

Just do it

JEJU CODING BASECAMP

YOUR BEHAVIORS MATTER
MORE THAN YOU THINK

MAKE A CHANGE

코딩 기초부터 서비스 런칭까지

Contents

1일차 오전

| | |
|----------------------------|----|
| 0. 프리뷰 ----- | 7 |
| 1. 로컬환경설정 ----- | 16 |
| 2. 서버환경설정 ----- | 25 |
| 3. Nginx 설치 및 구동(선택) ----- | 34 |

1일차 오후

| | |
|---------------|----|
| 1. HTML ----- | 37 |
| 2. CSS ----- | 41 |

1일차 저녁 - 2일차 오전

| | |
|-------------------------------------|----|
| 1. 자바스크립트의 기초 ----- | 47 |
| 2. 변수 ----- | 51 |
| 3. 자바스크립트의 연산 ----- | 56 |
| 4. 조건문 ----- | 61 |
| 5. 함수 ----- | 64 |
| 6. DOM(Document Object Model) ----- | 66 |
| 7. HTML FORM ----- | 64 |
| 8. 라이브러리 ----- | 71 |
| 9. jQuery의 필터와 실행방법 ----- | 73 |
| 10. 이벤트 ----- | 73 |

2일차 저녁

부트스트랩 ----- 79

3일차 오전 - 오후

Python 1부 ----- 149

3일차 저녁

홈페이지 기획 ----- 193

4일차 오전

Django ----- 233

1일차

오전

TUTORIAL 0. 프리뷰

일정표

| 시간 | Front-end | | Back-end | | 해커톤 |
|------------------|--------------------------------|---------------------------|--------------------|-------------------|-----------------------|
| | 월 | 화 | 수 | 목 | |
| 9:00 - 12:00 | front-end back-end 개요 설명 | Javascript & JQuery | Python 문법 | Django 활용 | |
| 12:00 - 13:30 | 점심 시간 | | | | 해 커 토 진 행 |
| 13:30 - 17:30 | HTML, CSS | 카카오 특강 | Python 문법 | Django 활용, 팀빌딩 | |
| 17:30 - 19:00 | 저녁 시간 및 개인 정비 | | | | |
| 19:00 - 21:00 | Javascript & JQuery | Bootstrap | 홈페이지 기획, 개발자 도구 | 자유코딩 | 심사 |

※ 상기 일정은 현장 사정에 따라 일부 변동될 수 있습니다.

상세カリ큘럼

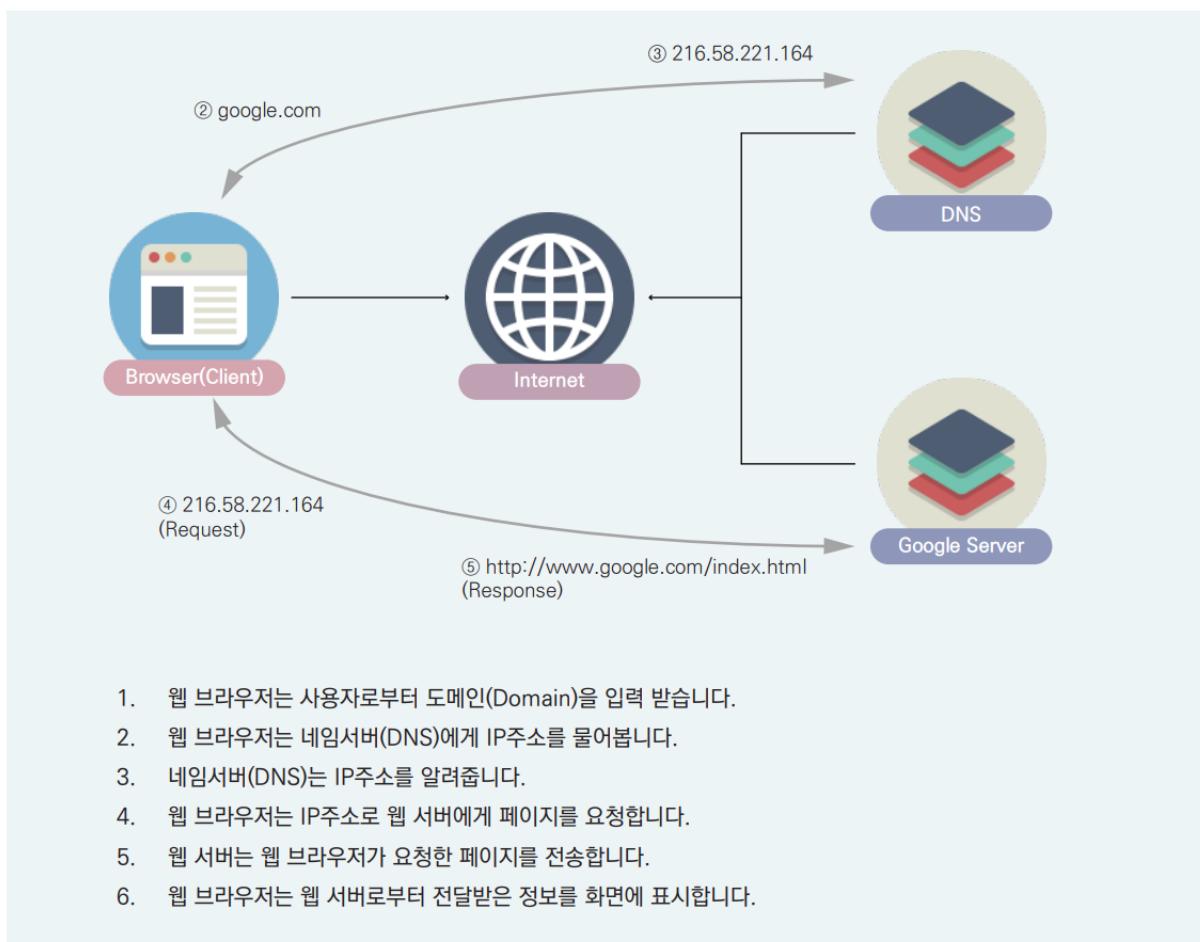
| 구분 | 교육시간 | | 세부 교육과정 | 장소 |
|----|---------------|----|---|-----|
| 월 | 09:00 ~ 12:00 | 3h | <ul style="list-style-type: none"> ▶ front-end, back-end 동작 방식 ▶ Network 동작 방식 | 카카오 |
| | 13:30 ~ 17:30 | 4h | <ul style="list-style-type: none"> ▶ HTML 태그, 동작 방식, 골격 구성 ▶ CSS 동작 방식 및 디자인 활용 | 카카오 |
| | 19:00 ~ 21:00 | 2h | <ul style="list-style-type: none"> ▶ Javascript 문법 및 응용 ▶ JQuery 문법 및 응용 | JDC |
| 화 | 09:00 ~ 12:00 | 3h | <ul style="list-style-type: none"> ▶ HTML, CSS, Javascript, JQuery를 활용한 웹 설계 ▶ 인터랙티브 웹 개발 | 카카오 |
| | 13:30 ~ 17:30 | 4h | <ul style="list-style-type: none"> ▶ 카카오 API(카카오 지도, 로그인 등) 응용 | 카카오 |
| | 19:00 ~ 21:00 | 2h | <ul style="list-style-type: none"> ▶ Web Framework를 이용한 웹 설계 개요 ▶ Bootstrap을 이용한 웹 개발 | JDC |
| 수 | 09:00 ~ 12:00 | 3h | <ul style="list-style-type: none"> ▶ 파이썬소개 및 개발환경 세팅 ▶ 타입과 변수, 연산자, 리스트, 튜플, 사전과 집합 | 카카오 |
| | 13:30 ~ 17:30 | 4h | <ul style="list-style-type: none"> ▶ 제어문(조건문과 반복 등) ▶ 함수와 클래스 | 카카오 |
| | 19:00 ~ 21:00 | 2h | <ul style="list-style-type: none"> ▶ 웹 서비스 기획 ▶ 카카오 Oven, FreeMind를 이용한 웹 기획 ▶ 개발자 도구(Atom 설정 및 버전관리를 위한 git) 설명 | JDC |
| 목 | 09:00 ~ 12:00 | 3h | <ul style="list-style-type: none"> ▶ 클라우드 서버 세팅 ▶ Django 세팅 및 아키텍쳐 설명 | 카카오 |
| | 13:30 ~ 17:30 | 4h | <ul style="list-style-type: none"> ▶ Django로 만들어보는 여행 블로그 ▶ 댓글 기능, 게시물 관리 기능 등 자주 쓰이는 기능 구현 ▶ 카카오 Open API를 활용한 Mesh-Up 웹 개발 | 카카오 |
| 금 | 09:00 ~ 12:00 | 3h | <ul style="list-style-type: none"> ▶ 카카오 투어 및 소개 ▶ 해커톤 및 멘토링 진행 | 카카오 |
| | 13:30 ~ 17:30 | 4h | <ul style="list-style-type: none"> ▶ 해커톤 및 멘토링 진행 | 카카오 |
| | 19:00 ~ 21:00 | 2h | <ul style="list-style-type: none"> ▶ 개인(팀)별 결과발표 | JDC |

※ 상기 일정은 현장 사정에 따라 일부 변동될 수 있습니다.

소통해요

- 어디서 오셨나요?
- 기대하는 것은 무엇인가요?
- 어떤 서비스를 만들고 싶으신가요?

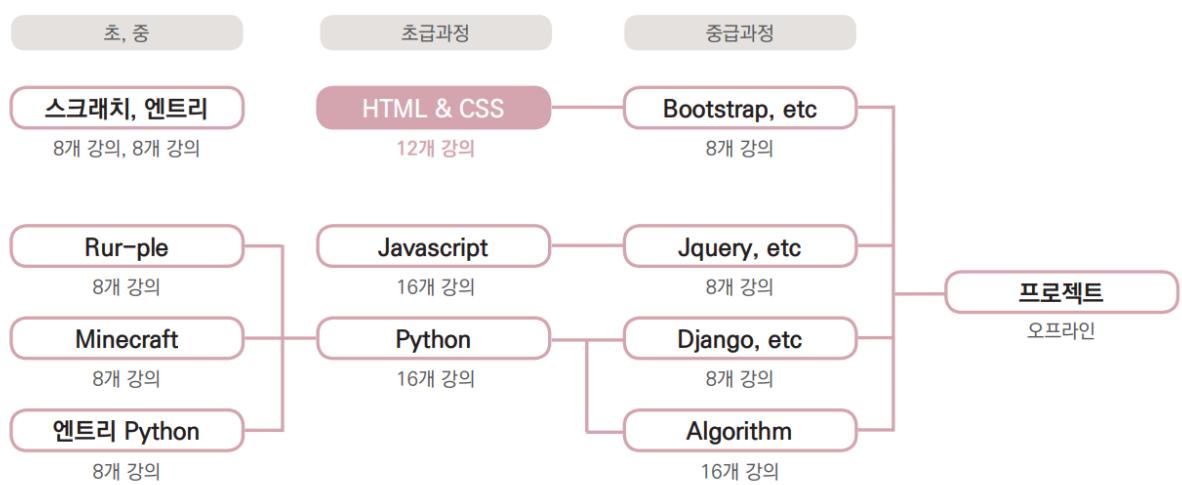
인터넷은 어떻게 구성되어 있나?



이미지 – **튜토리얼로 배우는 HTML & CSS**
디자인 – 김미경 팀장님



무엇을 어떻게 이용하여 개발할 것인가?



이미지 - 뷰토리얼로 배우는 HTML & CSS
디자인 - 김미경 팀장님

프로젝트는 어떻게 설계할 것인가?

1. 참가한 해커톤에서 완성 홈페이지 시연

- 시연
- usecase
- 프로젝트 문서

2. 구름IDE 세팅, 환경설정

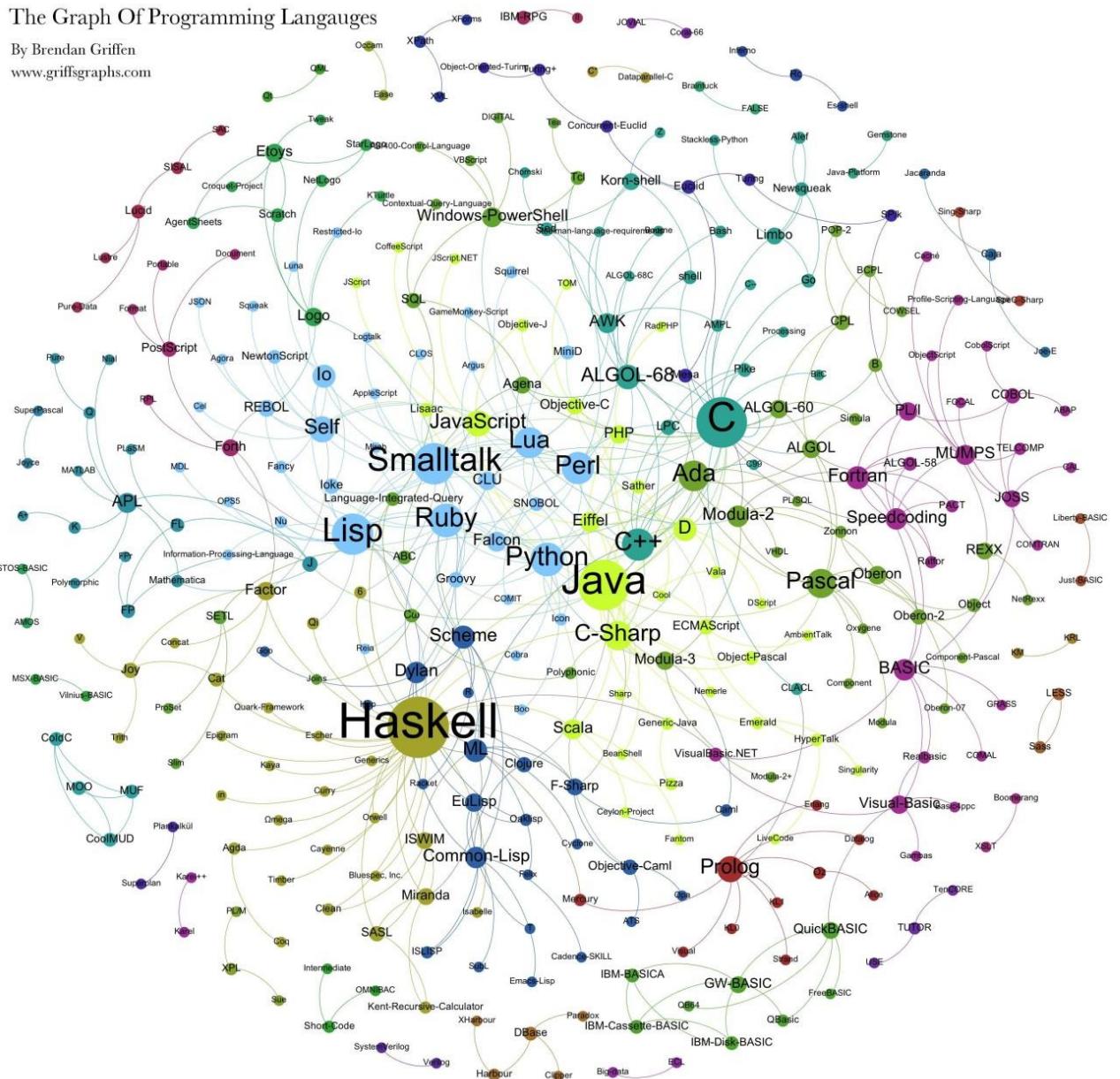
3. 구름 IDE에서 HTML, CSS, Javascript, Python 훑어보기

왜 파이썬(Python)인가?

The Graph Of Programming Languages

By Brendan Griffen

www.griffsgraphs.com



출처 <https://namu.wiki/w/프로그래밍 언어/종류>

왜 파이썬(Python)인가?

수 많은 언어 중 단연 주목받는 언어

직관적이고 배우기 쉬움

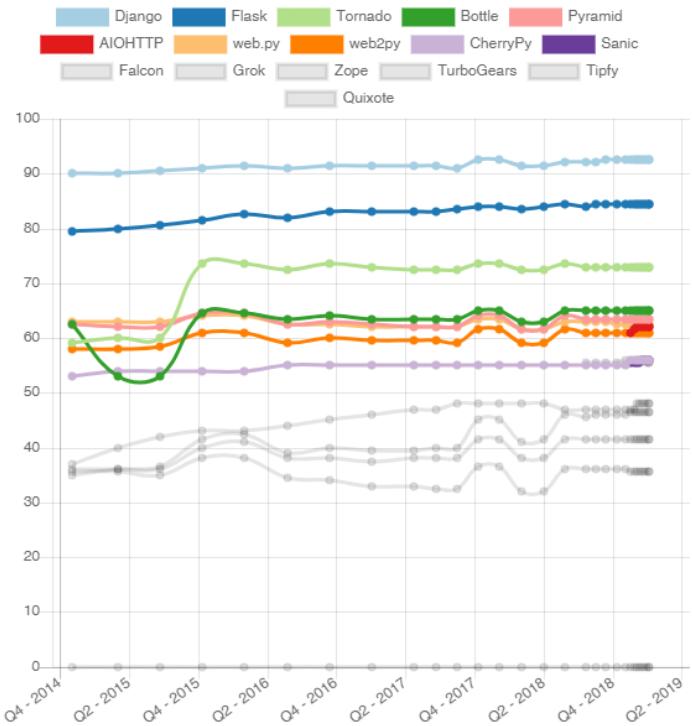
여러 Framework(Django, Flask 등)를 사용하여 개발을 쉽게 할 수 있음

해커의 언어라고도 불리움 - 라이브러리가 많아 할 수 있는 활동들이 많기 때문

왜 장고(Django)인가?

Python

| Framework | Score |
|------------|-------|
| Django | 92 |
| Flask | 84 |
| Tornado | 73 |
| Bottle | 65 |
| Pyramid | 63 |
| AIOHTTPP | 62 |
| web.py | 62 |
| web2py | 61 |
| CherryPy | 56 |
| Sanic | 56 |
| Falcon | 55 |
| Grok | 48 |
| Zope | 46 |
| TurboGears | 41 |
| Tipfy | 35 |
| Quixote | 0 |



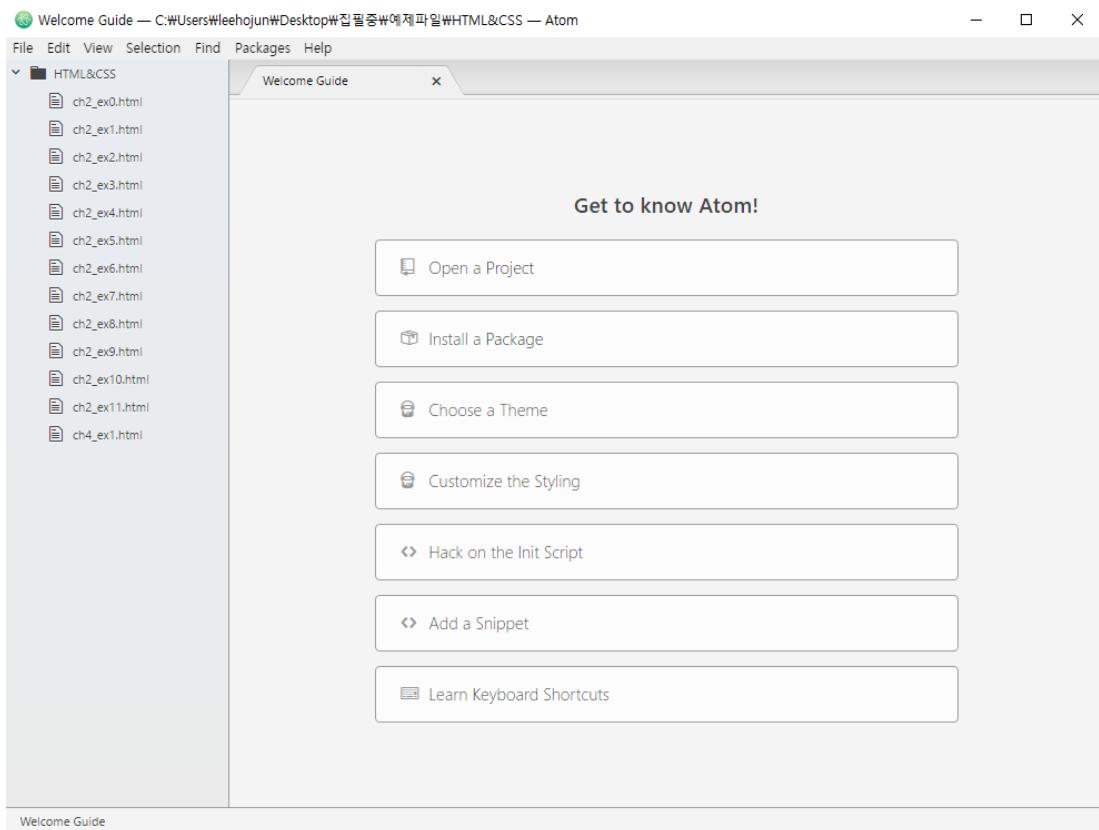
출처 <https://hotframeworks.com/languages/python>

Full-Stack Framework로 Front-end부터 Back-end까지 경험할 수 있음

약간(?) 무겁지만 뭐든지 만들 수 있음

Flask도 많이 쓰지만 이미 대세는 Django!

TUTORIAL 1 로컬 환경설정

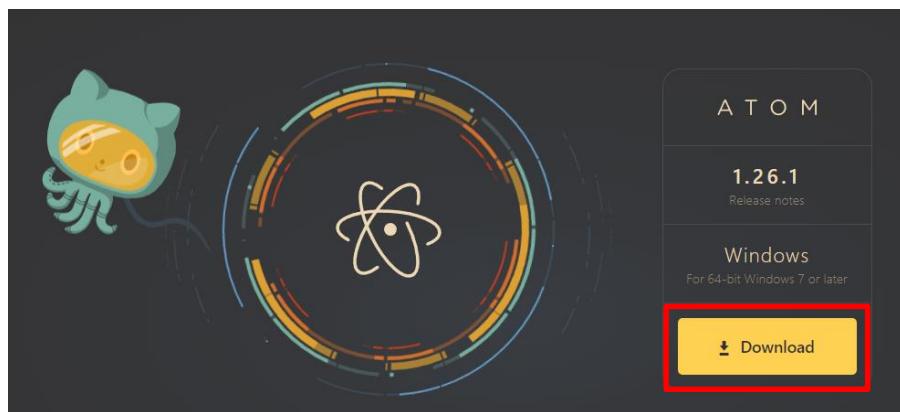


github에서 제작한 에디터입니다. git을 쓰시는 분들에게는 이보다 좋은 에디터는 없을 것으로 보입니다. 아직 git을 사용해보시지 않으셨다면 한번 사용해 보시길 권해드립니다. 추가 기능이 필요하시다면 패키지를 인스톨하여 원하는 에디터를 만들어갈 수 있습니다.

1. 로컬 환경설정 (계속)

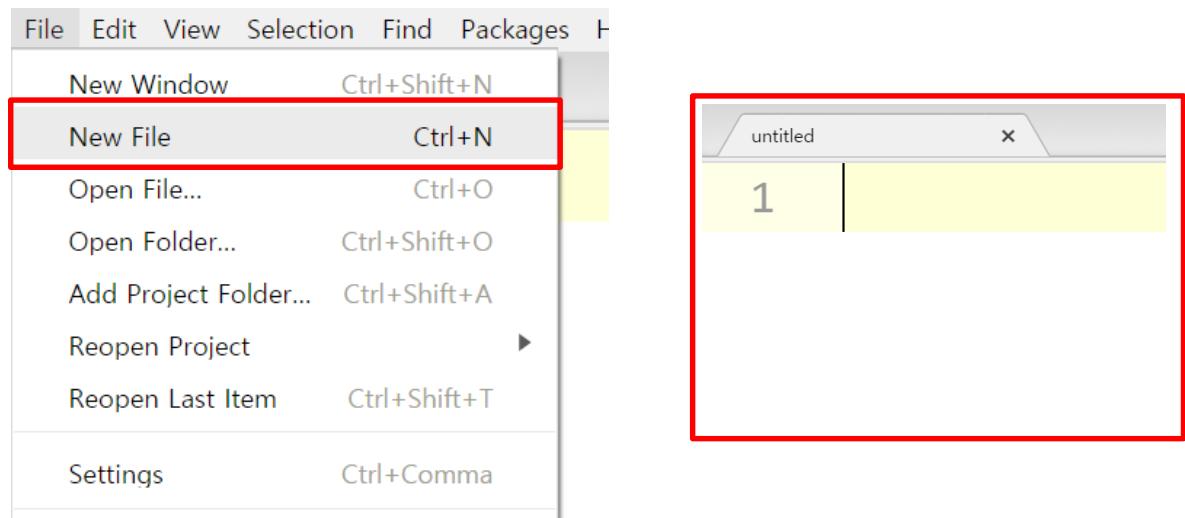
설치 방법

1) <https://atom.io> 홈페이지에 접속합니다. 구글에서 Atom을 검색하면 최상단에 있습니다. 설치 파일을 다운로드 받고 실행하셔서 설치를 진행하세요.



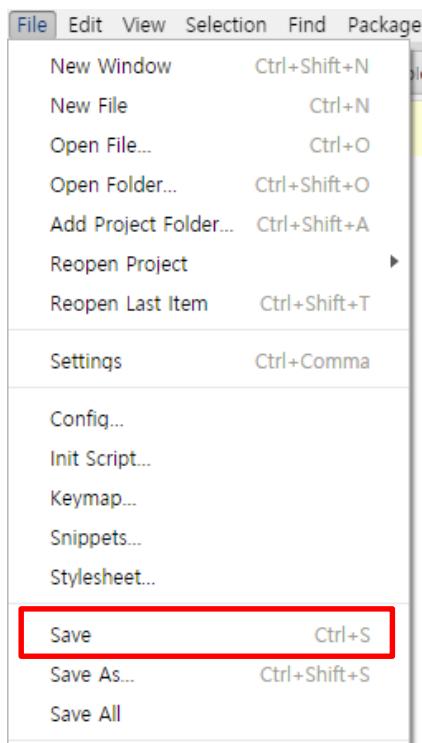
사용법

2) Atom을 실행하고 File > New File을 선택합니다. 단축키는 Ctrl + N 입니다.



1. 로컬 환경설정 (계속)

3) untitled 창에서 File, Save 를 클릭해 저장합니다. 저장할 때는 파일명 '.html'로 저장해야 합니다. '.html'로 저장한 파일에서는 코드하이라이팅이 적용됩니다.

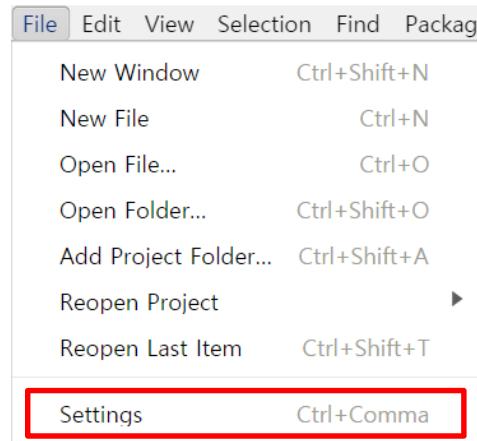


```
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>Document</title>
</head>
<body>
<h1>hello world</h1>
</body>
</html>
```

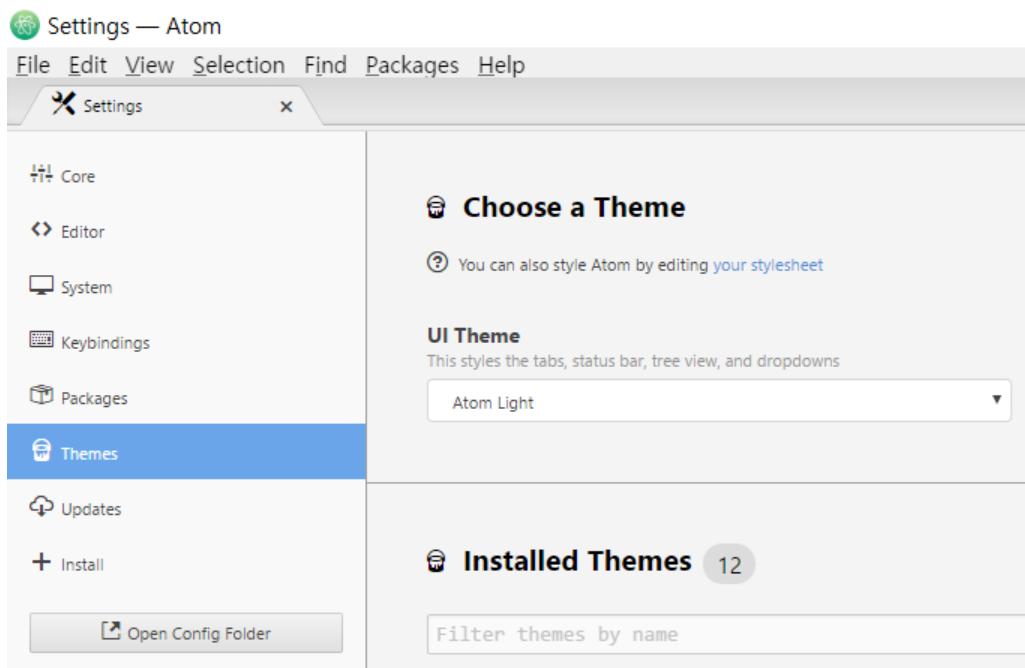
1. 로컬 환경설정 (계속)

테마 변경

1) File을 누르고 Settings를 누릅니다.



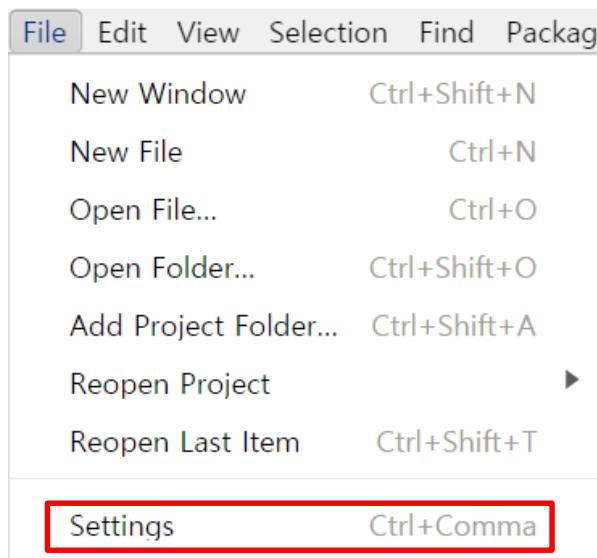
2) UI Theme에서 테마에 대한 설정을 변경할 수 있습니다. 이 책에서는 Atom Light 테마를 사용하도록 하겠습니다.



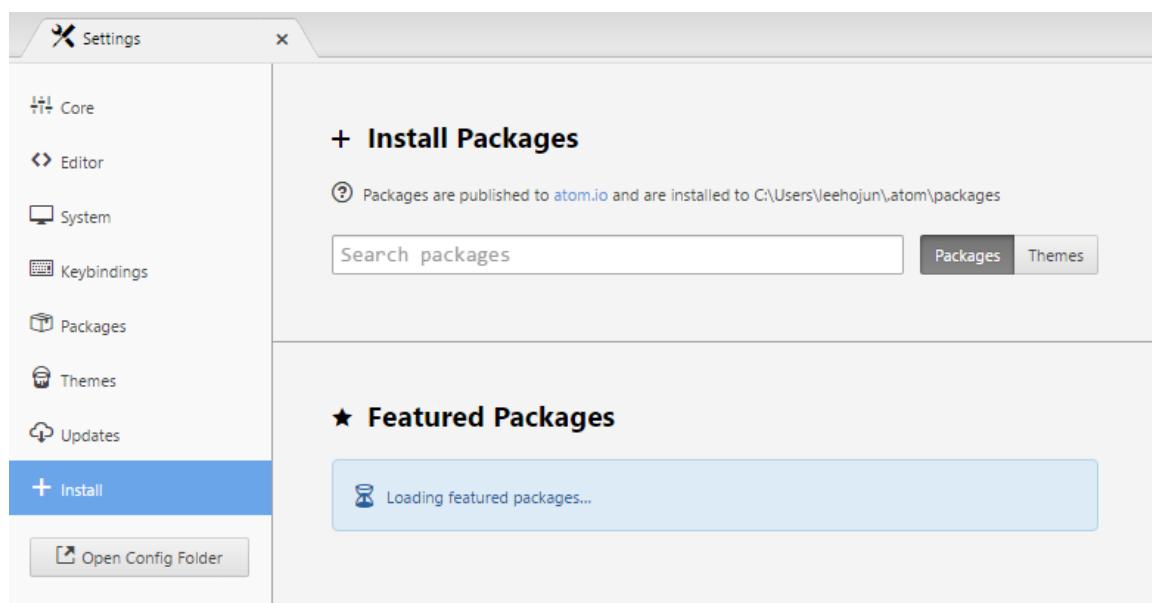
1. 로컬 환경설정 (계속)

유용한 기능

1) File을 누르고 Settings를 누릅니다.



2) Install에서 Editor를 Customizing할 수 있는 몇 가지 Packages를 설치하도록 하겠습니다.



1. 로컬 환경설정 (계속)

3) wrap이라 검색하고 엔터를 치면 wrap-guide가 나올 것입니다. 에디터 안에서 오른쪽에 라인을 그려주는 Package인데 지원 사용하는 편이 더 깔끔하기 때문에 Disable 시키도록 하겠습니다.

+ Install Packages

Packages are published to atom.io and are installed to C:\Users\leehojun\atom\packages

wrap

Packages Themes

wrap 0.1.0
An Atom package that does wrapping right.
danielchatfield Install

wrap-guide 0.40.2
Displays a vertical line at the 80th character in the editor. This packages uses the config value of `editor.preferredLineLength` when set.
atom Disable

4) 오른쪽 상단에 map이라 검색을 하면 minimap이 검색됩니다. Code mini Map을 그려주는 유용한 Package이니 설치하도록 합니다.

+ Install Packages

Packages are published to atom.io and are installed to C:\Users\leehojun\atom\packages

map

Packages Themes

minimap 4.28.2
A preview of the full source code.
atom-minimap Install

1. 로컬 환경설정 (계속)

5) emmet 패키지는 웹 관련 코드를 자동완성 시켜주는 패키지입니다. 상당히 유용하지만 사용법을 어느정도 숙지해야 하기 때문에 뒷장부터 사용법을 언급하도록 하겠습니다.

The screenshot shows the Atom package manager interface. At the top, there's a search bar with 'emmet' typed into it, and tabs for 'Packages' and 'Themes'. Below the search results, the 'emmet' package is listed. It has a version of '2.4.3', a description 'Emmet – the essential tool for web developers', and a download count of '1,878,644'. To the right of the package name is a green circular icon with a white arrow. On the far right of the listing are three buttons: 'Settings' (with a gear icon), 'Uninstall' (with a trash can icon), and 'Disable' (with a double slash icon). A green vertical bar is positioned to the right of the 'Disable' button.

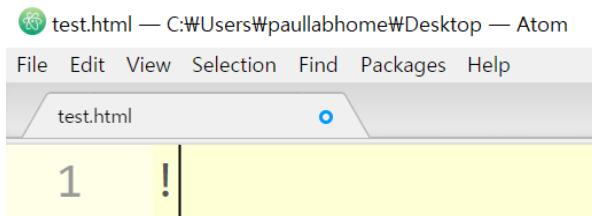
6) 이번에는 Editor로 들어와서 Soft Wrap에 체킹을 해줍니다. 라인이 허용치를 넘어가면 옆으로 스크롤이 생기는데 스크롤이 생기지 않고 아래로 넘어가도록 하는 옵션입니다.

The screenshot shows the Atom settings editor. The left sidebar has a tree view with categories like 'Core', 'Editor' (which is selected and highlighted in blue), 'URI Handling', 'System', 'Keybindings', 'Packages', 'Themes', 'Updates', and '+ Install'. At the bottom of the sidebar is a button labeled 'Open Config Folder'. The main area is titled 'Scroll Sensitivity' and describes how fast the editor scrolls when using a mouse or trackpad. It shows a 'Default: 40' input field. Below this are several checkboxes with descriptions: 'Show Cursor On Selection' (checked), 'Show Indent Guide' (unchecked), 'Show Invisibles' (unchecked), 'Show Line Numbers' (checked), 'Soft Tabs' (checked), and 'Soft Wrap' (checked). The 'Soft Wrap' checkbox is highlighted with a red rectangle. A tooltip for 'Soft Wrap' explains that it wraps lines that exceed the width of the window, and that when 'Soft Wrap At Preferred Line Length' is set, it follows the 'Preferred Line Length' setting.

1. 로컬 환경설정 (계속)

Emmet 패키지 사용법

1) emmet 패키지 사용법을 어느정도 숙지해 두는 것은 생산성을 높여줍니다. 가장 먼저 파일을 [.html로 저장](#)하였다면 !를 입력하고 [Tap](#)을 눌러보세요. 그럼 아래와 같이 기본 골격이 한번에 만들어지는 것을 볼 수 있습니다.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
6     content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible"
8     content="ie=edge">
9   <title>Document</title>
10 </head>
11 <body>
12 </html>
```

1. 로컬 환경설정 (계속)

```
<body>
  h1
  h1+h1
  h1*3
  h1{hello world}*3
  table>(tr>th*3)+(tr>td*3)
  h1#one+div.two+div.three.four.five
  h1.test$$*3
</body>
```

2) 이번에는 위와 같이 입력하고 [하나의 라인](#) 끝에서 Tab을 눌러보세요. 예를 들어 아래와 같이 h1 뒤에서 Tab을 눌러보십시오. 그러면 그 아래 화면과 같이 변하는 것을 볼 수 있습니다. 위의 예제를 모두 실습하면 그 원리에 대해 이해하실 수 있으실 겁니다.

1 h1|

1 <h1></h1>

1. 로컬 환경설정

이 책에서는 대부분 <script> 내용 </script> 안에 부분만 다루고 있습니다. 아래 캡쳐화면에서 빨간색 네모에 해당하는 부분입니다. 이에 유념하셔서 공부하시기 바랍니다.

< J00.html > Code

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>j00.html</title>
6  </head>
7  <body>
8
9  <script>
10 var 알파벳 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
11 var 길이 = txt.length;
12 document.write("알파벳의 길이: " + 길이 + '<br>')
13 </script>
14
15 </body>
16 </html>
```

TUTORIAL 2 서버 환경설정

어디에서 개발할 것인가?

클라우드 서비스는 크게 3가지로 나눌 수 있습니다.

1. SaaS(Software as a Service)

SaaS는 가장 사용자 단에 친밀한 서비스이며 네트워크를 통해 애플리케이션 기능을 이용할 수 있는 서비스입니다.

2. PaaS(Platform as a Service)

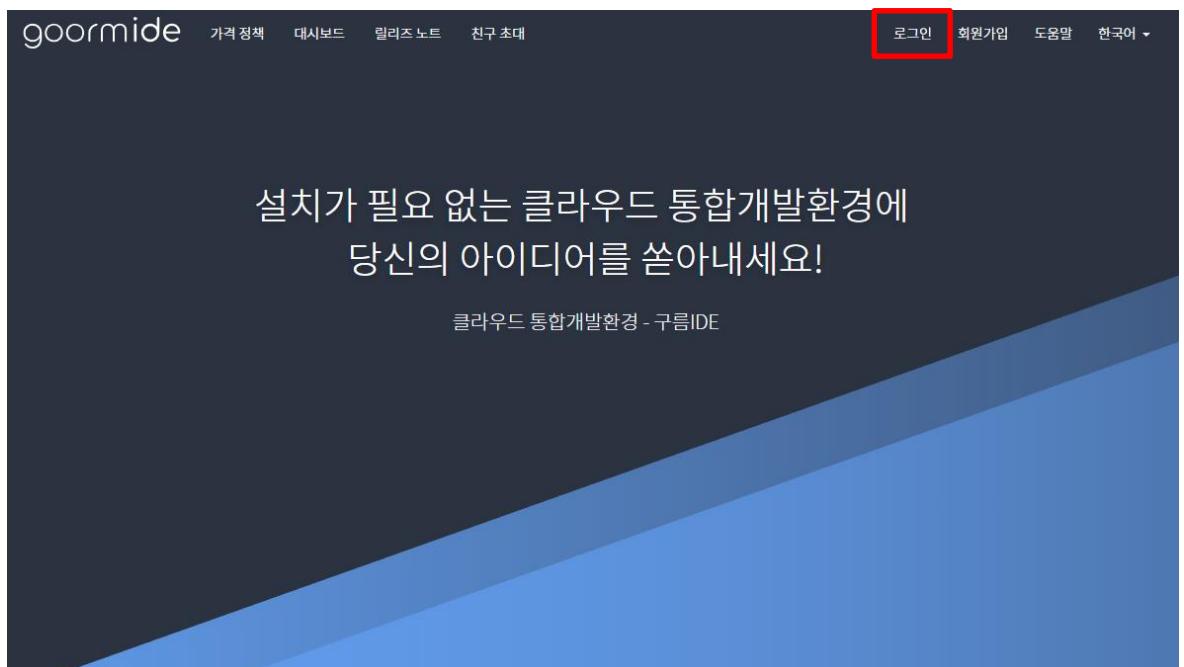
PaaS는 윈도우와 리눅스 같은 운영체제를 제공하고 개발 가능한 플랫폼도 함께 제공되는 클라우드 서비스입니다.

3. IaaS(infrastructure as a Service)

IaaS는 인프라를 제공하는 클라우드 서비스입니다. 기업에서 특히 많이 쓰입니다.

2. 서버 환경설정 (계속)

구름IDE 환경설정을 해보도록 하겠습니다. 아래 화면을 보시면서 빨간색 네모를 따라 그대로 따라하시면 됩니다.



<https://ide.goorm.io/>

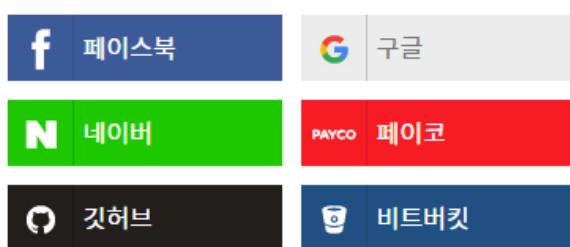
2. 서버 환경설정 (계속)

회원가입을 해야 합니다. 무료계정도 컨테이너 5개를 사용할 수 있는 강력한 서비스입니다. 회원가입을 하면 자동적으로 대시보드에 들어갑니다. 만약 메인화면으로 나오셨다면 대시보드를 눌러주세요.

로그인

이메일 저장

다른 서비스 계정으로 로그인



[비밀번호 재설정](#)

[회원가입](#)

goormide

[가격 정책](#)

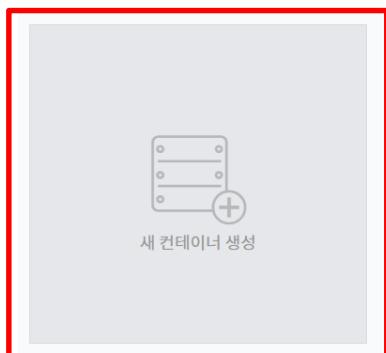
[대시보드](#)

[릴리즈 노트](#)

[친구 초대](#)

2. 서버 환경설정 (계속)

다음은 대시보드 화면입니다. 새 컨테이너 생성을 누르셔서 컨테이너를 생성해 주세요. 하나의 컨테이너는 하나의 컴퓨터를 세팅하는 것과 같습니다. Python을 선택하신 다음 생성하기를 눌러주세요.



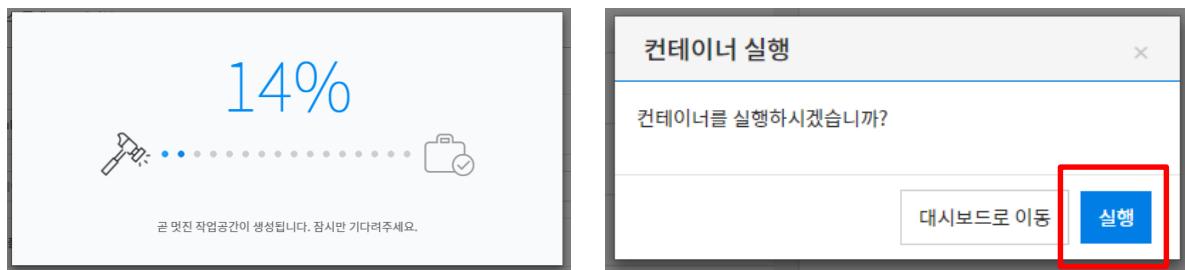
컨테이너 생성

컨테이너 생성

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|---|--------|-------|---------------|--------|-------|------------------|------------|------|------|-------|---------|-------------|----|------------------|---------|-------|--------|-------|------|-------|-----|-------|---------|-------|--------|-------------|-------|------|------------|-------|
| 소스 | <input checked="" type="radio"/> 템플릿 <input type="radio"/> Github <input type="radio"/> Bitbucket <input type="radio"/> Git / SVN <input type="radio"/> 압축파일 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 이름 | tutorialdjango | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 설명 | 여행블로그 튜토리얼로 배우는 <u>Django</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 소프트웨어 스택 선택 | <table border="1"><tr><td>Search</td></tr><tr><td>C/C++</td></tr><tr><td>Python</td></tr><tr><td>Django</td></tr><tr><td>Flask</td></tr><tr><td>Jupyter Notebook</td></tr><tr><td>TensorFlow</td></tr><tr><td>PyQt</td></tr><tr><td>JAVA</td></tr><tr><td>Maven</td></tr></table> <p>The "Python" option is highlighted with a red box.</p> <table border="1"><tr><td>프로젝트 유형</td><td>Python 프로젝트</td></tr><tr><td>OS</td><td>Ubuntu 14.04 LTS</td></tr><tr><td>Python3</td><td>3.4.3</td></tr><tr><td>Python</td><td>2.7.6</td></tr><tr><td>pip3</td><td>9.0.1</td></tr><tr><td>pip</td><td>9.0.1</td></tr><tr><td>Jupyter</td><td>4.3.0</td></tr><tr><td>Django</td><td>1.11.12 LTS</td></tr><tr><td>Flask</td><td>0.12</td></tr><tr><td>TensorFlow</td><td>1.3.0</td></tr></table> <p>A red box surrounds the "생성하기" (Create) button at the bottom right.</p> | Search | C/C++ | Python | Django | Flask | Jupyter Notebook | TensorFlow | PyQt | JAVA | Maven | 프로젝트 유형 | Python 프로젝트 | OS | Ubuntu 14.04 LTS | Python3 | 3.4.3 | Python | 2.7.6 | pip3 | 9.0.1 | pip | 9.0.1 | Jupyter | 4.3.0 | Django | 1.11.12 LTS | Flask | 0.12 | TensorFlow | 1.3.0 |
| Search | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C/C++ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter Notebook | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PyQt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JAVA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Maven | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 프로젝트 유형 | Python 프로젝트 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OS | Ubuntu 14.04 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python3 | 3.4.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | 2.7.6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip3 | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter | 4.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | 1.11.12 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | 0.12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | 1.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

2. 서버 환경설정 (계속)

컨테이너가 만들어지고 있다는 화면이 나오고 곧 컨테이너 생성이 완료되었다고 뜹니다. 컨테이너 실행을 누르셔서 컨테이너 안으로 들어가주세요. 만약 모르고 닫으셨다면 대시보드에서 해당 컨테이너 실행을 누르시면 됩니다.



This screenshot shows a detailed view of a container named "tutorialdjango". At the top, it displays the last update date: "2018. 4. 30. Updated" and a "+ 더보기" (More) button. Below this, the container's configuration is listed:

| | |
|----------|-----------------------|
| 소프트웨어 스택 | Python |
| 저장 공간 | 10GB |
| 항상 켜두기 | PREMIUM 상품 구매 시 사용 가능 |
| SSH | 컨테이너 실행 시 사용 가능 |

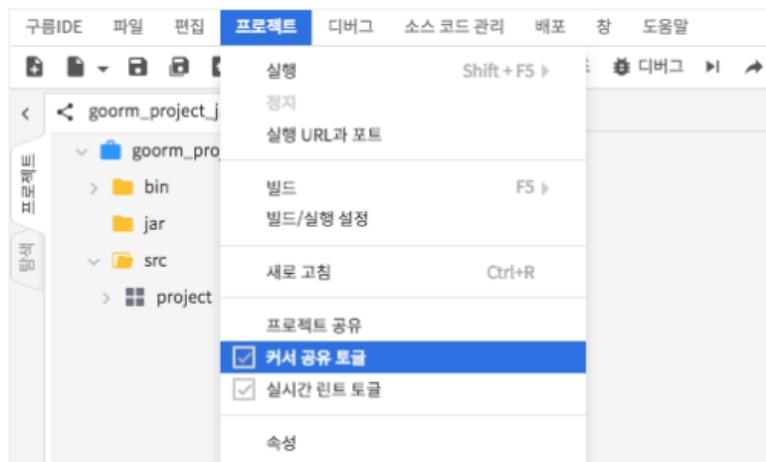
At the bottom of the page are two large buttons: a dark grey "실행" (Run) button with white text, which is also highlighted with a red box, and a blue "터미널 실행" (Terminal Run) button.

2. 서버 환경설정 (계속)

컨테이너가 로딩되는 화면입니다. 주로 구름에 TIP이나 유명인사의 명언이 나옵니다.

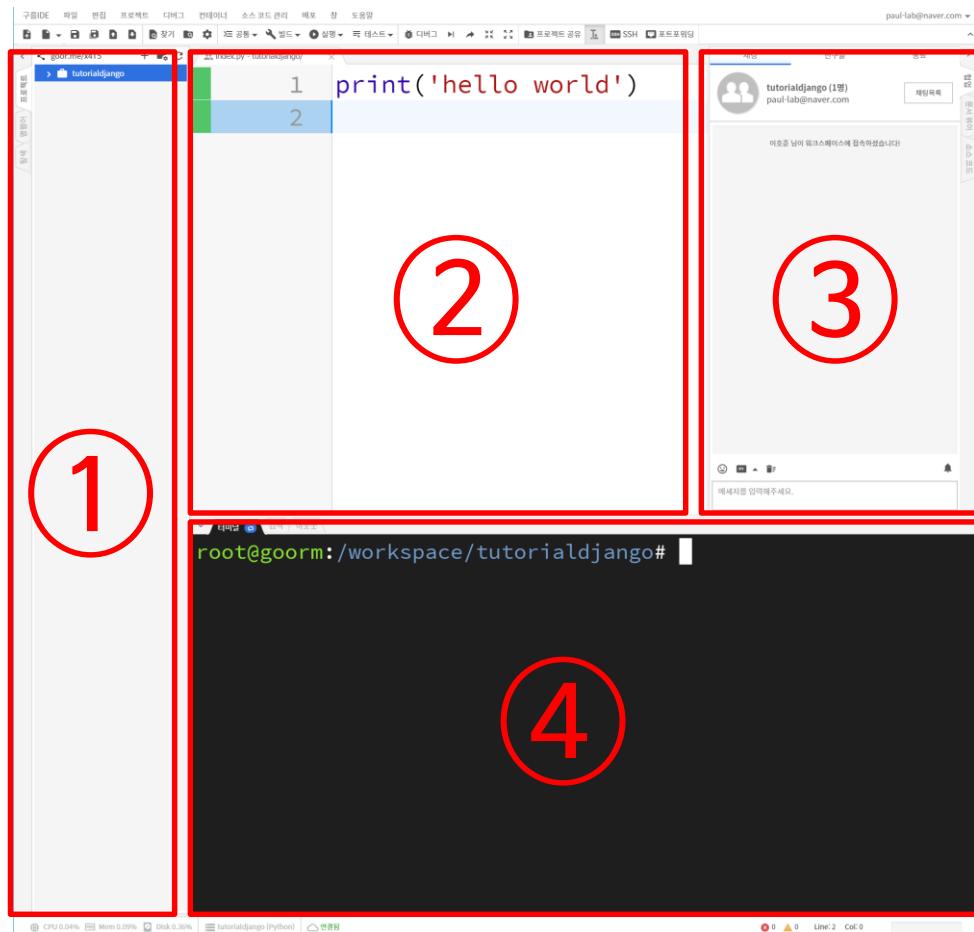


프로젝트 로딩 중입니다.



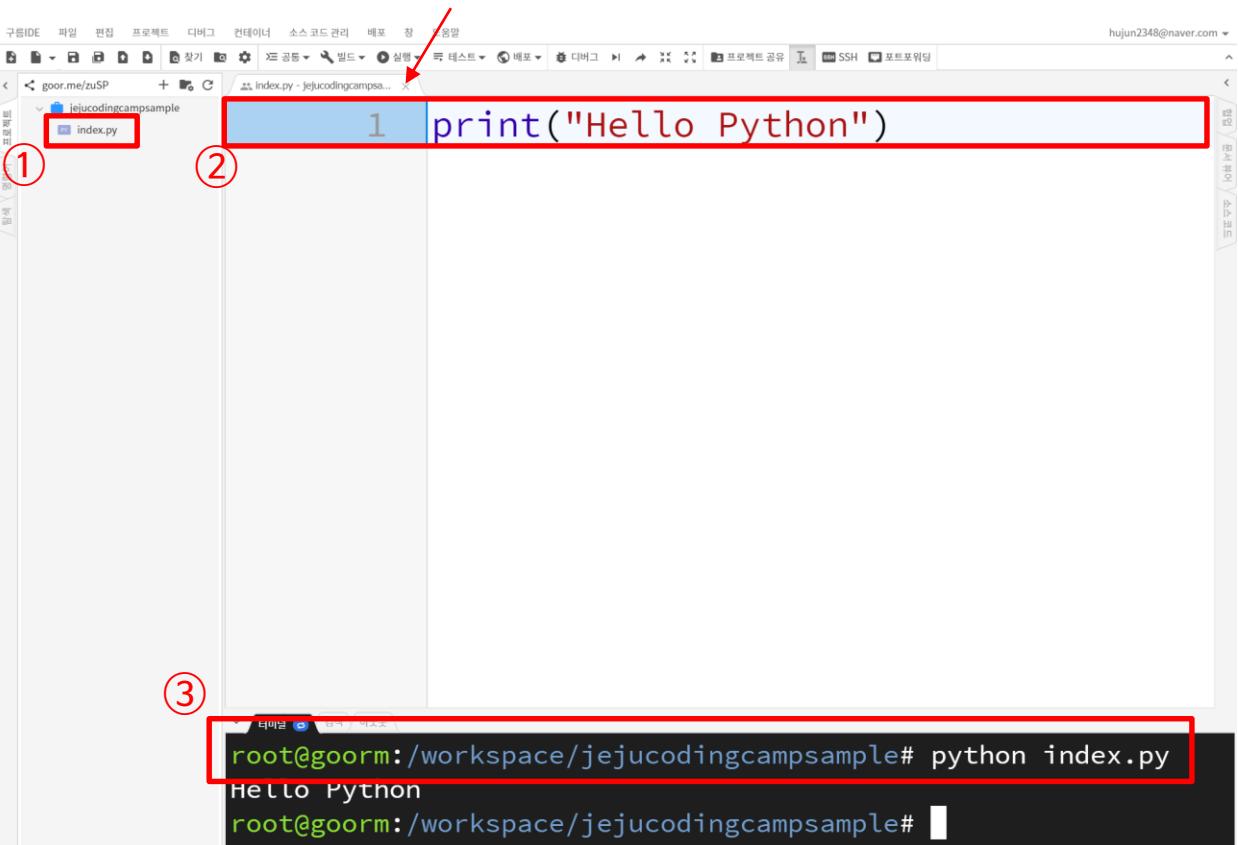
공유된 프로젝트에서 동료의 커서를 보고 싶지 않다면 '프로젝트 > 커서 공유 토글'의 체크 상태를 해제해 주세요.

2. 서버 환경설정 (계속)



1. 풀더 구조를 볼 수 있는 공간입니다. 우측 상단에 보시면 새로고침 버튼이 있어요. 콘솔창에서 뭔가 작업을 했는데 보이지 않는다면 새로고침을 해보시기 바랍니다.
2. txt, html, py파일 등 다양한 파일을 edit 할 수 있는 공간입니다.
3. 협업을 위한 공간이에요. 이 책에서는 사용하지 않는 기능입니다. 채팅 등 협업을 위한 다양한 기능을 제공합니다.
4. 콘솔창입니다. 우리는 대부분의 명령어를 이곳에 입력하게 됩니다.

2. 서버 환경설정 (계속)

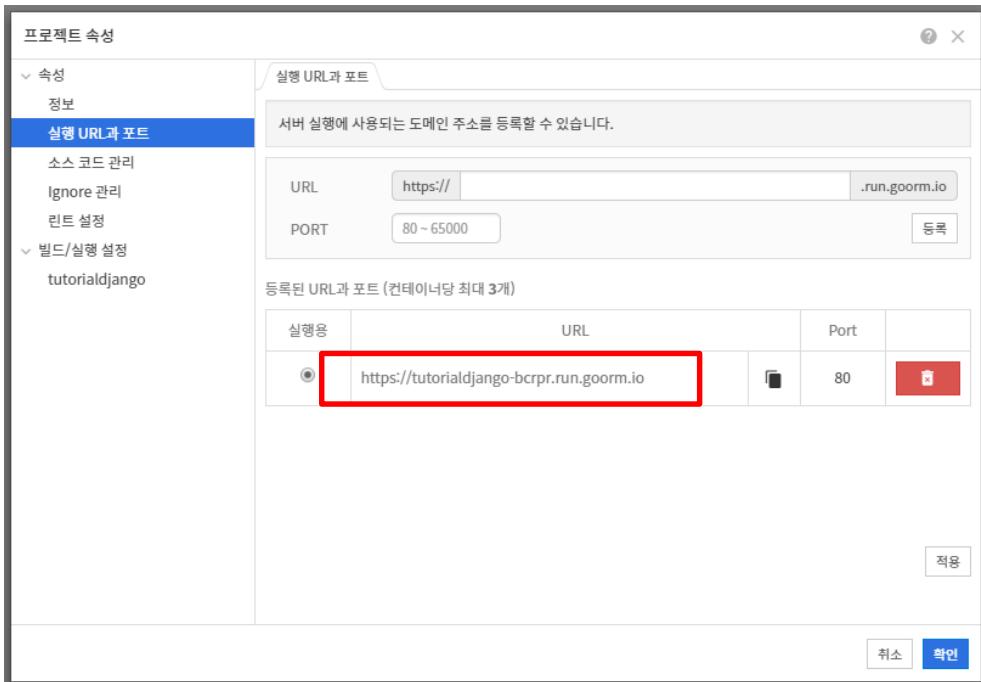
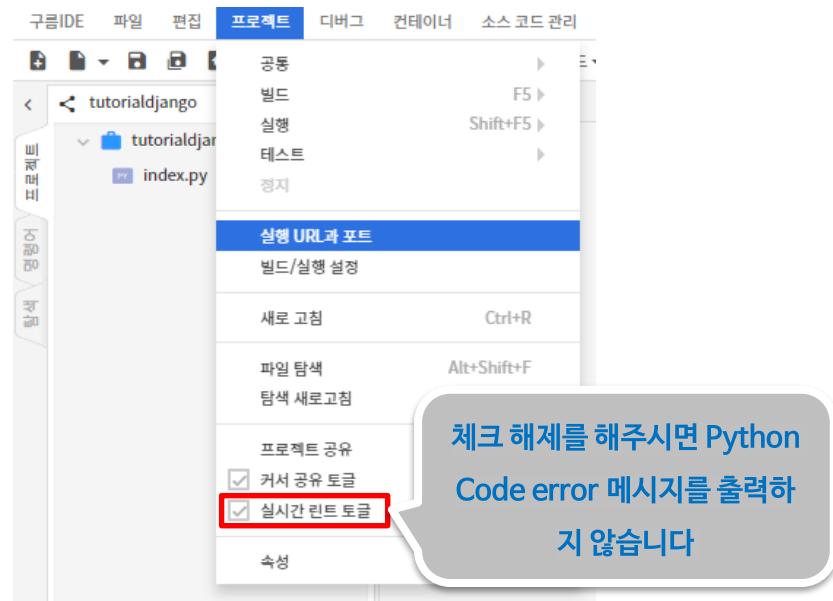


1. 왼쪽 프로젝트 바에서 index.py를 클릭합니다. 그러면 오른쪽 창에 해당 파일의 소스코드가 보입니다. 안에 있는 내용을 수정해보세요.
2. 안에 있는 내용을 수정하면 상단 화살표로 그려 넣은 곳이 '*'로 바뀝니다. 해당 표시는 저장이 안되었다는 표시입니다. Ctrl + S를 누르시면 저장이 되고 다시 'X'로 돌아옵니다.
3. 저장이 되신 다음 아래 콘솔창에서 python index.py를 입력해보세요. 작성하신 문서가 실행됩니다!

2. 서버 환경설정

프로젝트를 누르시고 실행 URL과 포트를 눌러주세요. 아래 빨간색 네모로 표시된 것이 우리가 앞으로 사용할 URL입니다. 아직은 실행되지 않습니다.

* 만약 등록이 되어 있지 않다면 위에서 원하는 url을 입력하고 PORT는 80으로 하여 등록버튼을 눌러주세요.



TUTORIAL 3 Nginx 설치 및 구동(선택)

이제 우리가 작성했었던 HTML, CSS를 서버로 구동시켜 보겠습니다.

* 여기서 부터는 선택사항입니다. 안해보셔도 괜찮습니다.

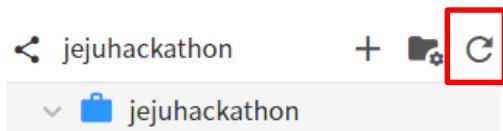
```
root@goorm:/workspace/여러분ID# python --virsion  
root@goorm:/workspace/여러분ID# mkdir web  
root@goorm:/workspace/여러분ID# cd web  
root@goorm:/workspace/여러분ID/web# sudo apt-get update  
root@goorm:/workspace/여러분ID/web# sudo apt-get install nginx  
root@goorm:/workspace/여러분ID/web# vi /etc/nginx/sites-available/default
```

vi 에디터를 이용하여

root /var/www/html; 을 root /workspace/컨테이너이름/web; 로 고치세요.
(i를 누르면 편집이 됩니다. 편집을 다 하신 후에는 ESC, :, wq!를 순서대로 입력하세요.)

```
root@goorm:/workspace/여러분ID/web# sudo service nginx start
```

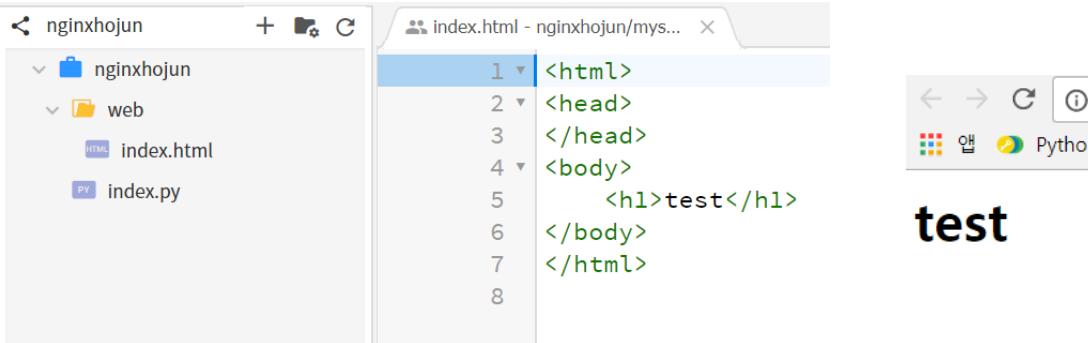
멈추고 싶을 때에는 sudo service nginx stop를 입력합니다.



작업하신 파일이 안보이실 때에는 새로고침 버튼을 클릭해주세요.

3. Nginx 설치 및 구동

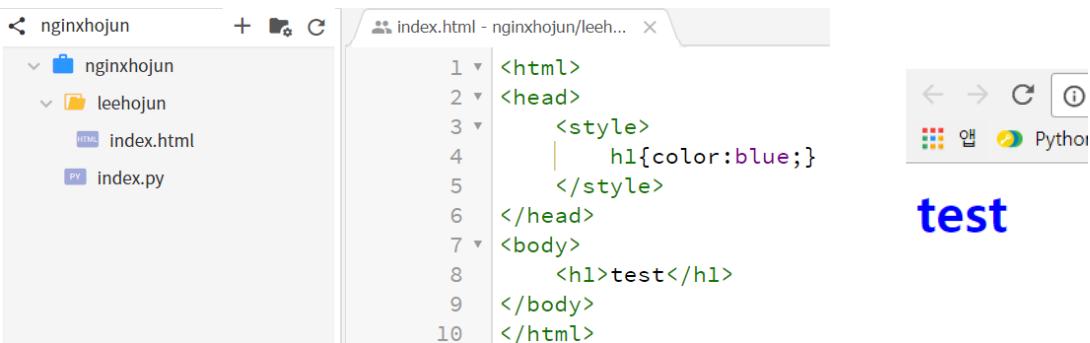
구름IDE에서 HTML, CSS, Javascript을 순차적으로 실행해보도록 하겠습니다.



The screenshot shows the Cloud IDE interface with the project structure on the left and the code editor on the right. The code editor displays the following HTML file:

```
1 <html>
2 <head>
3 </head>
4 <body>
5   <h1>test</h1>
6 </body>
7 </html>
```

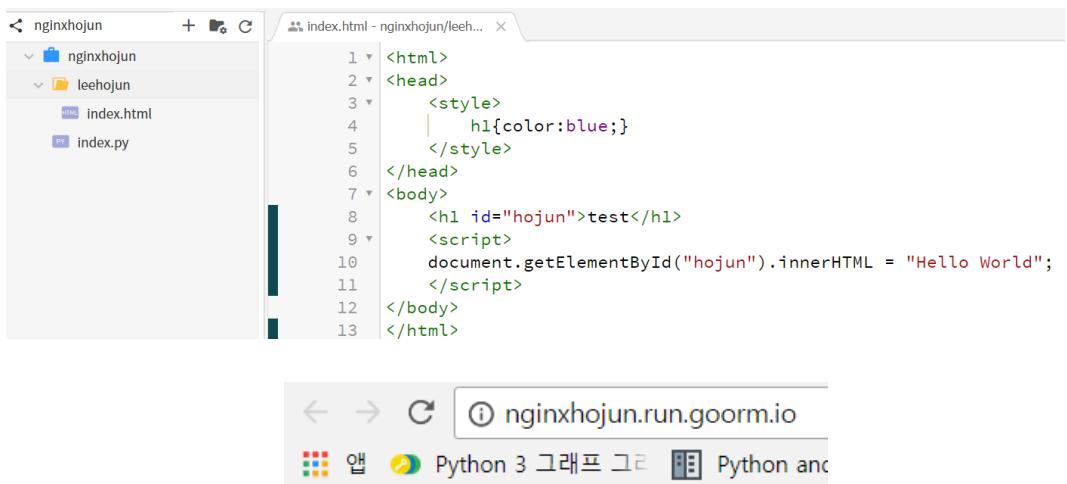
To the right of the code editor, the output window shows the word "test" in bold black font.



The screenshot shows the Cloud IDE interface with the project structure on the left and the code editor on the right. The code editor displays the following HTML file with a CSS style:

```
1 <html>
2 <head>
3   <style>
4     h1{color:blue;}
5   </style>
6 </head>
7 <body>
8   <h1>test</h1>
9 </body>
10 </html>
```

To the right of the code editor, the output window shows the word "test" in blue color.



The screenshot shows the Cloud IDE interface with the project structure on the left and the code editor on the right. The code editor displays the following HTML file with both CSS and JavaScript:

```
1 <html>
2 <head>
3   <style>
4     h1{color:blue;}
5   </style>
6 </head>
7 <body>
8   <h1 id="hojun">test</h1>
9   <script>
10    document.getElementById("hojun").innerHTML = "Hello World";
11  </script>
12 </body>
13 </html>
```

To the right of the code editor, the output window shows the word "Hello World" in blue color. Below the code editor, the browser address bar shows "nginxhojun.run.goorm.io".

Hello World

1일차

오후

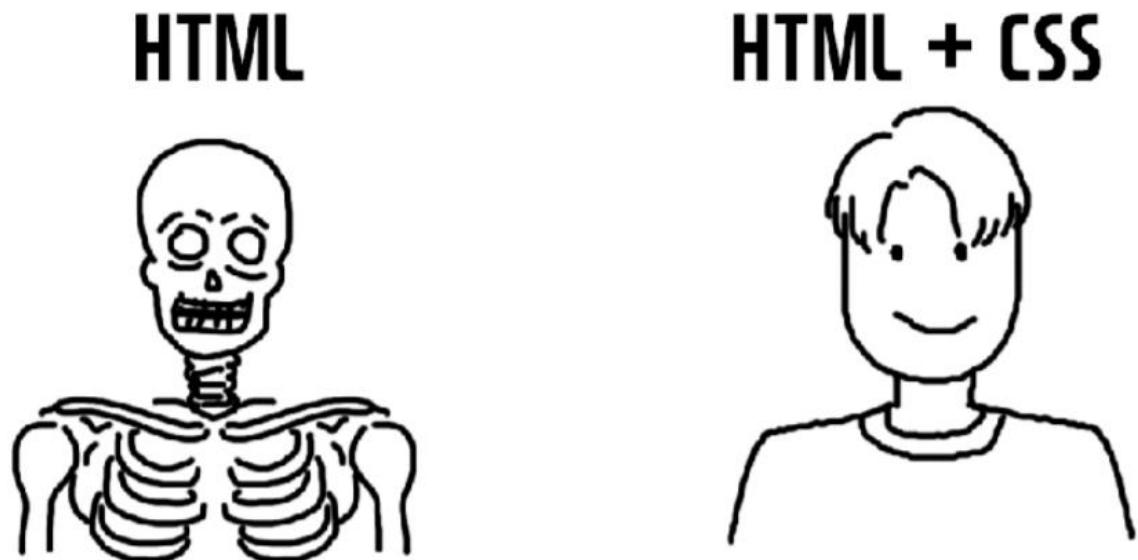
HTML & CSS

TUTORIAL 1 HTML

HTML과 CSS는 무엇일까요? 그리고 무슨 차이가 있을까요?

HTML은 Hyper Text Markup Language의 약자로, Hyper Text(하이퍼텍스트)는 링크로 연결된 텍스트 페이지, Markup Language(마크업 언어)는 태그 등을 이용해서 문서를 작성하도록 도와주는 언어를 의미합니다. 확장자명은 .html입니다.

CSS는 Cascading Style Sheets의 약자로, 웹 페이지의 전반적인 스타일을 작성해 저장해두는 시트를 의미합니다. 확장자명은 .css입니다.



홈페이지 시연 <http://studiomeal.com/>

1. HTML (계속)

XHTML이란 웹페이지를 제작하기 위해 사용되는 HTML4(지금 사용하고 있는 HTML5의 이전 버전)을 XML에 맞도록 재정의한 언어라고 할 수 있습니다. HTML5보다 조금 더 구조화된 형식과 엄격한 문법을 가지고 있는데요, 구체적으로 어떤 차이가 있는지 아래 표를 살펴봅시다.

| | XHTML1.0 | HTML5 |
|---|--|---------------------------------------|
| DOCTYPE (독타입, DTD; Document Type Declaration) | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> | <!DOCTYPE html> |
| character set (캐릭터 셋, charset) | <meta http-equiv="content-type" content="text/html; charset=UTF-8"> | <meta charset="UTF-8"> |
| type 표시 여부 | 생략 및 단축 불가 <link type="text/css" rel="stylesheet"> | 생략 및 단축 가능 <link rel="stylesheet"> |
| 태그 닫기 여부 | 단독으로 쓰이는 태그는 /> 으로 닫아야함 <hr /> | 닫아줄 필요 없음 <hr> |

- DOCTYPE : 웹페이지의 맨 처음에 선언되어 어떤 종류의 html인지 웹 브라우저에게 알려줍니다.
- meta charset : 해당 페이지의 인코딩 방식을 나타냅니다.

1. HTML (계속)

링크의 끓음입니다.

모든 링크 끓음에 nav를 사용할 필요는 없습니다. 일반저공로 메인 네비게이션 역할의 링크 끓음에 사용합니다.

페이지의 주요 내용의 흐름과 관련없는 다른 내용을 나타냅니다. 주로 사이드바 형태의 디자인으로 표현되며 각주에는 aside를 사용하지 않습니다.

Sectioning content의 풋터입니다. 풋터라고 해서 반드시 끝에 위치할 필요는 없습니다.

* Sectioning content : 헤딩과 풋터의 범위를 정의합니다.

`<h1> ... <h6>`

Sectioning content의 제목을 나타냅니다. 서브타이틀로 사용하면 안됩니다.

(틀린 예)

```
<header>
<h1>집에서 맛있는 피자 만들기</h1>
<h2>주위에서 쉽게 구입할 수 있는 재료를 중심으로
</h2>
</header>
```

Sectioning content의 헤더를 나타냅니다. 일반적으로 헤딩 엘리먼트를 포함하지만 반드시 그럴 필요는 없습니다. 헤딩 없이 목차, 검색폼 같은 내용을 header로 감싸도 무방합니다.

문서에서 일반적인 섹션의 의미입니다. 하나님의 주제로 그룹화된 컨텐츠를 담습니다. 각 섹션의 주제는 일반적으로 섹션 요소의 하위 항목으로 제목을 포함시켜 식별해야 합니다.

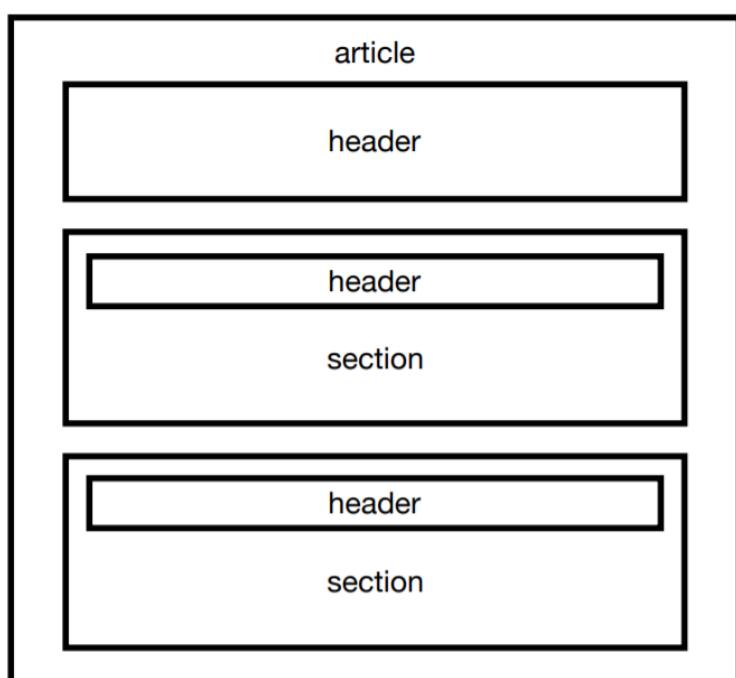
다른 곳에 단독으로 사용이 가능한 내용을 담는 엘리먼트입니다. 잡지의 기사, 포럼이나 블로그 게시글, 사용자가 남긴 댓글, 위젯 등이 될 수 있습니다. 중첩해서 사용할 경우 하위의 article들은 상위 article에 관련된 내용임을 의미합니다.

(좋은 예)

```
<header>
<h1>집에서 맛있는 피자 만들기</h1>
<p>주위에서 쉽게 구입할 수 있는 재료를 중심으로
</p>
</header>
```

1. HTML (계속)

앞서 배운 태그를 사용해서 레이아웃을 만들어 볼까요? 왼쪽 화면의 레이아웃을 코드로 작성하면 오른쪽과 같습니다.



<article>

<header>

내용

</header>

<section>

<header></header>

내용

</section>

<section>

<header></header>

내용

</section>

</article>

TUTORIAL 2 CSS

HTML를 예쁘게, 보기 좋게 꾸며주는 CSS를 적용하는 방법은 크게 3가지가 있습니다.

외부 CSS 파일 로드하기

```
<head>
  <link rel="stylesheet" href="css/foo.css">
</head>
```

HTML 안에 CSS 작성하기

```
<head>
  <style>
    body {font-size: 14px;}
  </style>
</head>
```

인라인 스타일

```
<body style="font-size: 14px;">
```

CSS 파일 안에 CSS 포함하기

```
@import "foo.css";
```

2. CSS (계속)

에릭 마이어의 reset CSS <https://meyerweb.com/eric/tools/css/reset/>
Normalize.css <https://necolas.github.io/normalize.css/>

각 브라우저들에 자체적으로 적용되어 있는 기본 스타일(여백 등)을 초기화 시켜서 원하는 디자인을 적용하기 쉽게 만들어 줍니다. 위의 두가지 방법을 일반적으로 많이 사용하며, 프로젝트에 맞게 직접 만들어 쓰기도 합니다.

7. CSS의 기본 형태

```
선택자  
body {  
    background-color: red;  
}
```

▲ body의 background-color를 red로 설정

```
선택자  
header a {  
    display: inline-block;  
}
```

▲ header에 포함되어 있는 a의 display를 inline-block으로 설정

2. CSS (계속)

아래의 내용 외에도 많은 선택자들이 있지만, 공부를 처음 시작할 때 우선 알면 좋은 것들 위주로 추려보았습니다.

- * 문서 내의 모든 엘리먼트를 선택.

body, header, h1, section 등 해당 이름을 가진 html 엘리먼트를 선택.

#foo id 선택자. `<div id="foo">`의 형태로, id는 중복이 불가능한 유일한 값.

.foo class 선택자. `<div class="foo">`의 형태로, 중복이 가능. class로 자주 쓰는 스타일을 정의해두고 재사용.

input[type="text"] input 엘리먼트 중 type 속성의 값이 text인 엘리먼트를 선택.

.foo > p class="foo"인 엘리먼트의 직계 자식 엘리먼트 중 p를 선택. 문서 내의 모든 엘리먼트를 선택.

.foo > p 설명 예시 code

```
<section class="foo">
  <header>
    <h1>집에서 맛있는 피자 만들기</h1>
    <p>주위에서 쉽게 구할 수 있는 재료를 중심으로</p>
  </header>
  <p>AAAAAA</p> ← .foo > p
</section>
```

.foo:first-child

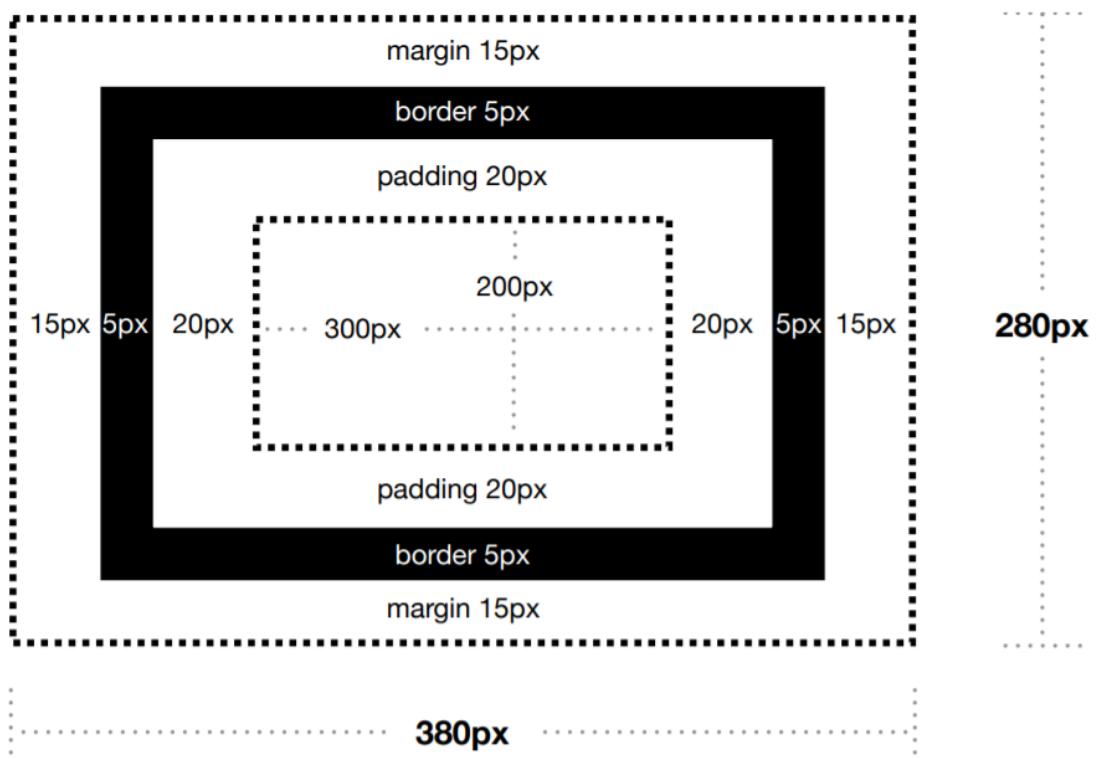
.foo:last-child

.foo:nth-child(n) class="foo"인 엘리먼트 중 첫번째, 마지막, n번째 자식인 엘리먼트를 선택.

.foot:first-child/.foo:last-child/.foo-nth-child(n) 설명 예시 code

```
<ul>
  <li class="foo">1</li> ← .foo:first-child
  <li class="foo">2</li>
  <li class="foo">3</li>      .foo:nth-child(3)
  <li class="foo">4</li> ←
  <li class="foo">5</li>      .foo:last-child
</ul>
```

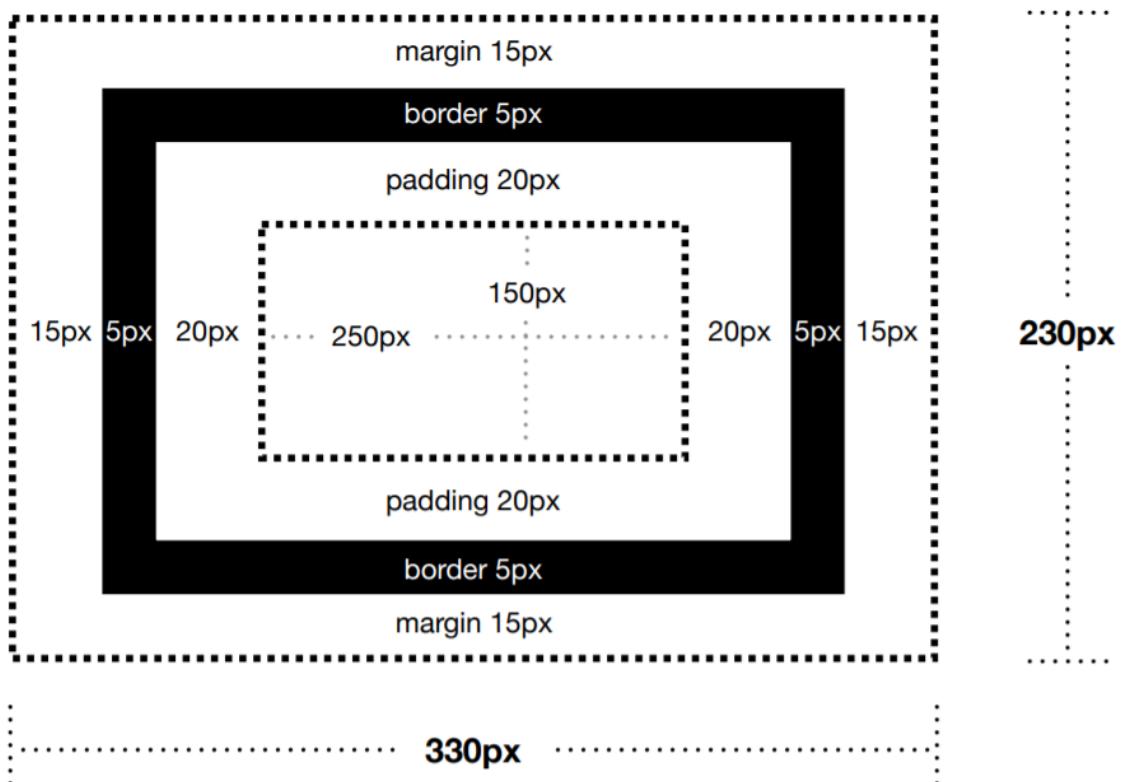
2. CSS (계속)



box-sizing: content-box (기본값)

width: 300px; height: 200px; padding: 20px; border: 5px solid black; margin: 15px;

2. CSS



box-sizing: border-box (border까지 width에 포함)

width: 300px; height: 200px; padding: 20px; border: 5px solid black; margin: 15px;

1일차

저녁

Javascript

TUTORIAL 1 자바스크립트의 기초

이번 시간에 배울 것은 바로

JavaScript 와 jQuery입니다. 우선 JavaScript부터 살펴볼까요?

자바스크립트란?

- HTML을 프로그래밍으로 제어한다
- 웹브라우저가 해석해서 실행할수 있는 유일한 프로그래밍 언어
- 요즘에는 게임프로그래밍, 서버프로그래밍 등 다양한 분야에서 쓰여지고 있음.

실습방법

- goorm IDE를 사용해 소스코드를 작성하고 실행합니다.
- 브라우저는 Chrome을 사용합니다

1. 자바스크립트의 기초

자바스크립트는 사용자의 이벤트를 받고 내장 객체들을 이용한 CSS나 태그의 스타일 관련 속성, 날짜, 텍스트 등을 조작 할 수 있습니다.

〈 J001.html 〉 Code (CSS)

```
<style>
  .box{
    height: 300px;
    width: 300px;
    background-color: blue;
    transition: all 2s;
  }
</style>
```

〈 J001.html 〉 Code (Body)

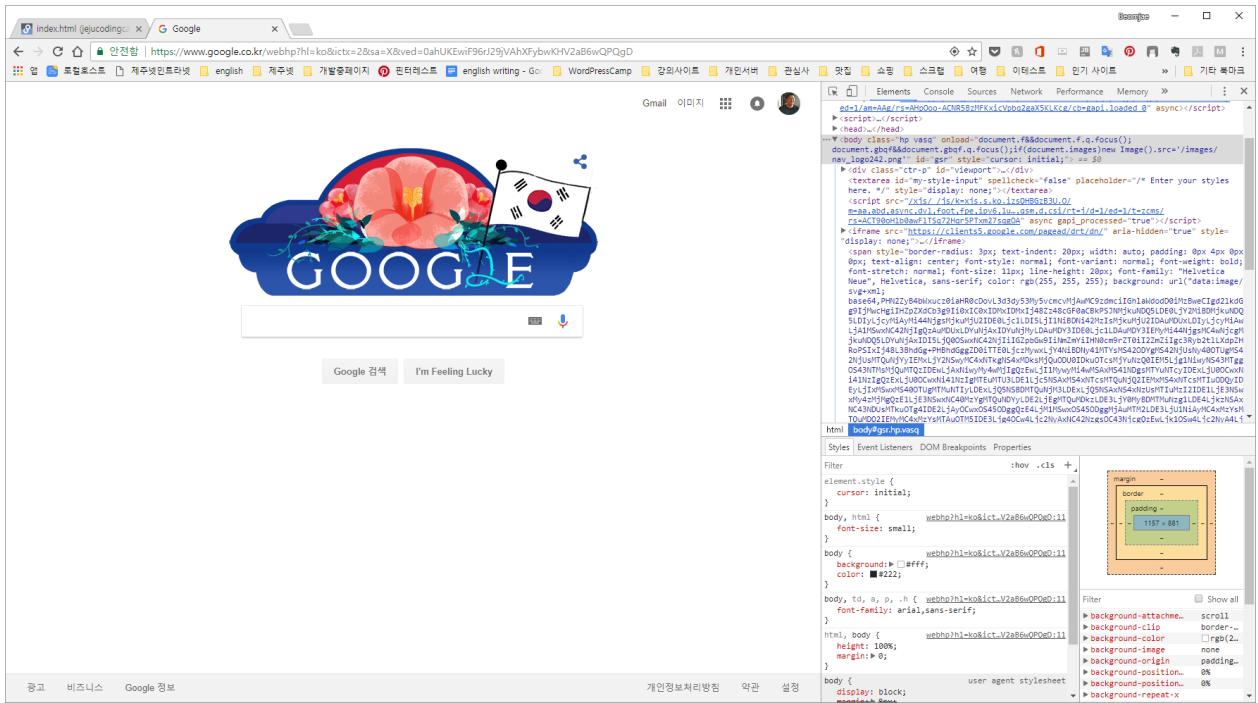
```
<!-- javascript 소개 -->
<h2>자바스크립트로는 어떤 것을 할 수 있나요?</h2>
<button type="button"
  onclick="document.getElementById('demoOne').innerHTML = Date()"
>Click me (One) </button>

<button type="button"
  onclick="document.getElementById('demoTwo').innerHTML = 'hello world'"
>Click me (Two)</button>
<button type="button"
  onclick="document.getElementById('demoThree').style.backgroundColor = 'red'"
>Click me (Three)</button>

<p id="demoOne"></p>
<p id="demoTwo"></p>
<p class="box" id="demoThree"></p>
```

1. 환경설정

단축키 F12 또는 Ctrl + Shift + i.



HTML에서 자바 스크립트 로드하기

INLINE

HTML내에 javascript를 포함하고 로드합니다.

SCRIPT

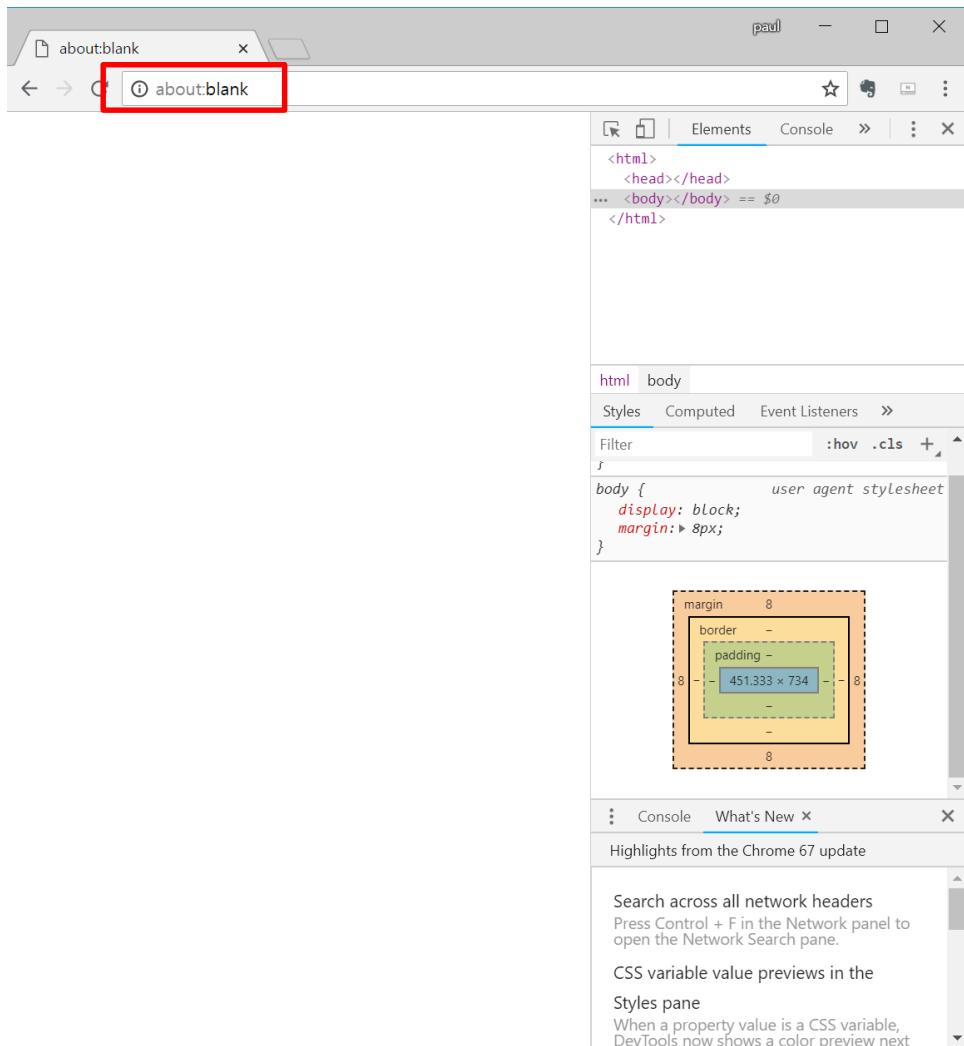
외부 파일로 저장을 하고 로드합니다.

1. 자바스크립트로 출력하기

크롬의 콘솔창은 Ctrl + Shift + i 버튼을 누르면 나오는 창입니다. 다음은 크롬에서

about:blank

페이지에 접속한 화면입니다. 이 페이지는 비어있는 페이지이며 간단한 javascript 연습을 하기 충분합니다.



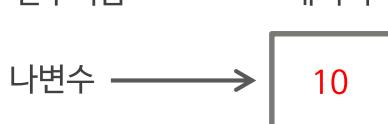
TUTORIAL 2 변수

변수는 ‘변할 수 있는 수’, ’변할 수 있는 정보’라는 뜻입니다.

프로그램을 만드는 데 필요한 숫자나 문자와 같은 데이터를 보관할 공간이 필요한데, 물건을 잠시 보관하는 상자 같은 역할을 하는 것이 변수입니다. 변수는 선언하고 할당하고 사용할 수 있습니다.

또한, 변수는 ‘변할 수 있는 수’이므로 지정된 값을 계속 바꿀 수 있습니다.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>변수</title>
5  </head>          변수이름           데이터
6  <body>          나변수 →
7      <script>
8          var 나변수 = 10
9      </script>
10 </body>
11 </html>
```



2. 변수(계속)

Code

```
var a;  
var myname;
```

자바스크립트에서 변수를 선언할 때에는 앞에 var을 써줍니다. 변수의 타입과 var 외 다른 키워드는 뒤에서 더 자세하게 알아보도록 하겠습니다.

Code

```
a = 1;  
my_name = "lee";
```

변수를 선언하는 방법은 다음과 같습니다.

Code

```
console.log(a);  
console.log(my_name);
```

변수에 저장된 값을 프로그램을 통해서 확인할 수 있습니다. 저희는 구글 chrome을 통해서 콘솔창에 찍히는 값을 확인해 보도록 하겠습니다.

2. 변수(계속)

| Code | Output |
|---|--------|
| <pre>1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>변수</title> 5 </head> 6 <body> 7 <script> 8 var 변수하나 = 10 9 var 변수둘 = 20 10 document.write(변수하나 + 변수둘) 11 </script> 12 </body> 13 </html></pre> | 30 |

우리가 인터넷으로 물건을 주문하면 주문한 물건의 크기에 걸맞는 택배상자에 담아서 오게 됩니다. 이와 같이 변수도 그 크기에 맞는 형식을 가지고 있습니다.

| | |
|----------|-------------------------|
| number | 숫자를 담는 상자 |
| string | 문자열을 담는 상자 |
| boolean | true, false를 담는 상자 |
| array | 여러 자료형을 한꺼번에 담을 수 있는 상자 |
| function | 함수를 담는 상자 |

변수의 자료형은 다양한 데이터를 용도에 맞게 쓰기 위해서 입니다. 보통 언어에서는 변수의 자료형과 함께 변수를 선언하지만 자바스크립트는 자료형을 함께 쓸 필요는 없습니다.

변수의 타입과 var 외 다른 키워드는 뒤에서 더 자세하게 알아보도록 하겠습니다.

2. 변수(계속)

| Code | Output |
|---|------------|
| <pre><!DOCTYPE html> <html> <head> <title>string</title> </head> <body> <script> var 문자열 = 'lee ho jun' document.write(문자열[0] + ' ') document.write(문자열[4] + ' ') document.write(문자열[7] + ' ') </script> </body> </html></pre> | h j |

여기서 문자열 자료형에 대해 좀 더 알아보고 가도록 하겠습니다. 문자열은

작은따옴표 (“ 예시 ”)나

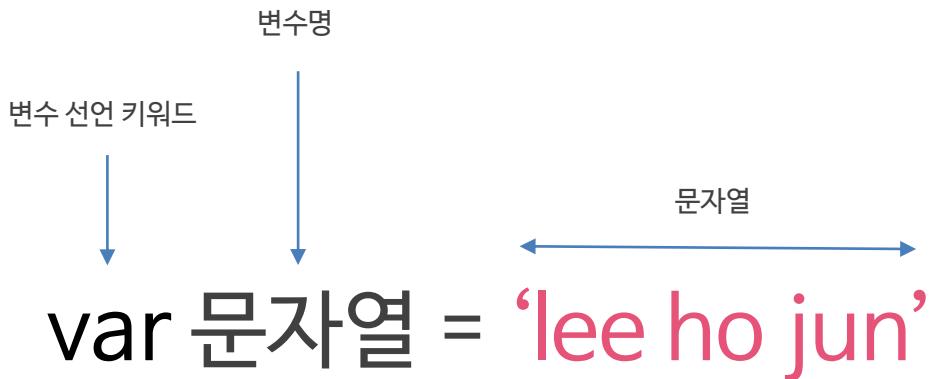
큰따옴표 (“ 예시 ”)로

둘러싼 것을 말합니다.

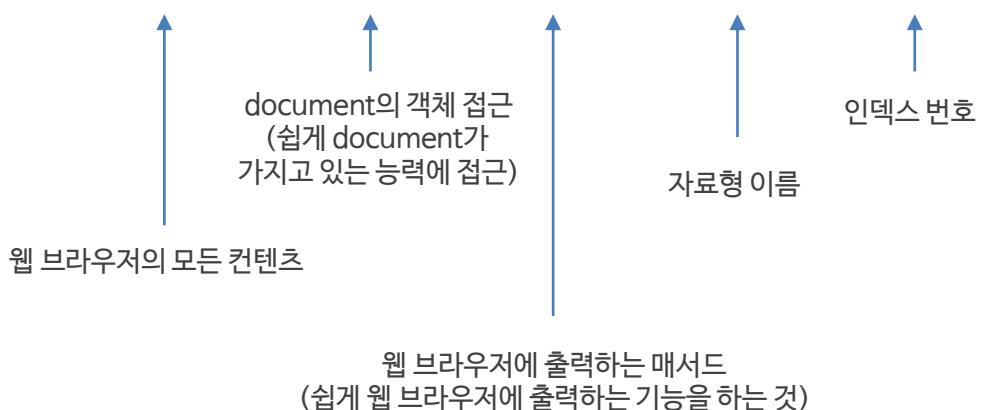
문자열은 순서가 있어요. 순서가 있는 자료형을 시퀀스형 자료형이라고 합니다. 순서는 0부터 시작하며 띄어쓰기도 문자로 취급합니다.

이 순서를 index라고 부르며 위와 같이 index로 호출하는 것을 indexing이라고 합니다.

2. 변수



`document.write(문자열[0])`



TUTORIAL 3 자바스크립트의 연산

자바스크립트는 HTML, CSS와는 다르게 다양한 산술, 대입 등의 연산자를 통해 숫자, 문자 등을 출력할 수 있습니다.

Code

```
<script>
    // Javascript
    var x, y, z;
    x = 5;
    y = 6;
    y++; // y = y + 1, 7
    x--; // x = x - 1, 4
    x--; // x = 3
    x += 10 // x = 13
    y *= 100 // y = 700
    z = x + y; // z = 713
    document.write(z);

    var lastname, firstname;
    lastname = "lee";
    firstname = "hojun";
    document.write(lastname + firstname);
</script>
```

3. 자바스크립트의 연산(계속)

오른쪽에 보이는 것처럼
713leehojun이 출력되었습니다.

x, y, z로 선언된 변수에 초기 값으로 x는 5, y는 6을 대입하였습니다.

Output

713leehojun

왼쪽에 보시는 바와 같이 x는
-- 후위 감소 산술 연산 2회
+= 덧셈 할당 연산 1회를 통해
변수의 값이 13이 되었습니다.

```
<script>
// Javascript
var x, y, z;
x = 5;
y = 6;
y++; // y = y + 1, 7
x--; // x = x - 1, 4
x--; // x = 3
x += 10 // x = 13
y *= 100 // y = 700
z = x + y; // z = 713
document.write(z);

var lastname, firstname;
lastname = "lee";
firstname = "hojun";
document.write(lastname + firstname);
</script>
```

y는 ++ 후위 증가 산술 연산 1회와
*= 곱셈 할당 연산을 통해
변수의 값이 700이 되었습니다.

z는 x(13)와 y(700)를 더한 값으로 713이며, write()함수를 통해 출력합니다.

자바스크립트에서는 피연산자가 하나라도 문자열이면 주로 결합을 수행하게 됩니다.

lastname, firstname으로 선언된 변수에 초기 값으로 문자열 lee와 hojun을 대입하였습니다. write()함수를 통해 결합된 결과값 문자열 leehojun을 출력합니다.

* 피연산자의 type coercion 은 연산자에 따라 달라집니다. 상세 내용은 <https://medium.freecodecamp.org/js-type-coercion-explained-27ba3d9a2839>를 참고해주세요. 예) '1' & 0

3. 자바스크립트의 연산(계속)

| 산술, 할당 연산자 | |
|------------|-----------------------|
| 변수++ | 변수 값에 1 더하기(후위) |
| ++변수 | 변수 값에 1 더하기(전위) |
| 변수-- | 변수 값에 1 빼기(후위) |
| --변수 | 변수 값에 1 빼기(전위) |
| += | 변수 값에 더한 결과값을 대입 |
| -= | 변수 값에 뺀 결과값을 대입 |
| *= | 변수 값에 곱한 결과값을 대입 |
| /= | 변수 값에 나눈 결과값을 대입 |
| %= | 변수 값에 나머지를 구한 결과값을 대입 |

3. 자바스크립트의 연산(계속)

객체의 크고 작고를 비교하거나 맞고 틀림을 비교하는 연산을 비교연산이라고 합니다.

비교 연산 Code

Output

| | | |
|-------------------------------------|--|-------|
| <script> | | |
| var x, y; | | |
| x = 10; | | true |
| y = 5; | | true |
| document.write(x > y); //true | | false |
| document.write(' '); | | false |
| document.write(x >= y); //true | | false |
| document.write(' '); | | false |
| document.write(x <= y); //false | | false |
| document.write(' '); | | false |
| document.write(x < y); //false | | false |
| document.write(' '); | | false |
| document.write(x == 10); //true | | false |
| document.write(' '); | | false |
| document.write(x === "10"); //false | | false |
| document.write(' '); | | false |
| document.write(x != y); //true | | false |
| </script> | | |

3. 자바스크립트의 연산

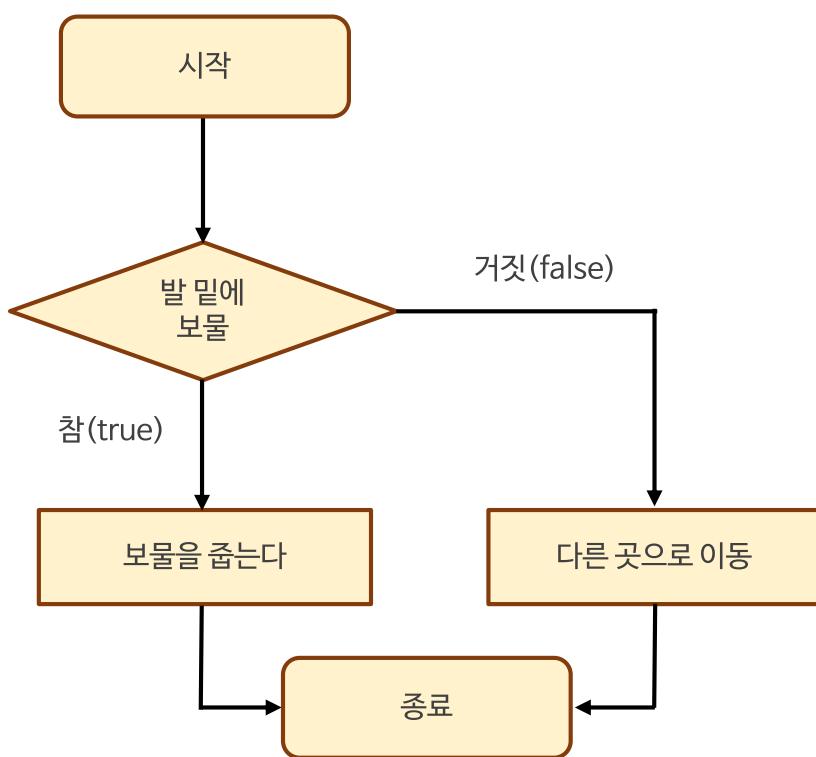
관계연산의 결과는 true, false로 나온다는 것을 기억해 주세요.

| 관계비교연산자 | |
|---------|--------------------------------|
| > | 왼쪽 값이 더 클 경우 true |
| >= | 왼쪽 값이 크거나 같을 경우 true |
| <= | 왼쪽 값이 작거나 같을 경우 true |
| < | 왼쪽 값이 작을 경우 true |
| == | 동일한 값(10=='10'은 true)일 경우 true |
| ===== | 동일한 값과 동일한 유형일 경우 true |
| != | 같지 않은 true |

TUTORIAL 4 조건문

조건이 참인지 거짓인지에 따라 계산이나 상황을 수행하는 프로그래밍 언어의 특징

조건문의 알고리즘



4. 조건문 (계속)

if문 Code

if-else문 Code

if

```
var test = 5;  
✓ if(test < 10) {  
    // codes  
}
```

if

```
var test = 5;  
if(test < 10) {  
    // codes  
} else {  
    // codes  
}
```

배열 Code

```
var arr = [ 'a', 'p', 'p', 'l', 'e' ];
```

1

2

3

4

5

arr

| | | | | |
|---|---|---|---|---|
| a | p | p | l | e |
|---|---|---|---|---|

4. 조건문

다음에는 반복문에 대해서 알아보도록 하겠습니다. 먼저 다음과 같은 배열이 존재한다고 가정해 봅시다.

배열 Code

```
var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];  
  
var text = "";
```

여기서 만약 배열 cars에 담긴 내용을 전부 text라는 변수에 넣고 싶다면 어떻게 해야 할까요? 저희가 지금까지 배운 내용으로는 다음처럼 길고 반복적인 작업을 통해 text에 배열의 내용을 넣을 수 있습니다.

기존 방식 Code

for문 Code

```
text += cars[0] + "<br>";    var i;  
text += cars[1] + "<br>";    for (i = 0; i < cars.length; i++) {  
text += cars[2] + "<br>";        text += cars[i] + "<br>";  
text += cars[3] + "<br>";    }  
text += cars[4] + "<br>";  
text += cars[5] + "<br>";
```

하지만 반복문을 사용하면 오른쪽에 나온 것처럼 간단하게 배열의 내용을 text에 넣을 수 있습니다.

TUTORIAL 5 함수

함수는 입력, 출력, 기능을 하나로 묶어 재사용 할 수 있도록 만드는 것입니다. 자바스크립트는 실행 코드들이 들어있는 **독립 블록 단위의 객체**인 함수를 사용할 수 있습니다. 여기서 객체란 데이터와 그 데이터를 포함한 모든 동작에 대해 말합니다.

함수 예시 코드

Output

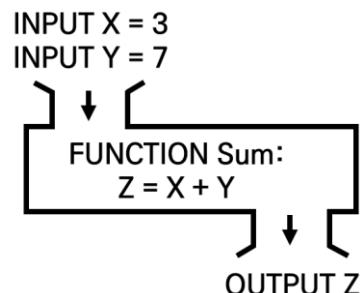
```
<!-- 함수 -->
<p id="result"></p>
<script>
function myFunction(x, y) { //함수의 정의
    z = x + y //함수의 기능
    return z; //함수의 결과값
}
//함수의 호출
document.getElementById("result").innerHTML = myFunction(4, 3);
</script>
```

7

함수는 기본적으로

function() {} :

형태의 구조를 가집니다.



소괄호 ()내에는 매개변수와 중괄호 {}내에는 실행코드와 return(결과) 값을 가지고 있습니다.

함수를 호출 하기 위해서는 선언된 함수의 매개변수 값을 인수로 전달합니다.

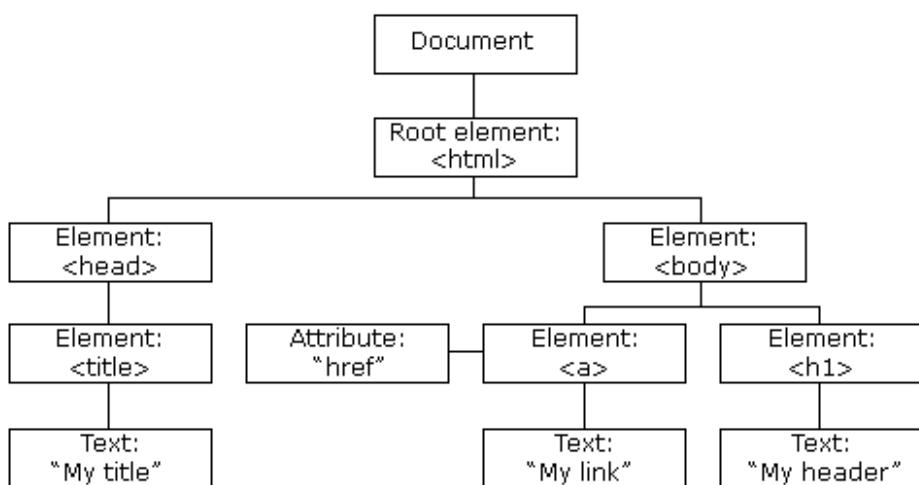
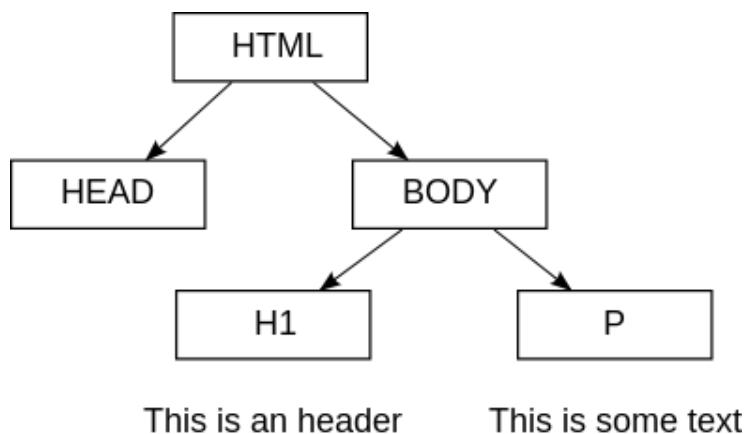
myFunction으로 선언된 함수에 인수로 4, 3을 전달하면 함수내 실행코드의 결과값인 7이 출력 할 수 있습니다.

5. 함수

```
함수명  
함수 선언 키워드  
function myFunction(x, y) { ← 함수 범위 시작  
    z = x + y; ← 함수의 기능  
    return z; ← OUTPUT  
}; ← 함수 범위 종료
```

```
document.write(myFunction(4, 3));  
↑  
웹 브라우저의 모든 컨텐츠  
↑  
document의  
객체 접근  
↑  
웹 브라우저에 출력하는 매서드  
↑  
함수 호출  
↑  
인덱스 번호
```

TUTORIAL 6 DOM(Document Object Model)



웹페이지가 로딩될 때 브라우저는 다음과 같은 객체들의 트리를 생성합니다. 이를 DOM이라 부릅니다. 자바스크립트는 이 DOM을 참조해 html요소를 제어할 수 있습니다.

TUTORIAL 7 HTML FORM

*Mandatory to fill

Firstname: *

Lastname:

Birthday: Day Month Year

Username: *

E-mail:

Website:

Password: *

Re-password: *

I agree to the terms & conditions.

Sign up

모두 인터넷을 하면서 다음과 같은 입력 폼을 본 적이 있을 것입니다. 이러한 입력 폼은 HTML의 <form> 코드로 작성됩니다. 다음 코드를 atom에서 작성해봅시다.

form 예시 Code

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

7. HTML FORM

그런 다음 작성한 html 파일을 브라우저 상에서 확인해 보면 다음과 같은 화면이 보일 것입니다.

First name:

Last name:

도전하기 : 구구단 만들기

Break Time

TUTORIAL 8 라이브러리

라이브러리란 자주 사용하는 코드를 재사용 할 수 있는 형태로 가공하여 프로그래밍 효율을 높여주는 코드들을 말합니다. 자바스크립트의 가장 유명한 라이브러리는 jQuery입니다.



jQuery는 빠르고, 작고, feature-rich한 자바스크립트 라이브러리입니다.

엘리먼트를 선택하는 강력한 방법과 선택된 엘리먼트를 효율적으로 제어할 수 있는 다양한 수단을 제공합니다.

다음은 jQuery에 대해 참고할 수 있는 사이트들입니다.

<http://jquery.com/>
<http://www.jqueryrain.com/>

8. 라이브러리(계속)

jQuery 로딩하기

먼저 구글에서 가장 최신의 jQuery cdn을 찾습니다.

jQuery CDN - Latest Stable Versions

Powered by  StackPath

jQuery Core

Showing the latest stable release in each major branch. [See all versions of jQuery Core.](#)

jQuery 3.x

- jQuery Core 3.2.1 - [uncompressed](#), [minified](#), [slim](#), [slim minified](#)

jQuery 2.x

- jQuery Core 2.2.4 - [uncompressed](#), [minified](#)

jQuery cdn Code

```
<head>
<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
</head>
```

cdn에서 복사해 온 코드를 본래 가지고 있던 html 파일에 붙여 넣습니다.

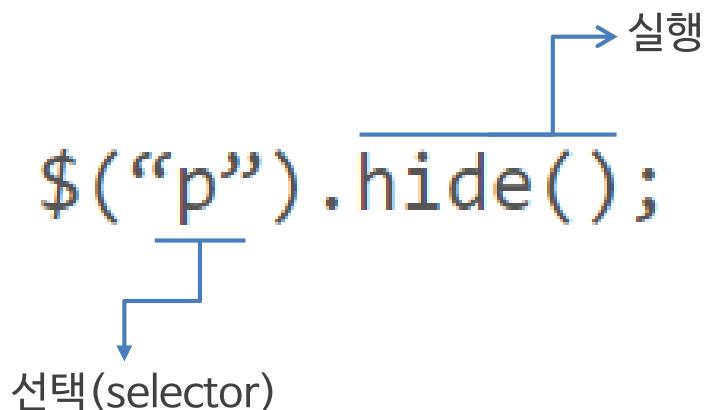
8. 라이브러리

jQuery는 다음과 같이 사용할 수 있습니다.

jQuery test Code

```
$("p").hide();
$("#test").show();
$(".test").click(function(){
    alert("클릭!");
});
```

jQuery는 \$("")의 부분에서 코드가 적용 될 부분을 작성합니다. 그 다음에 .hide(); 와 같이 실행했을 때 적용되는 코드를 넣습니다.



선택 방법은 기본 선택과 계층 선택이 있습니다. 기본 선택에는 element, .class, #d, all(*) 이 있으며, 계층선택에는 child()가 있습니다.

TUTORIAL 9 jQuery의 필터와 실행방법

기본필터

- :eq(index)
- :even
- :odd
- :first
- :last
- :gt(index)
- :lt(index)
- :not(filter)

속성필터

- :attributeContains – input [name*='man']
- attributeEquals – input [name='newsletter']

차일드 필터

- :first-child, :last-child
- :nth-child(2)
- :nth-child(even), :nth-child(odd)
- :nth-child(3n)

다양한 실행방법

컨텐츠 필터

- :contains(text)
- :empty
- :has(selector)

- 속성값 변경
- DOM 탐색
- DOM 편집
- CSS
- 이벤트
- 효과
- AJAX
- 유틸리티

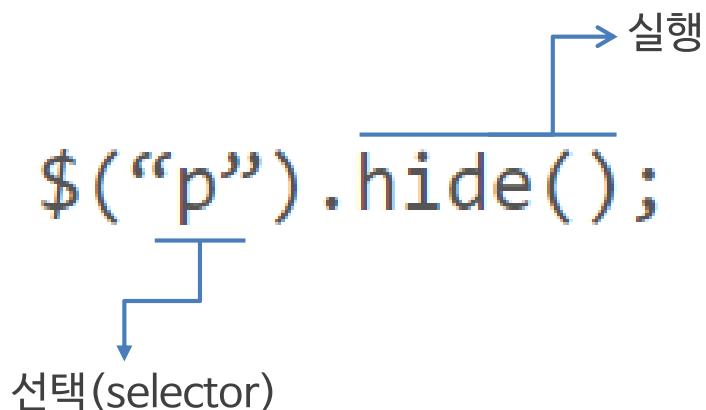
9. jQuery의 필터(계속)

jQuery에서 속성값을 불러오는 방법은 다음과 같습니다.

속성값 Code

```
$(“#name”).val(); #속성값 읽어오기  
$(“#name”).val(); #속성값 쓰기
```

jQuery는 \$(“”)의 부분에서 코드가 적용 될 부분을 작성합니다. 그 다음에 .hide(); 와 같이 실행했을 때 적용되는 코드를 넣습니다.



선택 방법은 기본 선택과 계층 선택이 있습니다. 기본 선택에는 element, .class, #d, all(*) 이 있으며, 계층선택에는 child()가 있습니다.

9. jQuery의 필터(계속)

이 파트에서는 jQuery에서 DOM을 탐색하는 방법에 대해 알아보도록 하겠습니다. DOM의 탐색은 일반적으로 선택자를 통해서 이루어집니다.

탐색 코드 1

```
<div class="wrapper">
  <h1 class="title">TITLE</h1>
  <ul class="items">
    <li class="item">item 1</li>
    <li class="item">item 2</li>
  </ul>
</div>
```

탐색 코드 2

```
$("span").parent();
$("div").children();
$("div").first();
$("div").last();
```

빨간 네모 박스가 쳐진 곳이 선택자에 해당하는 부분입니다.

`$("span").parent();`는 부모 선택자입니다. 이것은 부모 인자로 잡히는 모든 상위 인자를 반환합니다. `$("div").children();`는 자식선택자로, 선택된 요소의 자식에 해당하는 것을 전부 반환합니다.

`$("div").first();`는 선택한 요소 중 가장 첫번째를 반환합니다. 이 예시에서는 div 중 가장 처음 나오는 div를 반환합니다. 반대로 `$("div").last();`는 여러 개의 div 중 가장 마지막에 나오는 div를 반환합니다.

9. jQuery의 필터(계속)

다음으로는 DOM의 편집에 대해 알아보도록 하겠습니다. 먼저 제이쿼리의 문자를 편집하는 메소드에 대해 알아보도록 하겠습니다.

편집 코드 1

```
$("#test1").text("Hello World");
$("#test2").html("<b>Hello World</b>");
$("#test3").val("<b>Hello World</b>");
```

위 세 코드는 텍스트를 받아오거나 변경할 수 있게 합니다. 세 코드는 비슷해 보이지만 차이점이 존재합니다.

| 메소드 | 특징 |
|---------|-----------------------------|
| .text() | 선택한 요소의 텍스트 내용을 받아온다. |
| .html() | 선택한 요소의 html 코드를 설정하거나 받아온다 |
| .val() | form 요소의 값을 설정하거나 받아온다 |

편집 코드 2

```
$( "p" ).append("Some appended text.");
$( "p" ).prepend("Some prepended text.");
$( "#div1" ).remove();
```

append()는 객체를 선택한 객체(p)내의 가장 마지막 요소로 붙입니다. prepend()는 객체를 선택한 객체 내의 가장 첫번째 요소로 붙입니다. remove()는 선택한 객체를 삭제시킵니다.

9. jQuery의 필터(계속)

DOM 편집은 html 요소 뿐만 아니라 css도 변경할 수 있습니다. 아래의 코드를 함께 공부해 보도록 하겠습니다.

css 편집 코드

```
$("div").addClass("important");
$("h1, h2, p").removeClass("blue");
$("h1, h2, p").toggleClass("blue");
$("p").css("background-color");
$("p").css("background-color", "yellow");
```

먼저 addClass() 함수는 클래스를 추가하는 것이 가능합니다. 위 코드에서는 div에 important라는 클래스가 추가되게 됩니다.

.addClass() 적용 전

```
<div class="hello">hello world!</div>
```

예를 들어 위와 같은 코드에 \$("div").addClass("world")라는 코드를 실행하면 다음과 같이 클래스가 변화하게 됩니다.

.addClass() 적용 후

```
<div class="hello world">hello world!</div>
```

| 메소드 | 특징 |
|-----------------|----------------------------|
| .removeClass() | 선택한 클래스를 삭제한다 |
| .toggleClass() | 특정한 클래스의 추가 또는 제거를 한번에 한다 |
| .css("속성", "값") | css의 속성을 선택하고, 값을 변경할 수 있다 |

9. jQuery의 필터

이제부터는, jQuery의 effect에 대해 알아보도록 하겠습니다. effect를 사용하면 마치 파워포인트의 애니메이션과 같이 요소에 다양한 효과를 줄 수 있습니다.

effect Code

```
$("p").hide();
$("p").show();
$("p").toggle();
$("#div1").fadeIn();
$("#div1").fadeOut();
$("#div1").fadeToggle();
$("#div3").fadeIn(3000);
```

| 메소드 | 설명 |
|---------------|---|
| .hide() | 선택한 요소를 사라지게 한다 |
| .show() | 선택한 요소를 나타나게 한다 |
| .toggle() | 선택한 요소의 현재 표시상태에 따라 다르게 나타난다. 만약 선택한 요소가 사라진 상태라면 .show()를 실행하고 아닐 경우 반대를 실행한다. |
| ,fadeIn() | 선택한 요소의 CSS opacity 값을 높여가며 나타난다 |
| .fadeOut() | 선택한 요소의 CSS opacity 값을 낮춰가며 사라진다 |
| .fadeToggle() | 선택한 요소에 fadeIn()과 fadeout()을 번갈아가며 적용한다 |
| .fadeIn(3000) | 시간을 ms단위 또는 slow, fast등으로 넣어 효과의 시간을 조절 할 수 있다. |

TUTORIAL 10 이벤트

이번에는 이벤트에 대해 알아보도록 하겠습니다. 웹 페이지에서 사용자는 버튼을 클릭하거나, 마우스를 스크롤하거나, 필드의 내용을 바꾸는 등의 행동(action)을 합니다. 웹 페이지는 이러한 사용자의 행동에 대해 상호작용을 하여 이벤트를 발생시킵니다.

클릭 이벤트 코드

```
$("p").click(function(){
    $(this).hide();
});
```

현재 작성중인 html 코드에 다음을 적용해 보도록 하겠습니다. 이 코드에서는 <p>태그에서 .click()이라는 이벤트가 발생하면 <p> 태그에 해당하는 내용을 숨기도록 작성되었습니다. 이렇게 특정 요소의 이벤트를 제어하는 함수를 이벤트 핸들러(event handler)라 합니다.

마우스 이벤트 코드

```
$("#p1").mouseenter(function(){
    alert("You entered p1!");
});
$("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
});
```

이번에는 마우스 이벤트를 직접 실습해보면서 위 이벤트 핸들러가 어떤 동작을 발생시키는지 확인해 보도록 합시다.

10. 이벤트

하지만 저희가 지금 당장 모든 이벤트 핸들러에 대해서 학습할 수는 없습니다. 다른 이벤트에 대해서 알고 싶다면 다음 사이트(<https://api.jquery.com/category/events/>)로 들어가 원하는 이벤트에 대해 찾아보면 좋을 것입니다.

The screenshot shows the jQuery API Documentation homepage with a blue header. In the top right corner, there is a heart icon and the text "Your donations help fund the continued development and growth of jQuery." Below the header, there is a yellow button labeled "SUPPORT THE PROJECT". The main navigation bar includes links for "Download", "API Documentation", "Blog", "Plugins", and "Browser Support". A search bar with a magnifying glass icon is located on the right side of the header. The left sidebar contains a navigation tree with categories like "Ajax", "Attributes", "Callbacks Object", "Core", "CSS", "Data", "Deferred Object", "Deprecated", and "Shorthand Methods". The main content area is titled "Category: Events". It features a brief description: "These methods are used to register behaviors to take effect when the user interacts with the browser, and to further manipulate those registered behaviors." Below this, five event handlers are listed in boxes: ".bind()", ".blur()", ".change()", ".click()", and ".contextmenu()". Each box includes a small note about where it is also used: ".bind()" is also in "Event Handler Attachment", ".blur()" is also in "Events > Form Events | Forms", ".change()" is also in "Events > Form Events | Forms", ".click()" is also in "Events > Mouse Events", and ".contextmenu()" is also in "Events > Mouse Events".

혹은 구글에서 원하는 이벤트명과 jQuery를 함께 검색하면 원하는 결과를 찾을 수 있을 것입니다.

jQuery 플러그인

- jQuery UI
- jQuery validation
- jQuery lightbox
- Bootstrap
- etc

2일차

저녁

Bootstrap

TUTORIAL 0 부트스트랩

부트스트랩은 웹 프레임 워크입니다. 프레임 워크는 최소한의 작업으로 빠르게 결과물을 만들 수 있도록 구축해 놓은 종합 공구 세트라고 생각하시면 됩니다.

부트스트랩은 큰 모니터 화면으로 보던지 휴대폰처럼 작은 화면으로 보던지 상관 없이 사용자의 시점에 맞춰 최적화 되어 화면을 구성합니다. 이렇게 사용자의 시점에 맞춰 구축되는 사이트를 반응형 사이트라고 하는데 부트스트랩은 반응형 사이트를 구축하기에 최적화되어 있습니다.

부트스트랩은 form, button, table, navigator가 HTML, CSS로 미리 디자인 되어 있습니다. 부트스트랩 3에서 사용하던 대부분의 태그들이 그대로 살아있지만 많은 부분이 바뀌었기 때문에 새로 공부를 하셔야 합니다.

부트스트랩 4가 나온지 벌써 1년이 다 되었습니다. 그럼에도 아직 시중에는 부트스트랩 3 책만 나와있어 집필을 하게 되었습니다.

<https://www.w3schools.com>에서도 아직 부트스트랩 3를 기술해 놓았습니다. 한국어 공식 홈페이지는 로는 정식 번역된 것이 없으며 <http://bootstrap4.kr/>에서 공식 홈페이지의 한국어 번역이 어느정도 되어 있는 상태(Ver 4.0)입니다. 자세한 내용은 부트스트랩 공식 홈페이지인(영문) <https://getbootstrap.com/docs/4.1/content/reboot/>에서 확인하세요.

TUTORIAL 0.2 부트스트랩 다운로드

<https://getbootstrap.com/>와 <http://bootstrap4.kr/>에서 다운로드 받을 수 있습니다.

- 실습은 다운로드를 받지 않고 진행하니 다운로드를 안받으셔도 됩니다

- 영문 홈페이지 : <https://getbootstrap.com/> (Ver 4.2.1)



Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

[Get started](#)

[Download](#)

Currently v4.1.0



- 한글 홈페이지 : <http://bootstrap4.kr/> (Ver 4.0.0)



부트스트랩

세계에서 가장 널리 사용되는 프론트엔드 컴포넌트 라이브러리를 사용하여 반응형 웹, 모바일 우선 프로젝트를 구축하세요.

부트스트랩은 HTML, CSS, JS로 개발하기 위한 오픈소스 툴킷입니다. 아이디어를 빠르게 모델링하거나, Sass 변수와 mixin, 반응형 그리드 시스템, 그리고 jQuery 기반의 강력한 플러그인을 통해 전체 앱을 신속하게 제작할 수 있습니다.

[시작하기](#)

[다운로드](#)

현재 베포버전 v4.0.0-beta



0.2. 부트스트랩 다운로드 (계속)

한국어 홈페이지에 실려있는 글귀도 함께 실어 놓습니다.

컴파일된 CSS 와 JS

바로 사용할 수 있도록 컴파일된 **부트스트랩 v4.0.0-beta** 을 다운로드하여 쉽게 프로젝트에 추가할 수 있습니다. 여기에는 다음이 포함되어 있습니다. :

- 컴파일 되고 축소 된 CSS 번들 (기본값, 그리드 전용 및 리부트 전용)
- 컴파일 되고 축소 된 JavaScript 플러그인

여기에는 문서, 소스 파일 또는 JavaScript 의존성(jQuery 및 Popper.js)이 포함되지 않습니다.

[다운로드](#)

소스코드 파일

Sass 소스, JavaScript 및 문서 파일을 다운로드하여 자신만의 에셋 파일 라인으로 부트스트랩을 컴파일 하십시오. 이 옵션을 사용하려면 다음과 같은 추가 도구가 필요합니다.

- CSS 컴파일을 위한 Sass 컴파일러 (Libsass 또는 Ruby Sass 지원).
- CSS vendor prefixing을 위한 Autoprefixer

빌드 툴이 필요하다면 부트스트랩과 그 문서를 개발하기 위해 포함되어 있지만, 그 것들은 당신의 목적에 적합하지 않을 것입니다.

[소스 다운로드](#)

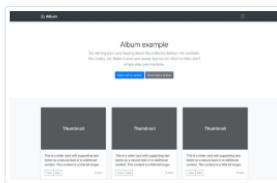
0.2. 부트스트랩 다운로드 (계속)

부트스트랩을 이용한다면 다음과 같은 여러 레이아웃을 만들 수 있습니다. 부트스트랩 공식 홈페이지에서 Examples 탭에서 다양한 레이아웃을 확인해보세요!

유료 템플릿 사이트는 <https://wrapbootstrap.com/>를 참고해보세요.

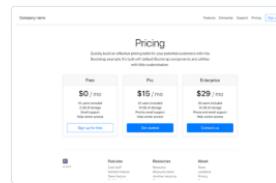
Custom components

Brand new components and templates to help folks quickly get started with Bootstrap and demonstrate best practices for adding onto the framework.



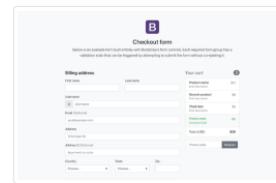
Album

Simple one-page template for photo galleries, portfolios, and more.



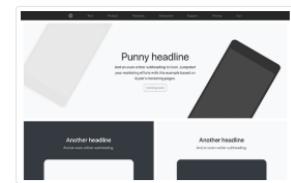
Pricing

Example pricing page built with Cards and featuring a custom header and footer.



Checkout

Custom checkout form showing our form components and their validation features.



Product

Lean product-focused marketing page with extensive grid and image work.



Cover

A one-page template for building simple and beautiful home pages.



Carousel

Customize the navbar and carousel, then add some new components.



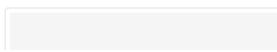
Blog

Magazine like blog template with header, navigation, featured content.



Dashboard

Basic admin dashboard shell with fixed sidebar and navbar.



| ID | 글 제목 | 번호 | 카테고리 | 작가 | 변경일 | 수정일 | 액션 |
|-----|--|----|--------------------|---------------------|------------------|------------------|---|
| 29 | Interview With David Dixon, Artist, Founder, and Curator of Cathouse FUNeral and Its Many Iterations | 1 | critic | Diana Seo Hyung Lee | 2018-02-28 20:59 | | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 43 | 한국과 일본의 미술, 노동자의 삶 | 1 | essay | 이혜령 | 2017-12-22 20:04 | | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 119 | 최현정과 심상학의 이율배반, 아름다움의 진지한 측면 | 2 | critic | 이나연 | 2017-12-08 16:22 | | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 109 | Stairs | 2 | essay | 김윤기(Claire) | 2017-12-08 16:19 | | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 68 | INVESTIGATING THE "GANGNAM STYLE" PHENOMENON: GLOBAL EUROCENTRISM, ASIAN MASCULINITY, AND PSY | 1 | critic | 고민디 | 이나연 | 2017-12-08 16:13 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 151 | Investigating the "Gangnam Style" Phenomenon: Global Eurocentrism, Asian Masculinity, and Psy | 2 | critic | 고민디 | Nayun Lee | 2017-12-08 16:13 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 152 | On <Spectacular Vernacular>: Exhibition of Parsons & Charlesworth, 9/19/2016 - 1/2/2017 (PART II) | 2 | critic | 리안 노연 | 김영주 | 2017-12-08 16:08 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 101 | 월질의 상호이동과 우주의 팽창 | 2 | essay | 이소영 | | 2017-12-08 16:07 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 148 | Safe zone-nowhere | 2 | critic | 이나연 | | 2017-12-08 16:05 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 154 | Life Cultivation | 2 | artist's statement | Sheryl Cheung | | 2017-12-08 15:57 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 146 | Self Portrait-The Crowd 7.8x3" Acrylic, conté, oil pastel on cigarette case 2017 | 2 | artist's statement | Suena | | 2017-12-08 15:36 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |
| 147 | Artist note | 2 | artist's statement | Hwa seo Yang | | 2017-12-08 15:16 | <button>글 조회</button> <button>조회</button> <button>수정</button> <button>삭제</button> |

TUTORIAL 1 컨테이너



```
<div class="container">  
  <!-- Content here -->  
</div>
```

Copy



```
<div class="container-fluid">  
  ...  
</div>
```

Copy

TUTORIAL 2 그리드

처음에는 그리드 시스템에 대해 알아볼 것입니다. 아래 공식 홈페이지 글에도 나와 있는 것처럼 부트스트랩은 전체 화면을 12개의 컬럼으로 나눠놓았습니다. 그 나눠놓은 컬럼에 무엇을 배치할 것인지 우리가 정해서 홈페이지를 만들 수 있습니다. 컬럼 12개가 모여 하나의 row가 됩니다.

Bootstrap grid examples

Basic grid layouts to get you familiar with building within the Bootstrap grid system.

Five grid tiers

There are five tiers to the Bootstrap grid system, one for each range of devices we support. Each tier starts at a minimum viewport size and automatically applies to the larger devices unless overridden.

| | | |
|-----------|-----------|-----------|
| .col-4 | .col-4 | .col-4 |
| .col-sm-4 | .col-sm-4 | .col-sm-4 |
| .col-md-4 | .col-md-4 | .col-md-4 |
| .col-lg-4 | .col-lg-4 | .col-lg-4 |
| .col-xl-4 | .col-xl-4 | .col-xl-4 |

Three equal columns

Get three equal-width columns **starting at desktops and scaling to large desktops**. On mobile devices, tablets and below, the columns will automatically stack.

| | | |
|-----------|-----------|-----------|
| .col-md-4 | .col-md-4 | .col-md-4 |
|-----------|-----------|-----------|

Three unequal columns

Get three columns **starting at desktops and scaling to large desktops** of various widths. Remember, grid columns should add up to twelve for a single horizontal block. More than that, and columns start stacking no matter the viewport.

| | | |
|-----------|-----------|-----------|
| .col-md-3 | .col-md-6 | .col-md-3 |
|-----------|-----------|-----------|

Two columns

Get two columns **starting at desktops and scaling to large desktops**.

| | |
|-----------|-----------|
| .col-md-8 | .col-md-4 |
|-----------|-----------|

Full width, single column

No grid classes are necessary for full-width elements.

2. 그리드 (계속)

앞서 말씀드린 것처럼 실습 캡쳐는 <body>와 </body> 안에 있는 태그들 중 <script>…</script> 를 생략하고 작성하도록 하겠습니다. Code와 같이 작성한다음 002.html로 저장을 하고 실행하면 아래와 같이 나옵니다.
빨간색 네모는 이해를 돋기 위해 작성한 것입니다. 1개의 row에 4 컬럼씩 할당했으므로 같은 넓이로 출력됩니다.

Code

```
14      <div class="container">
15          <div class="row">
16              <div class="col-md-4">
17                  <h1>hello</h1>
18              </div>
19
20              <div class="col-md-4">
21                  <h1>hello</h1>
22              </div>
23
24              <div class="col-md-4">
25                  <h1>hello</h1>
26              </div>
27          </div>
28      </div>
```

Output

hello

4컬럼

hello

4컬럼

hello

4컬럼

002.html

2. 그리드 (계속)

이번에는 한 개의 row를 더 주어 봤습니다. 빨간색 네모는 이해를 돋기 위한 것입니다. 위에는 4컬럼씩 주었기 때문에 3등분이 되었고 아래는 6컬럼씩 주었기 때문에 2등분이 되었습니다.

Code

```
14      <div class="container">          26      <div class="row">
15        <div class="row">            27          <div class="col-md-6">
16          <div class="col-md-4">        28            <h1>hello</h1>
17            <h1>hello</h1>        29          </div>
18          </div>            30          <div class="col-md-6">
19          <div class="col-md-4">        31            <h1>hello</h1>
20            <h1>hello</h1>        32          </div>
21          </div>            33          </div>
22          <div class="col-md-4">        34          </div>
23            <h1>hello</h1>
24          </div>
25      </div>
```

Output



003.html

2. 그리드 (계속)

아래 공식 홈페이지에서 보듯이 하나의 컬럼 단위를 다시 분할하는 것도 가능합니다. 실습은 해보지 않도록 하겠습니다.

Two columns with two nested columns

Per the documentation, nesting is easy—just put a row of columns within an existing column. This gives you two columns **starting at desktops and scaling to large desktops**, with another two (equal widths) within the larger column.

At mobile device sizes, tablets and down, these columns and their nested columns will stack.

| | | |
|-----------|-----------|-----------|
| .col-md-8 | | .col-md-4 |
| .col-md-6 | .col-md-6 | |

Mixed: mobile and desktop

The Bootstrap v4 grid system has five tiers of classes: xs (extra small), sm (small), md (medium), lg (large), and xl (extra large). You can use nearly any combination of these classes to create more dynamic and flexible layouts.

Each tier of classes scales up, meaning if you plan on setting the same widths for xs and sm, you only need to specify xs.

| | | |
|-------------------|------------------|------------------|
| .col-12 .col-md-8 | | .col-6 .col-md-4 |
| .col-6 .col-md-4 | .col-6 .col-md-4 | .col-6 .col-md-4 |
| .col-6 | .col-6 | |

Mixed: mobile, tablet, and desktop

| | | |
|-----------------------------|------------------|------------------|
| .col-12 .col-sm-6 .col-lg-8 | | .col-6 .col-lg-4 |
| .col-6 .col-sm-4 | .col-6 .col-sm-4 | .col-6 .col-sm-4 |

<https://getbootstrap.com/docs/4.1/examples/grid/>

2. 그리드

그리드 시스템에서 12컬럼을 모두 이용할 때 넓이를 100% 사용하고 싶다면 .container-fluid를 사용합니다.

컬럼 사이에 여백은 .row에 .no-gutters를 적용하면 padding과 margin을 제거할 수 있습니다.

공식 홈페이지에 나와있는 그리드 시스템에 대한 설명입니다. Nestable은 컬럼을 또 다른 컬럼으로 나눌 수 있다는 얘기입니다.

우리는 주로 md를 사용합니다.

| | 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 | | | | |

<https://getbootstrap.com/docs/4.1/layout/grid/>

TUTORIAL 3 이미지

이미지에 반응형과 스타일을 입히는 방법입니다. 아래 실행 이미지는 가로가 2000px인 이미지입니다. 3버전에서는 img-responsive를 사용했습니다.

```
<div class="container">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-2">
      hello
    </div>
    <div class="col-2">
      hello
    </div>
  </div>
</div>
```

Output

3_006.html



hello

hello

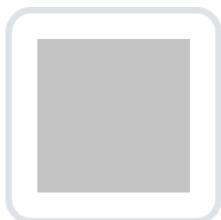
3. 이미지 (계속)

이미지 썸네일입니다. 작은이미지 보다는 큰 이미지에 활용됩니다.

```
<div class="container">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-2">
      hello
    </div>
    <div class="col-2">
      hello
    </div>
  </div>
</div>
```

Output

3_008.html



hello

hello

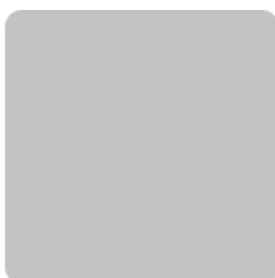
3. 이미지 (계속)

화면에는 보이지 않지만 이미지가 컬럼의 가장 오른쪽에 붙어있습니다. 또한 테두리가 미묘하게 깎여 있는 것을 볼 수 있습니다. 추가로 이미지를 둥그렇게 만드는 class는 rounded-circle입니다. 3 이전의 버전에서는 img-circle을 사용했었습니다.

```
<div class="container">
  <div class="row">
    <div class="col-2">
      
    </div>
    <div class="col-2">
      hello
    </div>
    <div class="col-2">
      hello
    </div>
  </div>
</div>
```

Output

3_009.html



hello

hello

3. 이미지 (계속)

Code

```

```

Output

681x250

Code

```

```

Output

200x200

3. 이미지

Code

```

```

Output

Circle:



TUTORIAL 4 테이블

다음으로는 부트스트랩에서 기본적으로 제공하고 있는 Table에 대하여 알아보도록 하겠습니다. Thead에는 thead-dark나 thead-light를 주어 thead를 부각시킬 수 있습니다. table-striped를 주면 테이블이 중간중간 강조된 격자무늬로 볼 수 있습니다.

Code

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">구분</th>
      <th scope="col">책이름</th>
      <th scope="col">판매량</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>해리포터</td>
      <td>100권</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>해리포터2</td>
      <td>200권</td>
    </tr>
  </tbody>
</table>
```

Output

| 구분 | 책이름 | 판매량 |
|----|-------|------|
| 1 | 해리포터 | 100권 |
| 2 | 해리포터2 | 200권 |

3_010.html

4. 테이블 (계속)

기본 테이블과 마찬가지로 table-striped를 주면 테이블이 중간중간 강조된 격자무늬로 볼 수 있습니다.

Code

```
<table class="table table-dark">
  <thead>
    <tr>
      <th scope="col">구분</th>
      <th scope="col">책이름</th>
      <th scope="col">판매량</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>해리포터</td>
      <td>100권</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>해리포터2</td>
      <td>200권</td>
    </tr>
  </tbody>
</table>
```

Output

| 구분 | 책이름 | 판매량 |
|----|-------|------|
| 1 | 해리포터 | 100권 |
| 2 | 해리포터2 | 200권 |

3_011.html

4. 테이블 (계속)

기본 테이블과 마찬가지로 table-striped를 주면 테이블이 중간중간 강조된 격자무늬로 볼 수 있습니다. table-dark에도 사용할 수 있습니다. table-borderless를 사용하면 테두리를 없앨 수 있습니다.

Code

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th scope="col">구분</th>
      <th scope="col">책이름</th>
      <th scope="col">판매량</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>해리포터</td>
      <td>100권</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>해리포터2</td>
      <td>200권</td>
    </tr>
  </tbody>
</table>
```

Output

| 구분 | 책이름 | 판매량 |
|----|-------|------|
| 1 | 해리포터 | 100권 |
| 2 | 해리포터2 | 200권 |

3_012.html

4. 테이블

Table을 꾸밀 수 있는 전체 색입니다. 필요에 따라 활용하시면 됩니다.

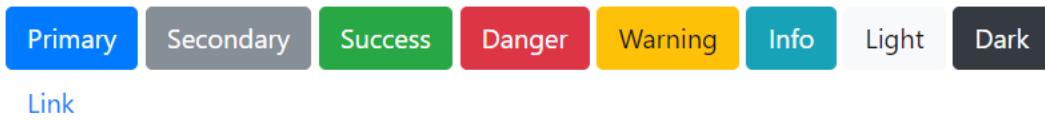
| Code | Output |
|---|--------|
| <table class="table"> | |
| <tbody> | |
| <tr class="table-active"><td>hello</td></tr> | hello |
| <tr class="table-primary"><td>hello</td></tr> | hello |
| <tr class="table-secondary"><td>hello</td></tr> | hello |
| <tr class="table-success"><td>hello</td></tr> | hello |
| <tr class="table-danger"><td>hello</td></tr> | hello |
| <tr class="table-warning"><td>hello</td></tr> | hello |
| <tr class="table-info"><td>hello</td></tr> | hello |
| <tr class="table-light"><td>hello</td></tr> | hello |
| <tr class="table-dark"><td>hello</td></tr> | hello |
| <tr class="bg-primary"><td>hello</td></tr> | hello |
| <tr class="bg-success"><td>hello</td></tr> | hello |
| <tr class="bg-warning"><td>hello</td></tr> | hello |
| <tr class="bg-danger"><td>hello</td></tr> | hello |
| <tr class="bg-info"><td>hello</td></tr> | hello |
| </tbody> | |
| </table> | |

3 013.html

TUTORIAL 5 버튼

부트스트랩에 버튼은 다양한 크기와 색을 지원합니다. 아래는 부트스트랩에서 기본적으로 제공하고 있는 버튼입니다.

Output



Code

```
1 <button type="button" class="btn btn-primary">Primary</button>
2 <button type="button" class="btn btn-secondary">Secondary</button>
3 <button type="button" class="btn btn-success">Success</button>
4 <button type="button" class="btn btn-danger">Danger</button>
5 <button type="button" class="btn btn-warning">Warning</button>
6 <button type="button" class="btn btn-info">Info</button>
7 <button type="button" class="btn btn-light">Light</button>
8 <button type="button" class="btn btn-dark">Dark</button>
9
10 <button type="button" class="btn btn-link">Link</button>
```

4_004_001.html

5. 버튼 (계속)

버튼은 <button> 요소와 함께 사용하도록 설계되었지만 다른 요소에도 사용할 수 있습니다. 아래 다양한 예제를 참고하여 홈페이지를 작성하시면 좋을 것 같습니다. 모두 동일한 모양의 버튼을 만들어줍니다.

Output

Link

Button

Input

Submit

Reset

Code

```
1 <a class="btn btn-primary" href="#" role="button">Link</a>
2 <button class="btn btn-primary" type="submit">Button</button>
3 <input class="btn btn-primary" type="button" value="Input">
4 <input class="btn btn-primary" type="submit" value="Submit">
5 <input class="btn btn-primary" type="reset" value="Reset">
```

4_004_002.html

5. 버튼 (계속)

아웃라인만 들어간 버튼을 만들 수도 있습니다. 기본 클래스를 `.btn-outline-*`로 바꾸면 버튼의 배경 이미지와 색상이 제거된 아웃라인 버튼을 만들 수 있습니다.

Output



Code

```
1 <button type="button" class="btn btn-outline-primary">Primary</button>
2 <button type="button" class="btn btn-outline-secondary">Secondary</button>
3 <button type="button" class="btn btn-outline-success">Success</button>
4 <button type="button" class="btn btn-outline-danger">Danger</button>
5 <button type="button" class="btn btn-outline-warning">Warning</button>
6 <button type="button" class="btn btn-outline-info">Info</button>
7 <button type="button" class="btn btn-outline-light">Light</button>
8 <button type="button" class="btn btn-outline-dark">Dark</button>
```

4_004_003.html

5. 버튼 (계속)

부트스트랩은 버튼의 사이즈를 빠르게 변경하기 위한 클래스를 제공합니다. `.btn-lg` (큰 버튼) 또는 `.btn-sm` (작은 버튼)을 추가하면 아래처럼 버튼의 크기가 조정됩니다.

Output



Code

```
1 <button type="button" class="btn btn-primary btn-lg">큰 버튼</button>
2 <button type="button" class="btn btn-secondary btn-lg">큰 버튼</button>
3
4 <button type="button" class="btn btn-primary btn-sm">작은 버튼</button>
5 <button type="button" class="btn btn-secondary btn-sm">작은 버튼</button>
```

4_004_004.html

5. 버튼 (계속)

.btn-block을 추가하면 부모의 전체 넓이만한 버튼을 만들 수도 있습니다.

Output

블록 레벨 버튼

블록 레벨 버튼

Code

```
1 <button type="button" class="btn btn-primary btn-lg btn-block">
2   블록 레벨 버튼</button>
3 <button type="button" class="btn btn-secondary btn-lg btn-block">
4   블록 레벨 버튼</button>
```

4_004_005.html

5. 버튼 (계속)

부트스트랩 버튼에는 활성과 비활성을 제공하고 있습니다. 버튼이 Acrive(활성)상태가 되면 버튼이 눌러져 보입니다.

Output

Primary link

Link

Code

```
1 <a href="#" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">
2 Primary link</a>
3
4 <a href="#" class="btn btn-secondary btn-lg active" role="button" aria-pressed="true">
5 Link</a>
```

4_004_006.html

5. 버튼 (계속)

버튼의 비활성 상태는 기존의 색 그대로를 나타냅니다. **disabled** 속성을 **<button>** 요소에 추가하여 버튼을 비활성 상태로 만들 수 있습니다. 활성화된 상태는 **.active**입니다.

Output

Primary button

Button

Code

```
1 <button type="button" class="btn btn-lg btn-primary" disabled>
2   Primary button</button>
3 <button type="button" class="btn btn-secondary btn-lg" disabled>
4   Button</button>
```

4_004_007.html

<a> 요소를 사용하는 비활성 버튼은 조금 다르게 동작합니다.

1. **<a>**는 **disabled** 속성을 지원하지 않으므로 비활성화 된 것으로 보이게 하려면 **.disabled** 클래스를 추가 해야 합니다.
2. anchor 버튼의 모든 **pointer-events**를 비활성화 합니다. 해당 속성을 지원하는 브라우저에서는 비활성화 된 커서가 전혀 표시되지 않습니다.
3. Disabled buttons(비활성상태 버튼)은 **aria-disabled="true"** 속성을 포함해야 합니다.
4. 비활성화 된 버튼에는 요소의 상태를 표시하기 위해 **aria-disabled="true"** 속성을 포함합니다.

Output

Primary link

Link

Code

```
1 <a href="#" class="btn btn-primary btn-lg disabled" role="button" aria-disabled="true">
2   Primary link</a>
3 <a href="#" class="btn btn-secondary btn-lg disabled" role="button" aria-disabled="true">
4   Link</a>
```

4_004_008.html

5. 버튼 (계속)

버튼 플러그인을 사용하면 버튼으로 더 많은 작업을 할 수 있게 됩니다. 토글 상태를 알기 위해서는 `data-toggle="button"`을 추가하여 버튼 `active` 상태를 토글합니다. 버튼을 미리 토글할 경우에는 `.active` 클래스와 `aria-pressed="true"`를 수동으로 `<button>`에 추가해야 합니다.

Output

Single toggle

Code

```
1 <button type="button" class="btn btn-primary"
2 data-toggle="button" aria-pressed="false"
3 autocomplete="off">
4 Single toggle
5 </button>
```

4_004_009.html

5. 버튼 (계속)

체크박스와 라디오 버튼에 부트스트랩에서 제공하고 있는 플러그인을 사용해보도록 하겠습니다.
부트스트랩의 `.button` 스타일은 `<label>`과 같은 요소에 적용되어 checkbox나 radio에 버튼토글
을 지정할 수 있습니다. 각각의 스타일에서 토글링할 수 있도록 변경된 버튼이 포함된
`.btn-group`에 `data-toggle="buttons"`를 추가할 수 있습니다.

버튼들의 체크상태는 버튼의 클릭 이벤트를 통해서만 업데이트됩니다. 다른 방법을 사용하여
`<input type="reset">` 또는 input의 `checked` 속성을 적용하여 입력을 업데이트하는 경우
`<label>`을 수동으로 토글해야 합니다.

기본 선택된 버튼은 적용하려면 `.active` 클래스를 input의 `<label>`에 수동으로 추가해야 합니다.

Output

체크박스 1 (기본선택) 체크박스 2 체크박스 3

Code

```
1 <div class="btn-group" data-toggle="buttons">
2   <label class="btn btn-secondary active">
3     <input type="checkbox" checked autocomplete="off"> 체크박스 1 (기본선택)
4   </label>
5   <label class="btn btn-secondary">
6     <input type="checkbox" autocomplete="off"> 체크박스 2
7   </label>
8   <label class="btn btn-secondary">
9     <input type="checkbox" autocomplete="off"> 체크박스 3
10  </label>
11 </div>
```

4_004_010.html

5. 버튼

Button plugin(버튼 플러그인)

Output

라디오 1 (기본선택) 라디오 2 라디오 3

Code

```
1 <div class="btn-group" data-toggle="buttons">
2   <label class="btn btn-secondary active">
3     <input type="radio" name="options" id="option1" autocomplete="off" checked> 라디오 1 (기본선택)
4   </label>
5   <label class="btn btn-secondary">
6     <input type="radio" name="options" id="option2" autocomplete="off"> 라디오 2
7   </label>
8   <label class="btn btn-secondary">
9     <input type="radio" name="options" id="option3" autocomplete="off"> 라디오 3
10  </label>
11 </div>
```

4_004_011.html

Methods(메소드):

`$().button('toggle')`

푸시 상태를 토글할 수 있습니다. 버튼에 활성화된 상태를 지정해보세요.

TUTORIAL 6 CARD

부트스트랩의 Card(카드)는 다양한 컨텐츠를 포함할 수 있는 확장성을 지니고 있습니다. 부트스트랩 버전 3에 익숙하시다면 panel(패널)이나 well(웰), thumbnail(썸브네일)로 대체가 가능합니다. Card(카드)는 Flexbox(플렉스박스)로 제작되어 쉽게 정렬할 수 있으며 다른 부트스트랩 컴포넌트와 잘 섞일 수 있습니다.

Code

```
1 <div class="card" style="width: 20rem;">>
2   
3   <div class="card-body">
4     <h4 class="card-title">Card title</h4>
5     <p class="card-text">
6       Some quick example text to build on the card title
7       and make up the bulk of the card's content.
8     </p>
9     <a href="#" class="btn btn-primary">Go somewhere</a>
10    </div>
11 </div>
```

Output

318x180

Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Go somewhere

4_006_001.html

6. CARD (계속)

카드의 body를 작성하기 위해서는 .card-body 클래스를 사용하면 됩니다. 아래 보이는 것이 기본 골격입니다.

Output

This is some text within a card body.

Code

```
1 <div class="card">
2   <div class="card-body">
3     This is some text within a card body.
4   </div>
5 </div>
```

4_006_002.html

6. CARD (계속)

다음은 공식 홈페이지에서 제공하고 있는 Card에 타이틀, 텍스트 링크 작성 방법입니다.

Titles, text, and links(타이틀, 텍스트, 링크):

- 타이틀은 `<h*`
- 서브타이틀은 `<h*`
- * `.card-body` item에 `.card-title`과 `.card-subtitle`을 배치하면 알맞게 정렬
- 링크는 `<a>` 태그에 `.card-link`를 추가하면 사용할 수 있음

Output

Card title
Card subtitle
Some quick example text to build on the card title and make up the bulk of the card's content.
[Card link](#) [Another link](#)

Code

```
1 <div class="card" style="width: 20rem;">>
2   <div class="card-body">
3     <h4 class="card-title">Card title</h4>
4     <h6 class="card-subtitle mb-2 text-muted">Card subtitle</h6>
5     <p class="card-text">
6       Some quick example text to build on the card title
7       and make up the bulk of the card's content.
8     </p>
9     <a href="#" class="card-link">Card link</a>
10    <a href="#" class="card-link">Another link</a>
11  </div>
12 </div>
```

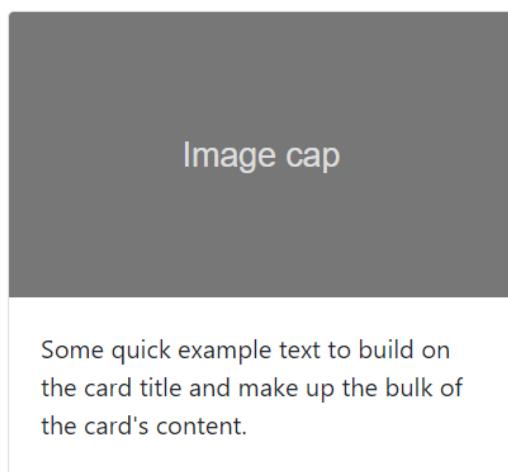
4_006_003.html

6. CARD (계속)

Card안에서 이미지를 사용하는 방법입니다.

- .card-img-top은 이미지를 카드의 윗 부분에 놓아줍니다.
- .card-text를 이용하여 텍스트는 카드에 추가될 수 있습니다.
- .card-text안의 텍스트는 HTML 태그로 꾸밀 수 있습니다.

Output



Code

```
1 <div class="card" style="width: 20rem;">
2   
3   <div class="card-body">
4     <p class="card-text">
5       Some quick example text to build on the card title
6       and make up the bulk of the card's content.
7     </p>
8   </div>
9 </div>
```

4_006_004.html

6. CARD (계속)

Card안에도 리스트 그룹을 만들 수 있습니다. Flush list group(플러시 리스트 그룹)을 이용해서 아래와 같이 카드 안에 컨텐츠 리스트를 만들어 보세요.

Output

1

Cras justo odio

Dapibus ac facilisis in

Vestibulum at eros

2

Featured

Cras justo odio

Dapibus ac facilisis in

Vestibulum at eros

Code

1

```
1 <div class="card" style="width: 20rem;">
2   <ul class="list-group list-group-flush">
3     <li class="list-group-item">Cras justo odio</li>
4     <li class="list-group-item">Dapibus ac facilisis in</li>
5     <li class="list-group-item">Vestibulum at eros</li>
6   </ul>
7 </div>
```

2

```
1 <div class="card" style="width: 18rem;">
2   <div class="card-header">
3     Featured
4   </div>
5   <ul class="list-group list-group-flush">
6     <li class="list-group-item">Cras justo odio</li>
7     <li class="list-group-item">Dapibus ac facilisis in</li>
8     <li class="list-group-item">Vestibulum at eros</li>
9   </ul>
10 </div>
```

6. CARD (계속)

카드를 보다 더 다채롭게 꾸미기 위해 여러 컨텐츠를 믹스해보세요. 아래 예제는 한번 타이핑을 해보시길 추천해 드립니다.

Code

Output

```
1 <div class="card" style="width: 18rem;">>
2   
5   <div class="card-body">
6     <h5 class="card-title">Card title</h5>
7     <p class="card-text">
8       Some quick example text to build on the card title
9       and make up the bulk of the card's content.
10    </p>
11  </div>
12  <ul class="list-group list-group-flush">
13    <li class="list-group-item">Cras justo odio</li>
14    <li class="list-group-item">Dapibus ac facilisis in</li>
15    <li class="list-group-item">Vestibulum at eros</li>
16  </ul>
17  <div class="card-body">
18    <a href="#" class="card-link">Card link</a>
19    <a href="#" class="card-link">Another link</a>
20  </div>
21 </div>
```

Image cap

Card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Cras justo odio

Dapibus ac facilisis in

Vestibulum at eros

Card link Another link

4_006_006.html

6. CARD (계속)

카드 안에 헤더와 푸터를 추가할 수 있습니다.

Output

Featured

Special title treatment

With supporting text below as a natural lead-in to additional content.

[Go somewhere](#)

Code

```
1 <div class="card">
2   <div class="card-header">
3     Featured
4   </div>
5   <div class="card-body">
6     <h5 class="card-title">
7       Special title treatment
8     </h5>
9     <p class="card-text">
10       With supporting text below
11       as a natural lead-in to additional content.
12     </p>
13     <a href="#" class="btn btn-primary">
14       Go somewhere
15     </a>
16   </div>
17 </div> 4_006_007.html
```

6. CARD (계속)

Card header(카드 헤더)는 `<h*`

요소에 `.card-header`를 추가하여 스타일링 할 수 있습
니다.

Output

Featured

Special title treatment

With supporting text below as a natural lead-in to additional content.

[Go somewhere](#)

Code

```
1 <div class="card">
2   <h5 class="card-header">
3     Featured
4   </h5>
5   <div class="card-body">
6     <h5 class="card-title">
7       Special title treatment
8     </h5>
9     <p class="card-text">
10       With supporting text below
11       as a natural lead-in to additional content.
12     </p>
13     <a href="#" class="btn btn-primary">
14       Go somewhere
15     </a>
16   </div>
17 </div>
```

4_006_008.html

6. CARD (계속)

Header and footer(헤더와 풋터) (예제 계속)

Output

1

Quote

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

 — Someone famous in *Source Title*

2

Featured

Special title treatment

With supporting text below as a natural lead-in to additional content.

Go somewhere

2 days ago

Code

```
1 <div class="card">
2   <div class="card-header">
3     Quote
4   </div>
5   <div class="card-body">
6     <blockquote class="blockquote mb-0">
7       <p>
8         Lorem ipsum dolor sit amet,
9         consectetur adipiscing elit.
10        Integer posuere erat a ante.
11      </p>
12      <footer class="blockquote-footer">
13        Someone famous in
14        <cite title="Source Title">
15          Source Title</cite>
16      </footer>
17    </blockquote>
18  </div>
19 </div>
```

```
2 <div class="card text-center">
3   <div class="card-header">
4     Featured
5   </div>
6   <div class="card-body">
7     <h5 class="card-title">
8       Special title treatment
9     </h5>
10    <p class="card-text">
11      With supporting text below
12      as a natural lead-in
13      to additional content.
14    </p>
15    <a href="#" class="btn btn-primary">
16      Go somewhere
17    </a>
18  <div class="card-footer text-muted">
19    2 days ago
20  </div>
21 </div>
```

6. CARD (계속)

Sizing(크기 조정)

2. Using utilities(유틸리티 사용하기):

빠르게 카드의 너비를 맞출 수 있는 sizing utilities를 사용해 보세요.

Output

1

Card title

With supporting text below as a natural lead-in to additional content.

Button

2

Card title

With supporting text below as a natural lead-in to additional content.

Button

Code

1

```
1 <div class="card w-75">
2   <div class="card-body">
3     <h4 class="card-title">
4       Card title
5     </h4>
6     <p class="card-text">
7       With supporting text below
8       as a natural lead-in
9       to additional content.
10    </p>
11    <a href="#" class="btn btn-primary">
12      Button
13    </a>
14  </div>
15 </div>
```

2

```
16 <div class="card w-50">
17   <div class="card-body">
18     <h4 class="card-title">
19       Card title
20     </h4>
21     <p class="card-text">
22       With supporting text below
23       as a natural lead-in
24       to additional content.
25     </p>
26     <a href="#" class="btn btn-primary">
27       Button
28     </a>
29   </div>
30 </div>
31 </div>
```

6. CARD (계속)

Sizing(크기 조정)

3. Using custom CSS(사용자 정의 CSS 사용하기):

너비를 맞추기 위해 stylesheets(스타일시트)나 inline style(인라인 스타일)을 통해 사용자 정의 CSS를 사용해 보세요.

Output

Special title treatment

With supporting text below as a natural lead-in to additional content.

Go somewhere

Code

```
1 <div class="card" style="width: 20rem;">
2   <div class="card-body">
3     <h4 class="card-title">
4       Special title treatment
5     </h4>
6     <p class="card-text">
7       With supporting text below
8       as a natural lead-in
9       to additional content.
10    </p>
11    <a href="#" class="btn btn-primary">
12      Go somewhere
13    </a>
14  </div>
15</div> 4_006_011.html
```

6. CARD (계속)

Text alignment(텍스트 정렬):

Text align classes를 통해 전체 혹은 일부분에서 텍스트 정렬을 빠르게 바꿀 수 있습니다.

| Code | Output |
|--|---|
| <pre>1 <div class="card" style="width: 20rem;"> 2 <div class="card-body"> 3 <h4 class="card-title"> 4 Special title treatment 5 </h4> 6 <p class="card-text"> 7 With supporting text below 8 as a natural lead-in 9 to additional content. 10 </p> 11 12 Go somewhere 13 14 </div> 15 </div> 16 17 <div class="card text-center" style="width: 20rem;"> 18 <div class="card-body"> 19 <h4 class="card-title"> 20 Special title treatment 21 </h4> 22 <p class="card-text"> 23 With supporting text below 24 as a natural lead-in 25 to additional content. 26 </p> 27 28 Go somewhere 29 30 </div> 31 </div> 32 33 <div class="card text-right" style="width: 20rem;"> 34 <div class="card-body"> 35 <h4 class="card-title"> 36 Special title treatment 37 </h4> 38 <p class="card-text"> 39 With supporting text below 40 as a natural lead-in 41 to additional content. 42 </p> 43 44 Go somewhere 45 46 </div> 47 </div></pre> | <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"><p>Special title treatment</p><p>With supporting text below as a natural lead-in to additional content.</p><p>Go somewhere</p></div> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"><p>Special title treatment</p><p>With supporting text below as a natural lead-in to additional content.</p><p>Go somewhere</p></div> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"><p>Special title treatment</p><p>With supporting text below as a natural lead-in to additional content.</p><p>Go somewhere</p></div> |

4_006_012.html

6. CARD (계속)

부트스트랩은 카드 내 네비게이션을 허용하고 있습니다. [nav components](#)를 이용해서 카드의 header(헤더)나 block(블록)에 네비게이션을 추가해보세요. (예제 계속)

Output

The screenshot shows a card component with a green header bar containing the word "Output". Below it is a navigation bar with three tabs: "Active" (selected), "Link", and "Disabled". The main content area contains the heading "Special title treatment" and the text "With supporting text below as a natural lead-in to additional content.". A blue button labeled "Go somewhere" is at the bottom.

Code

```
1 <div class="card text-center">
2   <div class="card-header">
3     <ul class="nav nav-tabs card-header-tabs">
4       <li class="nav-item">
5         <a class="nav-link active" href="#">
6           Active
7         </a>
8       </li>
9       <li class="nav-item">
10        <a class="nav-link" href="#">
11          Link
12        </a>
13      </li>
14      <li class="nav-item">
15        <a class="nav-link disabled" href="#">
16          Disabled
17        </a>
18      </li>
19    </ul>
20  </div>
```

```
21   <div class="card-body">
22     <h5 class="card-title">
23       Special title treatment
24     </h5>
25     <p class="card-text">
26       With supporting text below
27       as a natural lead-in
28       to additional content.
29     </p>
30     <a href="#" class="btn btn-primary">
31       Go somewhere
32     </a>
33   </div>
34 </div>
```

4_006_013.html

6. CARD (계속)

Output

Active Link Disabled

Special title treatment

With supporting text below as a natural lead-in to additional content.

Go somewhere

Code

```
1 <div class="card text-center">
2   <div class="card-header">
3     <ul class="nav nav-pills card-header-pills"> 23
4       <li class="nav-item"> 24
5         <a class="nav-link active" href="#"> 25
6           Active 26
7         </a> 27
8       </li> 28
9       <li class="nav-item"> 29
10        <a class="nav-link" href="#"> 30
11          Link 31
12        </a> 32
13      </li> 33
14      <li class="nav-item"> 34
15        <a class="nav-link disabled" href="#">
16          Disabled
17        </a>
18      </li>
19    </ul>
20  </div>
```

```
21   <div class="card-body">
22     <h5 class="card-title">
23       Special title treatment
24     </h5>
25     <p class="card-text">
26       With supporting text below
27       as a natural lead-in
28       to additional content.
29     </p>
30     <a href="#" class="btn btn-primary">
31       Go somewhere
32     </a>
33   </div>
34 </div>
```

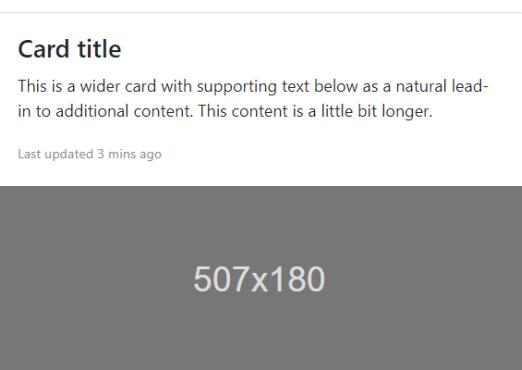
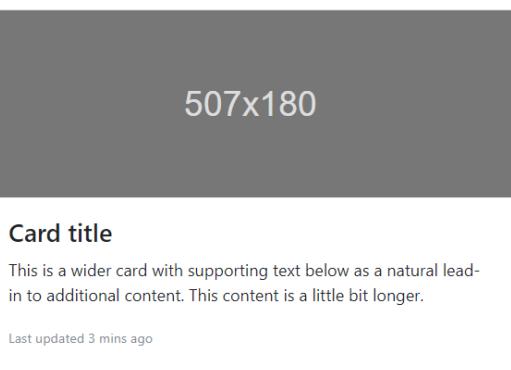
4_006_014.html

6. CARD (계속)

부트스트랩에는 카드에 이미지 작업을 위한 몇가지 옵션이 있습니다. 예를 들어 카드의 양 쪽 끝에 “image caps”를 추가하거나 카드 내용으로 이미지를 겹치거나 이미지를 카드에 간단하게 끼워 넣기를 할 수 있습니다.

Header(헤더), footer(풋터)와 마찬가지로 card(카드)는 상단 및 하단에 “image caps”이 포함될 수 있습니다.

Output



Code

```
1 <div class="card mb-3">
2   22
3   <div class="card-body">
4     <h4 class="card-title">
5       Card title
6     </h4>
7     <p class="card-text">
8       This is a wider card
9       with supporting text
10      below as a natural lead-in
11      to additional content.
12      This content is a little bit longer.
13    </p>
14    <p class="card-text">
15      <small class="text-muted">
16        Last updated 3 mins ago<
17      /small>
18    </p>
19  </div>
20 </div>
21 <div class="card">
22   <div class="card-body">
23     <h4 class="card-title">
24       Card title
25     </h4>
26     <p class="card-text">
27       This is a wider card with supporting text
28       below as a natural lead-in
29       to additional content.
30       This content is a little bit longer.
31     </p>
32     <p class="card-text">
33       <small class="text-muted">
34         Last updated 3 mins ago
35       /small>
36     </p>
37   </div>
38   
39 </div>
```

6. CARD (계속)

부트스트랩에서는 이미지를 카드 배경으로 바꾸고 카드의 텍스트를 오버레이할 수 있는 방법을 제공하고 있습니다.

Code

```
1 <div class="card bg-dark text-white">
2   
5   <div class="card-img-overlay">
6     <h5 class="card-title">
7       Card title
8     </h5>
9     <p class="card-text">
10       This is a wider card
11       with supporting text
12       below as a natural lead-in
13       to additional content.
14       This content is a little bit longer.
15     </p>
16     <p class="card-text">
17       Last updated 3 mins ago
18     </p>
19   </div>
20 </div>
```

Output

Card title

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

444x270

Last updated 3 mins ago

4_006_016.html

6. CARD (계속)

카드는 배경, 테두리, 색깔을 바꿀 수 있습니다. 아래 내용을 참고하세요.

Code

```
1 <div class="card text-white bg-primary mb-3" style="max-width: 18rem;">
2   <div class="card-header">Header</div>
3   <div class="card-body">
4     <h5 class="card-title">Primary card title</h5>
5     <p class="card-text">Some quick example text
6       to build on the card title
7       and make up the bulk of the card's content.</p>
8   </div>
9 </div>
10 <div class="card text-white bg-secondary mb-3" style="max-width: 18rem;">
11   <div class="card-header">Header</div>
12   <div class="card-body">
13     <h5 class="card-title">Secondary card title</h5>
14     <p class="card-text">Some quick example text
15       to build on the card title
16       and make up the bulk of the card's content.</p>
17   </div>
18 </div>
19 <div class="card text-white bg-success mb-3" style="max-width: 18rem;">
20   <div class="card-header">Header</div>
21   <div class="card-body">
22     <h5 class="card-title">Success card title</h5>
23     <p class="card-text">Some quick example text
24       to build on the card title
25       and make up the bulk of the card's content.</p>
26   </div>
27 </div>
28 <div class="card text-white bg-danger mb-3" style="max-width: 18rem;">
29   <div class="card-header">Header</div>
30   <div class="card-body">
31     <h5 class="card-title">Danger card title</h5>
32     <p class="card-text">
33       Some quick example text to build on the card title
34       and make up the bulk of the card's content.</p>
35   </div>
36 </div>
```

Output

Header

Primary card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Secondary card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Success card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Danger card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

6. CARD (계속)

Code

```
1 <div class="card text-white bg-warning mb-3" style="max-width: 18rem;">
2   <div class="card-header">Header</div>
3   <div class="card-body">
4     <h5 class="card-title">Warning card title</h5>
5     <p class="card-text">
6       Some quick example text
7       to build on the card title
8       and make up the bulk of the card's content.</p>
9   </div>
10 </div>
11 <div class="card text-white bg-info mb-3" style="max-width: 18rem;">
12   <div class="card-header">Header</div>
13   <div class="card-body">
14     <h5 class="card-title">Info card title</h5>
15     <p class="card-text">
16       Some quick example text
17       to build on the card title
18       and make up the bulk of the card's content.</p>
19   </div>
20 </div>
21 <div class="card bg-light mb-3" style="max-width: 18rem;">
22   <div class="card-header">Header</div>
23   <div class="card-body">
24     <h5 class="card-title">Light card title</h5>
25     <p class="card-text">
26       Some quick example text
27       to build on the card title
28       and make up the bulk of the card's content.</p>
29   </div>
30 </div>
31 <div class="card text-white bg-dark mb-3" style="max-width: 18rem;">
32   <div class="card-header">Header</div>
33   <div class="card-body">
34     <h5 class="card-title">Dark card title</h5>
35     <p class="card-text">
36       Some quick example text
37       to build on the card title
38       and make up the bulk of the card's content.</p>
39   </div>
40 </div>
```

Output

Header

Warning card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Info card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Light card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

Header

Dark card title

Some quick example text to build
on the card title and make up the
bulk of the card's content.

4_006_018.html

6. CARD (계속)

카드에도 테두리를 줄 수 있습니다. `border-color`를 바꾸기 위해서는 `border utilities`를 사용하면 됩니다. 부모요소의 카드나 카드 컨텐츠에 `.text-{color}` 클래스를 추가하면 됩니다.

| Code | Output |
|--|--|
| <pre>1 <div class="card border-primary mb-3" style="max-width: 18rem;"> 2 <div class="card-header">Header</div> 3 <div class="card-body text-primary"> 4 <h5 class="card-title">Primary card title</h5> 5 <p class="card-text"> 6 Some quick example text 7 to build on the card title 8 and make up the bulk of the card's content. 9 </p> 10 </div> 11 </div> 12 <div class="card border-secondary mb-3" style="max-width: 18rem;"> 13 <div class="card-header">Header</div> 14 <div class="card-body text-secondary"> 15 <h5 class="card-title">Secondary card title</h5> 16 <p class="card-text"> 17 Some quick example text 18 to build on the card title 19 and make up the bulk of the card's content. 20 </p> 21 </div> 22 </div> 23 <div class="card border-success mb-3" style="max-width: 18rem;"> 24 <div class="card-header">Header</div> 25 <div class="card-body text-success"> 26 <h5 class="card-title">Success card title</h5> 27 <p class="card-text"> 28 Some quick example text 29 to build on the card title 30 and make up the bulk of the card's content. 31 </p> 32 </div> 33 </div> 34 <div class="card border-danger mb-3" style="max-width: 18rem;"> 35 <div class="card-header">Header</div> 36 <div class="card-body text-danger"> 37 <h5 class="card-title">Danger card title</h5> 38 <p class="card-text"> 39 Some quick example text 40 to build on the card title 41 and make up the bulk of the card's content. 42 </p> 43 </div> 44 </div></pre> | <div style="border: 1px solid #ccc; padding: 10px;"><p>Header</p><p>Primary card title</p><p>Some quick example text to build on the card title and make up the bulk of the card's content.</p></div> <div style="border: 1px solid #ccc; padding: 10px;"><p>Header</p><p>Secondary card title</p><p>Some quick example text to build on the card title and make up the bulk of the card's content.</p></div> <div style="border: 1px solid #ccc; padding: 10px;"><p>Header</p><p>Success card title</p><p>Some quick example text to build on the card title and make up the bulk of the card's content.</p></div> <div style="border: 1px solid #ccc; padding: 10px;"><p>Header</p><p>Danger card title</p><p>Some quick example text to build on the card title and make up the bulk of the card's content.</p></div> |

6. CARD (계속)

카드 헤더와 풋터 테두리를 바꾸거나 지우면 좀 더 세련된 카드를 구현할 수 있습니다. `.bg-transparent`를 이용해서 `background-color`를 제거할 수도 있습니다.

Output

The screenshot shows a card with a green header containing the word 'Header'. The main body of the card is white and contains the text 'Success card title' in green and 'Some quick example text to build on the card title and make up the bulk of the card's content.' in grey. The footer is green and contains the word 'Footer'.

Code

```
1 <div class="card border-success mb-3" style="max-width: 20rem;">  
2   <div class="card-header bg-transparent border-success">Header</div>  
3   <div class="card-body text-success">  
4     <h4 class="card-title">Success card title</h4>  
5     <p class="card-text">Some quick example text  
6       to build on the card title  
7       and make up the bulk of the card's content.  
8     </p>  
9   </div>  
10  <div class="card-footer bg-transparent border-success">Footer</div>  
11 </div>
```

4_006_020.html

128

6. CARD (계속)

카드 안에 컨텐츠를 추가하기 위해서 부트스트랩은 카드의 레이아웃작업의 옵션들을 포함하고 있습니다.

그룹의 카드들을 같은 너비와 높이의 불어있는 columns(열)로 바꿔줄 수 있습니다.

카드 그룹은 동일한 사이즈로 만들기 위해 `display: flex;`를 사용할 수 있습니다. 풋터를 사용해서 카드 그룹을 사용할 경우, 컨텐츠는 자동적으로 정렬됩니다. 코드는 다음장에 넣어 두었습니다.

| Output | | |
|---|--|---|
| 194x180 | 194x180 | 194x180 |
| Card title This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer. <small>Last updated 3 mins ago</small> | Card title This card has supporting text below as a natural lead-in to additional content. <small>Last updated 3 mins ago</small> | Card title This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action. <small>Last updated 3 mins ago</small> |

6. CARD (계속)

Card groups(카드 그룹) (예시 계속)

Code

```
1  <div class="card-group">
2    <div class="card">
3      
4      <div class="card-body">
5        <h4 class="card-title">Card title</h4>
6        <p class="card-text">
7          This is a wider card
8          with supporting text
9          below as a natural lead-in
10         to additional content.
11         This content is a little bit longer.
12       </p>
13       <p class="card-text"><small class="text-muted">
14         Last updated 3 mins ago</small>
15       </p>
16     </div>
17   </div>
18   <div class="card">
19     
20     <div class="card-body">
21       <h4 class="card-title">Card title</h4>
22       <p class="card-text">
23         This card has supporting text
24         below as a natural lead-in
25         to additional content.</p>
26       <p class="card-text"><small class="text-muted">
27         Last updated 3 mins ago</small>
28       </p>
29     </div>
30   </div>
31   <div class="card">
32     
33     <div class="card-body">
34       <h4 class="card-title">Card title</h4>
35       <p class="card-text">
36         This is a wider card with supporting text
37         below as a natural lead-in to additional content.
38         This card has even longer content than the first
39         to show that equal height action.</p>
40       <p class="card-text"><small class="text-muted">
41         Last updated 3 mins ago</small>
42       </p>
43     </div>
44   </div>
45 </div>
```

4_006_021.html

6. CARD (계속)

Card groups(카드 그룹) (코드 계속)

| Output | | |
|---|--|---|
| 194x180 | 194x180 | 194x180 |
| Card title This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer. | Card title This card has supporting text below as a natural lead-in to additional content. | Card title This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action. |
| Last updated 3 mins ago | Last updated 3 mins ago | Last updated 3 mins ago |

6. CARD (계속)

Card groups(카드 그룹)

Code

```
1 <div class="card-group">
2   <div class="card">
3     
4     <div class="card-body">
5       <h4 class="card-title">Card title</h4>
6       <p class="card-text">
7         This is a wider card with supporting text
8         below as a natural lead-in to additional content.
9         This content is a little bit longer.
10      </p>
11    </div>
12    <div class="card-footer">
13      <small class="text-muted">Last updated 3 mins ago</small>
14    </div>
15  </div>
16  <div class="card">
17    
18    <div class="card-body">
19      <h4 class="card-title">Card title</h4>
20      <p class="card-text">
21        This card has supporting text
22        below as a natural lead-in
23        to additional content.
24      </p>
25    </div>
26    <div class="card-footer">
27      <small class="text-muted">Last updated 3 mins ago</small>
28    </div>
29  </div>
30  <div class="card">
31    
32    <div class="card-body">
33      <h4 class="card-title">Card title</h4>
34      <p class="card-text">
35        This is a wider card with supporting text
36        below as a natural lead-in to additional content.
37        This card has even longer content than the first
38        to show that equal height action.
39      </p>
40    </div>
41    <div class="card-footer">
42      <small class="text-muted">Last updated 3 mins ago</small>
43    </div>
44  </div>
45 </div>
```

4_006_022.html

6. CARD (계속)

너비와 높이가 동일하지만 붙어있지 않은 것을 카드 데크라고 말합니다. 푸터를 사용해서 카드 그룹을 사용할 경우 컨텐츠는 자동으로 정렬됩니다.

Output

The image shows three cards arranged horizontally under a green header bar labeled "Output". Each card has a dark gray header section labeled "174x200" and a white content section. The first card's content section contains the text: "Card title", "This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.", and "Last updated 3 mins ago". The second card's content section contains the text: "Card title", "This card has supporting text below as a natural lead-in to additional content.", and "Last updated 3 mins ago". The third card's content section contains the text: "Card title", "This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.", and "Last updated 3 mins ago".

| Card title | Card title | Card title |
|--|--|---|
| This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer. Last updated 3 mins ago | This card has supporting text below as a natural lead-in to additional content. Last updated 3 mins ago | This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action. Last updated 3 mins ago |

6. CARD (계속)

Card decks(카드 데크) (예제 계속)

Code

```
1 <div class="card-deck">
2   <div class="card">
3     
4     <div class="card-body">
5       <h4 class="card-title">Card title</h4>
6       <p class="card-text">
7         This is a longer card with supporting text
8         below as a natural lead-in to additional content.
9         This content is a little bit longer.
10      </p>
11      <p class="card-text"><small class="text-muted">
12        Last updated 3 mins ago</small>
13      </p>
14    </div>
15  </div>
16  <div class="card">
17    
18    <div class="card-body">
19      <h4 class="card-title">Card title</h4>
20      <p class="card-text">
21        This card has supporting text
22        below as a natural lead-in to additional content.
23      </p>
24      <p class="card-text"><small class="text-muted">
25        Last updated 3 mins ago</small>
26      </p>
27    </div>
28  </div>
29  <div class="card">
30    
31    <div class="card-body">
32      <h4 class="card-title">Card title</h4>
33      <p class="card-text">
34        This is a wider card with supporting text
35        below as a natural lead-in to additional content.
36        This card has even longer content than the first
37        to show that equal height action.
38      </p>
39      <p class="card-text"><small class="text-muted">
40        Last updated 3 mins ago</small>
41      </p>
42    </div>
43  </div>
44 </div>
```

4_006_023.html

6. CARD (계속)

Card decks(카드 데크) (코드 계속)

| Output | | |
|---|--|---|
| 174x180 | 174x180 | 174x180 |
| Card title This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer. | Card title This card has supporting text below as a natural lead-in to additional content. | Card title This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action. |
| Last updated 3 mins ago | Last updated 3 mins ago | Last updated 3 mins ago |

6. CARD (계속)

Card decks(카드 데크)

Code

```
1  <div class="card-deck">
2    <div class="card">
3      
4      <div class="card-body">
5        <h4 class="card-title">Card title</h4>
6        <p class="card-text">
7          This is a wider card with supporting text
8          below as a natural lead-in to additional content.
9          This content is a little bit longer.
10       </p>
11     </div>
12     <div class="card-footer">
13       <small class="text-muted">Last updated 3 mins ago</small>
14     </div>
15   </div>
16   <div class="card">
17     
18     <div class="card-body">
19       <h4 class="card-title">Card title</h4>
20       <p class="card-text">
21         This card has supporting text
22         below as a natural lead-in to additional content.
23       </p>
24     </div>
25     <div class="card-footer">
26       <small class="text-muted">Last updated 3 mins ago</small>
27     </div>
28   </div>
29   <div class="card">
30     
31     <div class="card-body">
32       <h4 class="card-title">Card title</h4>
33       <p class="card-text">
34         This is a wider card with supporting text
35         below as a natural lead-in to additional content.
36         This card has even longer content than the first
37         to show that equal height action.
38       </p>
39     </div>
40     <div class="card-footer">
41       <small class="text-muted">Last updated 3 mins ago</small>
42     </div>
43   </div>
44 </div>
```

6. CARD (계속)

카드는 .card-columns에서 CSS를 이용해 감싸주면 벽돌 형으로 정렬된 열을 구성 할 수 있습니다. 카드는 flexbox(플렉스 박스)가 아닌 CSS 열 속성을 사용하여 쉽게 정렬 할 수 있습니다. 카드는 위에서 아래로, 왼쪽에서 오른쪽으로 정렬됩니다.

카드 열이 있는 마일리지는 다를 수 있습니다. 카드가 열을 가로 지르지 못하게 하려면 `display: inline-block`에 `column-break-inside: avoid`로써 설정해야 합니다.

(코드 계속)

Output

368x160

Card title

This card has supporting text below as a natural lead-in to additional content.

Last updated 3 mins ago

Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat.

— Someone famous in *Source Title*

Card title

This card has supporting text below as a natural lead-in to additional content.

Last updated 3 mins ago

368x260

Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

— Someone famous in *Source Title*

Card title

This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.

Last updated 3 mins ago

6. CARD (계속)

Card columns(카드 칼럼)

Code

```
1 <div class="card-columns">
2   <div class="card">
3     
4     <div class="card-body">
5       <h4 class="card-title">Card title</h4>
6       <p class="card-text">
7         This card has supporting text
8         below as a natural lead-in
9         to additional content.</p>
10      <p class="card-text"><small class="text-muted">
11        Last updated 3 mins ago</small></p>
12    </div>
13  </div>
14  <div class="card bg-primary text-white text-center p-3">
15    <blockquote class="blockquote mb-0">
16      <p>Lorem ipsum dolor sit amet,
17         consectetur adipiscing elit.
18         Integer posuere erat.</p>
19    <footer class="blockquote-footer">
20      <small>
21        Someone famous in <cite title="Source Title">
22          Source Title</cite>
23      </small>
24    </footer>
25  </blockquote>
26 </div>
27 <div class="card text-center">
28   <div class="card-body">
29     <h4 class="card-title">Card title</h4>
30     <p class="card-text">
31       This card has supporting text
32       below as a natural lead-in
33       to additional content.</p>
34     <p class="card-text"><small class="text-muted">
35       Last updated 3 mins ago</small></p>
36   </div>
```

```
37   </div>
38   <div class="card">
39     
40   </div>
41   <div class="card p-3 text-right">
42     <blockquote class="blockquote mb-0">
43       <p>Lorem ipsum dolor sit amet,
44         consectetur adipiscing elit.
45         Integer posuere erat a ante.</p>
46     <footer class="blockquote-footer">
47       <small class="text-muted">
48         Someone famous in <cite title="Source Title">
49           Source Title</cite>
50       </small>
51     </footer>
52   </blockquote>
53 </div>
54 <div class="card">
55   <div class="card-body">
56     <h4 class="card-title">Card title</h4>
57     <p class="card-text">
58       This is a wider card with supporting text
59       below as a natural lead-in to additional content.
60       This card has even longer content than the first
61       to show that equal height action.</p>
62     <p class="card-text"><small class="text-muted">
63       Last updated 3 mins ago</small></p>
64   </div>
65 </div>
66 </div>
```

4_006_025.html

추가적인 코드를 이용해서 카드를 커스터마이징할 수 있습니다.

CSS의 .card-columns를 이용해서 커스터 마이징 해보세요.

```
.card-columns { @include media-breakpoint-only(lg) { column-count: 4; }
@include media-breakpoint-only(xl) { column-count: 5; } }
```

TUTORIAL 7 캐러셀

캐러셀은 이미지나 텍스트 슬라이드를 구성하는 슬라이드 쇼입니다. 마지막 캐러셀 이미지에서는 다시 처음 슬라이드로 돌아오게 됩니다.

부트스트랩 공식 홈페이지에서는 캐러셀을 ‘CSS 3D 트랜스폼과 약간의 자바스크립트로 제작된 시리즈 컨텐츠를 순환하는 슬라이드 쇼’라고 말하고 있습니다. 주의하셔야 할 부분은 캐러셀은 중첩을 허락하지 않으며 웹 접근성 표준을 준수하지 않는다는 것입니다. 또한 소스에서 JS를 빌드하는 경우 [util.js](#)가 필요합니다.

캐러셀은 자동으로 슬라이드 크기를 표준화하지 않습니다. 따라서 컨텐츠의 크기를 적절하게 조정하려면 추가 유ти리티 또는 사용자 정의 스타일을 사용해야 할 수도 있습니다.

.active 클래스 슬라이드당 하나씩 추가되어야 합니다. 그렇지 않으면 캐러셀이 보이지 않습니다. 하나의 페이지에 다양한 캐러셀을 사용할 경우, 선택적인 컨트롤을 위해 .carousel에 각각의 고유한 아이디를 부여해주세요. 컨트롤과 인디케이터 요소들의 경우 반드시 .carousel요소의 아이디를 맞추는 데이터 맞춤 속성을 가져야 합니다.

7. 캐러셀 (계속)

다음은 슬라이드만 있는 캐러셀입니다.

Output

첫번째 슬라이드

두번째 슬라이드

Code

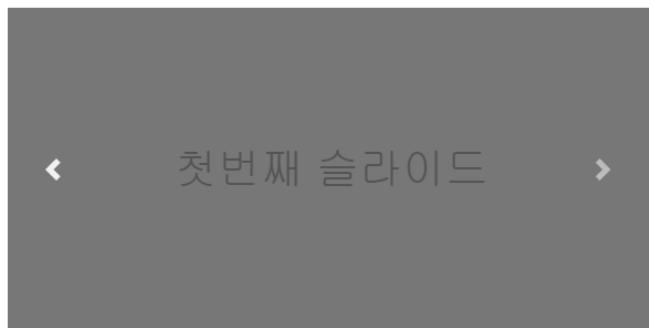
```
1 <div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel">
2   <div class="carousel-inner">
3     <div class="carousel-item active">
4       
5     </div>
6     <div class="carousel-item">
7       
8     </div>
9   </div>
10 </div>
```

4_007_001.html

7. 캐러셀 (계속)

이전과 다음 컨트롤이 필요한 경우 아래와 같이 사용할 수 있습니다.

Output



Code

```
1 <div id="carouselExampleControls"
2   class="carousel slide" data-ride="carousel">
3     <div class="carousel-inner">
4       <div class="carousel-item active">
5         
6       </div>
7     </div>
8     <a class="carousel-control-prev"
9       href="#carouselExampleControls"
10      role="button" data-slide="prev">
11       <span class="carousel-control-prev-icon"
12         aria-hidden="true"></span>
13       <span class="sr-only">이전</span>
14     </a>
15     <a class="carousel-control-next"
16       href="#carouselExampleControls"
17      role="button" data-slide="next">
18       <span class="carousel-control-next-icon"
19         aria-hidden="true"></span>
20       <span class="sr-only">다음</span>
21     </a>
22   </div>
```

4_007_002.html

7. 캐러셀 (계속)

컨트롤과 함께 인디케이터를 캐러셀에 추가할 수 있습니다. 슬라이드 하단에 보이는 것이 인디케이터입니다.

Code

```
<div id="carouselExampleIndicators"
      class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators"
        data-slide-to="0" class="active"></li>
    <li data-target="#carouselExampleIndicators"
        data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators"
        data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev"
     href="#carouselExampleIndicators"
     role="button" data-slide="prev">
    <span class="carousel-control-prev-icon"
          aria-hidden="true"></span>
    <span class="sr-only">이전</span>
  </a>
  <a class="carousel-control-next"
     href="#carouselExampleIndicators"
     role="button" data-slide="next">
    <span class="carousel-control-next-icon"
          aria-hidden="true"></span>
    <span class="sr-only">다음</span>
  </a>
</div>
```

Output

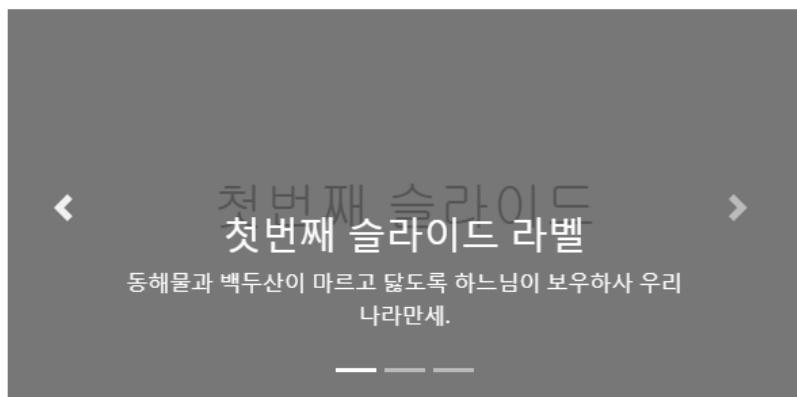


4_007_003.html

7. 캐러셀 (계속)

.carousel-item 내에 .carousel-caption 요소를 사용하여 슬라이드에 캡션을 추가할 수 있습니다. 선택적으로 `display utilites`를 사용하여 작은 뷔포트에 숨길 수 있습니다. 처음에는 `d-none`으로 숨기고 `d-md-block`으로 중각 크기의 디바이스에서 다시 가져옵니다.

Output



Code

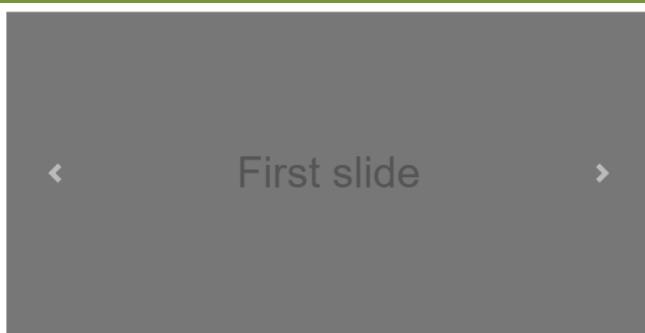
```
1 <div class="carousel-item">
2   
3   <div class="carousel-caption d-none d-md-block">
4     <h3>...</h3>
5     <p>...</p>
6   </div>
7 </div>
```

4_007_004.html

7. 캐러셀 (계속)

부트스트랩에 Crossfade(크로스 페이드)는 이미지가 animate(애니메이트)되는 대신 fadein fadeout되는 fade transition(페이드 트랜지션)을 사용하는 방법입니다. 캐러셀에 `.carousel-fade` 를 추가하면 됩니다.

Output



Code

```
1 <div id="carouselExampleFade"
2   class="carousel slide carousel-fade"
3   data-ride="carousel">
4   <div class="carousel-inner">
5     <div class="carousel-item active">
6       
9     </div>
10   </div>
11   <a class="carousel-control-prev"
12     href="#carouselExampleFade"
13     role="button" data-slide="prev">
14     <span class="carousel-control-prev-icon"
15
16       aria-hidden="true"></span>
17     <span class="sr-only">Previous</span>
18   </a>
19   <a class="carousel-control-next"
20     href="#carouselExampleFade"
21     role="button" data-slide="next">
22     <span class="carousel-control-next-icon"
23       aria-hidden="true"></span>
24     <span class="sr-only">Next</span>
25   </a>
26 </div>
```

4_007_005.html

7. 캐러셀 (계속)

데이터 속성을 사용하여 캐러셀의 위치를 쉽게 제어할 수 있습니다. `data-slide`는 `prev` 또는 `next` 키워드를 사용하여 현재 위치를 기준으로 슬라이드 위치를 변경합니다. 또는 `data-slide-to`를 사용해서 슬라이드 포인터를 0으로 시작하는 특정 인덱스로 이동하는 슬라이드 포인터 `data-slide-to="2"`에 원시 슬라이드 인덱스를 전달합니다.

`data-ride="carousel"` 속성은 캐러셀 페이지 로드시 애니메이션으로 표시하는데 사용됩니다. 동일한 캐러셀을 JavaScript 초기화 (중복 및 불필요한)방식과 함께 사용할 수는 없습니다.

Via JavaScript (JavaScript 방식):

```
$('.carousel').carousel() 를 사용해서 수동으로 캐러셀을 호출할 수 있습니다.
```

7. 캐러셀 (계속)

Options(옵션):

옵션은 data 속성 또는 JavaScript를 통해 전달될 수 있습니다. data 속성의 경우 data-interval="" 처럼 data-에 옵션명을 추가하십시오.

| 옵션명 | 유형 | 기본 | 설명 |
|----------|------------------|---------|---|
| interval | number | 5000 | 아이템 자동 순환 자연시간입니다. false 인 경우, 캐러셀은 자동으로 순환하지 않습니다. |
| keyboard | boolean | true | 캐러셀이 키보드 이벤트에 반응해야 하는지 여부를 나타내는 것입니다. |
| pause | string boolean | "hover" | "hover"로 설정하면 mouseenter에서 캐러셀의 순환을 일시 중지하고 mouseleave에서 캐러셀을 다시 시작합니다. false로 설정하면 캐러셀 위에 마우스를 올리면 일시 중지되지 않습니다. 터치 지원 디바이스에서 "hover"로 설정하면 touchend(사용자가 캐러셀과 상호 작용을 완료하면)에서 두번의 인터벌 동안 순환이 일시 중지됩니다. 이는 위의 마우스 동작에 추가 된 것입니다. |
| ride | string | false | 사용자가 첫번째 아이템을 수동으로 순환시킨 후 캐러셀을 자동 재생합니다. "carousel" 인 경우 로드시 캐러셀을 자동 재생 합니다. |
| wrap | boolean | true | 캐러셀이 지속적으로 순환 해야하는지 아니면 강력하게 정지를 해야하는지를 나타냅니다. |

7. 캐러셀 (계속)

Methods(메소드):

.carousel(options)

선택적 옵션 object를 사용해서 캐러셀을 초기화하고 아이템을 순환시킵니다.

```
$('.carousel').carousel({ interval: 2000 })
```

.carousel('cycle')

왼쪽에서 오른쪽으로 캐러셀 아이템을 순환 시킵니다.

.carousel('pause')

아이템이 순환하는 것을 중지합니다.

.carousel(number)

캐러셀을 특정 프레임(배열과 비슷한 0기반)으로 이동시킵니다. 대상 아이템이 표시되기 전에 (즉, slid.bs.carousel 이벤트가 발생하기 전에) 호출자에게 반환됩니다.

.carousel('prev')

이전 항목으로 이동합니다. 이전 항목이 표시되기 전에(즉, slid.bs.carousel 이벤트가 발생하기 전에) 호출자에게 반환됩니다.

.carousel('next')

다음 항목으로 이동합니다. 다음 항목이 표시되기 전에(즉, slide.bs.carousel 이벤트가 발생하기 전에) 호출자에게 반환됩니다.

7. 캐러셀

Events(이벤트):

부트스트랩의 캐러셀 클래스는 캐러셀 기능에 연결하기 위한 두 가지 이벤트를 노출합니다.

두 이벤트는 모두 다음과 같은 추가 속성을 가집니다.

- **direction** : 캐러셀이 슬라이드 되는 방향 (“left” 또는 “right”)
- **relatedTarget** : 활성화된(active) 아이템으로 제자리에 들어간 DOM요소입니다.
- **from** : 현재 아이템의 색인(index)
- **to** : 다음 아이템의 색인(index)

모든 이벤트는 캐러셀 자체에서 발생됩니다. (즉, `<div class="carousel">`에서)

| 이벤트 | 설명 |
|-------------------|---|
| slide.bs.carousel | 이 이벤트는 슬라이드 인스턴스 메서드가 호출 될 때 즉시 발생합니다. |
| slid.bs.carousel | 이 이벤트는 회전식 슬라이드 전환(transition)이 완료되면 시작됩니다. |

```
$('#myCarousel').on('slide.bs.carousel', function () { // do something... })
```

3일차

오전

Python

1부 목차

Step 1. 파이썬 알아보기

1.1 파이썬 알아보기

1.2 파이썬 설치

1.3 주석

Step 2.

2.1 타입

2.2 변수

2.3 논리 연산

2.4 함수

2.5 내장함수 사용하기

2.6 리스트

2.7 조건문

2.8 반복문

2.9 클래스

2.10 import

1부 목차

Step 3. 연산

3.1 operate

3.2 비교연산

3.3 논리연산

3.4 할당연산

3.5 bit연산

Step 4. 함수

4.1 들여쓰기

4.2 function

4.3 function 연습

4.4 global

4.5 function 응용

4.6 연산자 우선순위

Step 1

파이썬 알아보기

TUTORIAL 1 파이썬 알아보기

파이썬이란?

파이썬이란 프로그래밍 언어로 1989년 귀도반 로섬이 크리스마스를 심심하지 않게 보내기 위해 집에서 재미삼아 만들기 시작한 프로그래밍 프로젝트입니다.

파이썬은 이해하기 쉽고 간결한 문법으로 생산성이 높아 다양한 분야에서 활용되고 있고 여러 분야에서 선호하는 언어로 꼽히고 있습니다.

어디에 쓰이나요?

파이썬은 거의 모든 분야에 활용되고 있으나 특히 웹 프로그래밍(Django), 자동화 프로그래밍, 빅데이터 분석, 머신러닝에서 주목받고 있습니다.

1. 파이썬 알아보기 (계속)

본 과정은 Python을 로컬에 설치하여 진행하지 않습니다. 하지만 로컬에 설치하시고 사용하시고 싶으신 분들을 위해 다운로드하여 컴퓨터에 설치하는 법을 알려드립니다.

Python 다운로드 하기

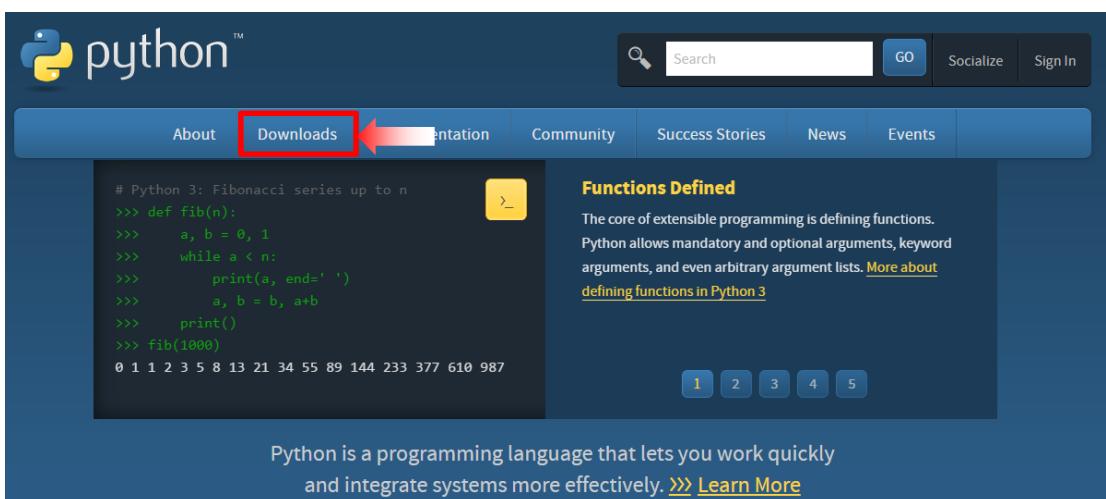
1) 구글' (www.google.co.kr)에서 "python"을 검색해 주세요



2) Welcome to Python.org를 클릭합니다.



3) [Download]를 클릭합니다.

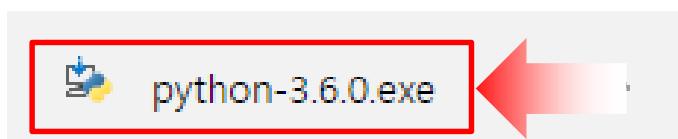


1. 파이썬 알아보기

4) [Download Python 3.6.0]을 클릭합니다.



5) 하단에 파일이 다운로드되면 클릭하여 설치할 수 있습니다.



TUTORIAL 2 환경설정

구름IDE 환경설정을 해보도록 하겠습니다. 아래 화면을 보시면서 빨간색 네모를 따라 그대로 따라하시면 됩니다.



<https://ide.goorm.io/>

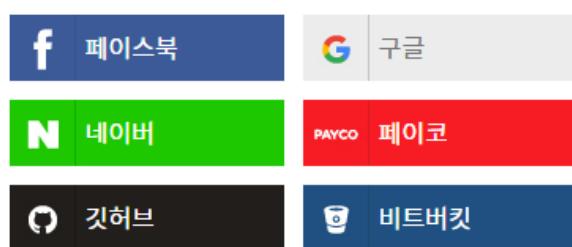
2. 환경설정 (계속)

회원가입을 해야 합니다. 무료계정도 컨테이너 5개를 사용할 수 있는 강력한 서비스입니다. 회원가입을 하면 자동적으로 대시보드에 들어갑니다. 만약 메인화면으로 나오셨다면 대시보드를 눌러주세요.

로그인

이메일 저장

다른 서비스 계정으로 로그인



[비밀번호 재설정](#)

[회원가입](#)

goormide

[가격 정책](#)

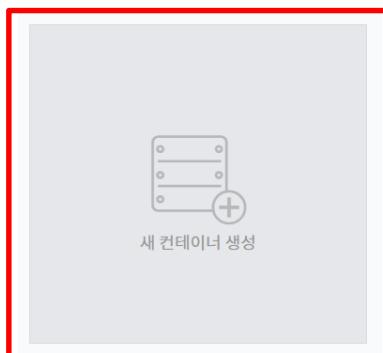
[대시보드](#)

[릴리즈 노트](#)

[친구 초대](#)

2. 환경설정 (계속)

다음은 대시보드 화면입니다. 새 컨테이너 생성을 누르셔서 컨테이너를 생성해 주세요. 하나의 컨테이너는 하나의 컴퓨터를 세팅하는 것과 같습니다. Python을 선택하신 다음 생성하기를 눌러주세요.



컨테이너 생성

컨테이너 생성

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|--|--------|-------|---------------|--------|-------|------------------|------------|------|------|-------|---------|-------------|----|------------------|---------|-------|--------|-------|------|-------|-----|-------|---------|-------|--------|-------------|-------|------|------------|-------|
| 소스 | <input checked="" type="radio"/> 템플릿 <input type="radio"/> Github <input type="radio"/> Bitbucket <input type="radio"/> Git / SVN <input type="radio"/> 압축파일 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 이름 | tutorialdjango | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 설명 | 여행블로그 튜토리얼로 배우는 <u>Django</u> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 소프트웨어 스택 선택 | <table border="1"><tr><td>Search</td></tr><tr><td>C/C++</td></tr><tr><td>Python</td></tr><tr><td>Django</td></tr><tr><td>Flask</td></tr><tr><td>Jupyter Notebook</td></tr><tr><td>TensorFlow</td></tr><tr><td>PyQt</td></tr><tr><td>JAVA</td></tr><tr><td>Maven</td></tr></table> <p>The "Python" option is highlighted with a red box.</p> <table border="1"><tr><td>프로젝트 유형</td><td>Python 프로젝트</td></tr><tr><td>OS</td><td>Ubuntu 14.04 LTS</td></tr><tr><td>Python3</td><td>3.4.3</td></tr><tr><td>Python</td><td>2.7.6</td></tr><tr><td>pip3</td><td>9.0.1</td></tr><tr><td>pip</td><td>9.0.1</td></tr><tr><td>Jupyter</td><td>4.3.0</td></tr><tr><td>Django</td><td>1.11.12 LTS</td></tr><tr><td>Flask</td><td>0.12</td></tr><tr><td>TensorFlow</td><td>1.3.0</td></tr></table> | Search | C/C++ | Python | Django | Flask | Jupyter Notebook | TensorFlow | PyQt | JAVA | Maven | 프로젝트 유형 | Python 프로젝트 | OS | Ubuntu 14.04 LTS | Python3 | 3.4.3 | Python | 2.7.6 | pip3 | 9.0.1 | pip | 9.0.1 | Jupyter | 4.3.0 | Django | 1.11.12 LTS | Flask | 0.12 | TensorFlow | 1.3.0 |
| Search | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C/C++ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter Notebook | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PyQt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JAVA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Maven | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 프로젝트 유형 | Python 프로젝트 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OS | Ubuntu 14.04 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python3 | 3.4.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | 2.7.6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip3 | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter | 4.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | 1.11.12 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | 0.12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | 1.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

취소 **생성하기**

The "Create" button is highlighted with a red box.

2. 환경설정 (계속)

컨테이너가 만들어지고 있다는 화면이 나오고 곧 컨테이너 생성이 완료되었다고 뜹니다. 컨테이너 실행을 누르셔서 컨테이너 안으로 들어가주세요. 만약 모르고 닫으셨다면 대시보드에서 해당 컨테이너 실행을 누르시면 됩니다.

The image consists of three screenshots from a cloud-based development environment:

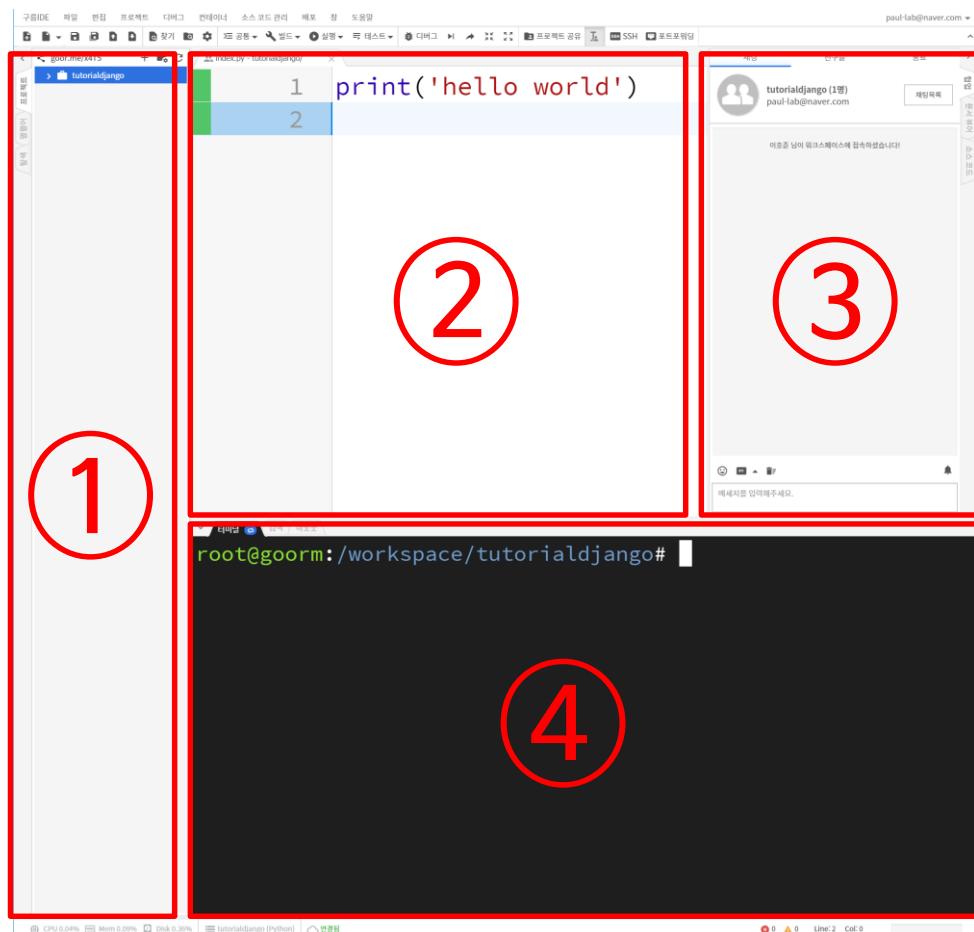
- Screenshot 1:** A progress bar indicating "14%" completion. Below it, a message says "곧 멋진 작업공간이 생성됩니다. 잠시만 기다려주세요." (A nice workspace will be created soon. Please wait a moment.)
- Screenshot 2:** A confirmation dialog titled "컨테이너 실행" (Container Execution) asking "컨테이너를 실행하시겠습니까?" (Would you like to run the container?). It has two buttons: "대시보드로 이동" (Move to Dashboard) and a blue "실행" (Run) button, which is highlighted with a red box.
- Screenshot 3:** The container details page for "tutorialdjango". It shows the last update was "2018. 4. 30. Updated". It lists the following specifications:
 - 소프트웨어 스택: Python
 - 저장 공간: 10GB
 - 항상 켜두기: PREMIUM 상품 구매 시 사용 가능
 - SSH: 컨테이너 실행 시 사용 가능A large red box highlights the "실행" (Run) button at the bottom of the page.

2. 환경설정 (계속)

컨테이너가 로딩되는 화면입니다. 주로 구름에 TIP이나 유명인사의 명언이 나옵니다.



2. 환경설정



1. 풀더 구조를 볼 수 있는 공간입니다. 우측 상단에 보시면 새로고침 버튼이 있어요. 콘솔창에서 뭔가 작업을 했는데 보이지 않는다면 새로고침을 해보시기 바랍니다.
2. txt, html, py파일 등 다양한 파일을 edit 할 수 있는 공간입니다.
3. 협업을 위한 공간이에요. 이 책에서는 사용하지 않는 기능입니다. 채팅 등 협업을 위한 다양한 기능을 제공합니다.
4. 콘솔창입니다. 우리는 대부분의 명령어를 이곳에 입력하게 됩니다.

Step 2

TUTORIAL 1 자료형

제일 처음으로 자료형(Data Type)에 대해 알아보도록 하겠습니다. 자료형이란 주어진 값이 어떤 종류인가를 표현하는 것입니다. 컴퓨터는 주어진 값이 문자인지 숫자인지에 따라 값을 저장하는 방법도 계산하는 방법도 다르기 때문에 자료형을 이해하는 것은 그 언어를 이해하는 기본이라고 할 수 있습니다.

다음 코드를 입력하고 결과를 확인해 보세요.

001.py

```
age = input('나이를 입력하세요: ')
intage = int(age) #형변환
print('제 나이는', age+age, '입니다')
print('제 나이는', intage+intage, '입니다')
print(type(age)) #age의 정체를 알 수 있어요.
print(type(intage)) #intage의 정체를 알 수 있어요.
print(dir(age)) #age의 속성들을 볼 수 있습니다.
print(dir(intage)) #intage의 속성들을 볼 수 있습니다.
```

1. 자료형(계속)

밑의 코드를 확인해 보면 다음과 같은 결과를 얻을 수 있을 것입니다.

Code

```
a = 10    #int
b = '10' #str
c = 10.1 #float
d = -1    #int
e = True #bool : True, False
print(a + a, type(a))
print(b + b, type(b))
print(dir(b))
print('leehojun'.count('e'))
```

Output

```
20 <class 'int'>
1010 <class 'str'>
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__',
 '__hash__', '__init__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__',
 '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__',
 '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold',
 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map',
 'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islower',
 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower',
 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition',
 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
 'translate', 'upper', 'zfill']
```

1. 자료형(계속)

숫자 자료형

앞장의 코드 결과 첫째 줄과 둘째 줄을 보면 같은 둘 다 `10+10`임에도 불구하고 다른 값이 나왔습니다. 이는 두 데이터의 자료형이 다르기 때문입니다. 첫째 줄에 사용된 값 `a`는 숫자 자료형으로 이루어져 있기 때문에 저희가 평소에 계산을 하면 나오는 '20'이라는 결과가 나온 것입니다. 20이라는 결과값 옆에 나온 `<class : int>`가 숫자형 자료형을 표현하는 것입니다.

숫자 자료형의 가장 많이 쓰이는 예로는 다음 두 개가 있습니다.

| | |
|------------|--|
| int(정수)형 | int형은 정수형으로 표현된 숫자형 데이터만 표현할 수 있습니다. ex) 1, 12, 123 |
| float(실수)형 | float는 소수점이 포함된 숫자형 데이터를 표현할 수 있습니다. ex) 3.14, 1.59 |

str

두번째 줄의 `<class : 'str'>`에서 str은 문자열(string)을 표현하는 자료형입니다. 문자열이란 단어나 문자로 이루어진 집합을 의미합니다 앞장 코드의 두번째 줄이 '1010'이라는 결과가 나온 것은 b는 str 즉 문자로 취급되기 때문입니다.

1. 자료형

dir()

dir()은 객체가 내장하고 있는 메소드를 보여줍니다. 메소드란 입력값을 받아 어떤 행동을 만들어 내는 것입니다. 예를 들어 메소드 중에 ‘upper’라는 메소드가 있었습니다. 한 번 이 메소드를 사용해 보도록 하겠습니다.

Code

```
name = 'jejucodingcamp'  
print(name.upper())
```

Output

JEJUCODINGCAMP

다음과 같이 문자가 전부 대문자로 바뀐 것을 볼 수 있습니다.

dir()로 알 수 있는 내장 메소드들은 전부 개발 할 때 유용하게 사용되는 경우가 많습니다.
dir()에 있는 메소드들이 어떤 기능을 하는지는 뒤에서 자세히 설명하도록 하겠습니다.

TUTORIAL 2 변수의 연산

다음에는 변수의 연산에 대해 알아보도록 하겠습니다. 다음 코드를 입력하고 결과를 확인해 보세요.

| Code | Output |
|-------------------------|------------------|
| a = 20 | 23 |
| b = 3 | 17 |
| print(a + b) | 6.66666666666667 |
| print(a - b) | 6 |
| print(a / b) #실수(float) | 60 |
| print(a // b) #정수(int) | 2 |
| print(a * b) | 8000 |
| print(a % b) #나머지 | |
| print(a ** b) #제곱 | |

| 종류 | 설명 |
|----|------------------------------------|
| + | 덧셈 연산자 |
| - | 뺄셈 연산자 |
| / | 나눗셈 연산자, float 형을 반환하므로 소수점까지 나온다. |
| // | 나눗셈 연산자, int 형을 반환하므로 정수형으로 나온다. |
| * | 곱셈 연산자 |
| % | 나눗셈을 했을 때의 나머지를 반환한다. |
| ** | 제곱 연산자, a의 b제곱을 반환한다. |

2. 변수의 연산

다음 코드를 보면 ‘`a += 10`’이라는 계산이 보입니다. 이는 `a`에 `a`와 어떤 숫자를 더한 값을 다시 저장하고 싶을 때 ‘`a = a + 10`’이라고 매번 사용하는 것이 비효율적이기 때문에 만들어진 연산입니다. `a += 10`과 ‘`a = a + 10`’은 같은 연산을 수행합니다.

| Code | Output |
|--|-------------------------|
| <pre>a = 10 a = a + 10 a = a - 5 a += 10 #3 번째 줄과 같은 연산 수행 print(a)</pre> | 25 [Finished in 0.234s] |

| | |
|---------------------|------------------------|
| <code>a += 1</code> | <code>a = a + 1</code> |
| <code>a -= 1</code> | <code>a = a - 1</code> |

TUTORIAL 3 논리 연산

다음으로 논리 연산(Boolean)에 대해 알아보겠습니다. 논리연산에서는 모든 계산 결과를 참 또는 거짓으로 반환합니다. 다음 코드를 입력하고 결과를 확인해 보세요.

| Code | Output |
|----------------------|--------|
| a = 10 | False |
| b = 3 | True |
| print(a == b) #False | True |
| print(a > b) #True | False |
| print(a >= b) | False |
| print(a < b) | True |
| print(a <= b) | True |
| print(a != b) | False |

| | |
|--------|--------------|
| a == b | a와 b가 같다 |
| a < b | a가 b보다 작다. |
| a > b | a가 b보다 크다. |
| a != b | a와 b는 같지 않다. |

3. 논리연산

다음에는 논리 연산의 다양한 활용에 대해 알아보겠습니다. 다음 코드를 보면 and 와 or 연산자와 bool() 연산자가 나와 있습니다. bool()은 숫자형일 경우 1이 참 0이 거짓이고 문자열은 빈칸이면 거짓, 빙칸이 아니면 참입니다.

| Code | Output |
|-------------------------------|--------|
| a = True #1 | False |
| b = False #0 | True |
| print(a and b) #and는 * | False |
| print(a or b) #or는 + | True |
| print(not a) | True |
| | True |
| print(bool(1)) | False |
| print(bool('hello')) | False |
| print(bool(' ')) | 0 |
| print(bool('')) | 15 |
| print(bool(0)) | 30 |
| | 45 |
| for i in range(100): | 60 |
| if i % 3 == 0 and i % 5 == 0: | 75 |
| print(i) | 90 |

TUTORIAL 4 함수

다음으로 함수에 대해 알아보겠습니다. 함수는 어떤 입력값에 대해 어떤 연산을 처리해 결과를 내놓습니다. 앞 장에서 dir()로 살펴보았을 때 다양한 내장함수들이 나왔는데, 저희는 이미 만들어진 내장함수 외에 자신의 상황에 맞는 함수를 만드는 것이 가능합니다. 먼저 간단한 함수를 생성해 보도록 하겠습니다.

Code

```
def leehojun(x):
    result = x * 2
    return result
print(leehojun(100))
```

Output

```
200
[Finished in 0.222s]
```

저희가 만든 함수는 x라는 숫자를 받으면 그것의 두 배를 해서 돌려주는 함수입니다. 결과로 200이 나오는 것을 확인할 수 있습니다.

4. 함수(계속)

보통의 함수는 입력값 > 함수 > 결과값(리턴값)이라는 구조를 가지고 있습니다. 여기서 알아두어야 하는 것은 함수의 결과값은 return을 통해서만 돌려 받을 수 있습니다.
아래는 return이 나오지 않습니다. return이 없을 때에는 return None이 들어갑니다.
* return과 print를 헷갈리시면 안됩니다.

| Code | Output |
|----------------|--------|
| def one(): | one |
| print('one') | two |
| print('two') | three |
| print('three') | one |
| one() | two |
| print(one()) | three |
| print('four') | None |
| | four |

위에서 print('three')의 들여쓰기를 하지 않고 앞으로 이동하여 출력해 보세요. 그리고 그 순서를 위 순서와 비교해보십시오.

print(one())을 하면 None이 출력되는 것을 알 수 있습니다. 위에서 언급해드린 것처럼 return None이 들어갔기 때문입니다.

4. 함수

pass를 했을 경우에도 return None이 자동으로 들어갑니다. 프로그래머는 default라는 말을 많이 쓰는데요. default는 입력하지 않아도 입력되는 값이라고 보시면 됩니다. pass는 default가 return None인 것이죠. 두번째 함수는 바로 return을 하고 있습니다. 이렇게 안에 코드 라인이 없어도 바로 return 해줄 수 있습니다. 세번째 함수는 print를 넣었는데요. 이것은 three라는 함수 안에 print라는 함수가 있는 형태인 것이죠.

마지막으로 함수를 재정의 해보았습니다. one이라는 함수 안에 two라는 함수를 넣어보았는데요. 이 함수는 지금 단계에서는 조금 어려우실 내용입니다. 팩토리 함수라고 하는데요. 들어가 있는 값을 기억합니다.(아래 예제 정도만 파악하고 넘어가도록 하겠습니다.)

| Code | Output |
|----------------------|----------------------|
| def one(): | 9 |
| pass | 16 |
| print(one()) | 9 |
| | 16 |
| def two(x): | hello world |
| return x**2 | [Finished in 0.231s] |
| def three(): | |
| print('hello') | |
| def one(X): | |
| def two(Y): | |
| return Y ** X | |
| return two | |
| f = one(2) | |
| print(f(3)) | |
| print(f(4)) | |
| f2 = one(3) | |
| print(f2(3)) | |
| print(f2(4)) | |
| hojun = print | |
| hojun('hello world') | |

4. 함수

global

다음 코드의 두 함수를 봅시다. one()함수의 결과와 changei()의 결과는 똑같이 20이지만 사실상 전혀 다르게 동작하고 있습니다. one()함수에서는 변수 a에 x의 값을 받아와 계산이 이루어진다. 즉 함수 밖의 x는 값은 10인 채이고, 함수 속의 a만 20으로 변하는 것입니다.

만약 함수 밖의 값을 함수 내에서 직접적으로 바꾸고 싶다면 어떻게 해야할까요? 이 때 전역변수 global을 사용합니다. global i는 함수 밖의 i를 그대로 가져오겠다는 뜻입니다. 그러므로 changei()함수에서는 함수 밖의 값 i도 20이 된 것입니다.

Code

```
x = 10
def one(a):
    a = a + 10
    return a
x = one(x)
print(x)
```

Output

```
20
20
[Finished in 0.269s]
```

```
i = 10
def changei():
    global i
    i += 10
changei()
print(i)
```

TUTORIAL 5 내장함수 사용하기

다음으로는 문자열을 다루는 방법에 대해 알아보겠습니다. 다음 문자열을 만들어 주세요.

Code

```
a = 'hello'  
name = 'hojun'  
nametwo = 'Python'  
namethree = 'paullab CEO leehojun'
```

문장 속에 문자열을 넣어 표현하는 방법은 일반적으로 두가지 방법이 있습니다. 가장 자주 사용되는 방법은 `.format()`을 사용하는 것입니다.

Code

```
print('hello ', name, '!!, my name', nametwo)  
print('hello {}!!, my name{}'.format(name, nametwo))
```

Output

```
hello hojun !!, my name Python  
hello hojun!!, my namePython
```

4. 함수

다음으로 문자열의 인덱싱과 슬라이싱에 대해 알아보도록 하겠습니다. 앞서 설명한 리스트의 index 처럼 문자열에도 index가 있습니다. 문자열에서는 각 문자 하나마다 index가 붙습니다. 공백 또한 문자로 취급되며, name = 'hojun'에서는 h,o,j,u,n 각각의 인덱스가 0,1,2,3,4가 됩니다.

맨 마지막 index의 값을 알기 위해서는 보통 string[-1]과 같은 방법을 사용하는데, 뒤에서 첫번째 요소라는 뜻입니다.

| Code | Output |
|------------------------|----------------------|
| print(name[0]) | h |
| print(name[1]) | o |
| print(name[2:4]) | ju |
| print(namethree[2::2]) | ulbCOleou |
| print(namethree[::-1]) | nujoheel OEC balluap |
| print(namethree[-1]) | n |

슬라이싱은 인덱스 값의 뒤음을 가져오는 데 사용합니다. 보통 string[시작값:끝 값:건너뛰는 값]과 같은 형태로 사용되며 건너뛰는 값을 생략하면 1만큼 뛰는 것으로 생각합니다. 예를 들어 name = 'hojun'에서 name[1:4:2]라는 슬라이싱을 사용하면 반환되는 값은 ou 일 것입니다.

4. 함수

| Code | Output |
|--|----------------------|
| <code>print(name.index('j'))</code> | 2 |
| <code>print(namethree.count('l'))</code> | 3 |
| <code>print(namethree.index('l'))</code> | 3 |
| <code>print(namethree.strip(' l '))</code> | paullab CEO leehojun |

슬라이싱은 인덱스 값의 묶음을 가져오는 데 사용합니다. 보통 string[시작값:끝 값:건너뛰는 값]과 같은 형태로 사용되며 건너뛰는 값을 생략하면 1만큼 뛰는 것으로 생각합니다. 예를 들어 name = 'hojun'에서 name[1:4:2]라는 슬라이싱을 사용하면 반환되는 값은 ou 일 것입니다.

TUTORIAL 6 집합 자료형

이번 시간에는 집합 자료형에 대해 알아보도록 하겠습니다.

리스트

제일 먼저 리스트에 대해 알아보도록 하겠습니다. 리스트는 집합 자료형 중 가장 흔하게 쓰이는 자료형으로 여러 개의 데이터를 하나의 변수로 관리할 수 있게 해줍니다. 리스트는 다음과 같이 사용합니다.

Code

```
l = [100, 200, 300, 400]  
INDEX 0   1   2   3
```

집합 자료형에서는 값의 위치를 통해 그 값을 불러올 수 있습니다. 이 때 그 위치를 index라고 부르며 0부터 시작합니다. 그러므로 위 코드에서 100은 l[0]에 위치합니다.

리스트 속에 리스트를 만들 수도 있습니다. 이를 중첩 리스트라고 부르는데 안의 값이 많아지면 관리하기 어렵기 때문에 자주 사용되지는 않지만 알아두는 편이 좋습니다. 값의 위치를 차근차근 따라가 보세요.

Code

```
l = [100, 200, 300, [1000, 2000, [1, 2]]]  
print(l[3])  
print(l[3][2])  
print(l[3][2][1])
```

Output

```
[1000, 2000, [1, 2]]  
[1, 2]  
2  
[Finished in 0.254s]
```

6. 집합 자료형 (계속)

이번에는 리스트를 다루는 법에 대해 알아보도록 하겠습니다. 리스트는 안의 값을 생성, 삭제, 수정 할 수 있어 가변값을 다룰 때 편리합니다.

리스트의 내장함수

append(a)는 리스트의 끝에 a라는 요소를 추가합니다. 다음의 예시 뿐만이 아니라 l.append([100])과 같은 형식으로 사용하면 리스트 속에 리스트를 추가할 수 있습니다.

| Code | Output |
|---|--------------------------------|
| <pre>l = [100, 200, 300, 400] print(type(l)) print(dir(l)) l.append(500) l.append(100) print(l)</pre> | [100, 200, 300, 400, 500, 100] |

count(a) 함수는 리스트에 포함된 요소 a의 개수를 셹니다. 저희가 만든 리스트 속에 100이라는 요소는 한 개 뿐이므로 결과로 1이 출력됩니다.

| Code | Output |
|---|--------|
| <pre>l = [100, 200, 300, 400] print(l.count(100))</pre> | 1 |

6. 집합 자료형 (계속)

| Code | Output |
|---|-------------------------------------|
| <pre>l.extend([100, 200, 300]) print(l)</pre> | [100, 200, 300, 400, 100, 200, 300] |

방금 설명한 append(a)를 사용하면 값을 하나씩 밖에 넣을 수 없습니다. 그럴 때 expand(a)를 사용하면 요소를 한번에 추가할 수 있습니다.

index() 함수는 이번 장 제일 처음 설명한 index 즉, 요소의 위치를 반환합니다.

| Code | Output |
|---|----------------------------------|
| <pre>l = [100, 200, 300, 400] print(l.index(100)) l.insert(3, 10000) print(l)</pre> | 0 [100, 200, 300, 10000, 400] |

append(a)는 리스트에 요소를 순서대로 추가합니다. 하지만 리스트 내 원하는 위치에 값을 추가하고 싶은 경우도 있을 것입니다. 그럴 때 insert(a, b)라는 함수를 사용할 수 있습니다. 이는 a라는 위치에 b라는 값을 추가하라는 의미입니다.

6. 집합 자료형 (계속)

리스트 내의 요소를 삭제하는 방법에는 2가지가 있습니다. `pop()`을 사용하면 리스트의 마지막 값을 리턴하고 삭제합니다. `pop(a)`를 사용하면 a 위치에 있는 값을 돌려주고 그 값을 삭제합니다.

| Code | Output |
|---------------------------------------|-----------------|
| <code>l = [100, 200, 300, 400]</code> | |
| <code>print(l.pop())</code> | 400 |
| <code>print(l)</code> | [100, 200, 300] |
| <code>l.remove(100)</code> | [200, 300] |
| <code>print(l)</code> | |

다음으로 리스트 내의 값을 정렬하는 방법에 대해 알아보겠습니다. 리스트 내 값을 오름차순으로 정리하는 함수가 `sort()`입니다. `reverse()`함수는 값을 내림차순으로 정리합니다.

| Code | Output |
|---------------------------------------|----------------------|
| <code>l = [100, 200, 300, 400]</code> | [100, 200, 300, 400] |
| <code>l.sort()</code> | [400, 300, 200, 100] |
| <code>print(l)</code> | [100, 200, 300, 400] |
| <code>l.reverse()</code> | [400, 300, 200, 100] |
| <code>print(l)</code> | |
| <code>print(sorted(l))</code> | |
| <code>print(l)</code> | |
| <code>print(reversed(l))</code> | |

다른 방법으로 `sorted(list)`와 같은 방법이 있습니다. `sort()`과 `sorted()`은 같은 내용을 반환하는 것처럼 보이지만 실제로는 조금 다릅니다. `sort()`, `reverse()`를 사용하면 원본의 순서가 변합니다. 하지만 `sorted()`, `reversed()` 함수는 원본의 복사본을 만들어 정렬하는 개념이기 때문에 원본은 변경되지 않습니다.

6. 집합 자료형 (계속)

리스트의 내장함수

다음 코드는 리스트 값 변환에 대해서 보여줍니다.

Code

```
t = (100, 200, 300, 400)
l = [100, 200, 300]
t2 = (100, 200, 300, 1)
```

```
print(t2)
l[0] = 1000
print(t2)
def change(x):
    print(id(x))
    x[0] = 10000
change(l)
print(id(l))
print(l)
```

Output

```
(100, 200, 300, [100, 200, 300])
(100, 200, 300, [1000, 200, 300])
3868792
3868792
[10000, 200, 300]
[Finished in 0.242s]
```

6. 집합 자료형

튜플

같은 집합 자료형으로 튜플에 대해 설명하도록 하겠습니다. 튜플은 리스트와 매우 흡사하지만 안에 있는 값을 생성, 삭제, 수정 할 수 없다는 특징을 가지고 있습니다. 리스트 내용을 바꿀 필요가 없다거나 프로그램 실행 도중 값이 변하지 않기를 바란다면 튜플을 사용하는 것이 좋습니다.

딕셔너리

딕셔너리 자료형은 key와 value 값을 함께 갖습니다. 딕셔너리는 순서가 없고, key 값이 중복 될 수 없으며 값을 추가하거나 삭제 할 수 있습니다.

Code

```
d = {'name': 'hojun', 'age': 34, 'height': 178}  
#dict{key:value}  
print(d['name'])  
d['money'] = 100  
print(d)
```

Output

```
hojun  
{'height': 178, 'money': 100, 'age': 34, 'name': 'hojun'}  
[Finished in 0.271s]
```

TUTORIAL 7 조건문

이번엔 조건문에 대해 알아보도록 하겠습니다. 개발을 하다보면 조건에 따라 결과가 다르게 나오도록 해야 할 때가 반드시 있습니다. 그럴 때 조건문을 사용하는데, 조건문은 조건이 참일때만 그 다음의 코드를 실행시킵니다.

Code

```
if True:  
    print('hello jeju')  
  
if False:  
    print('hello hojun')  
  
print('good job')
```

Output

```
hello jeju  
good job
```

기본적으로 아무 조건이 붙지 않으면 True이므로 ‘hello jeju’가 출력됩니다.

보통 if문은 if…,else…와 같이 사용되는데 이는 조건을 두 가지밖에 표현할 수 없습니다. 만약 조건이 여러 개라면 어떻게 해야 할까요?

7. 조건문

다음 코드는 score에 따라 엄마가 주는 용돈의 액수가 달라지는 코드입니다. 90점 이상이면 100만원을, 80점 이상이면 10만원을 나머지 경우에는 1만원을 받습니다. 조건이 총 3개이기 때문에 if…, elif…, else…로 나누어 지는 것을 볼 수 있습니다.

| Code | Output |
|---|--|
| <pre>score = 95 money = 1000 if score > 90: print('mom : i'm so happy') money += 1000000 elif score > 80: print('mom : i'm so happy') money += 100000 else: print('mom : i'm so happy') money += 10000 print(money)</pre> | mom : i'm so happy 1001000 hello world |

7. 조건문

while문을 사용해 보겠습니다. while문은

while <조건문>:

<수행할 문장>

과 같은 형식을 가지고 있습니다. 조건문이 참일 동안 문장을 수행합니다.

break / pass

Code

```
while True:  
    order = input('##')  
    if order == 'exit':  
        break  
    elif order == 'name':  
        print('hojun')  
    else:  
        pass  
  
    x = 0  
    while x < 10:  
        print('hello')  
        x += 1
```

위 코드에서 while문은 항상 True이므로 무한으로 수행 문장을 반복하게 됩니다. 이럴 때 break문을 사용하면 루프에서 강제적으로 빠져나올 수 있습니다.

pass는 참 거짓을 따라 실행할 때 아무것도 하지 않고 넘어가고 싶을 경우에 사용합니다.

TUTORIAL 8 반복문

for문은 while문과 비슷한 형태를 가진 반복문입니다. 특히 파이썬에서의 for문은 굉장히 직관적인 성격을 띠고 있습니다.

| Code | Output |
|--|--------|
| <code>print(100 in [100, 200])</code> | True |
| <code>for i in 'leehojun':</code> | l |
| <code> print(i)</code> | e |
| | e |
| | h |
| <code>for i in [100, 200, 300]:</code> | o |
| <code> print(i)</code> | j |
| | u |
| <code>for i in (100, 200, 300):</code> | n |
| <code> print(i)</code> | 100 |
| | 200 |
| | 300 |
| <code>for i in {100, 200, 300}:</code> | 100 |
| <code> print(i)</code> | 200 |
| | 300 |
| | 200 |
| | 100 |
| | 300 |

위 코드를 보면 for ... in 형태로 이루어져 있는 것을 알 수 있습니다. 여기서 i라는 변수에 리스트 안의 값이 차례대로 대입되어 print(i)로 인해 출력되고 이를 리스트의 마지막 요소 까지 반복합니다.

8. 반복문

range

range는 지정된 숫자 리스트를 자동으로 만들어주는 함수입니다. range(a, b)라는 리스트를 만들면 a이상 b미만의 리스트를 자동으로 만들어 줍니다. 만약 range(a, b, c)의 형태로 사용한다면 a이상 b 미만의 리스트를 c만큼 띄어서 만들겠다는 소리입니다.

range만으로도 편리하지만 특히 for문과 결합해 for in range 문으로 만들 때 큰 효과를 발휘합니다.

Code

```
print(range(10000)) #range(0, 10)
print(type(range(10))) #class 'range'

for i in range(10):
    print(i)
print('-----')
for i in range(2, 10, 2):
    print(i)
print('-----')
sumi = 0
for i in range(2, 101, 2):
    sumi += i
print(sumi)
```

Output

```
range(0, 10000)
<class 'range'>
0
1
2
3
4
5
6
7
8
9
-----
2
4
6
8
-----
2550
```

8. 반복문

연습문제 : 구구단

다음은 for in range문을 사용하여 2단, 3단, 구구단을 출력하는 코드입니다. 코드를 먼저 보지 마시고 for문을 활용하여 구구단을 만들어 보세요.

Code

```
for i in range(1, 10):
    print('{} X {} = {}'.format(2, i, 2*i))

for i in range(1, 10):
    print('{} X {} = {}'.format(3, i, 3*i))

for j in range(2, 10):
    for i in range(1, 10):
        print('{} X {} = {}'.format(j, i, j*i))

l = []
for i in range(10):
    l.append(i)

l2 = [i for i in range(10)]

l3 = ['{} X {} = {}'.format(i, j, i*j)
      for i in range(2, 10)
      for j in range(1, 10)]

print(l)
print(l2)
print(l3)
#리스트 안에 공백은 병합
l4 = [
    '제주코딩베이스캠프7기',
    '제주코딩베이스캠프8기',
]
```

TUTORIAL 9 클래스

저희는 앞서 함수에 대해 배웠습니다. 매우 편리한 기능이지만 만약 비슷한 기능의, 비슷한 이름을 가진 함수가 너무 많다면 관리가 어려울 것입니다. 이 때 클래스를 사용해서 비슷한 함수를 묶어 관리할 수 있습니다.

Code

```
class Car():
    maxSpeed = 300
    maxPeople = 5
    def move(self, x):
        print(x, '의 스피드로 움직이고 있습니다.')
    def stop(self):
        print('멈췄습니다.')
k3 = Car()
k5 = Car()
print(k3.maxSpeed)
print(k5.maxPeople)
k3.move(5)
print(type(k3))
print(dir(k3))
print(k3)
```

Output

```
300
5
5 의 스피드로 움직이고 있습니다.
<class '__main__.Car'>
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__',
 '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__le__', '__lt__',
 '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
 '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__',
 'maxPeople', 'maxSpeed', 'move', 'stop']
<__main__.Car object at 0x01A03030>
[Finished in 0.368s]
```

9. 클래스

클래스와 인스턴스

클래스와 인스턴스에 대해 알아봅시다. 일반적으로 클래스는 같은 모양의 빵을 무수히 만들 수 있는 빵틀로, 인스턴스는 그 빵틀로 만들어진 빵에 비유됩니다. 여기서 각 인스턴스들은 서로 독립적인 성격을 가집니다. 즉, 만약 객체 a가 바뀐다고 해도 b에는 영향을 주지 않습니다.

상속

클래스의 상속에 대해 알아보도록 하겠습니다. 상속을 사용하면 기존 클래스 자체를 변경하지 않고 클래스의 기능을 추가하거나 변경할 수 있습니다. 부모가 자식에게 재산을 상속해 주는 것처럼, 기능을 상속하는 클래스를 부모 클래스, 기능을 상속 받는 클래스를 자식 클래스라고 합니다.

Code/ 상속

```
class Car():
    maxSpeed = 300
    maxPeople = 5
    def move(self, x):
        print(x, '의 스피드로 움직이고 있습니다.')
    def stop(self):
        print('멈췄습니다.')

class HyCar(Car):
    battery = 100
    batteryKM = 300

hyk3 = HyCar()
print(hyk3.maxSpeed)
```

Output

```
300
[Finished in 0.24s]
```

TUTORIAL 10 import

다음은 import를 통해서 다른 파일에 있는 데이터를 가져오는 방법을 알아보도록 하겠습니다. 먼저 ‘test.py’ 파일을 생성해 주세요.

test.py

```
name = 'hojun'  
age = 34
```

두가지 방법으로 test.py에서 이름과 나이를 출력해 보도록 하겠습니다.

새로운 파일을 만들고 각각 결과를 확인해 보세요. 둘 다 같은 결과를 출력합니다.

Code

```
from test import age, name  
print(name, age)
```

Code

```
import test  
print(test.name, test.age)
```

Output

```
hojun 34  
[Finished in 0.256s]
```

지금은 간단한 예시로 설명했지만 실제 개발에서는 import는 매우 중요하게 다뤄집니다. 다뤄야 할 데이터양이 너무 많거나, 정보를 속성에 따라 나눠야 할 때가 많아 파일을 따로 생성한 후 import를 통해서 불러오는 게 더 효율적인 관리가 되기 때문입니다.

3일차

저녁

홈페이지 기획, 개발자 도구

홈페이지 기획하기

이번 시간 학습할 것

이번 시간에는 free mind나 oven을 이용해 홈페이지를 기획하는 방법에 대해 배워보겠습니다. 함께 기능 목록을 작성하고, 와이어 프레임을 그려보며 올바른 기획을 하는 방법에 대해 생각해 보도록 합시다.

이번 시간 주요 질문

기획을 하기에 앞서 먼저 해야 하는 질문은 다음과 같습니다

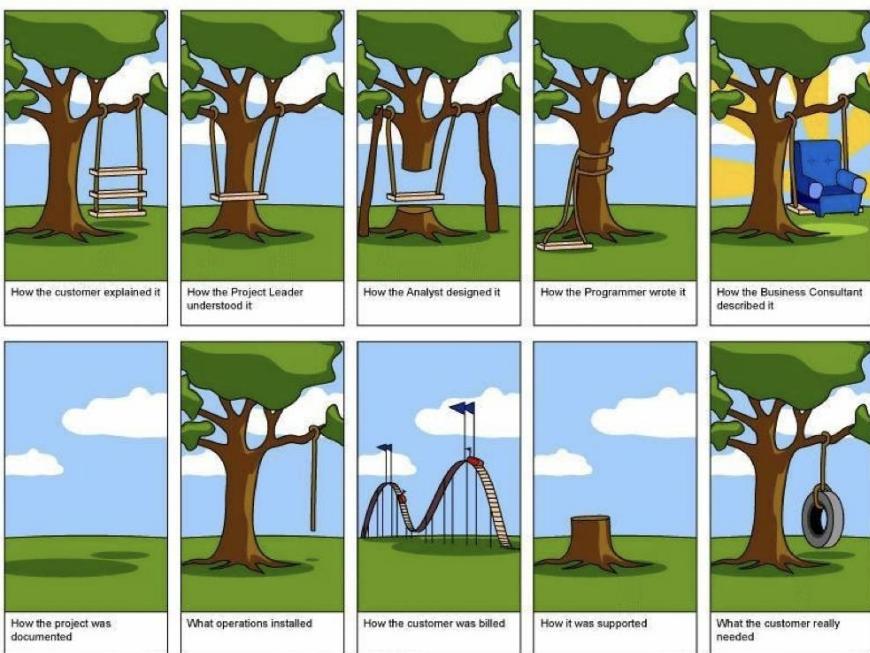
- 왜 웹사이트를 만드는가?
- 핵심 요소는 무엇인가?
- 누가 이 홈페이지를 방문하는가?
- 기능에는 무엇이 있나?

다음 질문들을 잘 생각해보며 이번 수업을 시작하도록 하겠습니다.

TUTORIAL 0 화면설계가 필요한 이유

보통 프로젝트를 할 때는 혼자서 하는 게 아닌 디자이너, 기획자, 개발자와 같은 다양한 사람이 참여하여 이루어집니다. 이 때 화면설계는 아이디어 발안자가 아닌 다른 사람들이 작업을 이해하고 아이디어를 소통하는데 중요한 역할을 합니다. 화면설계는 프로젝트의 청사진이자 목적지이기도 합니다.

만약 화면설계를 하지 않은 채 프로젝트를 진행한다면 아래와 같은 결과가 벌어지기도 합니다.



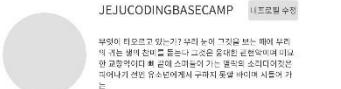
0. 화면설계의 필요성

다음은 화면 설계에 대한 예시입니다.

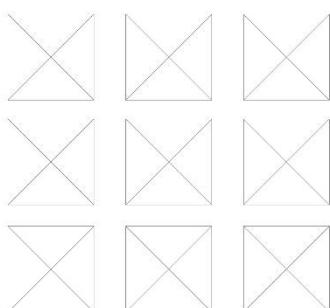


- 스토리보드 -

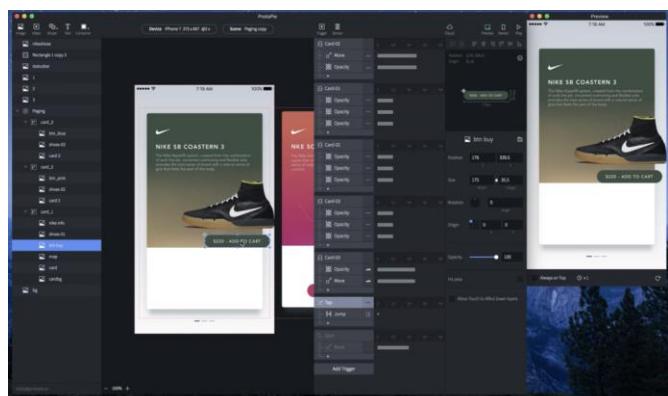
INSTACLONE



무엇이 티오르고 있는가? 누리 눈이 그것을 보는 데에 무리의 궁금증을 풀어주는 듯하다. 그것은 중대한 현상을 알며 미묘한 교향곡이다. 궁금증은 스며들어 가는 몽의 소리다. 이것은 피어나기 전인 유소년에게서 구하지 못할 바이며 시들어 가



- 와이어프레임 -



- 프로토타입 -

TUTORIAL 1 OVEN

이 챕터에서는 개발과는 직접적인 관련이 없지만 앞서 한 질문들을 바탕으로 웹사이트를 제작하는데 큰 도움이 되는 도구들을 소개하겠습니다. 저희가 지금 만들고자 하는 웹사이트 뿐만 아니라 앞으로 프로젝트를 기획하는데 큰 도움이 되는 도구들입니다.

Oven은 프로토타입을 만들기 적절한 툴입니다. input을 최소화 하여 화면 구성을 할 수 있습니다.

'쓸만한' 프로토타이핑을 설계하려면,
프로토타입 전용툴이 필요합니다.

"아.. 제가 화면 레이아웃을 잘못 생각했네요"

파워포인트로는 이렇게 하면 될 줄 알았는데
막상 디자인과 개발이 시작되고 나서 아차 싶었던 적이 있으신가요?
스크린 환경에는 실제 디바이스 환경에 기반한 기획이 필요합니다.
프리젠테이션 툴, 이런 프리젠테이션 용으로만 사용하세요.
Oven으로 보다 꼼꼼한 기획을 준비할 수 있습니다.



- 캡쳐화면 -

1. OVEN (계속)

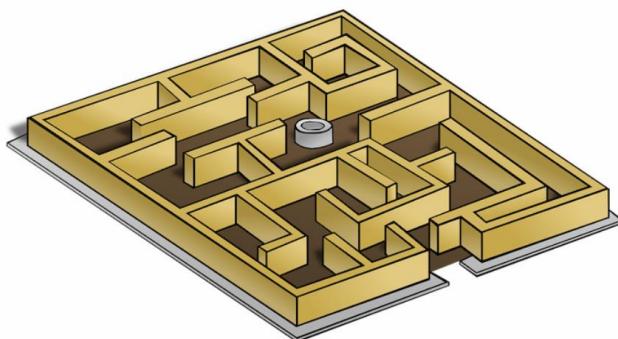
다음은 오븐에서 소개하는 내용입니다. 이미지 링크 기능, 설계 전 테스트 등을 장점으로 내세우고 있습니다.



개발의 첫 삽을 뜨기 전
미리 설계하고 테스트 해볼 수 있습니다.

"난 직접 코드짜는게 더 편해"

편하게 하겠지만... 기획과 설계없이 바로 진행하는 개발, 과연 생각대로 효율적일까요?
Oven의 100여가지 UI 컴포넌트들과 500여개에 달하는 벡터 아이콘으로 뚝딱뚝딱 미리 설계해보세요.
예상치 못했던 구멍들, 분명히 눈에 보이실겁니다.
시행착오도 그만큼 줄어들겠죠.



- 캡쳐화면 -

1. OVEN (계속)

또한 쉬운 페이지 설정과 아이콘, 컴포넌트를 제공하여 사용자가 쉽고 편하게 프로토타입을 만들 수 있도록 합니다. 다음 장에서는 간단한 튜토리얼 페이지를 작성해 봅니다.

Features

손쉬운 페이지 드로잉



생각하는 대로 혹은 보이는 대로 개체를 가져다 놓고, 편집하기만 하면 됩니다. 익숙하게 사용해왔던 그 방법 그대로 알아요. 그리고 다양한 스타일편집 기능과 효과 적용을 통해 마치 도화지에 그림 그리듯 쉬운 방식으로 실제 웹기반 프로토타이핑을 만들어 낼 수 있습니다.



100여개의 기본제공 컴포넌트

Oven은 13개 카테고리 총 100여개의 멋진 UI 컴포넌트를 제공합니다. 그리고 이 컴포넌트들은 원하는 형태로 얼마든지 변형해 사용할 수 있습니다.

1,200여개의 방대한 벡터아이콘



쉽고 명료한 오픈라이센스 아이콘들을 제공합니다. 더 놀라운 것은 이 모두가 어떠한 사이즈, 어떠한 컬러로도 변경 할 수 있는 벡터 아이콘이라는 점입니다.



다양하게 선택가능한 스크린

모바일과 태블릿, PC스크린 사이즈까지 다양한 사이즈 선택을 지원합니다. 또한 테스트할 모바일 기기의 화면 크기가 제각각이어도 걱정할 필요가 없이 Oven에서는 속편하게 하나로만 작업하시면 됩니다. 어떠한 사이즈의 폰에서도 자동스케일링되어 깨지지 않고 멋지게 보이니까요.

간편한 이미지업로드와 편집



직접 만든 아이콘 이미지나 이미 작업한 사진이 있다구요? 프로젝트 편집화면에 내컴퓨터의 이미지파일을 드래그&드롭하기만 하면 바로 유저 계정으로 업로드됩니다. 그리고 업로드한 이미지들은 이미지크롭/필터 기능을 이용해 새로운 이미지로 재탄생시킬 수 있습니다.



템플릿 동기화, 클릭 한번으로 모두를 변화시키는 마법

수십장의 페이지를 만들었는데, 상단의 공통로고가 바뀌었다구요? 더이상 일일히 페이지 수정을 하실 필요가 없습니다. '템플릿 동기화' 기능을 통해 클릭 한번으로 프로젝트 전체를 변경할 수 있습니다.

손쉬운 링크이동 설정

화면의 각 요소들이 페이징크로 연결되는 그 순간이야말로 프로토타입이 생명력을 갖게 되는 순간입니다. 각 요소에 대한 '링크연결 설정' 기능을 통해 외부 URL이나 프로젝트내 페이지를 연결하여 테스트해보세요.

문장 자동생성 기능

대부분의 프로토타입 툴이나 오피스 프로그램들은 텍스트 상자나 위젯을 가져다 놓으면 항상 같은 초기값(예. "텍스트")으로 페이지에 삽입이 됩니다. 결국 그럴듯한 프로토타입을 만들려면 글자 내용을 직접 타이핑해야 한다는 이야기지요. Oven에서는 빅스 사이즈에 따라 영문 또는 한글의 Lorem Ipsum (의미 없는 무작위 문장)을 자동으로 생성해줍니다.

Font 호환성

원도우에서 작성한 문서와 액에서 작성한 문서가 서로 달라보이는 현상은 바로 시스템 기본 폰트가 다르기 때문에 발생합니다. Oven은 이 문제를 해결하기 위해 오픈라이센스 폰트인 본고딕체(HanSans)를 한국어 환경에서의 기본 글꼴로 적용하였습니다.

QR로 찍고 바로 테스트

Oven은 '웹 테스트' 모드를 제공합니다. QR 또는 단축URL을 통해 PC는 물론 스마트디바이스에서 곧바로 작동가능한 웹테스트를 실행할 수 있습니다.

오브젝트 검색

수백개의 아이콘과 수많은 컴포넌트들을 일일히 찾아본다는 건 생각도 할 수 없습니다. 필요한 요소는 컴포넌트 검색창으로 검색해서 사용하세요.

나만의 컴포넌트

열심히 만든 컴포넌트, 다음에 또 쓰일지 모른다면 '보관함에 보내기'를 통해 보관해 둘 수 있습니다.

공유 옵션

프로젝트별 공유 옵션 목록을 통해 유저별 읽기권한을 설정할 수 있습니다. 또는 링크를 아는 유저는 누구나 로그인없이 접근할 수 있도록 할 수도 있습니다.

다른 파일로 내보내기

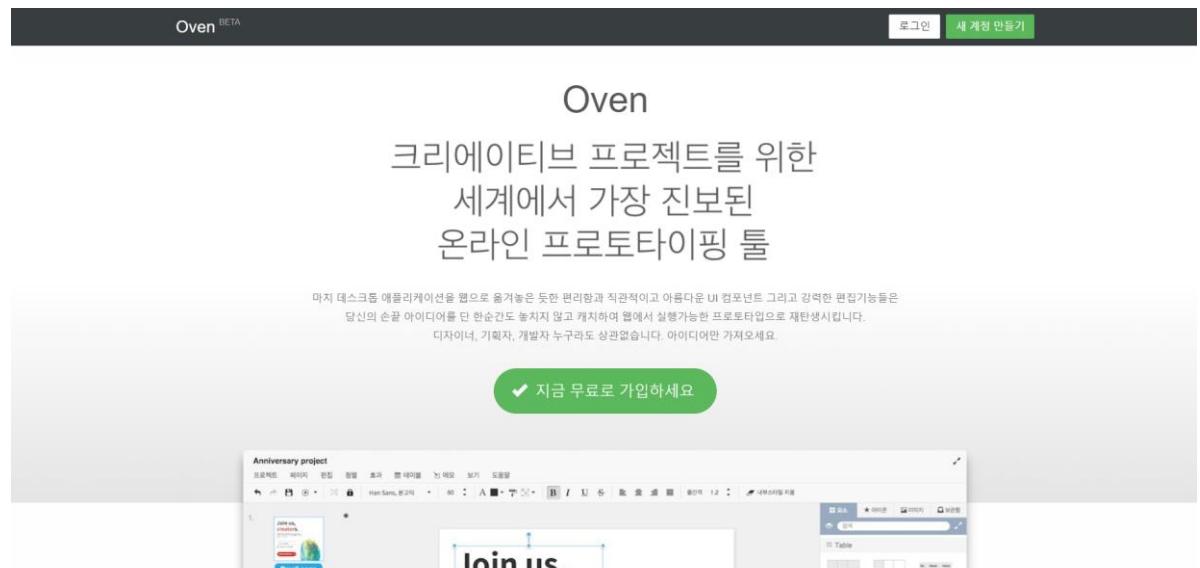
'페이지 내보내기'를 통해 JPG, PNG 또는 PDF파일로 내려받을 수 있습니다. 매일이나 다른 문서에 첨부하거나 문서출력을 해야 할 경우에 유용합니다.

- 캡쳐화면 -

1. OVEN (계속)

1) 초기화면 구성

- 로그인 또는 새 계정 만들기를 클릭하여 시작해주세요.



2) 시작화면

- 여기서는 새로운 프로젝트를 만들 수 있으며 만든 프로젝트를 다른 사람과 공유할 수도 있습니다.

프로젝트 대수보드

The screenshot shows the Project Dashboard. At the top, there is a search bar labeled "프로젝트 검색" and a button labeled "새로운 프로젝트 만들기". Below the search bar, there are two cards representing existing projects:

- mini note**: This card shows a preview of a document with several text entries. It includes a date stamp "2017년 8월 12일" and a button "로그인없이 텍스트 가능".
- 온오프믹스**: This card shows a preview of a page with the title "ABOUT PROJECTS". It includes a date stamp "2017년 8월 5일" and a button "로그인없이 텍스트 가능".

1. OVEN (계속)

3) 새 프로젝트 만들기

- 다양한 레이아웃에 맞게 프로젝트를 만들 수 있습니다.

➊ 새로운 프로젝트 만들기

프로젝트 이름 *

상세내용

기본 화면사이즈

| | |
|------------------------|------------------------|
| Mobile 세로모드, 360 px | Mobile 가로모드, 640 px |
| 600 px Tablet | 768 px Tablet |
| 978 px PC | 1200 px PC |

ⓘ 신규 페이지 생성 시 적용되는 화면너비 사이즈이며, 프로젝트 생성 후 페이지 별로 변경 가능합니다.

테스트URL 공유

로그인없이 보기 허용 특정 사용자에게만 허용

ⓘ 테스트페이지 URL을 전달받은 유저는 Oven 계정 생성이나 로그인을 하지 않아도 테스트페이지에 접근할 수 있습니다. 프로젝트 생성 후에는 '공유옵션'을 통해 변경 가능합니다.

새 프로젝트 만들기

1. OVEN(계속)

4) 편집화면 - 1

- HTML 5로 적용되어 있으며 화면 크기에 따라 반응하는 반응형 웹을 적용하였습니다.

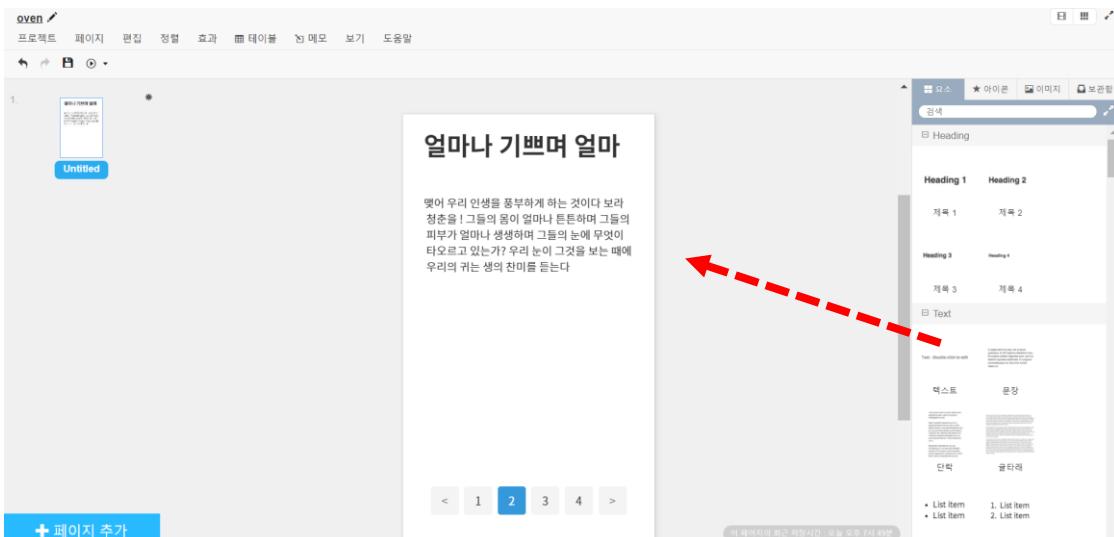
오른쪽에는 다양한 툴이 제공되는데 이해하기 쉽고 빠른 편집을 도와줍니다.



5) 편집화면 - 2

- 요소, 아이콘을 오른쪽 화면에서 드래그하여 UI를 편집할 수 있습니다.

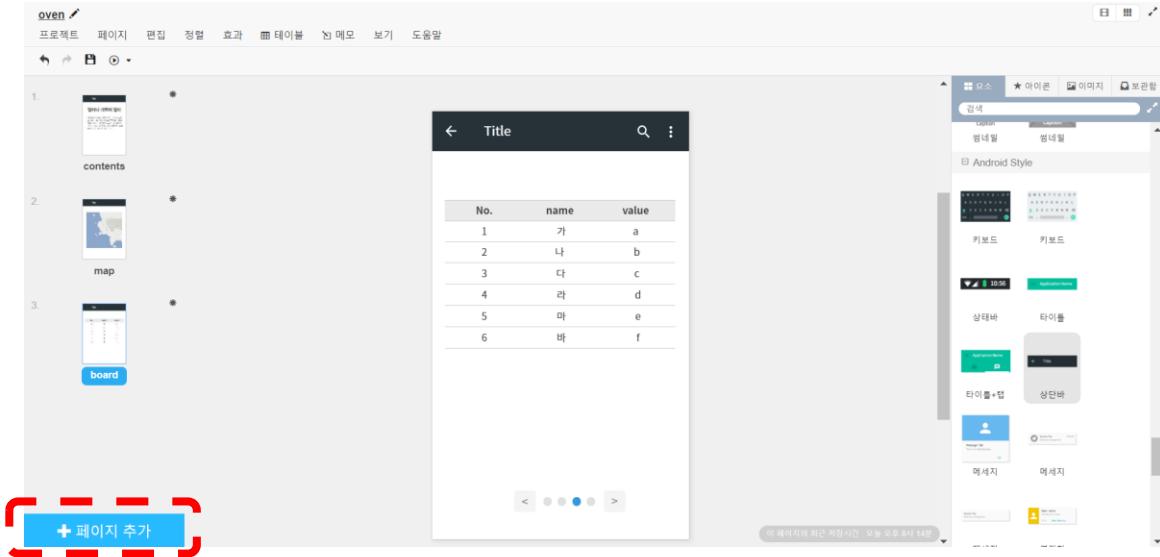
한글 입술이 자동으로 삽입됩니다.



1. OVEN(계속)

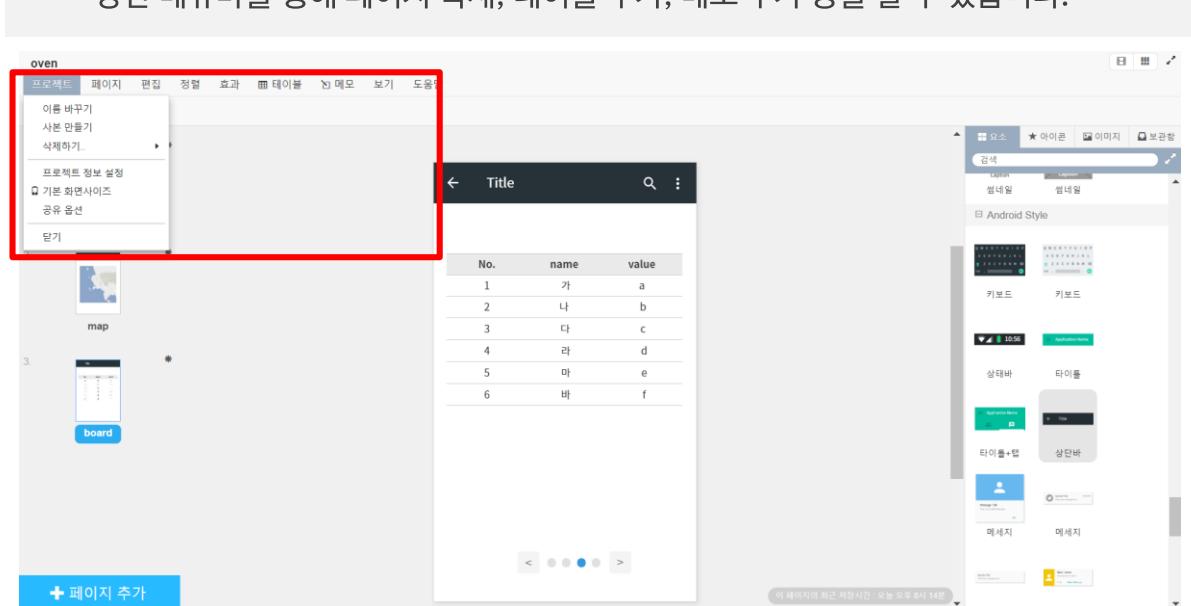
6) 편집화면 - 3

- 페이지 추가를 통하여 여러 화면을 만들 수 있고,
구분이 쉽게 화면 이름 설정이 가능합니다.
또한 클릭해서 이동할 수 있는 링크 페이지를 제공합니다.



7) 편집화면 - 4

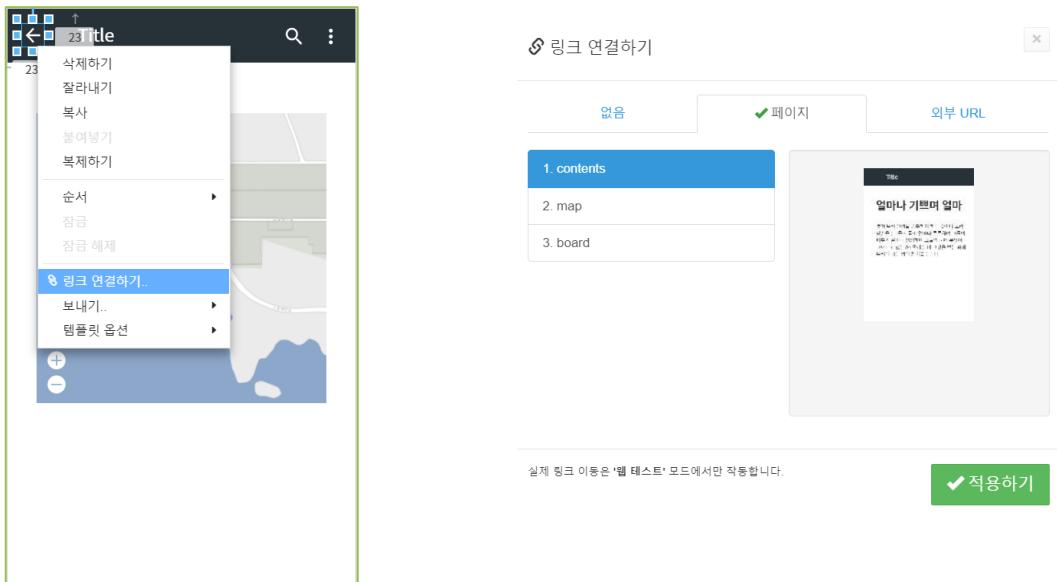
- 상단 메뉴바를 통해 페이지 복제, 테이블 추가, 메모 추가 등을 할 수 있습니다.



1. OVEN (계속)

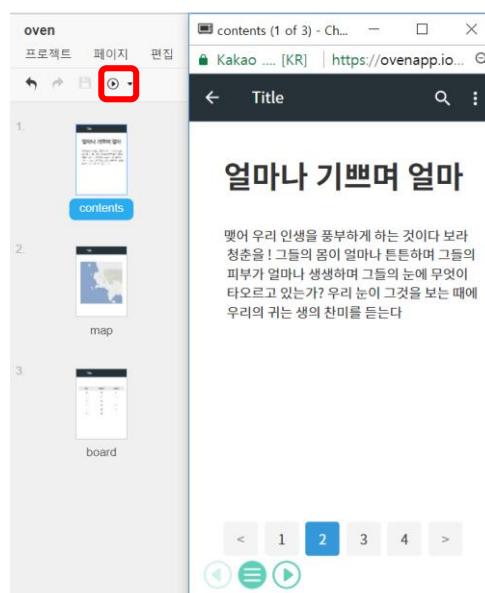
8) 링크연결

- 링크 연결하기를 통해 이미지, 버튼, 글귀 등에 페이지를 연결할 수 있습니다.



9) 웹 테스트

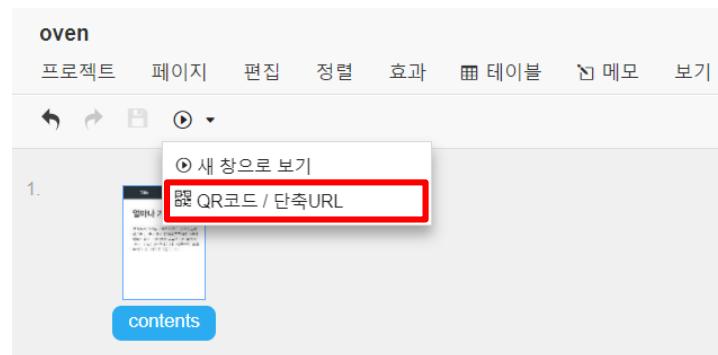
- 아래 버튼을 클릭하여 자신이 만든 것을 테스트를 하며
수정, 보완 등을 할 수 있습니다.



1. OVEN

10) QR코드 / 단축 URL

- 아래 버튼을 클릭하여 자신이 만든 것을 공유 또는 실행을 할 수 있습니다.



TUTORIAL 2 FREEMIND

FreeMind는 생각을 확장하기 좋은 프로그램입니다. 한국 이스트소프트에서 개발한 알마인드도 사용하기 좋습니다.

마인드 맵은 중심 주제를 가지고 키워드를 가지 단위로 나눠 시각화 할 수 있는 유용한 도구입니다.

린 스타트업, 애자일 기법을 사용하여 웹 서비스를 만들 때, 기획서 쓰느라 시간과 노력을 낭비하기보다는 마인드 맵을 활용하여 정리한 다음 바로 사이트를 구축해 사용자에 피드백을 받아가며 성장시킬 것을 추천해 드립니다.(물론 규모가 있는 회사에서는 힘든 방법입니다.)

2. FREEMIND (계속)

1) FreeMind 설치

- google에서 FreeMind를 검색하시고 download를 할 수 있는

‘FreeMind : Main Page’를 클릭해주세요.

A screenshot of a Google search results page for the query "freemind". The search bar contains "freemind". Below the search bar are navigation links: 전체 (selected), 이미지, 동영상, 뉴스, 도서, 더보기, 설정, and 도구. The search results section shows the first result: "FreeMind: Main Page" from freemind.sourceforge.net. The snippet includes a link to "Download" and a brief description of FreeMind as a Java-based mind-mapping software. There are also two other snippets on the right side of the result.

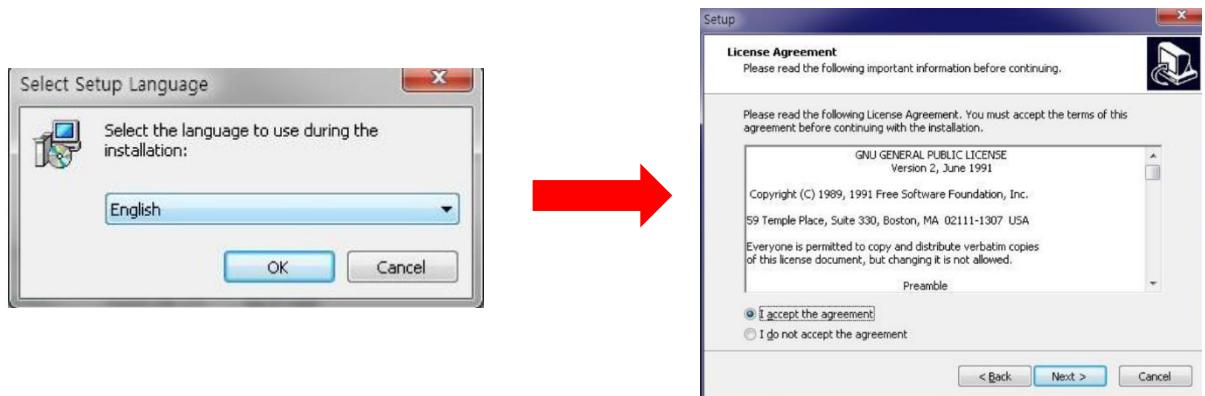
A screenshot of the FreeMind project's main page. The page title is "Main Page". On the left, there is a sidebar with a tree diagram of the site's structure and a navigation menu with links like Site Map, Main Page, Downloads, Asked questions, Screenshots, Documentation, Accessories, Scripting, Plugins, Linux, Development, Marketing, Essays, and Stuff. Below the sidebar is a search bar with "Go" and "Search" buttons. The main content area has tabs: page, discussion, view source, and history. The main heading is "FreeMind - free mind mapping software". The content describes FreeMind as a Java-based mind-mapping tool. A red box highlights the "Download" section, which contains a large "Download" button and a "See Download" link. At the bottom of the page, it says "Get yourself what more than 4 000 users download each day." followed by a "stats" link.

2. FREEMIND (계속)

2) OS 버전에 맞는 FreeMind를 다운로드 받아주세요.

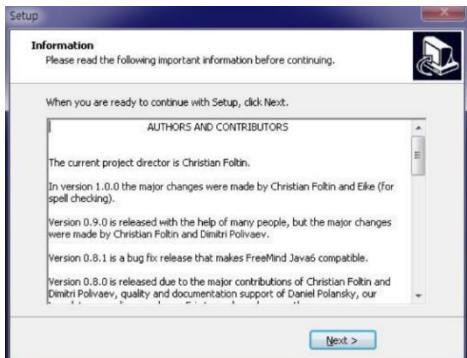
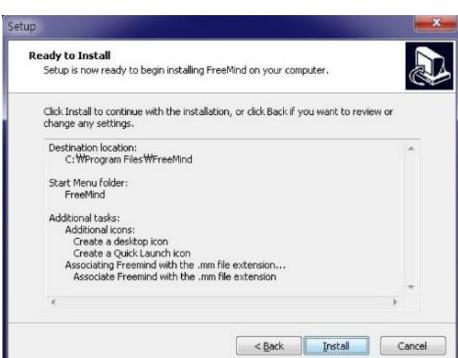
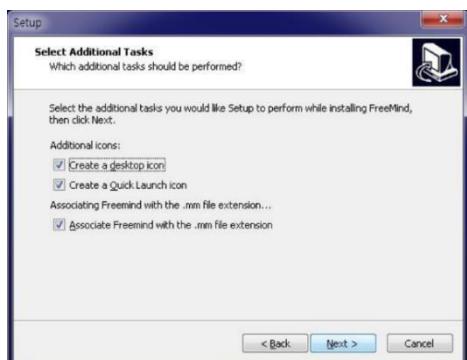
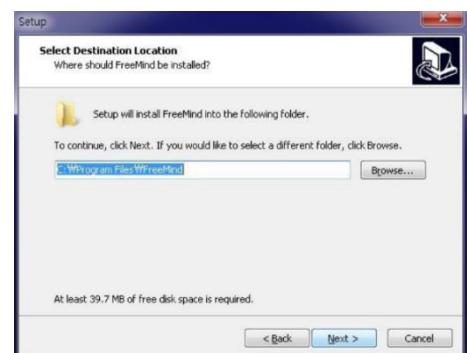
| OS | Download | Size (MB) | |
|-----------------|---|-----------|---|
| MS Windows | Windows Installer Min | 29 | A smaller version without SVG export and PDF export. |
| MS Windows | Windows Installer Max | 38 | All-inclusive version |
| Mac OS X | MAC OS X | 92 | All-inclusive version. Information for Mac Maverick (10.9) and reports the application to be broken. Please see here for the : |
| Mac OS X 10.6.8 | MAC OS X 10.6.8 | 150 | All-inclusive version, still version 1.0.0. Users of version 10.6.8, pl |
| Linux | Ubuntu Linux | ?? | Please consider first checking the release notes . |
| Linux | Debian Linux (currently still for the 0.9.0 version) | 2.6 | Please consider first checking the release notes . |
| Linux | SuSE 9.x Linux and other RPM-based distros like Fedora and Mandriva | 2.8 | See also release notes . |
| Any | binaries for any operating system | 29 | A smaller version without SVG export and PDF export. |
| Any | binaries for any operating system - max | 38 | All-inclusive version |
| eComStation | Download: WarpIN Installation package from netlabs.org | 7.6 | All-inclusive version |

3) 언어는 영어로 선택합니다. 영어로 설치해도 한국어로 작성이 가능합니다.



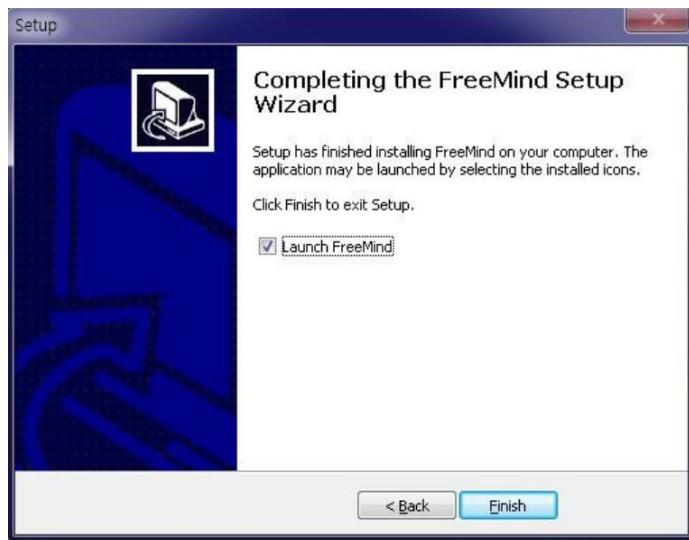
2. FREEMIND (계속)

4) Next를 눌러주세요.



2. FREEMIND (계속)

5)설치가 완료되었으면 freemind를 실행시켜 주세요.

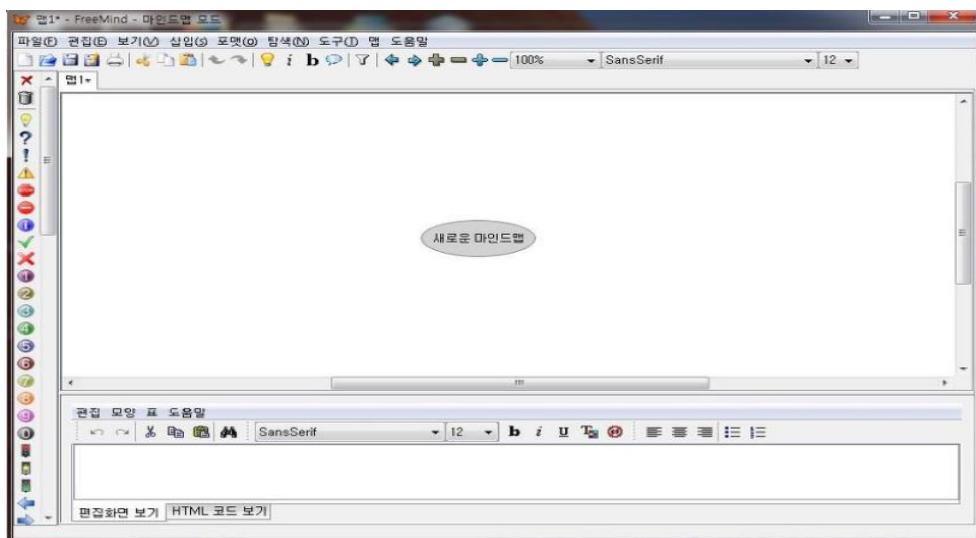


2. FREEMIND (계속)

주제어 설정

FreeMind를 열고 파일 > 새 파일 또는 Ctrl+N으로 새로운 맵의 작성을 시작합니다.

새 맵을 시작하면 "새로운 마인드맵"이라는 기본 주제어가 설정되는데 이 주제어를 클릭하여 주제어를 수정합니다.



2. FREEMIND (계속)

가지를 확장하고 핵심단어 설정

가지를 확장할 위치에서 Insert 키를 누르면 가지가 확장됩니다. 마우스로 가지를 확장할 노드를 마우스 우측 버튼을 클릭하여 팝업 메뉴>하위 노드로 가지를 확장할 수도 있습니다.

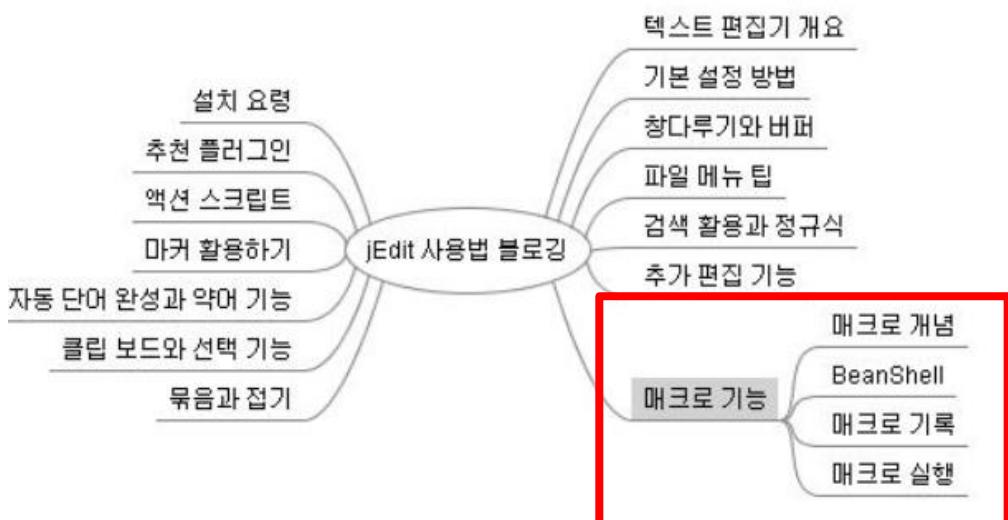
| | |
|------------------|--------------|
| 노드 수정 | F2 |
| 장문 편집창 열기 | Alt+Enter |
| Attributes 편집 | Alt+F9 |
| 노드 삭제 | Delete |
| 잘라내기 | Ctrl+X |
| 복사 | Ctrl+C |
| 선택한 노드만 복사 | Ctrl+Shift+C |
| 붙여넣기 | Ctrl+V |
| 하위 노드 | Insert |
| 위쪽에 같은 단계의 노드 추가 | Enter |
| 아래쪽에 단계의 노드 추가 | Shift+Enter |
| 접기(펴기) | Space |
| 아이콘(c) | ▶ |
| 내보내기(S) | ▶ |
| 포맷(O) | ▶ |
| 삽입(I) | ▶ |
| 준비된 스타일 적용(S) | ▶ |



2. FREEMIND (계속)

핵심단어에 부가지를 만들고 연관단어 작성

부가지를 확장하고 연관 단어를 작성하는 과정은 주제어에서 핵심 단어를 확장하는 과정과 동일합니다.



2. FREEMIND (계속)

노드에 특성을 부여하거나 강조

다음과 같이 이모티콘을 사용하여 노드의 특성을 부여하거나 강조 할 수 있습니다.



2. FREEMIND (계속)

추가적인 단축 기능들

새 마인드 맵 -

Ctrl+N이나 파일>새파일을 선택하면 "새로운 마인드맵"으로 시작할 수 있습니다.

노드 선택 및 수정 -

노드를 클릭하면 선택하여 수정할 수 있고 F2를 클릭해도 됩니다.

하위 노드 추가 -

선택 노드에서 Insert 키를 누르면 하위 노드를 추가합니다.

같은 레벨 추가 -

선택 노드에서 Enter 키를 누르면 같은 레벨로 노드를 추가합니다.

노드간 위치 바꾸기 -

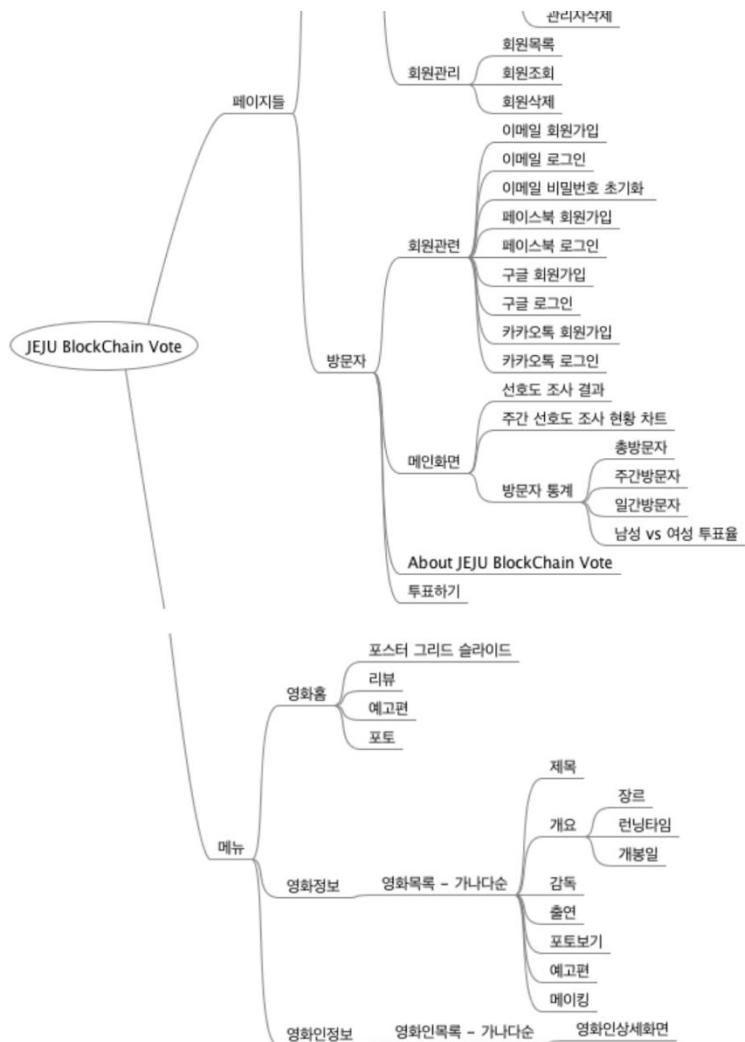
Ctrl+상하좌우 방향키로 노드 위치를 바꿀 수 있고, 마우스로 끌어다 놓기 해도 됩니다.

맵 저장 -

Ctrl+S 또는 파일>저장으로 파일을 저장할 수 있습니다.

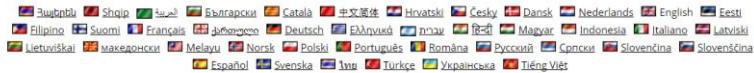
2. FREEMIND (계속)

밑의 마인드맵은 기획을 할 때 생각 할 만한 기능에 대한 예시입니다. 다음 기능 중 어떤 기능이 자신이 만들고자 하는 웹사이트에 필요한지를 생각하며 나만의 기획을 해보도록 합시다.



TUTORIAL 3 로렘 입숨

가끔 사이트의 프로토 타입 등을 만들거나 골격을 먼저 만든 후 추후에 내용을 채워넣어야 할 경우 의미 없는 문구를 골격에 넣어 사이트의 예시를 조금 더 실제와 비슷하게 보여줘야 하는 경우가 있습니다. 이 때 사용하는 것이 로렘 입숨으로 영문 버전과 한글 버전이 있으니 용도에 맞게 사용하시면 됩니다.



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..."

What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

한글입숨

무의미한 한글 텍스트 생성기입니다.

왜 사용하나요?

| 문단 수 | 문단 길이 | 텍스트 소스 |
|-----------------------|-------|--------|
| 3 | 길게 | 청춘예찬 |
| <button>생성하기</button> | | |

[Tweet](#) [G+](#) [Like](#) 좋아요 5개

Powered by [Thefron](#)

TUTORIAL 4 GIT

컴퓨터로 작업을 계속 하다보면 누구나 파일백업의 필요성을 느낍니다.

그래서 새로운 파일로 복사하고 뒤에 날짜와 특정단어를 붙여서 파일명을 지정합니다.

하지만 소스코드와 마찬가지로 파일 갯수가 많아지고 변경하는 횟수가 많아지면 백업과정이 점점 복잡해지게 됩니다. 또한 내가 변경한 내용에 대한 추적도 쉽지 않게 됩니다.

Git과 같은 형상관리도구를 사용하면 쉽게 백업을 할 수 있으며 특정버전으로 전환도 쉽게 됩니다.



프로젝트 결과 보고서_1126



프로젝트 결과 보고서_1126_수정



프로젝트 결과 보고서_1126_수정2



프로젝트 결과 보고서_최종



프로젝트 결과 보고서_최종_보고용



프로젝트 결과 보고서_최종_보고용_1127 수정



프로젝트 결과 보고서_최종_보고용_1127 최종



프로젝트 결과 보고서_최종_보고용_1127 최종_진짜진짜진짜 최종



프로젝트 결과 보고서_최종_보고용_1127 최종_진짜진짜최종

4. GIT (계속)

그렇다면 GIT을 사용하면 어떤 장점이 있을까요?

- 폴더내에 백업본을 지속적으로 만들 수 있습니다.
- 언제든지 원하는 백업본으로 내 폴더를 변경할 수 있습니다.
- 전체백업본을 서버에 업로드 할 수 있습니다.
- 또한 다른 사람이 업로드한 전체백업본을 쉽게 다운 받을 수 있습니다.
- 같은 소스코드를 여러 사람이 각자 작업하고 고통없이 합칠 수 있습니다.

따라서 팀프로젝트에 매우 유용한 도구입니다.



프로젝트 결과 보고서_1126



프로젝트 결과 보고서_1126_수정



프로젝트 결과 보고서_1126_수정2



프로젝트 결과 보고서_최종



프로젝트 결과 보고서_최종_보고용



프로젝트 결과 보고서_최종_보고용_1127 수정



프로젝트 결과 보고서_최종_보고용_1127 최종



프로젝트 결과 보고서_최종_보고용_1127 최종_진짜진짜진짜 최종



프로젝트 결과 보고서_최종_보고용_1127 최종_진짜진짜최종

4. GIT (계속)

Windows용 Git 다운로드 사이트 : <https://git-scm.com/download/win>

다운로드 사이트에서 파일을 다운로드 받아서 설치를 진행하면 됩니다.

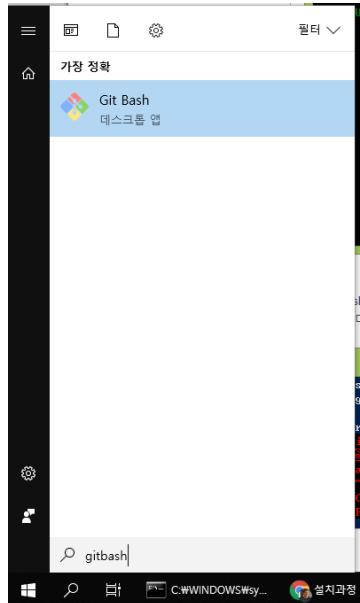


4. GIT (계속)

다음으로 Gitbash프로그램을 실행해보도록하겠습니다.

윈도우키 - gitbash검색 - 실행

맥에서는 터미널을 실행합니다.



다음으로 Gitbash프로그램을 실행해보도록하겠습니다.

윈도우키 - gitbash검색 - 실행

맥에서는 터미널을 실행합니다.

4. GIT (계속)

Git 버전확인명령을 입력하여 설치가 잘 되어있는지 확인합니다.

```
$ git --version
```



```
MINGW64:/c/Users/leebe
teebe@DESKTOP-NUI83BP MINGW64 ~
$ git --version
git version 2.12.0.windows.1
teebe@DESKTOP-NUI83BP MINGW64 ~
$
```

이 다음으로는 Git 기본설정을 해보도록 하겠습니다.

Git을 처음 설치한 뒤에 사용자정보를 등록해주어야 합니다.

Git을 통해 파일버전을 생성 할 때마다 이 사용자정보가 같이 등록되니 잘 등록하는 것이 좋습니다.

```
$ git config --global user.name "사용자 이름"
```

```
$ git config --global user.email "사용자의 이메일 주소"
```



```
MINGW64:/c/Users/leebe
teebe@DESKTOP-NUI83BP MINGW64 ~
$ git config --global user.name "Beomjae Lee"
teebe@DESKTOP-NUI83BP MINGW64 ~
$ git config --global user.email "beomjae@gmail.com"
teebe@DESKTOP-NUI83BP MINGW64 ~
$
```

4. GIT (계속)

처음 git을 설치하고 내 디렉토리에 초기화 하면 디렉토리 내부에 .git 폴더가 생성이 됩니다.

이후에 저장하는 모든 백업들이 이 폴더안에 생성이 됩니다.

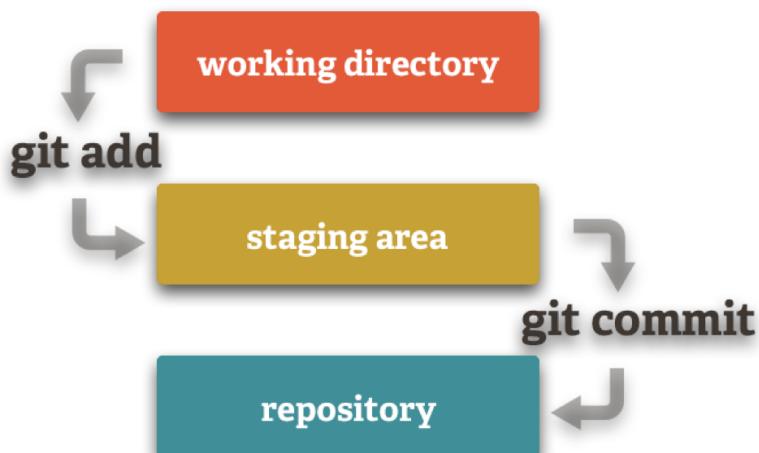
Git에서는 백업을 하는것을 '커밋' 이라고 부릅니다.

내가 작업중인 a.txt 파일을 커밋하려면 다음의 순서를 거칩니다.

- 1) a.txt파일 생성
- 2) add 명령을 통해 a.txt파일을 커밋대상으로 준비목록에 추가 시킵니다.
- 3) commit 명령을 통해 준비목록에 담아놓은 변경사항파일들을 커밋합니다.

명령어

```
$ git init  
$ touch a.txt  
$ git add a.txt  
$ git commit -m "first commit"
```



4. GIT (계속)

간단한 카페목록페이지를 만들면서 Git을 실습해보도록 하겠습니다.

1. 작업을 위한 카페디렉토리를 생성합니다.

```
$ mkdir cafe
```

```
$ cd cafe
```

2. git init 명령을 이용하여 디렉토리에서 git을 사용할 수 있도록 초기화 합니다.

```
$ git init
```

3. 작업을 위해 atom 에디터를 실행합니다.

```
$ atom .
```

4. 아톰에디터상에서 café.html 파일을 생성합니다.

아톰에디터 메뉴 - 파일 - 새파일 - cafe.html

5. 다음 내용을 입력하고 저장합니다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>제주카페</title>
</head>
<body>
<h1>제주카페</h1>
<ul>
<li>에이팩토리 커피앤톡스</li>
<li>아일랜드팩토리 풍류</li>
<li>에스프레소 라운지</li>
<li>애프터글로우</li>
<li>941Hz</li>
<li>날것</li>
<li>카페 이호동</li>
</ul>
</body>
</html>
```

4. GIT (계속)

6. 터미널에서 git add 명령을 이용해 파일을 커밋하기 위한 목록에 추가합니다.

```
$ git add .
```

7. 준비한 파일을 커밋메세지와 함께 커밋합니다.

```
$ git commit -m "start cafe project"
```

8. git log 명령을 이용해서 커밋기록을 조회해봅니다.

```
$ git log
```

9. cafe.html 파일에 내용을 추가합니다.

```
...
<li>카페 이호동</li>
<li>갤러리카페 다리</li>
<li>베이스노트</li>
<li>브와두스베이커리카페</li>
<li>소설다방</li>
<li>엘가커피 제주연동점</li>
<li>커피생각할때</li>
...
</ul>
</body>
</html>
```

10. git status 명령을 이용해 변경상태를 조회합니다.

```
$ git status
```

11. 변경한 내용을 다시 커밋합니다.

```
$ git add .
```

```
$ git commit -m "add more cafe"
```

4. GIT (계속)

이번에는 작업중에 파일을 원래 이전에 커밋한 내용으로 복구해보겠습니다.

12. café.html 파일에 아무 내용이나 추가 또는 수정하고 저장합니다.

```
...
<li>AAAAAAA</li>
<li>AAAAA</li>
<li>베이스노트</li>
...
</ul>
</body>
</html>
```

13. git status 명령을 이용해 변경된 내용을 확인합니다.

```
$ git status
```

14. git reset 명령을 이용해 파일을 이전에 커밋했던 상태로 되돌립니다.

```
$ git reset --hard
```

4. GIT (계속)

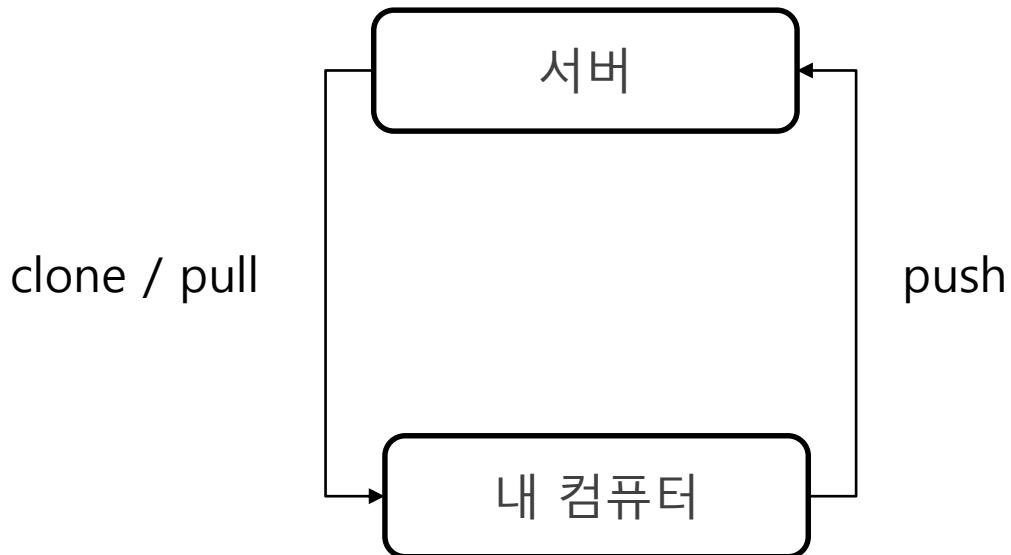
다음에는 Git 서버에 대해 배워보도록 합시다.

git은 커밋내역들을 내 컴퓨터 뿐만 아니라 git 서버에도 업로드하고 다운로드할 수 있는 기능이 있습니다.

프로젝트를 통째로 다운로드 받는것이 **clone** 입니다.

클론받은 곳에서 추가적으로 다운로드하는것이 **pull** 입니다.

내 컴퓨터에서 추가 커밋한 내용을 업로드 하는것이 **push** 입니다.



git서버의 기능을 제공하는 대표적인 서비스 사이트로는 github, bitbucket, gitlab 이 있습니다. 실습에서는 github를 사용하겠습니다.

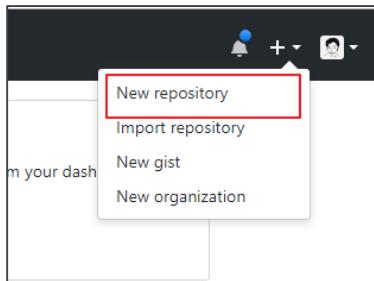
4. GIT (계속)

다음에는 github 사이트를 이용해 보도록 하겠습니다.

1. github 사이트에 가입을 합니다.

<https://github.com/>

2. 새 리포지토리(저장공간)을 생성합니다.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: beomjae / Repository name: hellocafe

Great repository names are short and memorable. Need inspiration? How about [redesigned-meme](#).

Description (optional): git서버 사용 실습

Visibility: Public (radio button) / Private (radio button, selected). Anyone can see this repository. You choose who can commit.

Initialize this repository with a README: This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None | ⓘ

Create repository

The screenshot shows the 'Create a new repository' form on GitHub. It includes fields for the owner (beomjae), repository name (hellocafe), a description (git서버 사용 실습), and visibility settings (Public or Private). The 'Private' option is selected and highlighted with a red box. At the bottom, there is a large green 'Create repository' button also highlighted with a red box.

4. GIT (계속)

3. Create repository를 클릭합니다
4. Quick setup화면에서 https를 클릭합니다.
5. 두번째단락 (…or push an existing repository from the command line) 안에 있는 코드를 복사합니다.

The screenshot shows the GitHub repository creation interface for a private repository named 'beomjae / hellocafe'. The 'Code' tab is selected. The 'Quick setup' section is highlighted with a red box around the 'HTTPS' button. Below it, a code block shows the command to initialize a local repository and push it to GitHub. A second red box highlights the command 'git push -u origin master'. At the bottom, a pro tip suggests using the URL for adding GitHub as a remote.

beomjae / hellocafe Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or **HTTPS** SSH <https://github.com/beomjae/hellocafe.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

...or create a new repository on the command line

```
echo "# hellocafe" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin https://github.com/beomjae/hellocafe.git  
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/beomjae/hellocafe.git  
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

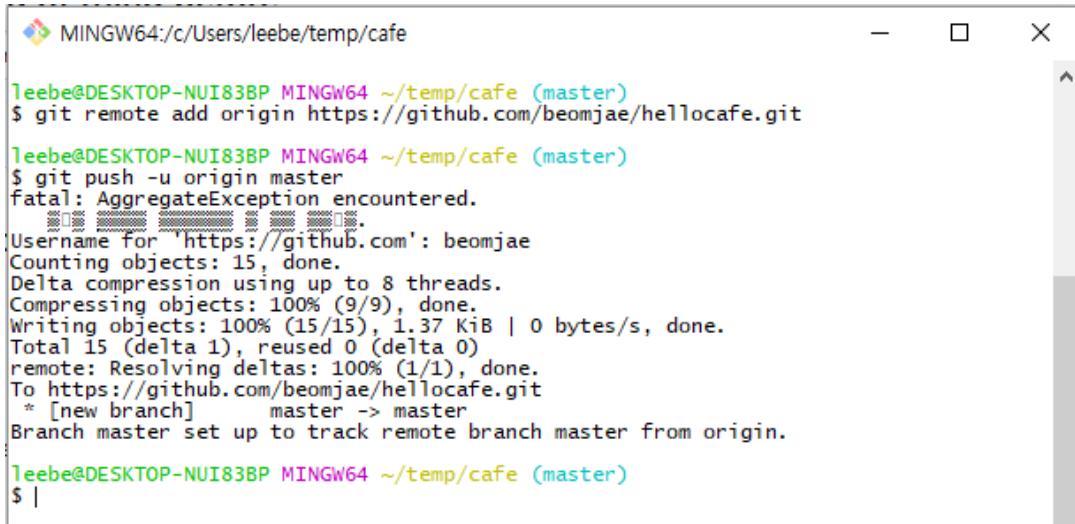
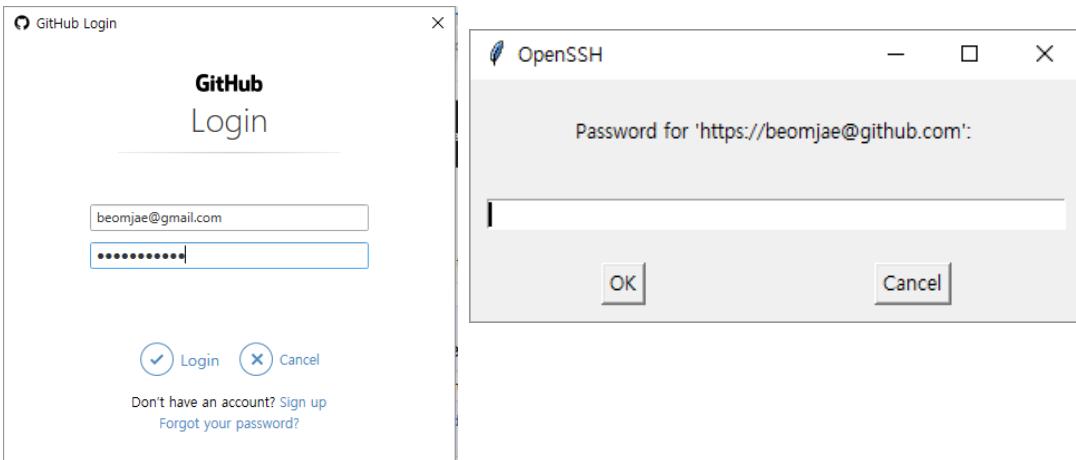
ProTip! Use the URL for this page when adding GitHub as a remote.

4. GIT (계속)

6. 터미널에 한줄씩 붙여넣기 합니다. 중간에 ID와 암호를 입력합니다.



```
MINGW64:/c/Users/leebe/temp/cafe
teebe@DESKTOP-NUI83BP MINGW64 ~/temp/cafe (master)
$ git remote add origin https://github.com/beomjae/hellocafe.git
teebe@DESKTOP-NUI83BP MINGW64 ~/temp/cafe (master)
$ git push -u origin master
```



```
MINGW64:/c/Users/leebe/temp/cafe
teebe@DESKTOP-NUI83BP MINGW64 ~/temp/cafe (master)
$ git remote add origin https://github.com/beomjae/hellocafe.git
teebe@DESKTOP-NUI83BP MINGW64 ~/temp/cafe (master)
$ git push -u origin master
fatal: AggregateException encountered.
Username for 'https://github.com': beomjae
Counting objects: 15, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (15/15), 1.37 KiB | 0 bytes/s, done.
Total 15 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/beomjae/hellocafe.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
teebe@DESKTOP-NUI83BP MINGW64 ~/temp/cafe (master)
$ |
```

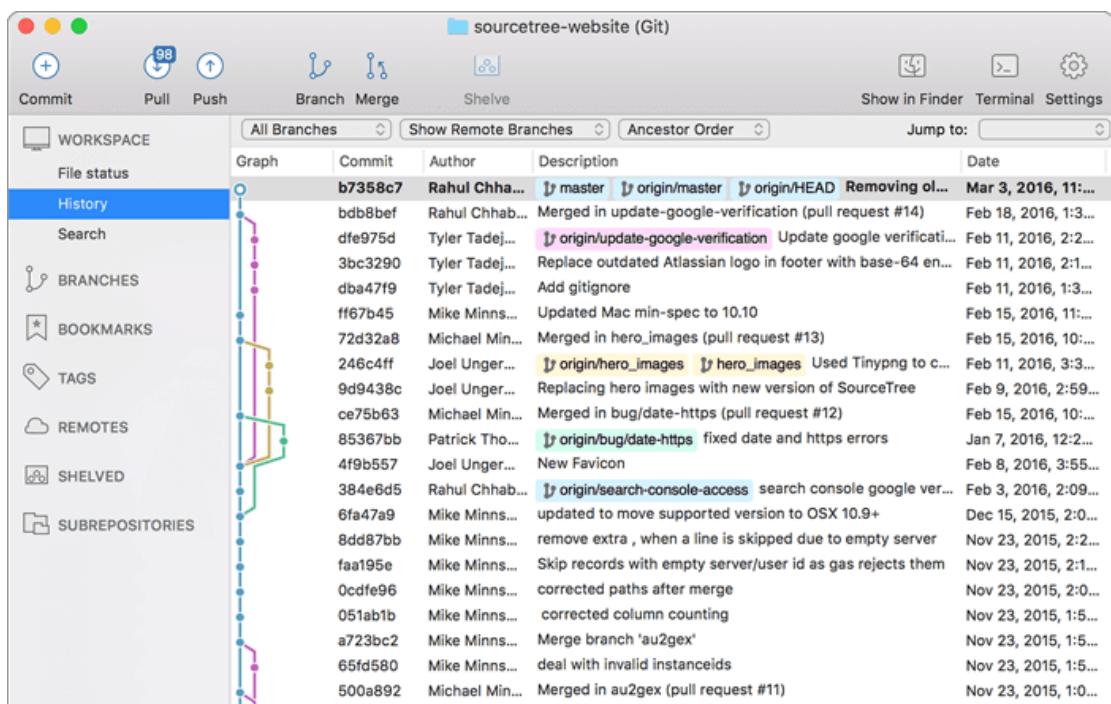
4. GIT (계속)

7. github 사이트의 hellocafe프로젝트에 다시 들어가보면 서버에 업로드된 내역을 확인 할 수 있습니다.

The screenshot shows the GitHub repository page for 'beomjae / hellocafe'. The repository is private, indicated by a lock icon. It has 1 watch, 0 stars, and 0 forks. The main navigation tabs are 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Insights', and 'Settings'. Below the tabs, there's a section titled 'git서버 사용 실습' with an 'Edit' button and a 'Manage topics' link. A summary bar shows '4 commits', '1 branch', and '0 releases'. The commit history lists three commits: 'beomjae add more cafe' (latest), 'cafe add more cafe', and 'a.txt a'. At the bottom, there's a note to 'Add a README with an overview of your project.' and a green 'Add a README' button. The footer includes links to GitHub's terms, privacy, security, status, help, contact, pricing, API, training, blog, and about pages.

4. GIT (계속)

많은 사람들이 GIT을 사용하고 있기 때문에 GIT을 조금 더 효율적으로 사용할 수 있도록 다양한 솔루션들이 나와 있습니다. 예를 들어, SOURCE TREE는 GIT는 간단한 인터페이스를 제공하여 효율적으로 자신의 리포지토리를 관리하게 해줍니다.



4일차

오전

Django

Django 목차

Step 1. 구름IDE 환경설정

1.1 Django설치 및 세팅 환경소개

1.2 환경설정

Step 2. Django로 Main page 만들기

Step 3. Django로 Cafelist page 만들기

Step 4. Django로 Cafe details page 만들기

Step 5. 댓글과 태그 기능 만들기

5.1 댓글 기능 만들기

5.2 태그 기능 만들기

Step 6. Django에 Template 입히기

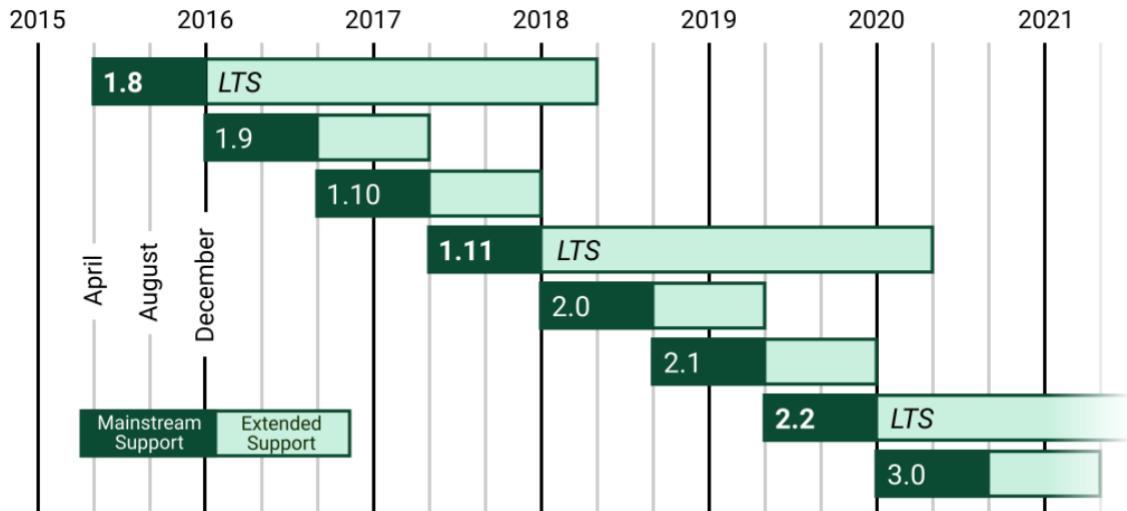
6.1 Django에 Template 입히기

6.2 서비스 배포

Step 7. 요약정리

Version Roadmap

Django 업데이트 로드맵입니다.



| Release Series | Latest Release | End of mainstream support ¹ | End of extended support ² |
|-----------------------|----------------|--|--------------------------------------|
| 2.1 | 2.1 | April 2019 | December 2019 |
| 2.0 | 2.0.8 | August 1, 2018 | April 2019 |
| 1.11 LTS ³ | 1.11.15 | December 2, 2017 | Until at least April 2020 |
| 1.10 | 1.10.8 | April 4, 2017 | December 2, 2017 |
| 1.9 | 1.9.13 | August 1, 2016 | April 4, 2017 |
| 1.8 LTS | 1.8.19 | December 1, 2015 | April 1, 2018 |
| 1.7 | 1.7.11 | April 1, 2015 | December 1, 2015 |
| 1.6 | 1.6.11 | September 2, 2014 | April 1, 2015 |
| 1.5 | 1.5.12 | November 6, 2013 | September 2, 2014 |
| 1.4 LTS | 1.4.22 | February 26, 2013 | October 1, 2015 |
| 1.3 | 1.3.7 | March 23, 2012 | February 26, 2013 |

<https://www.djangoproject.com/download/>

Step 1

구름IDE 환경설정

1. Django 설치 및 세팅 환경소개

Django는 파이썬 Web Framework 중 가장 사랑받는 Full-Stack Framework로 인스타그램, NASA, 우리가 사용할 댓글 관리 서비스인 Disqus 등에서 사용하고 있습니다.

Full-Stack Framework는 웹 서비스 개발에 필요한 모든 요소들이 한곳에 모여있는 종합선물 세트라고 생각하시면 편합니다. 보다 빠르고 편리하게 웹 서비스를 개발할 수 있도록 필요한 공구들을 모은 것이죠.

Django는 2005년 오픈소스로 시작되어 2018년 8월 Django 2.1이 나왔으며 아직 한국에는 2.1 관련된 책이 없습니다.

책에는 상세한 내용을 다루기보다 전체적인 맥락을 잡기 위해 한 사이클을 도는 것에 초점이 맞춰져 있으니 보다 자세한 내용은 공식 홈페이지에 문서를 참고하시면 좋을 것 같아요.

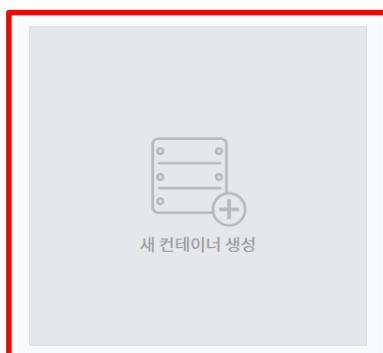
<https://www.djangoproject.com/>

공식 홈페이지에는 django에 대한 소개와 다운로드, 지원 문서를 제공합니다. 이 지원문서에 django에 대한 내용이 다 담겨 있으니 개발할 때 꼭 참고하시면서 개발하세요.

- * 1870 page에 방대한 양을 pdf로도 제공합니다.
 - <https://media.readthedocs.org/pdf/django/2.1.x/django.pdf>

2. 환경설정(계속)

앞서 컨테이너를 만들었지만 여기서 새 컨테이너를 생성하여 진행하도록 하겠습니다. 새 컨테이너 생성을 누르셔서 컨테이너를 생성해 주세요.



컨테이너 생성

The screenshot shows a configuration page for creating a new container. At the top, there are options for source (Template selected), name (tutorialdjango), and description (여행블로그 튜토리얼로 배우는 Django). The 'Software Stack Selection' section is expanded, showing various Python-related options like Django, Flask, Jupyter Notebook, TensorFlow, PyQt, Java, and Maven. A dropdown menu for 'Python Project Type' is open, with 'Python' selected. To the right, a list of system requirements is shown with checkmarks: OS (Ubuntu 14.04 LTS), Python3 (3.4.3), Python (2.7.6), pip3 (9.0.1), pip (9.0.1), Jupyter (4.3.0), Django (1.11.12 LTS), Flask (0.12), and TensorFlow (1.3.0). At the bottom, there are '취소' (Cancel) and '생성하기' (Create) buttons, with '생성하기' being highlighted by a red box.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------|---|--------|-------|---------------|--------|-------|------------------|------------|------|------|-------|----|------------------|---------|-------|--------|-------|------|-------|-----|-------|---------|-------|--------|-------------|-------|------|------------|-------|
| 소스 | <input checked="" type="radio"/> 템플릿 <input type="radio"/> Github <input type="radio"/> Bitbucket <input type="radio"/> Git / SVN <input type="radio"/> 암축파일 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 이름 | tutorialdjango | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 설명 | 여행블로그 튜토리얼로 배우는 Django | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 소프트웨어 스택 선택 | <table border="1"><tr><td>Search</td></tr><tr><td>C/C++</td></tr><tr><td>Python</td></tr><tr><td>Django</td></tr><tr><td>Flask</td></tr><tr><td>Jupyter Notebook</td></tr><tr><td>TensorFlow</td></tr><tr><td>PyQt</td></tr><tr><td>Java</td></tr><tr><td>Maven</td></tr></table> <p>프로젝트 유형</p> <p>Python 프로젝트</p> <table border="1"><tr><td>OS</td><td>Ubuntu 14.04 LTS</td></tr><tr><td>Python3</td><td>3.4.3</td></tr><tr><td>Python</td><td>2.7.6</td></tr><tr><td>pip3</td><td>9.0.1</td></tr><tr><td>pip</td><td>9.0.1</td></tr><tr><td>Jupyter</td><td>4.3.0</td></tr><tr><td>Django</td><td>1.11.12 LTS</td></tr><tr><td>Flask</td><td>0.12</td></tr><tr><td>TensorFlow</td><td>1.3.0</td></tr></table> | Search | C/C++ | Python | Django | Flask | Jupyter Notebook | TensorFlow | PyQt | Java | Maven | OS | Ubuntu 14.04 LTS | Python3 | 3.4.3 | Python | 2.7.6 | pip3 | 9.0.1 | pip | 9.0.1 | Jupyter | 4.3.0 | Django | 1.11.12 LTS | Flask | 0.12 | TensorFlow | 1.3.0 |
| Search | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C/C++ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter Notebook | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PyQt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Java | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Maven | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OS | Ubuntu 14.04 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python3 | 3.4.3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Python | 2.7.6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip3 | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| pip | 9.0.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Jupyter | 4.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Django | 1.11.12 LTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flask | 0.12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TensorFlow | 1.3.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

취소

생성하기

2. 환경설정(계속)

컨테이너가 만들어지고 있다는 화면이 나오고 곧 컨테이너 생성이 완료되었다고 뜹니다. 컨테이너 실행을 누르셔서 컨테이너 안으로 들어가주세요. 만약 모르고 닫으셨다면 대시보드에서 해당 컨테이너 실행을 누르시면 됩니다.

The image consists of three screenshots from a Docker interface:

- Screenshot 1:** A progress bar showing 14% completion. Below it, a message says "곧 멋진 작업공간이 생성됩니다. 잠시만 기다려주세요." (A nice workspace will be created soon. Please wait a moment.)
- Screenshot 2:** A confirmation dialog titled "컨테이너 실행" (Container Execution) asking "컨테이너를 실행하시겠습니까?" (Would you like to run the container?). It has two buttons: "대시보드로 이동" (Move to Dashboard) and a redboxed "실행" (Run) button.
- Screenshot 3:** A container details page for "tutorialdjango". It shows the last update was "2018. 4. 30. Updated". It lists the following specifications:
 - 소프트웨어 스택: Python
 - 저장 공간: 10GB
 - 항상 켜두기: PREMIUM 상품 구매 시 사용 가능
 - SSH: 컨테이너 실행 시 사용 가능A red box highlights the "실행" (Run) button at the bottom of the page.

2. 환경설정(계속)

가장 먼저 해야 할 것은 홈페이지 기획 및 구성입니다. 전체 URL 구조를 살펴보도록 하겠습니다.

1. https://사용자_지정_도메인.run.goorm.io/

- 메인페이지
- Template : index.html

2. https://사용자_지정_도메인.run.goorm.io/about

- 홈페이지 소개 페이지
- Template : about.html

3. https://사용자_지정_도메인.run.goorm.io/cafelist

- 메인페이지에서 카페들을 클릭했을 경우 이동
- Template : cafelist.html

4. https://사용자_지정_도메인.run.goorm.io/cafelist/cafe번호

- cafelist 페이지에서 카페 상세 정보 보기 를 눌렀을 때 이동, google map API 사용
- Template : cafedetails.html

5. https://사용자_지정_도메인.run.goorm.io/cafeenroll

- 메인페이지에서 카페등록을 클릭했을 경우 이동
- Template : enroll.html

2. 환경설정(계속)

사용자가 처음 볼 main page 입니다. 여기서 사용자가 원하는 지역을 선택하고 카페 찾기를 누르면 해당 지역의 카페가 찾아지는 구조로 되어 있습니다.



지도에서 원하는 지역을 선택하세요!
선택하지 않으면 전체 지역에서 찾습니다



카페찾기!

2018 © 제주카페찾기

2. 환경설정(계속)

about page 입니다. 안에 글귀는 의미 없는 로렘 입술으로 작성되어 있습니다. 여러분의 생각과 입맛대로 작성해보세요!



제주카페찾기

제주도 카페, 여기 다 있어 요!

섹션 제목

*Lorem ipsum dolor sit amet, consectetur adipisicing elit.
 Dolorem cupiditate molestias vel veritatis odio repellat quod
 velit, at delectus dolor tenetur, laudantium atque, inventore.
 Quia obcaecati accusantium consequuntur dignissimos fugit.
 제주코딩베이스캠프에서 제작하는 예제 웹사이트입니다.*

2. 환경설정(계속)

cafelist page 입니다. 해당 사진이나 글귀를 눌렀을 경우 상세 페이지로 이동합니다.

제주카페찾기

서비스 소개 카페들 + 카페등록

찾고 싶은 카페 이름

Q



제주코딩카페

개발자와 디자이너들을 위한 카페예요.

• 제주시 조천읍 함덕00길 000
• 064-000-0000
• @jejucodingcafe



CAFE 마트료시카

러시아 인형 장식이 인상적인 러시아풍 카페예요.
• 제주시 조천읍 함덕00길 000
• 064-000-0000
• @jejucodingcafe

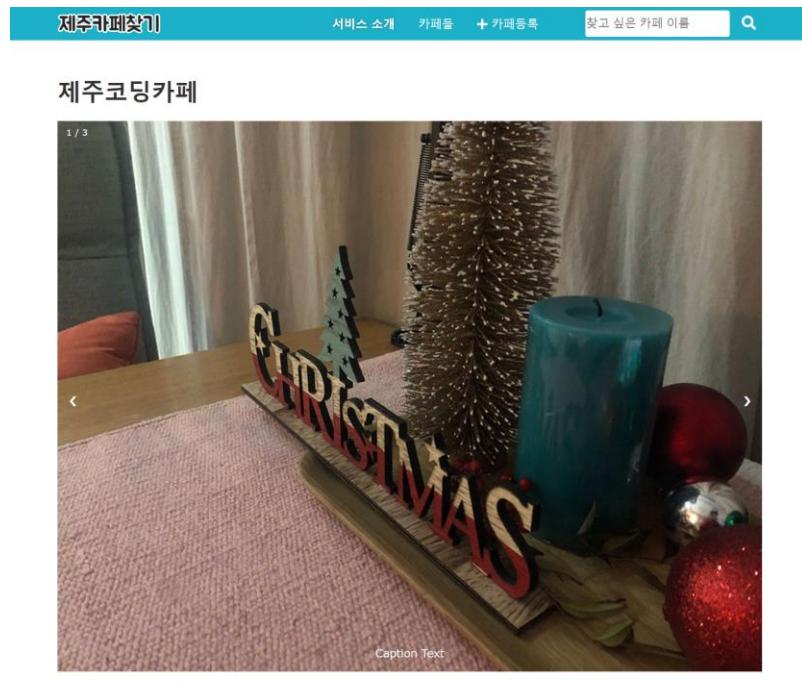


크리스마스 카페

크리스마스에만 문을 여는 카페입니다.
• 제주시 조천읍 함덕00길 000
• 064-000-0000
• @jejucodingcafe

2. 환경설정(계속)

cafedetails page 입니다. 카페의 사진 3장, 설명, 주소와 위치를 나타내고 google MAP API를 활용하여 지도에 위치를 나타내줍니다.



개발자와 디자이너들을 위한 카페예요.

📍 제주시 조천읍 험목00길 000
📞 064-000-0000
👤 @jejucodingcafe

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolorem cupiditate molestias vel veritatis odio repellat quod velit, at delectus dolor tenetur, laudantium atque, inventore. Quia obcaecati accusantium consequuntur dignissimos fugit.
제주코딩베이스캠프에서 제작하는 예제 웹사이트입니다.



2018 © 제주카페찾기

2. 환경설정(계속)

cafe 등록 페이지입니다. 해당 페이지는 admin 페이지에서 실행할 수 있기 때문에 없어도 되는 페이지이지만 form 연습 차원에서 튜토리얼 안에 넣어놨습니다.

The screenshot shows a registration form for a cafe on a website. The top navigation bar includes links for '서비스 소개' (Service Introduction), '카페들' (Cafes), '+ 카페등록' (Register Cafe), a search bar with placeholder '찾고 싶은 카페 이름' (Name of cafe you want to find), and a magnifying glass icon.

카페등록

카페 위치: 구좌읍, 남원읍, 대정읍, 서귀포시내, 성산읍, 안덕면, 애월읍, 우도면, 제주시내, 조천읍, 표선면, 한경면, 한림읍

카페이름: [Input field]

주소: [Input field]

전화번호: [Input field]

카페소개: [Input field]

사진: 파일 선택 (Selected file: 없음)

등록하기

2. 환경설정(계속)

본격적인 화면 구성입니다. 해당 명령은 4번 콘솔창에서 입력해주세요.

```
root@goorm /workspace/tutorialdjango# mkdir mysite
```

```
root@goorm /workspace/tutorialdjango # cd mysite
```

```
root@goorm /workspace/tutorialdjango/mysite# pip3 install virtualenv
```

```
root@goorm /workspace/tutorialdjango/mysite# virtualenv myvenv
```

```
root@goorm /workspace/ tutorialdjango /mysite# source myvenv/bin/activate
```

```
(myvenv)root@goorm /workspace/ tutorialdjango /mysite#
```

```
pip3 install django==2.1
```

```
(myvenv)root@goorm /workspace/ tutorialdjango /mysite#
```

```
django-admin startproject tutorialdjango .
```

```
(myvenv)root@goorm /workspace/ tutorialdjango /mysite#
```

```
python manage.py migrate
```

'startproject 컨테이너이름' 뒤에 '.'이 있으니 주의해 주세요.

2. 환경설정(계속)

```
root@goorm:/workspace/tutorialdjango# mkdir mysite
root@goorm:/workspace/tutorialdjango# cd mysite
root@goorm:/workspace/tutorialdjango/mysite# pip3 install virtualenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/b6/30/96a02b2287098b23b875
2efe84f747b64d523721dc2b5/virtualenv-16.0.0-py2.py3-none-any.whl (1.9MB)
    100% |████████████████████████████████| 1.9MB 469kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-16.0.0
You are using pip version 9.0.3, however version 18.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
root@goorm:/workspace/tutorialdjango/mysite# virtualenv myvenv
Using base prefix '/usr/local'
New python executable in /workspace/tutorialdjango/mysite/myvenv/bin/python3.6
Also creating executable in /workspace/tutorialdjango/mysite/myvenv/bin/python
Installing setuptools, pip, wheel...done.
root@goorm:/workspace/tutorialdjango/mysite# source myvenv/bin/activate
```

```
(myvenv) root@goorm:/workspace/tutorialdjango/mysite# pip3 install django==2.1
Collecting django==2.1
  Downloading https://files.pythonhosted.org/packages/51/1a/e0ac7886c7123a03814178d7517dc822af0fe
03322/Django-2.1-py3-none-any.whl (7.3MB)
    100% |████████████████████████████████| 7.3MB 5.3MB/s
Collecting pytz (from django==2.1)
  Downloading https://files.pythonhosted.org/packages/30/4e/27c34b62430286c6d59177a0842ed90dc789c
8779a/pytz-2018.5-py2.py3-none-any.whl (510kB)
    100% |████████████████████████████████| 512kB 6.4MB/s
Installing collected packages: pytz, django
Successfully installed django-2.1 pytz-2018.5
(myvenv) root@goorm:/workspace/tutorialdjango/mysite# django-admin startproject tutorialdjango .
```

- Please be careful ‘.’ is followed by ‘startproject container name’

```
(myvenv) root@goorm:/workspace/tutorialdjango/mysite# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

2. 환경설정(계속)

사용하셨던 명령어를 해석하는 부분입니다.

```
root@goorm:/workspace/컨테이너명# mkdir mysite
```

mkdir은 디렉토리를 만들어주는 명령어 입니다. mysite라는 폴더를 만들 것입니다.

```
root@goorm:/workspace/컨테이너명# cd mysite
```

cd명령어는 change directory 명령어 입니다. 우리가 앞서 만든 mysite라는 폴더로 이동합니다.

```
root@goorm:/workspace/컨테이너명/mysite# pip3 install virtualenv
```

가상환경 설정을 위한 명령어 입니다. 가상환경 설정은 프로젝트 관리를 편하게 해주고 버전별 충돌을 막아줍니다.

```
root@goorm:/workspace/컨테이너명/mysite# # virtualenv myvenv
```

가상환경을 설정하겠다는 명령입니다.

```
root@goorm:/workspace/컨테이너명/mysite# source myvenv/bin/activate
```

이 명령어를 실행하고 앞서 실행한 가상환경으로 들어가겠다는 명령어 입니다.

이 명령어는 구름IDE 컨테이너를 다시 실행할때마다 쳐주셔야 합니다.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite# pip3 install django==2.1
```

패키지까지 설치하기 위한 명령어 입니다.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite#
```

django-admin startproject 컨테이너이름 .

현재 있는 폴더에 프로젝트를 ‘컨테이너이름’으로 만들어 시작하겠다는 명령어예요.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite# python manage.py migrate
```

Migrate 명령어는 하편에서 설명해 드리도록 하겠습니다. 쉽게 말해 DB에 값을 넣는 작업입니다.

2. 환경설정(계속)

(TensorFlow를 사용하실 경우의 명령어입니다.
Django만 사용하실 분은 위에 페이지를 참고해주세요.)

다른 환경에서 설치하실 경우 <https://www.tensorflow.org/install/> 홈페이지를 참고하여 버전확인을 꼭 하세요.(우분투나 파이썬 버전에 따라서 TensorFlow가 작동하지 않을 수 있습니다.)

```
root@goorm:/workspace/여러분ID# export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.12.1-cp34-cp34m-linux_x86_64.whl
```

```
root@goorm:/workspace/여러분ID# sudo pip3 install --upgrade $TF_BINARY_URL
```

```
root@goorm:/workspace/여러분ID# apt-get install python-virtualenv
```

```
root@goorm:/workspace/여러분ID# virtualenv --system-site-packages ~/tensorflow
```

```
root@goorm:/workspace/여러분ID# source ~/tensorflow/bin/activate
```

```
(tensorflow)root@goorm:/workspace/여러분ID# source ~/tensorflow/bin/activate
```

```
(tensorflow) root@goorm:/workspace/여러분ID# mkdir mysite
```

```
(tensorflow) root@goorm:/workspace/여러분ID# cd mysite
```

```
(tensorflow) root@goorm:/workspace/여러분ID/mysite# pip3 install django==2.1
```

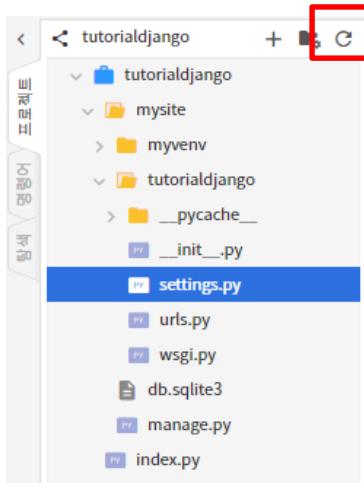
```
(tensorflow) root@goorm:/workspace/여러분ID/mysite#  
django-admin startproject tutorialdjango .
```

```
(tensorflow) root@goorm:/workspace/여러분ID/mysite# python manage.py migrate
```

'startproject 컨테이너이름' 뒤에 '.'이 있으니 주의해 주세요.

2. 환경설정(계속)

이제 settings.py로 이동하셔서 28번째 줄을 아래와 같이 고쳐주세요. 그런 다음 아래 명령어를 콘솔에 입력해주세요.



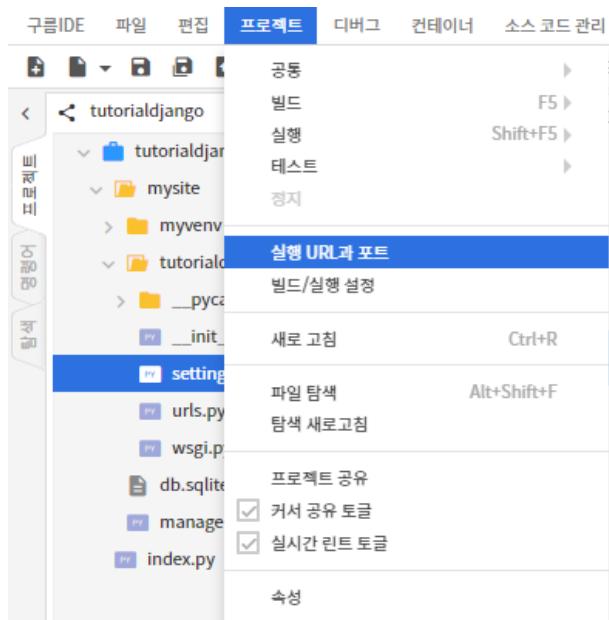
28 ALLOWED_HOSTS = ['*']

Settings.py에 28번째 줄에 가셔서 해당 값을 '*'로 바꿔 주십시오. 모든 사용자의 접속을 허락하겠다는 것입니다.

(myvenv)root@goorm:/workspace/컨테이너명/mysite# python manage.py runserver 0:80

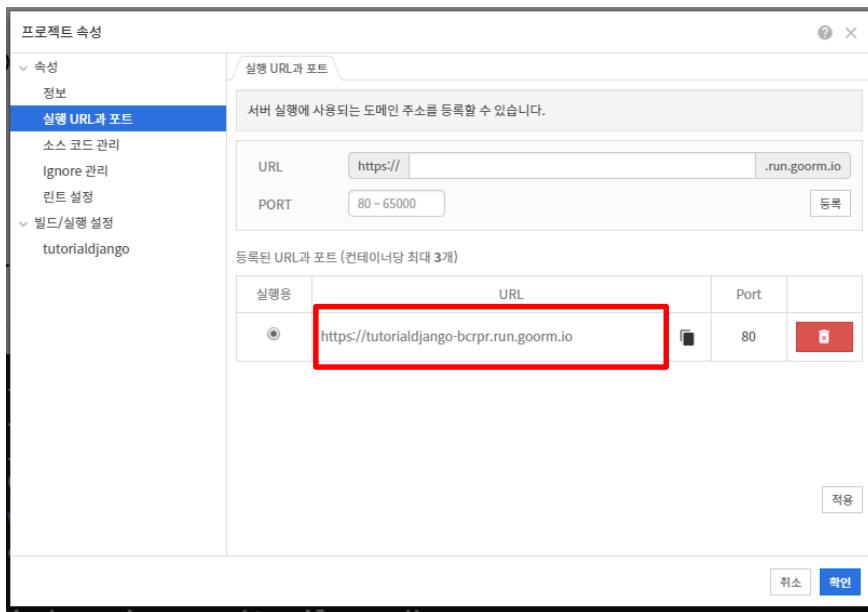
```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py runserver 0:80
Performing system checks...

System check identified no issues (0 silenced).
April 30, 2018 - 07:10:49
Django version 2.0.4, using settings 'tutorialdjango.settings'
Starting development server at http://0:80/
Quit the server with CONTROL-C.
```



2. 환경설정

해당 url을 클릭하시면 아래와 같이 실행화면이 뜹니다. Django가 실행되고 있어요!



django View release notes for Django 2.0



Step 2

Django로 Main page 만들기

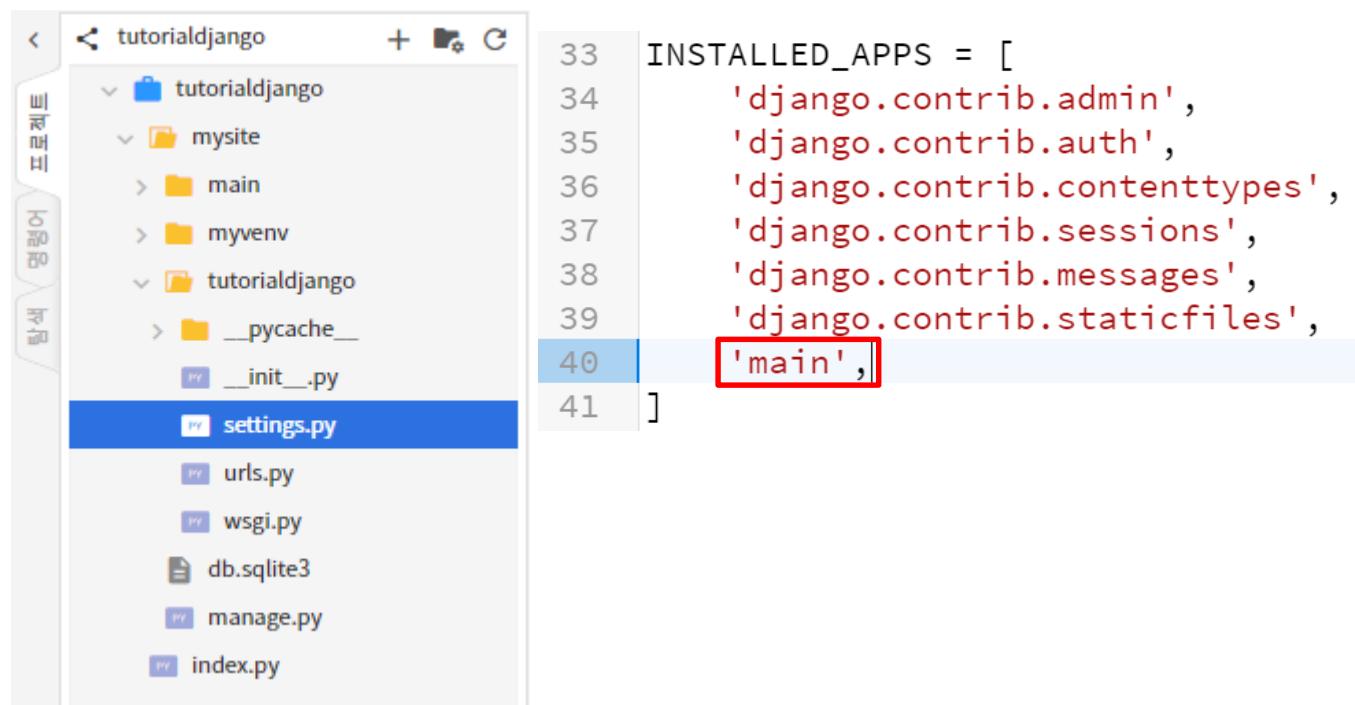
1. Django로 Main page 만들기(계속)

Django는 여러 개의 앱을 동시에 만들어 조립하는 과정으로 보실 수 있어요. 우리는 간단한 실습이기 때문에 main이라는 앱만을 만들 것입니다. Settings에 가셔서 설치되어 있는 앱 목록에 main을 추가해주세요. 이 작업을 하지 않으면 앱이 구동하지 않습니다. 여러 개의 앱을 만들 경우 모두 여기 등록해 주셔야 합니다.

* Main뒤에 쉼표가 있으니 꼭 체크하세요.

```
root@goorm:/workspace/컨테이너명/mysite# python manage.py startapp main
```

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py startapp main  
(myvenv)root@goorm:/workspace/tutorialdjango/mysite#
```



The screenshot shows a file explorer on the left and a code editor on the right. The file explorer displays a project structure with a 'tutorialdjango' folder containing a 'mysite' folder with a 'main' subfolder, and other files like 'myvenv', 'tutorialdjango', '_pycache_'. The 'settings.py' file is selected in the file list. The code editor shows the 'INSTALLED_APPS' configuration:

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     'main', // Line 40 is highlighted with a red box  
41 ]
```

1. Django로 Main page 만들기(계속)

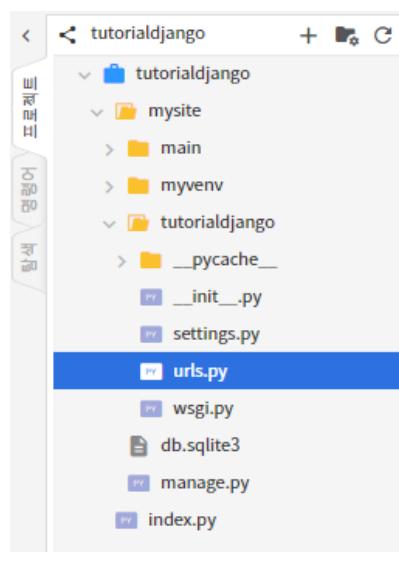
이제 urls.py를 작성할 것입니다. urls는 사용자가 어떤 url을 사용하여 들어오느냐에 따라 어떤 화면을 보여줄지를 결정하게 됩니다. 여기서

from main.views import index는

main폴더 안에 views.py파일 안에 index라는 함수를 연결시켜 주겠다는 뜻입니다. 아직 만들 어지지 않았기 때문에 오류가 뜹니다.

또한 path(“, index),

의 의미는 사용자가 뒤에 무언가 붙지 않은 있는 그대로의 url을 입력하고 들어왔을 경우 index라는 것을 연결시켜 주겠다는 얘기입니다. 여기서 index에는 index.html을 연결시켜 줄 것입니다.



The screenshot shows the PyCharm IDE interface with the project structure on the left and the code editor on the right. The project structure tree shows a 'mysite' folder containing 'main', 'myenv', and 'tutorialdjango' subfolders. 'tutorialdjango' contains '__pycache__', '__init__.py', 'settings.py', and 'urls.py'. The 'urls.py' file is selected in the editor. The code in 'urls.py' is:

```
16 from django.contrib import admin
17 from django.urls import path
18 from main.views import index
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', index),
23 ]
```

Below the code editor, the path 'mysite > tutorialdjango > urls.py' is displayed.

1. Django로 Main page 만들기(계속)

이제 mysite > main > views.py로 들어와 index라는 함수를 만듭니다. 여기서 사용자가 index.html을 볼 수 있게끔 연결을 해줍니다.

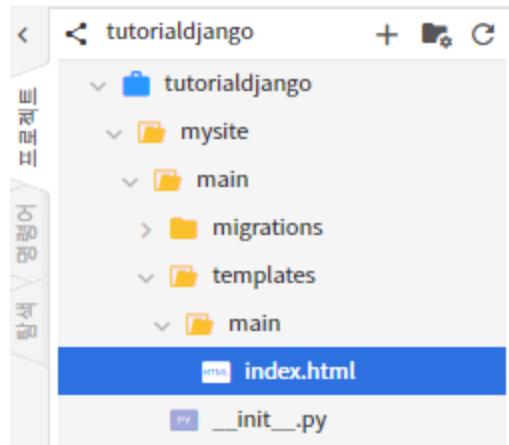
mysite > main > views.py

```
1 from django.shortcuts import render
2
3 def index(request):
4     return render(request, 'main/index.html')
```

1. Django로 Main page 만들기(계속)

Template을 연결 시키는 과정으로 **mysite** > **main** > **templates** > **main** > **index.html** 을 만들어 주세요. 기본적인 테스트를 위하여 아래 내용을 붙여 넣어 주세요.

```
<html>
<head>
    <title>Django!</title>
</head>
<body>
    <h1>Test!</h1>
</body>
<html>
```



```
1  <html>
2  <head>
3      <title>Django!</title>
4  </head>
5  <body>
6      <h1>Test!</h1>
7  </body>
8  <html>
```

1. Django로 Main page 만들기(계속)

이제 다시 아래 명령어를 콘솔창에 입력하시고 프로젝트 > 실행URL과 포트를 클릭하신 다음 URL을 클릭하셔서 실행되는 화면을 보십시오.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite# python manage.py runserver 0:80
```

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py runserver 0:80
Performing system checks...

System check identified no issues (0 silenced).
April 30, 2018 - 07:34:58
Django version 2.0.4, using settings 'tutorialdjango.settings'
Starting development server at http://0:80/
Quit the server with CONTROL-C.
Performing system checks...
```

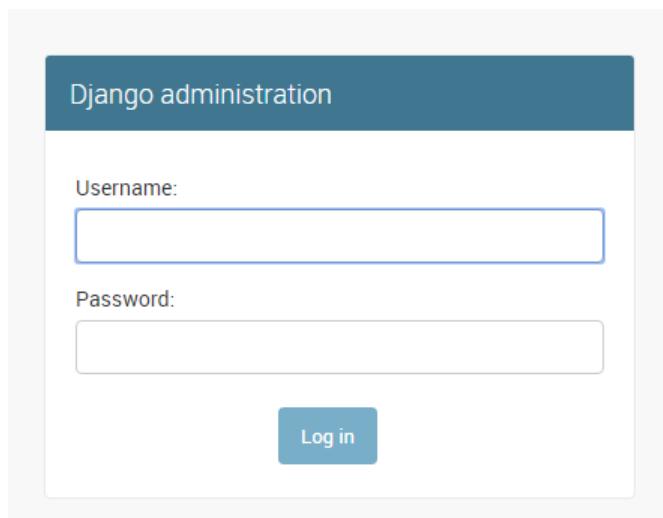


Test!

1. Django로 Main page 만들기(계속)

이번에는 뒤에 /admin을 입력하셔서 admin 페이지가 열리는지 확인해주세요. 여기서는 게시물을 올리거나 수정하거나 삭제할 수 있으며 user에 대한 관리도 할 수 있습니다. 아직 superuser를 만들지 않았으므로 접속 할 수 없습니다.

urls.py에서 보았던 admin/ 부분이 이 페이지를 뜻하는 것입니다. 보안상 실 서비스에서는 열어두지 않는 것이 좋습니다.

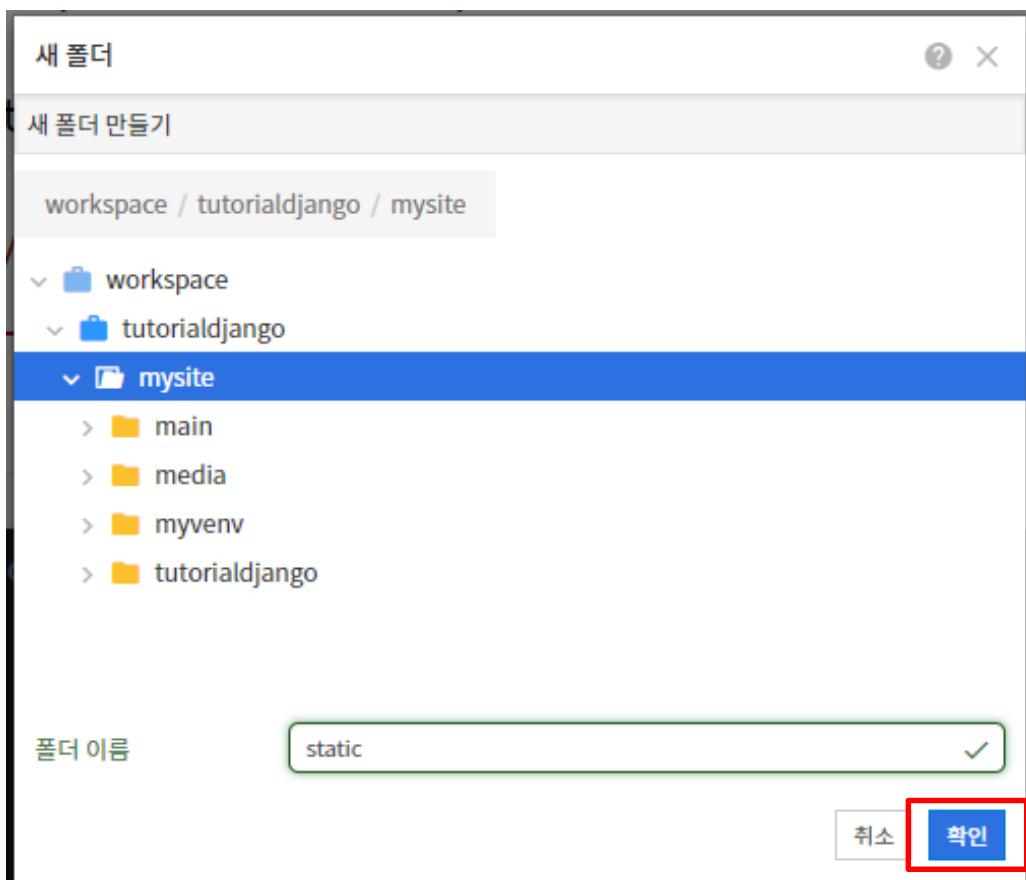


<https://여러분URL/admin>

1. Django로 Main page 만들기(계속)

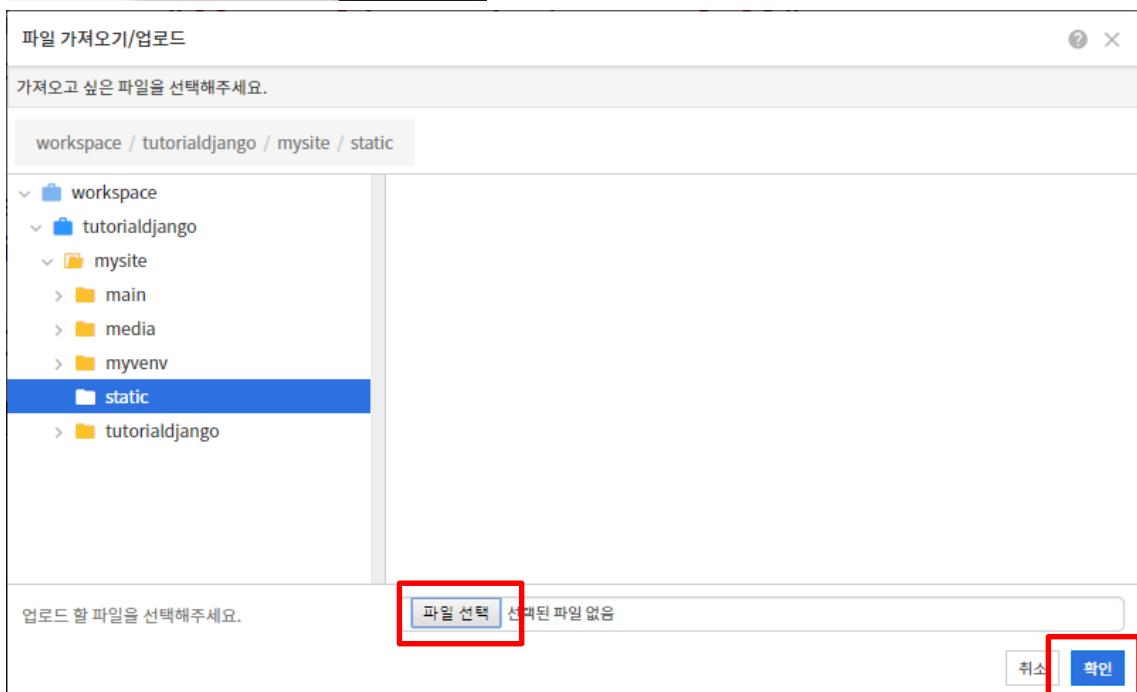
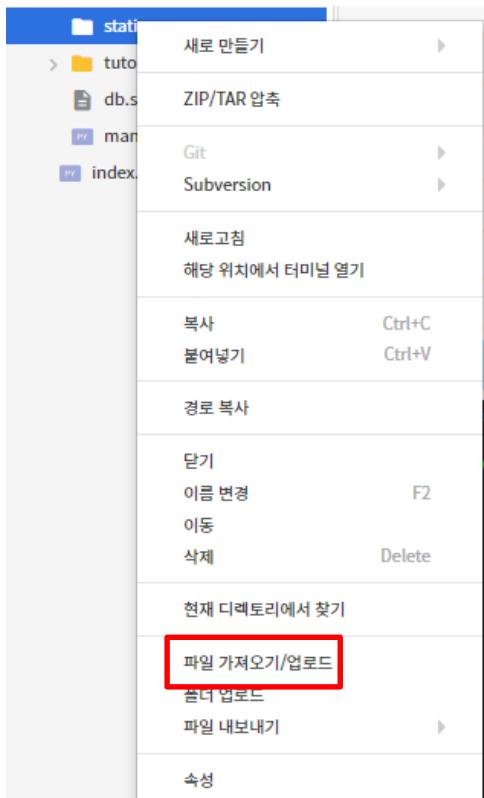
다음으로는 메인페이지에서 이미지를 불러오는 방법에 대해 소개하려 합니다. Django에서는 일반적인 HTML, CSS에서 지원하는 상대경로를 지원하지 않습니다. 우선 정적 파일들이 저장될 static이라는 폴더를 mysite 밑에 만들어 줍니다.

1) 새폴더 만들기



1. Django로 Main page 만들기(계속)

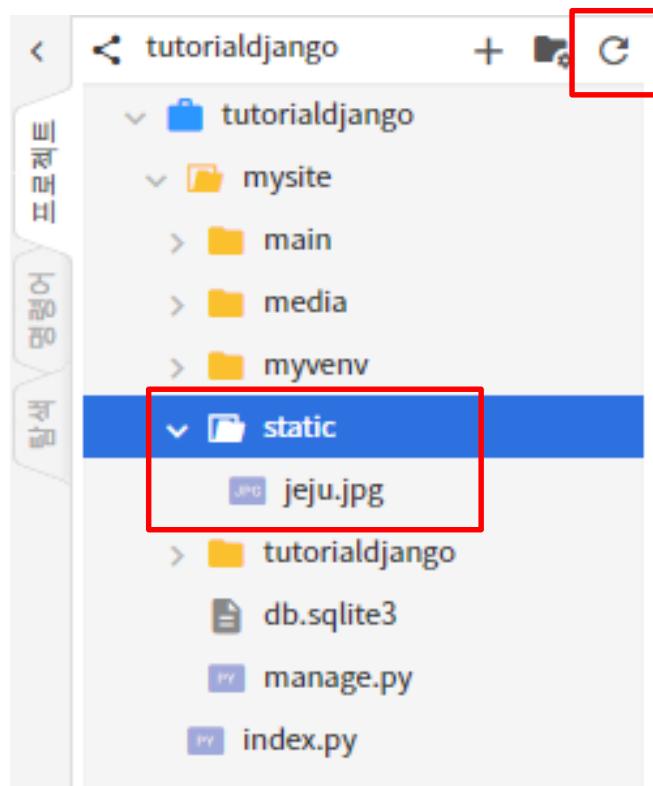
2) 파일 업로드



1. Django로 Main page 만들기(계속)

3) 파일 업로드 확인

혹시 이미지가 안떴을 경우에는 새로고침을 눌러주세요.



1. Django로 Main page 만들기(계속)

다음은 static 폴더를 django와 연결시켜 주는 작업이 남았습니다. settings.py로 이동하셔서 아래와 같이 수정해주세요. 뒤에 콤마가 있으니 빼먹지 않도록 주의해주세요.

Time Zone 변경하고 가도록 하겠습니다.

```
121 STATIC_URL = '/static/'  
122 STATICFILES_DIRS = (  
123     os.path.join(BASE_DIR, 'static'),  
124 )
```

```
109 TIME_ZONE = 'Asia/Seoul'  
110  
111 USE_I18N = True  
112  
113 USE_L10N = True  
114  
115 USE_TZ = True
```

1. Django로 Main page 만들기(계속)

이제 index.html로 들어가 static 파일을 호출하는 code를 넣어주세요. { % 문법 %}의 구조는 template tag라고 불립니다. 자세한 내용은 Django 공식문서 (<https://docs.djangoproject.com/en/2.1/ref/templates/builtins/>)를 참고하세요.

```
1 <html>
2 <head>
3     <title>Django!</title>
4 </head>
5 <body>
6     <h1>Test!</h1>
7     {% load staticfiles %}
8     
9 </body>
10 </html>
```

tutorialdjango/mysite/main/templates/main/index.html

1. Django로 Main page 만들기(계속)

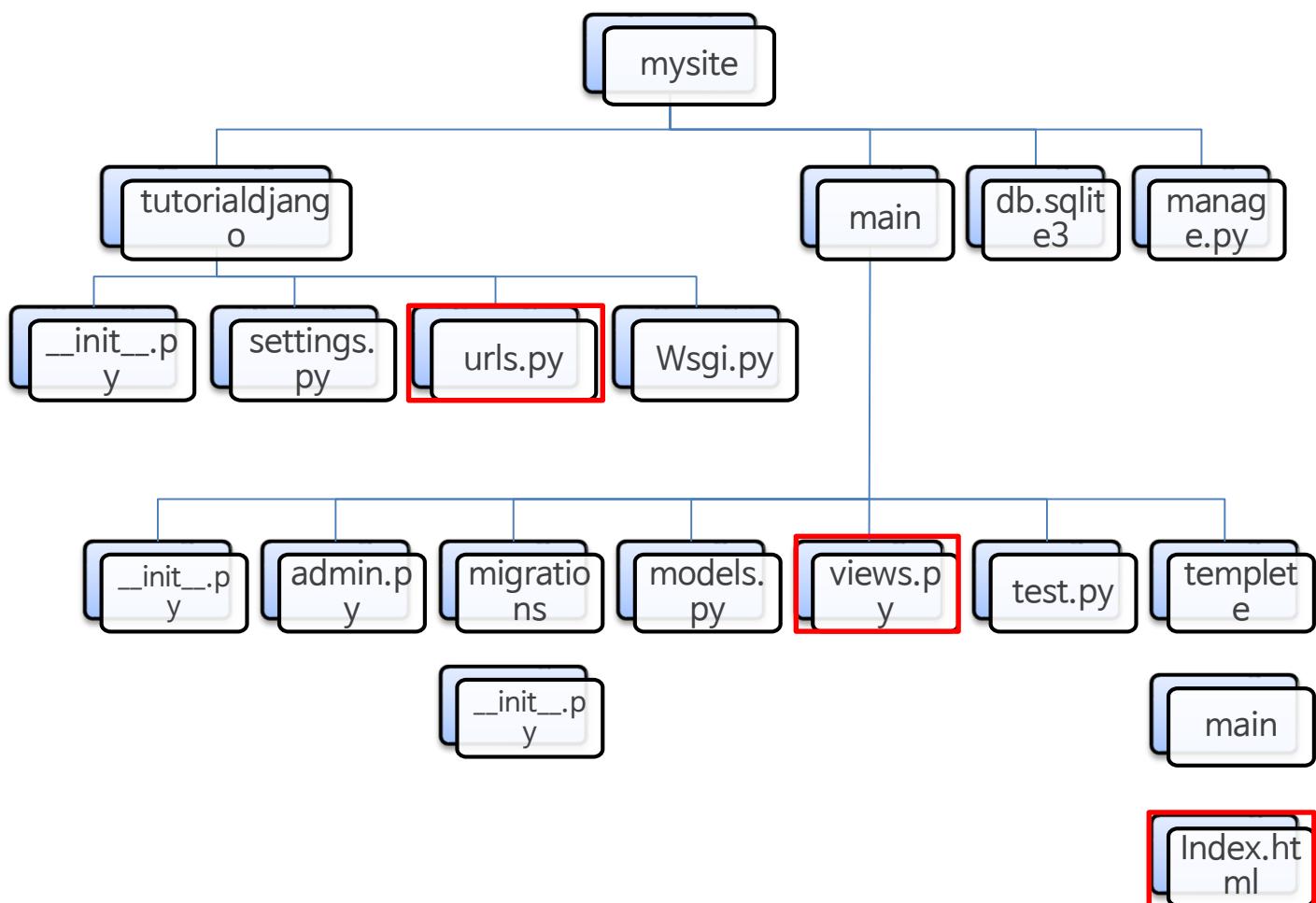
콘솔창에 python manage.py runserver 0:800이 입력이 안되어 있으시다면 입력하시고 프로젝트 > 실행URL과 포트를 누르신 다음 URL을 클릭하시면 아래와 같은 화면이 나옵니다.

Test!



1. Django로 Main page 만들기(계속)

현재 폴더 구성 트리입니다. 빨간색으로 네모를 친 것이 우리가 수정한 파일입니다.



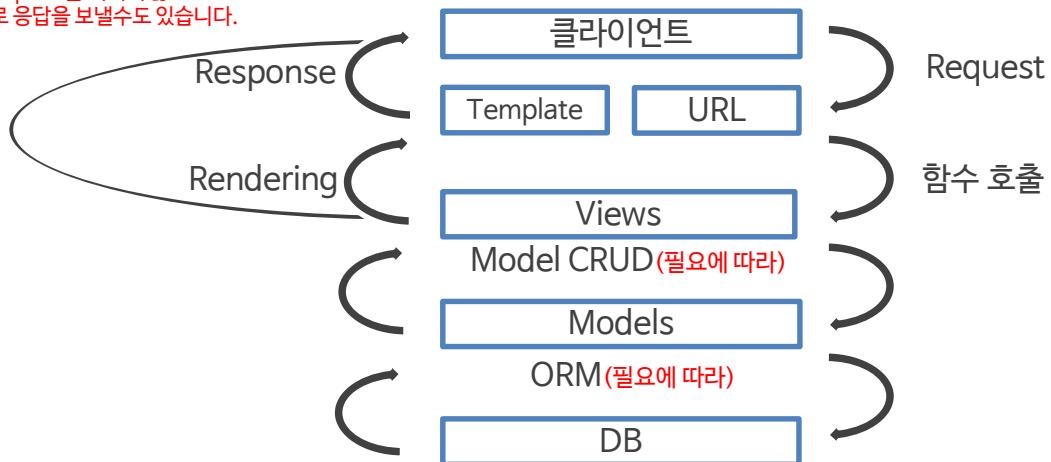
1. Django로 Main page 만들기

Django는 MVC모델을 MTV라고 부릅니다. 용어가 바뀌었을 뿐이지 기본적인 개념은 같습니다.

아래 표는 Django 작동원리를 도식화 한 것입니다. 8장에서 좀 더 상세하게 다룹니다.

MTV(빨간색으로 표시된 부분은 설계에 따라 작동하지 않을 수도 있습니다.)

Template를 거치지 않고
바로 응답을 보낼 수도 있습니다.



Model - DB와 연결된 Python Class

Template - 사용자에게 response될 Client View

View - Django에서 처리한 데이터를 Template에게 전달

각 문자는 다음과 같이 표준 SQL문으로 대응 가능하다.

| 이름 | 조작 | SQL |
|--------------------|-----------|--------|
| Create | 생성 | INSERT |
| Read(또는 Retrieve) | 읽기(또는 인출) | SELECT |
| Update | 갱신 | UPDATE |
| Delete(또는 Destroy) | 삭제(또는 파괴) | DELETE |

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create(생성), Read(읽기), Update(갱신), Delete(삭제)를 묶어서 일컫는 말입니다.

<https://ko.wikipedia.org/wiki/CRUD>

Step 3

Django로 Cafelist page 만들기

1. Django로 Cafelist page 만들기(계속)

이번에는 포스팅이 모아져 있는 cafelist 페이지를 만들어 보도록 하겠습니다. 우선 urls.py 파일을 수정하고 views.py 파일을 수정합니다.

```
from django.contrib import admin
from django.urls import path
from main.views import index, cafelist

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index),
    path('cafelist/', cafelist),
]
```

tutorialdjango/mysite/tutorialdjango/urls.py

```
from django.shortcuts import render

def index(request):
    return render(request, 'main/index.html')

def cafelist(request):
    return render(request, 'main/cafelist.html')
```

tutorialdjango/mysite/main/views.py

1. Django로 Cafelist page 만들기(계속)

이제 html 파일을 만들 차례입니다. blog.html 파일을

[mysite/main/templates/main/cafelist.html](#)

에 작성해주세요.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>cafelist</title>
5  </head>
6  <body>
7      <h1>cafelist</h1>
8  </body>
9  </html>
```

[/mysite/main/templates/main/cafelist.html](#)

1. Django로 Cafelist page 만들기(계속)

만약 중간에 한번 나갔다 오셨다면 아래 명령어처럼 쳐주세요. 계속해서 진행하고 계셨다면
python manage.py runserver 0:80
만 입력하시면 됩니다.

프로젝트 > 실행 URL과 포트를 누르시고 URL을 클릭하신 다음 뒤에 /cafelist라고 치시면
cafelist페이지가 나옵니다.

```
root@goorm:/workspace/tutorialdjango# cd mysite/
root@goorm:/workspace/tutorialdjango/mysite# source myvenv/bin/activate
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py runserver 0:80
Performing system checks...

System check identified no issues (0 silenced).
May 02, 2018 - 06:25:59
Django version 2.0.4, using settings 'tutorialdjango.settings'
Starting development server at http://0:80/
Quit the server with CONTROL-C.
```

실행화면(<https://tutorialdjango-bcrpr.run.goorm.io/cafelist/>)



A screenshot of a web browser window. The address bar shows the URL <https://jejucodingcampsample-iqusj.run.goorm.io/cafelist/>. The page content is a single word "cafelist" in a large, bold, black font.

cafelist

1. Django로 Cafelist page 만들기(계속)

이제 각각의 포스팅에 저장될 공간을 만들어보도록 하겠습니다. 우선 name과 content를 만듭니다. 사진을 저장할 공간은 다른 챕터에서 만들도록 하겠습니다.

tutorialdjango/mysite/main/models.py

```
from django.db import models

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    content = models.TextField()
```

Ctrl + C를 누르시고 빠져나오신 다음 아래 명령어를 입력해주세요.

(myvenv)root@goorm:/workspace/컨테이너명/mysite#

python manage.py makemigrations main

(myvenv)root@goorm:/workspace/컨테이너명/mysite#

python manage.py migrate

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py makemigrations main
Migrations for 'main':
  main/migrations/0001_initial.py
    - Create model Post
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions
Running migrations:
  Applying main.0001_initial... OK
```

1. Django로 Cafelist page 만들기(계속)

Admin 사이트에 등록 할 수 있도록 합니다. 이 공간에서 admin 사이트 설정을 할 수 있습니다.

tutorialdjango/mysite/main/admin.py

```
from django.contrib import admin  
from .models import Cafe  
  
admin.site.register(Cafe)
```

1. Django로 Cafelist page 만들기(계속)

이제 Superuser를 만듭니다. Superuser는 게시물을 삭제, 수정, 저장할 수 있으며 다른 유저를 관리할 수 있습니다.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite#  
python manage.py createsuperuser
```

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py createsuperuser  
Username (leave blank to use 'root'): tutorialdjango  
Email address: tutorialdjango@gmail.com  
Password:  
Password (again):  
Superuser created successfully.
```

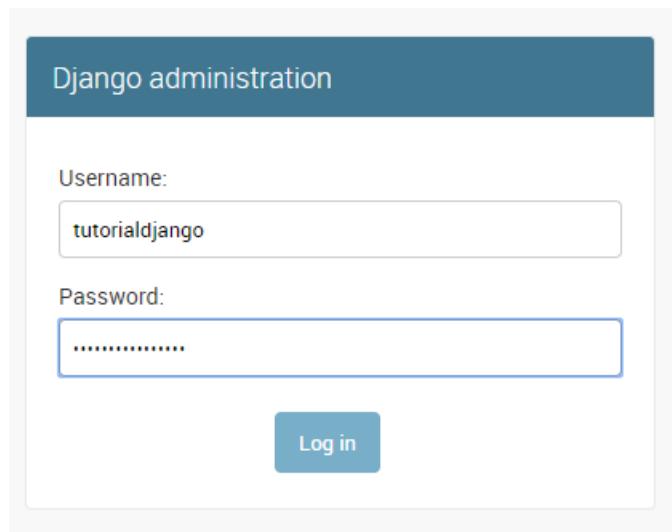
Admin user를 만들 것입니다.

비밀번호는 원래 콘솔창에 안뜨니 입력하시고 엔터를 치시면 됩니다.

```
python manage.py runserver 0:80
```

를 치셔서 Django를 실행시킨 다음 프로젝트 > 실행 URL과 포트를 누르시고 URL을 클릭하신다음 뒤에 /admin라고 치시면 admin페이지가 나옵니다.

<https://tutorialdjango-bcrpr.run.goorm.io/admin/>



1. Django로 Cafelist page 만들기(계속)

Admin 사이트 안으로 들어오셨다면 게시물을 올려보도록 하겠습니다. 구분을 위해서 postname과 contents의 내용을 다르게 입력해주세요.

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

| | | |
|--------|-------|--------|
| Groups | + Add | Change |
| Users | + Add | Change |

MAIN

Cafes

+ Add Change

- 캡쳐화면 -



Django administration

WELCOME, LEEHOJUN | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Cafes > Add cafe

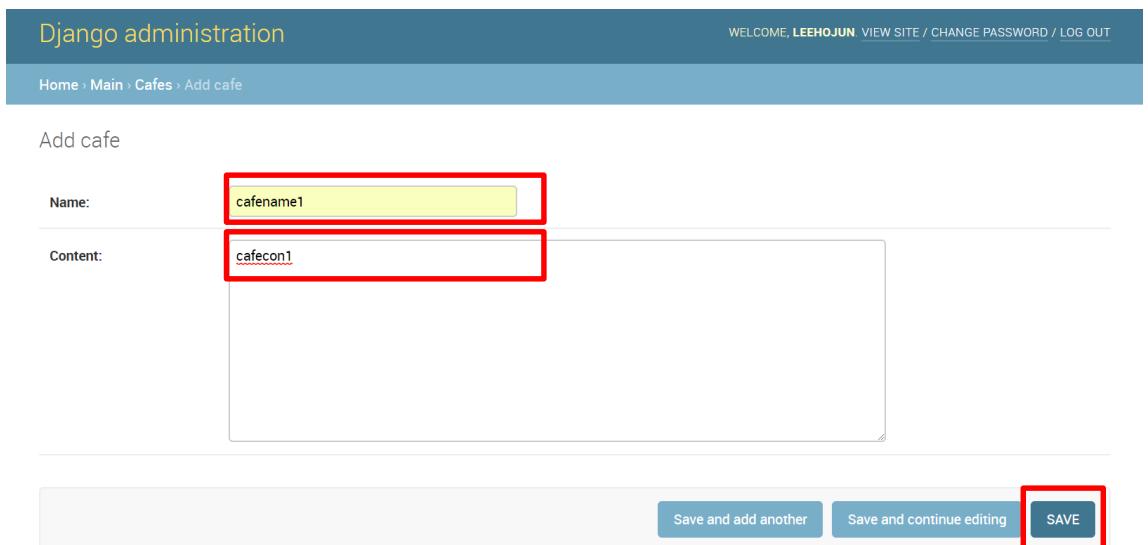
Add cafe

Name: cafename1

Content: cafecon1

Save and add another Save and continue editing SAVE

- 캡쳐화면 -



1. Django로 Cafelist page 만들기(계속)

SAVE를 클릭하시면 제목이 name_1이 아닙니다. 제목을 name으로 출력하라는 명령이 없었기 때문입니다. 이게 제목이라는 것을 아는 것은 우리뿐이라는 것이죠. models.py로 들어가서 다시 제목이 무엇인지 설정해줄게요.

The screenshot shows the Django admin 'Posts' list page. At the top, it says 'Django administration' and 'WELCOME, TUTORIALDJANGO. VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below that, the breadcrumb navigation shows 'Home > Main > Posts'. A green success message box contains the text 'The post "Post object (1)" was added successfully.' On the left, there's a 'Select post to change' sidebar with an 'ADD POST +' button. The main area lists one post: 'Post object (1)' with a checkbox next to it. A red box highlights this checkbox. Below the list, it says '1 post'.

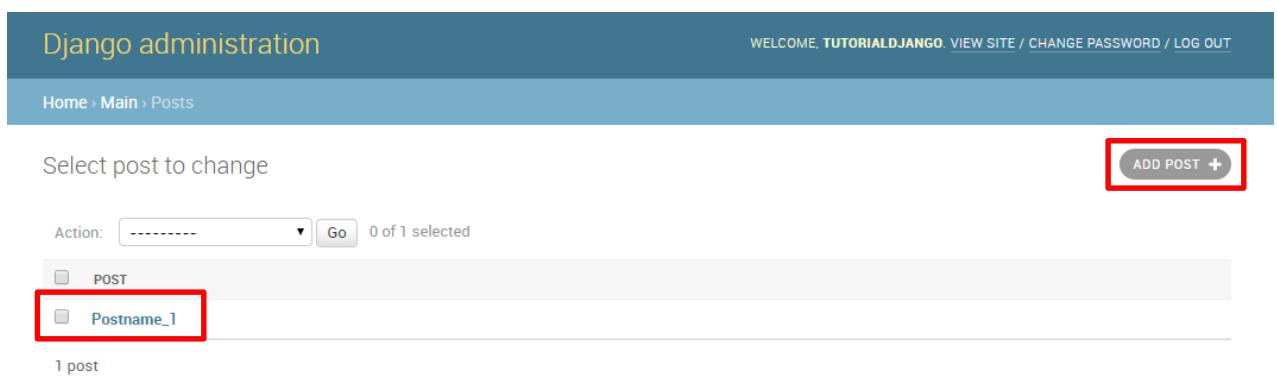
```
mysite/main/models.py
from django.db import models

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    content = models.TextField()

    def __str__(self):
        return self.name
```

1. Django로 Cafelist page 만들기(계속)

브라우저 창에서 새로고침을 해보시면 이름이 바뀐 것을 볼 수 있습니다. 이제 새로운 포스팅을 몇 개 작성해 보겠습니다.(나중에 삭제할 게시물입니다.)



Django administration

WELCOME, TUTORIALDJANGO. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Posts

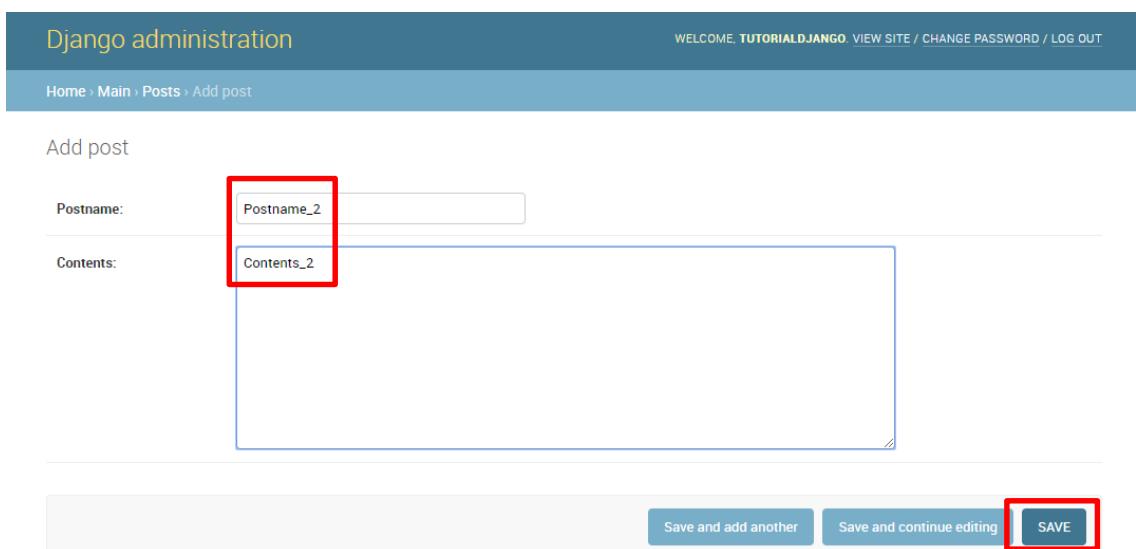
Select post to change

Action: ----- Go 0 of 1 selected

POST
 Postname_1

1 post

게시물 추가



Django administration

WELCOME, TUTORIALDJANGO. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Posts > Add post

Add post

Postname: Postname_2

Contents: Contents_2

Save and add another Save and continue editing **SAVE**

1. Django로 Cafelist page 만들기(계속)

Test를 위해 3개의 게시물을 작성하였습니다. 이번에는 이 게시물을 cafelist 페이지에 띄어 보도록 하겠습니다. views 파일을 아래와 같이 수정해주세요.

The screenshot shows the Django administration interface for a 'Cafes' model. At the top, it says 'Django administration' and 'WELCOME, LEEHOJUN'. Below that, there's a breadcrumb navigation: 'Home > Main > Cafes'. A main title 'Select cafe to change' is centered above a list of three items. Each item has a checkbox next to its name: 'CAFE', 'cafename3', 'cafename2', and 'cafename1'. A button labeled 'ADD CAFE +' is located in the top right corner. Below the list, it says '3 cafes'.

tutorialdjango/mysite/main/views.py

```
from django.shortcuts import render
from .models import Cafe

def index(request):
    return render(request, 'main/index.html')

def cafelist(request):
    cafelistobj = Cafe.objects.all()
    return render(request, 'main/cafelist.html', {'cafelistobj':cafelistobj})
```

1. Django로 Cafelist page 만들기(계속)

Template tag를 사용하여 DB에 입력된 내용을 모두 가져옵니다. 템플릿 테그와 랭귀지에 관련된 자세한 내용은

* 공식 문서 : <https://docs.djangoproject.com/en/2.1/>

* templates language :

<https://docs.djangoproject.com/en/2.1/ref/templates/language/>

를 참고해주세요.

```
<!DOCTYPE html>
<html>
<head>
    <title>cafelist</title>
</head>
<body>
    <h1>cafelist</h1>
    <table>
        {% for list in cafelistobj %}
        <tr>
            <td>{{list.name}}</td>
            <td>{{list.content}}</td>
        </tr>
        {% endfor %}
    </table>
</body>
</html>
```

← → C ⌂ 🔒 h

cafelist

cafename1 cafecon1
cafename2 cafecon2
cafename3 cafecon3

- 실행화면 -

1. Django로 Cafelist page 만들기

소스코드보기를 하면 우리가 작성했던 템플릿 랭귀지가 보이지 않습니다.

이렇게 사용자에게 보여질 때에는 html 형식으로 넘겨주게 됩니다.



```
1 | <!DOCTYPE html>
2 | <html>
3 | <head>
4 |   <title>cafelist</title>
5 | </head>
6 | <body>
7 |   <h1>cafelist</h1>
8 |   <table>
9 |
10|     <tr>
11|       <td>cafename1</td>
12|       <td>cafecon1</td>
13|     </tr>
14|
15|     <tr>
16|       <td>cafename2</td>
17|       <td>cafecon2</td>
18|     </tr>
19|
20|     <tr>
21|       <td>cafename3</td>
22|       <td>cafecon3</td>
23|     </tr>
24|
25|   </table>
26| </body>
27| </html>
```

- 소스보기 -

Step 4

Django로 cafe details page 만들기

1. Django로 Cafe details page 만들기(계속)

챕터가 넘어왔습니다. 아래 명령어는 구름IDE에서 다시 접속하실 때마다 실행해 주셔야 하는 명령어입니다.

```
root@goorm:/workspace/컨테이너명# cd mysite
```

```
root@goorm:/workspace/컨테이너명/mysite# source myvenv/bin/activate
```

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite#
```

myvenv가 붙지 않은 상태에서 그동안 명령어를 치셨다면 지금이라도 컨테이너를 삭제해버리시고 처음부터 다시 하시길 권장해 드립니다.

(myvenv)는 (myvenv)가 붙지 않은 환경과 완전히 다른 환경이라고 보시면 됩니다.

1. Django로 Cafe details page 만들기(계속)

이제 blog페이지에서 post를 눌렀을 때 나오는 상세 posting 화면을 만들어 보도록 하겠습니다.
홈페이지 전체 구성 중에서는 3번을 작업하는 중에 있습니다.

3. http://사용자아이름.run.goorm.io/post번호

- blog 페이지에서 post를 눌렀을 때 이동
- Template : details.html

```
from django.contrib import admin
from django.urls import path
from main.views import index, cafelist, cafedetails

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index),
    path('cafelist/', cafelist),
    path('cafelist/<int:pk>', cafedetails),
]
tutorialdjango/mysite/tutorialdjango/urls.py
```

```
from django.shortcuts import render
from .models import Cafe

def index(request):
    return render(request, 'main/index.html')

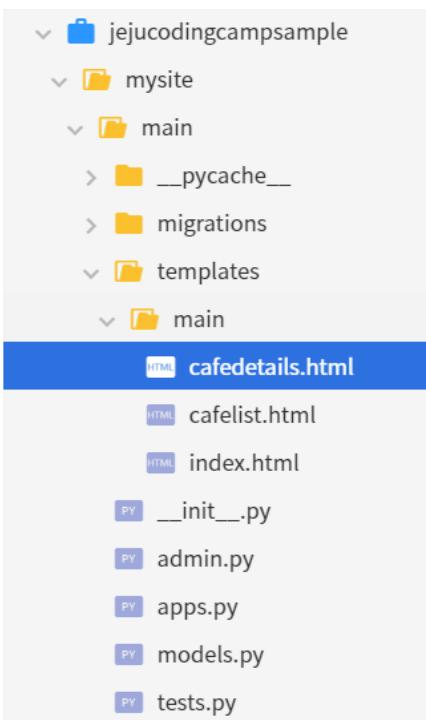
def cafelist(request):
    cafelistobj = Cafe.objects.all()
    return render(request, 'main/cafelist.html', {'cafelistobj':cafelistobj})

def cafedetails(request, pk):
    cafeobj = Cafe.objects.get(pk=pk)
    return render(request, 'main/cafedetails.html', {'cafeobj':cafeobj})
```

tutorialdjango/mysite/main/views.py

1. Django로 Cafe details page 만들기(계속)

이제 아래 화면처럼 main 안에 템플릿은 3개가 됩니다. 마지막 템플릿입니다.(꾸미는 것은 7장에서 합니다.)



```
<!DOCTYPE html>
<html>
<head>
    <title>cafelist</title>
</head>
<body>
    <h1>cafelist</h1>
    <p>{{cafeobj.name}}</p>
    <p>{{cafeobj.content}}</p>
</body>
</html>
```

1. Django로 Cafe details page 만들기(계속)

아래 url패턴처럼 blog/post번호를 입력하시면 아래 페이지처럼 출력이 됩니다.

<https://tutorialdjango-bcrpr.run.goorm.io/cafelist/1/>

← → C ⌂ 🔒 https://jejucodingcampsample-iqusj.run.goorm.io/cafelist/1/

cafedetails

cafename1

cafecon1

- 실행화면 -

1. Django로 Cafe details page 만들기(계속)

간단하게 blog화면에서 post제목과 내용을 클릭하였을 때 해당 posting으로 이동하도록 만들어 보도록 하겠습니다.

'<https://tutorialdjango-bcrpr.run.goorm.io/>' 이 부분은 여러분의 URL로 교체해 주세요.

`tutorialdjango/mysite/main/templates/main/cafelist.html`

```
<!DOCTYPE html>
<html>
<head>
  <title>cafelist</title>
</head>
<body>
  <h1>cafelist</h1>
  <table>
    {% for list in cafelistobj %}
      <tr onclick="location.href='https://jejucodingcampsample-iqusj.run.goorm.io/cafelist/{{list.pk}}'">
        <td>{{list.name}}</td>
        <td>{{list.content}}</td>
      </tr>
    {% endfor %}
  </table>
</body>
</html>
```

1. Django로 Cafe details page 만들기(계속)

연결된 화면입니다. 해당 post 제목과 내용을 클릭하면 해당 포스팅으로 이동합니다.

cafelist

cafename1 cafecon1

cafename2 cafecon2

cafename3 cafecon3



cafedetails

cafename1

cafecon1

1. Django로 Cafe details page 만들기(계속)

이번에는 반대로 posting 상세 내용에서 목록으로 넘어가는 버튼을 만들어 보도록 하겠습니다.
아래 url은 적절하게 수정해주세요.

컨테이너명/mysite/main/templates/main/cafedetails.html

```
<!DOCTYPE html>
<html>
<head>
  <title>cafelist</title>
</head>
<body>
  <h1>cafedetails</h1>
  <p>{{cafeobj.name}}</p>
  <p>{{cafeobj.content}}</p>
  <a href="https://jejucodingcampsample-iqusj.run.goorm.io/cafelist/">목록</a>
</body>
</html>
```

1. Django로 Cafe details page 만들기(계속)

이제 사진이 들어갈 부분을 만들어 보도록 하겠습니다. tutorialdjango > mysite > main > models.py로 이동하겠습니다. 여기서 null=True를 주게 되면 기존에 있는 게시판 게시물은 null 값으로 들어가게 됩니다.

tutorialdjango/mysite/main/models.py

```
from django.db import models

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    mainphoto = models.ImageField(blank=True, null=True)
    content = models.TextField()

    def __str__(self):
        return self.name
```

1. Django로 Cafe details page 만들기(계속)

위와 같이 수정하셨다면 이제 사진을 처리할 수 있는 라이브러리를 설치 해야 합니다. Pillow라는 라이브러리이며 아래와 같이 설치 할 수 있습니다.

```
(myvenv)root@goorm:/workspace/컨테이너명/mysite# pip3 install pillow==2.9.0
(myvenv)root@goorm:/workspace/컨테이너명/mysite#
python manage.py makemigrations
(myvenv)root@goorm:/workspace/컨테이너명/mysite#
python manage.py migrate
(myvenv)root@goorm:/workspace/컨테이너명/mysite#
python manage.py runserver 0:80
```

```
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# pip3 install pillow==2.9.0
Collecting pillow==2.9.0
  Downloading https://files.pythonhosted.org/packages/62/7d/a6decd77e926197621ca13f63e4
462e1be18a8b5928b692f7ae24/Pillow-2.9.0.tar.gz (9.3MB)
    100% |████████████████████████████████| 9.3MB 5.1MB/s
```

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py makemigrations
Migrations for 'main':
 main/migrations/0002_post_mainphoto.py
 - Add field mainphoto to post
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions
Running migrations:
  Applying main.0002_post_mainphoto... OK
(myvenv)root@goorm:/workspace/tutorialdjango/mysite#
```

```
(myvenv)root@goorm:/workspace/tutorialdjango/mysite# python manage.py runserver 0:80
Performing system checks...

System check identified no issues (0 silenced).
May 02, 2018 - 07:38:06
Django version 2.0.4, using settings 'tutorialdjango.settings'
Starting development server at http://0:80/
Quit the server with CONTROL-C.
```

1. Django로 Cafe details page 만들기(계속)

* Change post

Django administration

WELCOME, TUTORIALDJANGO. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Posts > Postname_1

Change post

Postname: Postname_1

Mainphoto: 파일 선택 선택된 파일 없음

Contents: Contents_1

Delete Save and add another Save and continue editing SAVE

* Add post

Django administration

WELCOME, TUTORIALDJANGO. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Posts > Add post

Add post

Postname:

Mainphoto: 파일 선택 선택된 파일 없음

Contents:

Save and add another Save and continue editing SAVE

1. Django로 Cafe details page 만들기(계속)

이제 사진이 저장될 공간을 설정해주도록 하겠습니다. 이 설정을 하지 않으면 사진을 업로드 할 경우 mysite 바로 아래로 파일이 들어가게 됩니다.

tutorialdjango/mysite/tutorialdjango/settings.py

```
126     MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
127     MEDIA_URL = '/media/'
```

이미지 파일은 총 3개를 올려보도록 하겠습니다.



cafe_0



cafe_1



cafe_2

순서대로 café_0부터 올리겠습니다.

Django administration

WELCOME, LEEHOJUN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Main > Cafes > cafename1

Change cafe

HISTORY

Name:

Mainphoto: 선택된 파일 없음

Content:

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)

1. Django로 Cafe details page 만들기(계속)

첫번째로 jeju의 파일을 업로드 해보았습니다. 업로드한 파일을 확인하기 위해 해당 포스팅을 클릭합니다.

The screenshot shows the Django admin interface for the 'Cafes' model. At the top, there is a breadcrumb navigation: Home > Main > Cafes. Below the navigation, a green success message box contains the text: "The cafe 'cafename1' was changed successfully." The main content area is titled "Select cafe to change". It features an "Action:" dropdown menu set to "-----" and a "Go" button. To the right of the dropdown, it says "0 of 3 selected". Below this, there is a list of three cafe entries, each with a selection checkbox:

- CAFE
- cafename3**
- cafename2**
- cafename1** (This entry is highlighted with a red rectangular border.)

At the bottom left of the list area, it says "3 cafes".

1. Django로 Cafe details page 만들기(계속)

클릭하니 page를 찾을 수 없다고 뜹니다. 이미지 url을 찾을 수 없기 때문에 발생되는 애러입니다. 이미지 url을 설정하려 가보도록 하겠습니다.

Name:

Mainphoto: Currently: Change: 선택된 파일 없음

Content:

Page not found (404)

Request Method: GET

Request URL: http://jejucodingcampsample-iqusj.run.goorm.io/media/cafe_0.jpg

Using the URLconf defined in `tutorial/django.urls`, Django tried these URL patterns, in this order:

1. `admin/`
2.
3. `cafelist/`
4. `cafelist/<int:pk>`

The current path, `media/cafe_0.jpg`, didn't match any of these.

1. Django로 Cafe details page 만들기(계속)

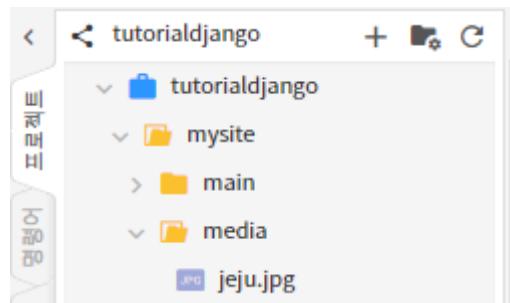
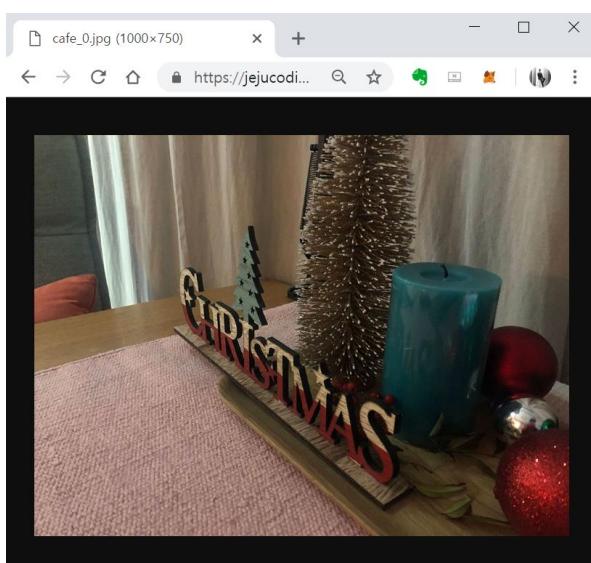
아래와 같이 urlpatterns를 넣어주세요. 그러면 이미지가 정상적으로 보입니다.

* 나머지 2개의 사진도 업로드 해주세요.

```
from django.contrib import admin
from django.urls import path
from main.views import index, cafelist, cafedetails
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index),
    path('cafelist/', cafelist),
    path('cafelist/<int:pk>', cafedetails),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```



1. Django로 Cafe details page 만들기(계속)

이번에는 cafedetails page에서 해당 이미지를 불러와 보도록 하겠습니다. 빨간색 네모 부분은 여러분 홈페이지 주소로 바꿔주세요. 해당 내용의 자세한 내용은 공식 홈페이지 file부분 <https://docs.djangoproject.com/en/2.1/topics/files/> 을 참고해주세요.

tutorialdjango/mysite/main/templates/main/postdetails.html

```
<!DOCTYPE html>
<html>
<head>
    <title>cafelist</title>
</head>
<body>
    <h1>cafedetails</h1>
    <p>{{cafeobj.name}}</p>
    <p>{{cafeobj.content|linebreaks}}</p>
    {% if cafeobj.mainphoto %}
        
    {% endif %}
    <a href='https://jecodingcampsample-iqusj.run.goorm.io/cafelist/'>목록</a>
</body>
</html>
```

- 공식홈페이지 -

Using files in models

When you use a `FileField` or `ImageField`, Django provides a set of APIs you can use to deal with that file.

Consider the following model, using an `ImageField` to store a photo:

```
from django.db import models

class Car(models.Model):
    name = models.CharField(max_length=255)
    price = models.DecimalField(max_digits=5, decimal_places=2)
    photo = models.ImageField(upload_to='cars')
```

Any `Car` instance will have a `photo` attribute that you can use to get at the details of the attached photo:

```
>>> car = Car.objects.get(name="57 Chevy")
>>> car.photo
<ImageFieldfile: chevy.jpg>
>>> car.photo.name
'cars/chevy.jpg'
>>> car.photo.path
'/media/cars/chevy.jpg'
>>> car.photo.url
'http://media.example.com/cars/chevy.jpg'
```

This object – `car.photo` in the example – is a `File` object, which means it has all the methods and attributes described below.

1. Django로 Cafe details page 만들기

이미지가 제대로 뜯 것을 확인할 수 있습니다.

cafedetails

cafename1

cafecon1



목록

Step 5

댓글과 태그 기능 만들기

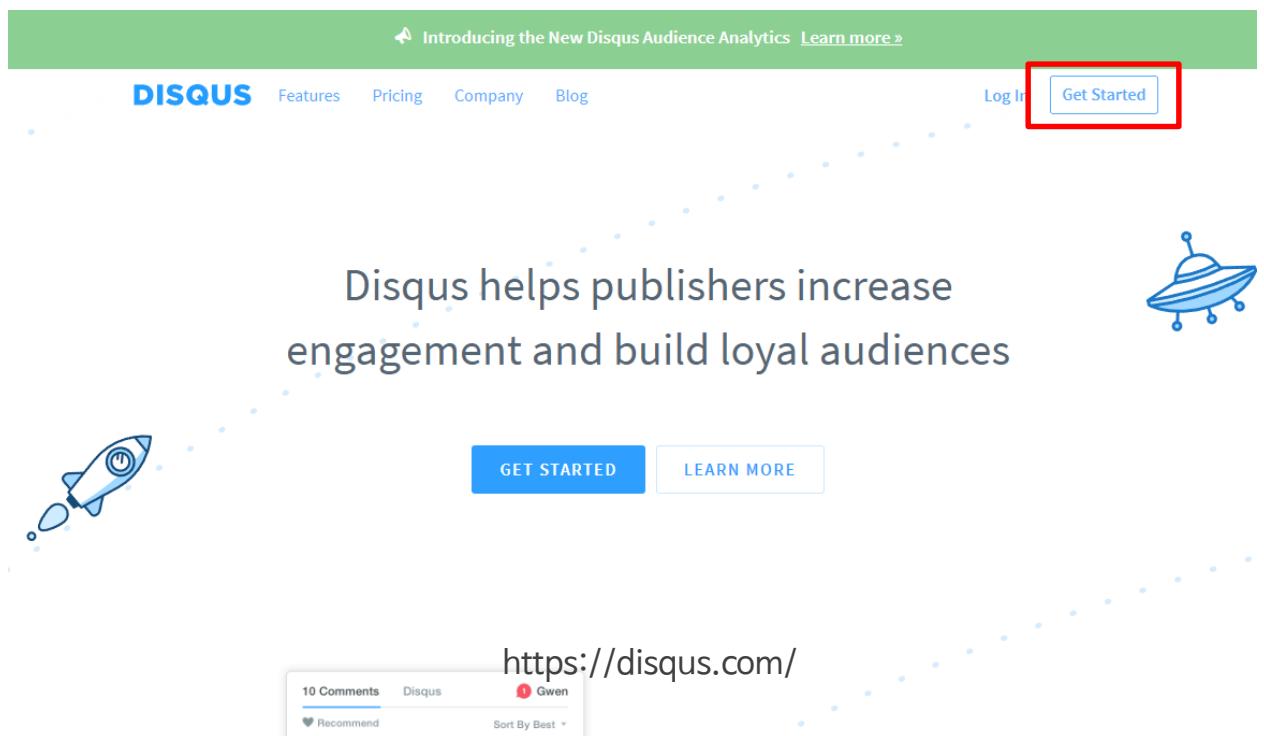
1. 댓글 기능 만들기 (계속)

이제 댓글을 달 수 있도록 설정해 보도록 하겠습니다 아래 명령어를 사용하여 disqus를 Django 내에 설치할 수도 있지만 우리는 좀 더 편한 방법을 사용해 보도록 하겠습니다.

disqus 설치하기

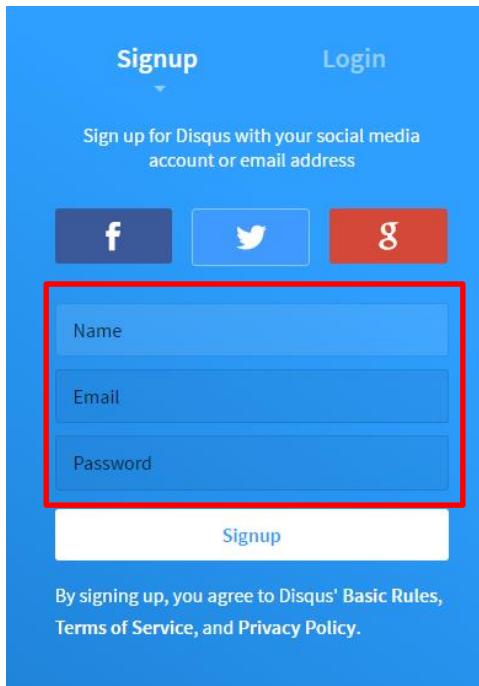
(myvenv)root@goorm:/workspace/컨테이너명/mysite# pip install django-disqus

* 우리는 설치하지 않는 방법을 사용하도록 하겠습니다. 따라서 위 명령어를 사용하지는 않습니다. 그러나 설치하셔서 사용하시는 분도 계시기 때문에 명시해 두도록 하겠습니다.



1. 댓글 기능 만들기 (계속)

이제 댓글을 달 수 있도록 설정해 보도록 하겠습니다 아래 명령어를 사용하여 disqus를 Django 내에 설치할 수도 있지만 우리는 좀 더 편한 방법을 사용해 보도록 하겠습니다.



Name, E-mail 등은 관리하기 쉽도록
구름 IDE ID, PW 사용

Disqus helps publishers increase
engagement and build loyal audiences



1. 댓글 기능 만들기 (계속)

아래 내용을 작성하신 다음(작성하는 내용이 중요하지는 않습니다.) Create Site를 눌러주세요.
우리는 BASIC으로 설치를 하도록 하겠습니다.

Create a new site
All fields are required.

Site Owner  paul lab
To associate a different account as the site owner,
[login with a different account](#)

Website Name
Your unique disqus URL will be: tutorialdjango.disqus.com
[Customize Your URL](#)

Category

Language

Create Site

Basic
Free, Ads Supported
Any number of total daily pageviews
Get all of the core Disqus features with the option to configure ads.

- ✓ Comments Plug-in
- ✓ Advanced Spam Filters
- ✓ Moderation Tools
- ✓ Basic Analytics
- ✓ Configurable Ads

[Learn more about Disqus' Basic features](#)

Plus
\$10 \$9 per month
Under 50,000 total daily pageviews
Get everything in Disqus Basic and the option to turn ads on and off.

- ✓ Everything In Basic
- ✓ Direct Support
- ✓ Ads Optional

Pro
\$99 \$89 per month
Under 150,000 total daily pageviews
Get everything in Disqus Basic, Plus, and extra premium features.

- ✓ Everything in Plus
- ✓ Priority Support
- ✓ Single Sign-On
- ✓ Advanced Analytics
- ✓ Shadow Banning
- ✓ Timeouts
- ✓ Email Subscriptions

[Learn more about Disqus' Pro features](#)

POPULAR

Subscribe Now

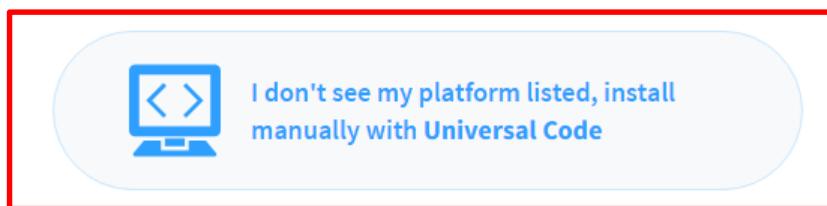
No credit card required **Start Trial (30 days)**

Free
For small, personal, non-commercial sites who do not run any ads. This plan includes everything in Plus except Direct Email Support.

Subscribe Now

1. 댓글 기능 만들기 (계속)

다음을 누르시면 여러 플랫폼이 나오는데 맨 아래 있는 것을 클릭하겠습니다. 그런 다음 1번이 나와 있는 모든 소스를 복사해두세요. (나중에 홈페이지에 붙여넣을 것입니다.)



1

Place the following code where you'd like Disqus to load:

```
<div id="disqus_thread"></div>
<script>

/**
 * RECOMMENDED CONFIGURATION VARIABLES: EDIT AND UNCOMMENT THE SECTION BELOW TO INSERT
 * DYNAMIC VALUES FROM YOUR PLATFORM OR CMS.
 * LEARN WHY DEFINING THESE VARIABLES IS IMPORTANT:
 * https://disqus.com/admin/universalcode/#configuration-variables*/
 */

var disqus_config = function () {
this.page.url = PAGE_URL; // Replace PAGE_URL with your page's canonical URL variable
this.page.identifier = PAGE_IDENTIFIER; // Replace PAGE_IDENTIFIER with your page's unique identifier
variable
}
```

A screenshot of the Disqus configuration settings page. The page title is 'Configure Disqus'. It contains various input fields for site configuration: Appearance (Color scheme: Auto, Typeface: Auto), Website Name (tutorialdjango), Website URL, Comment Policy URL, Comment Policy Summary (example: We welcome relevant and respectful comments. Off-topic comments may be removed.), Category (Tech), Description, and Language (Korean). At the bottom right, a blue button labeled 'Complete Setup' is highlighted with a red box.

1. 댓글 기능 만들기 (계속)

작동하는지 한번 붙여보겠습니다. Postdetails.html로 오셔서 body 닫기 태그 위에 해당 소스를 붙여주세요. (주석은 보기 편하게 일부러 지웠습니다.)

tutorialdjango/mysite/main/templates/main/cafedetails.html

```
<!DOCTYPE html>
<html>
<head>
    <title>cafelist</title>
</head>
<body>
    <h1>cafedetails</h1>
    <p>{{cafeobj.name}}</p>
    <p>{{cafeobj.content|linebreaks}}</p>
    {% if cafeobj.mainphoto %}
        
    {% endif %}
    <a href="https://jejucodingcampsample-iqusj.run.goorm.io/cafelist/">목록</a>
    <div id="disqus_thread"></div>
<script>
(function() { // DON'T EDIT BELOW THIS LINE
var d = document, s = d.createElement('script');
s.src = 'https://codingcampsamplepage.disqus.com/embed.js';
s.setAttribute('data-timestamp', +new Date());
(d.head || d.body).appendChild(s);
})();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_noscript">comments powered by Disqus.</a></noscript>
</body>
</html>
```

1. 댓글 기능 만들기 (계속)

실행된 화면입니다. 목록 아래 댓글창이 달립니다.

cafedetails

cafename1

cafecon1



목록

0 Comments codingcampsamplepage

2 Hojun Lee

Recommend

Tweet

Share

Sort by Best



Start the discussion...

Be the first to comment.

Subscribe

Add Disqus to your site

Disqus' Privacy Policy

DISQUS

1. 댓글 기능 만들기 (계속)

이번에는 게시된 날짜와 수정된 날짜를 추가해 보도록 하겠습니다.

tutorialdjango/mysite/main/models.py

```
from django.db import models

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    mainphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    content = models.TextField()

    def __str__(self):
        return self.name
```

```
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py makemigrations
Migrations for 'main':
  main/migrations/0003_auto_20190101_0707.py
    - Add field modifiedDate to cafe
    - Add field publishedDate to cafe
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions
Running migrations:
  Applying main.0003_auto_20190101_0707... OK
```

1. 댓글 기능 만들기 (계속)

게시된 날짜와 수정된 날짜가 추가되어 있는 것을 확인할 수 있습니다.

Home > Main > Cafes > cafename1

Change cafe HISTORY

Name:

Mainphoto: Currently: [Clear](#)
Change: [파일 선택](#) 선택된 파일 없음

PublishedDate: Date: Today
Time: Now |

Note: You are 9 hours ahead of server time.

ModifiedDate: Date: Today
Time: Now |

Note: You are 9 hours ahead of server time.

Content:

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)

1. 댓글 기능 만들기 (계속)

올린 모든 게시물들을 수정해 보도록 하겠습니다.

Home > Main > Cafes > cafename1

Change cafe

HISTORY

Name:

Mainphoto: Currently: cafe_0.jpg Change: 선택된 파일 없음

PublishedDate: Date: Today | Time: Now | Note: You are 9 hours ahead of server time.

ModifiedDate: Date: Today | Time: Now | Note: You are 9 hours ahead of server time.

Content:

1. 댓글 기능 만들기

이번에는 blog에 수정된 날짜를 함께 올려보도록 하겠습니다.

tutorialdjango/mysite/main/templates/main/cafe.html

```
<!DOCTYPE html>
<html>
<head>
    <title>cafe</title>
</head>
<body>
    <h1>cafe</h1>
    <table>
        {% for list in cafelistobj %}
            <tr onclick="location.href='https://jeiusjusj.run.goorm.io/cafe/{list.pk}'">
                <td>{{list.name}}</td>
                <td>{{list.content}}</td>
                <td>{{list.publishedDate}}</td>
                <td>{{list.modifiedDate}}</td>
            </tr>
        {% endfor %}
    </table>
</body>
</html>
```

<https://tutorialdjango-bcrpr.run.goorm.io/cafe/>

cafe

cafe
cafe
cafe

| name | content | publishedDate | modifiedDate |
|----------|----------|-------------------------|-------------------------|
| cafecon1 | cafecon1 | Jan. 1, 2019, 7:09 a.m. | Jan. 1, 2019, 7:09 a.m. |
| cafecon2 | cafecon2 | Jan. 1, 2019, 7:10 a.m. | Jan. 1, 2019, 7:10 a.m. |
| cafecon3 | cafecon3 | Jan. 1, 2019, 7:10 a.m. | Jan. 1, 2019, 7:10 a.m. |

2. 태그 기능 만들기 (계속)

이번에는 태그를 달 수 있는 패키지은 taggit을 설치 해보도록 하겠습니다. 태그를 이용하여 검색이 가능하도록 구현할 수 있으며 태그의 별도 페이지를 만들 수도 있습니다.

(myvenv)root@goorm:/workspace/컨테이너명/mysite# pip3 install django-taggit

```
^C(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# pip3 install django-taggit
Collecting django-taggit
  Downloading https://files.pythonhosted.org/packages/39/ed/3607874788740674b7533586fc6b3
9585882f0c8709e9a8c36/django_taggit-0.23.0-py2.py3-none-any.whl (44kB)
    100% |██████████| 51kB 713kB/s
Requirement already satisfied: Django>=1.11 in ./myvenv/lib/python3.6/site-packages (from
taggit) (2.1)
Requirement already satisfied: pytz in ./myvenv/lib/python3.6/site-packages (from Django>
taggit) (2018.7)
Installing collected packages: django-taggit
Successfully installed django-taggit-0.23.0
```

tutorialdjango/mysite/main/models.py

```
from django.db import models
from taggit.managers import TaggableManager

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    mainphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    content = models.TextField()
    tag = TaggableManager(blank=True)

    def __str__(self):
        return self.name
```

2. 태그 기능 만들기 (계속)

App을 설치하였으니 settings에서 taggit을 추가해 주도록 하겠습니다. DB에 반영하기 위해 makemigrations 명령과 migrate 명령을 사용합니다.

```
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'main',
41     'taggit',
42 ]
```

tutorialdjango/mysite/tutorialdjango/settings.py

```
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py makemigrations
Migrations for 'main':
 main/migrations/0004_cafe_tag.py
 - Add field tag to cafe
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py migrate
Operations to perform:
 Apply all migrations: admin, auth, contenttypes, main, sessions, taggit
Running migrations:
 Applying taggit.0001_initial... OK
 Applying taggit.0002_auto_20150616_2121... OK
 Applying main.0004_cafe_tag... OK
```

2. 태그 기능 만들기 (계속)

Tag가 작동하는지 확인하기 위해 admin page로 들어오면 하단에 tags라고 tag가 모여진 것을 볼 수 있습니다. 또한 각각에 페이지에 tag 입력 폼이 달린 것을 볼 수 있습니다. 우리는 이곳에 포스트 순서대로 ‘애월읍, 제주시, 조천읍’을 입력할 것입니다.

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups + Add Change
Users + Add Change

MAIN

Cafes + Add Change

TAGGIT

Tags + Add Change

Home > Main > Cafes > cafename1

Change cafe

HISTORY

Name: cafename1

Mainphoto: Currently *cafe_0.jpg* Clear
Change: 파일 선택 선택된 파일 없음

PublishedDate: Date: 2019-01-01 Today |
Time: 07:09:26 Now |

Note: You are 9 hours ahead of server time.

ModifiedDate: Date: 2019-01-01 Today |
Time: 07:09:27 Now |

Note: You are 9 hours ahead of server time.

Content: cafecon1

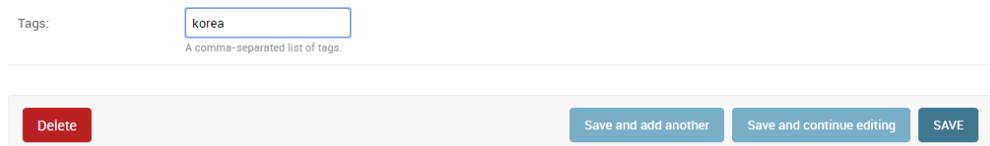
Tags: 애월읍

A comma-separated list of tags.

Delete Save and add another Save and continue editing SAVE

2. 태그 기능 만들기 (계속)

Tags를 입력하고 SAVE를 누른다음 cafedetails에서 출력을 해보도록 하겠습니다.



tutorialdjango/mysite/main/templates/main/postdetails.html

```
<body>
    <h1>cafedetails</h1>
    <p>{{cafeobj.name}}</p>
    <p>{{cafeobj.content|linebreaks}}</p>
    {% if cafeobj.mainphoto %}
        
    {% endif %}
    <p>{{cafeobj.tag.names}}</p>
    <a href="https://jecodingcampsample-iqus:</a>
        <div id="disqus_thread"></div>
    <script>
        (function() { // DON'T EDIT BELOW THIS LINE
            var d = document, s = d.createElement('script');
            s.src = 'https://codingcampsamplepage.disqus.com/embed.js';
            s.setAttribute('data-timestamp', +new Date());
            (d.head || d.body).appendChild(s);
        })();
    </script>
    <noscript>Please enable JavaScript to view the </noscript>
    comments powered by Disqus.</a>
</body>
```

2. 태그 기능 만들기 (계속)

출력은 되는데 Query Set이라는 문구가 붙습니다.

cafedetails

cafename1

cafecon1



<QuerySet ['애월읍']>

목록

0 Comments codingcampsamplepage

2 Hojun Lee -

Heart Recommend

[Tweet](#)

[Share](#)

Sort by Best ▾



Start the discussion...

Be the first to comment.

[Subscribe](#)

[Add Disqus to your site](#)

[Disqus' Privacy Policy](#)

DISQUS

2. 태그 기능 만들기 (계속)

for문으로 순회를 하면 여러 개의 태그도 출력할 수 있습니다. 우선은 1개만 출력해보도록 하겠습니다. 뒤에 .0을 붙이면 가장 첫번째 것을 출력합니다.

```
<h1>cafedetails</h1>
<p>{{cafeobj.name}}</p>
<p>{{cafeobj.content|linebreaks}}</p>
{%
  if cafeobj.mainphoto %
    
  {% endif %}
<p>{{cafeobj.tag.names.0}}</p>
<a href="https://jejucodingcampsample-iqus:</a>
```

애월읍

목록

0 Comments

codingcampsamplepage

 Recommend

 Tweet

 Share



Start the discussion...

2. 태그 기능 만들기 (계속)

이번에는 경도와 위도를 표현할 수 있도록 models.py 파일을 다뤄보도록 하겠습니다.
models.py 파일을 수정하면 makemigrations와 migrate를 해야 한다는 사실 잊지 않으셨죠?

```
from django.db import models
from taggit.managers import TaggableManager

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    lat = models.FloatField(null=True)
    lng = models.FloatField(null=True)
    mainphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    content = models.TextField()
    tag = TaggableManager(blank=True)

    def __str__(self):
        return self.name
```

```
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py makemigrations
Migrations for 'main':
    main/migrations/0005_auto_20190101_0806.py
        - Add field lat to cafe
        - Add field lng to cafe
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py migrate
Operations to perform:
    Apply all migrations: admin, auth, contenttypes, main, sessions, taggit
Running migrations:
    Applying main.0005_auto_20190101_0806... OK
```

2. 태그 기능 만들기 (계속)

샘플로 구글맵에서 각 지역을 대표하는 장소를 검색하여 좌표를 알아낸 다음 아래와 같이 입력합니다.

1번 게시물 - 33.4619638,126.3273297 - 애월읍사무소

2번 게시물 - 33.6628574,126.0112614 - 제주시청

3번 게시물 - 33.5380993,126.631991 - 조천읍사무소

Home > Main > Cafes > cafename1

Change cafe

| | |
|------------|---|
| Name: | cafename1 |
| Lat: | 33.4619638 |
| Lng: | 126.3273297 |
| Mainphoto: | Currently: cafe_0.jpg <input type="button" value="Clear"/> Change: <input type="button" value="파일 선택"/> 선택된 파일 없음 |

Home > Main > Cafes > cafename2

Change cafe

| | |
|------------|---|
| Name: | cafename2 |
| Lat: | 33.6628574 |
| Lng: | 126.0112614 <input type="button" value="▼"/> |
| Mainphoto: | Currently: cafe_1.jpg <input type="button" value="Clear"/> Change: <input type="button" value="파일 선택"/> 선택된 파일 없음 |

Home > Main > Cafes > cafename3

Change cafe

| | |
|------------|---|
| Name: | cafename3 |
| Lat: | 33.5380993 |
| Lng: | 126.631991 |
| Mainphoto: | Currently: cafe_2.jpg <input type="button" value="Clear"/> Change: <input type="button" value="파일 선택"/> 선택된 파일 없음 |

2. 태그 기능 만들기 (계속)

이번에는 사진 한장을 더 추가할 수 있도록 models.py 파일을 수정하겠습니다. 마찬가지로 makemigrations와 migrate를 입력해주세요.

```
from django.db import models
from taggit.managers import TaggableManager

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    lat = models.FloatField(null=True)
    lng = models.FloatField(null=True)
    mainphoto = models.ImageField(blank=True, null=True)
    subphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    content = models.TextField()
    tag = TaggableManager(blank=True)

    def __str__(self):
        return self.name
```

```
Migrations for 'main':
main/migrations/0006_cafe_subphoto.py
  - Add field subphoto to cafe
```

```
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions, taggit
Running migrations:
  Applying main.0006_cafe_subphoto... OK
```

2. 태그 기능 만들기 (계속)

이번에는 cafelist에서 필요한 항목들을 추가해 보도록 하겠습니다. 마찬가지로 makemigrations와 migrate를 입력해주세요.

* 완료된 후 Admin 창에서 임의의 값들을 넣어놓겠습니다.

```
from django.db import models
from taggit.managers import TaggableManager

class Cafe(models.Model):
    name = models.CharField(max_length = 50)
    lat = models.FloatField(null=True)
    lng = models.FloatField(null=True)
    mainphoto = models.ImageField(blank=True, null=True)
    subphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    content = models.TextField()
    locate = models.TextField(null=True)
    phone = models.TextField(null=True)
    insta = models.TextField(null=True)
    tag = TaggableManager(blank=True)

    def __str__(self):
        return self.name
```

```
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py makemigrations
Migrations for 'main':
  main/migrations/0007_auto_20190103_0407.py
    - Add field insta to cafe
    - Add field locate to cafe
    - Add field phone to cafe
(myvenv) root@goorm:/workspace/jejucodingcampsample/mysite# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions, taggit
Running migrations:
  Applying main.0007_auto_20190103_0407... OK
```

2. 태그 기능 만들기

공식문서 :

<https://docs.djangoproject.com/en/2.1/ref/models/fields/#django.db.models.FloatField>

공식문서에 보시면 blank와 null에 관련된 내용이 있습니다.

null

If **True**, Django will store empty values as **NULL** in the database. Default is **False**.

blank

If **True**, the field is allowed to be blank. Default is **False**.

Note that this is different than **null**. **null** is purely database-related, whereas **blank** is validation-related. If a field has **blank=True**, form validation will allow entry of an empty value. If a field has **blank=False**, the field will be required.

공식문서에 null과 blank에 대한 내용

Step 6

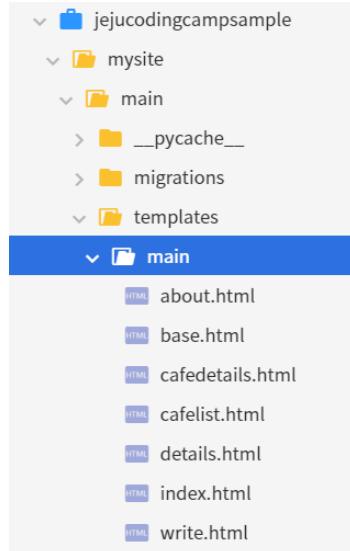
Django에 Template 입히기

1. Django Template 입히기 (계속)

이제 전에 들었던 수업의 UI를 Django에 입히는 작업이 남아 있습니다. 우선 아래 링크에서 파일을 다운로드 받아

1. css폴더, js폴더, images폴더를 서버에 static 폴더로 가져옵니다.
2. '.html' 파일을 열어 모든 내용을 복사해서 main > '.html'에다 넣겠습니다.

* 우클릭 하여 '파일 가져오기/업로드' 버튼을 누르신 다음 파일을 선택하셔서 업로드하시면 됩니다.

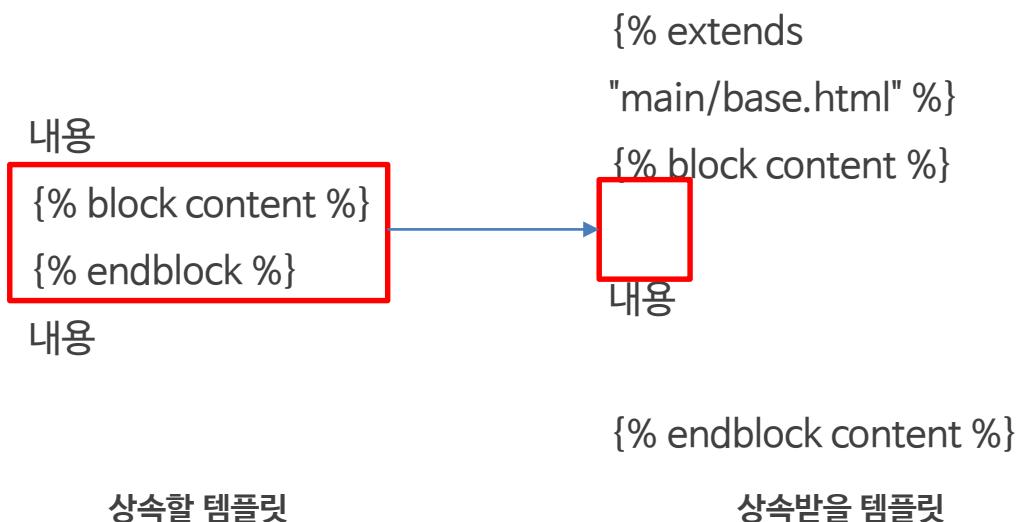


* 모든 templates 소스는 paullab.co.kr/jejucodingcampsample.zip에 있습니다.

1. Django Template 입히기 (계속)

템플릿 상속:

업로드 한 파일이 10개, 메뉴가 바뀐다면 10개의 파일을 수정해야 하는 상황이 발생됩니다. 이러한 상황을 막기 위해 Django에서는 템플릿 상속을 지원합니다. 하나의 템플릿을 만들어 여러 개의 템플릿에서 상속받을 수 있도록 하면 메뉴를 하나의 템플릿으로 만들어 다른 파일이 상속할 수 있게 한 것이죠.



1. Django Template 입히기 (계속)

업로드한 템플릿 중

1. about.html 파일 URL 연결
2. write 파일 URL 연결

을 해보도록 하겠습니다.

```
from django.contrib import admin
from django.urls import path
from main.views import index, about, write, cafelist, cafedetails
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index),
    path('about/', about),
    path('write/', write),
    path('cafelist/', cafelist),
    path('cafelist/<int:pk>', cafedetails),
]
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

views.py

from django.shortcuts import render
from .models import Cafe

def index(request):
    return render(request, 'main/index.html')

def about(request):
    return render(request, 'main/about.html')

def write(request):
    return render(request, 'main/write.html')
```

urls.py

1. Django에 Template 입하기 (계속)

템플릿 테그, 템플릿 변수, 기타 기능 구현은 라이브로 진행하도록 하겠습니다.

The screenshot shows the homepage of the Jeju Cafe Search website. At the top, there's a blue header bar with the text "제주카페찾기" (Jeju Cafe Search) on the left, followed by "서비스 소개", "카페들", and "카페등록" (Create Cafe). On the right side of the header is a search bar with the placeholder "찾고 싶은 카페 이름" (Name of the cafe you want to find) and a magnifying glass icon. Below the header is a large, bold title "지도에서 원하는 지역을 선택하세요!" (Select the desired area on the map!). Underneath this title is a smaller subtitle "선택하지 않으면 전체 지역에서 찾습니다." (If not selected, it will search the entire region). The main feature is a map of Jeju Island with various districts labeled: Seogwipo (서귀포), Hallim-eup (한림읍), Hwaseong-eup (화성읍), and surrounding areas like Gwakji-eup (곽지읍), Dongdo-myeon (동도면), and Seogwipo-si (제주시). Below the map is a large blue button with the text "카페찾기!" (Find Cafe!).



[서비스 소개](#)[카페들](#)[+ 카페등록](#)

검색

제주카페찾기 | 서비스 소개 | 카페들 | + 카페등록 | 찾고 싶은 카페 이름 | 🔎

제주카페찾기 | 서비스 소개 | 카페들 + 카페등록 | 찾고 싶은 카페 이용 | Q.



cafename1

개발자와 디자이너들을 위한 카페예요.

- cafe1번 주소
 - cafe1번 번호
 - cafe1번 인스타



Caption Text

- cafe2번 주소
 - cafe2번 번호
 - cafe2번 인스턴스



cafename2

개발자와 디자이너들을 위한 카페에요.

- 📍 cafe2번 주소
 - 📞 cafe2번 전화번호
 - 👤 cafe2번 인스타



Three Christmas ornaments are arranged together. On the left is a Santa Claus figure with a red suit, white beard, and a red and white striped hat. In the center is a snowman with a black top hat, a red pom-pom nose, and a green scarf. On the right is a reindeer with a brown coat, large black antlers, and a red collar.

cafename3

www.w3.org/2001/XMLSchema#

- cafe3번 주소
• cafe3번 전화번호

1. Django Template 입하기

검색 기능 구현을 위한 소스코드입니다. 주석은 일부러 남겨둔 것이니 나중에 주석을 풀고 실행해보세요.

```
def index(request):
    if request.method == "POST":
        #print(request)
        #print(request.POST.keys())

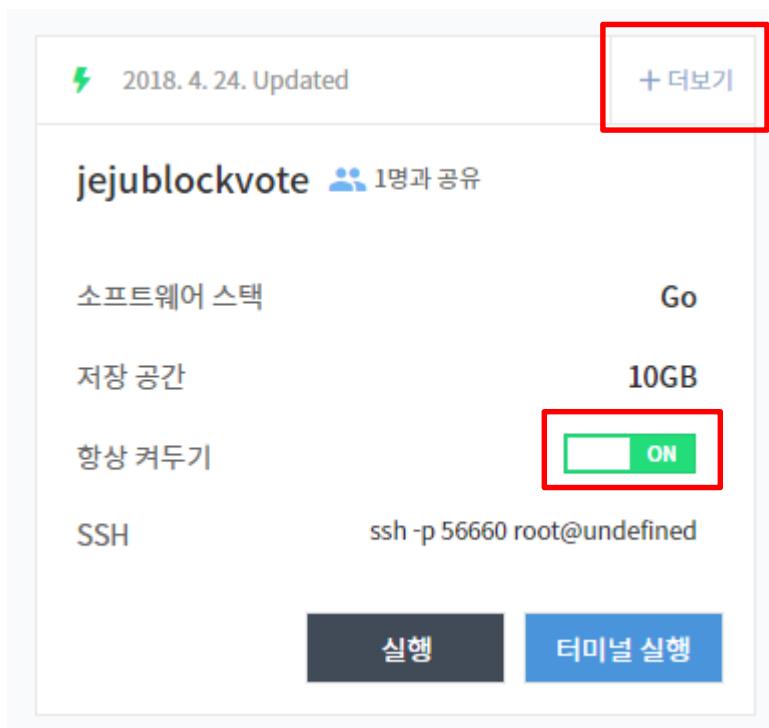
        keys = list(request.POST.keys())
        keys.remove('csrfmiddlewaretoken')
        print(keys[0])

        cafelistobj = []
        cafelistobjall = Cafe.objects.all()
        #print(cafelistobj)
        #print(type(cafelistobj))
        #print(dir(cafelistobj))

        for key in keys:
            for cafelist in cafelistobjall:
                #print(cafelist['tag'], key, cafelist.tag == key)
                cafetag = list(cafelist.tag.values())
                print(cafetag[0]['name'], key, cafetag[0]['name'] == key)
                if cafetag[0]['name'] == key:
                    cafelistobj.append(cafelist)
        print(cafelistobj)
        return render(request, 'main/cafelist.html', {'cafelistobj':cafelistobj})
    return render(request, 'main/index.html')
```

2. 서비스 배포 (계속)

구름 IDE에서는 항상 켜두기 기능을 프리미엄 계정에서만 제공되고 있습니다. 클릭만 하시면 항상 켜두기가 활성화 됩니다. 더보기를 클릭하셔서 다음 장으로 넘어가세요.



2. 서비스 배포 (계속)

실행 URL과 포트에서 추가를 누르시고 구매하신 URL을 입력합니다. 자동실행 스크립트는 Django가 실행되기 위한 최소한의 스크립트를 적어놓는 것이 좋습니다. 저는 주로을 넣어놓고 실행해 놓습니다.

```
source /workspace/컨테이너이름/mysite/myvenv/bin/activate  
python3 /workspace/컨테이너이름/mysite/manage.py runserver 0:80
```

The screenshot shows the 'Container Settings' page for a container named 'jejucodingcampsample'. The 'Auto-execution Script' section contains two lines of code:

```
source /workspace/jejucodingcampsample/mysite/myvenv/bin/activate  
python3 /workspace/jejucodingcampsample/mysite/manage.py runserver 0:80
```

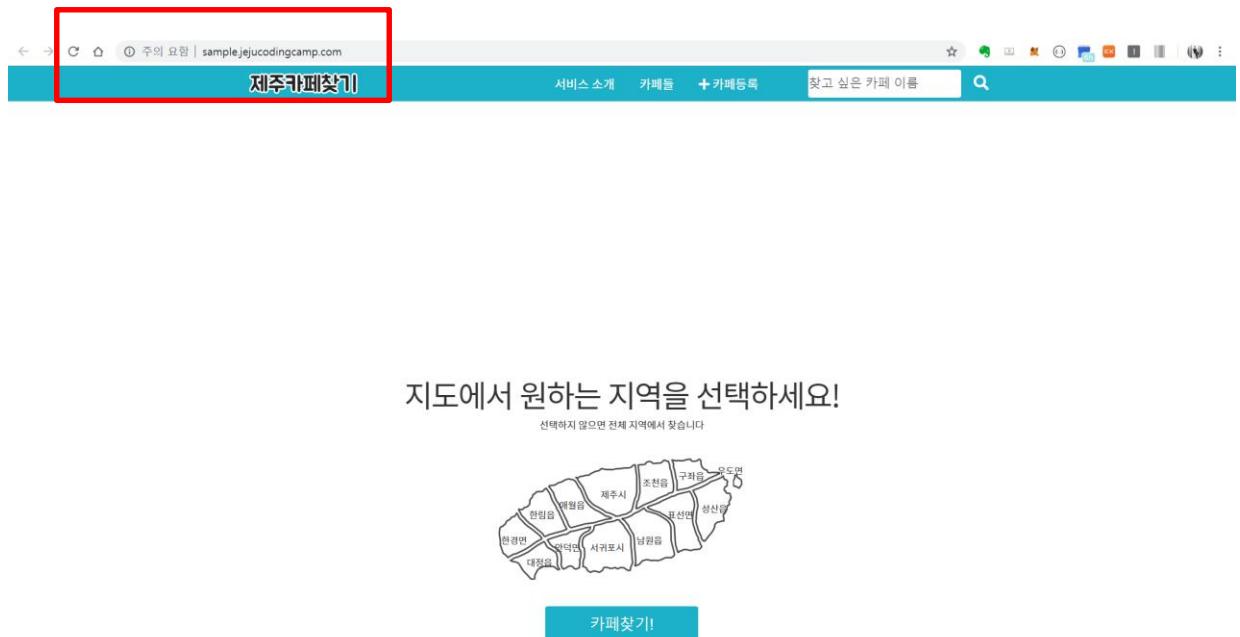
The 'Network Management' section shows a single entry:

| URL | Port |
|---|------|
| jejucodingcampsample-iqusj.run.goorm.io | 8080 |
| sample.jejuencodingcamp.com | 80 |

2. 서비스 배포

URL이 제대로 연결되고 작동하는지 확인해보세요.

<http://sample.jejucodingcamp.com/>



Step 7

요약정리

TUTORIAL 0 Django

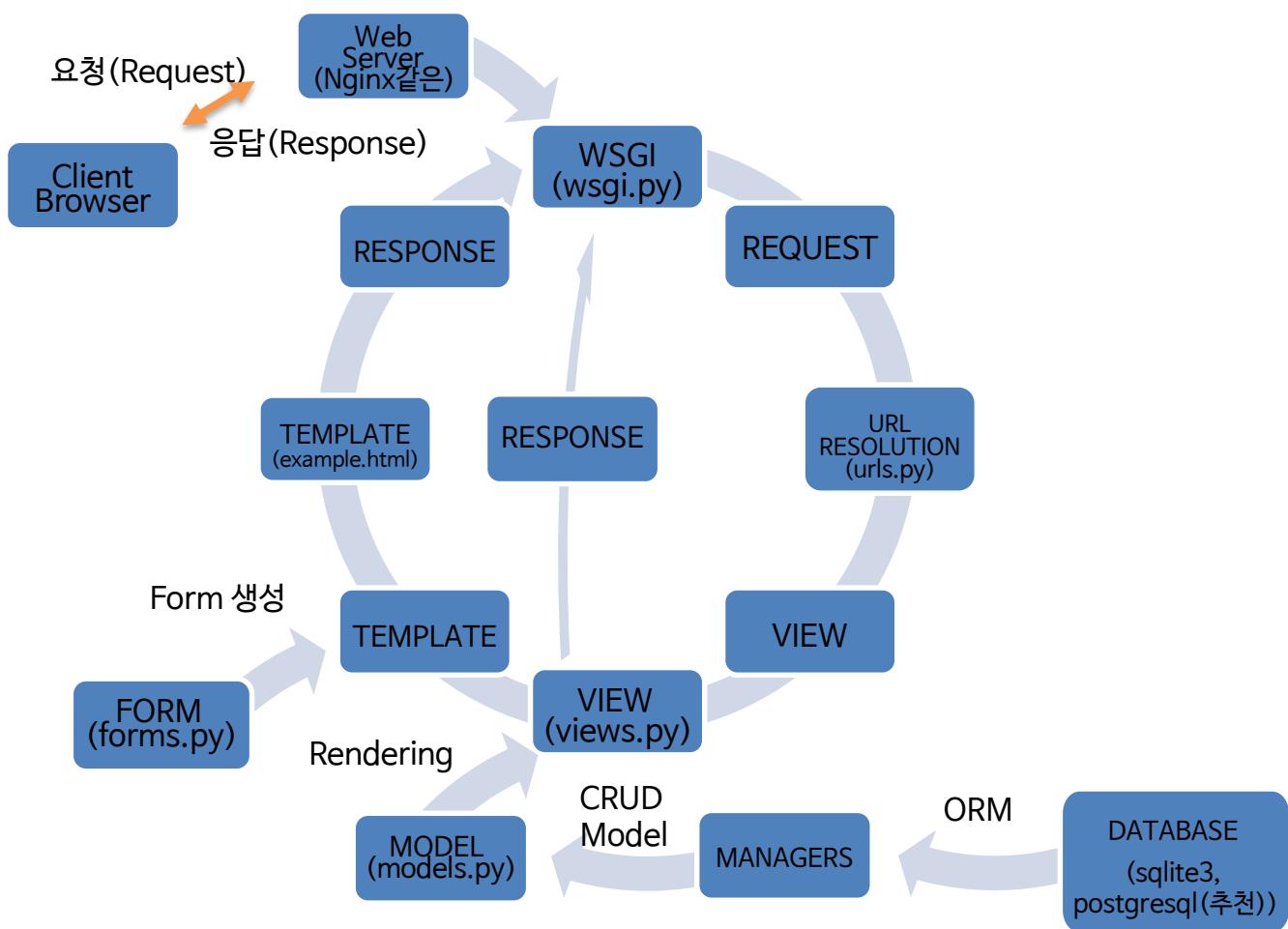
Django는 Python으로 빠르게 웹을 개발하기 위해 나온 Web Framework입니다. 대부분의 기능들이 자동화 되어 있으며 공식문서도 잘 되어 있어 공식문서 정독을 한번 하면 다른 문서는 필요 없을 정도로 상세 내용을 가지고 있습니다.

우리는 지금까지 간단한 여행 블로그를 만들어 보았습니다. 그러나 Django에 극히 일부분만을 다룬 것입니다. 더 강력한 Django의 기능을 사용하고 싶으시다면 지금 바로 공식문서를 정독하시길 권해드립니다.

아래는 Django의 기능을 전체 도식화 정리 해놓은 것입니다. 보시면서 놓치신 것이 없으신지 체크해보시기 바랍니다.

* 해당 책에서는 form에 대해 다루지 않았습니다.

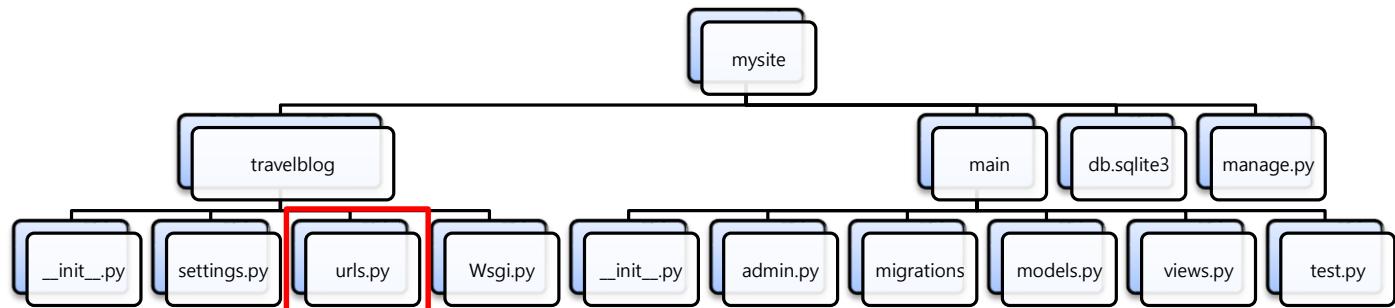
Django



TUTORIAL 1 URL 설계

챕터에서 수정한 순서대로 빨간색 네모를 그려보았습니다. 처음에는 urls.py에서 url 파싱을 하였습니다. 아래 코드는 우리가 수정한 코드 전문입니다.

Urls.py



```
from django.contrib import admin
from django.urls import path
from main.views import index, blog, postdetails
from django.conf.urls.static import static
from django.conf import settings

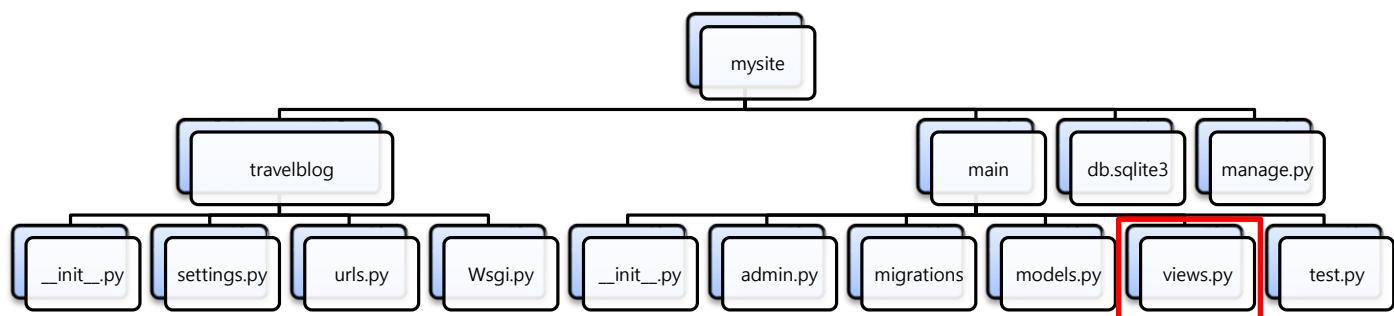
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index),
    path('blog/', blog),
    path('blog/<int:pk>', postdetails),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

TUTORIAL 2 VIEW 로직 설계

URL 설계가 끝났다면 URL에서 연결해준 각종 함수를 views에서 만들어 줍니다.

views.py



```
from django.shortcuts import render
from .models import Post

def index(request):
    postlist = Post.objects.all()
    return render(request, 'main/index.html', {'postlist':postlist})

def blog(request):
    postlist = Post.objects.all()
    return render(request, 'main/blog.html', {'postlist':postlist})

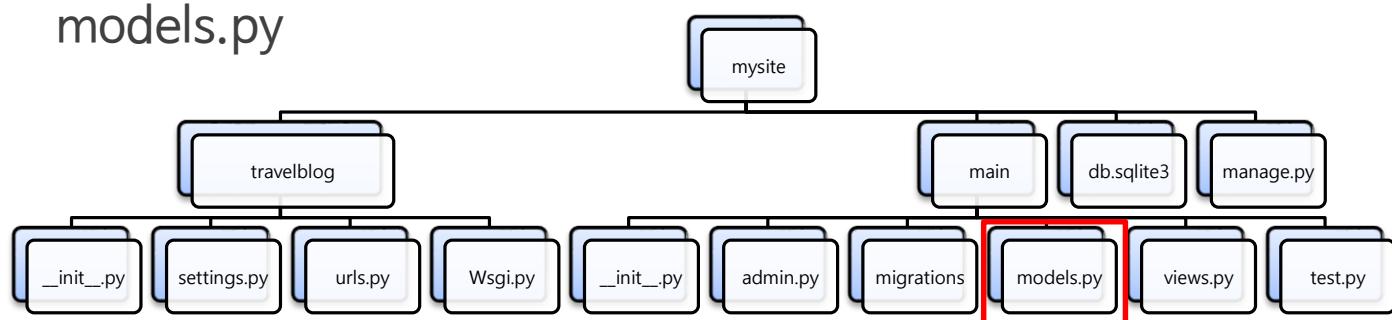
def postdetails(request, pk):
    postlist = Post.objects.get(pk=pk)
    return render(request, 'main/postdetails.html', {'postlist':postlist})
```

TUTORIAL 3 MODEL 데이터베이스 설계

아래 코드는 후에 작성하였지만 설계를 제대로 하셨다면 세번째에서 작업할 사항입니다. 각종 게시물 등 필요한 model을 설계합니다.

전부 작성한 후에는 makemigrations와 migrate로 DB에 반영합니다.

models.py



```
from django.db import models
from taggit.managers import TaggableManager

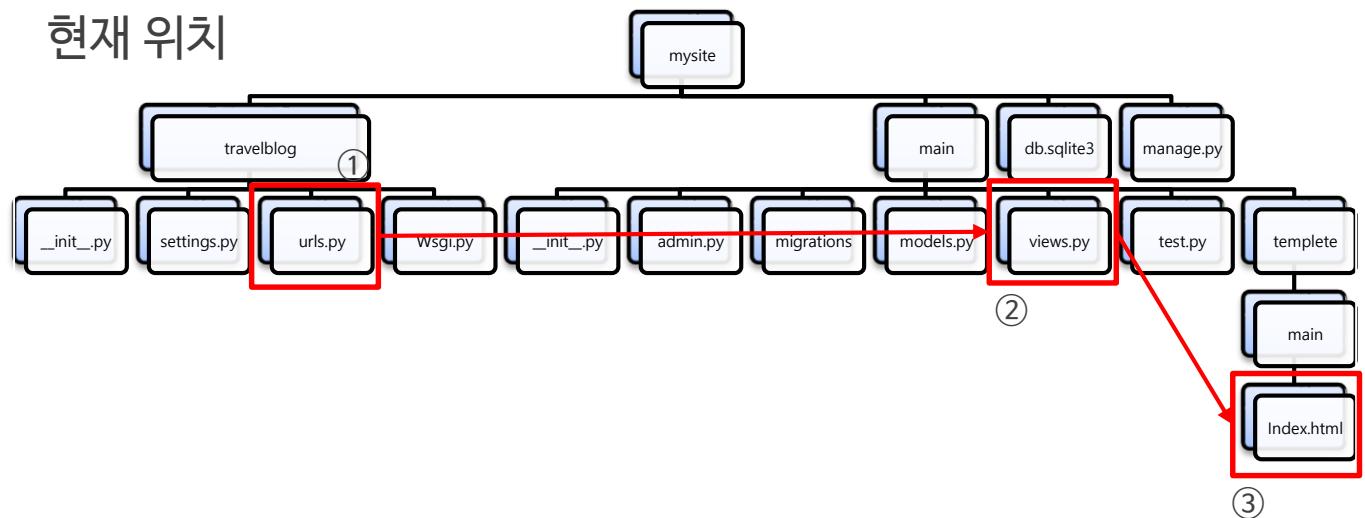
class Post(models.Model):
    postname = models.CharField(max_length=50)
    Lat = models.FloatField(null=True)
    Lng = models.FloatField(null=True)
    mainphoto = models.ImageField(blank=True, null=True)
    publishedDate = models.DateTimeField(blank=True, null=True)
    modifiedDate = models.DateTimeField(blank=True, null=True)
    contents = models.TextField()
    tag = TaggableManager(blank=True)

    def __str__(self):
        return self.postname
```

TUTORIAL 4 Tamplates

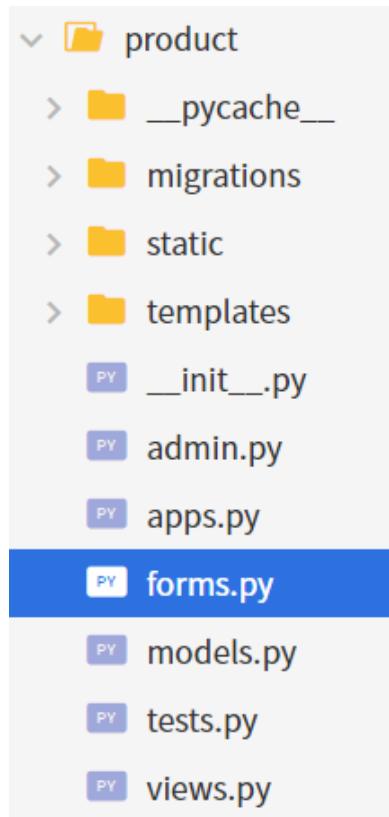
이번에는 Template을 작성 해보겠습니다. 중간에 모델 설계를 하지 않았다면 3번째 작업할 사항입니다. 우리는 templates라는 폴더 아래 main이라는 폴더를 만들어서 그 안에 각종 템플릿을 작성하였습니다.

현재 위치



TUTORIAL 5 PRODUCT 페이지에서 관리하기

(번외편 – 이 부분은 응용이 필요하신 분만 따라하세요.)



App 안에 form.py파일을 만들고 그 안에 아래와 같이 입력합니다.

```
1 from django import forms
2 from .models import Product
3
4 class PostForm(forms.ModelForm):
5     class Meta:
6         model = Product
7         fields = ('productname', 'contents',)
```

TUTORIAL 5 PRODUCT 페이지에서 게시물 작성

(번외편 – 이 부분은 응용이 필요하신 분만 따라하세요.)

```
1 from django.shortcuts import render
2 from .models import Product
3 from .forms import PostForm
4
5 # Create your views here.
6 def index(request):
7     return render(request, 'product/index.html')
8
9 def product(request):
10    productlist = Product.objects.all()
11    if request.method == "POST":
12        form = PostForm(request.POST)
13        if form.is_valid():
14            post = form.save(commit=False)
15            post.author = request.user
16            post.save()
17            return render(request, 'product/product.html', {'productlist':productlist})
18    else:
19        form = PostForm()
20    return render(request, 'product/product.html', {'form':form, 'productlist':productlist})
```

form으로 전달된 값이 POST인지 확인하여 처리합니다.

Product 페이지에서 게시물 작성하게 하기

(번외편 – 이 부분은 응용이 필요하신 분만 따라하세요.)

```
<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">submit</button>
</form>
```

form.as_p 템플릿 태그를 사용하면 Django가 모델에 있는 form을 템플릿으로 전달해줍니다.

Product 페이지에서 게시물 작성하게 하기

(번외편 – 이 부분은 응용이 필요하신 분만 따라하세요.)

Productname:

Contents:

form.as_p

• Productname:

• Contents:

form.as_ul

Productname: Contents:

form.as_table

코딩 기초에서 서비스 런칭까지 4박5일

제주 코딩 베이스캠프

코딩 기초부터 기획, 서비스 배포까지 4박 5일. 우리는 단지 열정 있는 청년들에게
열정만큼 성장할 수 있는 기회를. 서비스 구축 성공경험을 통해 이 취업난
시대에 자신감, 자존감, 성취감, 자립할 수 있는 능력을 배양할 수 있도록. 또,
이러한 경험이 단순히 금액이 높다는 이유로 또 한번에 좌절이 되지 않도록.
이 행사를 진행합니다.