Hello,

KDT 웹 개발자 양성 프로젝트

5기!

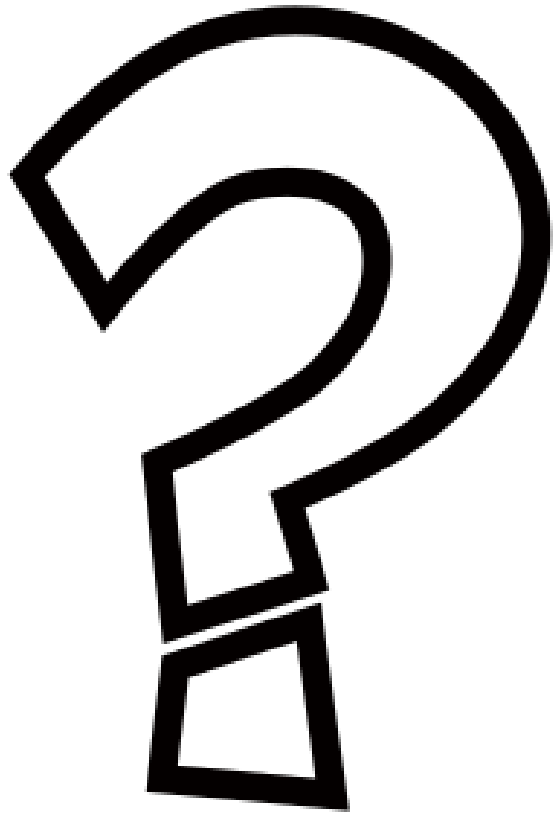
with


이렇게 된 이상
수업을
시작한다!
할
할
오케이!!

Why?

우리는 이런 언어들을 배울 겁니다!

이 언어들은 왜 나누어져 있을까요?




OutdoorNews

이 언어들은 왜 나누어져 있을까요?


PURPOSE

그럼 컴퓨터 언어는?

- 웹 페이지를 만들고 싶다면?
- 앱을 만들고 싶다면?
- 서버를 구현하고 싶다면?
- 해커의 공격을 방어하고 싶다면?
- 인공 지능을 개발하고 싶다면?
- 소형 시계에 들어가는 프로그램을 만들어야 한다면?

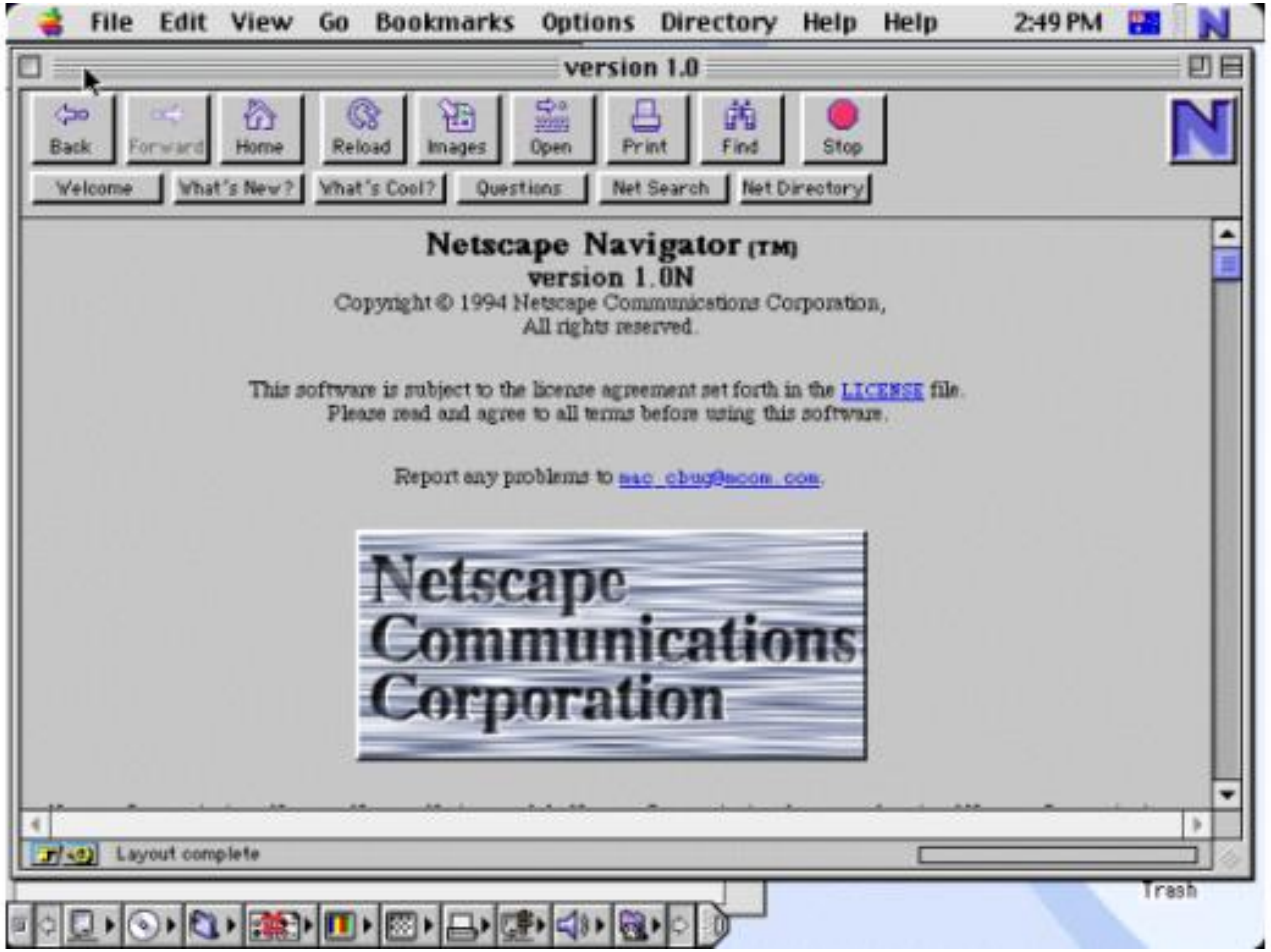
What?

HTML

- 왜? 생겼을까요?

역사적으로 인류는!

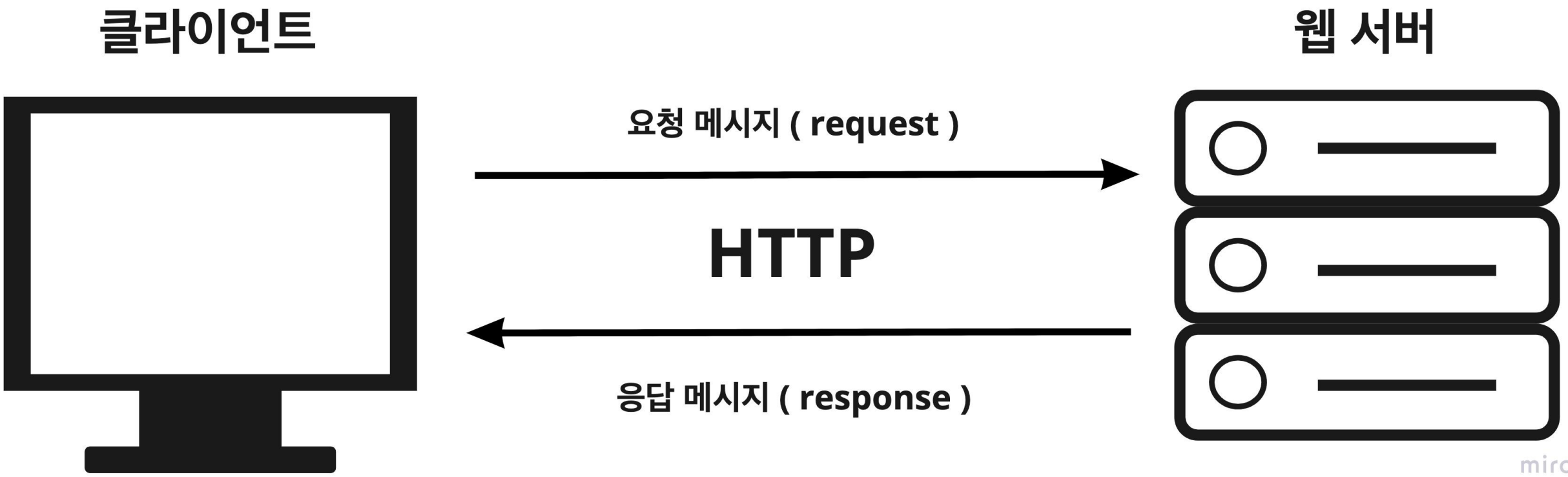
인터넷의 탄생과, HTML



탄생! 인터넷(International Network)

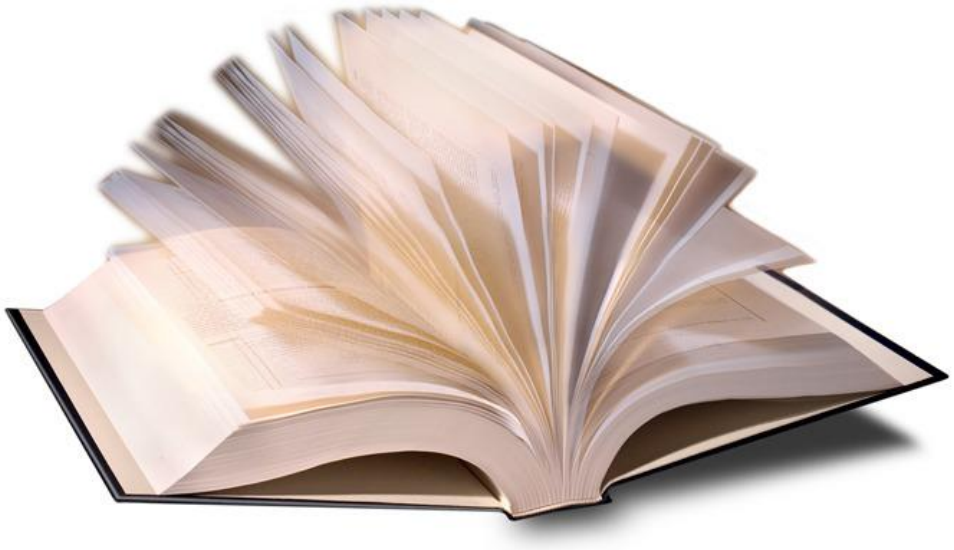
- 1973, '빈튼 서프'와 '밥 간'이 인터네셔널 네트워크 발명
- 1980, 미국 국방성이 연구용으로 'ARPA NET'을 구축
- 1983, 미국 국방성이 군사용 네트워크는 'MIL NET'으로 분리하여 'ARPA NET'은 민간용으로 사용 시작
- 1989, '팀 버너스리'가 월드 와이드 웹의 하이퍼텍스트시스템, URL, HTTP, HTML을 설계 및 개발
- 1993, HTML 1.0 발표
- https://www.w3.org/Protocols/HTTP/AsImplemented.html

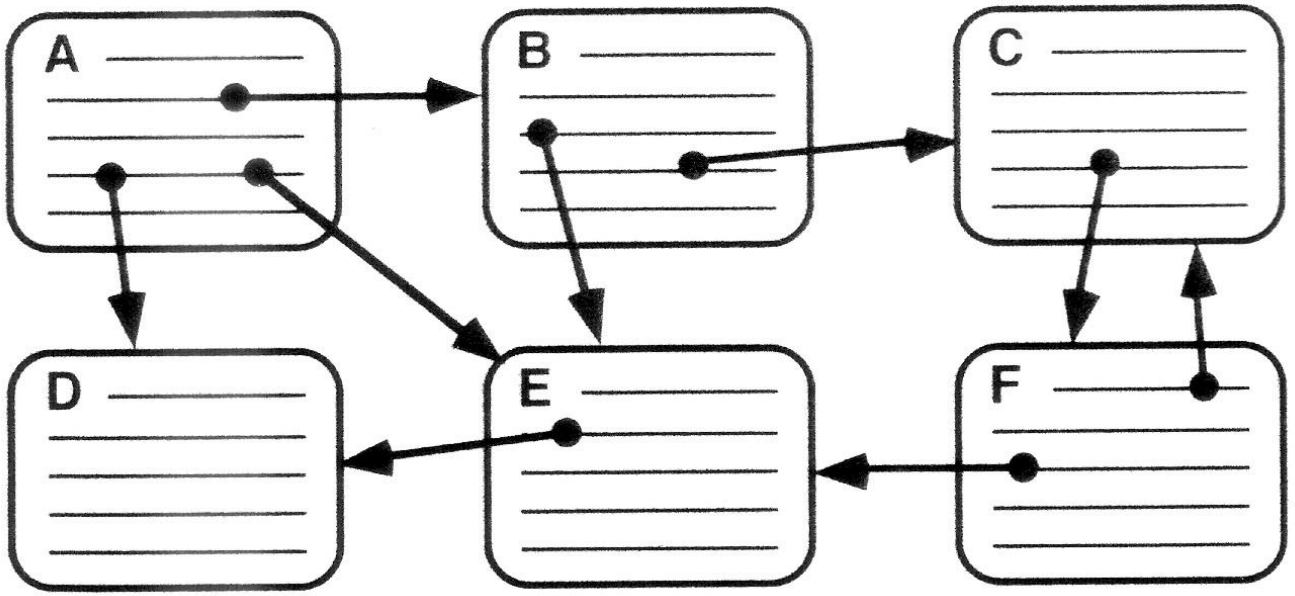
HTTP(Hyper Text Transfer Protocol)



HyperText Markup Language

- HyperText?





HyperText Markup Language

- Markup

HTML

- 공식 문서 확인
- https://developer.mozilla.org/ko/docs/Learn/HTML/Introduction_to_HTML/Getting_started

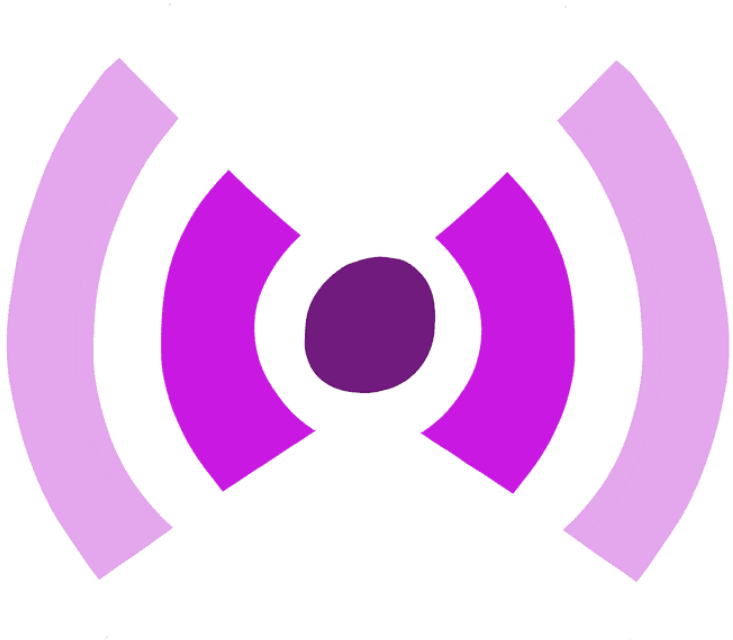

드디어 실습이구나

VS Code 설치

Visual Studio Code

https://code.visualstudio.com/

Live Server



Live Server v5.7.9

Ritwick Dey | 30,163,373 | ★★★★★(424)

Launch a development local Server with live reload feature for static & dynamic pages

사용 안 함 | 제거

이 확장은 전역적으로 사용하도록 설정되었습니다.

세부 정보 기능 기여도 변경 로그 런타임 상태

[Wanna try LIVE SERVER++ (BETA) ? It'll enable live changes without saving file. https://github.com/ritwickdey/vscode-live-server-plus-plus]

Live Server

Live Server loves your multi-root workspace

> **Live Server for server side pages like PHP. Check Here**

> ***[For 'command not found error' #78]***

vscode marketplace v5.7.9 | downloads 56M | rating 4.3/5 (424)

travis branch passing | license MIT

Launch a local development server with live reload feature for static & dynamic pages.

Prettier

Formatting

Formatting?

README.md — prettier-config-example

README.md

```
# Prettier support for other languages

This folder has JavaScript, CSS, HTML, JSON and even this Markdown file all formatted using
Prettier

Note that while formatting [index.js](index.js) Prettier uses 2 spaces per tab, but in the
Markdown code blocks it uses 4 spaces.

```js
const code = true;if (code) {console.log('code is on')}
```

This is because we override options inside the .prettierrc.json file.
```

Formatting

- Code의 스타일을 통일 시켜 줍니다!
- 함수의 소괄호와 중괄호는 띄울 것인지? 세미 콜론은 찍을 것인지? 탭을 누르면 몇 칸을 띄울 것인지? 등등등
- 문법이 아닌 코드의 스타일을 통일 시켜줘서 가독성을 높이고 버그를 예방합니다!
- Prettier 를 사용!

Prettier - Code formatter

v9.10.4

Prettier | 28,403,214 | ★★★★½(375) | 스폰서

Code formatter using prettier

사용 안 함 | 제거

이 확장은 전역적으로 사용하도록 설정되었습니다.

세부 정보 | 기능 기여도 | 변경 로그 | 종속성 | 런타임 상태

Prettier Formatter for Visual Studio Code

Prettier is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maxi
line length into account, wrapping code when necessary.

JavaScript · TypeScript · Flow · JSX · JSON

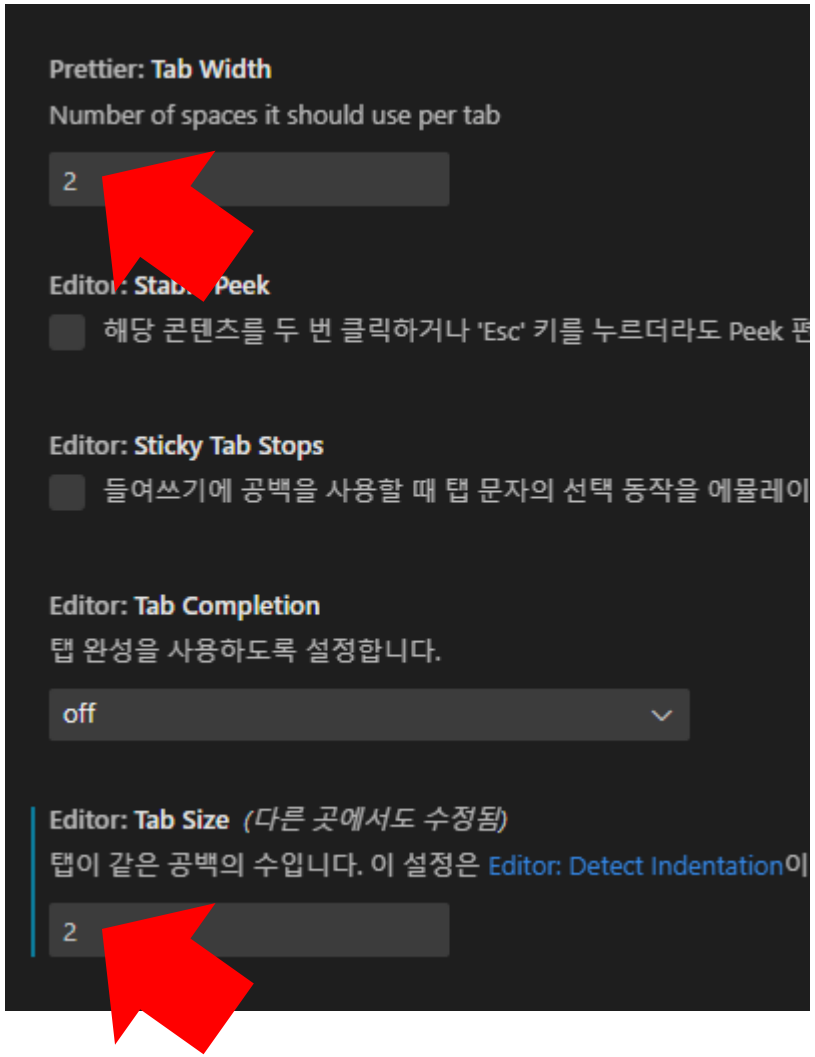
CSS · SCSS · Less

HTML · Vue · Angular HANDLEBARS · Ember · Glimmer

GraphQL · Markdown · YAML

Your favorite language?

Main passing | downloads 160M | installs 28M | code style prettier | follow prettier unparseable json response


Prettier: Tab Width
Number of spaces it should use per tab
2
Editor: Stab Peek
해당 콘텐츠를 두 번 클릭하거나 'Esc' 키를 누르더라도 Peek 편
Editor: Sticky Tab Stops
들여쓰기에 공백을 사용할 때 탭 문자의 선택 동작을 에뮬레이
Editor: Tab Completion
탭 완성을 사용하도록 설정합니다.
off
Editor: Tab Size (다른 곳에서도 수정됨)
탭이 같은 공백의 수입니다. 이 설정은 Editor: Detect Indentation이
2

Do it!

<html>

Hello, World!

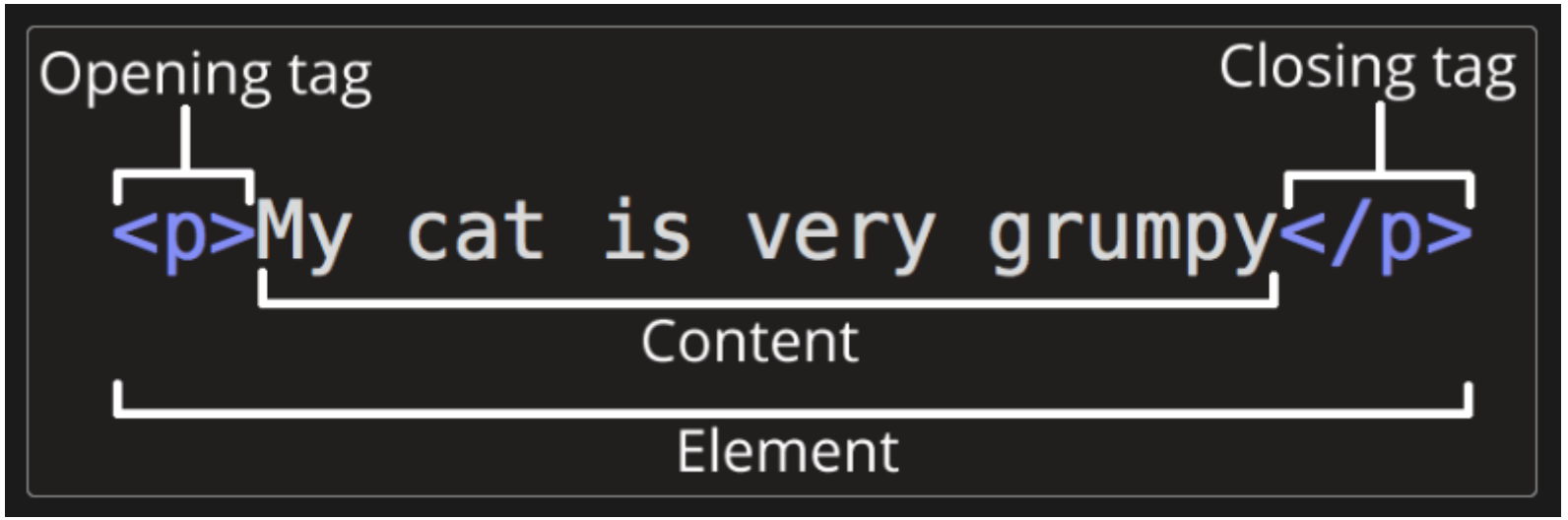
<html>


3
5
JS

HTML
기초 문법

요소

HTML 기본 문법, 요소(Element)



- HTML 요소(Element)는 시작 태그(Opening tag)와 종료 태그(Closing tag) 그리고 태그 사이에 위치한 내용(Content)로 구성

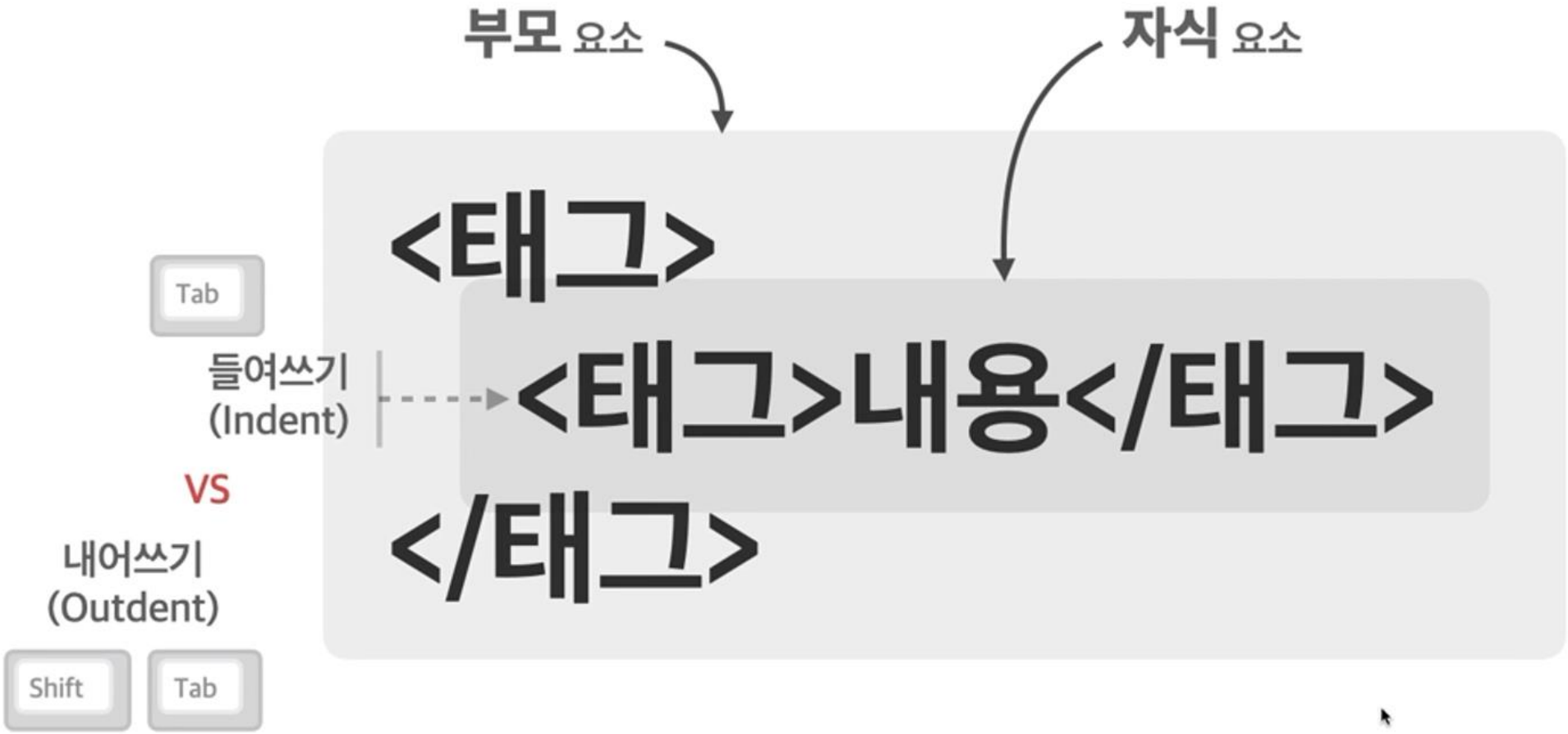
```
<h1>Hello, HTML world!</h1>
```

Hello, HTML world!

중첩

```
<!DOCTYPE>
<html>
  <head>
    <title>페이지 타이틀</title>
  </head>
  <body>
    <h1>여기는 제목입니다.</h1>
    <p>여기는 문장입니다.</p>
  </body>
</html>
```

HTML 기본 문법, 중첩(Nested)



<태그> 자식 요소

<태그>

<태그>내용</태그>

</태그>

</태그>

```
<태그>
    <태그>
        <태그>내용</태그>
    </태그>
</태그>
```

하위(후손) 요소

```
<태그>
    <태그>
        <태그>내용</태그>
    </태그>
</태그>
```

부모 요소

<태그>

<태그>

<태그>내용</태그>

</태그>

</태그>

상위(조상) 요소

```
<h1>
  <span>>Hello, HTML World!</span>
</h1>
```

Hello, HTML world!

빈 요소

HTML 기본 문법, 빈 요소(Empty)

- 내용이 없는 요소(가질 수 없거나, 필요 X)
- 빈 요소는 내용이 없으며, 속성만 소유
-
, <hr>, <img>, <input>, <link>, <meta> 등이 존재

빈(Empty) 태그!

<태그></태그>

종료(닫힌) 태그???

편리함!

HTML 1/2/3/4/5

<태그>

VS

<태그 />

안전함!

XHTML / HTML5

```
<h1>
  <span>Hello,<br />HTML World!</span>
</h1>
```

Hello,
HTML World!

주석

HTML 기본 문법, 주석(Comments)

```
1  <!--주석은 컴퓨터가 아닌 사람을 위한 것이며
2      화면에 표시되지 않는다.-->
3  <p>Hello, KDT World!</p>
```

- 주석(comment)는 개발자에게 코드를 설명하기 위해 사용
- 브라우저는 주석을 화면에 표시하지 않는다.

실습, 간단한 HTML 문서 만들기

- 지금 까지 배운 <h1>, <hr>, 주석을 사용하여 아래 그림과 같은 간단한 HTML 문서 만들기~!

첫번째 실습입니다!

- 주석을 사용하여 자신이 하고싶은 말 전달하기!

태그의 속성

HTML 기본 문법, 속성(Attribute)

Attribute Value

```
<태그 속성="값">내용</태그>
```

기능의 확장

- 속성(Attribute)이란 요소의 성질, 특징을 정의
- 속성은 요소에 추가적 정보(예를 들어 이미지 파일의 경로, 크기 등)를 제공한다.

이미지를 삽입하는 요소(태그)!
어떤 이미지를 삽입해야 하지???
이미지 이름은 뭐지??

이미지의 경로

이미지의 이름
(대체 텍스트 / Alternate Text)

```
<img src="./cat.jpg" alt="고양이" />
```

<input />

사용자가 데이터를 입력하는 요소(태그)!
어떤 데이터 타입을 입력 받을 거지?

hello world

화면에 출력!

데이터의 타입

text
사용자에게 일반 텍스트를 입력 받음!

```
<input type="text" />
```

화면에 출력!

데이터의 타입

checkbox
사용자에게 체크 여부를 입력 받음!

```
<input type="checkbox" />
```

HTML
문서의 구조

HTML 구조 설명

<head>

<body>

HTML 구조 설명

<html>

<head>

<body>

</html>

HTML 구조 설명

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>안녕하세요! HTML5</p>
  </body>
</html>
```

- HTML5 문서는 반드시 <!DOCTYPE html> 으로 시작하여 문서 형식을 HTML5로 지정
- 실제적인 HTML 문서는 2번째 행부터 시작, <html>과 </html> 사이에 작성

HTML 구조 설명

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>안녕하세요! HTML5</p>
  </body>
</html>
```

- <head>와 </head> 사이에는 <body> 태그의 정보를 읽어 들이기 위한 정보가 존재 (제목, 저장 방식, 브라우저의 크기 등등)
- 웹 브라우저에 출력되는 모든 요소는 <body>와 </body> 사이에 위치한다

<head>

<head> 태그

- 메타 데이터(데이터를 위한 데이터)를 포함하기 위한 태그
- <body> 의 내용을 읽기 위한 정보를 표기
- 하나의 웹 페이지에 하나만 존재
- 당연히 <head> 부터 읽어 들이고, <body> 를 나중에 읽습니다!

<head> 태그

- HTML 문서의 title, meta 에 대한 데이터로 화면에 표시되지 않음
- Title : HTML 문서의 제목 → 브라우저 탭에 표시
- Meta : 페이지 설명, 키워드, 저자, 화면 크기 등의 정보. 주로 브라우저 또는 검색 엔진에서 사용

<head> 태그, +@ 요소들

- Style : HTML 문서의 Style 정보 정의
- Link : 외부 리소스와의 연결 정보를 정의(CSS 파일 연계에 사용)
- Script : Javascript 를 정의

기초 코드를 불러와 봅시다!

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```

<meta>, charset

```
<meta charset="UTF-8" />
```

- 문자 인코딩에 대한 요약 정보를 나타내는 속성 → 문자가 어떤 방식으로 인코딩 되었는지를 표현
- 영문과 한글을 모두 사용하기 위해서는 UTF-8 을 많이 사용
- 생략해도 큰 문제는 없지만, 브라우저가 임의로 정하기 때문에 사용하고자 하는 문자가 깨질 가능성이 있으므로 표기하는 것을 추천

<meta>, http-equiv(alent)

- 해당 페이지에서 사용하는 HTTP 통신에 대한 정보를 나타낼 때 사용

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

- 최신 버전의 IE 를 사용하라고 하는 http-equiv 태그

```
<meta http-equiv="refresh" content="10" />
```

- 페이지를 10초에 한번 씩, 새로 고침 하라는 http-equiv 태그

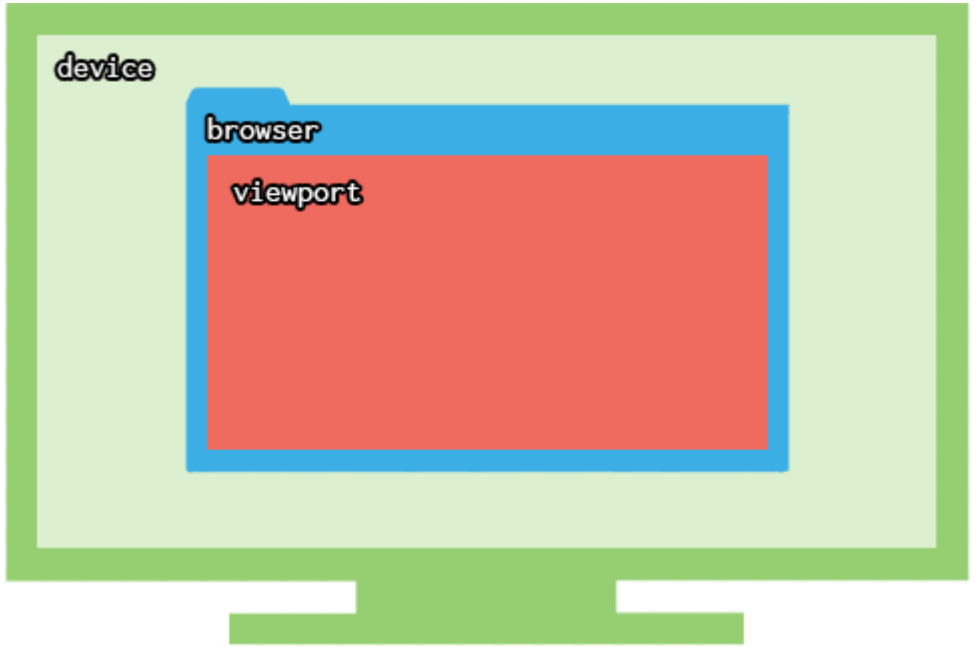
<meta>, name

- 페이지의 정보를 name 과 conetent 의 쌍으로 표기
- 저자(author), 설명(description), 키워드(keywords), 뷰포트(viewport) 등에 사용

```
<meta name="author" content="이효석" />
<meta name="description" content="KDT 5기 수업 코드" />
<meta name="keywords" content="웹개발자, HTML, CSS, JS, REACT" />
```

<meta>, viewport

- 실제로 화면에 보여지는 영역을 의미



<meta>, viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- 뷰 포트를 기기 넓이에 맞추고, 초기 배율은 1로 설정!
- 해당 정보를 바탕으로 웹 문서의 기본 적인 배율 조정이 일어 납니다!

<meta>, viewport 체험하기

- 뷰포트 메타 태그를 주석 처리하고 보기

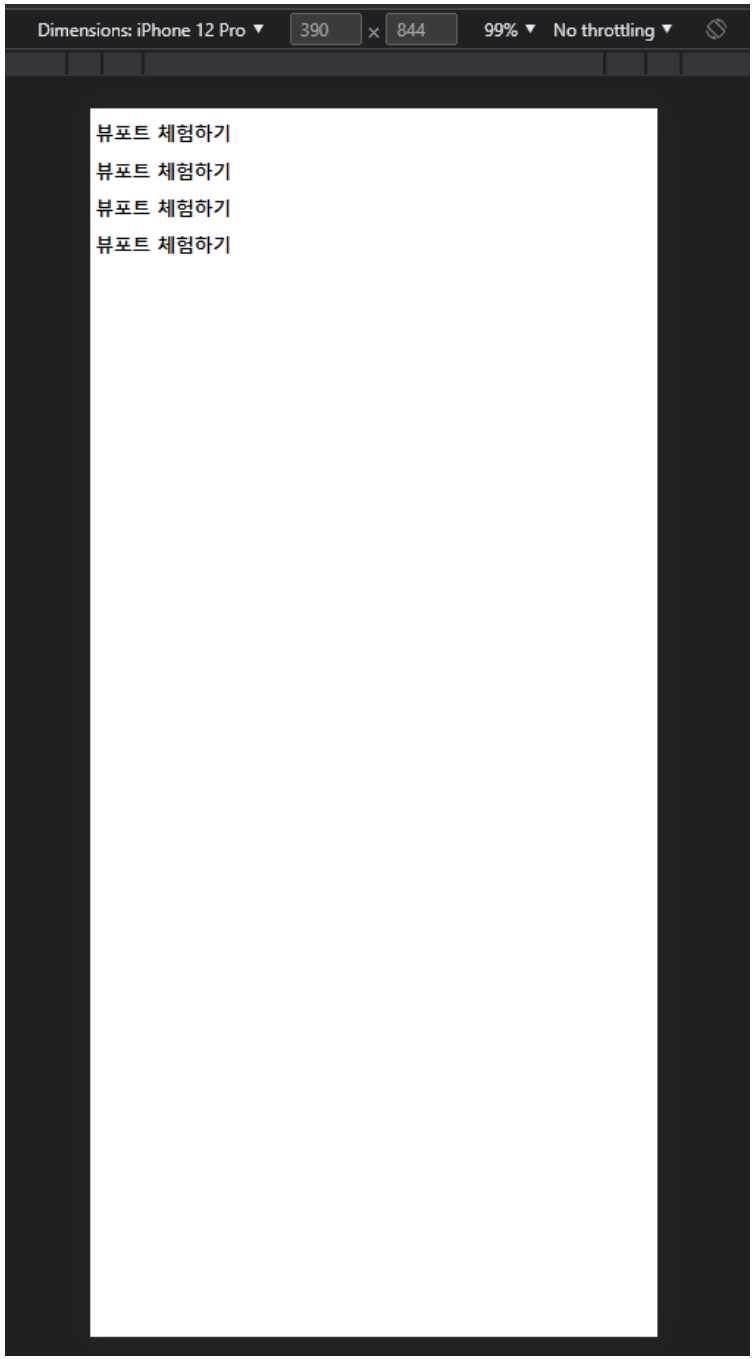
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <!-- <meta name="viewport" content="width=device-width, initial-scale=1.0" /> -->
    <title>Document</title>
  </head>
  <body>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
  </body>
</html>
```

뷰포트 체험하기

뷰포트 체험하기

뷰포트 체험하기

뷰포트 체험하기



<meta>, viewport 체험하기

- 뷰포트 메타 태그를 주석 해제하고 보기

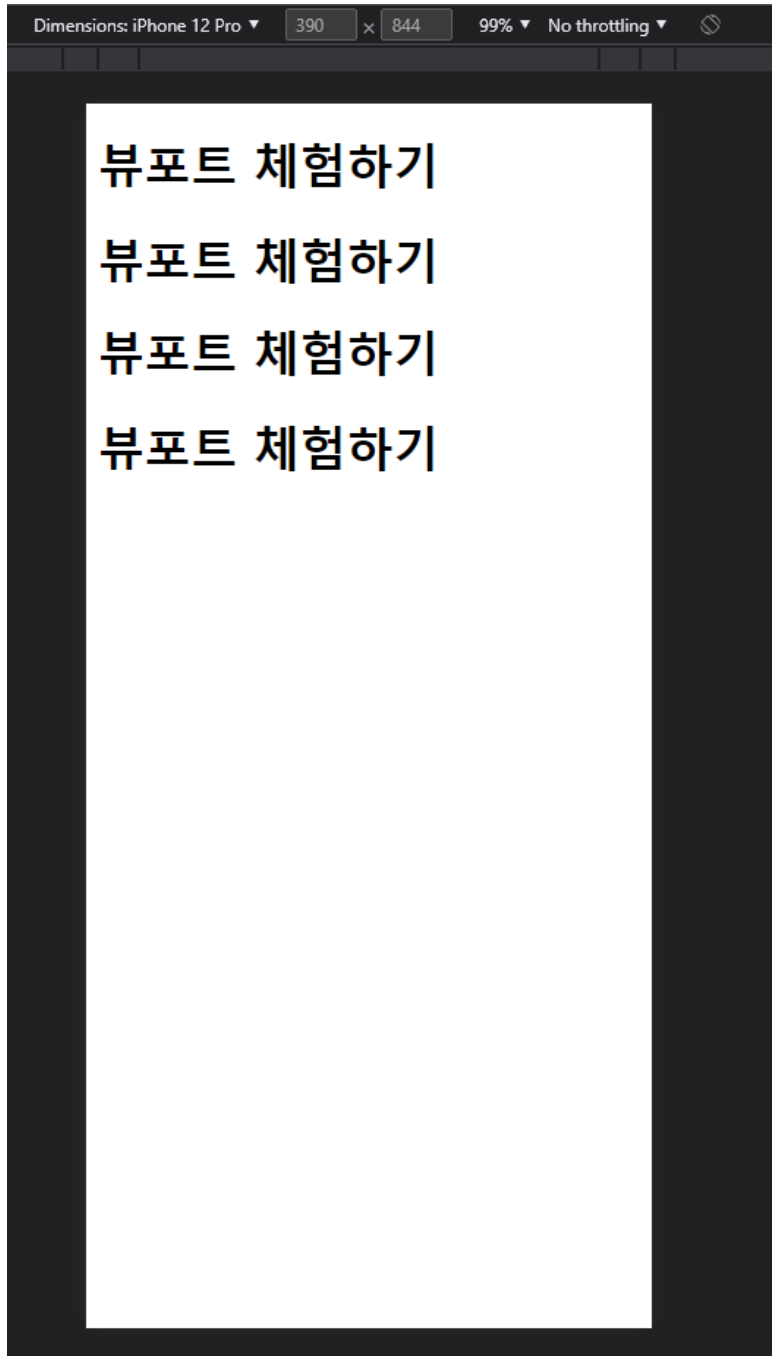
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
    <h1>뷰포트 체험하기</h1>
  </body>
</html>
```

뷰포트 체험하기

뷰포트 체험하기

뷰포트 체험하기

뷰포트 체험하기



<body>

<body> 태그

- HTML 문서의 내용을 담는 태그
- 웹페이지를 구성하는 대부분의 요소가 body 태그 내에 기술
- HTML 태그를 사용하여 사용자가 원하는 문서를 작성

<tag>

글자를 다루는 <tag>

제목 태그, <h1~6></h1~6>

- 제목을 뜻하는 Heading 의 약자, h 사용!
- 자동 줄 바꿈! 왜? 제목이니까!
- 하나의 HTML 문서에는 하나의 h1 태그를 권장
- 웹 검색 엔진이 제일 먼저 검색하는 태그!

Semantic Web

- 태그 자체로 컨텐츠에 대한 의미를 명확하게 설명할 수 있는 것
  - 개발자가 의도한 요소의 의미가 명확하게 나타남
  - <header>, <footer>, <img>, <title> 등등
- 검색 엔진이 웹 페이지를 검색 할 때, Semantic 요소를 검색

본문 태그,

- 본문을 뜻하는 paragraph 의 약자, 사용!
- 본문을 적기 위한 태그!

줄 바꿈 태그,

- 과연
은 무엇의 약자 일까요?
- 줄을 바꿔 준답니다!
- 그런데.... 그냥 엔터 치면 안되나요?
- 그럼 공백은요?

형식화 된 글자, <pre></pre>

- 형식화(preformatted)의 약자 <pre>를 사용
- 우리가 에디터에 쓴 대로 웹에서도 보여준답니다!
- 개발자들도
,   사용하기 귀찮기 때문이죠!
- 그런데 막상 거의 안씁니다... ☺

수평 줄, <hr>

- 수평으로 된 줄을 그어 줍니다!

문자 꾸미기 태그들

- <b> : 두껍게!
- : 두껍게! + Semantic 한 의미를 지님
- <i> : 이탤릭
- : Emphasized, 강조!
- <del> : 중간 줄!
- <u> : 밑 줄!

이미지 삽입하기

이미지를 넣어 봅시다!

- 이미지를 넣을 때 사용하는 <img> 태그
- 빈 요소, 닫는 태그가 필요하지 않음

이미지의 경로

이미지의 이름
(대체 텍스트 / Alternate Text)

```
<img src="./cat.jpg" alt="고양이" />
```

이미지 위치는!?

- Src 속성을 사용하여 이미지의 위치를 알려 줍니다~!
- 컴퓨터의 파일 또는, 인터넷의 주소(url)을 입력할 수 있습니다.

```
<img src="study.png" />
<img
  src="https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/dog-puppy-on-
garden-royalty-free-image-1586966191.jpg?crop=0.752xw:1.00xh;0.175xw,0&resize=1200:*"
/>
```

Alt(ernative)?

- 이미지의 대체 텍스트
- 이미지를 제대로 못 불러 왔을 때, 텍스트로 해당 이미지가 무었이었는지를 표시
- 시각장애인 분들이 웹을 사용할 때 사용하기 때문에 필수!

```
<img src="study.png" alt="공부를 결심하는 사진" />
```

크기 조절

- 속성 width 와 height 를 사용하여 크기를 조절

```
<img src="study.png" alt="공부를 결심하는 사진" width="300" height="300" />
```

- 가로 세로 중 하나의 크기만 입력하면 이미지의 비율에 맞춰서 나머지 한쪽은 자동으로 결정!

```
<img src="study.png" alt="공부를 결심하는 사진" height="500" />
```

문서의
영역 구분하기

컨테이너 태그

- 다른 요소 여럿을 묶어서 관리하기 위한 용도로 사용되는 태그 → '컨테이너'
- 컨텐츠나 레이아웃에 아무런 영향을 주지 않음
- 대표적으로 , 을 사용

```
<h1>이젠 정말 코딩 뿐이야</h1>
<p>코딩 열심히 해야지!</p>
<hr />
<img src="study.png" alt="공부를 결심하는 사진" width="300" />
<p>정말 공부 뿐이야</p>
```

```
<div>
  <h1>이젠 정말 코딩 뿐이야</h1>
  <p>코딩 열심히 해야지!</p>
  <hr />
</div>
<div>
  <img src="study.png" alt="공부를 결심하는 사진" width="300" />
  <p>정말 공부 뿐이야</p>
</div>
```

태그의 전역 속성

태그의 전역 속성

- 모든 HTML 태그에서 공통으로 사용할 수 있는 속성
- ID : 요소에 고유한 이름을 부여하는 식별자 속성
- CLASS : 요소를 그룹 별로 묶을 수 있는 식별자 속성
- STYLE : CSS 를 적용할 때 사용하는 속성
- TITLE : 요소의 추가 정보를 제공하는 속성, 이미지에 툴팁 제공.

```
<div id="text" class="division">
  <h1>이젠 정말 코딩 뿐이야</h1>
  <p>코딩 열심히 해야지!</p>
  <hr />
</div>
<div id="image" class="division">
  <img
    src="study.png"
    alt="공부를 결심하는 사진"
    width="300"
    title="나는 결심 했다"
  />
  <p>정말 공부 뿐이야</p>
</div>
```

실습, 일기 쓰기!

- 지금까지 배운 내용을 전부 사용하여 일기 쓰기
- 텍스트 태그, 이미지 태그, 문서 영역 구분, 태그 전역 속성을 전부 사용하여 오늘의 일기를 써주세요!
- 총 3가지의 영역으로 구분하여 일기를 작성하세요 ☺

링크

HTML의 꽃 하이퍼링크, <a>

- 기존 문서나 텍스트의 선형성, 고정성의 제약에서 벗어나 사용자가 원하는 정보를 취득할 수 있는 기능을 제공
- Anchor 의 약자인 <a> 태그 사용
- 속성 값
  - href : Hypertext Reference 의 약자, 이동할 페이지의 링크
  - target : 링크 된 문서를 열었을 때 문서가 열릴 위치 표시
    - _blank(새 탭) / _self(현재 탭)

HTML의 꽃 하이퍼링크, <a>

- Href 에는 전화 또는 메일 주소 등을 지정 가능

```
<a href="tel:010-3930-1325" target="_self">전화 걸기</a>
<a href="mailto:xenosign@naver.com" target="_blank">메일 쓰기</a>
```

목록

목록

- 연관 있는 항목들을 나열하는 태그
- 순서 있는 목록과, 순서 없는 목록으로 구분

목록, or

- 순서 없는 목록
- 순서 있는 목록
- 은 속성 사용 가능
  - <ol type="?"> : 말머리 기호 변경(1, A, a, I, i)
    - 1 / A / a / I / i
  - <ol start="?"> : 시작 값 변경
  - <ol reversed> : 역순으로 시작

중첩 목록!

```html
<!DOCTYPE html>
<html>
  <body>
    <h2>중첩 목록</h2>
    <ul>
      <li>Coffee</li>
      <li>
        Tea
        <ol>
          <li>Black tea</li>
          <li>Green tea</li>
        </ol>
      </li>
      <li>Milk</li>
    </ul>
  </body>
</html>
```

- **직관적**으로 부모자식 관계 파악 가능!

실습, 카페 음료 문서 작성

- 카페 이름은 <h1>
- 태그와 <li> 사용해서 메뉴 작성, 음료 최소 3개
- 메뉴 이름은 <h2>
- 메뉴의 가격은 태그와 <li> 태그로 표현하기
- 메뉴 설명은 <p> 태그 사용
- 메뉴끼리는 <hr /> 태그로 구분
- 문자 꾸미기 태그 전부 사용하기!

실습, 카페 음료 문서 작성

Tetz 카페

- **아이스 아메리카노**
    a. **3500원**

시원한 아메리카노

- **라떼는 말이야**
    A. *7000원*

따뜻한 라떼

- **달달한 핫 초코**
    i. *3000원*

달콤한 핫 초코

- **달달한 핫 초코**
    i. ~~3000원~~

달콤한 핫 초코

- **달달한 핫 초코**
    i. 3000원

달콤한 핫 초코

사용자 입력 받기

선택 메뉴를 만드는, <select>

- 선택 메뉴(드롭 다운)를 만드는 태그!
- <select> : select 폼 생성
  - Name : select 박스의 이름
- <option> : select 폼의 옵션 값 생성
  - Value : 실제적으로 전달 되는 값
  - Selected : 최초에 선택 된 값으로 설정
- <optgroup> : option 을 그룹화
  - Label : optgroup 이름 설정
- Disabled : 옵션은 보이지만 선택을 못하도록 설정

```
<select name="menu">
  <optgroup label="coffee">
    <option value="americano">아메리카노</option>
    <option value="latte">라떼</option>
    <option value="cappuccino">카푸치노</option>
  </optgroup>
  <optgroup label="other">
    <option value="lemonade">레모네이드</option>
    <option value="hotchoco">핫초코</option>
    <option value="smoothie">스무디</option>
  </optgroup>
</select>
```

입력 값 받기! <input>

- type
  - Button
  - Text
  - Password
  - Checkbox
  - Radio
  - Date
  - Color

버튼, type="button"

- 버튼을 생성
- 주로 특정 기능을 수행 시킬 때 사용

텍스트, type="text"

- 텍스트 입력 값을 받는 폼을 생성
- 우리가 입력하는 ID 입력 부분?
- 텍스트 값을 입력 받아 전달하는 기능

패스워드, type="password"

- 비밀번호 값을 받는 폼을 생성
- 입력 값을 자동으로 안보이게 처리
- 중요 or 비밀 텍스트 값을 전달하는 기능

체크 박스, type="checkbox"

- 여러 선택지 중 여러 개를 선택 가능한 체크 박스 생성
- 속성
  - Name : 체크 박스의 이름, 같은 분류의 체크 박스는 같은 이름으로 설정
  - Value : 체크 박스가 실제로 전달하는 값을 지정
  - Checked : 화면 최초 로딩 시에 선택 된 상태로 로딩

```
<input type="checkbox" name="menu" value="coffee" /> 커피
<input type="checkbox" name="menu" value="coffee" /> 커피
<input type="checkbox" name="menu" value="coffee" /> 커피
```

라디오 버튼, type="radio"

- 여러 선택지 중 하나만 선택 가능한 라디오 버튼 생성
- 속성
  - Name : 라디오 버튼의 이름, 같은 name 을 가지는 라디오 버튼은 하나만 선택이 가능 → 하나를 선택하면 다른 선택 값이 취소 됨
  - Value : 라디오 버튼이 실제로 전달하는 값을 지정
  - Checked : 화면 최초 로딩 시에 선택 된 상태로 로딩

```
<h3>radio</h3>
<input type="radio" name="gender" value="male" checked /> 남자<br />
<input type="radio" name="gender" value="female" /> 여자<br />
<hr />
```

날짜 선택, type="date"

- 특정 날짜(년/월/일)를 선택
- 속성
  - Name : 날짜 선택 폼 이름
- type="datetime-local"
  - 시간 까지 선택 가능!
  - 기존은 "datetime" 을 사용하였으나 시간은 시간대의 영향을 받기 때문에 정확한 데이터 값 전달이 불가능 하여 지금은 사용 X

실습, 카페 음료 주문 문서 작성

- Input 태그를 활용해서 주문자 이름 받기
- 전역 속성 id 사용하여 구역 나누기 + <hr> 태그로 선 긋기
- 단체 주문 파트
  - Checkbox 로 각각 메뉴 여러 개를 주문할 수 있도록 구성
  - 주문 날짜 입력 받기
- 개별 주문 파트
  - Radio 로 하나의 메뉴만 주문 할 수 있도록 구성
  - 주문 날짜 및 시간 까지 입력 받기

Tetz 카페

주문자 이름을 입력 하세요

단체 주문

☐ 커피 ☐ 차 ☐ 라떼

주문 날짜 입력

연도-월-일

개별 주문

○ 커피 ○ 차 ○ 라떼

주문 날짜 및 시간 입력

연도-월-일 -- --:--

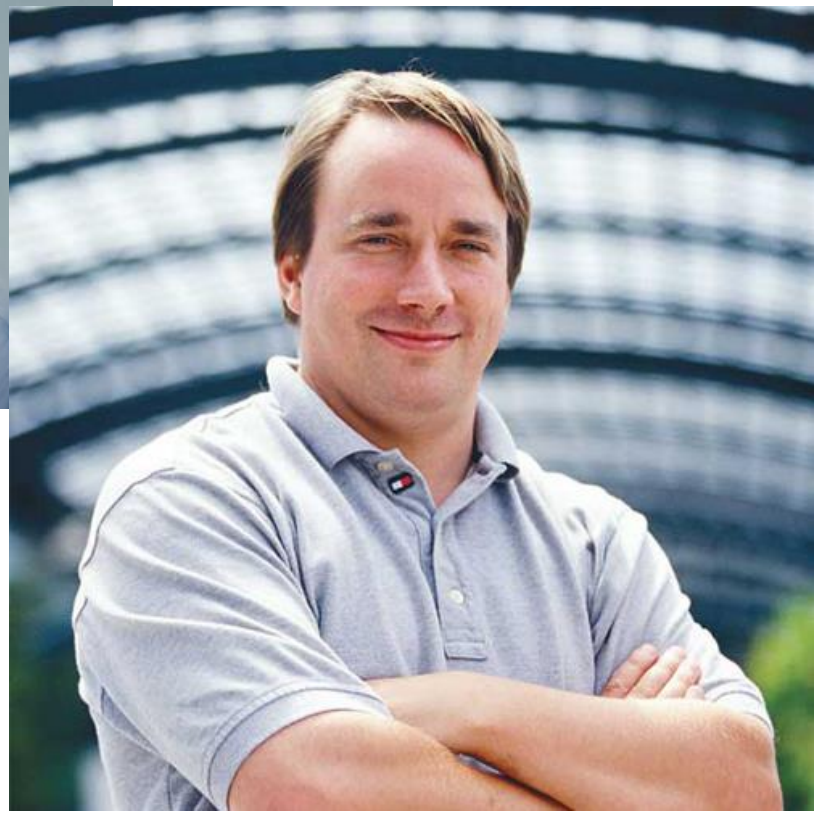
git

git

Git(깃)은 컴퓨터 파일의 변경사항을 추적하고
여러 사용자들 간에 해당 파일 작업을 조율하기 위한
대표적인 버전 관리 시스템(VCS)입니다.

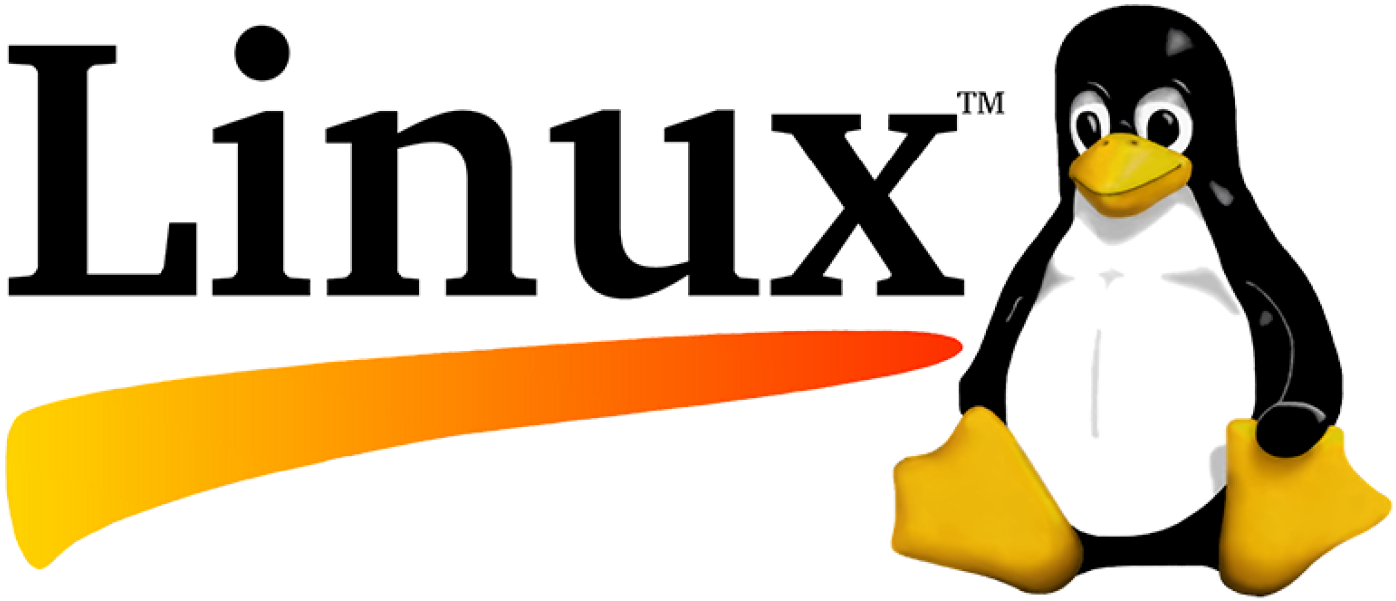
VCS(Version Control System)

세계 3대 개발자!?



Linus Benedict Torvalds

Linux(Linus + unix)



VCS(Version Control System)

Git이 하는 일은!

버전 관리

버전 관리


v_0.1
v_1.0
v_1.0_최종
v_1.3_제발_최종
v_1.5_더 이상은Naver


Commit
Commit
Commit
Commit
Commit

Git의 특징!

- 폴더 단위로 관리! → 하나의 프로젝트 단위로 관리
- Git에 대한 모든 정보는 .git 폴더에 있습니다
  - 일종의 블랙 박스!
- 커밋은 로컬(= 여러분의 컴퓨터)에서만 일어나는 행위입니다!
- 따라서, 깃 허브는 여러분의 로컬에 있는 깃 정보를 온라인에 저장해 주는 서비스를 제공하는 것입니다!
- 깃 허브의 리포지토리는 여러분 로컬의 폴더의 개념과 같아요!

깃 설치하기!

GitHub

https://github.com/

Desktop

https://desktop.github.com/

Git, 설치하기(Windows)

https://git-scm.com/

Download for Windows

Click here to download the latest (**2.36.1**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **24 days ago**, on 2022-05-09.

Other Git for Windows downloads

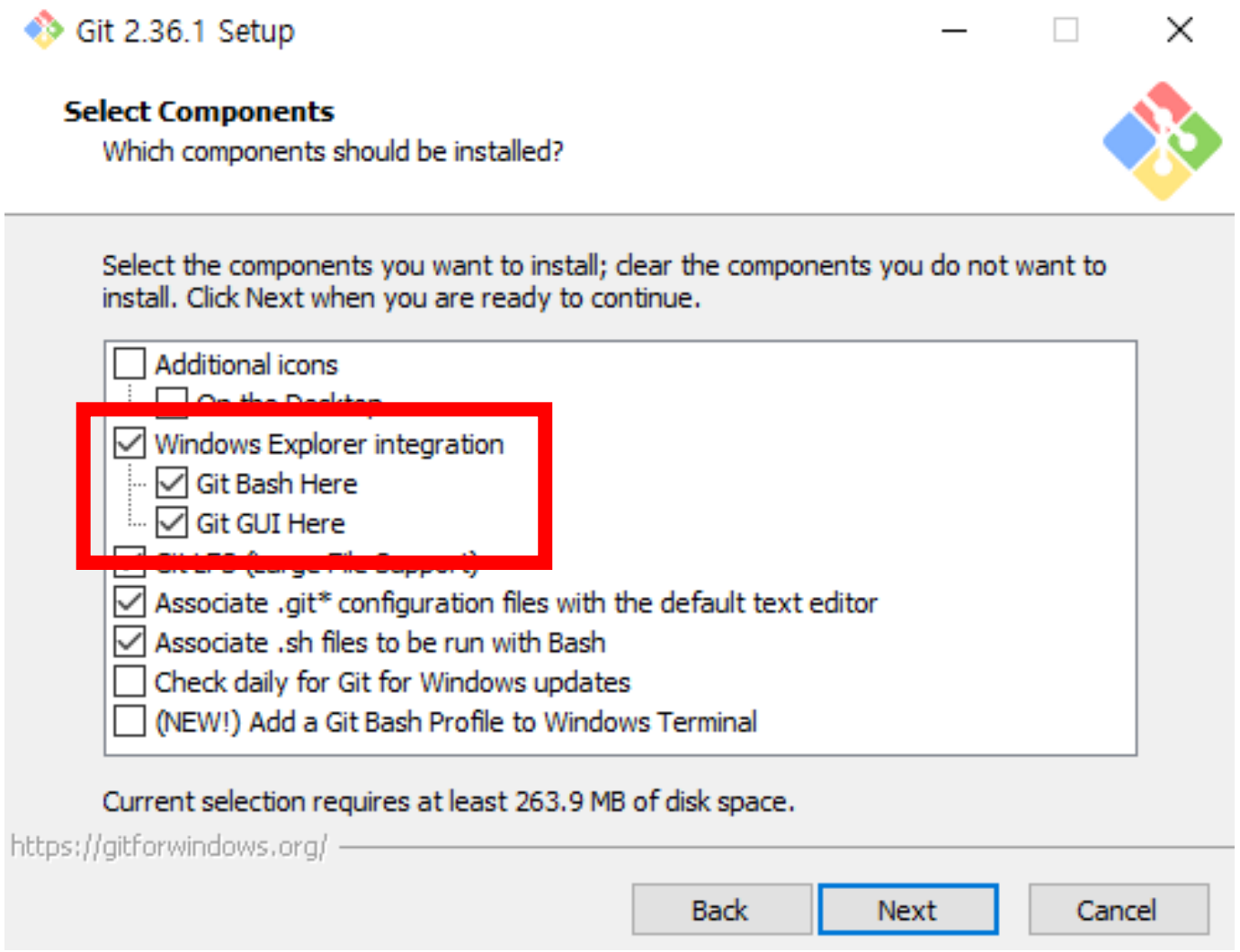
Standalone Installer
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

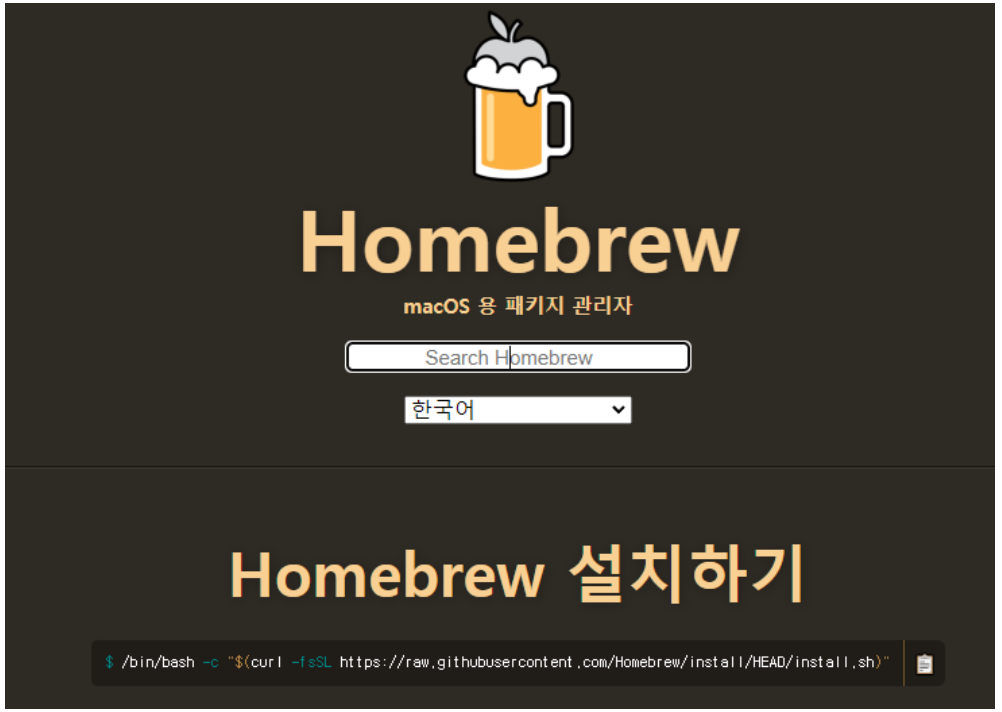
Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

Git, 설치하기(Windows)



Git, 설치하기(Mac)



https://brew.sh/index_ko

brew install git

Commit?

project

로컬에 설치

사용자
(local, 로컬)

Git hub 사용을 위한 세팅 시작!

- git init
- git config --global init.defaultBranch main
- 개행 문자 관련 처리(윈도우는 CR, LF / 맥은 LF 만)
  - git config --global core. autocrlf true (Window, CRLF → LF)
  - git config --global core.autocrlf input (Mac, LF 만 사용)
- git config --global user.name “프로필 이름”
- git config --global user.email “이메일 주소”
- git config --lobal --list

```
# 개행 문자(Newline) 설정
## macOS
$ git config --global core.autocrlf input
## Windows
$ git config --global core.autocrlf true

# 사용자 정보
## 커밋(버전 생성)을 위한 정보 등록
$ git config --global user.name 'YOUR_NAME'
$ git config --global user.email 'YOUR_EMAIL'

# 구성 확인
## Q키를 눌러서 종료!
$ git config --global --list
```

project

로컬에 설치

사용자
(local, 로컬)

```
$ git init
# 현재 프로젝트에서 변경사항 추적(버전 관리)을 시작.
```

git

사용자
(local, 로컬)

master

index.html

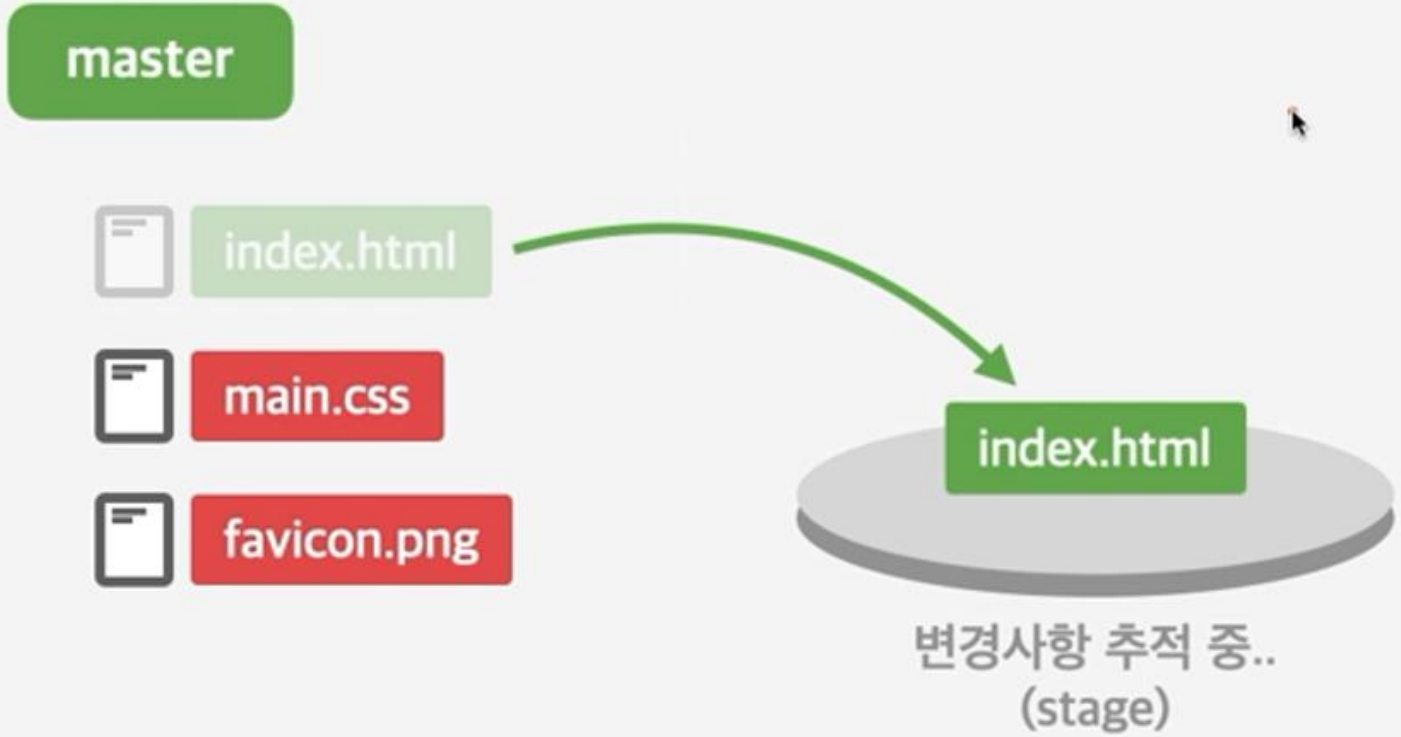
main.css

favicon.png

변경사항 추적 중..
(stage)

```
$ git add index.html
# 변경사항을 추적할 특정 파일(index.html)을 지정.
```

git 사용자 (local, 로컬)



```
$ git add .
# 모든 파일의 변경사항을 추적하도록 지정.
```

git 사용자 (local, 로컬)

master

```
$ git commit -m '프로젝트 생성'
# 메시지(-m)와 함께 버전을 생성.
```

git

사용자
(local, 로컬)

master

- index.html
- main.css
- favicon.png

● 프로젝트 생성

Git, 커밋 메시지!?

```
$
```

git 사용자 (local, 로컬)

master

- index.html
- main.css
- favicon.png
- main.js 파일 생성

프로젝트 생성

```
$ git add .
# 모든 파일의 변경사항을 추적하도록 지정.
```

git 사용자 (local, 로컬)

master

- index.html
- main.css
- favicon.png
- main.js

프로젝트 생성

```
$ git commit -m 'main.js 추가'
# 메시지(-m)와 함께 버전을 생성.
```

git 사용자 (local, 로컬)

master

- index.html
- main.css
- favicon.png
- main.js

main.js 추가

프로젝트 생성

```
$
```

git 사용자 (local, 로컬)

master

index.html 수정함

main.css 수정함

favicon.png

main.js

main.js 추가

프로젝트 생성

```
$ git add .
# 모든 파일의 변경사항을 추적하도록 지정.
```

git 사용자 (local, 로컬)

master

- index.html
- main.css
- favicon.png
- main.js

main.js 추가

프로젝트 생성

```
$ git commit -m 'index.html 수정'
# 메시지(-m)와 함께 버전을 생성.
```

git

사용자
(local, 로컬)

master

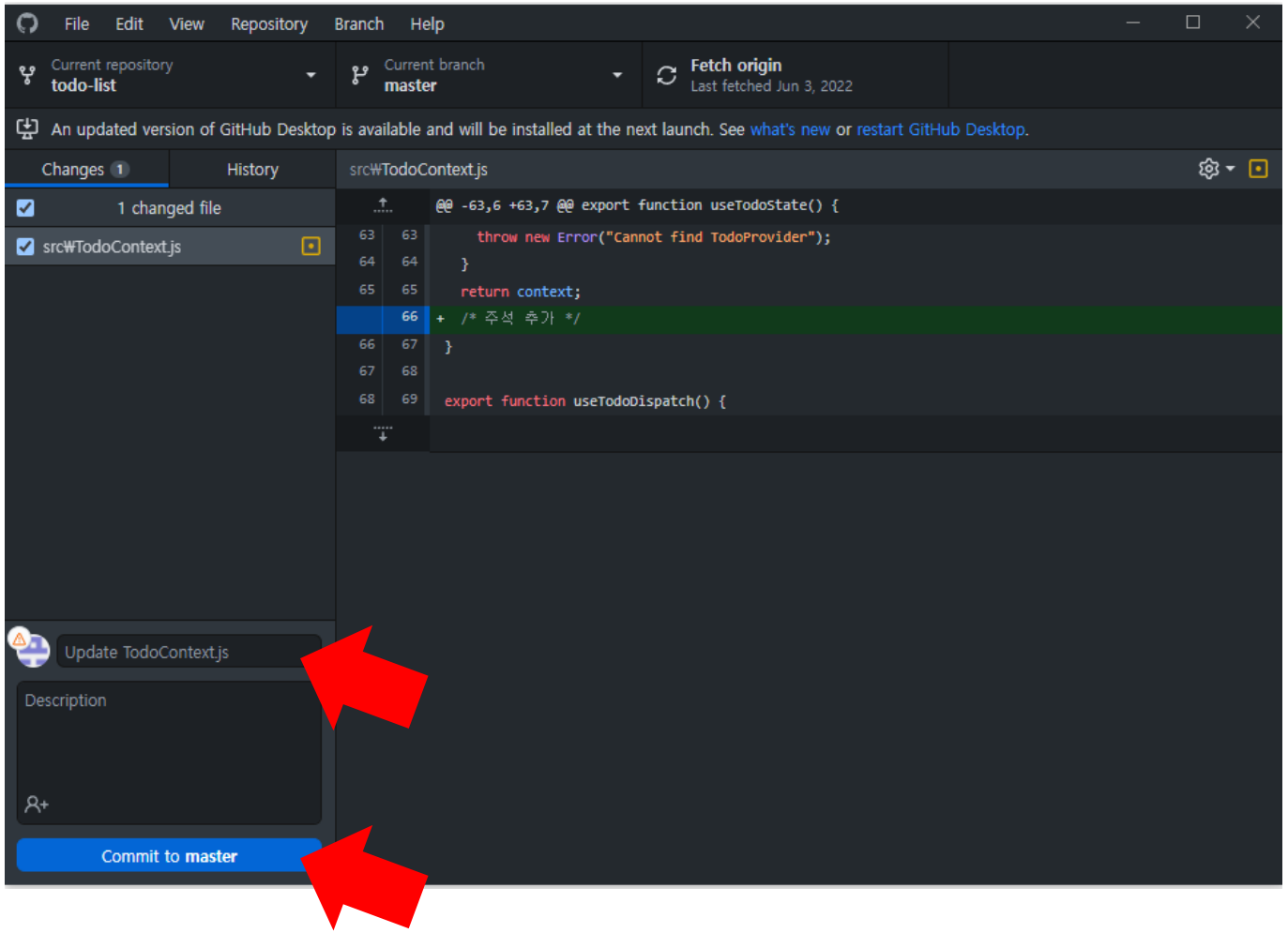
- index.html
- main.css
- favicon.png
- main.js

- index.html 수정
- main.js 추가
- 프로젝트 생성

Git, 커밋하기 on CLI

- git status
- Untracked files?
- git add 가 추천 되네요?
- git add .
- git status
- git commit -m “커밋 메세지”
- git log

Git, 커밋하기 on GUI



실습, 깃 커밋 연습

- 깃 커밋을 위한 빈 폴더 만들기
- 해당 폴더에 index.html 파일 생성
- 해당 내역 커밋하기(by CLI)
- 해당 폴더에 test.html 파일 생성
- 해당 내역 커밋하기(by CLI)
- 각각의 파일을 수정하고 커밋하기(by CLI)
- 같은 작업을 깃 허브 데스크탑으로도 수행 (별도의 폴더 생성)

Push!?


MEMZ
BSOD INCOMING
ENJOY THE NYAN CAT


GitHub

GitHub
(remote, 원격)

원격 저장소
(Repository)

git 사용자 (local, 로컬)

GitHub (remote, 원격)

```
$ git remote add origin https://github.c...
# origin이란 별칭으로 원격 저장소를 연결.
```

master

- index.html
- main.css
- favicon.png
- main.js

원격 저장소 (Repository)

git 사용자 (local, 로컬)

GitHub (remote, 원격)

```
$ git push origin master
# origin이란 별칭의 원격 저장소로 버전 내역 전송.
```

master

- index.html
- main.css
- favicon.png
- main.js

원격 저장소 (Repository)

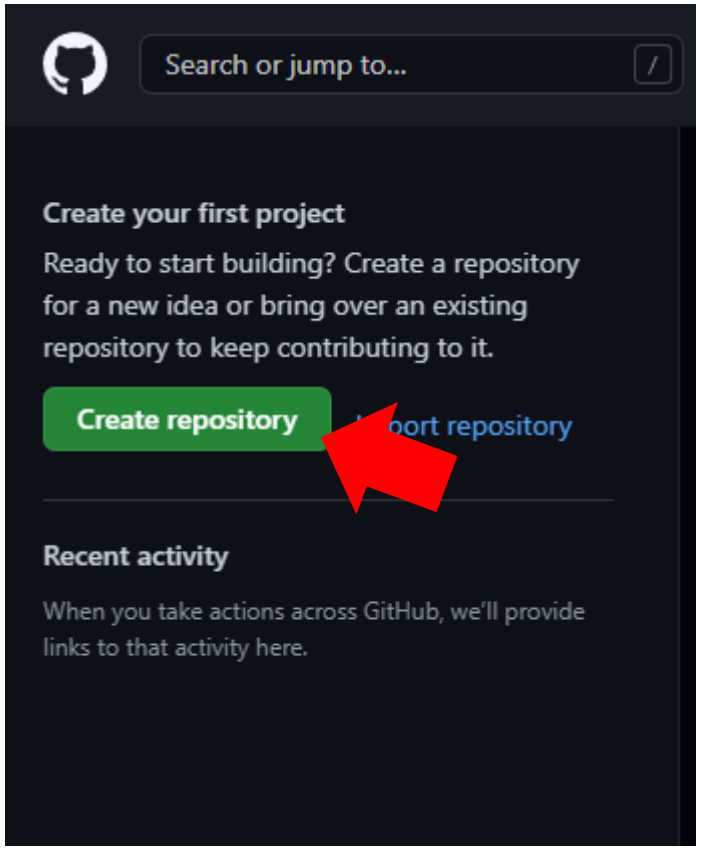
Push, 온라인에 깃을 저장하자!

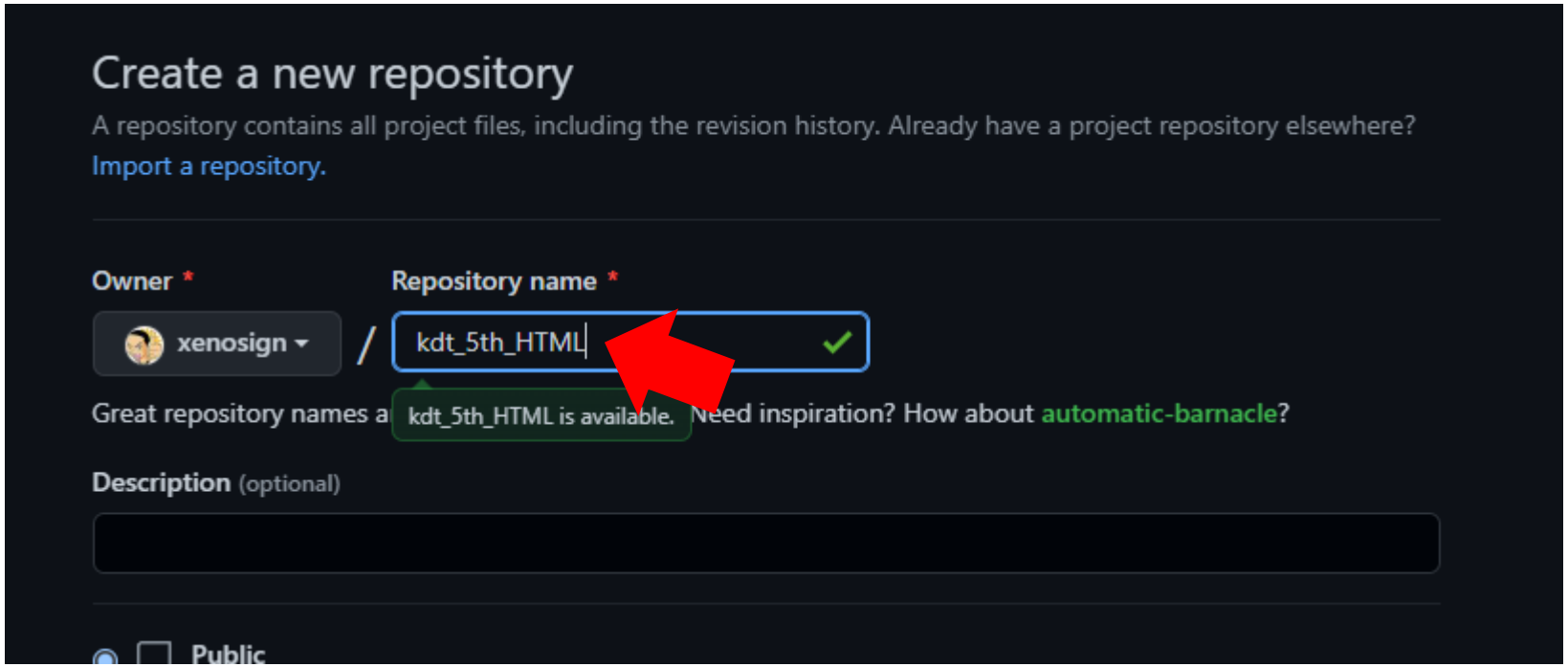
- Commit 은 여러분의 컴퓨터에서만 일어나는 일입니다!
- 즉, 온라인에 해당 내역을 올리지 않으면 github은 아무것도 모르는 거죠

Push, 온라인에 깃을 저장하자!

- Github에 여러분의 커밋 내역을 올리려면 반드시 push를 해주어야 합니다!
- Git이 관리하는 폴더 하나(=프로젝트) → github 리포지토리 하나
  - 두 개를 하나에 올리면, 코드의 변화를 추적하는 깃의 특성상 황당하겠죠?
  - 자동차 2대의 블랙 박스를 하나의 블랙 박스에 넣는 것과 비슷합니다!

Github 에 올리기! – Repo 생성





Github 에 올리기! – 안내 따라하기!

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

```
echo "# kdt_1st_HTML" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/kdtTetz/kdt_1st_HTML.git
git push -u origin main
```

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

```
git remote add origin https://github.com/kdtTetz/kdt_1st_HTML.git
git branch -M main
git push -u origin main
```

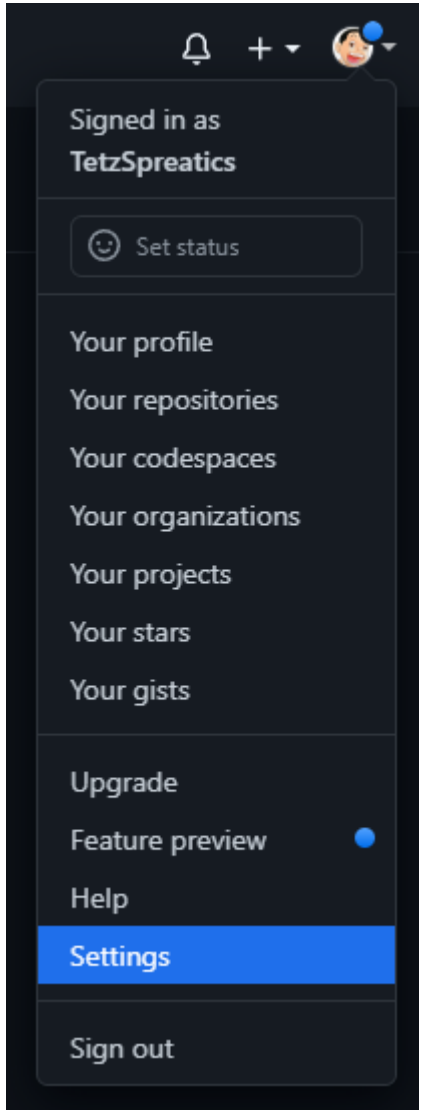
- 상황에 맞는 코드를 복사해서 Vscode 에 붙여 넣기!

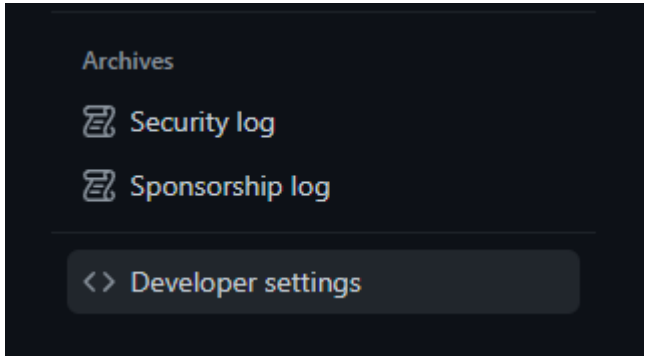
Github 에 올리기!(Push)

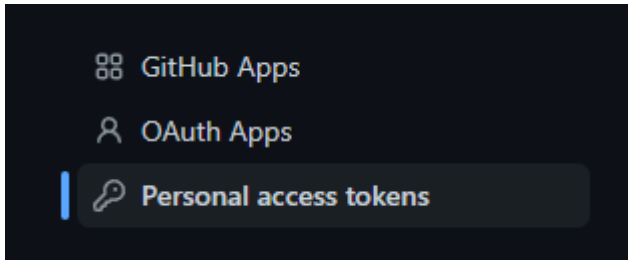
- echo "# ss" >> README.md
- git init
- git add .
- git commit -m "커밋 메세지"
- git branch -M main
- git remote add origin https://github.com/TetzSpreatics/ss.git
- git push -u origin main

문제 사항 처리

Access Token 발행!







토큰 저장해 두기!

Settings / Developer settings

- GitHub Apps
- OAuth Apps
- Personal access tokens
  - Fine-grained tokens (Beta)
  - Tokens (classic)

Personal access tokens (classic)

Generate new token | Revoke all

Tokens you have generated that can be used to access the GitHub API.

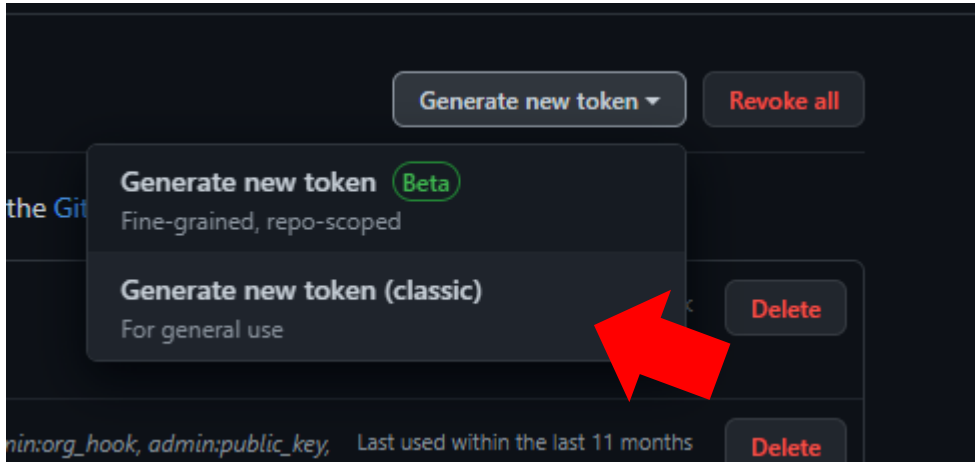
git — *repo*
Last used within the last week — Delete

⚠ This token has no expiration date.

toy project — *admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, delete:packages, delete_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages*
Last used within the last 11 months — Delete

Expired *on Sat, May 28 2022.*

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.


Generate new token
Revoke all
Generate new token
Beta
Fine-grained, repo-scoped
Generate new token (classic)
For general use
Delete
Last used within the last 11 months
Delete

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What's this token for?

Expiration *

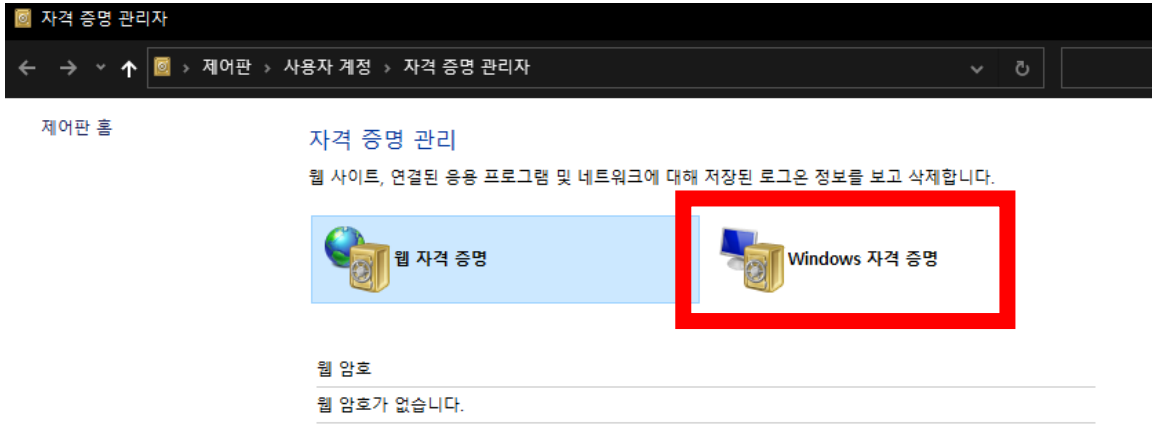
30 days ken will expire on Sat, Feb 25 2023

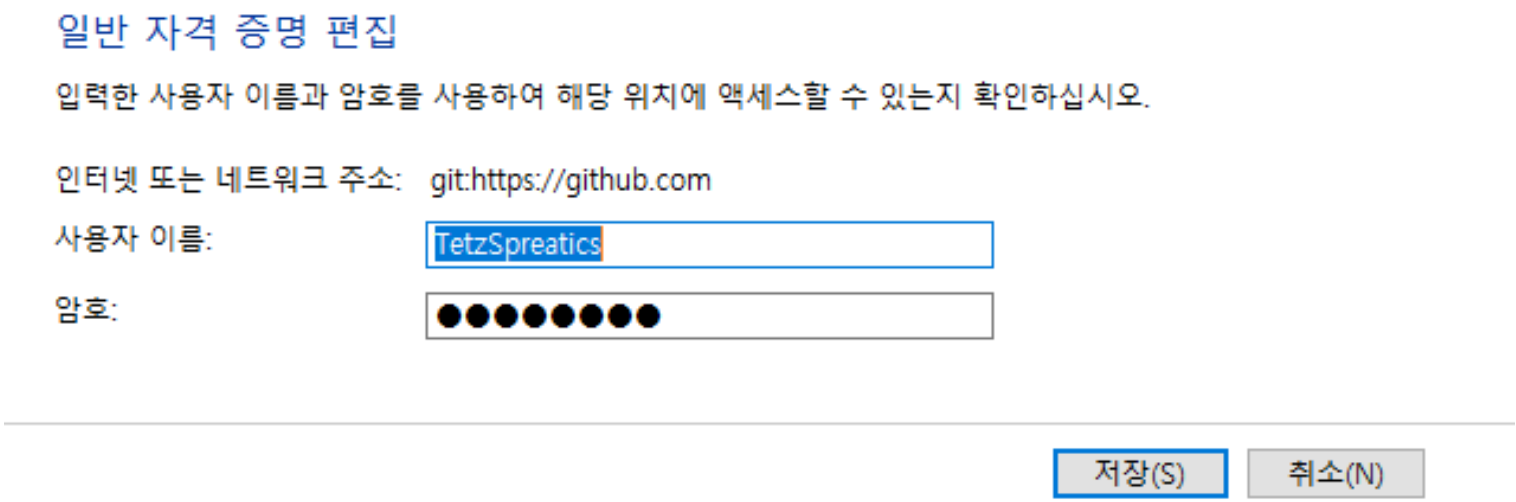
Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

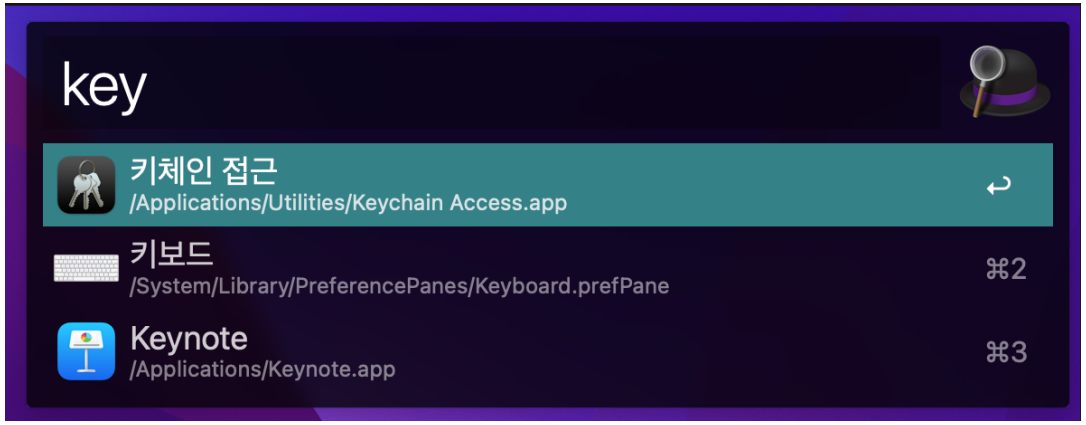
repo	Full control of private repositories
repo:status	Access commit status
repo_deployment	Access deployment status
public_repo	Access public repositories
repo:invite	Access repository invitations
security_events	Read and write security events
workflow	Update GitHub Action workflows
write:packages	Upload packages to GitHub Package Registry

For Windows


자격 증명 관리자
제어판 › 사용자 계정 › 자격 증명 관리자
제어판 홈
자격 증명 관리
웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그온 정보를 보고 삭제합니다.
웹 자격 증명
Windows 자격 증명
웹 암호
웹 암호가 없습니다.


일반 자격 증명 편집
입력한 사용자 이름과 암호를 사용하여 해당 위치에 액세스할 수 있는지 확인하십시오.
인터넷 또는 네트워크 주소: git:https://github.com
사용자 이름: TetzSpreatics
암호: ●●●●●●●●
저장(S)
취소(N)

For Mac


key
키체인 접근
/Applications/Utilities/Keychain Access.app
↩
키보드
/System/Library/PreferencePanes/Keyboard.prefPane
⌘2
Keynote
/Applications/Keynote.app
⌘3


키체인 접근
git
기본 키체인
로그인
로컬 항목
시스템 키체인
시스템
시스템 루트
모든 항목
암호
보안 메모
내 인증서
키
인증서
github.com
종류: 인터넷 암호
계정: xenosign
위치: https://github.com
수정일: 2022. 2. 27. 오후 9:41:42
이름
종류
수정일
만료
키체인
로그인
로그인
로그인
github.com
속성
접근 제어
이름: github.com
종류: 인터넷 암호
계정: xenosign
위치: https://github.com
설명:
암호 보기:
변경 사항 저장

블랙 박스와 비슷합니다!

- 하나의 프로젝트 = 하나의 폴더 = 한대의 차
- A 라는 차가 사고가 났을 때, B 라는 차의 블랙 박스를 확인하면 될까요?
- 또, 급하다고 해서 C 라는 차의 블랙 박스 데이터를 D 라는 차에 덮어 씌워도 될까요?

실습, 깃 푸쉬 연습

1. 실습을 위한 github repository 생성
2. 1의 폴더에 main.html 파일 생성 후 커밋
3. 해당 내역을 생성한 github repository 에 푸쉬하기

모를 심어 봅시다!

Signed in as
TetzSpreatics

Set status

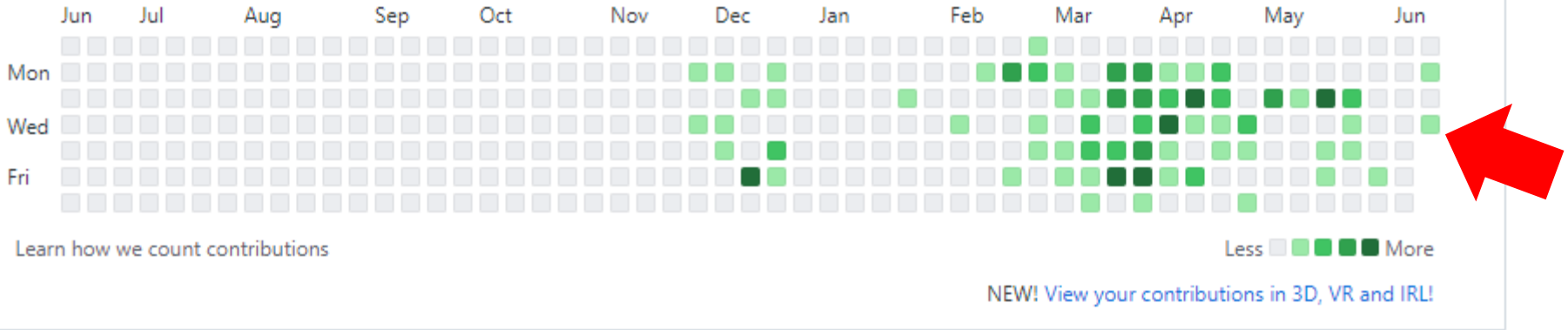
Your profile
Your repositories
Your codespaces
Your organizations
Your projects
Your stars
Your gists

Upgrade
Feature preview
Help
Settings

Sign out

238 contributions in the last year

Contribution settings



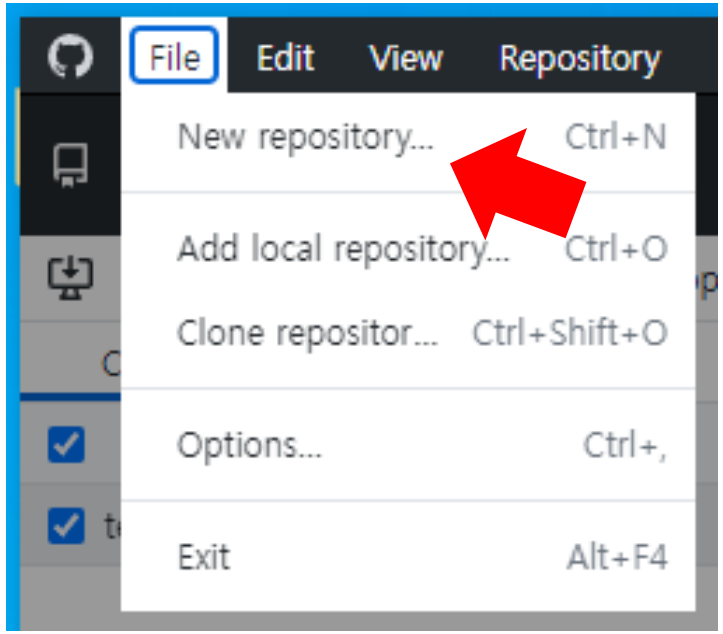
Learn how we count contributions

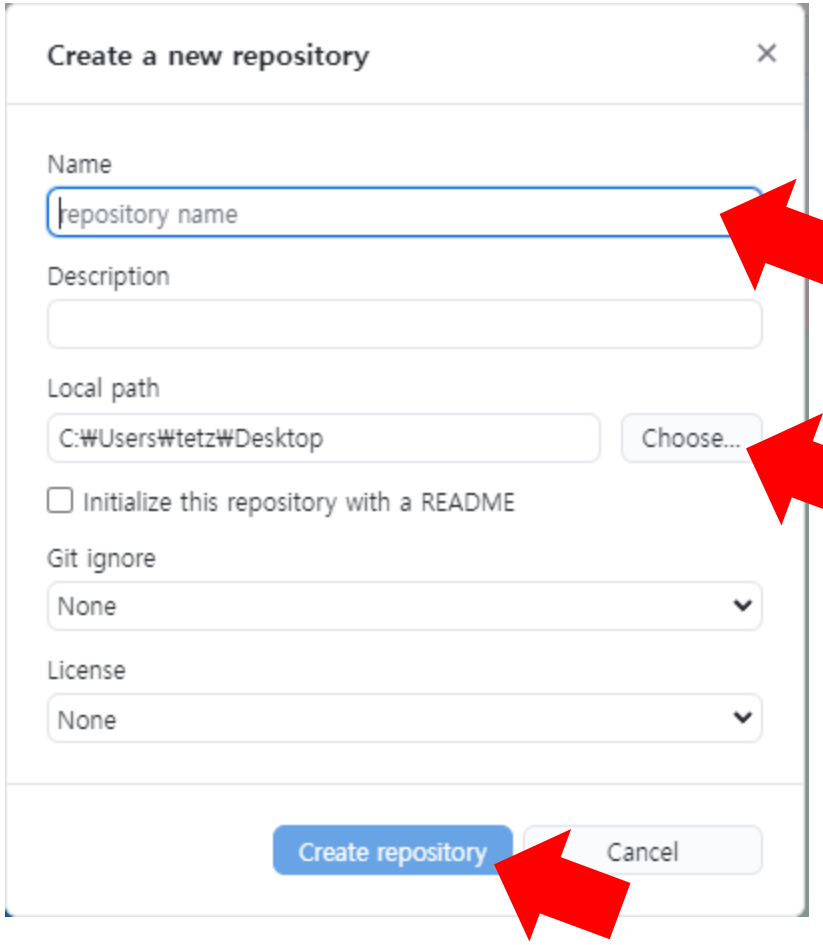
Less More

NEW! View your contributions in 3D, VR and IRL!

귀찮으면

Github desktop


File
Edit
View
Repository
New repository... Ctrl+N
Add local repository... Ctrl+O
Clone repositor... Ctrl+Shift+O
Options... Ctrl+,
Exit Alt+F4

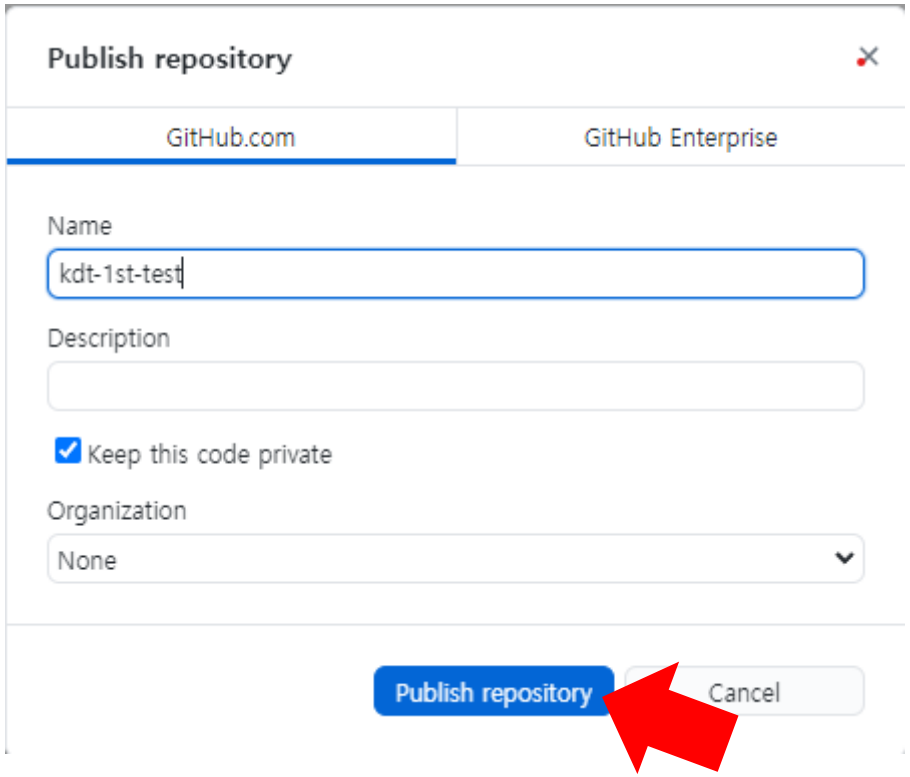

Create a new repository
Name
repository name
Description
Local path
C:₩Users₩tetz₩Desktop
Choose...
Initialize this repository with a README
Git ignore
None
License
None
Create repository
Cancel

Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

Always available in the toolbar for local repositories or Ctrl P

Publish repository



Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl Shift A

Open in Visual Studio Code


수고하셨습니다!