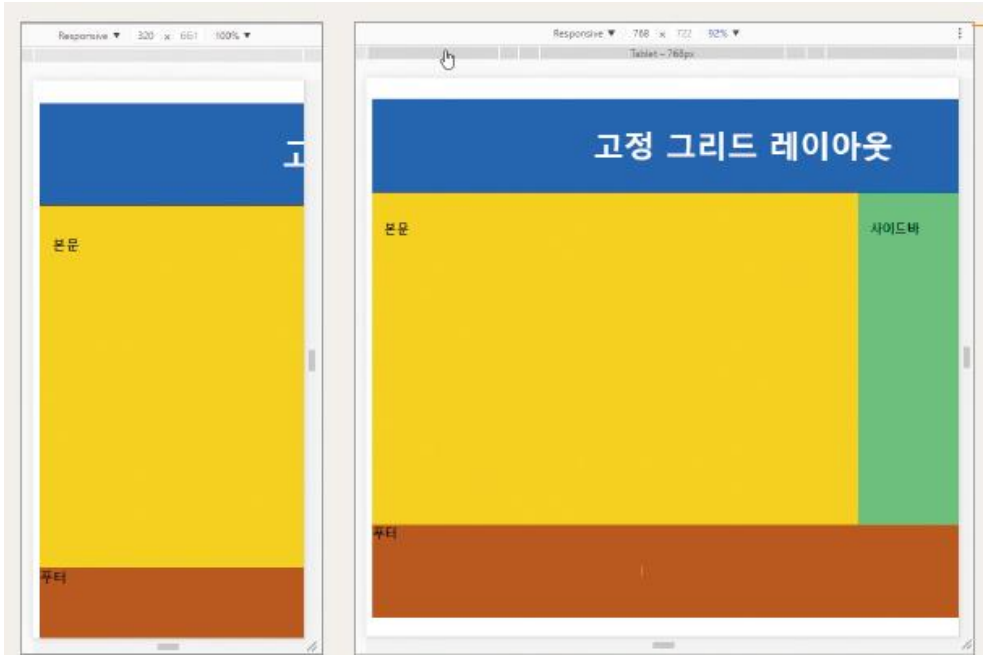
문서 좌우에 10px씩의 패딩이 있다고 가정한 레이아웃 →



가변 그리드 레이아웃

고정 그리드 레이아웃일 경우

- 문서의 맨 바깥 부분을 #wrapper 요소로 묶고 너비를 960px로 지정
- 헤더와 본문, 사이드 바, 푸터를 배치. 이 때 너비는 px 값.
- 화면 너비가 좁아질 경우 내용의 일부가 가려질 수 있음.




```
<style>
#wrapper {
  width:960px;
  margin:0 auto;
}
header {
  width:960px;
  height:120px;
}
.content {
  float:left;
  width:600px;
}
.right-side {
  float:right;
  width:300px;
}
footer {
  clear:both;
  width:960px;
}
</style>
```

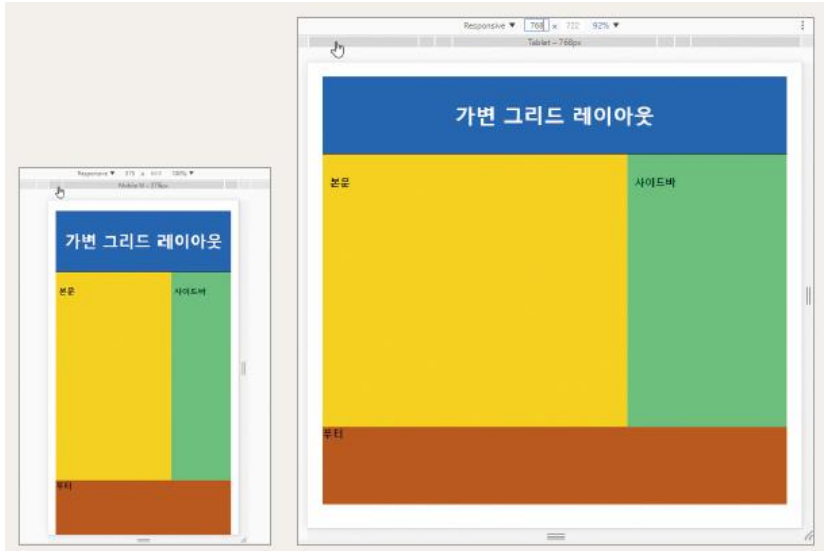
가변 그리드 레이아웃

가변 그리드 레이아웃 만들기

① 전체를 감싸는 요소의 너비를 %로 변환 (화면에 꽉 차게 하고 싶다면 100%, 여유를 두려면 적당히)

② 전체를 감싸는 요소의 너비를 기준으로 각 요소의 너비를 계산

```
(요소의 너비 / 콘텐츠 전체를 감싸는 요소의 너비) * 100
```




```
<style>
#wrapper {
  width:96%;
  margin:0 auto;
}
header {
  width:100%;
  height:120px;
}
.content {
  float:left;
  width:62.5%;
  height:400px;
  padding:1.5625%;
  background-
color:#ffd800;
}
.right-side {
  float:right;
  width:31.25%;
  height:400px;
  padding:1.5625%;
  background-
color:#00ff90;
}
footer {
  clear:both;
  width:100%;
  height:120px;
  background-
color:#c3590a;
}
```

가변 요소

가변 글꼴

em 단위

부모 요소 폰트의 대문자 M 너비를 1em으로 지정. 1em=16px

$$\text{글자 크기(em)} = \frac{\text{글자 크기(px)}}{16\text{px}}$$

```
<style>
  .header-text{ font-size:2em; }
  .content { font-size: 1.5em; }
  .right-side { font-size: 1.5em; }
  footer { font-size: 1.5em; }
</style>
```

rem 단위

- em 단위는 부모 요소가 중첩될 경우 글자 크기가 계속 달라짐
- rem은 처음부터 기본 크기를 지정하고 그것을 기준으로 글자 크기 지정

```
<style>
  body { font-size:16px; }
  .header-text{ font-size:2rem; }
  .fluid-text { font-size:1.5rem; }
</style>
```

가변 요소

가변 이미지

- 브라우저 창의 너비가 변하더라도 이미지 너비 값은 변하지 않음 → 브라우저 화면 너비를 줄일 경우 이미지 일부가 가려짐
- 가변 이미지(fluid image)로 만들면 창의 너비에 따라 이미지 너비도 조절됨

1) CSS를 이용한 방법

이미지를 감싸고 있는 부모 요소만큼만 커지거나 작아지도록 max-width 속성 값을 100%로 지정

```
<style>
  .content img {
    max-width:100%;
    height:auto;
  }
</style>
```

가변 요소

가변 이미지

2) <img> 태그와 srcset 속성

화면 너비 값이나 픽셀 밀도에 따라 고해상도의 이미지 파일 지정 가능

```
기본형  <img src="<이미지>" srcset="<이미지1>[, <이미지2>, <이미지3>, ...]">
```

예)

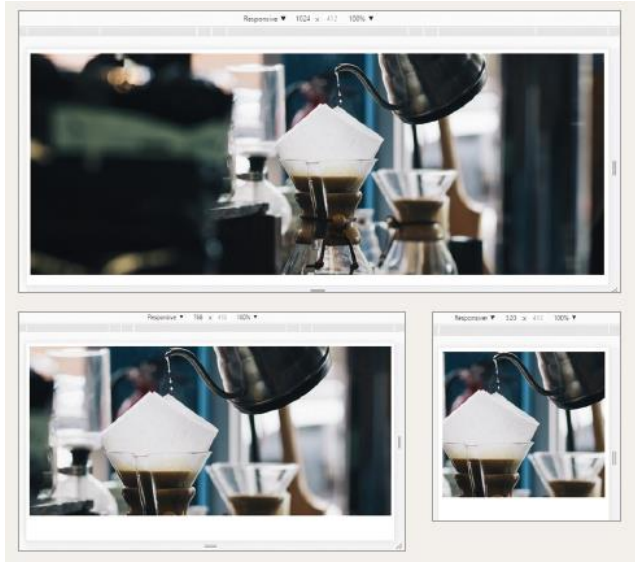
```
<img src="images/pencil.jpg"
srcset="images/pencil-hd.jpg 2x" alt="색연필 제품 이미지">
```

3) <picture> 태그와 <source> 태그

화면 해상도뿐만 아니라 화면 너비에 따라 다른 이미지 파일 표시

속성	설명
srcset	이미지 파일의 경로
media	srcset에 지정한 이미지를 표시하기 위한 조건(속성 값은 14-4 미디어 쿼리 참고).
type	파일 유형
sizes	파일의 크기

```
<picture>
  <source srcset="images/shop-large.jpg" media="(min-width:1024px)">
  <source srcset="images/shop-medium.jpg" media="(min-width:768px)">
  <source srcset="images/shop-small.jpg" media="(min-width:320px)">
  <img src="images/shop.jpg" alt="fill with coffee"
style="width:100%;">
```



가변 요소

가변 비디오

CSS를 사용해 max-width 속성을 100%로 지정

```
<style>
  video { max-width: 100%; }
</style>

<video autoplay loop src=”assets/cars.mp4”> </video>
```