

반응형 웹 디자인

목차

01 반응형 웹 디자인 요소

02 가변 그리드

03 미디어 쿼리

04 반응형 웹 페이지 제작

학습목표

- 반응형 웹 디자인의 동작 원리를 이해한다.
- 가변 그리드 기법을 이해하고 레이아웃에 적용하는 방법을 익힌다.
- 미디어 쿼리의 문법을 이해하고 웹 페이지 제작에 적용하는 방법을 익힌다.
- 가변 그리드 레이아웃과 미디어 쿼리를 이용하여 반응형 웹 페이지를 제작하는 방법을 익힌다.

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 웹 페이지가 모든 화면 크기와 해상도에서 잘 렌더링되도록 하면서도 사용성을 보장하는 웹 디자인 접근 방식
- 이선 마코트(Ethan Marcotte)가 처음 도입

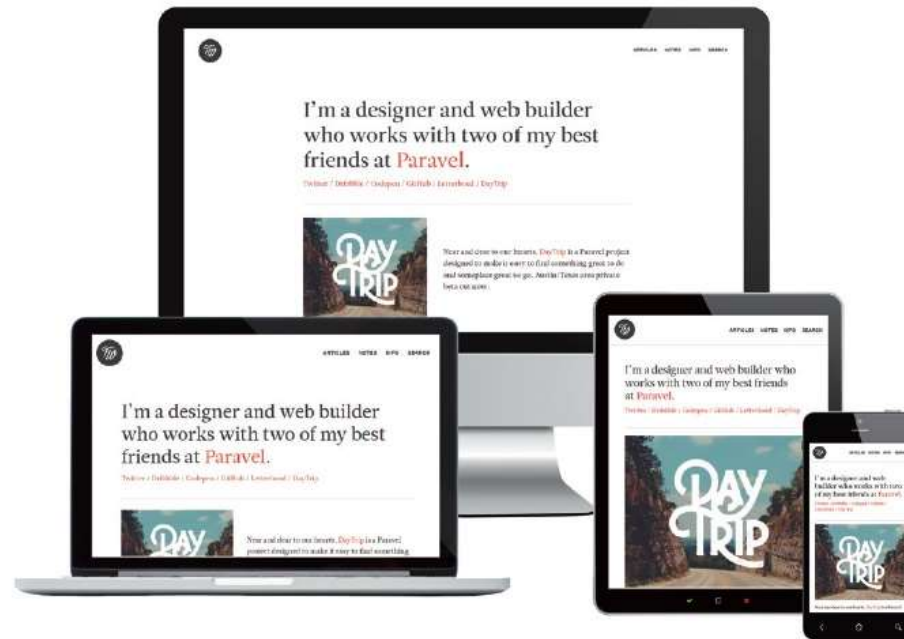


그림 14-1 반응형 웹 디자인이 적용된 사이트의 예(<http://trentwalton.com>)

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 콘텐츠를 보는 데 사용되는 모든 기기에 대응할 수 있는 레이아웃을 만드는 데 사용되는 일련의 모범 사례
- 유지보수가 효율적
- 검색엔진 최적화(SEO)에 유리
- 모바일 환경에서도 적응하게끔 만들어진 반응형 웹은 모바일을 전용으로 하는 사이트에 비해 무겁습니다
- 웹 브라우저는 스펙 및 사양이 제각기 다르기 때문에 하나의 웹 브라우저에서는 잘 반응하던 HTML 소스가 다른 웹 브라우저에서는 디자인이 깨지는 경우가 발생할 수 있습니다.

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 반응형 디자인을 만들기 위해 가장 많이 사용하는 기능은 뷰포트 너비이며, min-width와 max-width, width 등의 미디어 기능을 활용해 뷰포트가 특정 너비 이상 또는 이하인 경우 CSS를 적용할 수 있습니다
- 모바일 브라우저의 뷰를 실제 크기인 1로 설정

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

- 미디어쿼리(Media Query)는 screen과 print로 나뉘는데, 미디어 속성을 사용하여 조건을 지정

```
<link href="css파일이름.css" media="screen and (min-width: 400px) and (max-width: 1000px)" rel="stylesheet">
```

```
<link rel="stylesheet" type="text/css" media="screen" href="cssscreen.css">  
<link rel="stylesheet" type="text/css" media="print" href="cssprint.css">
```

- 미디어 타입(Media Type) - all : 모든 미디어 타입, print : 프린터, screen : 컴퓨터 화면, speech : 스크린 리더

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

```
//미디어 쿼리 구문
@media 미디어 타입 (조건(너비 및 높이)) {
  (CSS 입력하는 부분)
}
```

```
//@import로 미디어 쿼리 적용하기
@import url(cssscreen.css) screen;
@import url(csspprint.css) print;
```

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 스마트폰은 세로 해상도를 기본, 태블릿은 가로로 세로를 기본으로 하고 가로를 추가 한다, 데스크탑은 가로를 기본으로 한다
- 스마트폰은 가로 ~768px(최대), 태블릿 세로인 경우 (최소)768px~992px(최대), 가로화면은 (최소) 992px~1200px(최대), 데스크탑은 최소 1200px~ 로 나누는 경우,

```
@media (max-width: 767px){  
    //모바일  
}  
  
@media (min-width: 768px) and (max-width: 991px) {  
    // 태블릿 세로  
}  
  
@media (min-width: 992px) and (max-width: 1199px) {  
    // 태블릿 가로  
}  
  
@media (min-width: 1200px) {  
    // 데스크탑 일반  
}
```


1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 조건(너비 및 높이) : 반응형 디자인을 만들기 위해 가장 많이 사용하는 기능은 뷰포트 너비이며, min-width와 max-width, width 등의 미디어 기능을 활용해 뷰포트가 특정 너비 이상 또는 이하인 경우 CSS를 적용할 수 있습니다.

```
@media screen and (width: 600px) {  
    body {  
        color : red;  
    }  
}
```

- width(혹은 height)는 브라우저의 창 크기에 사용할 수 있습니다. 따라서 min-혹은 max- 접두사를 붙이게 되면 최솟값인지 최댓값인지 표시할 수 있습니다.

```
@media screen and (max-width: 400px) {  
    body {  
        color: blue;  
    }  
}
```

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 방향성 : 세로 모드인지 가로 모드인지 검사하여 CSS 스타일을 주고 싶은 경우 , orientation으로 검사 할 수 있습니다.

```
@media (orientation: landscape) {  
  body {  
    color: rebeccapurple;  
  }  
}
```

- 논리곱(and) 미디어 쿼리 : and를 사용해 미디어 기능을 합칠 수 있습니다.

```
@media screen and (min-width: 400px) and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인(responsive web design)

- 논리합(or) 미디어 쿼리 : 콤마로 분리를 한다면 미디어 쿼리의 해당 조건 중 어느 하나를 만족시킬 때 CSS 스타일을 적용시킬 수 있습니다.

```
@media screen and (min-width: 600px), screen and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

- 부정(not) 미디어 쿼리 : not 연산자를 사용하게 되면 미디어 쿼리의 의미를 반대로 적용시킵니다.

```
@media not all and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

1. 반응형 웹 디자인 요소

■ 반응형 웹 디자인의 단위

- 고정된 픽셀보다는 크기를 상대적으로 설정할 수 있는 %와 em 단위를 사용

표 14-1 스마트폰 해상도의 예

기종	해상도(px)
갤럭시 S5, 갤럭시 노트 3	1920×1080
갤럭시 S6/S6 엣지, 갤럭시 노트 4/5	2560×1440
아이폰 6	1334×750
아이폰 6 Plus	1920×1080

1. 반응형 웹 디자인 요소

- font-size는 브라우저의 사용자 설정에 따라 달라질 수 있습니다.
- font-size 기본값은 보통 16px로 설정되어 있으며, 이는 1rem 또는 100%에 해당합니다

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <style>
06 body {
07     font-size:100%;
08 }
09 #a {
10     font-size:150%;
11 }
12 #b {
13     font-size:2em;
14 }
15 </style>
16 </head>
17 <body>
18     Hello! I'm Genius!!!
19 <div id="a">Hello! I'm Genius!!!</div>
20 <div id="b">Hello! I'm Genius!!!</div>
21 </body>
22 </html>
```

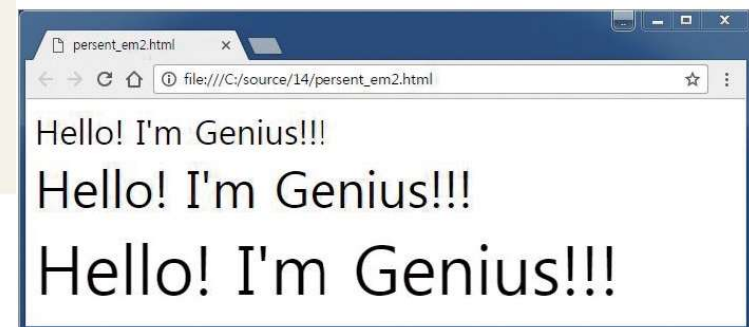
▼ 실행 결과



1. 반응형 웹 디자인 요소

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <style>
06 body {
07     font-size:200%;
08 }
09 #a {
10     font-size:150%;
11 }
12 #b {
13     font-size:2em;
14 }
15 </style>
16 </head>
17 <body>
18     Hello! I'm Genius!!!
19 <div id="a">Hello! I'm Genius!!!</div>
20 <div id="b">Hello! I'm Genius!!!</div>
21 </body>
22 </html>
```

▼ 실행 결과



1. 반응형 웹 디자인 요소

■ 뷰포트(viewport)

- 스마트폰의 브라우저에서 웹 페이지를 보여주는 영역
- <meta> 태그를 이용하여 뷰포트를 설정

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 </head>
06 <body>
07 
08 <h2>속초 대포항</h2>
09 <p>예전에는 배를 댈 공간이 없을 정도로 항구에 어선이 많았고 새벽에 고기를 잡아 돌아온 어선
    으로 북적였다. 항구로 들어오는 진입로 양옆에는 건어물 가게와 횃집이 늘어서 있고, 어판장 쪽에
    는 활어 난전이 형성되어 있었다. 요즘은 동해 고속도로가 개통되어 현대적인 시설의 호텔과 콘도,
    깔끔한 횃집이 많이 들어섰다.</p>
10 </body>
11 </html>
```

브라우저가 사용하는 **고정된 뷰포트 크기**에 따라 렌더링됩니다.
모바일 브라우저에서 **고정 뷰포트 크기(약 980px)**를 기준으로
계산됩니다.



1. 반응형 웹 디자인 요소

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <meta name="viewport" content="width=device-width, initial-scale=1.0">
06 </head>
07 <body>
08 
09 <h2>속초 대포항</h2>
10 <p>예전에는 배를 달 공간이 없을 정도로 항구에 어선이 많았고 새벽에 고기를 잡아 돌아온 어선
    으로 북적였다. 항구로 들어오는 진입로 양옆에는 건어물 가게와 횡집이 늘어서 있고, 어판장 쪽에
    는 활어 난전이 형성되어 있었다. 요즘은 동해 고속도로가 개통되어 현대적인 시설의 호텔과 콘도,
    깔끔한 횡집이 많이 들어섰다.</p>
11 </body>
12 </html>
```

viewport를 기기의 실제 화면 너비로 설정
이미지가 화면 너비에 맞춰 축소되어 화면 가로
너비를 채우게 됩니다.



1. 반응형 웹 디자인 요소

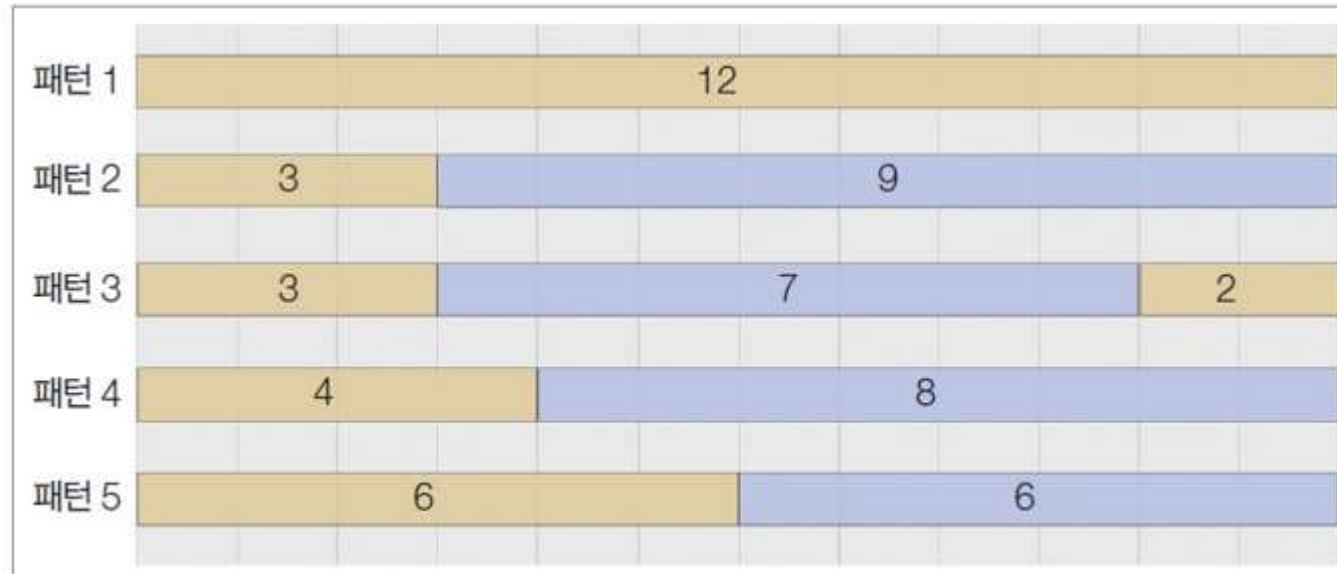
■ 뷰포트의 속성

속성	설명
width	픽셀 단위로 뷰포트의 너비를 설정할 수 있으며 기본 값은 device-width
height	픽셀 단위로 뷰포트의 높이를 설정할 수 있으며 기본 값은 device-height
initial-scale	초기 배율을 나타내는 것으로 1.0이 기본 값이며 0.5는 두 배 축소, 2.0은 두 배 확대를 의미
user-scalable	값을 'yes'로 설정하면 사용자가 화면을 확대 또는 축소할 수 있고 'no'로 설정하면 화면을 확대 또는 축소할 수 없으며 기본 값은 'yes'
minimum-scale	사용자가 축소할 수 있는 최솟값을 설정하며 기본 값은 0.25
maximum-scale	사용자가 확대할 수 있는 최댓값을 설정하며 기본 값은 5.0

2. 가변 그리드

■ 가변 그리드

- 격자를 이용하여 브라우저의 크기에 따라 레이아웃 요소의 너비를 쉽게 설정
- 반응형 웹 페이지는 대부분 그리드를 기반으로 요소를 배치



2. 가변 그리드

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <meta name="viewport" content="width=device-width, initial-scale=1.0">
06 <style>
07 * {
08     box-sizing:border-box;
09     margin:0;
10     padding:0;
11 }
12 ul {
13     list-style-type:none;
14 }
15 body {
16     font-family:"맑은 고딕", "돋움";
17     font-size:100%;
18 }
19 .c1 {width:8.33%; float:left; padding:15px;}
20 .c2 {width:16.66%; float:left; padding:5px;}
21 .c3 {width:25%; float:left; padding:15px;}
22 .c4 {width:33.33%; float:left; padding:15px;}
23 .c5 {width:41.66%; float:left; padding:15px;}
24 .c6 {width:50%; float:left; padding:15px;}
25 .c7 {width:58.33%; float:left; padding:15px;}
26 .c8 {width:66.66%; float:left; padding:15px;}
27 .c9 {width:75%; float:left; padding:15px;}
28 .c10 {width:83.33%; float:left; padding:15px;}
29 .c11 {width:91.66%; float:left; padding:15px;}
30 .c12 {width:100%; float:left; padding:15px;}
31 header {
32     height:80px;
33     padding:20px;
34     border:solid 1px #ff0000;
35     background-color:#66c0e7;
36 }
37 aside {
38     color:#ffffff;
39     border:solid 1px #ff0000;
40 }
41 aside li {
42     padding:10px;
43     margin:5px;
44     background-color:#b52d89;
45 }
46 section#main {
47     padding:15px;
48     border:solid 1px #ff0000;
49 }
50 section#main p {
51     padding:10px;
52     font-size:0.95em;
53     line-height:130%;
54 }
55 footer {
56     height:60px;
57     padding:10px;
58     text-align:center;
59     border:solid 1px #ff0000;
60     background-color:#eeeeee;
61 }
```

2. 가변 그리드

```
62 </style>
63 </head>
64 <body>
65   <header class="c12">
66     <h1>루바토의 사진 아카데미</h1>
67   </header>
68
69   <aside class="c3">
70     <ul>
71
72       <li>사진의 역사</li>
73       <li>노출이란?</li>
74       <li>카메라 동작 모드</li>
75       <li>렌즈의 종류</li>
76     </ul>
77   </aside>
78
79   <section id="main" class="c9">
80     <h1>사진의 역사</h1>
81
82     <p> 사진이 발명되기 이전에도 사람들은 카메라 옵스큐라는 장치로 피사체의 이미지를 얻는 방법을 알고 있었다. 13세기나 14세기 무렵에 발명된 카메라 옵스큐라는 한쪽 면에 구멍이 뚫린 암실 박스로, 암실 박스의 구멍으로 빛이 들어와 박스 안의 반대쪽 벽면에 이미지가 형성된다.</p>
83
84     <p> 아날로그 카메라는 카메라 옵스큐라의 작동 원리를 그대로 따른 것이다. 차이점은 카메라의 경우 암실에 맞힌 상을 뒤집기 위해 거울을 사용하고, 필름을 이용하여 이미지를 저장한다는 것이다.</p>
85
86     <p> 그 후 카메라 옵스큐라는 발전을 거듭하여 오늘날의 카메라에 이르게 되었다. 이제 예술의 한 장르로 인정받는 사진은 우리의 생활 곳곳에 스며들어 있다.</p>
87   </section>
88
89   <footer class="c12">
90     COPYRIGHT(C) 루바토 All Rights Reserved.
91   </footer>
92 </body>
93 </html>
```

▼ 실행 결과



3. 미디어 쿼리

■ 미디어 쿼리(media query)

- 웹 페이지가 사용되는 미디어의 유형에 따라 CSS를 다르게 적용하는 기술
- 미디어 유형으로는 print, screen, tv, all 등이 있음.

```
<!-- 생략 -->
05 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
06 <style>
07 body {
08     background-color:#00ff00; /* 초록색 */
09 }
10
11 @media only screen and (max-width: 600px) {
12     body {
13         background-color:#ffff00; /* 노란색 */
14     }
15 }
16 </style>
17 </head>
18 <body>
19
20 </body>
21 </html>
```

▼ 실행 결과



(a) 해상도 너비가 600픽셀을 초과하는 경우



3. 미디어 쿼리

```
<!-- 생략 -->
05 <meta name="viewport" content="width=device-width, initial-scale=1.0">
06 <style>
07 * {
08     box-sizing:border-box;
09     margin:0;
10     padding:0;
11 }
12 ul {
13     list-style-type:none;
14 }
15 body {
16     font-family:"맑은 고딕", "돋움";
17     font-size:100%;
18 }
19 /* 너비가 768픽셀을 초과하는 데스크톱 컴퓨터 등 */
20 .c1 {width:8.33%; float:left; padding:15px;}
21 .c2 {width:16.66%; float:left; padding:15px;}
22 .c3 {width:25%; float:left; padding:15px;}
23 .c4 {width:33.33%; float:left; padding:15px;}
24 .c5 {width:41.66%; float:left; padding:15px;}
25 .c6 {width:50%; float:left; padding:15px;}
26 .c7 {width:58.33%; float:left; padding:15px;}
27 .c8 {width:66.66%; float:left; padding:15px;}
28 .c9 {width:75%; float:left; padding:15px;}
29 .c10 {width:83.33%; float:left; padding:15px;}
30 .c11 {width:91.66%; float:left; padding:15px;}
31 .c12 {width:100%; float:left; padding:15px;}
32 /* 너비가 768픽셀 이하인 스마트폰 */
```

```
33 @media only screen and (max-width: 768px) {
34     [class*="c"] {
35         width:100%;
36     }
37 }
38 header {
39     height:80px;
40     padding:20px;
41     background-color:#66c0e7;
42 }
43 aside {
44     color:#ffffff;
45 }
46 aside li {
47     padding:10px;
48     margin:5px;
49     background-color:#b52d89;
50 }
51 section#main {
52     padding:15px;
53 }
54 section#main p {
55     padding:10px;
56     font-size:0.95em;
57     line-height:130%;
58 }
59 footer {
60     height:60px;
61     padding:10px;
62     text-align:center;
63     background-color:#eeeeee;
64 }
```

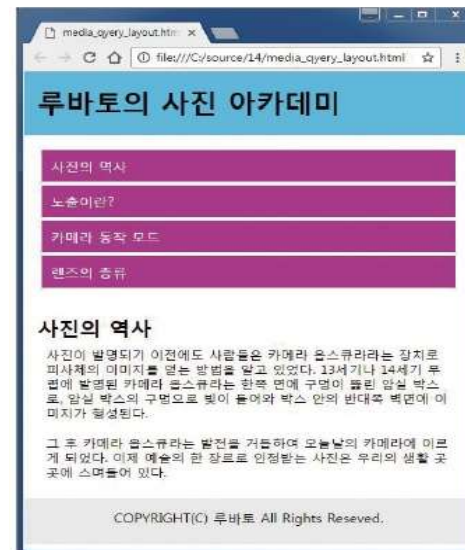
3. 미디어 쿼리

```
65 </style>
66 </head>
67 <body>
68   <header class="c12">
69     <h1>루바토의 사진 아카데미</h1>
70   </header>
71
72   <aside class="c3">
73     <ul>
74       <li>사진의 역사</li>
75       <li>노출이란?</li>
76       <li>카메라 동작 모드</li>
77       <li>렌즈의 종류</li>
78     </ul>
79   </aside>
80
81   <section id="main" class="c9">
82     <h2>사진의 역사</h2>
83     <p>사진이 발명되기 이전에도 사람들은 카메라 옵스큐라라는 장치로 피사체의 이미지를 얻는 방법을 알고 있었다. 13세기나 14세기 무렵에 발명된 카메라 옵스큐라는 한쪽 면에 구멍이 뚫린 암실 박스로, 암실 박스의 구멍으로 빛이 들어와 박스 안의 반대쪽 벽면에 이미지가 형성된다.</p>
84     <p>그 후 카메라 옵스큐라는 발전을 거듭하여 오늘날의 카메라에 이르게 되었다. 이제 예술의 한 장르로 인정받는 사진은 우리의 생활 곳곳에 스며들어 있다.</p>
85   </section>
86
87   <footer class="c12">
88     COPYRIGHT(C) 루바토 All Rights Reserved.
89   </footer>
90 </body>
91 </html>
```

▼ 실행 결과



(a) 해상도 너비가 768픽셀을 초과하는 경우



(b) 해상도 너비가 768픽셀 이하인 경우

4. 반응형 웹 페이지 제작

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="utf-8">
05 <meta name="viewport" content="width=device-width, initial-scale=1.0">
06 <style>
07 * {
08     box-sizing:border-box;
09     margin:0;
10     padding:0;
11 }
12 ul {
13     list-style-type:none;
14 }
15 body {
16     font-family:"맑은 고딕", "돋움";
17     font-size:100%;
18 }
19 [class*="c_"] {
20     float:left;
21     padding:10px;
22 }
23 /* 너비가 600픽셀 미만인 스마트폰을 위한 기본 설정 */
24 [class*="c_"] {
25     width:100%;
26 }
27 /* 너비가 600픽셀 이상인 태블릿을 위한 설정 */
28 @media only screen and (min-width: 600px) {
29     .c_m_1 {width:8.33%;}
```


4. 반응형 웹 페이지 제작

```
30 .c_m2 {width:16.66%;}
31 .c_m3 {width:25%;}
32 .c_m4 {width:33.33%;}
33 .c_m5 {width:41.66%;}
34 .c_m6 {width:50%;}
35 .c_m7 {width:58.33%;}
36 .c_m8 {width:66.66%;}
37 .c_m9 {width:75%;}
38 .c_m10 {width:83.33%;}
39 .c_m11 {width:91.66%;}
40 .c_m12 {width:100%;}
41 }
42 /* 너비가 768픽셀 이상인 데스크톱 컴퓨터, 노트북 등 */
43 @media only screen and (min-width: 768px) {
44 .c_1 {width:8.33%; }
45 .c_2 {width:16.66%; }
46 .c_3 {width:25%; }
47 .c_4 {width:33.33%; }
48 .c_5 {width:41.66%; }
49 .c_6 {width:50%; }
50 .c_7 {width:58.33%; }
51 .c_8 {width:66.66%; }
52 .c_9 {width:75%; }
53 .c_10 {width:83.33%; }
54 .c_11 {width:91.66%; }
55 .c_12 {width:100%; }
56 }
57 header {
58     height:80px;
59     padding:20px;
60     background-color:#66c0e7;
61 }
```

4. 반응형 웹 페이지 제작

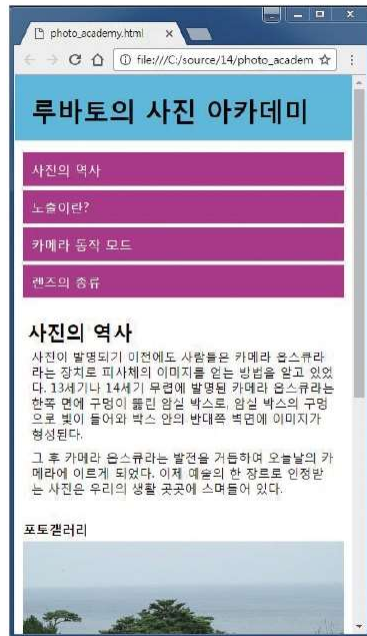
```
62 aside#menu {
63     color:#ffffff;
64 }
65 aside#menu li {
66     padding:10px;
67     margin:5px 0 0 0;
68     background-color:#b52d89;
69 }
70 section#main {
71     padding:15px;
72 }
73 section#main p {
74     padding:5px;
75     font-size:0.95em;
76     line-height:130%;
77 }
78 aside#photos img {
79     width:100%;
80 }
81 aside#photos li {
82     margin:5px 0 0 0;
83 }
84 footer {
85     height:60px;
86     padding:10px;
87     clear:both;
88     text-align:center;
89     background-color:#eeeeee;
90 }
91 </style>
92 </head>
93 <body>
94     <header class="c12">
95         <h1>루바토의 사진 아카데미</h1>
```

4. 반응형 웹 페이지 제작

```
96 </header>
97
98 <aside id="menu" class="c_m3 c_3">
99   <ul>
100     <li>사진의 역사</li>
101     <li>노출이란?</li>
102     <li>카메라 동작 모드</li>
103     <li>렌즈의 종류</li>
104   </ul>
105 </aside>
106
107 <section id="main" class="c_m9 c_6">
108   <h1>사진의 역사</h1>
109   <p> 사진이 발명되기 이전에도 사람들은 카메라 옵스큐라는 장치로 피사체의 이미지를 얻는 방
    법을 알고 있었다. 13세기나 14세기 무렵에 발명된 카메라 옵스큐라는 한쪽 면에 구멍이 뚫린 암
    실 박스로, 암실 박스의 구멍으로 빛이 들어와 박스 안의 반대쪽 벽면에 이미지가 형성된다.</p>
110   <p> 그 후 카메라 옵스큐라는 발전을 거듭하여 오늘날의 카메라에 이르게 되었다. 이제 예술의
    한 장르로 인정받는 사진은 우리의 생활 곳곳에 스며들어 있다.</p>
111 </section>
112 <aside id="photos" class="c_m12 c_3">
113   <h4>포토갤러리</h4>
114   <ul>
115     <li></li>
116     <li></li>
117   </ul>
118 </aside>
119
120 <footer class="c12">
121   COPYRIGHT(C) 루바토 All Rights Reseved.
122 </footer>
123 </body>
124 </html>
```

4. 반응형 웹 페이지 제작

▼ 실행 결과



(a) 기기 너비가 600픽셀 미만인 경우



(b) 기기 너비가 600픽셀 이상, 768픽셀 미만인 경우



(c) 기기 너비가 768픽셀 이상인 경우

동적 그리드 시스템 (Dynamic Grid System)

- 고정된 폭이 아니라 콘텐츠나 뷰포트의 크기에 따라 유동적으로 변경되는 그리드입니다.
- 비율(%), flexible units (em, rem, fr) 등을 활용하여 자동으로 크기를 조정합니다.
- 고정 폭이 아니라 상황에 따라 자동 크기 조절

```
.container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
}
```

/*

요소 크기가 200px 이상이면 여러 개, 작으면 하나만 나오게 됩니다 (자동으로 늘어남)*/

반응형 그리드 시스템 (Responsive Grid System)

- 반응형 그리드 시스템 (Responsive Grid System)
- 뷰포트 크기(브라우저 너비)에 따라 레이아웃이 바뀌는 시스템
- 미디어 쿼리(@media)를 사용해서 화면 크기별로 다른 그리드 설정을 적용합니다.

```
@media (min-width: 768px) {  
  .column {  
    width: 50%;  
  }  
}  
  
@media (max-width: 767px) {  
  .column {  
    width: 100%;  
  }  
}  
/* 화면이 크면 2열, 작으면 1열로 바뀌는 구조. */
```

CSS3 Grid Layout (Grid 시스템의 최신 표준)

- 2차원 레이아웃을 만들 수 있는 CSS의 공식 레이아웃 시스템
- 행(rows)과 열(columns)을 동시에 제어 가능
- 2017년부터 브라우저에 정식 지원
- display: grid 사용
- grid-template-rows, grid-template-columns로 영역 정의
- grid-area, grid-row, grid-column으로 요소 배치
- 미디어 쿼리와 함께 사용하면 반응형 그리드도 가능
- fr, minmax(), auto-fit 등 다양한 유닛 사용

CSS3 Grid Layout (Grid 시스템의 최신 표준)

■

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr); /* 3개의 같은 크기 칼럼 */  
  gap: 10px;  
}  
  
.item {  
  background: lightblue;  
}
```

```
<div class="container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
</div>  
<!--  
.container는 3열 그리드를 구성하고 .item들이 자동으로 배치됨  
-->
```


Bootstrap

5. Bootstrap

■ Bootstrap

- CSS, JavaScript, HTML로 구성된 프론트엔드 프레임워크
- 반응형 웹 개발을

5. Bootstrap

■ Bootstrap

	Bootstrap 3	Bootstrap 4	Bootstrap 5
출시 연도	2013	2018	2021
레이아웃	Float 기반	Flexbox 기반 Grid	
css preprocessor	LESS 사용	Sass(SCSS) 사용	Sass(SCSS)
jQuery 의존성	필수	필수	제거됨(vanilla JS 사용)
유틸리티 클래스	제한적	강화됨(.m-3, .p-2 등)	대폭증가, 커스텀 유틸리티
브라우저 지원	IE8 이상	IE10이상	IE11 지연중단
Color System	제한적	\$theme-colors 변수	CSS 변수 지원, 테마 확장 쉬움
Grid System	4단계(xs, sm, md, lg)	5단계(xs, sm, md, lg, xl)	6단계(xxl)까지
Icon	Glyphicons 내장	제거됨(외부사용)	제거됨

5. Bootstrap

■ Bootstrap 설치

방법1 : HTML 파일의 <head> 태그 CDN 링크 선언 : 인터넷 연결이 필요하며, 설치 없이 빠르게 시작할 수 있습니다.

```
<!-- Latest compiled and minified CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">

<!-- Latest compiled JavaScript -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"> </script>
```

방법2 : npm 또는 yarn 설치 (프론트엔드 프로젝트)

```
npm install bootstrap
```

```
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min.js';
```

방법3 : <http://getbootstrap.com>에서 파일을 다운로드하여 프로젝트에 추가

5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap은 HTML 구조를 기반으로 작동하므로, 아래의 기본 HTML 템플릿을 사용합니다

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <title>Bootstrap Example</title>
</head>
<body>
  <div class="container">
    <h1 class="text-center">Hello, Bootstrap!</h1>
  </div>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"> </script>
</body>
</html>
```

5. Bootstrap

■ Bootstrap HTML 기본 구조

부트스트랩은 그리드 시스템 기능을 사용하여 반응형 레이아웃을 쉽게 만들 수 있습니다.

```
<div class="container">
  <div class="row">
    <div class="col">Column 1</div>
    <div class="col">Column 2</div>
    <div class="col">Column 3</div>
  </div>
</div>
```

```
<div class="container">
  <div class="row">
    <div class="col-sm-6 col-md-4">Small and Medium</div>
    <div class="col-sm-6 col-md-4">Small and Medium</div>
    <div class="col-sm-12 col-md-4">Small Full, Medium Half</div>
  </div>
</div>
```

5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap 은 버튼, 네비게이션 바, 모달 등 다양한 기본 UI 컴포넌트를 제공합니다.

```
<button class="btn btn-primary">Primary Button</button>
<button class="btn btn-secondary">Secondary Button</button>
```

```
<!-- 네비게이션 바 -->
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Brand</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav">
    <span class="navbar-toggler-icon"> </span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
    </ul>
  </div>
</nav>
```

5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap 은 버튼, 네비게이션 바, 모달 등 다양한 기본 UI 컴포넌트를 제공합니다.

```
<!-- 모달 (팝업창) -->
<!-- Button to trigger modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-
target="#myModal">
  Launch Modal
</button>
<!-- Modal Structure -->
<div class="modal fade" id="myModal" tabindex="-1" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal title</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body">
        This is a modal body.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```


5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap 은 미디어 쿼리를 통해 화면 크기에 따라 다르게 동작하도록 도와줍니다.

```
<div class="d-none d-md-block">Desktop Only</div>  
<div class="d-block d-md-none">Mobile Only</div>
```

5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap 은 기본 스타일을 커스터마이징 할 수 있습니다

```
<style>
  .btn-primary {
    background-color: #ff5733; /* 새로운 색상 */
  }
</style>
```

부트스트랩 소스 파일을 가져와 SASS를 사용해 수정할 수 있습니다.

5. Bootstrap

■ Bootstrap HTML 기본 구조

Bootstrap 은 JavaScript 기반 컴포넌트(모달, 드롭다운 등)에 데이터 속성으로 간단히 액션을 설정할 수 있으며, 필요에 따라 직접 제어도 가능합니다

```
var myModal = new bootstrap.Modal(document.getElementById('myModal'));  
myModal.show();
```