

HTML 기초 및 VScode 셋팅

HTML/CSS/JS

2022/09/21 17:01

<http://blog.naver.com/zhsks880/222880610094>

*Visual Studio Code 설치(VSCode)

- 1.구글에서 vscode 검색해서 비주얼스튜디오 코드를 다운 받아서 설치하고
 - 2.라이브러리 가서
 - 3.korean (한국어패치)
 - 4.beautify (코드정렬)
 - 5.liveserver (코드 친거 바로 확인하는거)
- >설치하고, 환경설정가서 단축키
->파일->설정->바로가기 키->검색:beautify->beautify file 앞에 + 누르고
->beautify 컨트롤+알트+로 설정하자

*HTML 기초(Hyper Text Markup Language)

- 1.태그: 명령어 형태가 <> 로 되어있는것들
 - 2.요소: 태그의 시작<>과 끝</>의 한쌍
 - 3.마크업: 요소들을 이용하여 웹문서를 작성
 - 4.속성: 태그의 요소에 지정하는것들
- ex: a href="#" #"

*부모요소와 자식요소

-태그는 중첩 기능으로 부모 자식요소를 단계적 구조로 만들수 있음

```
<nav>
<h1>대전 맛집</h1>
<ul>
<li>성심당1</li>
<li>성심당2</li>
<li>성심당3</li>
</ul>
</nav>
```

*블록 요소

-블록요소는 마크업시 자동으로 줄바꿈이 일어나는 요소(세로배치)

-블록요소 안에는 또다른 블록요소, 인라인 요소 포함 가능

-너비: 사용가능한 최대치 까지 점유(화면비율)

-크기: 가로, 세로의 크기 지정 가능

-대표태그: div, h1~h6, p 태그 등

<h1>HTML 공부중~ 지금은 블록 요소를 확인 중입니다.</h1>

<div>블록 요소의 성질</div>

<p>블록 요소는 수직으로 배치됩니다.</p>

*인라인 요소

-인라인요소는 마크업시 자동으로 줄바꿈이 안되는 요소(가로배치)

-인라인요소 안에는 텍스트와 인라인 요소만 가능하며 블록 요소는 포함 불가능

-너비: 필요한 만큼(쓰인 만큼)

-크기: 가로, 세로 크기 지정 못함

-대표태그: span, a , img 태그 등

<h1>인라인 요소</h1>

인라인 요소는 수평으로 배치됩니다.

인라인 요소 링크입니다.

HTML 태그 종류

HTML/CSS/JS

2022/09/22 16:46

<http://blog.naver.com/zhsks880/222881485853>

태그 명	설명
h1~h6	-문서의 제목이나 목차 지정 시 사용 -h1이 폰트가 제일 크고 나머지 순차적으로 작아짐
header	-웹페이지의 소개 및 메뉴등에 사용 -웹페이지 로고, 검색폼, 메뉴 네비게이션 등을 포함하는 요소로 사용
footer	-사이트 최 하단에 위치 -작성자, 저작권 정보, 관련 문서 등 내용 저장 용도
article	-사이트 안에서 독립적으로 구분해서 배포하는 영역 -게시판, 블로그 글, 매거진, 뉴스 기사 등에 사용
section	-문서의 일반적인 영역 - section->h1~h6 형식으로 사용
aside	-주로 사이드 바나 광고 배너를 마크업 할때 사용
nav	-문서의 부분 중 다른페이지로의 링크를 보여주는 영역 -메뉴, 목차, 색인 등 사용
div	-아무 의미가 없는 영역을 구성할때 사용 -CSS로 꾸며서 사용할 목적으로 사용 -의미를 가진 다른 요소(article, nav등) 가 적절하지 않을때만 사용
ol	-각 항목(li) 의 정렬된 목록 -자식 태그로 li 만 사용가능
ul	-각 항목(li) 의 정렬되지 않은 목록 -자식 태그로 li 만 사용가능
li	-ol, ul 의 자식 태그로 단독 사용 불가
p	-하나의 문단을 설정할 때 사용 -문단의 분리를 위해 설정
hr	-빈태그 -화면 줄긋기

인라인 텍스트 태그

a	-하이퍼링크 생성 시 사용(URL생성)
	-a href="주소"
abbr	-머리글자 선택 시 사용
b	-독자의 주의를 끌기위한 용도 사용
	-요약키워드, 리뷰의 제품명 등 굵게 표시할 부분에 사용
mark	-하이라이팅 할 때 사용
em	-단순한 의미강조
	-정보통신 보조기기등에서 강하게 발음
strong	-아주 높은 중요도를 표시
	-굵고 크게!
span	-인라인 텍스트 콘텐츠를 위한 공용 컨테이너로 별 의미는 없음
	-블록요소의 div와 흡사함
br	-빈태그
	-문장의 줄바꿈

멀티미디어 태그

img	src	-이미지 URL(필수)
	alt	-이미지의 대체텍스트(필수)
		-이미지 깨질때 나올 텍스트
	width	-이미지의 가로 너비
	height	-이미지의 세로 너비
	srcset	-브라우저에게 제시할 이미지 URL과 원본크기의 목록을 정의(x축, y축 으로 구분)
	sizes	-미디어 조건과 해당 조건일 때 이미지 최적화 크기의 목록을 정의

<audio src=""></audio>

audio	autoplay	-준비되면 바로 재생
	controls	-제어 메뉴 표시(플레이 바 만들기)
	loop	-재생이 끝나면 다시 처음부터 재생(무한반복)
	src	-콘텐츠 URL
	muted	-음소거 여부

<video src=""></video>

video	autoplay	-바로 재생 (true, false 값 지정)
	controls	-제어 메뉴 표시 (true, false 값 지정)
	loop	-무한반복 (true, false 값 지정)
	src	-콘텐츠 URL
	muted	-음소거 (true, false 값 지정)
	poster	-동영상 썸네일 이미지 URL
	width	-동영상 가로너비
	height	-동영상 세로너비
<iframe src="" frameborder="0"></iframe>		
iframe	name	-프레임의 이름
	src	-포함할 문서의 URL
	width	-프레임의 가로너비
	height	-프레임의 세로너비
	allowfullscreen	-전체화면 모드 사용 여부(true,false 값 지정)
	frameborder	-프레임 테두리 사용 여부(0,1 값 지정)

표만드는 태그

<table></table>

<table border="1">

tr -표의 행 추가

<tr>

th -표의 열 추가

<th>이름</th>

td -표의 열의 데이터 삽입

<th>나이</th>

<!-- 세로 셀 병합: rowspan -->

<td rowspan="4" align="center">

<button>세로버튼</button>

</td>

</tr>

<tr>

<td>김철수</td>

<td>30살</td>

</tr>

<tr>

<td>박영희</td>

<td>25살</td>

```
</tr>
<tr>
<!-- 가로 셀 병합: colspan -> 병합하고 싶은 셀의
개수를 작성 -->
<td colspan="4" align="center">
<button>버튼입니다!</button>
</td>
</table>
```

HTML 입력양식, 전역속성

HTML/CSS/JS

2022/09/23 17:06

http://blog.naver.com/zhsks880/222882385613

HTML 입력양식

form

-form 요소는 웹 서버에 정보를 제출하기 위한 양식

<form action=""></form>

action	-전송한 정보를 처리할 URL	URL형식	
name	-양식의 이름 설정		
method	-서버로 전송할 HTTP방식	GET, POST	
autocomplete	-사용자가 이전에 입력한 값으로 자동완성 기능을 사용할 것인지 선택	on, off	
novalidate	-서버로 전송시 양식 데이터의 유효성을 검사하지 않도록 지정	논리값(true, false)	
target	-서버로 전송 후 응답받을 방식을 지정	_self _blank	

input

-input 요소는 사용자에게 입력 받을 데이터 양식 지정

-빈태그

<input type="text">

type	-입력받을 데이터의 종류		
	text	일반 텍스트	
	password	비밀번호	별별별별처리
	email	이메일	
	search	검색	
	checkbox	체크박스	
	radio	라디오 버튼	같은 name 속성 그룹 내 하나만 선택 가능
	number	숫자	
	button	일반버튼	<button>처럼 사용

	submit	서버 전송 버튼	form 내용을 서버로 전송
	reset	초기화	form 내부 양식 모두 제거
	file	파일 업로드	
	range	범위 컨트롤	min, max, step, value속성 사용
	hidden	보이지 않지만 서버로 전송할 내용	value속성으로 값을 지정
value	-양식의 초기값		
name	-양식의 이름		
placeholder	-사용자가 입력할 값의 힌트	text	type이 text, search, tel, url, email인 경우만 가능
checked	-양식이 선택된 상태로 시작	논리값	type이 checkbox, radio 일 경우만 가능
disabled	-양식을 비활성화		
readonly	-수정 불가능한 읽기전용		
button	<button></button>		-input의 type속성을 통해 버튼을 만들 수도 있지만 좀 더 명시적인 버튼을 만들 수도 있음 -input 내에 만드는 버튼은 텍스트 값 지정밖에 할수 없지만, button 요소로 생성 시 , , 등 다른 요소를 사용하여 복잡한 렌더링도 가능함 -form양식 안에 작성 -여러줄 의 일반 텍스트 폼 양식
textarea	<textarea name="" id="" cols="30" rows="10"></textarea>		-rows속성으로 행 수 지정 가능 -form양식 안에 작성
fieldset & legend	<fieldset> <legend>Coffe Size</legend> <input type="radio" name="size">Tall <input type="radio" name="size">Grande <input type="radio" name="size">Venti </fieldset>		-fieldset: 테두리를 만들어서 어울리는 제목으로 그룹화 시킨다 -form양식 안에 들어가는건 다 name 이 붙는다. 그래야 입력받을 수 있음 -form양식 안에 작성
select	<select name="drink">		-드롭다운 형식 작성


```

<optgroup label="Coffee">
<option value="ame">Americano</option>
<option value="esp">Espresso</option>
<option value="cl">Caffe Latte</option>
<option value="capp">Cappuccino</option>
</optgroup>
</select>

```

-select 안에 optgroup 으로 그룹화 시킬수 있음
 -select 안에 들어갈 값들은 option 으로 목록화 하여 삽입해야 함
 -option value = 내가 얻을 값 설정

HTML 전역속성

: 전역속성이란 모든 HTML 요소에서 공통 사용 가능한 속성을 말하는 것

class

```

<button id="join-btn" class="btn">Join</button>
-> class는 .btn

```

-class 속성은 요소의 별칭을 지정
 -공백을 통해 여러개의 클래스를 지정할 수 있음
 -CSS 나 JS의 요소 선택기를 통해서 요소를 제어할 때 사용

id

```

<button id="join-btn" class="btn">Join</button>
-> id 는 #join-btn

```

-id 속성은 문서에서 고유한 식별자를 정의할 때 사용
 -class 속성은 중복 설정이 가능하지만 id는 하나만 설정이 가능함
 -CSS에서는 대부분 class 속성으로 지정을 많이하고, JS에서는 id 속성을 많이 사용함(절대는 아님)

style

```

<style></style>

```

-style 속성은 요소에 직접 스타일을 지정할 때 사용 됨
 -</title> 과 </head> 사이에 사용을 주로 하며 html 본문에 작성 시 인라인 스타일이라고 함

title

```

<h1 title="정말 크다!!">우리집 크다</h1>

```

-title 속성은 요소에 마우스를 포커스 했을 때의 설명을 지정

data-*

-사용자 정의 데이터 속성을 지정하면 html 에 JS에서 이용할 수 있는 데이터를 저장해 두는 용도로 사용

*상대경로

: 나를 기준으로 상대적으로 찾아가는 경로

./(생략 가능)

***절대경로**

:절대적 기준으로 찾아가는 경로

:http://~~ , c:\game\riot

*HTML에서 특수기호를 제공하는 문법이 있긴 하나 크게 쓰이지는 않음

 => 스페이스 공백

< => <

> => >

& => &

CSS, 선택자

HTML/CSS/JS

2022/09/26 13:11

<http://blog.naver.com/zhsks880/222884540050>

*CSS

: Cascading Style Sheets 의 줄임말로 HTML등의 마크업 언어로 작성된 문서가 웹사이트에 표현되는 방법을 정해주는 언어

->레이아웃, 폰트, 색상처리 등 꾸미기를 위한 언어라고 보면 됨.

*내부 스타일 시트

-> html 문서 안에 <style>태그로 기재하는 형식

-> html 상단의 <head></head> 사이에 작성하는게 일반적임

*외부 스타일 시트

-> 외부 css 파일을 만들어 두어 html 문서에 파일명으로 연결해주는 형식

-> link rel = "stylesheet" href="css/style.css" 이런 형식

*인라인 스타일 시트

-> html body 내부의 요소에 직접 스타일을 하는 방식

-> 유지보수에 부담을 주므로 사용을 권고하지는 않음(비상시에는 쓸 수도 있음)

*선택자(Selector)

-> CSS로 UI를 디자인 할 때 "어디" 에 해당하는 지목하는 것

-> h1{color: blue; background: yellow;} << h1 영역

*CSS 선택자

선택자 명	선택자 설명	선택자 기호
전체선택자	-모든 태그를 지정할 때 사용 -"*" 별 사용	*
태그선택자	-태그 이름을 그대로 가져다 쓰는 선택자 -"태그명" 사용	p{color: blue;} 등
클래스선택자	-html 내부에 class를 불러다 쓰는 선택자 -"." 점 사용	.class{color: blue;} 등

아이디선택자	-html 내부에 id를 불러다 쓰는 선택자 -"#" 삽 사용	#id{color: blue;} 등
자식선택자	-직속 자식만 선택 -">" 화살표 사용	ul>.orange{color: orange;}
후손선택자	-바로 인접 자식이 아닌 후손 요소중에 선택이 가능 -" " 스페이스바 사용	div .red-color{color: orange;}
인접형제선택자	-현재 요소 바로 뒤에 나오는 형제 요소 한개만 선택 -"+" 기호 사용	.orange + li {color: orange;}
일반형제선택자	-현재 요소 바로 뒤에 나오는 형제 요소 전체 선택 -"~" 기호 사용	.orange ~ li {color: orange;}
속성선택자	-속성을 선택해서 사용 -"[속성명]" 사용	input[type]{color: red;}
가상선택자		
hover	-마우스 포인트 위에 놓을 시 효과 줌 -" :hover" 사용	a:hover {font-size: 20px;}
focus	-마우스 포인트 클릭 시 효과 -" :focus" 사용	input:focus {border-color: red;}
first-child	-first: 요소 중 첫번째 요소 지목 -" :first-child" 사용	.fruits>li:first-child {color: red;}
last-child	-last: 요소 중 마지막 요소 지목 -" :last-child" 사용	.fruits>li:last-child {color: red;}
nth-child	-요소 중 n번째 요소 -" :nth-child()" 사용	.fruits>li:nth-child(4) {color: red;}
nth-of-type	-같은 종류의 요소 중 n번째 요소 -" :nth-of-type()" 사용	.fruits>li:nth-of-type(1) {color: red;}
before	-해당 요소 앞에 가상 요소 추가 -" ::before" 사용	ul li::before {color: red;}
after	-해당 요소 뒤에 가상 요소 추가 -" ::after" 사용	ul li::after {color: red;}

Box 속성

HTML/CSS/JS

2022/09/26 14:46

<http://blog.naver.com/zhsks880/222884629573>

*단위

px	-픽셀 단위를 뜻하며, 화면에 고정적인 길이를 의미하며 절대적인 수치를 고정시킴	width: 100px;
%	-부모요소 대비 비율지정	width: 50%;
em	-자기 자신의 font-size에 영향을 받음 -h1 font-size: 10px; 이고 width: 5em 으로서는 가로 길이는 5 X 10px 해서 50px이 됨	width: 5em;
rem	-html 기본 폰트사이즈 로 계산됨 -html 기본 폰트사이즈 16px이므로, 3rem=48px 로 처리 됨 -css초기화 시키면 안되는 단위설정	width: 3rem;
vw	-뷰포트(사용자 모니터 환경)를 기준으로 설정되는 길이 단위 -0~100 사이 값 지정이 가능함 :100vw = 사용자 모니터 가로 가득!	width: 100vw;
vh	-뷰포트(사용자 모니터 환경)를 기준으로 설정되는 길이 단위 -0~100 사이 값 지정이 가능함 :100vh = 사용자 모니터 세로 가득!	height: 100vh;
	-가로 길이 지정	
	-세로 길이 지정	

*속성

	-요소의 바깥쪽 여백을 지정	margin: 10px 20px 30px 40px;
	-px, em, % 등 단위로 지정	위 오른 아래 왼
	-기본값 0	margin: 10px 20px 40px;
margin	-값을 %로 지정하면 부모요소의 width의 비율로 여백을 가짐	위 (좌우) 아래
	-좌우 마진을 auto로 지정하면 박스가 중앙 정렬	margin: 10px 40px;
	-박스 밖으로 밀어내기	(위아래) (좌우)
		margin: 10px;
		(상하좌우)

padding	-요소의 내부 여백을 지정	padding: 10px 20px 30px 40px;
	-px, em, % 등 단위로 지정	위 오른 아래 왼
	-기본값 0	padding: 10px 20px 40px;
	-값을 %로 지정하면 부모요소의 width의 비율로 여백을 가짐	위 (좌우) 아래
	-패딩을 사용하면 그만큼 박스 크기가 증가함	padding: 10px 40px;
	-박스 안에서 밀어넣기	(위아래) (좌우) padding: 10px; (상하좌우)
border	-요소의 테두리선을 지정	
	-border-width : 선의 두께	
	-border-style : 선의 종류	border: 1px solid red;
box-sizing	->solid, dashed, dotted, double, none 등 있음	
	-border-color : 선의 색상	
	-padding, border 에 의해 박스 사이즈가 조절되는 경우를 제어할 때 사용	-box-sizing: border-box; ->원래 지정한 사이즈를 해치지 않게 하기 위해 때 사용
display	-요소가 화면에 보여지는 특성을 지정	
	-값으로는	
	block: 블록 요소를 지정	display: block;
	inline: 인라인 요소를 지정	
	inline-block: 인라인 요소이면서 가로,세로 너비 지정 가능한 요소로 지정	
	none: 요소를 사라지게 함	
overflow	-박스 안의 내용이 박스보다 클 경우 넘치는 부분을 제어	
	-값으로는	
	visible: 전체를 다 보여줌	overflow: auto;
	hidden: 넘치는 콘텐츠를 숨김	
	auto: 넘치는 콘텐츠가 존재할 경우에만 스크롤바를 생성	
	scroll: 넘치는 콘텐츠가 없어도 스크롤바를 생성	
Font 속성		
font-style	-글자 스타일(기울기) 지정	font-style: italic;

	-값으로는 italic: 이탤릭체 oblique: 기울어진 글자	
	-글자 두께 지정 -값으로는 normal: 기본 두께 bold: 두꺼운글자, 700값 bolder: 부모요소보다 더 두껍게 lighter: 부모요소보다 더 얇게 숫자 100~900까지 설정이 가능하지만 대부분 bold 만 함	font-weight: bold;
font-weight		
	-글자 크기 지정 -값으로는 px, em(16px), %	font-size: 30px;
font-size		
	-줄 높이를 지정 -값으로는 normal, 숫자, 단위, % 가 있으나, 높이값이랑 같이 하면 그 줄에 폰트가 꽉차게 표시 됨	line-height: 100%;
line-height		
	-글꼴 설정 -값으로는 지정하고 싶은 폰트 모양으로 불러와서 셋팅 가능	font-family: arial, "open sans", "바탕", sans-serif;
font-family		
	-문자 정렬 방식 -값으로는 left, right, center, justify 가 있으나, 대부분 center(가운데 정렬로 씀)	text-align: center;
text-align		
	-문자의 장식 지정 -값으로는 none, underline, overline, line-through 가 있 으나, 대부분 underline 을 많이 씀	text-decoration: underline;
text-decoration		
Background 속성		
background-color	-요소의 배경 색상을 지정	background-color: red; background: red; <<이렇게 써도 됨
background-image	-요소의 배경에 이미지를 삽입 -크게 쓸일이 잘 없음 요즘은...	background-image: www.haha.img;

	-배경 이미지의 크기를 설정	
	-값으로는	
	auto: 원본이미지 크기 표시	
	px, em, % 등으로 지정도 가능	
background-size	cover: 배경 이미지를 박스 크기 위로 덮는거로 생각하자	background-size: contain;
	이미지가 잘릴 수 있음	
	contain: 배경 이미지를 박스 크기 안에 넣는걸로 생각하자	
	박스의 빈 공간이 보일 수 있음	
Box 띄우기		
	-요소를 좌우 방향으로 띄움	
	-가장 많이 쓰는거 같음, float 처리 후 html문서	
float	안에 해당 클래스 가서 clearfix 해줘야 박스가 내 려 았음	float: left; .class clearfix <<<이걸 꼭 해줘라
	-값으로는	
	left, right 있음	
Box position		
	-형제 요소들에게 영향을 받거나 주게 되는 형태	
	로 2번박스가 이동해도 3번의 위치는 변하지 않 음	position: relative; bottom: 40px;
relative	-기준에 있던 자리는 그대로 유지하고 나만 가게 되며 기존 요소랑 겹친다	left: 160px;
	-기준 자기 위치는 사라짐	
	-position을 가지고 있는 부모가 기준이 되며 position:relative 필수	position: absolute; right: 100px;
absolute	-부모요소에서 position이 없으면 body기준으로 움직임	bottom: 150px;
	-무조건 body기준으로 지정되며, 스크롤을 해도 해당 위치 고수 함(따라가질 않음)	position: fixed; right: 10px;
fixed	-맨위로가기 버튼 을 지정할 때 많이 씀	top: 50px;
	-sticky속성은 top, bottom, right, left 중에 하 나를 필수로 설정해야 합니다	postion: sticky;
sticky	-설정한 위치에 도달하기 전까지는 static처럼 동	top: 0;

작하다가 설정한 위치에 도달하면 스크롤 영역내
에서 fixed 처럼 동작함
-웹페이지 스크롤 하다 사라지거나 아예 밑으로
가거나 하게 할 때 사용

-transition-property: 전환 효과를 사용할 속성
명

-transition-duration: 전환 효과의 지속시간 설
정

transition

-transition-timing-function: 타이밍 함수를 지
정

transition: 2s cubic-bezier(0.8, -0.2, 0.94,
0.4);

-transition-delay: 전환 효과의 대기시간 설정

-박스 늘리고 줄이고 이런거 할때 썼는데 요즘도
쓸라나...

transform

-박스 풍차돌리기 할때 썼던건데, 이것도 요즘 쓸
라나..

transform: skew(30deg, 50deg);

animation

-애니메이션 속성은 JS 가 동반 되는것으로 추후
에 JS쪽에서 배우는걸로 다시 작성

float 사용시

HTML/CSS/JS

2022/09/27 16:46

<http://blog.naver.com/zhsks880/222885699579>

```
.clearfix::after {  
content: '';  
display: block;  
clear: both;  
}
```

부트스트랩을 숙지하자

HTML/CSS/JS

2022/09/29 12:53

<http://blog.naver.com/zhsks880/222887326081>

트위터 회사에서 만들었으며, 부트스트랩 현재는 5버전까지 있으나, 3버전으로 공부를 하게 된다. IE호환성 때문에...그리고 3버전까지만 한글로 된 설명서가 있고 4부터는 다 영문임

사이드 포폴 사이트, reset CSS

HTML/CSS/JS

2022/10/04 11:45

<http://blog.naver.com/zhsks880/222891302335>

뭐할지 생각해라

***reset.css**

HTML 본문에

Head 구문 안에 아래 주소 그대로 복사해서 넣기

<link rel="stylesheet" href="<https://cdn.jsdelivr.net/npm/reset-css@5.0.1/reset.min.css>">

Js_변수, 데이터타입

HTML/CSS/JS

2022/11/09 11:50

<http://blog.naver.com/zhsks880/222924081443>

- *기본적으로 자바스크립트는 html 위에서 동작함.
- *스크립트가 시작된다는 뜻의 <script> 태그를 사용.
- *자바스크립트는 동적 프로그래밍 언어.
- *자바스크립트는 "" ' ' 두개다 사용 가능하지만 JS에선 " 를 많이 씀
- *자바는 ' ' 단일 문자열을 나타내는거니까 헛갈려 하지 마라
- *f12 개발자 도구-> Console 탭에서 해당 sysout=console.log 확인 가능함

*변수선언 : 자바와는 다르게 타입 지정은 할 필요 없으며, var(옛날방식), let(변수선언), const(상수선언) 등으로 선언하여 사용 가능합니다.

- *식별자 : 변수나 함수에 들어가는 이름을 일컫는말
- : 숫자로만 이루어지거나, 숫자로 시작하는거 허용하지 않음
- (자바와 동일)
- : 특수기호 오면 허용하지 않음 ("_", "\$" 두개만 가능)
- (자바와 동일)

var 700 = 365; (에러)
var 7num = 7; (에러)
var num7 = 7; (정상)
var n7u7m7 = 7; (정상)

*데이터 타입

1. 숫자타입

- : 자바와는 다르게 숫자의 형변환등이 필요 없이 알아서 계산이 된다.
- : int, float, long 등 불필요함
- : 10 / 4 = 2.5 로 자동 계산된다(자바는 2로 계산됨)
- # typeof : 변수의 타입 확인이 가능하다
- var integer = 100;
- console.log(type of integer);

2. 문자타입

: `"`, `'`, ``` 세가지 따옴표 타입 안에 작성이 가능함

템플릿 리터럴 : 백틱(```) 으로 작성하면 공백과 들여쓰기, 줄개행 다 가능함

: 나머지 두개 타입은 줄개행은 안됨 `'` & `"`

: 템플릿 리터럴은 ES6 이후 버전부터 가능하게 됨

3. 기타타입

#논리 타입 : `false`, `true` 라는 키워드 있음

: 논리표현 `0=false`, 0 이외의 모든 숫자는 `true` 로 반환

(C언어와 동일)

: 자바는 `boolean` 형으로 `true` or `false` 만 가능하지만 다른 언어들은 숫자로 표현함

#`null` 타입 : 값이 없다고 의도적으로 표현하고 싶을 때 `null`을 대입함

: 의미는 자바와 동일하나 `null`표기가 되면 어딘가 `null` 선언을 해둔거임

#`undefined` : 애초에 값이 매겨지지 않았을 때 표현

: 진짜로 값이 할당이 안된것임.

JS_연산자,조건문,반복문

HTML/CSS/JS

2022/11/10 14:58

<http://blog.naver.com/zhsks880/222925144186>

*브라우저입출력

prompt함수

- > alert 창 처럼 경고 창으로 입력 받는걸 띄움
- > 브라우저에서만 지원되는 함수
- > 리턴값이 String 이라 문자+문자 형식으로만 지정 됨
- > 숫자로 받고 싶을때는 앞에 + 기호 붙여야 함

```
var number1 = +prompt('첫번째 숫자를 입력하세요!');
```

*비교연산자

- > == 연산자 : 암묵적으로 문자열 내부에 숫자가 있다면 자동으로 숫자로 변환 후 비교
- > === 연산자 : 기존 자바처럼 타입까지 고려해서 비교해줌

```
var a= 5, b='5';
```

```
console.log(a == b); => 동등 비교 (값을 자동으로 변환하면서 계산해서 가끔 잘못된 결과값이 도출되는 경우도 생김)
```

```
console.log(a === b); => 일치 비교
```

- > 문자열 비교 : 영어의 대문자와 소문자를 철저히 비교해서 처리함, 문자는 아스키코드로 이루어 진거라서 당연히 값 이 틀리게 됨(자바와 동일)

```
console.log('apple' === 'APPLE'); => false
```

- > 문자열의 대소비교는 유니코드 문자번호 기준으로(자바에서는 안되는 기능) 영어대문자 < 영어소문자 < 한글 순서로 대 소 비교가 된다고 생각하면 됨

```
console.log('peach' < 'zebra'); => true
```

```
console.log('Zebra' > 'apple'); => false
```

*산술 연산자

- > 자바와 동일하지만 차이점으로는 나누기는 자동으로 소수점으로 반환해서 값을 줌

자바: $10 / 4 = 2$

자바스크립트: $10 / 4 = 2.5$

-> 자바에서 사용하는 단항 산술 연산자도 사용 가능하며 기능도 같음(++ , --)

-> 후위연산(자바와 동일)

```
var a = 10, result;  
result = a++;  
=> result=10 / a=11
```

->전위 연산자(자바와 동일)

```
var b = 10, result;  
result = ++b;  
-> result = 11 / b=11
```

***삼항연산자(자바와 동일)**

조건 ? 참이면 여기 : 거짓이면 여기;

```
var money = +prompt('액수를 입력하라');  
var food = money >= 5000 ? '김치찌개' : '라면';  
alert('메뉴는: ' + food);
```

***조건문(자바와 동일)**

-> if
-> else if
-> else
-> 자바와 동일하게 사용하며 형변환이 필요없어서 if 안에 조건 달 때 편함

***반복문(자바와 동일)**

-> while
-> for
: for(let n=1; n<=10; n++){
total += n
}

JS_타입캐스팅(명시적,암묵적변환,Truthy,Falsy)

HTML/CSS/JS

2022/11/11 12:11

<http://blog.naver.com/zhsks880/222925897287>

*명시적변환

String() : 문자열이 아닌 값을 문자열로 변환하는 방법

var a = 10, b = 20;

1번방법:

var result = a + String(b);

2번방법:

result = a.toString() + b.toString();

result = 1020이 나옴

Number() : 숫자가 아닌 값을 숫자타입으로 변환하는 방법

var m = '10', n = '3.14';

1번방법:

result = Number(m) + Number(n);

2번방법:

result = parseInt(m) + parseFloat(n);

result = 13.14 가 나옴

Boolean() : 논리 타입이 아닌 값을 논리 타입으로 변환하는 방법

var t = 'Hello', f = '';

console.log(Boolean(t));

결과값 true

console.log(Boolean(f));

결과값 false

*암묵적 변환

덧셈 연산만 숫자를 문자로 암묵적으로 변환해서 String + String 계산함

let ex = 10 + '20';

결과는 1020

나머지 연산자 등은 숫자로 변환해서 계산함 int - int / int * int 등...

```
let ex = 10 * '3';
```

결과는 30

```
let ex = 10 - '3';
```

결과는 7

```
let ex = 10 / 3;
```

결과는 3.33333333

문자열 타입의 암묵적 숫자변환 / prompt 함수 쓸때 처럼 + 를 붙여서 숫자로 변환

```
let ex2 = +'550';
```

결과는 숫자 550

console.log(typeof ex2); << typeof 를 사용하여 ex2 변수의 타입을 확인이 가능

논리타입을 암묵적으로 숫자 변환

```
let ex3 = +true;
```

결과는 1

```
let ex3 = +false;
```

결과는 0

*Truthy, Falsy

//undefined, null, 0, NaN, ''(빈 문자열) 은 모두 false

//값이 제대로 안들어오면 false 로 인식하여 ok2,5,8 만 출력이 됨

```
if('') console.log('OK1'); //false
```

```
if('hello') console.log('OK2');
```

```
if(undefined) console.log('OK3'); //false
```

```
if(null) console.log('OK4'); //false
```

```
if(350) console.log('OK5');
```

```
if(NaN) console.log('OK6'); //false
```

```
if(0) console.log('OK7'); //false
```

```
if([123]) console.log('OK8');
```

JS_배열

HTML/CSS/JS

2022/11/11 17:58

<http://blog.naver.com/zhsks880/222926203941>

*배열선언(자바와 똑같음)

```
: let fruits = ['딸기', '포도', '복숭아', '사과'];  
console.log(fruits);
```

*배열 길이 확인

```
: .length  
console.log(fruits.length);
```

*배열에 데이터 추가

```
: 자바랑 틀리게 그냥 인덱스 번호랑 값만 지정하면 들어감  
: fruits[4] = '파인애플';  
-> 이제는 ['딸기', '포도', '복숭아', '사과', '파인애플'];  
: fruits[fruits.length] = '망고';  
-> 이제는 ['딸기', '포도', '복숭아', '사과', '파인애플', '망고'];
```

*배열의 반복문

1번

```
for(var i=0; i<fruits.length; i++){  
fruits[i] };
```

2번

```
for(let ele of fruits){  
${ele} };
```

*다중배열

```
var ourStudentScores = [  
[67, 89, 92, 32],  
[100, 87, 55, 67],  
[23, 44, 100, 22]  
];
```

```
console.log(ourStudentScores);
```

-> 0, 1, 2 번째 배열들 모두 출력

```
console.log(ourStudentScores[1]);
```

-> 1번째 배열 값들 모두 출력

```
console.log(ourStudentScores[0][2]);
```

-> 0번째 배열의 2번째 인덱스값

indexOf()

: 배열 요소의 인덱스를 알려줌

: 배열내에 없는 값 전달 시 -1 리턴함

includes()

: 배열 요소의 존재 확인

slice()

: 배열의 부분추출 / 끝 범위는 해당 안됨

reverse()

: 배열을 역순으로 배치함

join()

: 배열의 요소들을 연결한 문자열로 반환

concat()

: 배열을 결

for-of 문

: 배열의 값을 얻어낼 때 사용하며 자바의 forEach 문과 유사함

: 자바의 항상 for문과 비슷함

for-in 문

: 배열의 인덱스값이나 key값을 변수에 저장한다

: 배열에서는 index 객체에서는 key 값을 추출

pop

: 배열의 마지막 요소를 제거하고 제거한 요소를 반환

push

: 배열의 끝에 요소를 추가

: 자바의 list.add 와 같음

shift

: 배열의 맨 첫번째 요소를 제거하고, 제거한 요소를 반환

unshift

: 배열의 맨 첫번째 위치에 요소를 추가

splice

: 배열의 특정 요소를 제거

: 인덱스 번호로 지정하고 갯수를 지정하는거라 slick 와는 성격이 다름

: 배열내의 삭제도 가능하고, 추가도 가능

JS_DOM

HTML/CSS/JS

2022/11/14 14:22

<http://blog.naver.com/zhsks880/222928352997>

-> 자바스크립트에서 돔이란 html 문서내에 있는 요소들 input 이건 title 이건 기타 등등 요소들을 지목해서 변수화 시킨 다음 에디트를 가능하게 해서 html 을 다시 만들거나 복제할 필요없이 사용하기 위함

-> 태그 선택을 했을때 객체(object)로 표현해서 해당 태그를 구성하는 부분(텍스트/CSS/html 속성)들을 제어할 수 있음

-> 태그들을 자바스크립트에서 제어할 수 있도록 처리하는 문법들의 모음이라고 생각하면 좋음

-> 변수는 해당 태그를 선택할때 사용하는 구문을 좀더 간결하게 쓰기위해서 해당 태그를 참조하는 역할이라고 보면 됨

-> 변수를 사용안해도 작업은 가능 (복제는 아님)

JS 이벤트 전파

HTML/CSS/JS

2022/11/15 12:42

<http://blog.naver.com/zhsks880/222929215914>

전파방해

JS BOM

HTML/CSS/JS

2022/11/16 18:01

<http://blog.naver.com/zhsks880/222930446043>

돔 친구 봄

JS $\frac{\circ}{\circ} \frac{\circ}{\circ}$

HTML/CSS/JS

2022/11/18 13:57

<http://blog.naver.com/zhsks880/222931998353>

0 L 2 0 L 2 0 L 2 0 L

JQuery

HTML/CSS/JS

2022/11/29 14:57

<http://blog.naver.com/zhsks880/222941323579>

- > 자바스크립트를 조금 더 편하게 사용하게 해주는 라이브러리
- > 자바스크립트에 비해 생산성이 높다
- > 태그를 선택자로 한번에 선택할 수 있고 제어할수도 있음

*설치

->1번방법

jquery 공홈 -> 다운로드 -> [Download the compressed, production jQuery 3.6.1](#) 클릭 -> 코드라서 우클릭 하고 다른 이름으로 저장으로 하면 js파일로 저장 됨 (학원에서는 Vscode 에 파일 다운로드 받아서 사용함)

->2번방법

Using jQuery with a CDN -> 링크로 라이브러리 땡겨 오는거 됨(인터넷이 되야함) -> <https://releases.jquery.com/> ->

`<script src="https://code.jquery.com/jquery-3.6.1.js"`

`integrity="sha256-3zIB5s2uwoUzrXK3BT7AX3FyvojsraNFxCc2vC/7pNI=" crossorigin="anonymous"></script>`

이런식으로 코드 넣어서 사용해도 됨

*jQuery 시작 시

-> `$(function(){})`

*Js 로 코드 작성한거 그대로 구글에 복붙하고 jquery 로 검색하면 변경하는거 그대로 알려줌

-> `document.getElementById('temp');`는 HTML DOM 객체를 리턴하고,

`$('#temp');` 는 jQuery 객체를 리턴한다.

따라서, HTML DOM 객체를 사용해야 하는 경우에는,

`$('#temp')[0]`로 jQuery 객체의 첫 번째 element를 불러와야 HTML DOM 객체로 활용할 수 있다.

JQ_정규표현식

HTML/CSS/JS

2022/12/07 11:48

<http://blog.naver.com/zhsks880/222948551962>

* ID 체크 (영어 대문자, 소문자, 숫자 섞어서 4~14글자 까지)

```
: const getIdCheck = RegExp(/^[a-zA-Z0-9]{4,14}$/);
```

* PASS 체크(영어 대문자, 소문자, 특수문자 섞어서 글자제한 없음)

```
: const getPwCheck = RegExp(/[a-zA-Z0-9].*([!@#$%^&*?_~])|([!@#$%^&*?_~].*[a-zA-Z0-9])/);
```

* 이름 체크(한글만 가능)

```
: const getNameCheck = RegExp(/^[가-힣]+$/);
```

-> 정규표현식변수.test('검증하고 싶은 값') -> return boolean type

-> 정규표현식에 어긋난 값이면 false, 올바른 값이면 true.

JQ_ajax 호출

HTML/CSS/JS

2022/12/07 12:42

<http://blog.naver.com/zhsks880/222948592674>

ajax 호출

: 클라이언트에서 서버와 비동기 통신을 진행하게 도와주는 ajax 함수

: 각 방식에 따른 ", " 를 잘 찍어주자

: ajax 에서 json 타입으로 넘겨 줄때는 값을 가져온 변수를 담아서 객체화 해서 보내주는게 편함

*체크하고 값들을 변수 및 객체화 시작

```
if(chk1 && chk2){
```

```
const id = $('#signInId').val();
```

```
const pw = $('#signInPw').val();
```

```
const login = {
```

```
'account' : id,
```

```
'password' : pw
```

```
};
```

```
$.ajax({
```

```
type: 'POST', =>서버에 전송하는 HTTP 방식 (post or get 방식 처리할 거 정하기)
```

```
url: '/user/checkId', =>서버에 요청할 url 작성
```

```
headers: {
```

```
'Content-type': 'application/json'
```

```
}, =>요청 헤더 정보(전송하고자 하는 데이터의 타입)
```

```
dataType: 'text', =>서버로부터 응답 받을 데이터의 형태
```

```
data: id, =>서버로 전송할 데이터(현재는 id만 / 일반 텍스트라 그냥 던져줌)
```

```
success: function(result){ =>통신 성공 시 function의 매개변수로 저장됨
```

```
=> function의 매개값은 Controller 에서 return 되는 string 값을 받아와서 밑에 있는 if 문에 조건을 할때 사용하게 됨
```

```
console.log('통신성공!: ' + result); =>서버와 통신 성공 시 실행할 내용
```

```
if(result === 'available'){
```

```
$('#user_id').css('background', 'aqua');
```

```
$('#idChk').html('<b style="font-size: 14px; color: green;">[아이디사용이 가능합니다]</b>');
```

```
} else {
```

```

$('#user_id').css('background', 'pink');
$('#idChk').html('<b style="font-size: 14px; color: red;">[아이디가 중복되었습니다]</b>');
}
},
error: function(status, error){
console.log('통신실패');
console.log(status, error);

}

```

* 사용자의 값이 여러개를 객체화 시켜서 json 으로 만들어서 던지는거

```

$('#signIn-btn').click(function() {

if(chk1 && chk2){
const id = $('#signInId').val();
const pw = $('#signInPw').val();

const login = {
'account' : id,
'password' : pw
};

//ajax 시작
$.ajax({
type: 'post',
url: '/user/loginCheck',
contentType: 'application/json',
dataType: 'text',
data: JSON.stringify(login),
success: function(data){
if(data === 'idFail'){
//console.log('아이디가 없다');
$('#signInId').css('background', 'pink');
$('#idCheck').html('<b style="font-size: 14px; color: red;">[존재하지 않는 아이디입니다.]</b>');
$('#signInId').val('');
$('#signInId').focus(); // # focus : 커서 이동 및 스크롤 해당 위치로 이동

```

```

chk1 = false, chk2 = false;
} else if(data === 'pwFail'){
//console.log('비번 틀림');
$('#signInPw').css('background', 'pink');
$('#pwCheck').html('<b style="font-size: 14px; color: red;">[비밀번호가 틀렸습니다.]</b>');
$('#signInPw').val('');
$('#signInPw').focus();
chk2 = false;
} else{
//console.log('로그인성공');
alert('로그인을 환영합니다');
location.href='/';
}

},
error: function(){
console.log('통신실패:');
}
}) //ajax 끝

} else {
alert('입력값을 다시 확인하세요');
}
}); // 로그인 버튼 이벤트 처리 끝

```

JS_KeyCode 정리

HTML/CSS/JS

2022/12/07 17:36

<http://blog.naver.com/zhsks880/222948967834>

Key Name	Key Code	Event Key	Event Code
backspace	8	Backspace	Backspace
tab	9	Tab	Tab
enter	13	Enter	Enter
shift(left)	16	Shift	ShiftLeft
shift(right)	16	Shift	ShiftRight
ctrl(left)	17	Control	ControlLeft
ctrl(right)	17	Control	ControlRight
alt(left)	18	Alt	AltLeft
alt(right)	18	Alt	AltRight
pause/break	19	Pause	Pause
caps lock	20	CapsLock	CapsLock
escape	27	Escape	Escape
space	32		Space
page up	33	PageUp	PageUp
page down	34	PageDown	PageDown
end	35	End	End
home	36	Home	Home
left arrow	37	ArrowLeft	ArrowLeft
up arrow	38	ArrowUp	ArrowUp
right arrow	39	ArrowRight	ArrowRight
down arrow	40	ArrowDown	ArrowDown
print screen	44	PrintScreen	PrintScreen
insert	45	Insert	Insert
delete	46	Delete	Delete
0	48	0	Digit0
1	49	1	Digit1
2	50	2	Digit2

3	51	3	Digit3
4	52	4	Digit4
5	53	5	Digit5
6	54	6	Digit6
7	55	7	Digit7
8	56	8	Digit8
9	57	9	Digit9
a	65	a	KeyA
b	66	b	KeyB
c	67	c	KeyC
d	68	d	KeyD
e	69	e	KeyE
f	70	f	KeyF
g	71	g	KeyG
h	72	h	KeyH
i	73	i	KeyI
j	74	j	KeyJ
k	75	k	KeyK
l	76	l	KeyL
m	77	m	KeyM
n	78	n	KeyN
o	79	o	KeyO
p	80	p	KeyP
q	81	q	KeyQ
r	82	r	KeyR
s	83	s	KeyS
t	84	t	KeyT
u	85	u	KeyU
v	86	v	KeyV
w	87	w	KeyW
x	88	x	KeyX
y	89	y	KeyY
z	90	z	KeyZ
left window key	91	Meta	MetaLeft

right window key	92	Meta	MetaLeft
select key	93	ContextMenu	ContextMenu
numpad 0	96	0	Numpad0
numpad 1	97	1	Numpad1
numpad 2	98	2	Numpad2
numpad 3	99	3	Numpad3
numpad 4	100	4	Numpad4
numpad 5	101	5	Numpad5
numpad 6	102	6	Numpad6
numpad 7	103	7	Numpad7
numpad 8	104	8	Numpad8
numpad 9	105	9	Numpad9
multiply	106	*	NumpadMultiply
add	107	+	NumpadAdd
subtract	109	-	NumpadSubtract
decimal point	110	.	NumpadDecimal
divide	111	/	NumpadDivide
f1	112	F1	F1
f2	113	F2	F2
f3	114	F3	F3
f4	115	F4	F4
f5	116	F5	F5
f6	117	F6	F6
f7	118	F7	F7
f8	119	F8	F8
f9	120	F9	F9
f10	121	F10	F10
f11	122	F11	F11
f12	123	F12	F12
num lock	144	NumLock	NumLock
scroll lock	145	ScrollLock	ScrollLock
audio volume mute	173	AudioVolumeMute	
audio volume down	174	AudioVolumeDown	
media player	175	AudioVolumeUp	

audio volume up	181	LaunchMediaPlayer	
launch application 1	182	LaunchApplication1	
launch application 2	183	LaunchApplication2	
semi-colon	186	;	Semicolon
equal sign	187	=	Equal
comma	188	,	Comma
dash	189	-	Minus
period	190	.	Period
forward slash	191	/	Slash
Backquote	192	`	Backquote
open bracket	219	[BracketLeft
back slash	220	₩	Backslash
close bracket	221]	BracketRight
single quote	222	'	Quote

promise 함수(콜백지옥탈출)

HTML/CSS/JS

2022/12/21 10:54

<http://blog.naver.com/zhsks880/222961592667>

*Deferred: JQuery에서 제공하는 promise 함수

: 먼저 실행 되어야 할 함수 안에 제이쿼리 형식으로 변수로 선언을 한다

: 콜백 방식을 사용하는 비동기 처리 모델들은 순서가 보장되지 않는다는 문제가 있음

: 비동기 방식처리는 요청에 대한 처리가 병렬방식이라(콜백지옥) 함수의 처리 순서를 직렬방식으로 보장하기 위해 사용

```
let deferred = $.Deferred();
```

->그리고 나서 ajax 안에서 사용

```
if(userId !== ''){  
$.ajax({  
type: 'post',  
url: '<c:url value="/snsBoard/listLike" />',  
data: userId,  
contentType: 'application/json',  
success: function(result){  
console.log('result: ' + result);  
deferred.resolve(result);  
},  
error: function(){  
console.log('관리자에게 문의하세요');  
//처리에 실패 시 rejected 함수 호출  
deferred.rejected();  
}  
}); // end ajax
```

-> 그 다음에 // # promise 방식 : .done <실행될 함수, .fail < 실행에 실패 된 함수

```
getListLike().done(function(result){  
getList(1, true, result);
```

```
})  
.fail(function(error){  
  console.log(error);  
});
```

이렇게 처리해야 콜백 지옥에서 빠져 나올수 있다.

구글맵 api 키

HTML/CSS/JS

2022/12/23 14:04

<http://blog.naver.com/zhsks880/222963518894>

AlzaSyCJI_MuKSisirgqN8k27PwIEUYt1v9F2b4

Git 설정

HTML/CSS/JS

2022/12/26 15:12

<http://blog.naver.com/zhsks880/222966103997>

*형상관리 프로그램의 일종

*깃과 깃허브는 다름

*깃으로 프로그램 소스코드 관리

*깃허브는 프로젝트를 같이 공유 하는것

*깃

: 로컬저장소를 관리한다

<https://git-scm.com/>

Git

About The advantages of Git compared to other source control systems. Documentation Command reference pages, Pro Git book content, videos and other material. Downloads GUI clients and binary releases for all major platforms. Community Get involved! Bug reporting, mailing list, chat, development and ...
git-scm.com

: 깃 메인 홈페이지 접속-> download for windows gogo-> git 을 설치해 준다

: git gui 는 어느정도 인터페이스가 있는것

: git bash 는 리눅스 기반 명령어 치는 것

: 프로젝트 폴더내에서 우클릭 git bash 클릭-> 시작

*깃허브 안에는 3가지 공간이 있음

1. working directory -> 현재 쓰고 있는 폴더
2. staging area -> 코드 변경 사항들이 올라가는 공간
3. Repository -> 깃허브에 저장되는 곳 (1+2)

* clear -> 내용 다 지우기

1. git config --global user.email "zhsks880@naver.com"

이 폴더를 이 유저 계정에 등록한다

2. \$ git config --global user.name "전 상 현"

이름 설정

3. \$ git config --list

설정된 계정 정보가 뜸

4. \$ git init

: 로컬저장소 생성

: 보기->숨김 하면 .git 폴더 숨어서 생성 됨

5. \$ git status

: 해당 창에 대한 정보 표현

: 수정한 사항이 있으면 빨강게 보임

```
ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   a.txt

no changes added to commit (use "git add" and/or "git commit -a")

ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$
```

6. \$ git add a.txt

: staging area 로 보냄 -> 이과정을 스테이징 한다~라고 함(대기상태임)

: 대기상태에서 파일의 변경이 일어나면 다시 대기 상태로 돌리고 커밋해야함

: git rm --cached 파일명 : 해당 파일을 staging area 에서 삭제

: git add . : 몽땅 저장(점은 한칸 띄고 써라)

7. \$ git commit -m "처음커밋"

: 커밋하기 깃허브로 송송

8. \$ git log --all --oneline

: git log 만 쳐봐도 됨

: --all --oneline 은 에디션임
: 커밋 이후 변경 로그를 확인

9. \$ git difftool 821ce1e b6fd3a3
: git 변경 사항을 확인하는 툴인데 불편함....vscode로 보는게 나음
: 나오는건 :qa 명령어 입력 해야함

```
ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ git log --all --oneline
821ce1e (HEAD -> master) 두 번째 커밋입니다
b6fd3a3 처음 커밋

ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ git diff 821ce1e

ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ git diff 921ce1e b6fd3a3
fatal: ambiguous argument '921ce1e': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ git diff 821ce1e b6fd3a3
diff --git a/a.txt b/a.txt
index 6a6efb3..daf4d01 100644
--- a/a.txt
+++ b/a.txt
@@ -1,2 +1 @@
-안녕하세요 오늘은 2022년 12월 26일입니다
-한 줄 추가합니다 ~~~
\ No newline at end of file
+안녕하세요 오늘은 2022년 12월 26일입니다
\ No newline at end of file

ict03-10@DESKTOP-UAAHEJI MINGW64 ~/Desktop/test_git (master)
$ |
```

* branch 작업
: master 라는 공간안에다가 커밋을 계속 하다가 보면 진행되던 시점으로 돌아가기가 힘들게 되는 단점을 보완하기 위해서 브랜치 작업을 하기로 함
: 예로 write 브랜치 & 회원 브랜치 & 프로젝트 메인으로 나누어 가다가 나중에 통합하기
: 브랜치를 나누기 전에는 최소한 한번이라도 commit 해야함(프로젝트 초기세팅)

1. \$ git branch newfunc
: 새로운 브랜치 newfunc 하나를 만들

2. \$ git switch newfunc

: 새로 만든 브랜치로 이동

3. git add . 해서 새로 만들 파일을 새로운 브랜치에다가 애드 및 커밋

4. \$ git log --oneline --all --graph

: 갈라진 브랜치까지 표현해주는 명령어

5. \$ git merge newfunc

: 머지하는 기본 방법(3가지 방법이 있음)

: 단점이 한번에 다 하는거다 보니까 커밋내용이 다 출력되서 굉장히 보기가 힘들게 됨

: master 로 이동하고나서 newfunc 브랜치를 덮을 때 사용함

: 파일명이 같을때 내용이 서로 다르면 변경이 제대로 안됨-> 파일 안에 열어보면 Git 이 Head 와 newfunc 부분을 가르쳐 주는 부분을 내가 선택하고 그리고 나서 다시 git add -> commit 해야함

: 브랜치 작업 할 때는 파일명을 겹치게 하지 않게 하는게 속편함

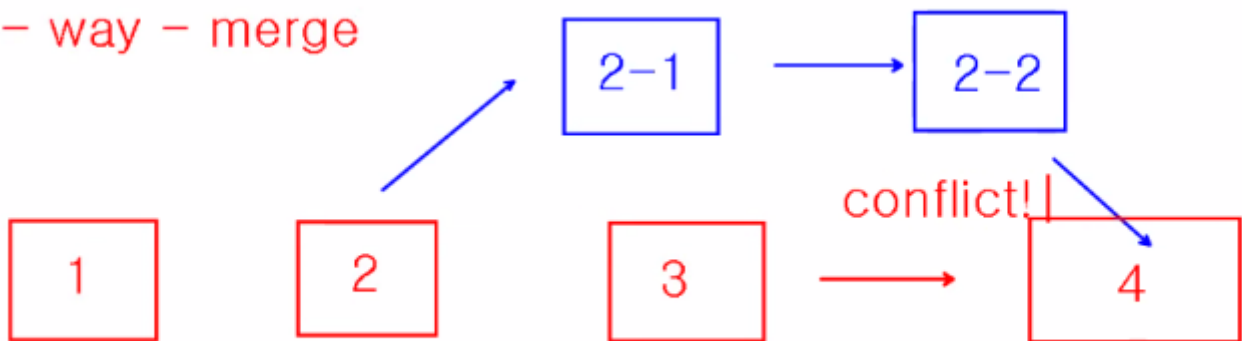
*머지의 종류 및 방법

* 3 way merge :

명령어 : git switch master

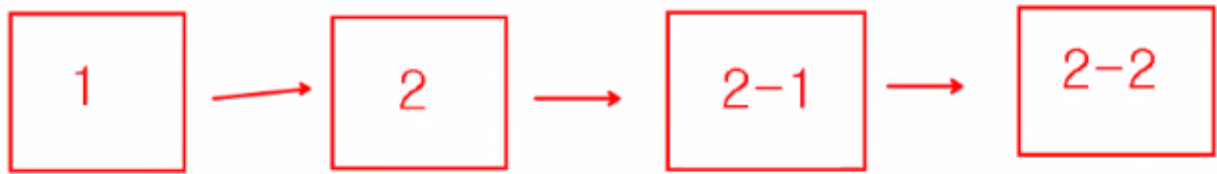
명령어 : git merge newfunc

3 - way - merge



* fast forward merge :

fast-forward merge

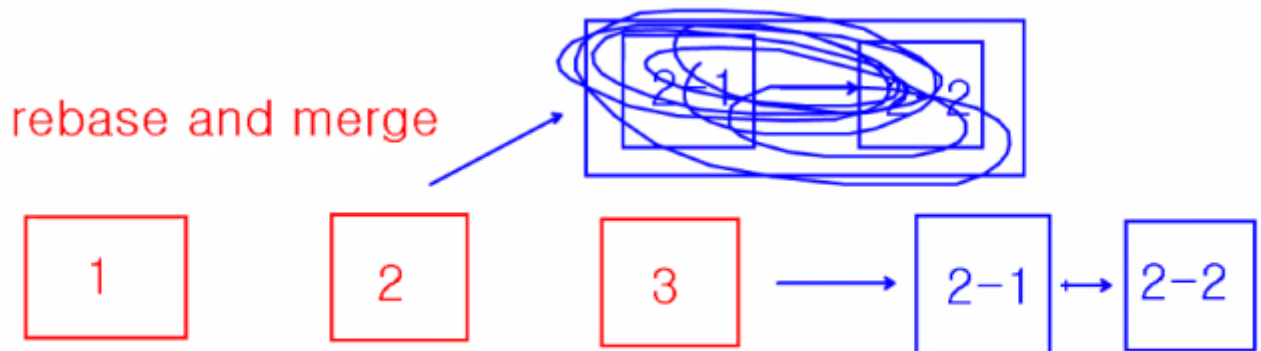


+

* rebase and merge :

명령어 : git switch master

명령어 : git rebase newfunc



* squash and merge : 많이 사용함(그래프가 굉장히 깔끔해짐 / master 위주)

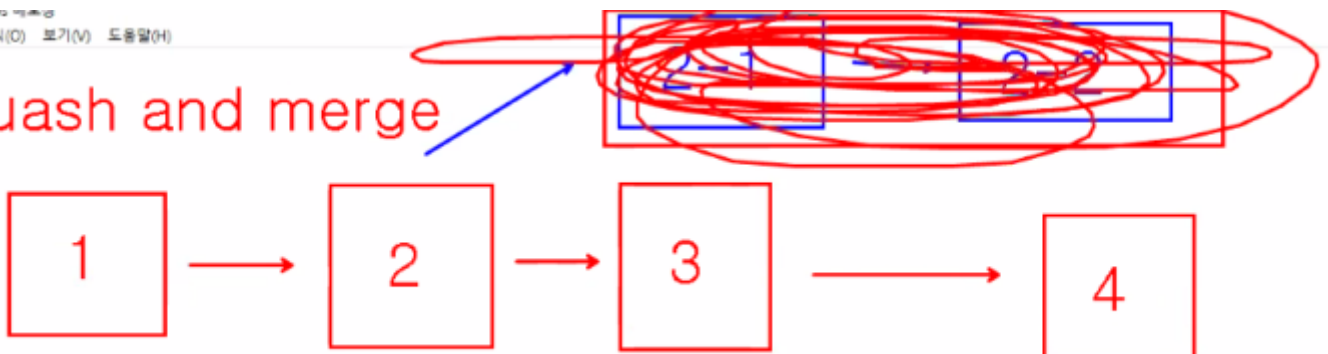
명령어 : git switch master

명령어 : git merge --squash newfunc

4명 · 118184/102 · 10/10/2020

편집(E) 서식(O) 보기(V) 도움말(H)

squash and merge



*이전으로 돌아가는 방법(게임에서 checkPoint 를 커밋이라 생각해라)

```
$ git restore --source 75b2f48 b.txt
```

: source 다음에 75b2 커밋명 + b.txt 는 파일 명

: source 안적고 파일명만 적으면 최근 파일이 변경된 시점으로 돌아감

* \$ git commit -am "한번에 커밋" : add 와 동시에 커밋 메세지 작성까지 한번에 됨 개꿀

* 이전 커밋 삭제

```
$ git revert 235b870
```

: 커밋생성 아이디 적어주면 a.txt 만든거 자체를 없애버림

: 깃 로그 찍어보면 살아는 있음

* 이전 커밋으로 복원

```
$ git reset --hard 75b2f48
```

: reset 시점 이후를 아예 못찾게 됨(나름 최후의 수단)

Github 설정

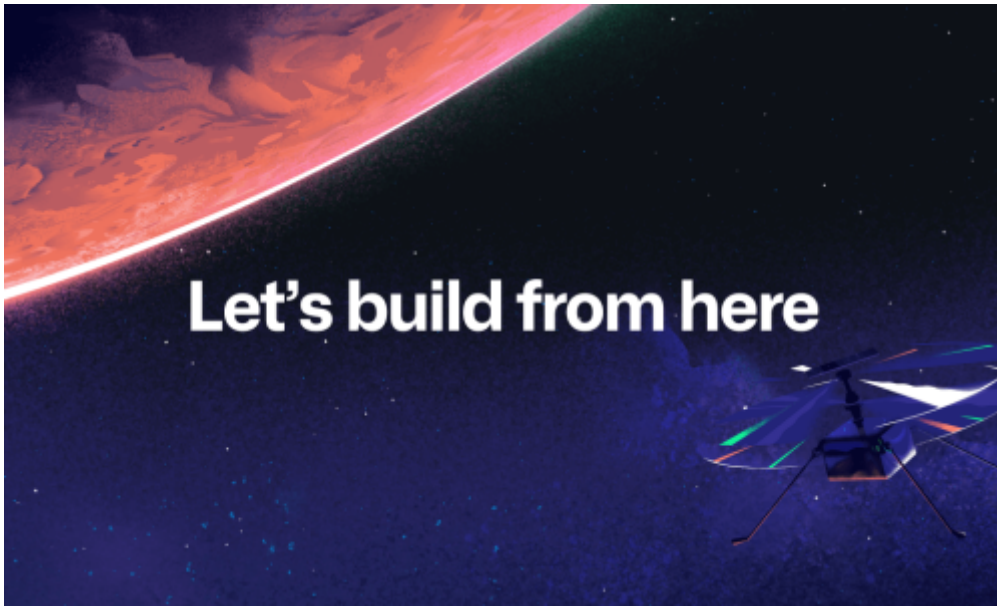
HTML/CSS/JS

2022/12/27 12:08

<http://blog.naver.com/zhsks880/222967009803>

*깃허브로 가기

<https://github.com/>



GitHub: Let's build from here

GitHub is where over 94 million developers shape the future of software, together. Contribute to the open source community, manage your Git repositories, review code like a pro, track bugs and feat...

github.com

1. repository 생성
2. Add a README file 나중에 만들어도 무방함

.gitignore : 스프링에서 사용한 properties 같은것들이나 기타 STS설정 파일등 은 제외하고 올릴때 사용

: gitignore 완성 시켜주는 사이트 따로 있음 / <https://www.toptal.com/developers/gitignore/>



Create useful .gitignore files for your project

Create

[Source Code](#)

[Command Line Docs](#)

[Watch Video Tutorial](#)



[gitignore.io](https://www.gitignore.io)

Create useful .gitignore files for your project

www.toptal.com

: .gitignore 아래에 설명 다시 기재함

Created by <https://www.toptal.com/developers/gitignore/api/eclipse>

Edit at <https://www.toptal.com/developers/gitignore?templates=eclipse>

Eclipse

.metadata

bin/

tmp/

*.tmp

*.bak

*.swp

*~.nib

local.properties

.settings/

.loadpath

.recommenders

External tool builders

.externalToolBuilders/

Locally stored "Eclipse launch configurations"

*.launch

PyDev specific (Python IDE for Eclipse)

*.pydevproject

CDT-specific (C/C++ Development Tooling)

.cproject

CDT- autotools

.autotools

Java annotation processor (APT)

.factorypath

PDT-specific (PHP Development Tools)

.buildpath

sbteclipse plugin

.target

Tern plugin

.tern-project

TeXlipse plugin

.texlipse

STS (Spring Tool Suite)

.springBeans

Code Recommenders

.recommenders/

Annotation Processing

.apt_generated/

.apt_generated_test/

Scala IDE specific (Scala & Java development for Eclipse)

.cache-main

.scala_dependencies

.worksheet

Uncomment this line if you wish to ignore the project description file.

Typically, this file would be tracked if it contains build/dependency configurations:

#.project

Eclipse Patch

Spring Boot Tooling

.sts4-cache/

End of <https://www.toptal.com/developers/gitignore/api/eclipse>

내 주소 복사해서

<https://github.com/zhsks880/testwork.git>



[zhsks880/testwork](https://github.com/zhsks880/testwork)

학원 test 용. Contribute to zhsks880/testwork development by creating an account on GitHub.
github.com

어제 만들어 둔 merong 폴더(깃 로컬폴더) 안에서 git bash 를 뜬다

1. `$ git push -u https://github.com/zhsks880/testwork.git master`

: -u 하면 저 저장소로 다음을 기억해서 사용함 : 다음부터 주소 안쳐도 되는것

: 맨뒤에 master 라고 붙이면 마스터로 올린다



*로컬 폴더에서 변경이 생기면

Git bash 를 해서 어제 한 작업처리한다

`git add .`

`git commit -m ""`

`git push` (위에서 -u 해놔서 됨)

`$ git remote add origin https://github.com/zhsks880/testwork.git`

: 깃 저장소에 origin 이라는 이름을 붙히고 내 주소를 등록해놓음

`$ git push origin master`

: 그 다음부터는 이렇게 올리면 된다

: 내꺼 레파지토리만 푸쉬가 가능함(남은 못함)

*레파지토리에서 가져오기

복사해 오고 싶은 레파지토리 이동해서 Code 누르면 해당 레파지토리 주소가 나옴

Zip파일로 다운도 가능함

: 새폴더에 가서 `$ git init -> $ git clone https://github.com/zhsks880/testwork.git ->` 내려 받게 됨

: A -> 레파지토리 -> B 형식에서 B에서 뭔가 파일을 바꿔서 push 를 하면 A가 push를 못함

: 그래서 에러 메시지를 보면 레파지토리->A 로 pull 한 후에 push 해줘야함(통일화)

: `$ git pull origin master` 다음에

: `$ git push origin master` 다시 올리기

: 그래서 대부분 pull 부터 하고 push 를 진행한다.

*동업자 등록하기(Collaborator)

-> settings 가기 -> 좌측에 Collaborators 가기 -> 초대하고 싶은 사람 아이디 및 이메일로 해서 콜라보 등록하기 -> 끄을

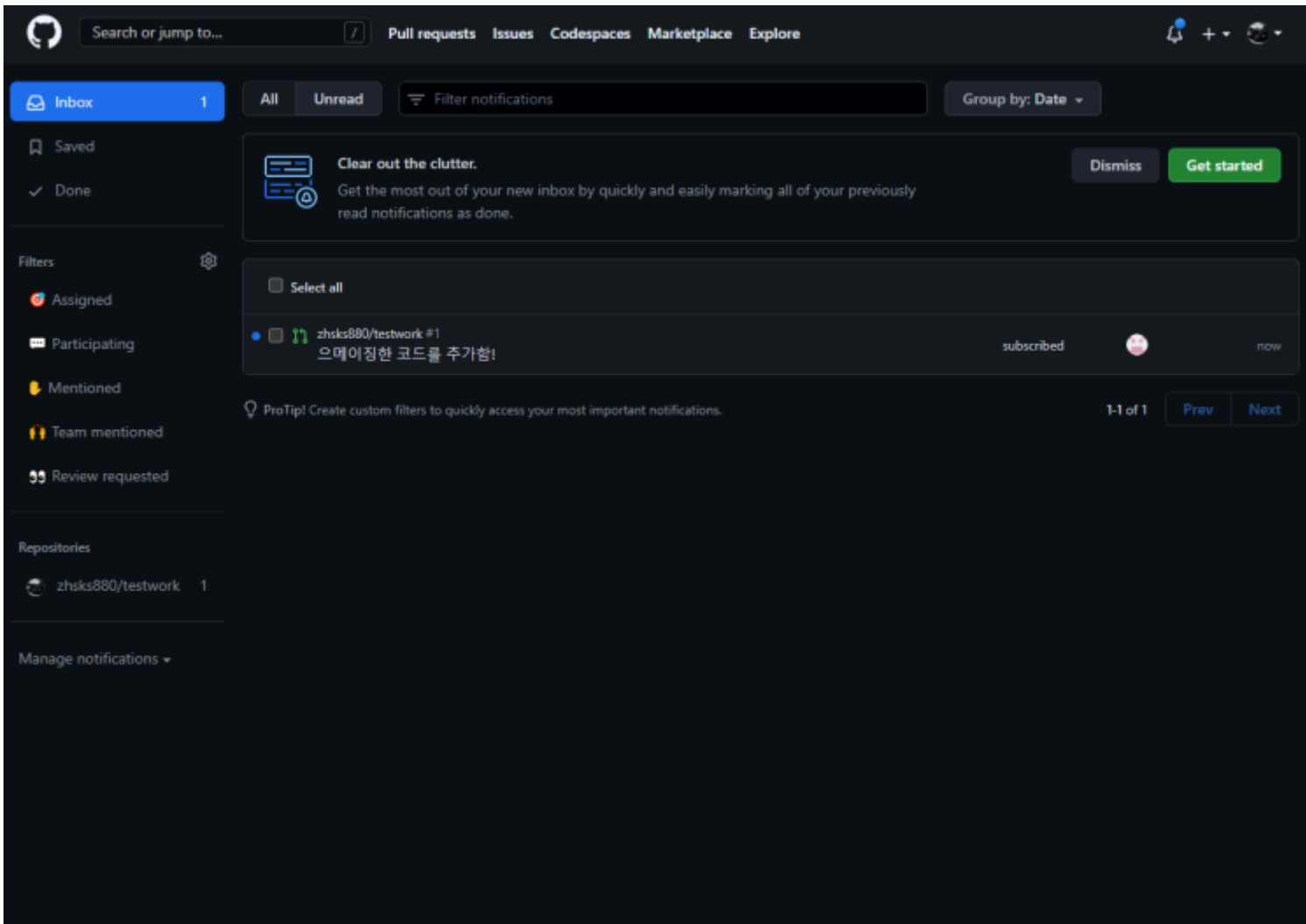
~

*에러 날 경우

- : git pull 동업자주소 --allow-unrelated histories
- : 어거지로 끌고 올 경우 사용
- : 내가 최신인지 동료의 최신인지 깃허브에서 판단이 안되어서 오류가 나올 시 끌고와서 확인해본다
- : conflict 난거를 확인해서 최신화 하고 싶은거를 남겨두고 다시 파일을 수정한다
- : 그리고 나서 다시 add -> commit 해줘서 레파지토리를 최신화 한다

*동업자 등록 안하고 협업 하는 방법(Folk 기능)

- : 남의 레파지토리 들어가서 우측 위에 Fork 기능사용
- : 내가 Fork 해온 것을 내가 마음대로 할수 있고, 그리고 나서 원작자에게 permission 을 얻는다
- : 내꺼 레파지토리로 푸쉬까지 끝내고나서 pull request 눌러서 요청 한다
- : 요청자의 코드를 확인하고 pull request 로 하면 합쳐서 가능하다



zhsk880 / testworkPublic

PinUnwatch1ForkStar0

<> CodeIssuesPull requests1ActionsProjectsWikiSecurityInsightsSettings

으메이징한 코드를 추가함! #1

Openleekyoungmin321 wants to merge 1 commit into zhsk880:master from leekyoungmin321:master

Conversation0Commits1Checks0Files changed1+1-0

leekyoungmin321 commented 2 minutes agoFirst-time contributor

제발 제 코드를 허가에 주세요
안그러면 너무 슬플 것 같아요~~~

으메이징한 코드를 추가함!Ba24948

Add more commits by pushing to the master branch on leekyoungmin321/testwork-1.

Require approval from specific reviewers before merging
Branch protection rules ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull requestYou can also open this in GitHub Desktop or view command line instructions.

ReviewersNo reviewsStill in progress? Convert to draft

AssigneesNo one—assign yourself

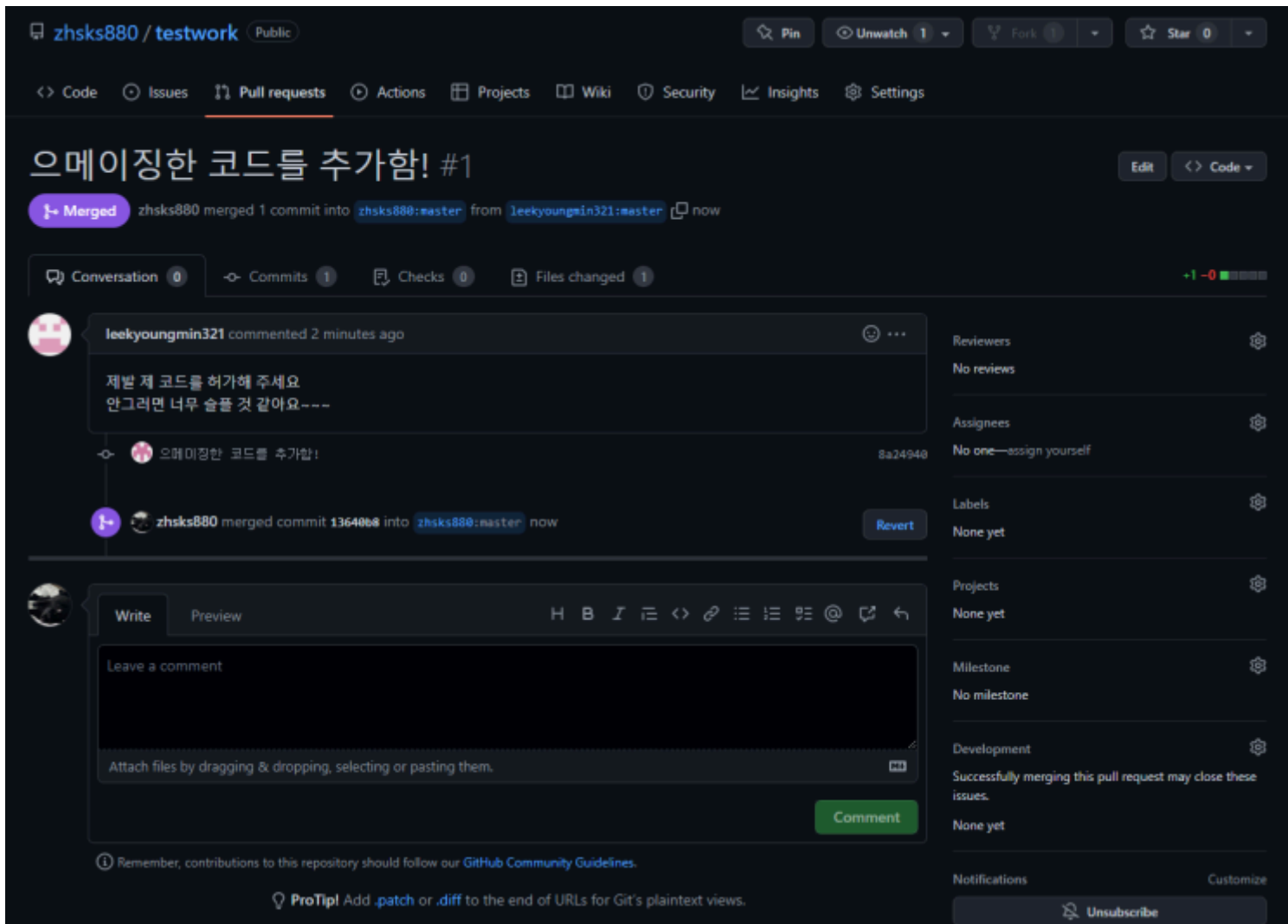
LabelsNone yet

ProjectsNone yet

MilestoneNo milestone

DevelopmentSuccessfully merging this pull request may close these issues.

Notifications



*vscode 로 사용하는 방법

- : 좌측에 소스제어 클릭
- : 마켓가서 git lens / gir graph 찾아서 설치하기
- : 새터미널 열고 터미널에서 바로 git 관련 명령어 사용해도 됨

*gitignore 설정

- : git bash 에서 커맨드
- : touch .gitignore (이그노어 파일 생성)
- : <https://www.toptal.com/developers/gitignore>



Create useful .gitignore files for your project

Create

[Source Code](#)

[Command Line Docs](#)

[Watch Video Tutorial](#)



gitignore.io

Create useful .gitignore files for your project

www.toptal.com

사이트에 가서 해당 IDE 입력해서 깃에 추가 시 제외시킬 파일등을 확인해서 코드 복사

: .gitignore 파일을 텍스트 or 워드패드 등으로 열어서 위의 코드 복붙하기

: 기존에 .gitignore 를 잘못 실행한 경우에는

1) git rm: 로컬과 원격 저장소 모두에서 파일 혹은 경로 삭제

2) git rm --cached: 원격 저장소에서만 파일 혹은 경로 삭제

git rm -r --cached . (한칸띄고 점 까지) 입력

: git add .

: git commit -m "ignore추가 및 생성완료"