



WELCOME LIONS!

2019 1ST SESSION

# 목차

1. 웹이란?
2. HTML & CSS
3. 개발을 위한 환경 구축하기

# 웹이란?



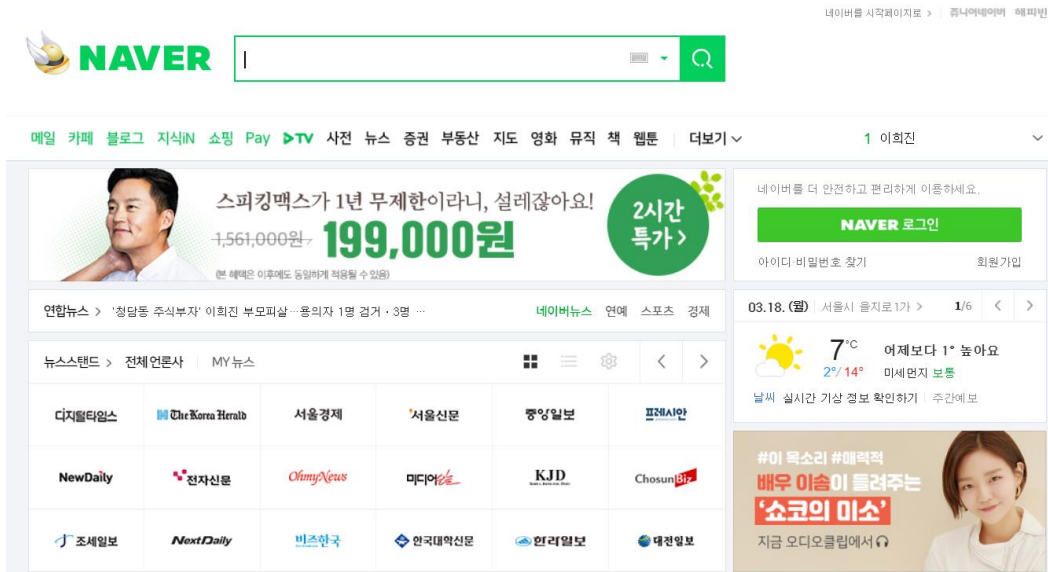
웹: 월드 와이드 웹의 줄임말

특징: 하이퍼링크 개념을 기반으로 하는 인터넷 서비스

⇒ 그럼 하이퍼링크란?

하이퍼텍스트 문서 안에서 모든 형식의 자료를 보여줄 수 있는 고리.

## 우리가 생각하는 웹



## 컴퓨터가 생각하는 웹

```
<html lang="ko">
<head>
<meta charset="utf-8">
<meta name="Referer" content="origin">
<meta http-equiv="Content-Script-Type" content="text/javascript">
<meta http-equiv="Content-Style-Type" content="text/css">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=1100">
<meta name="apple-mobile-web-app-title" content="NAVER" />
<meta name="robots" content="index,nofollow">
<meta name="description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요"/>
<meta property="og:title" content="네이버">
<meta property="og:url" content="https://www.naver.com/">
<meta property="og:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png">
<meta property="og:description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요"/>
<meta name="twitter:card" content="summary">
<meta name="twitter:title" content="">
<meta name="twitter:url" content="https://www.naver.com/">
<meta name="twitter:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png">
<meta name="twitter:description" content="네이버 메인에서 다양한 정보와 유용한 콘텐츠를 만나 보세요"/>

<link rel="shortcut icon" type="image/x-icon" href="/favicon.ico" />

<link rel="stylesheet" type="text/css" href="https://pm.pstatic.net/css/main_v190219.css"/>
<link rel="stylesheet" type="text/css" href="https://pm.pstatic.net/css/webfont_v170623.css"/>
<link rel="stylesheet" type="text/css" href="https://ssl.pstatic.net/sstatic/search/pc/css/api_atomp_181122.css"/>

<script type="text/javascript" src="https://pm.pstatic.net/js/c/nlog_v181107.js"></script>

<script type="text/javascript" src="https://ssl.pstatic.net/tveta/libs/assets/js/common/min/probe.min.js"></script>
<script type="text/javascript">
var nsc = "navertop.v3";
document.domain = "naver.com";
var jindoAll = "";
var iframeLazyLoad = false;
if (!!window.console) {window.console={};window.console["log"]=function({})}
var isLogin = false;
function refreshLcs(etc) {etc = etc ? etc : {};if(document.cookie.indexOf("nrefresh=1") != -1) {etc["mrf"]="1";} else {etc["pan"]="sow";}return etc;}

</script>
<title>NAVER</title>
</head>
```

# 웹이란?

HTML(HyperText Markup Language)이란

=> 웹을 이루는 가장 기초적인 구성요소.

⇒ 웹 페이지의 내용을 서술하고 정의하는 데 사용함.

HyperText

+

Markup Language

연관된 데이터를 링크로 연결하여  
사용자가 필요한 정보를 찾을 수  
있도록 도와주는 정보탐색 구조

태그 등을 이용하여  
문서나 데이터의 구조를  
명기하는 언어의 한 가지

# 웹이란?



CSS(Cascading Style Sheets)란

⇒ 웹페이지의 모양과 표현을 서술하는 데 사용함.

구성: 속성

+

선택자

HTML콘텐츠가 표시되는 방법을  
바꾸기 위한  
값을 가짐

바꾼 속성값을  
적용할 요소를 선택함.



**HTML**



**JS**



**CSS**



# 웹이란?



개발자 도구로 멋사 홈페이지 이미지를 다운받아보자!

1. 모든 요소는 사각형이다.
2. 요소들의 중심으로 들어가면 내가 보고 있는 화면과 똑같은 글이 있다.
3. 개발자도구로 수정이 가능하지만 내 컴퓨터에서만 바뀐다.



# HTML

## HTML5의 구성

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>likelion first class </title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first HTML example</p>
  </body>
</html>>
```

<!DOCTYPE html> : html5양식의 문서이다  
<html>~</html> : 문서의 시작과 끝을 나타낸다.  
<head>~</head> : 웹 페이지에 표시되지 않고,  
문서의 메타데이터를 넣는다.  
    <title> : 브라우저의 제목을 표시한다.  
    <meta> : 메타데이터를 넣는다.  
<body>~</body> : 웹 페이지에 표시하고 싶은  
내용을 넣는다.

## 기본 문법

<div id = "sample"> LIKELION </div>

시작태그      속성 + 속성값      내용      종료태그

## 자주 사용하는 태그

<div>	Division, 경계를 나누는 역할, 태그 자체에는 의미가 없음.
<span>	텍스트 영역 지정
<p>	Paragraph. 단락의 역할. 줄 간격을 띄움
<h1~h6>	Head. 제목 태그. 숫자가 커질수록 글자의 크기가 줄어듬.
<img>	image. href의 속성값으로 이미지가 삽입될 경로를 적어주어야 함.

## 자주 사용하는 태그

<code>&lt;ol&gt;</code>	Ordered list. 순서가 있는 목록.
<code>&lt;ul&gt;</code>	Unordered list. 순서가 없는 목록.
<code>&lt;li&gt;</code>	List. 목록을 나타내는 태그.
<code>&lt;table&gt;, &lt;th&gt;, &lt;tr&gt;, &lt;td&gt;</code>	표를 그리는 태그.표의 제목, 행, 열을 만든다.

## 자주 사용하는 태그

<code>&lt;form&gt;</code>	받은 정보를 전송하는 태그. action 과 name을 통해서 서버로 전송.
<code>&lt;input&gt;</code>	사용자에게 입력값을 받을 때 사용. type의 속성값이 데이터의 이름이 됨.
<code>&lt;textarea&gt;</code>	여러줄의 text를 입력할 때 사용

## CSS의 적용법

1. 인라인: 태그 안에 style 속성을 이용하여 적용한다.
2. 내부 적용: <head>태그 안에 <style>태그를 사용하여 적용한다.
3. 외부 파일로 적용: 하나의 파일에 만들어 그 참조 내용을 <head> 태그 안에 적용해 사용한다.

-> 전체 웹사이트의 모양을 변경할 수 있기 때문에 유지 보수에 용이하기 때문!

# CSS

## CSS

Likelion.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>LIKELION</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

선택자 속성

값

style.css

```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}

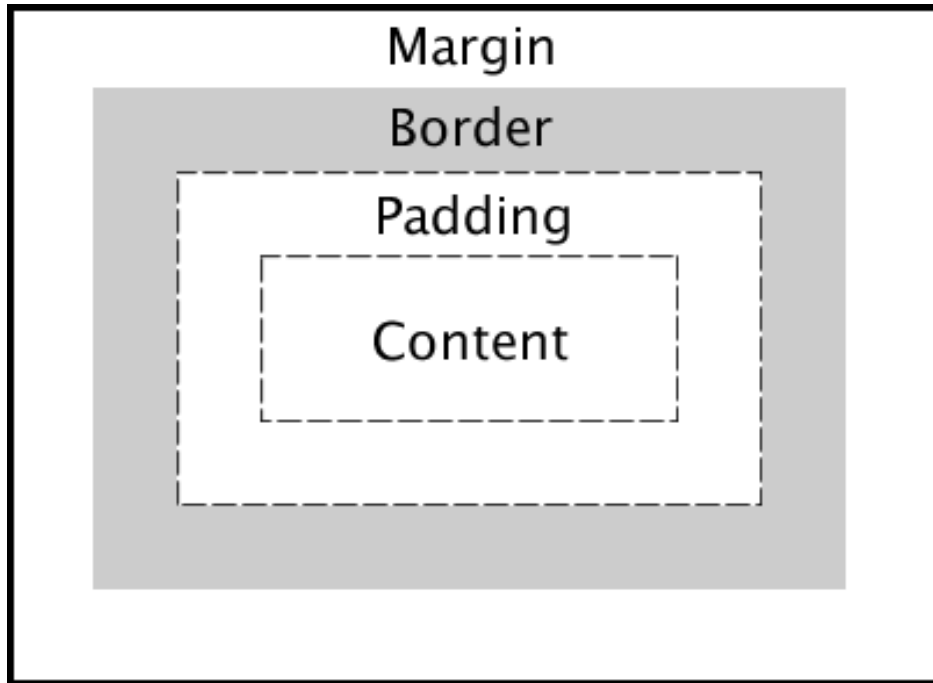
p {
  color: red;
}
```



## 선택자의 종류

태그 선택자	페이지 상에 선택자로 지정한 HTML요소를 모든 요소에 적용하는 경우	p{ }
전체 선택자	스타일을 적용하기 위한 페이지에서 모든 HTML요소를 선택	* { }
클래스 선택자	한 페이지에서 여러 번 사용 가능. 다른 클래스 속성과도 중첩 가능	<div id = "hello"> <p class = "lion">안녕</p> </div>
ID선택자	로고, 상단 메뉴 등 한 페이지에 한 번만 사용되는 곳에 속성을 부여할 때	----- #hello { background-color: gray; } .lion { color: yellow; }

## CSS의 박스 모델



Margin: border 외부의 영역. 여백을 만들 수 있음.  
Border: content가 들어있는 상자를 감싸는 테두리/ 경계  
Padding: 사이의 투명한 영역

Content: 텍스트 및 이미지가 나타나는 상자의 내용

즉! Box의 크기를 지정할 때

Box의 총 가로길이는  $\text{width} + \text{padding} + \text{border}$   
Box의 총 세로길이는  $\text{height} + \text{padding} + \text{border}$

임을 고려하자~

## 박스의 display 속성

1. `display: block;` => 가로 길이가 기본적으로 100%이며 줄바꿈되어 보인다.  
크기 속성을 지정할 수 있다. Ex) `<div>`, `<p>`
2. `display: inline;` => 줄바꿈이 되지 않고 크기 속성을 지정할 수 없다. Ex) `<span>`
3. `display: inline-block;` => block과 inline의 중간 형태.  
줄바꿈이 되지 않지만 크기를 지정할 수 있다.
4. `display: none;` => 태그가 보이지 않고 영역도 차지하지 않음.  
Visibility 를 hidden으로 설정한 것과는 다름!!

## 박스의 position

요소에 사용되는 위치 지정방법

1. position: static; => 페이지의 정상적인 흐름대로 배치
2. position: relative; => 위, 아래, 왼쪽, 오른쪽의 속성을 설정하면 그에 맞게 상대적으로 배치
3. position: fixed; => 위치가 있는 요소를 고정시켜 스크롤 해도 항상 고정됨
4. position: absolute => 좌표를 통해 위치를 지정해줌.



**django**

# 개발을 위한 환경 구축하기

1. Python 3.6 이상의 버전을 설치 합니다.
2. Git을 설치합니다.
3. 코드를 작성할 툴인 VSCode를 설치합니다.

Warning!

<<윈도우 기반의 환경 구축 매뉴얼입니다>>

# 개발을 위한 환경 구축하기

## 1. Python 설치하기

파이썬 공홈 -> [다운로드](#) -> 최신 버전 설치 (3으로 시작하는 파일)

⇒ 파이썬 2와 3의 차이 : 파이썬 3에서는 모든 변수가 객체로 처리됨  
정수 / 정수의 결과가 실수 형으로 처리됨  
Print문의 괄호 필수!

등등...



# 개발을 위한 환경 구축하기

## 2. VSCode 설치하기

[Visual studio code](#) [공홈](#) -> 다운로드

지금 우리가 사용하는 툴들을 가리켜 텍스트 에디터라고 함!

Atom, Sublime text 등등 종류는 많지만 우리는 vscode를 쓴다!

혹시 설치되어 있는 텍스트 에디터가 있다면 튜터에게 이야기 해 주세요!

# 개발을 위한 환경 구축하기

## 3. Git 설치하기

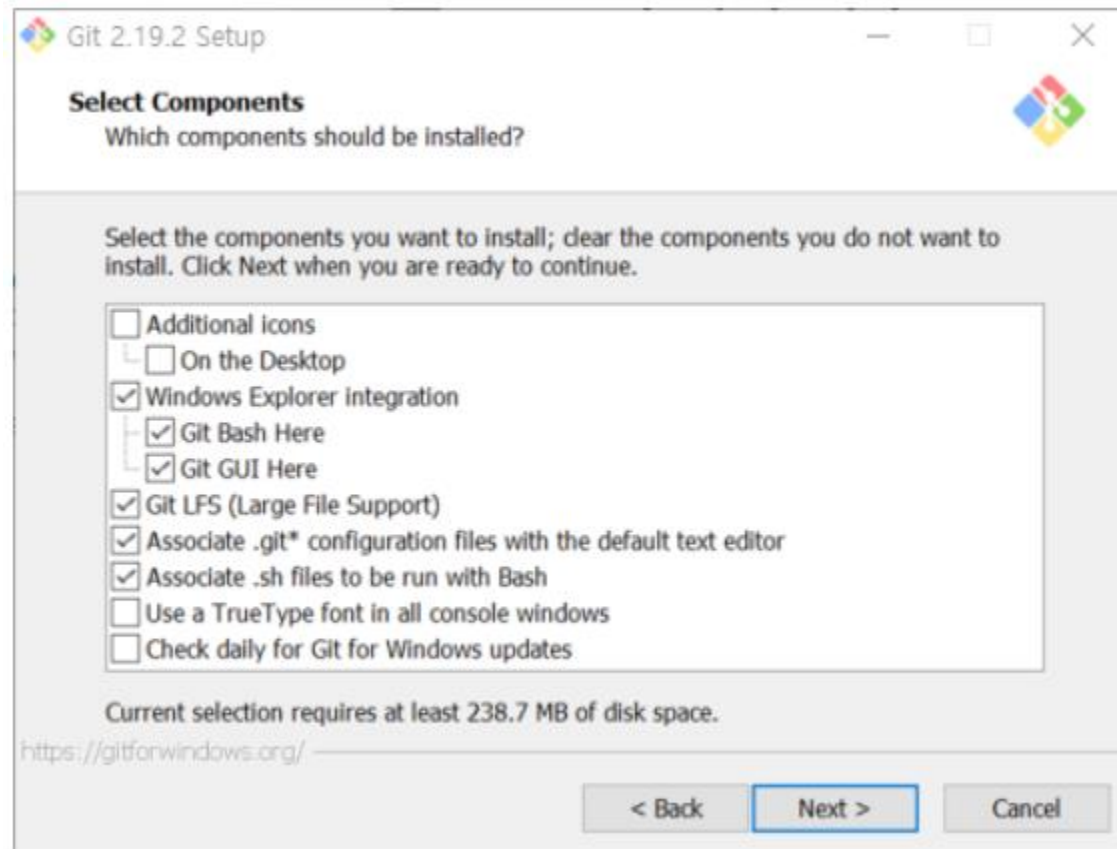
[Git 공홈](#) -> 다운로드

Git : 내가 작성한 소스코드를 편하게 관리할 수 있게 도와주는 버전 관리 툴.

여기서부터는 튜터와 함께 넘어갑니다.

# 개발을 위한 환경 구축하기

## 3. Git 설치하기



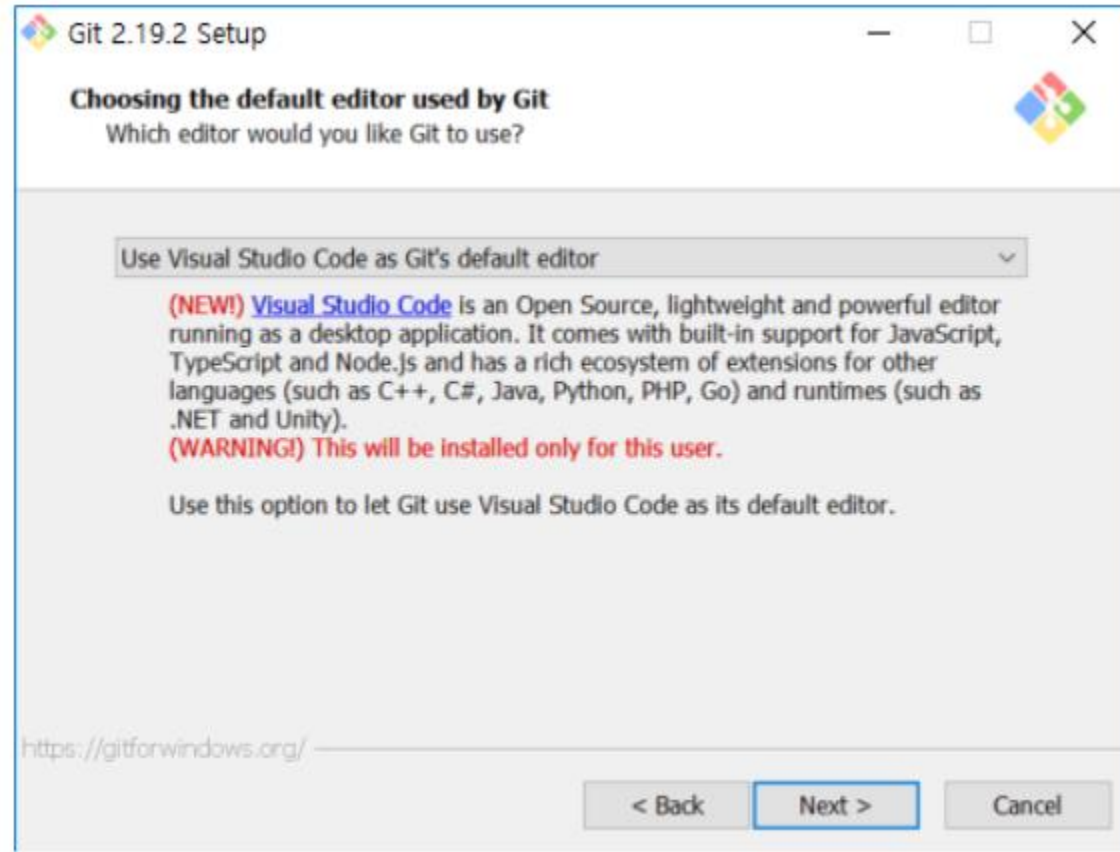
**NEXT**

# 개발을 위한 환경 구축하기

## 3. Git 설치하기

미리 설치해둔  
에디터가 있는  
사람은

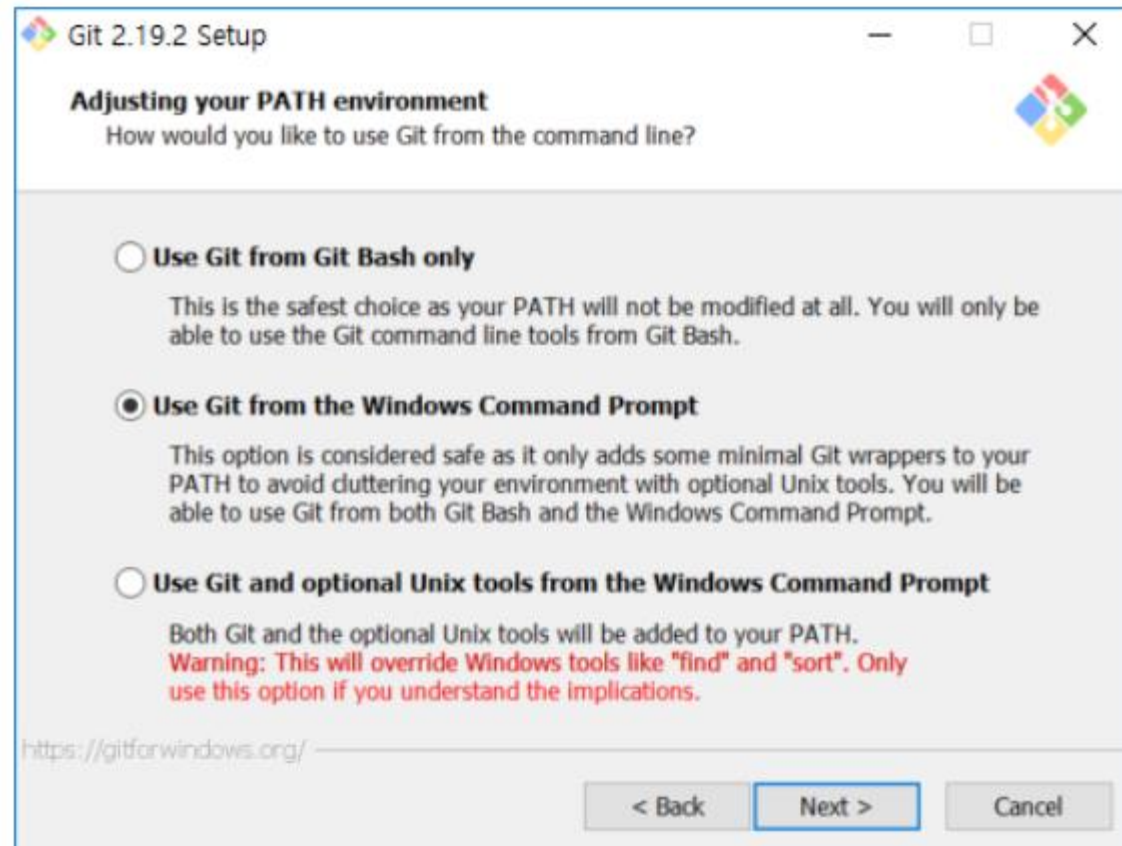
손!!!!



**NEXT**

# 개발을 위한 환경 구축하기

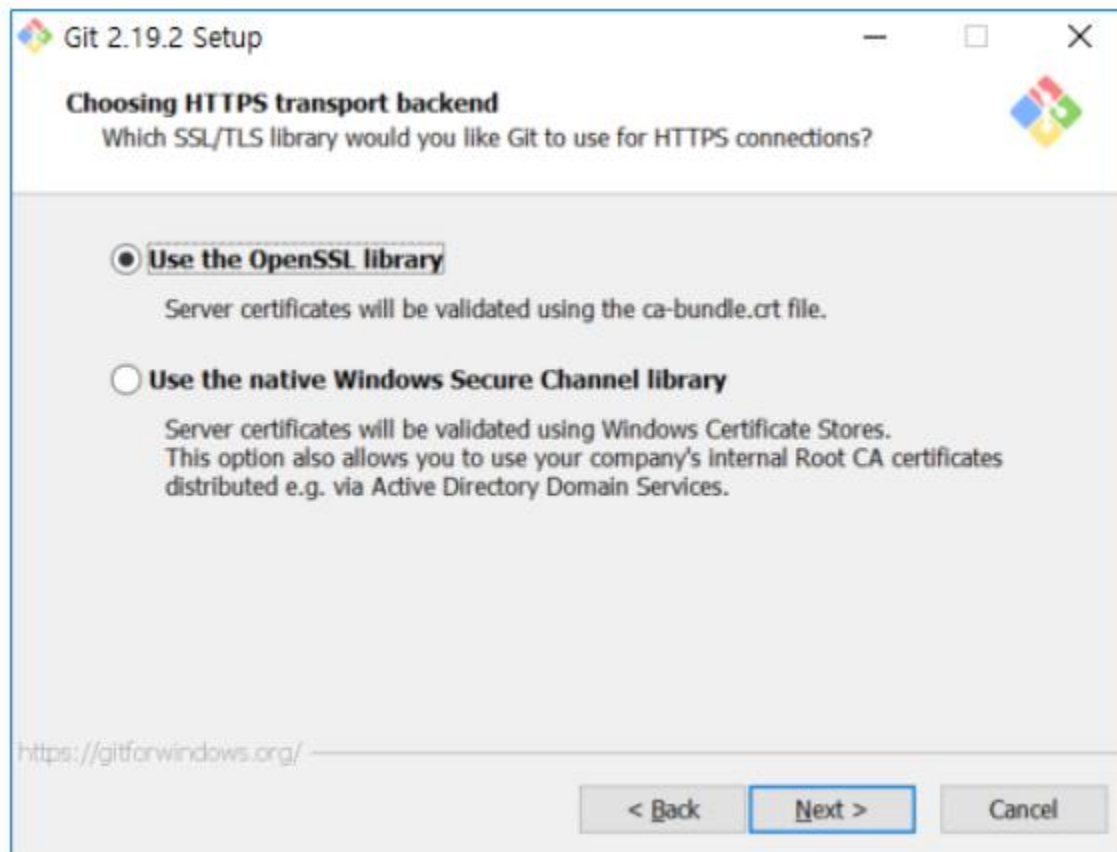
## 3. Git 설치하기



**NEXT**

# 개발을 위한 환경 구축하기

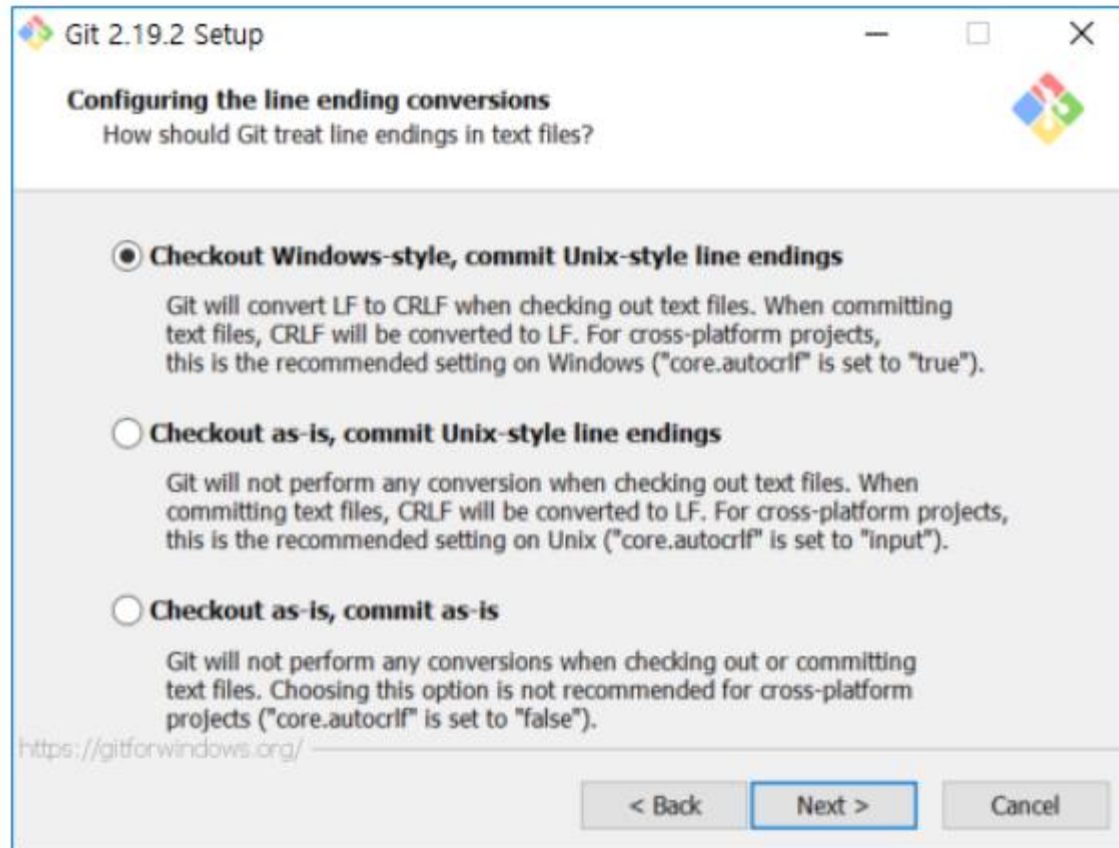
## 3. Git 설치하기



**NEXT**

# 개발을 위한 환경 구축하기

## 3. Git 설치하기

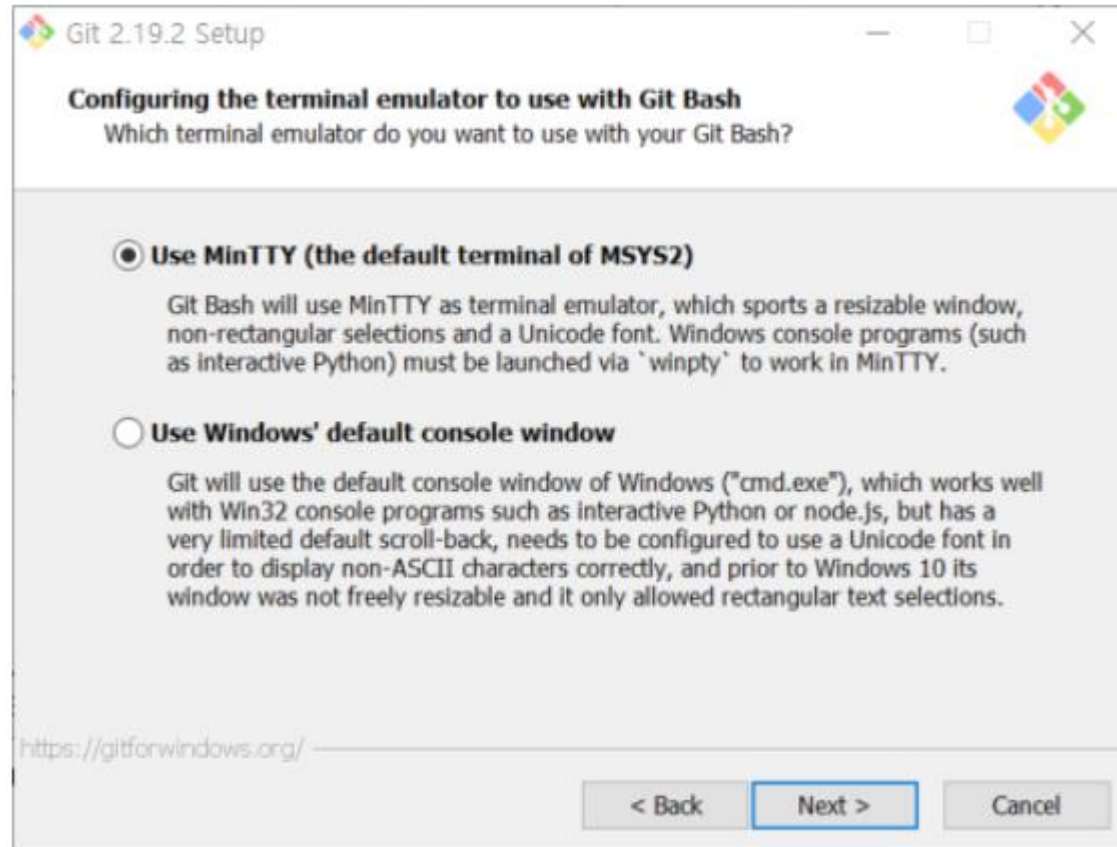


**NEXT**



# 개발을 위한 환경 구축하기

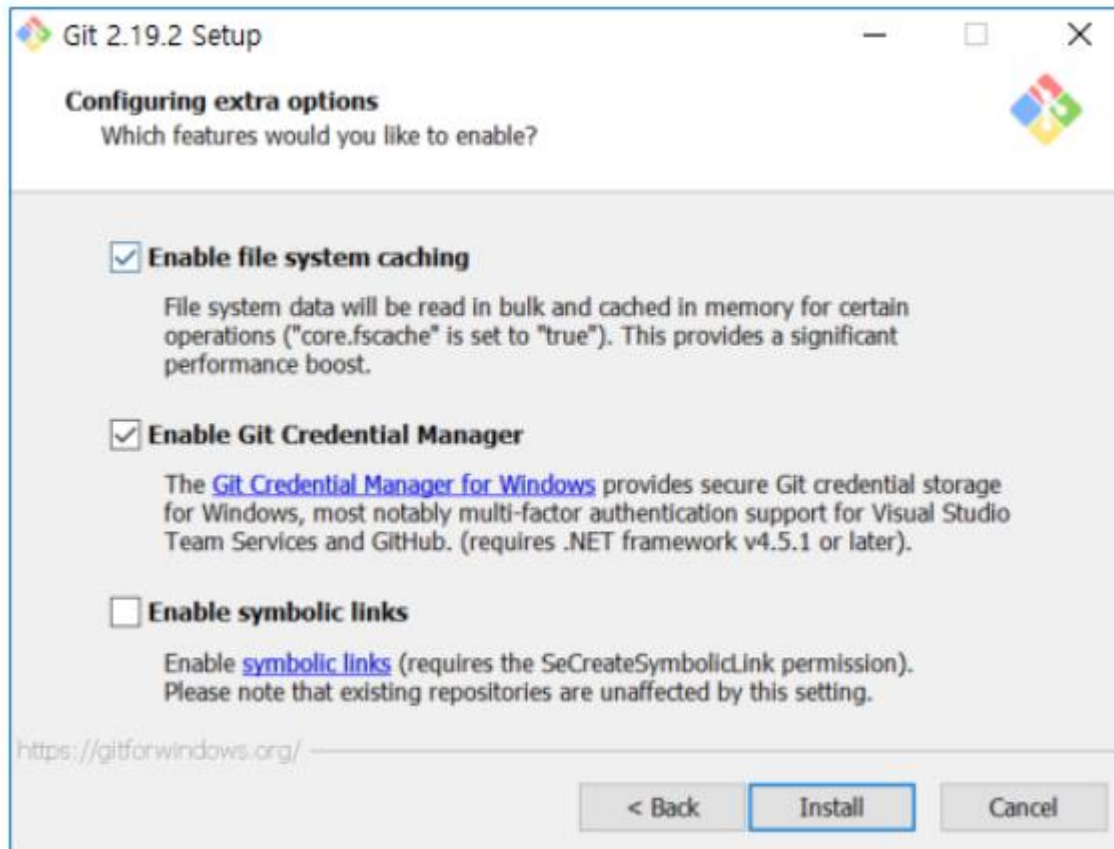
## 3. Git 설치하기




**NEXT**

# 개발을 위한 환경 구축하기

## 3. Git 설치하기



## 개발을 위한 환경 구축하기

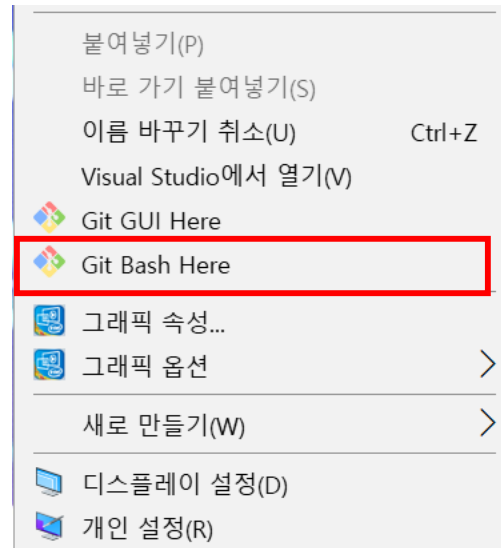


모든 설치가 완료되면 이제부터 환경 세팅을 하자!

# 개발을 위한 환경 구축하기

터미널을 사용해봅시다.

바탕화면에 우클릭 ->



```
MINGW64:/c/Users/노하은/Desktop/likelion

노하은@bluepicture MINGW64 ~/Desktop
$ mkdir likelion

노하은@bluepicture MINGW64 ~/Desktop
$ cd likelion

노하은@bluepicture MINGW64 ~/Desktop/likelion
$ |
```

mkdir: make directory / cd: change directory

# 개발을 위한 환경 구축하기

```
노 하 은 @bluepicture MINGW64 ~/Desktop/like1ion  
$ code .|
```

Ctrl + shift + ` 로 터미널 창 합치기

잘 되는 지 보기 위해서 파이썬 버전 확인해봅시다!

```
$ python --version
```

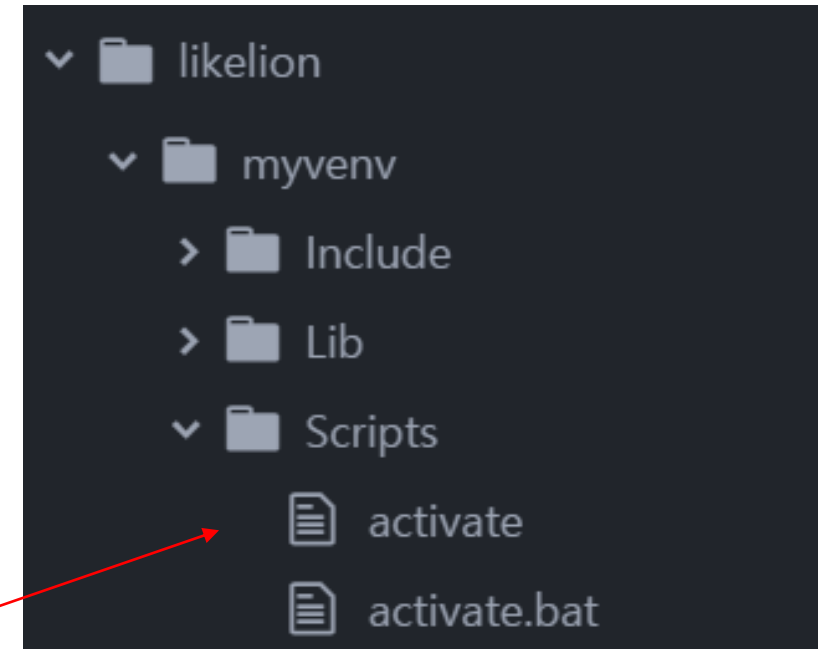
# 개발을 위한 환경 구축하기

## 가상환경 생성 및 실행하기

```
노 하 은 @bluepicture MINGW64 ~/Desktop/likelion
$ python -m venv myenv

노 하 은 @bluepicture MINGW64 ~/Desktop/likelion
$ ls
myenv/

노 하 은 @bluepicture MINGW64 ~/Desktop/likelion
$ source myenv/Scripts/Activate
[myenv]
노 하 은 @bluepicture MINGW64 ~/Desktop/likelion
$ |
```



가상환경 켜진 것 확인하기!

ls : list

# 개발을 위한 환경 구축하기

## 가상환경 생성 및 실행하기

```
노하은@bluepicture MTN64 ~/Desktop/likelion
$ pip install django
Collecting django
  Using cached https://files.pythonhosted.org/packages/c7/87/fbd666c4f87591ae25b7bb374298e8629816e87193c4099d3608ef11fab9/Django-2.1.7-py3-none-any.whl
Collecting pytz (from django)
  Using cached https://files.pythonhosted.org/packages/61/28/1d3920e4d1d50b19bc5d24398a7cd85cc7b9a75a490570d5a30c57622d34/pytz-2018.9-py2.py3-none-any.whl
Installing collected packages: pytz, django
Successfully installed django-2.1.7 pytz-2018.9
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(myvenv)
```



# 개발을 위한 환경 구축하기

## 가상환경 생성 및 실행하기

```
bluepicture@bluepicture-MINGW64 /Desktop/like1ion
$ python -m pip install --upgrade pip
Collecting pip
  Using cached https://files.pythonhosted.org/packages/d8/f3/413bab4ff08e1fc4828dfc59996d721917df8e8583ea85385d51125dceff/pip-19.0.3-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 18.1
    Uninstalling pip-18.1:
      Successfully uninstalled pip-18.1
Successfully installed pip-19.0.3
(myvenv)
```

끄을!

