

Website Decoration with Bootstrap

Responsive Web Design

: Concept



- ▶ 웹 초창기의 웹 디자인 : 일반적으로 PC용 웹 브라우저로 접속
 - ▶ 당시 점유율이 높았던 Netscape, Internet Explorer에 잘 보이도록 하는 것이 목표
- ▶ 대안 브라우저 등장 이후의 웹 디자인 : 파이어폭스, 크롬, 오페라 등
 - ▶ 주요 브라우저에 똑같이 동작하도록 만드는 것이 목표
- ▶ 스마트폰 등장 이후의 웹 디자인
 - ▶ 다양한 스크린 사이즈를 가진 장치들의 등장



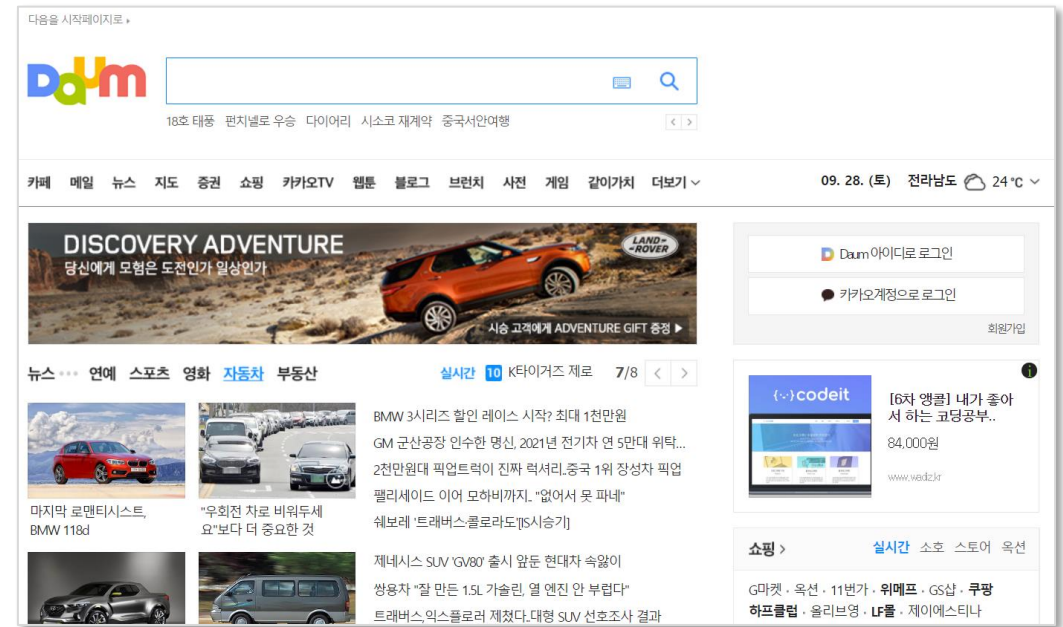
Responsive Web Design : Concept

- ▶ 다양한 화면 사이즈에 대응하기 위한 움직임
 - ▶ 많이 활용되는 화면 크기에 맞는 웹 페이지를 별도로 만든다
 - ▶ 사용자 접속 브라우저의 USER-AGENT 정보에 따른 분기
 - ▶ 문제점
 - ▶ 스마트폰 스크린 사이즈의 다양화
 - ▶ 스마트패드, 스마트 TV, 게임 콘솔 등 인터넷에 접속할 수 있는 기기 종류의 다양화
 - ▶ 다양해진 스크린 사이즈에 맞는 개별 웹 페이지를 모두 제작할 수는 없다!

모바일에서의 접속



PC에서의 접속

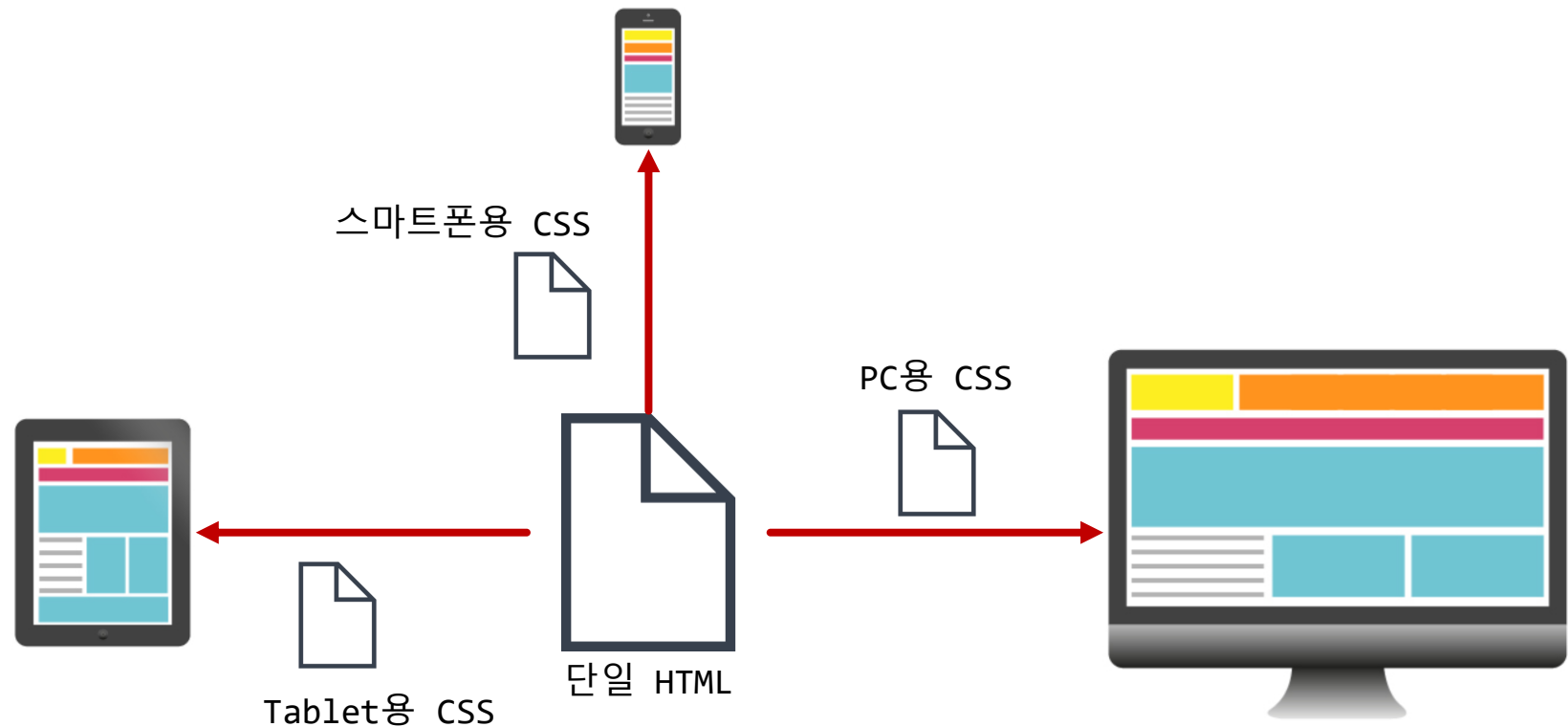


Responsive Web Design

: Concept



- ▶ 데이터를 표시하는 단 하나의 웹 페이지를 작성
- ▶ 사용자가 접속하는 기기(또는 환경)에 맞춰 화면 및 요소들의 크기를 조정, 모든 기기에 최적화된 화면 레이아웃을 보여주자! (Responsive)

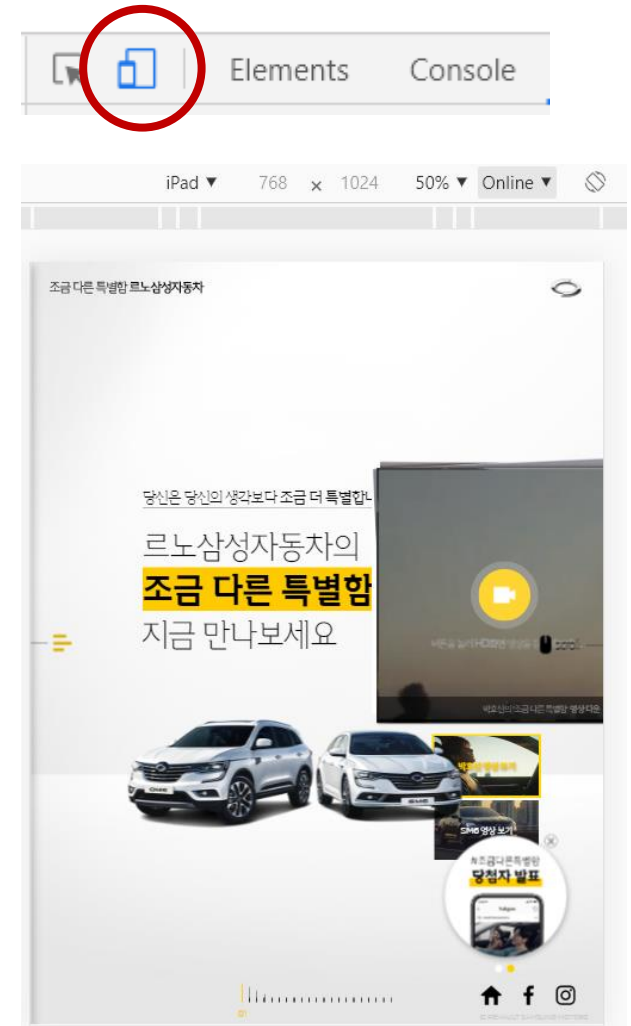


Responsive Web Design

: Concept



- ▶ Responsive Web Design 체험해 보기
 - ▶ Responsive Web Design Database : <http://rwdb.kr/>
 - ▶ 브라우저 개발자 도구의 디바이스 모드 활용하여 테스트
- ▶ Responsive Web Design으로 얻게 되는 장점
 - ▶ 기기의 화면 크기나 해상도에 구애 받지 않음
 - ▶ 모든 사용자가 자신의 환경에 최적화된 사용자 경험을 얻을 수 있다
 - ▶ 화면의 방향 변경 등에 적절하게 반응하는 웹 페이지를 구축할 수 있다
 - ▶ 모든 장치에 적합한 웹 사이트를 별도 구축하지 않아도 된다
 - ▶ 장치별 웹 페이지를 분리하지 않으므로 언제나 같은 주소를 사용할 수 있다
 - ▶ 유지 관리가 용이해진다



Responsive Web Design

: Concept



▶ Responsive Web Design의 단점

- ▶ 이전 브라우저 버전과 호환성의 문제가 있을 수 있다
 - ▶ 주로 활용하는 기술인 Media Query는 Internet Explorer 9 이상부터 지원
 - ▶ 하위 버전의 브라우저를 지원하기 위해서는 브라우저를 체크해 하위 버전용 페이지로 이동하거나 하위 버전 지원을 위한 별도 라이브러리를 추가해야 함
- ▶ 사이트 디자인이 단순
 - ▶ 기기의 해상도에 적절히 레이아웃이 바뀌어야 하므로 복잡한 레이아웃, 그래픽 효과가 많은 페이지에는 적합하지 않을 수 있다

▶ Adaptive Web Design(AWD)

- ▶ 서버 혹은 브라우저에서 사용자의 브라우저를 체크, 그에 적합한 Resource들을 별도로 불러오거나, 적절한 레이아웃을 개별 구성
- ▶ 반응형 웹이 레이아웃을 재배치하는 개념이라면 적응형 웹은 기기에 따라 별도의 레이아웃과 리소스를 사용하므로 보다 세밀한 부분까지 조정 가능

Media Query

: 미디어 쿼리와 뷰포트



- ▶ CSS3의 모듈 중 하나, 사이트에 접속하는 기기별로 적용할 CSS를 다르게 지정해 주는 기술
- ▶ 뷰포트(Viewport)
 - ▶ 사용자의 장치에서 실제 콘텐츠가 표시되는 부분을 가리키는 용어
 - ▶ PC의 1px와 스마트폰의 1px은 다르다.
 - ▶ 같은 너비 기준을 PC와 스마트폰에 동일하게 적용해서는 안된다
 - ▶ 해결책은 스마트 폰 화면을 웹 문서 너비에 맞추는 것이 아니라 웹 문서의 콘텐츠를 스마트 폰의 화면 너비에 맞춰야 한다
- ▶ 뷰포트의 설정(예)

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

Media Query

: viewport 메타 태그



▶ viewport 메타 태그의 속성

속성명	내용
width	뷰포트의 너비. 기본값은 device-width
height	뷰포트의 높이. 기본값은 device-height
initial-scale	초기 배율. 기본값은 1
user-scalable	사용자가 페이지를 확대, 축소할 수 있는지 여부 지정. 기본값은 yes
minimum-scale	확대/축소할 수 있는 최솟값(너비 기준). 기본값은 0.25
maximum-scale	확대/축소할 수 있는 최댓값(너비 기준). 기본값은 5.0
target-densityDpi	고해상도 단말기의 화면 해상도에 맞추어 수 있도록 페이지 확대. 기본값은 device-dpi

Media Query



▶ 미디어 쿼리 구문

- ▶ 미디어 쿼리를 사용하기 위해서는 `style` 태그 안에 다음과 같은 구문을 선언

```
@media [only | not] media-type and (media-feature)
```

- ▶ 혹은 외부 CSS를 `link`로 연결할 때, 장치별로 다른 리소스를 불러올 수 있다

```
<link rel="stylesheet"  
      media="media-type and [only|not] (media-feature)">
```

▶ 주요 미디어 유형(Media Type)

Media Type	설명
all	모든 미디어. 기본값
screen	스마트폰, PC 등의 화면
print	출력 장치(프린터)
tv	음성과 영상이 출력되는 TV
projection	프로젝터

Media Query

: 조건



▶ 미디어 쿼리에서 사용하는 주요 조건들

- ▶ orientation을 제외하면 min-, max- 접두사로 최솟값, 최댓값을 표현할 수 있음

조건		설명
디스플레이	width	웹 문서의 너비
	height	웹 문서의 높이
	device-width	기기의 너비
	device-height	기기의 높이
	orientation	화면의 방향. landscape(가로) or portrait(세로)
	aspect-ratio	화면 비율
	device-aspect-ratio	기기의 화면 비율
	resolution	기기 해상도
색상 관련	color	기기의 색상 비트 수
	color-index	기기의 색상 수

Media Query

: 중단점



- ▶ 미디어 쿼리 작성시 화면 크기에 따라 적용할 기준 지점을 명시해야 하며 이 각 지점을 중단점(Break Point)이라 한다

- ▶ 예) 화면 너비가 1280px 이하일 때 적용할 스타일의 정의

```
@media screen and (max-width: 1280px) {  
    /* 화면 너비가 1280px보다 작은 경우 적용할 CSS */  
}
```

- ▶ 예) 화면 너비가 640px 이상, 1280px 이하일 때 적용할 스타일의 정의

```
@media screen and (min-width: 640px) and (max-width: 1280px) {  
    /* 화면 너비가 640px ~ 1280px 사이일 경우 적용할 CSS */  
}
```

개발자 팁:

- 미디어 쿼리를 쉽게 작성하기 위한 다양한 유틸리티 혹은 웹사이트가 존재
- <https://simplecss.eu/>

Media Query

: 실습



- ▶ Resonsive.zip 프로젝트를 다운 받아 스크린 사이즈별 CSS를 적용해 봅니다
 - ▶ 단계 1) 모든 형식의 디스플레이에 적용되는 CSS를 적용합니다.
 - ▶ 내용은 supports > 01-basic-css.txt에 있습니다.
 - ▶ 단계 2) 태블릿용 (768px 이상에 적용) CSS를 적용합니다.
 - ▶ 내용은 supports > 02-tables-css.txt에 있습니다.
 - ▶ 미디어 쿼리는 다음의 내용을 따릅니다.

```
@media screen and (min-width: 768px) {  
    /* 화면 너비가 768px보다 큰 경우 적용할 CSS */  
}
```

- ▶ 단계 3) 대형 화면 (1024px 이상에 적용) CSS를 적용합니다.
 - ▶ 내용은 supports > 03-computer-css.txt에 있습니다.
 - ▶ 미디어 쿼리는 다음의 내용을 따릅니다.

```
@media screen and (min-width: 1024`px) {  
    /* 화면 너비가 1024px보다 클 경우 적용할 CSS */  
}
```

Media Query

: 실습 2



- ▶ 미디어 쿼리별로 선택적으로 CSS를 로드하는 실습입니다.
 - ▶ 단계 1) 기존 적용되고 있는 CSS는 모두 주석 처리합니다.
 - ▶ 단계 2) 미디어 쿼리별로 CSS 파일을 만듭니다.
 - ▶ 01-default-css.txt -> css/default.css
 - ▶ 02-tablet-css.txt -> css/tablet.css
 - ▶ 03-computer-css.txt -> css/computer.css
 - ▶ 단계 3) link 태그의 media 속성에 미디어 쿼리를 명시하여 스크린 사이즈별 CSS를 선택적으로 로딩합니다.

```
<link rel="stylesheet" href="css/default.css" /><!-- 기본 CSS -->
<link rel="stylesheet" href="css/tablet.css"
      media="screen and (min-width:768px)" /><!-- 태블릿 CSS -->
<link rel="stylesheet" href="css/computer.css"
      media="screen and (min-width:1024px)" /><!-- 데스크탑 CSS -->
```

Bootstrap

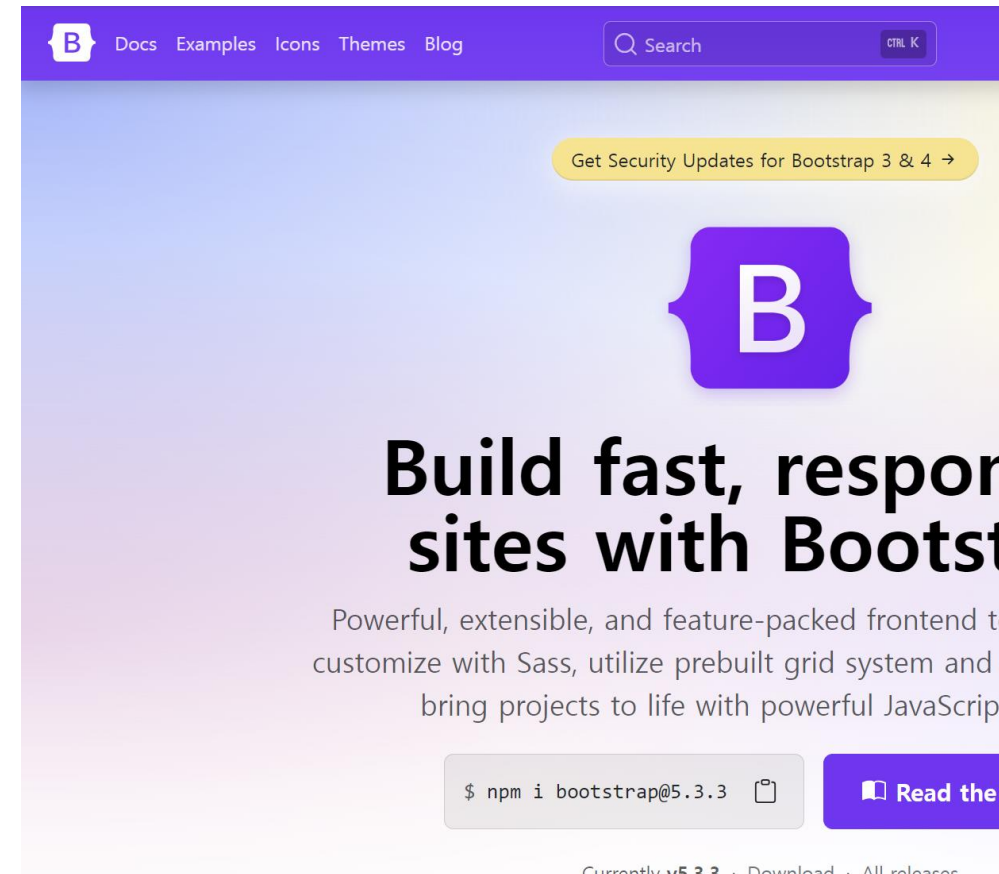


▶ CSS Framework

- ▶ 반응형 웹이 대중화되면서 반응형 웹을 만드는 기본 기술인 가변 그리드와 미디어 쿼리를 쉽게 사용하기 위한 Framework들이 다수 등장
- ▶ Foundation, Skeleton, Bootstrap 등

▶ Bootstrap 사용의 장점

- ▶ 반응형 그리드 시스템을 손쉽게 사용할 수 있도록 설계
- ▶ 다양한 컴포넌트, JavaScript 라이브러리를 포함
- ▶ 여러 가지 플러그인을 활용해 손쉽게 기능을 확장할 수 있음
- ▶ Twitter(X)의 주도로 작성되어 신뢰도와 품질이 높고 레퍼런스(적용 사이트)가 다양하다



Bootstrap

: 부트스트랩 소스의 연결



▶ 방법 1. 다운로드해서 링크

- ▶ 부트스트랩 공식 사이트(<https://getbootstrap.com/>)에서 직접 파일을 다운로드하여 웹 콘텐츠영역의 적절한 위치에 복사하여 사용
 - ▶ 일반적 사용의 경우, min 버전으로 충분

▶ 방법 2. CDN으로 링크 (추천)

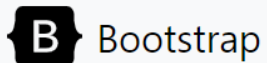
- ▶ Contents Delivery Network 서버의 파일을 링크하여 사용
- ▶ 부트스트랩 다운로드 페이지의 BootstrapCDN 항목을 참조하여 페이지에 링크

Bootstrap

: 첫 번째 부트스트랩 페이지 구성



- ▶ 부트스트랩 홈페이지 > Footer > Starter template
 - ▶ 많은 수의 템플릿들이 마련되어 있음
- ▶ 부트스트랩 홈페이지 > Docs > Introduction 항목을 확인
 - ▶ [실습] Quick Start 항목을 참고, `template.html`을 만들어 봅니다.
 - ▶ 특히 Bootstrap의 CSS, JS가 제대로 연결되었는지 확인합니다.



Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.3.3.

Links

Home

Docs

Examples

Icons

Guides

Getting started

Starter template

Webpack

Parcel

Projects

Bootstrap 5

Bootstrap 4

Icons

RFS

Community

Issues

Discussions

Corporate sponsors

Open Collective

Bootstrap

: 첫 번째 부트스트랩 페이지 구성



▶ 일반적인 템플릿 구성

```
<!doctype html>
<html lang="ko">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bootstrap demo</title>
  <link href="./bootstrap/css/bootstrap.min.css" rel="stylesheet" />
</head>

<body>
  <h1>Hello, world!</h1>
  <script src="./bootstrap/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

 bootstrap.bundle.js는 popper.js와 통합된 버전

Bootstrap

: 첫 번째 부트스트랩 페이지 구성



▶ popper.js를 포함하지 않은 템플릿 구성

```
<!doctype html>
<html lang="ko">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bootstrap demo</title>
  <link href="./bootstrap/css/bootstrap.min.css" rel="stylesheet" />
</head>

<body>
  <h1>Hello, world!</h1>
  <script src="./bootstrap/js/bootstrap.min.js"></script>
</body>

</html>
```

bootstrap.js는 popper.js가 통합되지 않은 버전

- popper.js의 기능이 필요하지 않은 경우 전체 번들 크기를 줄이기 위해
- popper.js의 특정 버전을 필요로 할 때는 popper.js를 포함하지 않음

Bootstrap Layout

: container



- ▶ 부트스트랩은 반응형 웹 구현을 위해 12컬럼 그리드 시스템을 이용
 - ▶ 부트스트랩에는 css가 미리 정의되어 있어, 클래스 스타일을 잘 이해하면 다양한 장치에 적합한 반응형 웹 레이아웃을 손쉽게 만들 수 있음
 - ▶ bootstrap.css의 container 클래스 선언부를 확인해 봅시다.
 - ▶ Q> 부트스트랩이 활용하는 중단점은 각각 몇 px입니까?

```
.container {  
  width: 100%;  
  padding-right: 15px;  
  padding-left: 15px;  
  margin-right: auto;  
  margin-left: auto;  
}
```

container 클래스의 기본 정의



```
@media (min-width: 576px) {  
  .container { max-width: 540px; }  
}
```

```
@media (min-width: 768px) {  
  .container { max-width: 720px; }  
}
```

```
@media (min-width: 992px) {  
  .container { max-width: 960px; }  
}
```

```
@media (min-width: 1200px) {  
  .container { max-width: 1140px; }  
}
```

장치의 스크린 사이즈별 미디어 쿼리 정의

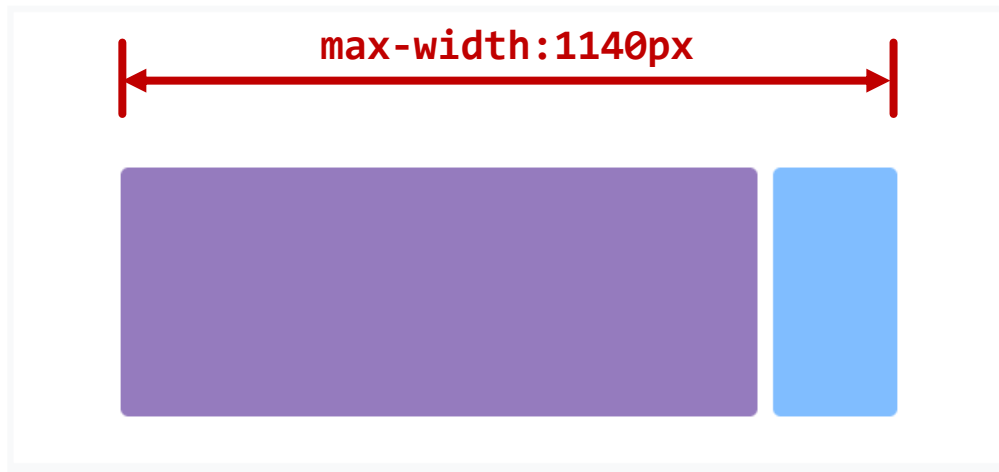
Bootstrap Layout

: container vs container-fluid

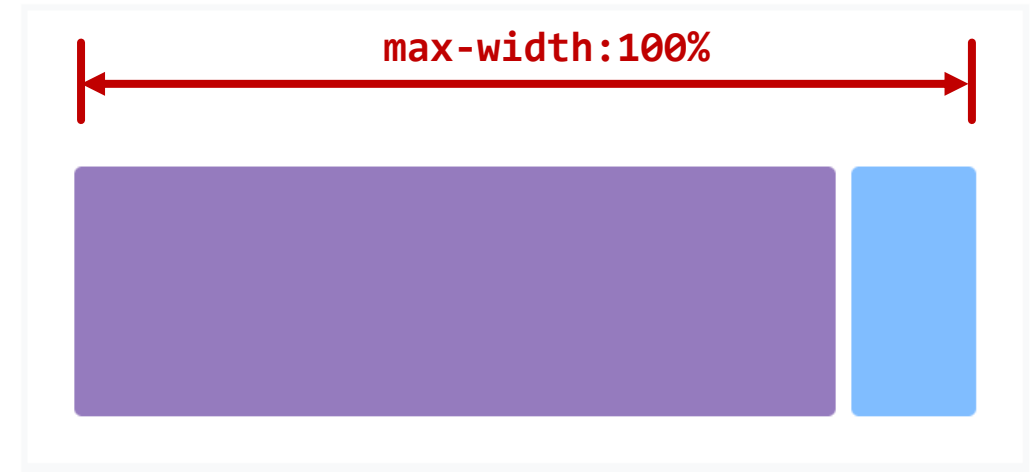


- ▶ container 클래스
 - ▶ 화면의 폭에 따라 변화하며 최대 1140px 사이즈
 - ▶ 좌우에 15px씩의 padding
- ▶ container-fluid
 - ▶ 화면의 가로폭 전체를 사용
 - ▶ 좌우에 15px씩의 padding

두 가지 형태의 container를 중첩해서
사용해서는 안된다



container



container-fluid

Bootstrap Layout

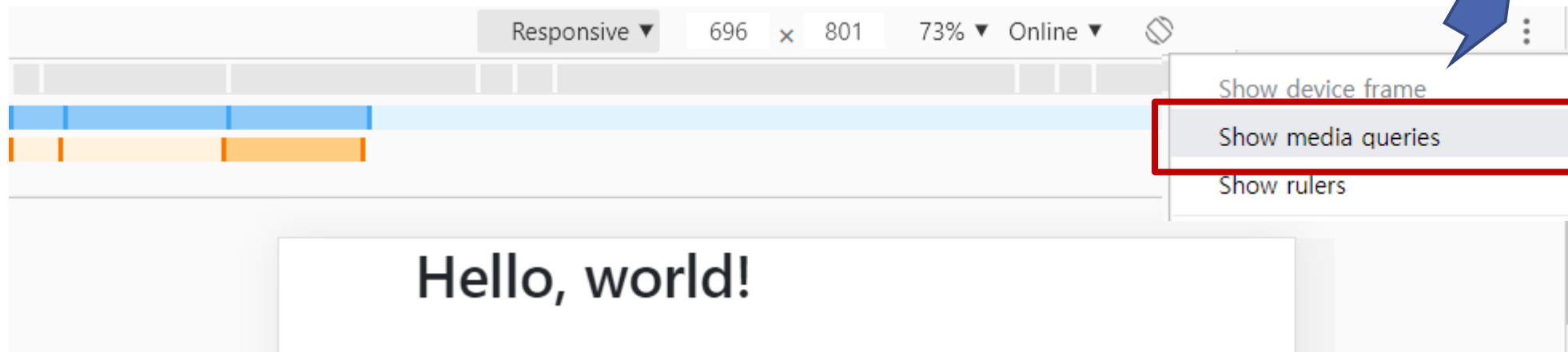
: container 적용 및 미디어 쿼리 중단점 확인



- ▶ 앞서 작성한 Starter Page의 내용을 container로 묶어 봅니다.
페이지 렌더링의 변화와 중단점을 확인해 봅니다.

```
...  
  <div class="container">  
    <h1>Hello, world!</h1>  
  </div>  
...
```

- ▶ 개발자 도구 > Device Toolbar > Show Media Queries 옵션을 이용,
웹 페이지 내에 적용되어 있는 미디어 쿼리의 중단점을 확인할 수 있음



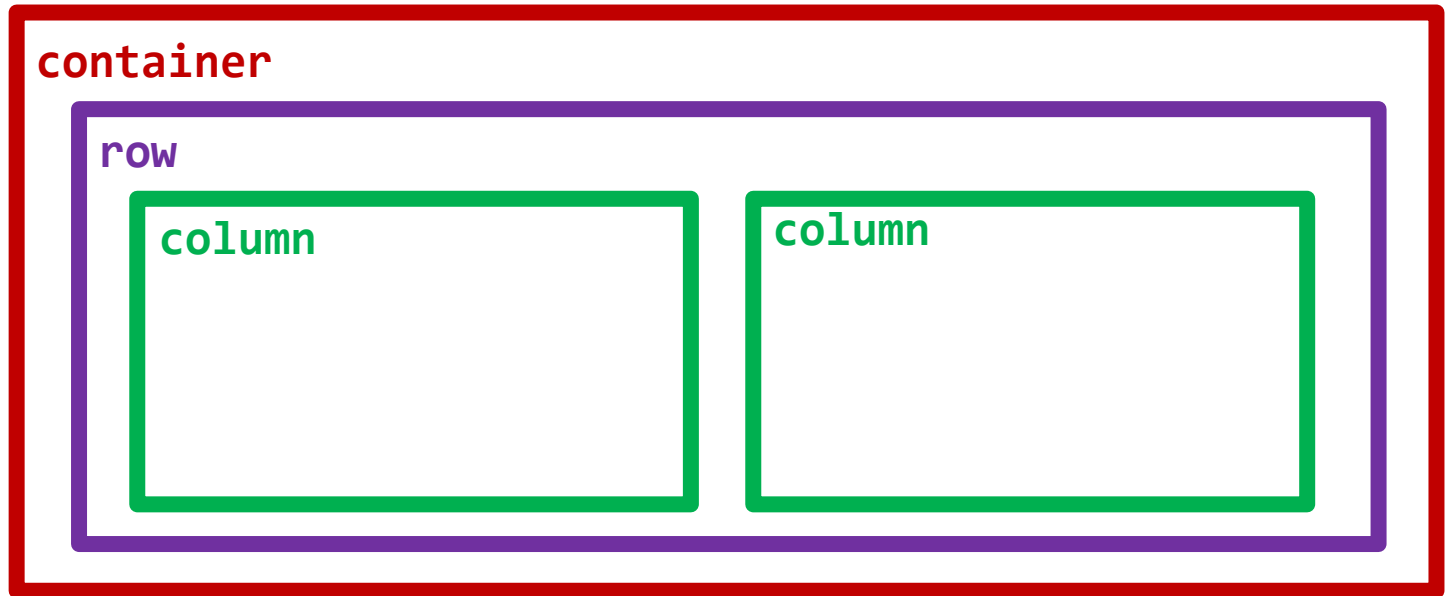
Bootstrap Layout

: container, row, col



- ▶ container : 콘텐츠를 감싸는(Wrapping) 요소. Grid 시스템의 필수 요소
- ▶ row : container 내에서 컬럼들을 감싸는 요소
- ▶ column : 실제 콘텐츠가 배치되는 영역
 - ▶ * 그리드 레이아웃 구성시 반드시 .row를 먼저 배치하고, 그 내부에 column을 배치하도록 한다

```
...  
<div class="container">  
  <div class="row">  
    <div class="col"></div>  
    <div class="col"></div>  
  </div>  
  ...  
</div>  
...
```



Bootstrap Layout

: column



▶ 부트스트랩은 콘텐츠 표시를 위해 12 Column Grid 시스템을 사용

▶ 컬럼 지정을 위한 기본 형식

.col-화면종류-컬럼개수

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Bootstrap Layout

: column 균등 분할



- ▶ 화면종류, 컬럼개수를 지정하지 않으면 자동으로 레이아웃을 맞춤

- ▶ 예1) 한 row에 두 개의 column을 표시

```
<div class="row">  
  <div class="col">1 of 2</div>  
  <div class="col">2 of 2</div>  
</div>
```

1 of 2

2 of 2

- ▶ 예2) 한 row에 세 개의 column을 표시

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

1 of 3

2 of 3

3 of 3

col 1

col 2

col 3

col 4

col 5

col 6

col 7

col 8

col 9

col 10

col 11

col 12

Bootstrap Layout

: column 화면-컬럼 지정 분할



- ▶ 화면종류, 컬럼개수를 지정하여 특정 화면 크기의 장치에 사용할 컬럼의 비율을 지정할 수 있음

- ▶ 예1) md size의 스크린에 한 row에 4컬럼, 8컬럼씩을 배분
- ▶ md보다 큰 화면은 이 그리드 구성대로 표시

```
<div class="row">  
  <div class="col-md-4">1 of 2</div>  
  <div class="col-md-8">2 of 2</div>  
</div>
```

1 of 2

2 of 2

- ▶ 예2) 한 row에 세 개의 column을 표시 가운데 컬럼만 10컬럼 점유

```
<div class="row">  
  <div class="col">1 of 3</div>  
  <div class="col-10">2 of 3</div>  
  <div class="col">3 of 3</div>  
</div>
```

1 of 3

2 of 3

3 of 3

col 1

col 2

col 3

col 4

col 5

col 6

col 7

col 8

col 9

col 10

col 11

col 12

Bootstrap Layout

: 특정 화면 크기에서 화면 요소 표시/감춤



▶ 특정 화면 크기의 장비에서 화면 출력 여부를 제어할 수 있다

▶ 예) CSS의 display 속성을 none, inline, block 등으로 변경

속성	설명
.d-none	화면에서 공간 점유를 해제(감춤)
.d-inline	요소를 인라인 레벨로 표시
.d-block	요소를 블록 레벨로 표시
.d-sm-none	작은 화면일 때 요소를 감춤
.d-sm-block	작은 화면일 때 요소를 블록 레벨로 표시
.d-md-none	중간 화면일 때 요소를 감춤
.d-md-block	중간 화면일 때 요소를 블록 레벨로 표시

▶ 더 많은 옵션은 부트스트랩 홈페이지 > Documentation > Utilities > Display 섹션에서 확인

Components

: Button



- ▶ 컴포넌트 : 웹 제작에 자주 사용하는 요소들의 CSS를 미리 정의해 둔 것
 - ▶ 부트스트랩 홈페이지 > Documentation > Components 항목 참조
 - ▶ 일부 컴포넌트들은 단순히 CSS 뿐만 아니라 jQuery 및 bootstrap.js와 함께 추가 기능을 수행하기도 한다
- ▶ Button 컴포넌트
 - ▶ 입력 폼, 대화상자 등에서 행동(Action) 혹은 기능을 수행하는 컴포넌트
 - ▶ 기본적으로는 <button> 태그를 위해 만들어진 컴포넌트이지만, a, input 등의 요소에도 적용할 수 있음
 - ▶ 기본 클래스: .btn
 - ▶ 보조 클래스: 버튼의 용도, 사이즈 등 버튼을 다양하게 꾸미기 위한 보조 클래스가 있음

Components

: Button



▶ 용도에 따른 버튼 스타일의 지정

▶ 적용 클래스 규칙

```
class="btn btn-용도"
```

▶ Examples

```
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-light">Light</button>  
<button type="button" class="btn btn-dark">Dark</button>  
<button type="button" class="btn btn-link">Link</button>
```

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

Link

Components

: Button



▶ 용도에 따른 버튼 스타일의 지정(Outline Button)

▶ 적용 클래스 규칙

```
class="btn btn-outline-용도"
```

▶ Examples

```
<button type="button" class="btn btn-outline-primary">Primary</button>  
<button type="button" class="btn btn-outline-secondary">Secondary</button>  
<button type="button" class="btn btn-outline-success">Success</button>  
<button type="button" class="btn btn-outline-danger">Danger</button>  
<button type="button" class="btn btn-outline-warning">Warning</button>  
<button type="button" class="btn btn-outline-info">Info</button>  
<button type="button" class="btn btn-outline-light">Light</button>  
<button type="button" class="btn btn-outline-dark">Dark</button>
```

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

Components

: Button



▶ 버튼의 크기 지정

- ▶ 별도의 CSS를 사용하여 픽셀 단위로 사이즈를 지정할 수도 있지만, 부트스트랩이 지정한 크기를 지정할 수도 있음

```
class="btn btn-lg"  
class="btn btn-sm"
```

▶ Examples

```
<!-- 큰 버튼 .btn-lg -->  
<button type="button" class="btn btn-primary btn-lg">Primary Big</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<!-- 작은 버튼 .btn-sm -->  
<button type="button" class="btn btn-primary btn-sm">Primary Small</button>
```

Primary Big

Primary

Primary Small

Components

: Alert



▶ Alert 컴포넌트

- ▶ 사용자에게 피드백을 전달하기 위한 메시지 표시 상자
- ▶ 기본 클래스: `.alert`
- ▶ 보조 클래스: 피드백 메시지의 용도를 지정하기 위한 다양한 보조 클래스들
- ▶ Alert 내에 링크를 포함하려면 `a` 태그에 `.alert-link` 클래스를 부여하면 좋음
- ▶ Example

```
<div class="alert alert-primary" role="alert">  
  중요한 메시지 알림 예제 <a href="#" class="alert-link">외부 링크</a>.  
</div>
```

중요한 메시지 알림 예제 외부 링크.

Components

: Alert



- ▶ Alert 버튼에 닫기 버튼 및 기능 추가

```
<div class="alert alert-info alert-dismissible fade show" role="alert">  
  중요한 메시지가 있습니다.  
  자세한 내용을 보시려면 <a href="#" class="alert-link">여기</a>를 클릭하세요.  
  <button type="button" class="btn-close"  
    data-bs-dismiss="alert"  
    aria-label="Close"></button>  
</div>
```

시려면 여기 를 클릭하세요.



Components

: NavBar



- ▶ NavBar 컴포넌트 : 사이트 상단에서 메인 메뉴로 활용
 - ▶ 화면이 넓을 때는 모든 내용이 표시되지만, 표시할 영역이 마땅치 않을 때는 숨기고 사용자가 터치할 때(사용자가 필요로 할 때) 화면에 표시해 주는 편리한 메뉴 기능을 손쉽게 구현할 수 있다
 - ▶ 기본 클래스 : .navbar (nav) 태그로 작성한다
 - ▶ 어두운 배경에 어울리는 내비게이션 바는 .navbar-dark로, 그렇지 않은 경우는 생략한다
- ▶ Examples

```
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">  
  <!-- navbar-expand-md : 중간 화면 이상에서는 내비게이션을 모두 펼침 -->  
  <!-- navbar-dark : 어두운 배경에 맞는 내비게이션 스타일 -->  
  <!-- fixed-top : 내비게이션을 항상 위쪽에 고정한다 -->  
  <!-- bg-dark : 내비게이션 배경을 어둡게 지정-->  
</nav>
```

Components

: NavBar



▶ Brand Sub Component

- ▶ 회사, 서비스 명, 제품 명 등을 표기하는 보조 컴포넌트
- ▶ 클래스 : .navbar-brand
- ▶ NavBar 컴포넌트 내에서 a 링크 태그를 이용할 경우 .nav-link 클래스를 이용한다

▶ Examples

```
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">  
  <div class="navbar-brand">My Home</div>  
</nav>
```

My Home

Components

: NavBar



▶ NavBar Toggler Sub Component

- ▶ 화면 사이즈가 줄어들 때(내용을 표현할 여유 공간이 넉넉하지 않을 때) 메뉴를 축약해 표시하는 클래스 (일명 햄버거 버튼)
- ▶ 클래스 : .navbar-toggler

▶ Examples

My Home



```
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
  <div class="navbar-brand">My Home</div>
  <button class="navbar-toggler" type="button"
    data-bs-toggle="collapse"
    data-bs-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
</nav>
```

Components

: NavBar



▶ NavBar Collapse Sub Component

- ▶ 네비게이션 바 중, 나타났다 사라졌다 할 부분에 `.collapse`와 `.navbar-collapse` 클래스 적용
- ▶ 실제 메뉴 아이템에는 `nav-item` 적용
- ▶ `.mr-auto` : 오른쪽 마진 자동 적용

▶ Examples



```
...  
<div class="collapse navbar-collapse" id="navbarSupportedContent">  
  <ul class="nav navbar-nav mr-auto">  
    <li class="nav-item active"><a class="nav-link">Home</a></li>  
    <li class="nav-item"><a class="nav-link">Features</a></li>  
    <li class="nav-item"><a class="nav-link">About</a></li>  
  </ul>  
</div>  
...
```



웹 디자인 팁 & 도우미 사이트들

디자인 무지렁이를 위한

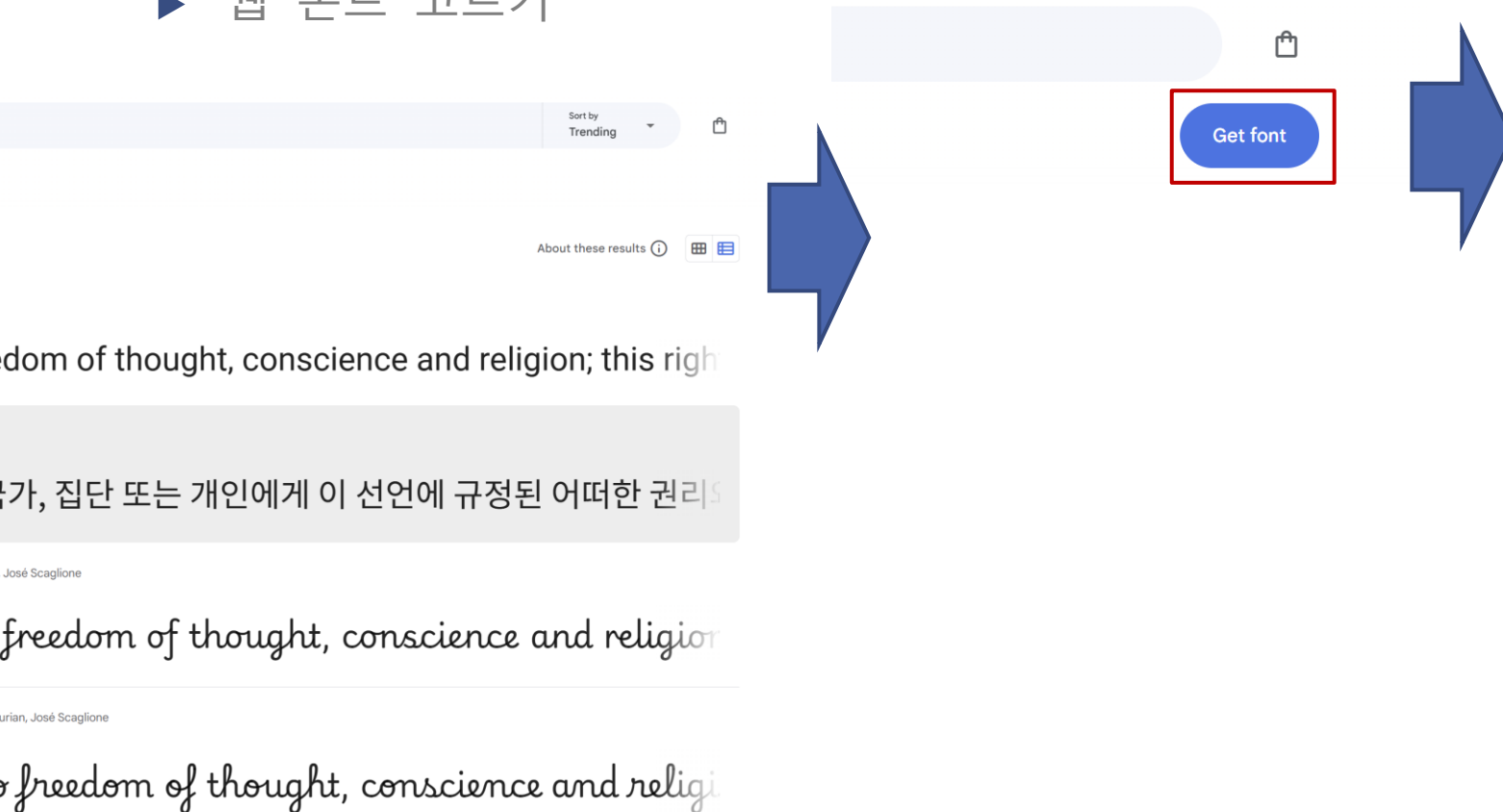
- ▶ 웹 폰트(Web Font) : 방문자 컴퓨터에 웹 페이지에서 사용한 폰트가 없을 경우 웹 페이지와 함께 저장된 폰트 정보를 다운로드하여 제작자가 의도한 디자인을 보여주는 웹 전용 폰트
- ▶ 웹 폰트는 TTF/OTF, WOFF, WOFF2, EOT 등 다양한 형식이 있음
 - ▶ 웹 폰트 사용시 글꼴의 라이선스를 반드시 확인하도록 한다
- ▶ 웹 폰트 지정 CSS의 설정의 예

```
@font-face {  
  font-family: Blackout, Arial Black;  
  src: local('Blackout'),  
       url('Blackout.eot'),  
       url('Blackout.woff') format('woff'),  
       url('Blackout.ttf') format('truetype')  
}
```

Google Web Font



- ▶ <https://fonts.google.com/>
- ▶ 저작권 없는 무료 웹 폰트를 제공하는 서비스
- ▶ 웹 폰트 고르기



Web

Android

iOS

Flutter

☒ <link> ☐ @import

Embed code in the <head> of your html

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100..900&display=swap" rel="stylesheet">
```

Copy code

Noto Sans KR: CSS class for a variable style

```
// <uniquifier>: Use a unique and descriptive class name
// <weight>: Use a value from 100 to 900

.noto-sans-kr-<uniquifier> {
  font-family: "Noto Sans KR", sans-serif;
  font-optical-sizing: auto;
  font-weight: <weight>;
  font-style: normal;
}
```

Copy code

[Legacy browser support for variable fonts](#)

[Read our FAQ](#)

[API Docs](#)

Google Web Font



- ▶ 생성된 코드를 자신의 웹 페이지에 포함

```
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link  
    href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100..900&  
display=swap" rel="stylesheet">
```


Lorem Ipsum



- ▶ 출판이나 그래픽 디자인 분야에서 폰트, 타이포그래피, 레이아웃 등 시각적 연출을 보여줄 때 사용하는 표준 채우기 텍스트

- ▶ Dummy Contents

- ▶ <https://www.lipsum.com/>

- ▶ Lorem Ipsum을 쉽게 만들어주는 Generator 서비스

5

☒ paragraphs

☐ words

☐ bytes

☐ lists

☒ Start with 'Lorem ipsum dolor sit amet...'

Generate Lorem Ipsum



Lorem Ipsum

"Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit..."
"There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain..."

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam commodo sem lorem, ac aliquet sem viverra vitae. Sed tincidunt mattis nibh, in ultricies dolor sollicitudin a. Nunc vulputate mauris viverra odio pharetra dignissim. Vivamus leo risus, euismod at justo ac, feugiat accumsan enim. Sed eleifend placerat lacus, sed tincidunt ante mollis sed. Pellentesque venenatis facilisis purus, sit amet finibus ipsum maximus vitae. Integer sodales massa et lobortis semper. Etiam varius purus at leo ornare, eget bibendum urna pretium. In eget tempus nunc. Nunc ac sagittis tellus. Mauris imperdiet, nisl id gravida accumsan, arcu turpis ullamcorper tortor, et feugiat leo urna in turpis. Donec viverra fermentum sapien, et commodo metus pretium vitae.

- ▶ 참고: 한국어 Lorem Ipsum

- ▶ <http://guny.kr/stuff/klorem/>

Place Holder Image



- ▶ 웹 디자인 작업시 레이아웃 등 시각적 요소 확인을 위해 이미지 위치에 실제 콘텐츠 대신 삽입(Place Holder)하는 이미지

- ▶ <http://placeimg.com/>



<http://placeimg.com/640/480/animals/sepia>



```

```

Lorem Picsum



- ▶ Lorem Ipsum의 이미지 버전
- ▶ 웹 디자인 작업시 레이아웃 등 시각적 요소 확인을 위해 이미지 위치에 실제 콘텐츠 대신 삽입(Place Holder)하는 이미지
- ▶ <https://picsum.photos/>



The Lorem Ipsum for photos.

```

```

