

## 1. Grid System

- 1)Bootstrap은 반응형 Web Design을 기반으로 쉽고 강력한 layout 기능을 제공한다.
- 2)Bootstrap은 기기별로 각각 class를 나누어서 설정해 사용할 수 있다.
- 3)Handphone과 같은 작은 device부터 Tablet과 Desktop 그 이상의 대형 device까지 지원한다.

## 2. Media Queries

## 1)CSS를 활용하여 반응형 Web Coding하기

## ①meta tag 포함하기

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## ②다양한 Device size 별 Media query 예시

```
@media screen and (min-width: 1600px) { ... }
@media screen and (min-width: 1280px) and (max-width:1599px) { ... }
@media screen and (min-width: 1024px) and (max-width:1279px) { ... }
@media screen and (min-width: 780px) and (max-width:1023px) { ... }
@media screen and (min-width: 640px) and (max-width:779px) { ... }
@media screen and (min-width: 480px) and (max-width:639px) { ... }
@media screen and (min-width: 479px) { ... }
```

## ③원하는 Device 선택하여 Media query별 Style coding 하기

```
/*Device 장치가 최소 1200 이상일 때 : Laptop Computer 큰 Monitor 用 */
@media min-width: 1200px) {
    html, body { background-color:purple}
}
/*Device 장치가 최소 992 이상일 때 : Laptop Computer 작은 Monitor 用 */
@media min-width: 992px) {
    #wrap{width:80%}
}
/*Device 장치가 최소 768 이상일 때 : Tablet 用 */
@media min-width: 768px) {
    html, body { background-color:orange}
    #content_wrap #content{font-size:1.5em;}
}
/*Device 장치가 최소 767 이하일 때 : Smart Phone 用 */
@media max-width: 767px) {
    html, body { background-color:pink;}
    #content_wrap #content{font-size:2em;}
    nav ul li{width:100%}
}
```

## 2)Lab

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Bootstrap Hello, World </title>
<style>
    * {padding : 0; margin:0;}
    body{
        background-color: pink;
    }
    #wrapper{
```

```
59         border:1px solid gray;
60         width:80%;
61         margin:0 auto;
62         font-size:1.5vw; /*화면의 사이즈가 커짐에 따라 글씨의 크기도 같이 커진다. */
63     }
64     nav li{
65         list-style-type: none;
66         float: left;
67         border: 1px solid gray;
68         width:20%;
69         box-sizing: border-box;
70         text-align: center;
71     }
72     nav ul{
73         overflow: hidden;
74     }
75     header h1{
76         text-align: center;
77         margin-bottom: 20px;
78     }
79     header{
80         padding:20px 0 0 0;
81         border:1px solid gray;
82     }
83     .content{
84         padding:5px;
85     }
86     footer{
87         border:1px solid gray;
88         height:100px;
89         line-height: 100px;
90         text-align: center;
91     }
92     @media(max-width:767px){
93         body{
94             background-color: yellow;
95         }
96         #wrapper{
97             width:100%;
98             font-size: 2em;
99         }
100        nav li{
101            width:100%;
102        }
103    }
104    @media(min-width:768px){
105        body{
106            background-color: orange;
107        }
108        #wrapper{ /* 992px 이하일 때는 100% */
109            width:100%;
110        }
111    }
112    @media(min-width: 992px){
113        body{
114            background-color: lightblue;
115        }
116        #wrapper{ /* 992px 이상일 때는 80% */
```

```

117         width:80%;
118     }
119 }
120 @media(min-width:1200px){
121     body{
122         background-color:rgba(221, 96, 200, 0.664)
123     }
124     #wrapper{
125         width:80%;
126     }
127 }
128 </style>
129 </head>
130 <body>
131     <div id="wrapper">
132         <header>
133             <h1>반응형 웹</h1>
134             <nav>
135                 <ul>
136                     <li>menu1</li>
137                     <li>menu2</li>
138                     <li>menu3</li>
139                     <li>menu4</li>
140                     <li>menu5</li>
141                 </ul>
142             </nav>
143         </header>
144         <div class="content">
145             <h2>반응형 Web은 Media Query를 사용하여 Styling 합니다.</h2>
146             <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Voluptatibus, ut fugit
147             officiis similique aperiam eos esse consectetur, sed ad ab omnis. Voluptatibus
148             doloremque officiis eos cupiditate earum facere accusamus consequuntur.</p>
149         </div>
150         <footer>
151             <h1>designed by sj</h1>
152         </footer>
153     </div>
154 </body>
155 </html>

```

### 3. Bootstrap에서는 구조의 최상위 Container 요소 안쪽에 div.row로 행을 만든다.

#### 1).row

-.container 또는 .container-fluid 안에 .row로 행을 만든다.

#### 2).col-\*-\*

-.row안에 .col-\*-\*로 열을 만든다.

-첫번째 \*에는 xs, sm, md, lg 중의 하나가, 두번째 \*에는 1부터 12까지의 수 중의 하나가 들어간다.

-xs : 항상 적용된다.

-sm : 가로 해상도 768px 이상에서 적용

-md : 가로 해상도 992px 이상에서 적용

-lg : 가로 해상도 1200px 이상에서 적용

-1 ~ 12 : 행을 12등분하여 그중 몇 개를 사용할지 결정.

#### 3)Lab

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

173 <meta name="viewport" content="width=device-width, initial-scale=1">
174 <title>Bootstrap 101 Template</title>
175 <!-- Bootstrap -->
176 <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
    integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l
    9a69psu" crossorigin="anonymous">
177 <style>
178     div {border : 1px solid gray;}
179 </style>
180 </head>
181 <body>
182     <div class="container">
183         <div class="row">
184             <div class="col-xs-6">div.col-xs-6</div>
185             <div class="col-xs-4">div.col-xs-4</div>
186             <div class="col-xs-2">div.col-xs-2</div>
187         </div>
188     </div>
189     <div class="container-fluid">
190         <div class="row">
191             <div class="col-xs-6">div.col-xs-6</div>
192             <div class="col-xs-4">div.col-xs-4</div>
193             <div class="col-xs-2">div.col-xs-2</div>
194         </div>
195     </div>
196
197     ...
198 </body>
199

```

-위의 예는 6:4:2일 경우,  $6+4+2=12$ 이므로, 행을 가득 채운다.

-만일 5:5의 경우,  $5+5=10$ 이므로, 오른쪽에 남는 부분이 생긴다.

-만일 5:5:4의 경우,  $5+5+4=14$ 이므로, 12를 넘게 만드는 마지막 열이 다음 줄로 넘어가게 된다.

#### 4).col-sm-\*

-가로 해상도가 768px이상에서 적용된다.

-예를 들어 .col-sm-6은 가로 해상도 768px이상에서 행의 6/12을 가로 크기로 하겠다는 뜻

-만약 가로 해상도가 768px 미만이라면 행 전체를 가로 크기로 한다.

```

209 <div class="container">
210     <div class="row">
211         <div class="col-sm-6">div.col-sm-6</div>
212         <div class="col-sm-6">div.col-sm-6</div>
213     </div>
214     <div class="row">
215         <div class="col-sm-8">div.col-sm-8</div>
216         <div class="col-sm-4">div.col-sm-4</div>
217     </div>
218     <div class="row">
219         <div class="col-sm-8">div.col-sm-8</div>
220         <div class="col-sm-4">div.col-sm-4</div>
221     </div>
222 </div>
223

```

#### 5).col-md-\*

-가로 해상도 992px 이상에서 적용된다.

-예를 들어 .col-md-6은 가로 해상도 992px 이상에서 행의 6/12을 가로 크기로 하겠다는 뜻

-만약 가로 해상도가 992px 미만이라면 행 전체를 가로 크기로 한다.

228  
229 6).col-lg-\*  
230 -가로 해상도 1200px 이상에서 적용  
231 -예를 들어 .col-lg-6은 가로 해상도 1200px 이상에서 행의 6/12을 가로 크기로 하겠다는 뜻  
232 -만약 가로 해상도가 1200px 미만이라면 행 전체를 가로 크기로 한다.  
233  
234

#### 235 4. Grid Systems

236 /\* Extra small devices (phones, less than 768px) \*/  
237 /\* No media query since this is the default in Bootstrap \*/  
238  
239 /\* Small devices (tablets, 768px and up) \*/  
240 @media (min-width: @screen-sm-min) { ... }  
241  
242 /\* Medium devices (desktops, 992px and up) \*/  
243 @media (min-width: @screen-md-min) { ... }  
244  
245 /\* Large devices (large desktops, 1200px and up) \*/  
246 @media (min-width: @screen-lg-min) { ... }  
247

248 1) 매우 작은 기기(Mobile Phone) : < 768px  
249 -Grid 적용 : 항상  
250 -Container class 최대너비 : None(auto)  
251 -Class 접두사 : .col-xs-  
252 -Column 수 : 12  
253 -Column 최대 너비 : Auto  
254 -사이 너비 : 30px (Column의 양쪽에 15px 씩) --> 이를 테면 margin을 의미  
255 -중첩 : Yes  
256 -Offset : Yes  
257 -Column 순서 : Yes  
258

259 2) 작은 기기(태블릿) : >= 768px  
260 -Grid 적용 : 분기점보다 크면 적용  
261 -Container class 최대너비 : 750px  
262 -Class 접두사 : .col-sm-  
263 -Column 수 : 12  
264 -Column 최대 너비 : 62px  
265 -사이 너비 : 30px (Column의 양쪽에 15px 씩)  
266 -중첩 : Yes  
267 -Offset : Yes  
268 -Column 순서 : Yes  
269

270 3) 중간 기기(데스크탑) : >= 992px  
271 -Grid 적용 : 분기점보다 크면 적용  
272 -Container class 최대너비 : 970px  
273 -Class 접두사 : .col-md-  
274 -Column 수 : 12  
275 -Column 최대 너비 : 81px  
276 -사이 너비 : 30px (Column의 양쪽에 15px 씩)  
277 -중첩 : Yes  
278 -Offset : Yes  
279 -Column 순서 : Yes  
280

281 4) 큰 기기(데스크탑) : >= 1200px  
282 -Grid 적용 : 분기점보다 크면 적용  
283 -Container class 최대너비 : 1170px  
284 -Class 접두사 : .col-lg-  
285 -Column 수 : 12

```

286 -Column 최대 너비 : 97px
287 -사이 너비 : 30px (Column의 양쪽에 15px 씩)
288 -중첩 : Yes
289 -Offset : Yes
290 -Column 순서 : Yes
291
292 5)12칸으로 나누는 이유는 보통 1/2, 1/3, 1/4, 1/6 으로 나누기 때문에 이 나눔의 최소공배수인 12임.
293
294
295 5. Grid System > .col-*-* 여러 개 사용하기
296 1).col-*-*을 여러 개 사용하면 가로 해상도에 따라 열의 가로 크기를 다르게 정할 수 있음.
297
298 2)예
299 -<div class="col-xs-4 col-sm-6">
300 -가로 해상도 767px 이하에서는 col-xs-4가 적용되어 4/12
301 -786px 이상에서는 col-sm-6이 적용되어 6/12를 열의 가로 크기로 한다.
302
303 3)Lab
304 <!DOCTYPE html>
305 <html lang="en">
306 <head>
307 <meta charset="UTF-8">
308 <meta name="viewport" content="width=device-width, initial-scale=1.0">
309 <title>Document</title>
310 <link rel="stylesheet"
311 href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
312 integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg
313 119a69psu" crossorigin="anonymous">
314 <style>
315 div {
316 border:1px solid gray;
317 }
318 </style>
319 </head>
320 <body>
321 <div class="container-fluid">
322 <div class="row">
323 <div class="col-xs-8 col-sm-6 col-md-4 col-lg-3">col-xs-8 col-sm-6 col-md-4
324 col-lg-3</div>
325 <div class="col-xs-4 col-sm-6 col-md-8 col-lg-9">col-xs-4 col-sm-6 col-md-8
326 col-lg-9</div>
327 </div>
328 </div>
329 </body>
330 </html>
331 -xs에서는 8:4, sm에서는 6:6, md에서는 4:8, lg에서는 3:9
332 -Broser에서 좌우 화면의 넓이를 조절해가며 확인하기
333
334 4)Lab
335 <div class="container-fluid">
336 <div class="row">
337 <div class="col-sm-6 col-md-4 col-lg-8">col-xs-8 col-sm-6 col-md-4 col-lg-8</div>
338 <div class="col-sm-6 col-md-8 col-lg-4">col-xs-4 col-sm-6 col-md-8 col-lg-4</div>
339 </div>
340 </div>
341 -xs에서는 단일 Column으로 나타남
342
343 5)Layout 다루기

```

339 -Bootstrap의 Grid system을 이용하면 HTML 요소에 적절한 class 값만 추가하여 layout을 만들 수 있다.  
 340 -기본 원칙은 다음과 같다.  
 341 --.container 또는 .container-fluid 바로 아래에 .row를 넣는다.  
 342 --.row 바로 아래에 .col-\*-\*를 넣는다.  
 343 --.container를 사용하면서 좌우로 꽉 찬 배경을 넣으려면, .container를 감싸는 새로운 요소를 만든다.  
 344 -Bootstrap Homepage에 다양한 예제가 있는데, 처음 볼 때는 분석이 어려울 수 있다.  
 345 -아래에 있는 단순한 형태의 layout을 이해하면 예제 분석이 훨씬 쉬워질 것이다.

## 6)Lab

347 -전체를 감싸는 요소는 좌우로 꽉 찬 요소(container 혹은 container-fluid)를 사용하고 각 행별로 .row를 사용  
 348 하여 가로 줄을 계속 만들어주며 그 안에서 요소의 layout이 12분의 몇을 차지하는지 col-\*-\*를 사용하여  
 contents를 감싸주고 그 안에서 text를 넣어준다.  
 349 -글자크기는 h1 ~ h6 class명을 활용하면 글자크기 style 변경이 가능하다.  
 350 -<https://getbootstrap.com/docs/3.4/css/#type>  
 351 -나머지 style은 css로 만든다.

```

352 <!DOCTYPE html>
353 <html lang="en">
354 <head>
355   <meta charset="UTF-8">
356   <meta name="viewport" content="width=device-width, initial-scale=1.0">
357   <title>Document</title>
358   <link rel="stylesheet"
359     href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
360     integrity="sha384-HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg
361     119a69psu" crossorigin="anonymous">
362   <style>
363     h1 {
364       border:1px solid gray;
365     }
366     .row:nth-of-type(1){
367       background-color: #02CBFB;
368     }
369     .row:nth-of-type(2){
370       background-color: #81C1E1;
371     }
372     .row:nth-of-type(3){
373       background-color: #AFF3FC;
374     }
375     .row:nth-of-type(4){
376       background-color: #02CBFB;
377     }
378   </style>
379 </head>
380 <body>
381   <div class="container-fluid">
382     <div class="row">
383       <div class="col-xs-12"><h1 class="text-center">lorem</h1></div>
384     </div>
385     <div class="row">
386       <div class="col-sm-8">
387         <h1 class="text-center">ipsum</h1>
388       </div>
389       <div class="col-sm-4">
390         <h1 class="text-center">dolor</h1>
391       </div>
392     </div>
393   </div>
  
```

```

392         <div class="col-sm-4">
393             <h1 class="text-center">lorem</h1>
394         </div>
395         <div class="col-sm-4">
396             <h1 class="text-center">lorem</h1>
397         </div>
398         <div class="col-sm-4">
399             <h1 class="text-center">lorem</h1>
400         </div>
401     </div>
402     <div class="row">
403         <div class="col-xs-12">
404             <h1 class="text-center">Lorem</h1>
405         </div>
406     </div>
407 </div>
408 </body>
409 </html>

```

```

410
411 6. Offset(margin 개념)
412 -css/grid.css
413 [class*="col-"]{
414     background-color:#eee; /* silver */
415     border:1px solid black;
416     font-size:24px;
417 }
418
419 .row{
420     margin-top:10px;
421 }
422
423 <div class="row" style="float:right">
424     <div class="col-md-2">.col-md-2 Button</div>
425     <div class="col-md-2">.col-md-2 Button</div>
426     <div class="col-md-2">.col-md-2 Button</div>
427 </div>
428 -이 결과는 예상과 달리 오른쪽으로 정렬되지 않는다.
429 -그 이유는 각 div의 class="col-md-2"가 원래 설정되어 있는 Style과 style="float:right"과 충돌하기 때문이
430 다.
431 -화면상에서는 크기와 위치 모두 어그러져 있다.
432 -따라서 float 같은 Style을 이용하지 않고 bootstrap에서 제공하는 Grid sytem을 이용하면 된다.

```

```

433
434 1)방법1
435 <div class="row">
436     <div class="col-md-2">.col-md-2</div>
437     <div class="col-md-2">.col-md-2 Button</div>
438     <div class="col-md-1">.col-md-1</div>
439     <div class="col-md-2">.col-md-2 Button</div>
440     <div class="col-md-1">.col-md-1</div>
441     <div class="col-md-2">.col-md-2 Button</div>
442     <div class="col-md-1">.col-md-1</div>
443     <div class="col-md-1">.col-md-1</div>
444 </div>
445 이렇게 한 후, .col-md-1의 text를 아래와 같이 지운다.
446 <div class="row">
447     <div class="col-md-2"></div>
448     <div class="col-md-2">.col-md-2 Button</div>

```



```

449     <div class="col-md-1"></div>
450     <div class="col-md-2">.col-md-2 Button</div>
451     <div class="col-md-1"></div>
452     <div class="col-md-2">.col-md-2 Button</div>
453     <div class="col-md-1"></div>
454     <div class="col-md-1"></div>
455 </div>
456 -그렇지만 text를 지운 부분의 상단의 라인이 있기 때문에 어색하다.
457 -그 이유는 bootstrap에서 .col로 시작하는 class는 상단에 border를 주었기 때문이다.
458 -그래서 이 작업을 별도로 하면 좋지만 별로 좋은 code라고 보지 않는다.
459 -화면의 너비는 줄여보면 더 확실히 안좋다는 것을 알 수 있다.
460
461 2)방법2
462 -그래서 offset이 필요하다.
463 -offset은 이전에서부터 얼마나 떨어져 있으라는 상대영역의 뜻
464 -아래와 같이 하면 된다.
465 <div class="row">
466     <div class="col-md-2 col-md-offset-2">.col-md-2 .md-offset-2 Button</div>
467     <div class="col-md-2 col-md-offset-1">.col-md-2 .md-offset-1 Button</div>
468     <div class="col-md-2 col-md-offset-1">.col-md-2 .md-offset-1 Button</div>
469 </div>

```

## 7. Grid Nesting

```

473 -css/grid.css
474 [class*="col-"]{
475     background-color:#eee; /* silver */
476     border:1px solid black;
477     font-size:24px;
478 }
479
480 .row{
481     margin-top:10px;
482 }

```

1)요소들의 크기와 위치를 특정영역안으로도 정의할 수 있다.

2)아래와 같이 이렇게 하면

```

486 <div class="row">
487     <div class="col-sm-9"> <!--전체 화면의 3/4을 차지한다. -->
488         Level 1 : .col-sm-9
489         <div class="row">
490             <div class="col-xs-8 col-sm-6"> <!-- 전체 화면의 1/2이 아니라 Level1 영역의 1/2이다. -->
491                 Level 2: .col-xs-8 .col-sm-6 <!-- 즉 자기가 속한 영역에서의 분할영역이다. -->
492             </div> <!--화면의 사이즈가 더 줄어들면(xs), 8:4로 분할된다. -->
493             <div class="col-xs-4 col-sm-6">
494                 Level 2: .col-xs=4 .col-sm-6
495             </div>
496         </div>
497     </div>
498 </div>

```

3)또한 아래와 같이 해보면

```

501 <div class="row">
502     <div class="col-sm-9">
503         Level 1 : .col-sm-9
504         <div class="row">
505             <div class="col-xs-6 col-sm-3"> <!--Level 1의 영역중 xs는 1/2씩 나뉘이고 -->
506                 Level 2: .col-xs-6 .col-sm-3 <!-- sm에서는 1/4, 1/4 offset 하고 -->

```

```

507         </div>
508         <div class="col-xs-6 col-sm-6 col-sm-offset-3"> <!-- 나머지 2/4를 차지한다. -->
509             Level 2: .col-xs-4 .col-sm-6 col-sm-offset-3
510         </div>
511     </div>
512 </div>
513 </div>
514
515 4)이렇게 중첩해서 사용하면 전체 화면을 분할하는 것 뿐만 아니라 특정 영역을 offset과 같이 사용함으로써 특정 영역을
    분할하는 것도 가능하다.
516
517
518 8. Column ordering : push, pull
519 -css/grid.css
520 [class*="col-"]{
521     background-color:#eee; /* silver */
522     border:1px solid black;
523     font-size:24px;
524 }
525
526 .row{
527     margin-top:10px;
528 }
529
530 1)offset과 비슷하지만, push와 pull로 상대위치 정하기
531 2)기준은 왼쪽에서 오른쪽으로, 위에서 아래로 push한다.
532 3)오른쪽으로 2칸 push하기
533 <div class="row">
534     <div class="col-md-9 col-md-push-2" style="background-color:orange">col-md-9
        col-md-push-2</div>
535     <div class="col-md-3" style="background-color:red">col-md-3</div>
536 </div>
537 4)이번에는 왼쪽으로 4칸 pull 하기
538 <div class="row">
539     <div class="col-md-9 col-md-push-2" style="background-color:orange">col-md-9
        col-md-push-2</div>
540     <div class="col-md-3 col-md-pull-4" style="background-color:red">col-md-3
        col-md-pull-4</div>
541 </div>
542
543
544 9. 복습하기
545 1)기본
546
547 <style>
548     [class*="col-"]{
549         background-color:#eee; /* silver */
550         border:1px solid black;
551         font-size:24px;
552     }
553
554     .row{
555         margin-top:10px;
556     }
557 </style>
558
559 <body>
560     <div class="container-fluid">

```

```

561     <div class="row">
562         <div class="col-md-1">col-md-1</div>
563         <div class="col-md-1">col-md-1</div>
564         <div class="col-md-1">col-md-1</div>
565         <div class="col-md-1">col-md-1</div>
566         <div class="col-md-1">col-md-1</div>
567         <div class="col-md-1">col-md-1</div>
568         <div class="col-md-1">col-md-1</div>
569         <div class="col-md-1">col-md-1</div>
570         <div class="col-md-1">col-md-1</div>
571         <div class="col-md-1">col-md-1</div>
572         <div class="col-md-1">col-md-1</div>
573         <div class="col-md-1">col-md-1</div>
574     </div>
575     <div class="row">
576         <div class="col-md-8">col-md-8</div>
577         <div class="col-md-4">col-md-4</div>
578     </div>
579     <div class="row">
580         <div class="col-md-4">col-md-4</div>
581         <div class="col-md-4">col-md-4</div>
582         <div class="col-md-4">col-md-4</div>
583     </div>
584     <div class="row">
585         <div class="col-md-6">col-md-6</div>
586         <div class="col-md-6">col-md-6</div>
587     </div>
588 </div>
589 </body>

```

## 2) offset 사용하여 격자만들기

```

592 <div class="container-fluid">
593     <div class="row">
594         <div class="col-md-1">col-md-1</div>
595         <div class="col-md-1 col-md-offset-1">col-md-1</div>
596         <div class="col-md-1 col-md-offset-1">col-md-1</div>
597         <div class="col-md-1 col-md-offset-1">col-md-1</div>
598         <div class="col-md-1 col-md-offset-1">col-md-1</div>
599         <div class="col-md-1 col-md-offset-1">col-md-1</div>
600     </div>
601     <div class="row">
602         <div class="col-md-1 col-md-offset-1">col-md-1</div>
603         <div class="col-md-1 col-md-offset-1">col-md-1</div>
604         <div class="col-md-1 col-md-offset-1">col-md-1</div>
605         <div class="col-md-1 col-md-offset-1">col-md-1</div>
606         <div class="col-md-1 col-md-offset-1">col-md-1</div>
607         <div class="col-md-1 col-md-offset-1">col-md-1</div>
608     </div>
609 </div>

```

## 3) 단순 offset 연습

```

612 <div class="container-fluid">
613     <div class="row">
614         <div class="col-md-4">col-md-4</div>
615         <div class="col-md-4 col-md-offset-4">col-md-4 col-md-offset-4</div>
616     </div>
617     <div class="row">
618         <div class="col-md-3 col-md-offset-3">col-md-3 col-md-offset-3</div>

```

```

619     <div class="col-md-3 col-md-offset-3">col-md-3 col-md-offset-3</div>
620 </div>
621 <div class="row">
622     <div class="col-md-6 col-md-offset-3">col-md-6 col-md-offset-3</div>
623 </div>
624 </div>

```

## 627 10. Available classes(Bootstrap 숨기기)

### 628 1)특정 요소를 숨기거나 드러낼 때 사용

	매우 작은 기기 (모바일폰 < 768px)	작은 기기 (테블릿 >= 768px)	중간 기기 (데스크탑 >= 992px)	큰기기 (데스크탑 >= 1200px)
631				
632	.visible-xs-*    보임	숨겨짐	숨겨짐	숨겨짐
633	.visible-sm-*    숨겨짐	보임	숨겨짐	숨겨짐
634	.visible-md-*    숨겨짐	숨겨짐	보임	숨겨짐
635	.visible-lg-*    숨겨짐	숨겨짐	숨겨짐	보임
636	.hidden-xs        숨겨짐	보임	보임	보임
637	.hidden-sm        보임	숨겨짐	보임	보임
638	.hidden-md        보임	보임	숨겨짐	보임
639	.hidden-lg        보임	보임	보임	숨겨짐

### 641 2)아래의 예를 보자

```

642 <div class="row">
643     <div class="col-xs-1 hidden-xs">.col-xs-1 hidden-xs</div>
644     <div class="col-xs-2 hidden-sm">.col-xs-1 hidden-sm</div>
645     <div class="col-xs-3 hidden-md">.col-xs-1 hidden-md</div>
646     <div class="col-xs-6 hidden-lg">.col-xs-1 hidden-lg</div>
647 </div>

```

649 -화면이 md 에서는 md가 안보이고 점차 화면을 좁히면 md는 나타나고 sm은 없어진다.

650 -더 줄이면 xs가 사라지고 sm과 md가 나타난다.

651 -만일 화면을 더 크게 하면 lg는 사라지게 된다.

### 653 3)아래의 예를 보자.

```

654 <div class="row">
655     <div class="col-xs-2 visible-xs">.col-xs-2 visible-xs</div>
656     <div class="col-xs-4 visible-sm">.col-xs-4 visible-sm</div>
657     <div class="col-xs-6 visible-md">.col-xs-6 visible-md</div>
658 </div>

```

659 -특정 화면 크기에서는 나오지 않는 hidden과 달리 visible은 특정 화면 사이트에서만 나타나게 된다.

660 -그래서 해당 사이즈가 되면 한 개씩만 나오게 된다.