

## KWEB Study: Week 0

1. 2학기 준스 안내
2. Back-end Intro
3. Node.js 소개
4. 개발환경 구축
5. 실습(?) 및 과제



# Back-end, start

KWEB 2학기 준회원 스터디



# 2학기에 당도한걸 환영하네!

- HTML을 넘고, CSS를 건너, Javascript를 찾아온 여러분 고생하셨습니다!
- 스터디는 아웃 카운트 뽀뽀하고.. 숙제도 귀찮고.. 하기 싫고 (\*같고..)
- 하지만.. 웹의 세계는 멀고도 멍니다. 웹은 방대하고 빠르게 변하는 분야니까요!
  - ✓ React.js, Angular, Vue.js, Node.js, koa.js, Express.js, SASS, LESS, Bootstrap, Spring, Django etc.
- 학교 교육과정에서는 배울 수 없는 실무와 가까운 이론과 실습을 KWEB에서 가져가실 수 있길 바랍니다!
- 준회원 여러분 2학기도 파이팅! (외주를 하는 그날까지.. 쓰읍)



# 이제 뭘 배울 것인가?

- 드디어 웹의 꽃! Back-end로 들어갑니다~ 서버 쪽!
- 2학기는 1학기와는 다르게 실무, 코딩 중심으로 스터디가 진행될 예정입니다.
  - ✓ 이론(40분) + 실습(40분), 총 80분
- Back-end 스터디에서 KWEB이 사용할 플랫폼은 요즘 가장 Hot하게 이용되는 Node.js 입니다.
- 오른쪽 그림들은 2학기 준스에서 다뤄볼 Topic들입니다!
- DB도 사용해볼 예정이니 여러분 만의 홈페이지를 하나 만 들어 가실 수 있길 바랍니다~





# 스터디 진행 Plan

- KWEB 준회원 스터디에서 Node.js 다루는 것은 올해가 두번째입니다. 작년 자료를 바탕으로 최신 경향을 담아 스터디 전체의 진행과정을 Renewal해 진행할 예정입니다.

주차	일시	내용
0주차	09.14.	Intro + Node.js Overview
1주차	09.17. - 09.21.	Module, Node.js 패키지 매니저, Git 기초
2주차	10.01. - 10.05.	ECMAScript & 비동기
3주차	10.08. - 10.12.	HTTP & HTTPS
4주차	11.05. - 11.09.	koa.js & Routing
5주차	11.12. - 11.16.	Database (NoSQL) & Restful API
6주차	11.19. - 11.23.	Database (RDB) & 프로젝트 I
7주차	11.26. - 11.30.	User Auth - Session & Token & 프로젝트 II
8주차	12.03. - 12.07.	Review & 프로젝트 II



# Back-end란?

- 웹 개발의 세계는 크게 두 진영으로 나뉘어집니다. 바로 Front-end와 Back-end입니다.
- Back-end: 프로그램의 뒷 부분, 즉 서버를 비롯하여 서버와 클라이언트 사이의 연결 다리를 개발하는 것
  - ✓ 서버 사이드(Server Side) 라고도 하며 이를 개발하는 사람을 백엔드 개발자, 서버 사이드 개발자라 합니다. (실제로 서버 개발자라 하는 경우도 있지만 서버 개발자는 보다 넓은 개념으로 사용됩니다!)
- Back-end 개발자의 경우 무수히 많은 언어가 있고 상황에 따라, 현장에 따라 각각 다른 언어를 사용합니다.
  - ✓ 한국에서는 스타트업을 제외하고는 Java, JSP, Spring, MySQL이 대표적인 Back-end 개발 스택입니다.
  - ✓ JSP의 대체로 PHP, ASP, 현재 전세계에서 가장 핫한 서버 사이드 플랫폼인 **Node.js** 등이 있습니다.
- (그럼 Front-end는 다 배웠나? 그건 아니고.. 크흠 1학기 내용은 WEB의 기본 언어들을 배웠다고 보시면 됩니다. 컴과로 따지면 컴프1,2 같은 포지션?)



# 잠깐, Front-end!

- Front-end: 프로세스의 처음, 즉 사용자가 프로그램에서 가장 처음으로 접하게 되는 부분을 개발하는 것
  - ✓ Client Side, 즉 사용자의 입장에서 설계되고 이용하고 싶은 UI를 만드는 것이 목표입니다.
  - ✓ 덧붙여 사용자에게 다양한 데이터를 여러가지 형태로 입력 받아 Back-end가 사용할 수 있는 규격으로 만들어 주는 것까지 Front-end 개발자의 역할입니다.
- 웹 퍼블리셔도 프론트엔드 개발자에 속하며, 프론트엔드 개발자에게는 디자인적인 감각, UI/UX에 대한 심도 있는 이해가 요구됩니다.
  - ✓ 프론트 엔드 개발자에게 필요한 언어는 사실 어느정도 구분된 범위에서 학습이 가능합니다.
  - ✓ 필수적으로 꼽는 HTML, CSS, Javascript 외에 다른 것들도 전문적으로 학습하면 큰 도움이 되고 필요한 것들이지만, 이 3가지에 대한 전문 지식이 충분하면 뛰어난 프론트 엔드 개발자라 할 수 있습니다.
- 요즘 웹 세계에서 흔히 들을 수 있는 Angular, React.js, Vue.js 들이 Front-end Framework 입니다.
- UI/UX의 중요성이 부각되는 현재에는 Front-end 역시 웹에서 중요한 축을 담당하고 있습니다.

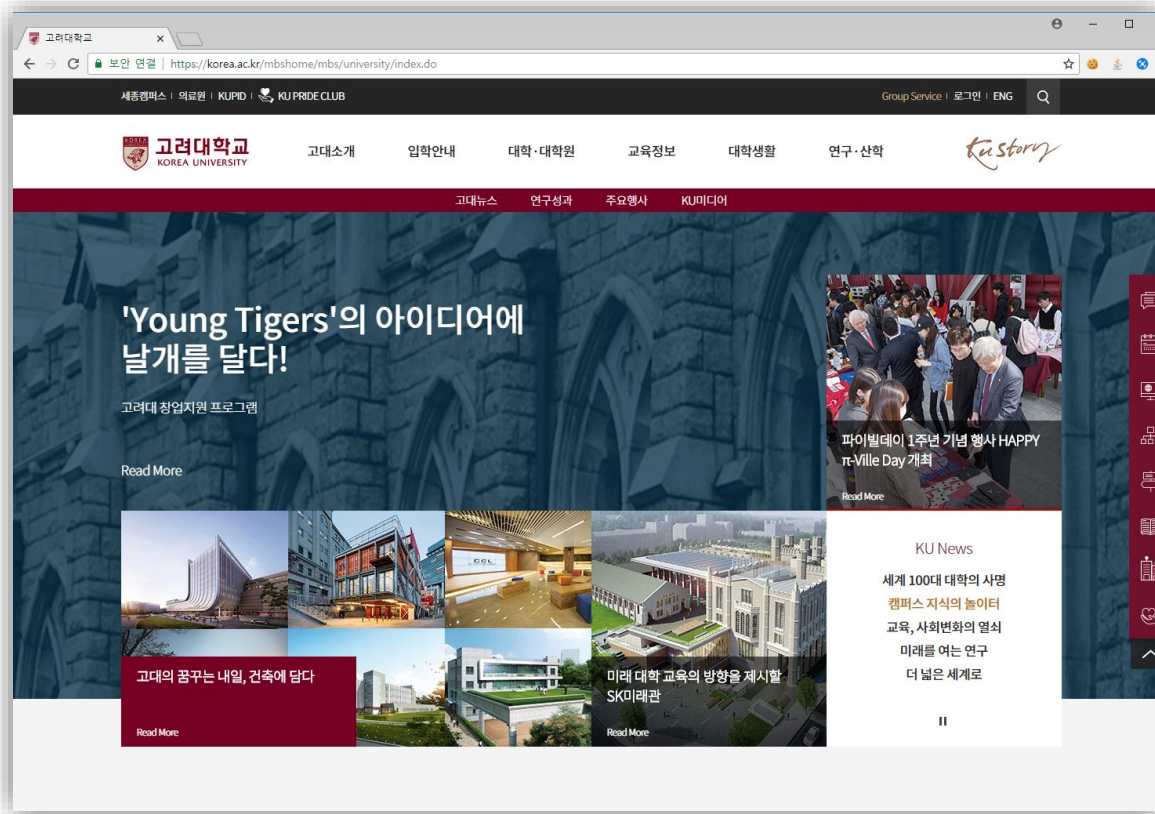


# 서버란?

- 우리는 PC, 스마트폰에서 Application이라 불리는 프로그램들을 사용합니다. 하나의 단말기 내에서 내장되어 있는 API로만 작동되는 프로그램도 있고, 외부에서 가져오는 프로그램도 많습니다.
- 최근의 많은 Application들은 대부분의 데이터를 Local이 아닌 인터넷을 통해 다른 장소에서 가져와 보여주는 것이 일반적입니다.
  - ✓ Facebook, 네이버 등의 데이터도 결국 웹 브라우저라는 프로그램이 인터넷을 이용해 웹 서버에 웹 문서를 달라고 요청하는 형태입니다.
- 이와 같은 형태에서 우리는 Client와 Server를 구분할 수 있습니다.
  - ✓ 클라이언트(Client): 단말에 데이터를 달라고 요청하는 프로그램
  - ✓ 서버(Server): 다른 곳에서 요청 받은 명령을 처리해 주는 프로그램
- 웹의 세계에선 Chrome, Safari, Edge 등의 웹 브라우저가 Client, 사용자(Client)로부터 요청을 받아서 처리, 집행하는 역할을 하는 소프트웨어 또는 그것을 실행하는 컴퓨터가 Server가 됩니다.



# 웹 서버 동작 예시



사용자가 브라우저에 URL 입력 (https://korea.ac.kr)

DNS 서버에서 해당 서버의 IP를 찾음

Client는 Server에게 웹 페이지 요청 (GET 방식 /)

Server는 만들어 둔 웹 문서를 Client에게 전송

HTML, CSS, Javascript 등을 이용하여  
기술적으로 웹 페이지를 렌더링\*

\* 렌더링(rendering): 요청에 따라 템플릿의 내용을 채워 넣는 일





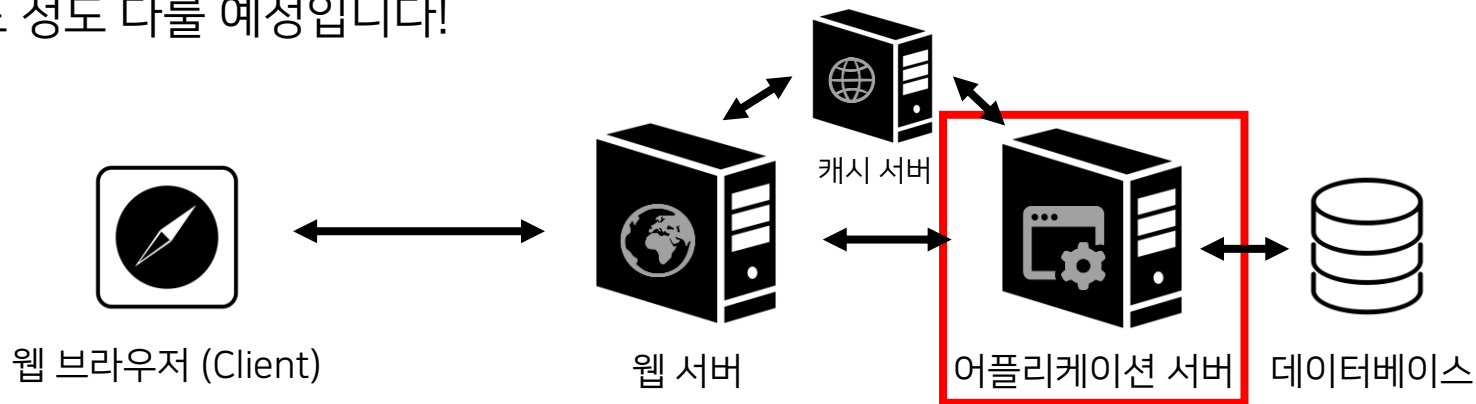
# HTTP & HTTPS

- 앞의 과정을 좀 더 상세히 살펴보려면 http에 대한 이해가 필요합니다.
- 우리는 보통 웹사이트를 http 혹은 https 프로토콜을 사용하여 호출합니다.
  - ✓ 홈페이지 주소의 :// 앞에 붙는 글자들을 생각하면 좀 더 다가오실 겁니다!
- 웹 개발자인 이상 이 부분에 대해 어느 정도는 알고 있어야 합니다. (서버와 통신하는 프로토콜이므로 사실 웹 개발자가 아니어도 개발자라면 이 부분에 대해 어느 정도 지식은 알아야 겠죠?)
- But, 지금 보기에는 다소 시간이 부족하므로 3주차에 가볍게 다뤄보도록 합시다!
  - ✓ 지금은 HTTP의 기본 규칙은 앞에서 배운 Server/Client 모델을 따른다는 정도?만 짚고 넘어갑시다.



# Back-end를 구축하려면?

- 일반적으로 웹에서의 Back-end 구성에는 아래 3가지 요소가 필수적으로 사용됩니다.
  - ✓ 웹 서버: 클라이언트의 요청을 받아서 처리 (Apache, Nginx)
  - ✓ 웹 서버사이드: 클라이언트의 요청을 처리하는 로직 기술 (Node.js, PHP, JSP, ASP)
  - ✓ 데이터베이스 관리 시스템(DBMS): 자료저장 (MySQL, MongoDB)
- 2학기 준회원 스터디에서는 아래 그림에서의 어플리케이션 서버 개발에 관한 스터디를 진행합니다. (일반적으로 Back-end 개발이라 함은 어플리케이션 서버 개발을 의미합니다.)
  - ✓ Node.js 같은 경우는, 웹 서버의 영역까지 어느 정도 커버할 수 있긴 합니다.
  - ✓ DB도 어느 정도 다룰 예정입니다!





# Node.js 설치

- 자연스럽게 Node.js로 넘어가려 했으나 설치하는데 시간이 걸리므로 설치부터 하도록 합니다.
- 설치 URL: <https://nodejs.org>
  - ✓ 오늘 (2018.09.14.)을 기준으로 8.12.0 LTS 버전을 다운로드 받으시면 됩니다.
- 존스에선 LTS 버전을 사용하니 LTS로 다운받아주세요! (실제 서비스에도 보통 LTS를 사용합니다.)
  - ✓ LTS 버전: 최신 버전이 아닌 사용한 지 2년 정도 지나 안정성과 신뢰성이 검증된 버전
  - ✓ 현재 버전: 최신 버전으로 Node.js의 최신 기술들이 탑재되어 있음





- Node.js는 Javascript를 이용하는 서버사이드 플랫폼입니다. → 서버를 구축할 수 있다!
  - ✓ 자세히 말하면, 오픈 소스 JavaScript 엔진인 구글 V8에 비동기 이벤트 처리 라이브러리인 libuv를 결합한 플랫폼인데 언젠간 각자 알아보도록 합시다. → V8 위에서 동작하는 이벤트 처리 I/O 프레임워크
- Javascript는 Front-End에서 주로 활용되었으나! Node.js를 기점으로 Back-End 기술까지 제어할 수 있도록 역할이 혁신적으로 확장되었습니다.
- 2009년 Ryan Dahl이 최초로 고안해냈으며, 현재도 그가 일하던 Joyent라는 회사에서 Node.js의 개발 및 운영을 담당하고 있습니다. (Linux 버전: 2009년 Node.js 최초 출시 / Windows 버전: 2011년 7월 출시)
- 2018.09. 기준 LTS 버전은 8.x 이며, 현재 10.x 가 개발이 진행되고 있는 버전입니다.
  - ✓ 짝수 번째 버전들만 LTS로 사용되며, 10.x 버전이 2018년 10월부터 LTS로 사용됩니다. (중간고사 후부터 사용!)
  - ✓ LTS나 개발 버전 등에 관한 정보는 오른쪽의 URL을 참조해주세요! → <https://github.com/nodejs/Release>



# npm (Node.js Package Manager)

- 2011년 출시된 Node.js 를 위한 패키지 관리 및 배포 시스템입니다.
  - ✓ Node.js 사용시, 모듈 설치 등에 npm install 어쩌구저쩌구 등 해서 많이 사용할겁니다.
  - ✓ 심지어, Node.js 실행시도 npm start 하면 된다는 점!
- Node.js 사용하는 이유 중 하나가 바로 생태계입니다. npm은 이러한 생태계를 활용할 수 있게 해줍니다.
- 패키지 관리 시스템이라 딱히 크게 설명할 바는 없고! 1주차에서 더 다룰 예정이므로 다음주에 또다른 패키지 매니저인 yarn과 함께 더 알아보시다~
- **참고!** Node.js 설치 시 npm이 자연스럽게 함께 설치되어서 이미 여러분 컴퓨터에서 사용할 수 있습니다~
  - ✓ Console 창에 npm -v 치면 확인하실 수 있을겁니다!



# Development Environment

- 2학기 준스에서 기본적으로 사용할 개발환경은 아래와 같습니다.
  - ✓ 서버사이드 플랫폼: Node.js (Javascript 사용)
  - ✓ IDE: WebStorm
  - ✓ 코드 에디터: Visual Studio Code (보조로 Sublime Text)
  - ✓ DB 관리툴: Windows면 HeidiSQL / Mac이면 DBeaver or Sequel Pro → 본인에게 편한 거 아무거나 쓰세요!
  - ✓ 버전관리 및 과제제출: Git & GitHub
- 오늘 필요한 것은 Node.js, 코드 에디터와 Git이며 일단 목표는 위의 프로그램을 모두 설치하는 것입니다!
- Node.js는 앞에서 설치했고! WebStorm, Visual Studio Code, Git Bash 를 설치해봅시다.
  - ✓ DB 관리툴은 지금 당장 필요없기도 하고, 본인 편한 거 쓰시면 되니! 나중에 설치하셔도 됩니다~



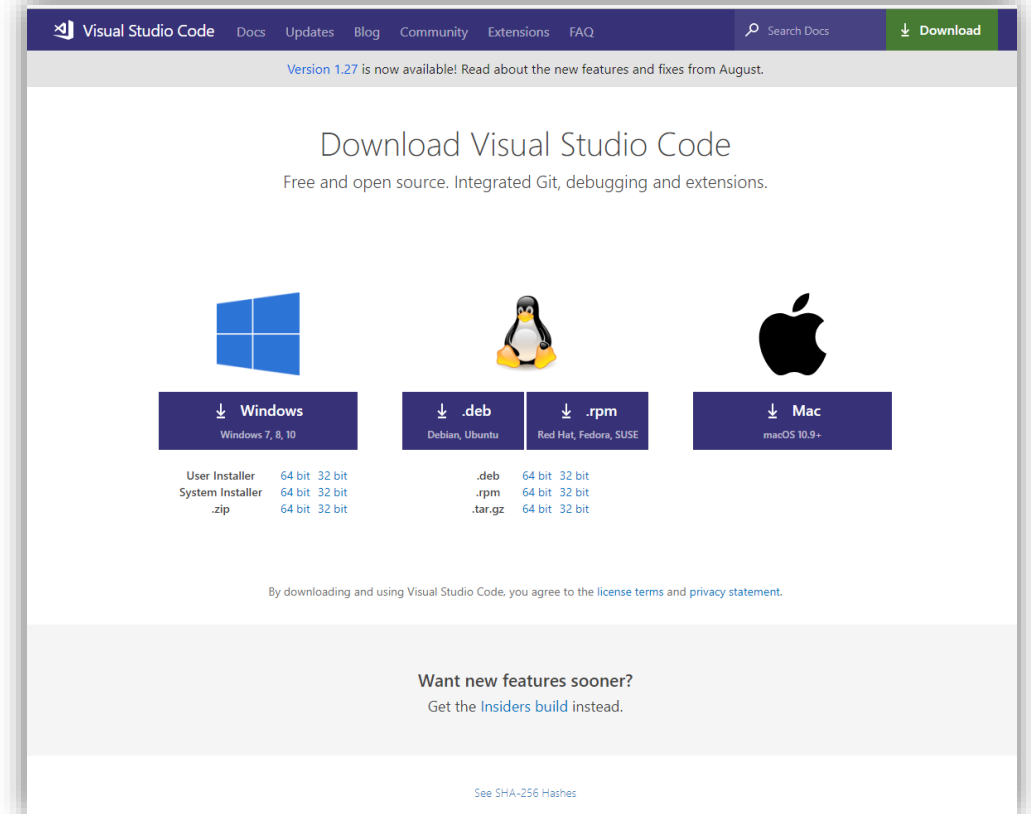
# WebStorm

- JetBrains사에서 제작한 웹 개발계에 요즘 핫한 IDE입니다. Node.js, Node.js + Express.js, Angular, React.js 등 다양한 라이브러리를 지원합니다.
- 원래는 굉장히 비싸지만 우리는 대학생이므로 무료로 이용해보도록 합시다.
  - ✓ <https://www.jetbrains.com/student/>
  - ✓ 위의 사이트로 이동해 Apply Now를 눌러 대학생 계정을 인증 받으시면 됩니다. 그럼 JetBrains사의 다양한 프로그램들을 무료로! 이용하실 수 있습니다. 1년에 한 번씩 추가 인증을 요구합니다~ (by 이메일)
- 아래의 URL에서 설치하시면 됩니다. Next만 누르면 설치되므로 별도 안내사항은 없고! 설치 후에 대학생 인증 받은 jetbrains 계정으로 로그인하시면 정상적으로 활성화됩니다~
  - ✓ URL: <https://www.jetbrains.com/webstorm/download/>
- 2학기 준회원스터디에서 활용할 IDE입니다~ (지금은 설치만 하고! 중간고사 이후에 활용할 예정입니다.)



# Visual Studio Code

- Microsoft에서 제작한 텍스트 에디터, 코드 에디터입니다.
  - ✓ Windows, Mac, Linux 모두 지원하며 Git, 터미널 내장 등 편리한 기능들을 지원합니다. (환경 구축하면 컴파일도 가능!)
- 1학기 때 사용했던 코드 에디터를 Sublime Text를 2학기에선 Visual Studio Code로 대체합니다.
  - ✓ 본인은 Sublime Text 혹은 Atom 등 다른 것을 사용하고 싶으시면 다른 것 쓰셔도 무방합니다~
- 설치하는 무료이므로 아래 URL에서 OS에 맞게 하시면 되고, 별다른 것없이 Next만 누르시면 됩니다!
  - ✓ URL: <https://code.visualstudio.com/>

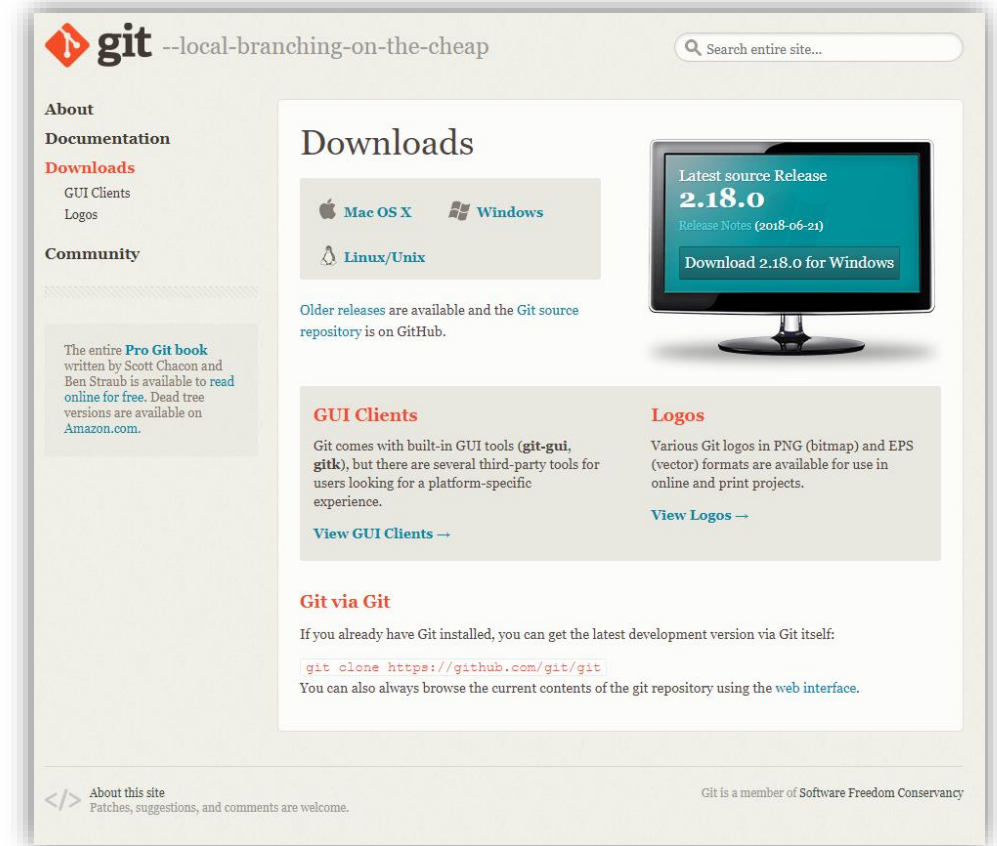






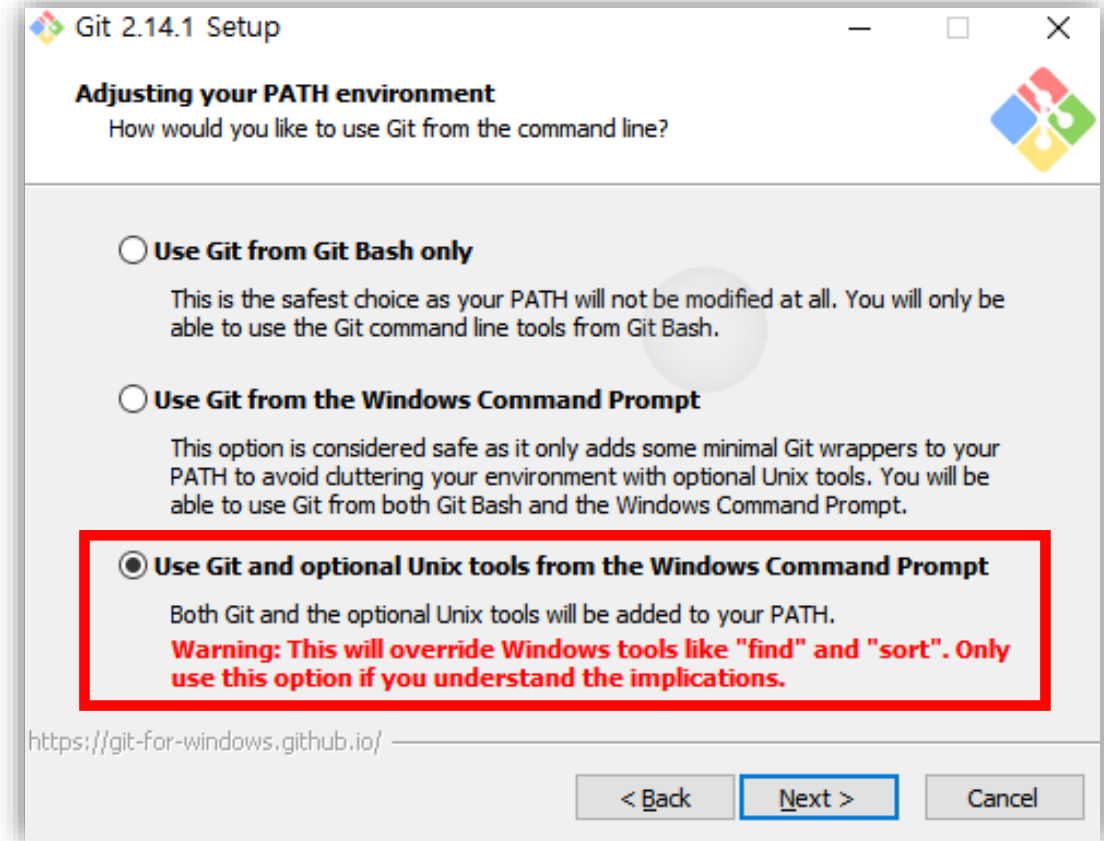
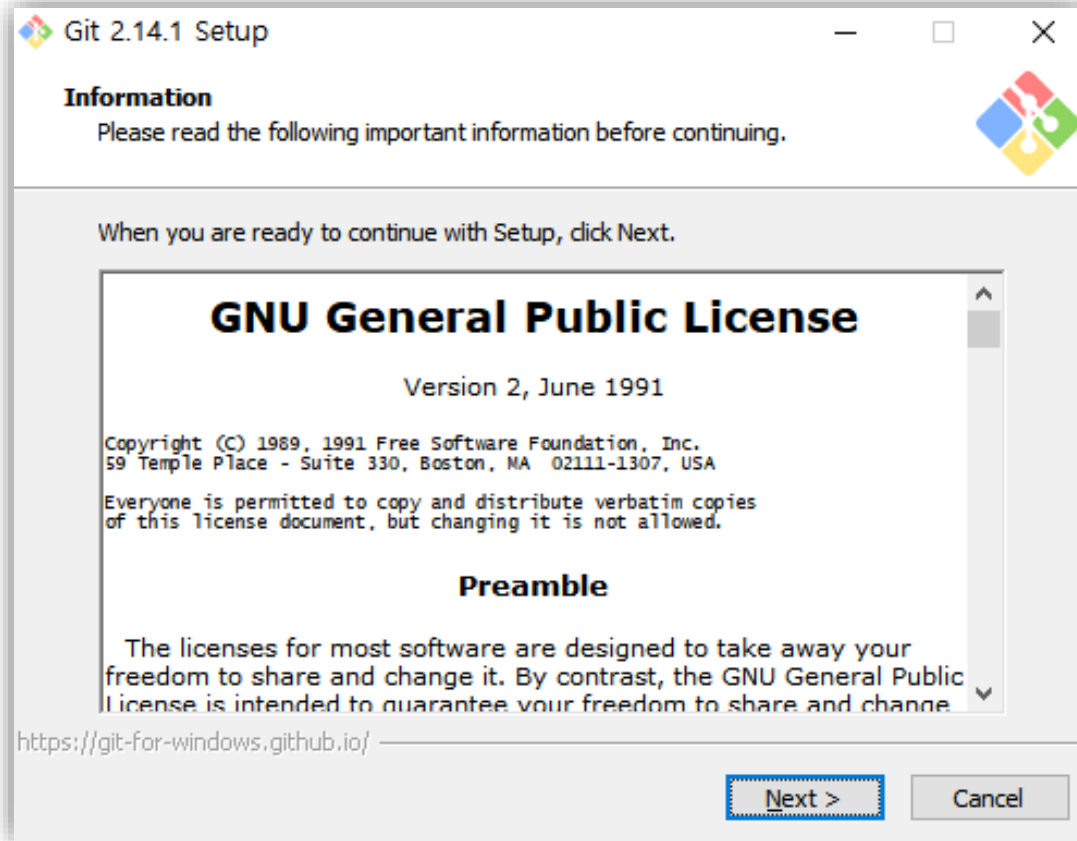
# Git Bash

- Git command를 console 창에서 사용할 수 있게 해줍니다.
- 덤으로, 모든 Windows 창에서 Git Bash 열기가 가능해집니다. cmd처럼 사용 가능합니다~
- Git 사용법은 컴퓨터학과라면 꼭 알아야 하니 나중에 배울 때 잘 배워 두면 쓸모가 많습니다!
- 아래 URL에서 본인 컴퓨터의 OS 버전에 맞게 클릭해주시면 됩니다. (자동 설치 안되면 CPU 따라 클릭해서 설치!)
  - ✓ URL: <https://git-scm.com/downloads>
  - ✓ 참고! Mac은 Xcode 설치 시 자동 설치 됩니다. 최신 버전 업데이트만 해줍니다. + PATH 설정도!





# Git Bash 설치 - Windows



# 실습

---

Node.js 예제 코드 실행시켜 보기

- 오늘의 실습은 간단합니다. (40분? 10분도 안 걸릴 듯) Node.js를 설치했으니 홈페이지의 예시 코드를 실행시켜봅시다.
- Console 환경에서 Node.js를 사용하는 방법과 간단한 웹 서버를 구동하는 법에 대한 실습입니다.
- 크게 실습 1,2로 나뉘어져 있습니다. (사실 안 나뉘어도 될 거 같은데 구색을 맞추려고 나눴습니다 ㅋㅋ)
  - ✓ 실습 1 - Console 환경: Console 환경을 이용하는 Node.js 사용법 접해보기
  - ✓ 실습 2 - 파일 이용: 파일을 이용하는 Node.js 사용법 접해보기
- (사실 따지고 보면 실습 따라가면 과제가 끝날겁니다. 인증샷 남기기~)



# 실습 1 - Console 환경

- Console 환경을 사용해봅시다!
  - ✓ Windows 사용자라면 cmd(명령 프롬프트)나 방금 설치한 Git Bash를 열어줍니다.
  - ✓ Mac 사용자라면 Terminal을 열어줍니다.
- 아래의 command를 입력해 Node.js, npm 그리고 Git의 버전을 알아봅시다.
  - ✓ Node.js 버전: `node -v`
  - ✓ npm 버전: `npm -v`
  - ✓ Git 버전: `git --version`

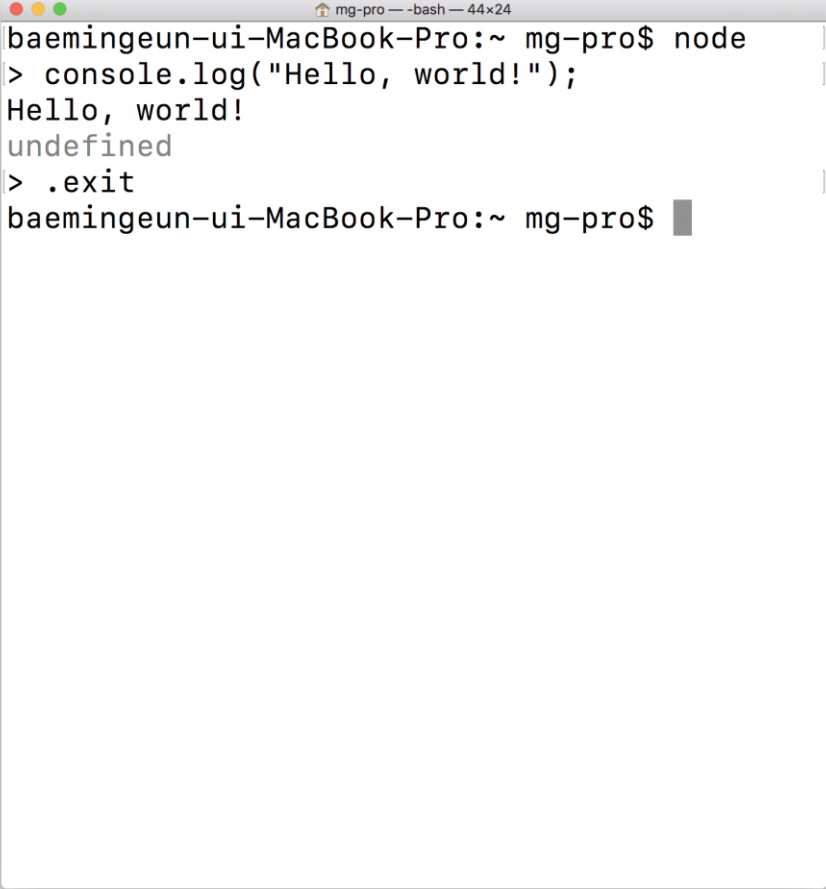
A screenshot of a macOS Terminal window. The title bar shows 'mg-pro' and '-bash' with a size of '52x24'. The terminal text shows the execution of three commands to check the versions of Node.js, npm, and Git.

```
baemingeun-ui-MacBook-Pro:~ mg-pro$ node -v
v8.11.3
baemingeun-ui-MacBook-Pro:~ mg-pro$ npm -v
6.4.1
baemingeun-ui-MacBook-Pro:~ mg-pro$ git --version
git version 2.18.0
baemingeun-ui-MacBook-Pro:~ mg-pro$
```



# 실습 1 - Console 환경

- 이번에는 Console 환경에서 Node.js를 이용해 'Hello, world!' 를 출력해봅시다.
- 먼저, Console 창에 node 를 입력합니다.
- 그 다음, `console.log("Hello, world!");` 를 입력해봅시다.

A screenshot of a macOS terminal window titled 'mg-pro - bash - 44x24'. The terminal shows the following sequence of commands and output:

```
baemingeun-ui-MacBook-Pro:~ mg-pro$ node  
> console.log("Hello, world!");  
Hello, world!  
undefined  
> .exit  
baemingeun-ui-MacBook-Pro:~ mg-pro$
```



# 실습 1 - Console 환경

- 이외에 Node.js의 기능들은 각자 하고 싶은 대로 이용해봅시다! 아래는 예시들입니다.

```
baemingeun-ui-MacBook-Pro:~ mg-pro$ node
> const array = [0,1,2,3,4];
undefined
> array
[ 0, 1, 2, 3, 4 ]
> console.log(array);
[ 0, 1, 2, 3, 4 ]
undefined
> array[0] = -1
-1
> array
[ -1, 1, 2, 3, 4 ]
>
```

```
baemingeun-ui-MacBook-Pro:~ mg-pro$ node
> const object = {
... key: 1,
... 'string_key': 2,
... array: [1,2,3],
... object: {
..... 'a': 'b'
..... }
... }
undefined
> object
{ key: 1, string_key: 2, array: [ 1, 2, 3 ],
  object: { a: 'b' } }
>
```

```
baemingeun-ui-MacBook-Pro:~ mg-pro$ node
> let text = "AAAABBBBCCCC";
undefined
> text.replace("A","X");
'XAAABBBBCCCC'
> text
'AAAABBBBCCCC'
> text.replace(/A/gi,"X");
'XXXABBBBCCCC'
> text
'AAAABBBBCCCC'
> "AAAABBBBCCCC".replace(/A/gi,"X");
'XXXABBBBCCCC'
> .exit
baemingeun-ui-MacBook-Pro:~ mg-pro$
```



## 실습 2 - 파일 이용

- 이번에는 외부 파일을 가져와 Node.js를 이용해 'Hello, world!' 를 출력해봅시다.
- 코드 에디터나 다른 툴들을 활용해 hello.js 를 작성해봅시다. hello.js에는 아래와 같은 코드가 있으면 됩니다.
  - ✓ `console.log("Hello, world!");`

```
JS hello.js x
1 console.log("Hello, world!");
```

- hello.js가 저장되어 있는 디렉토리에서 Git Bash (혹은 Terminal)을 열어 `node hello.js` 를 입력해 봅시다.

A screenshot of a terminal window titled 'kweb -- bash -- 53x24'. The prompt is 'baemingeun-ui-MacBook-Pro:kweb mg-pro\$'. The command 'node hello.js' has been entered and executed, resulting in the output 'Hello, world!'. The prompt is now 'baemingeun-ui-MacBook-Pro:kweb mg-pro\$' with a cursor.

```
baemingeun-ui-MacBook-Pro:kweb mg-pro$ node hello.js
Hello, world!
baemingeun-ui-MacBook-Pro:kweb mg-pro$
```





## 실습 2 - 파일 이용

- 이번엔 Node.js의 예제 코드를 실행시켜봅시다. 실습할 코드는 아래 URL에서 보실 수 있습니다.
  - ✓ URL: <https://nodejs.org/dist/latest-v8.x/docs/api/synopsis.html>
- index.js로 저장한 후, index.js가 있는 디렉토리에서 Git Bash (혹은 Terminal)을 열어 node index.js 를 입력해 실행시켜 봅시다.

```
JS index.js  x
1  const http = require('http');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.setHeader('Content-Type', 'text/plain');
9    res.end('Hello World!\n');
10 });
11
12 server.listen(port, hostname, () => {
13   console.log(`Server running at http://${hostname}:${port}/`);
14 });
```

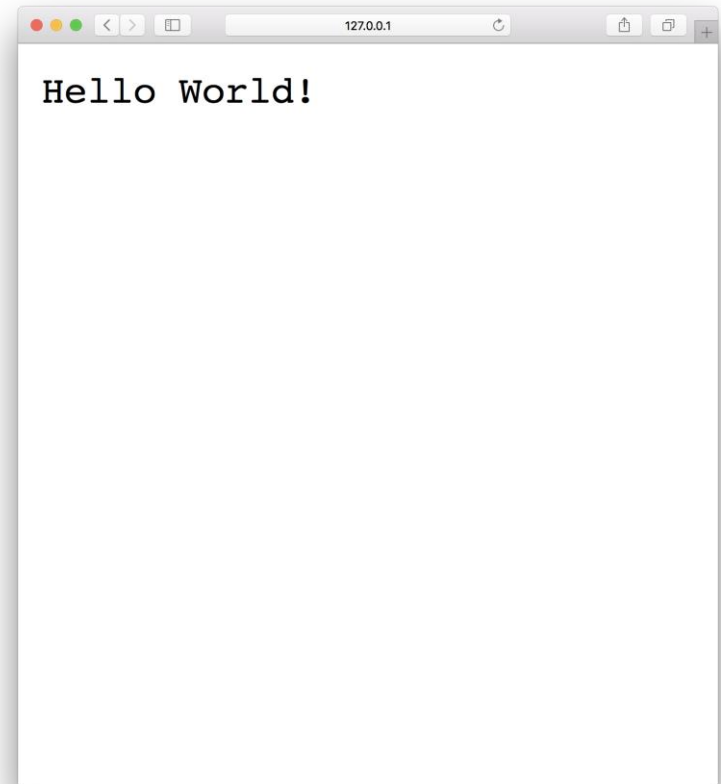
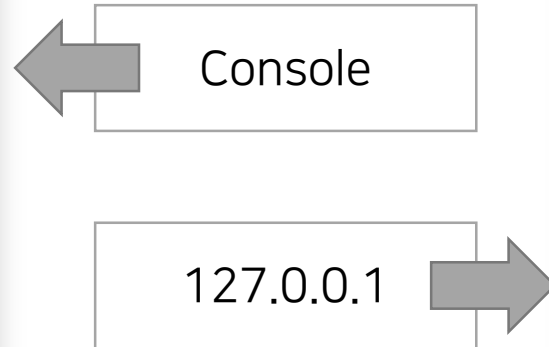


## 실습 2 - 파일 이용

- 앞 슬라이드의 과정을 따르고, Console 창에서 확인할 수 있는 URL을 브라우저로 접속해봅시다.
  - ✓ `http://127.0.0.1:3000` 혹은 `http://localhost:3000` 에서 결과를 확인할 수 있을 것입니다.

A terminal window titled "kweb — node index.js — 53x24". The prompt is "baemingeun-ui-MacBook-Pro:kweb mg-pro\$". The command entered is "node index.js". The output is "Server running at http://127.0.0.1:3000/".

```
baemingeun-ui-MacBook-Pro:kweb mg-pro$ node index.js
Server running at http://127.0.0.1:3000/
```





# 과제 (~ 9/17 17:00)

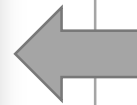
- 과제기한: 2018년 9월 17일 월요일 오후 5시까지
- 이번주 과제는 Node.js 를 사용해 웹 서버를 실행시켜 본인의 학번과 이름 띄우기입니다.
  - ✓ 보이는 웹페이지에는 본인의 학번과 이름이 보여야 합니다.
  - ✓ 서버가 구동될 때, Console 창에서는 'KWEB 과제 실행중' 이라는 문구가 보여야 합니다.
- 참고로 대부분 인코딩이 깨져 한글이 정상적으로 출력되지 않을 것입니다. 해결해서 제출하세요!
  - ✓ Hint! charset=utf-8
- 제출방법: 올바르게 작동되는 사진을 찍어 준회원 톡방에 인증샷 남겨주세요! (스크린 캡처도 됩니다~)



# 과제 제출 예시

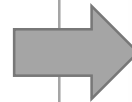
- 아래와 같은 항목들을 캡처해서 준회원 잡담방(카톡)에 올리시면 됩니다!
  - ✓ 스크린샷이 2개인데 꼭 그럴 필요는 없고 한 화면에 다 담은 캡처를 올리셔도 좋습니다~ (사실 이게 좋습니다.)

```
baemingeun-ui-MacBook-Pro:kweb mg-pro$ node index.js
KWEB 과 제 실행 중
```



Console

127.0.0.1





That's all for today!