

# Django - CSS(템플릿 꾸미기)

## CSS란?

CSS(Cascading Style Sheets)는 HTML와 같이 마크업언어(Markup Language)로 작성된 웹사이트의 외관을 꾸미기 위해 사용되는 언어이다.

## 부트스트랩

부트스트랩(Bootstrap)은 유명한 HTML과 CSS프레임워크이다.

<https://getbootstrap.com/>

## 부트스트랩 설치

부트스트랩을 설치하려면, `.html` 파일 내 `<head>` 에 이 링크를 넣어야 한다.

```
blog/templates/blog/post_list.html
```

```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
```

이 링크는 프로젝트에 새 파일을 추가하는 게 아니고, 인터넷에 있는 파일을 연결하는 것이다.

# Django Girls Blog

published: 2020년 5월 1일 8:33 오후

## notice

This is a test notice.

Hello World

Hello Django~

published: 2020년 5월 1일 8:33 오후

## 정적파일

정적 파일은 CSS와 이미지 파일에 해당한다. 이 콘텐츠는 요청 내용에 따라 바뀌는 것이 아니기 때문에 모든 사용자들이 동일한 내용을 볼 수 있다.

## 정적파일을 넣는 위치

서버에서 collectstatic를 실행할 때 처럼, 장고는 “admin”앱에서 정적 파일을 어디서 찾아야하는지 이미 알고 있다. 이제 `"blog"` 앱에 정적파일을 추가하면 된다.

`"blog"` 앱 안에 `static` 라는 새 폴더를 만든다. :

```
djangogirls
├── blog
│   ├── migrations
│   ├── static
│   └── templates
└── mysite
```

장고는 `app` 폴더 안에 있는 `static` 폴더를 자동으로 찾을 수 있다. 이 콘텐츠를 정적 파일로 사용하게 되는 것이다.

## CSS 파일 생성

CSS파일을 만들어 나만의 스타일을 가진 웹페이지를 만들 것이다. `static` 디렉토리 안에 `css` 라고 새로운 디렉토리를 생성한다. 그리고 `css` 디렉토리 안에 `blog.css` 라는 파일을 생성한다.

```
djangogirls
├── blog
│   └── static
│       └── css
│           └── blog.css
```

컴퓨터는 특별한 코드를 사용해 색상을 나타낸다. #으로 시작해 알파벳(A-F)와 숫자(0-9) 중 6개를 조합해 hexa 코드(hexacode)로 나타낸다. 이 곳에서 원하는 색상 코드를 찾거나, <https://www.colorpicker.com/> 또는 빨간색(red) 또는 녹색(green) 등 미리 명시된 색상을 사용할 수 있다.

`blog/static/css/blog.css` 파일에 아래 코드를 추가. :

```
h1 a {
    color: #FCA205;
}
```

`h1 a` 는 CSS 셀렉터(Selector)라고 한다. `h1` 요소 안에 어떠한 `a` 요소를 넣어 스타일을 적용 할 수 있다는 것이다. 예를 들어 `<h1><a href="">link</a></h1>` 라면, 오렌지색 `#FCA205` 으로 바꿀 수 있다.

CSS 파일에서는 HTML 파일에 있는 각 요소들에 스타일을 정의할 수 있다. 요소를 식별하는 첫 번째 방법은 요소 이름을 사용하는 것이다. `a`, `h1`, `body` 와 같은 것은 모두 요소 이름이다, 또 `class` 속성이나 `id` 속성에 의해 요소를 식별합니다. 클래스와 ID는 요소에 직접 부여한 이름이다. 클래스는 요소 그룹을 정의하고 ID는 특정 요소를 가리킨다. 아래 코드와 같이 태그 이름은 `a`, 클래스는 `external_link` 또는 ID는 `link_to_wiki_page` 로 사용하여 태그를 통해 요소를 식별할 수 있다.

```
<a href="https://en.wikipedia.org/wiki/Django" class="external_link" id="link_to_wiki_page">
```

이제 CSS를 HTML에 추가해보자. `blog/templates/blog/post_list.html` 파일을 열고 맨 처음 줄에 이 라인을 추가한다.

```
blog/templates/blog/post_list.html

{% load static %}
```

여기서 정적파일을 로드 하는 것이다.

다음 `<head>` 와 `</head>` 사이에 부트스트랩 CSS파일 링크 다음에 아래 코드를 추가한다.

```
blog/templates/blog/post_list.html

<link rel="stylesheet" href="{% static 'css/blog.css' %}">
```

코드를 잘 작성 했다면 아래와 같은 코드 모습이다.

```
blog/templates/blog/post_list.html

{% load static %}
<html>
  <head>
    <title>Django Girls blog</title>
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
    <link rel="stylesheet" href="{% static 'css/blog.css' %}">
  </head>
  <body>
    <div>
      <h1><a href="/">Django Girls Blog</a></h1>
    </div>

    {% for post in posts %}
      <div>
        <p>published: {{ post.published_date }}</p>
        <h1><a href="">{{ post.title }}</a></h1>
        <p>{{ post.text|linebreaksbr }}</p>
      </div>
    {% endfor %}
  </body>
</html>
```

# Django Girls Blog

published: 2020년 5월 1일 8:33 오후

## notice

This is a test notice.

Hello World

Hello Django~

published: 2020년 5월 1일 8:33 오후

## TEST

TEST content

안녕하세요~

published: 2020년 5월 1일 11:10 오후

## Sample title

Test

왼쪽 여백을 주려면 아래 코드 추가

```
blog/static/css/blog.css
```

```
body {
```

```
padding-left: 15px;
}
```

# Django Girls Blog

published: 2020년 5월 1일 8:33 오후

## notice

This is a test notice.

Hello World

Hello Django~

published: 2020년 5월 1일 8:33 오후

## 제목 폰트 변경

`blog/templates/blog/post_list.html` 파일 내 `<head>` 부분에 아래 코드를 추가.

```
blog/templates/blog/post_list.html
```

```
<link href="//fonts.googleapis.com/css?family=Lobster&subset=latin,latin-e
xt" rel="stylesheet" type="text/css">
```

이 라인은 Google 글꼴 (<https://www.google.com/fonts>)에서 **Lobster** 라는 글꼴을 가져온 것이다.

그리고 CSS파일 `blog/static/css/blog.css` 에서 `h1 a` 선언 블록 (중괄호{와}) 사이의 코드를 찾고 `font-family : 'Lobster';` 를 중괄호 사이에 추가하고 페이지를 새로고침 한다.

```
blog/static/css/blog.css
h1 a {
  color: #FCA205;
  font-family: 'Lobster';
}
```

# Django Girls Blog

published: 2020년 5월 1일 8:33 오후

## notice

This is a test notice.

Hello World

Hello Django~

published: 2020년 5월 1일 8:33 오후

앞서 말했듯이, CSS는 클래스 개념을 가지고 있다. 클래스는 HTML의 코드 일부에 이름을 붙이고 그 부분만 특정 스타일을 적용할 수 있게 해준다. 각기 다른 div를 (예를 들어 하나는 제목에 있고 하나는 글 본문에 있는 경우) 구분할 때 유용하다.

제목이 포함된 `div`에 `page-header`라는 클래스명을 붙여보자. :

```
blog/templates/blog/post_list.html

<div class="page-header">
  <h1><a href="/">Django Girls Blog</a></h1>
</div>
```

그리고 블로그 게시글을 나타내는 `div`에 `post`라고 클래스명을 붙여보자

```
blog/templates/blog/post_list.html

<div class="post">
  <p>published: {{ post.published_date }}</p>
  <h1><a href="">{{ post.title }}</a></h1>
  <p>{{ post.text|linebreaksbr }}</p>
</div>
```

이제 `blog/static/css/blog.css` 파일에 아래 코드를 복사, 붙여넣기 하자.

```
blog/static/css/blog.css
```

```
.page-header {  
    background-color: #ff9400;  
    margin-top: 0;  
    padding: 20px 20px 20px 40px;  
}
```

```
.page-header h1, .page-header h1 a, .page-header h1 a:visited, .page-header h1 a:active {  
    color: #ffffff;  
    font-size: 36pt;  
    text-decoration: none;  
}
```

```
.content {  
    margin-left: 40px;  
}
```

```
h1, h2, h3, h4 {  
    font-family: 'Lobster', cursive;  
}
```

```
.date {  
    color: #828282;  
}
```

```
.save {  
    float: right;  
}
```

```
.post-form textarea, .post-form input {  
    width: 100%;  
}
```

```
.top-menu, .top-menu:hover, .top-menu:visited {  
    color: #ffffff;  
    float: right;  
    font-size: 26pt;  
    margin-right: 20px;  
}
```

```
.post {  
    margin-bottom: 70px;  
}
```

```
.post h1 a, .post h1 a:visited {  
    color: #000000;  
}
```



blog/templates/blog/post\_list.html 파일 안에 아래 코드를 넣어보자

```
blog/templates/blog/post_list.html
```

```
<div class="content container">
  <div class="row">
    <div class="col-md-8">
      {% for post in posts %}
        <div class="post">
          <div class="date">
            <p>published: {{ post.published_date }}</p>
          </div>
          <h1><a href="">{{ post.title }}</a></h1>
          <p>{{ post.text|linebreaksbr }}</p>
        </div>
      {% endfor %}
    </div>
  </div>
</div>
```

# Django Girls Blog

published: 2020년 5월 1일 8:33 오후

## *notice*

This is a test notice.

Hello World

Hello Django~