

# 알튜비튜2

## OT



안녕하세요! 알튜비튜 2기의 본격적인 시작에 앞서 여러 안내를 위해 제작한 OT 자료입니다.  
PS(Problem Solving)를 위한 기본적인 개념을 배우고, 알튜비튜에서 활용할 다양한 플랫폼  
사용법과 우리의 지켜야 할 규칙들을 알려드릴게요.

# 01

## 알튜비튜2 소개



## 김서연

컴퓨터 공학  
주 업무 : 코드리뷰, 질의응답  
사용 언어: c++  
Github : flowersayo  
Baekjoon : flowersayo0811



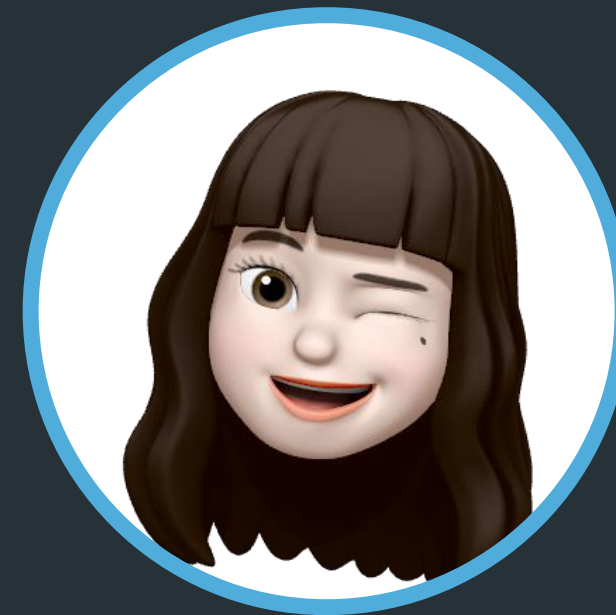
## 김재서

컴퓨터 공학  
주 업무 : 코드리뷰, 질의응답  
사용 언어: c++  
Github : jaeseo222  
Baekjoon : jaeserrr



## 민경원

컴퓨터 공학  
주 업무 : 코드리뷰, 질의응답  
사용 언어: c++  
Github : mkkwkw  
Baekjoon : mkkw0228



## 박주은

컴퓨터 공학  
주 업무 : 코드리뷰, 질의응답  
사용 언어: c++  
Github : hoit1302  
Baekjoon : hoit1302



## 박지영

컴퓨터 공학  
주 업무 : 튜터링 수업  
사용 언어: c++, python3  
Github : ZERO-black  
Baekjoon : pjy0616



## 박한울

사이버 보안  
주 업무 : 코드리뷰, 질의응답  
사용 언어: python3  
Github : nanometre380  
Baekjoon : 380380380



## 배수아

컴퓨터 공학  
주 업무 : 튜터링 수업  
사용 언어: c++  
Github : bsa0322  
Baekjoon : bsa0322



## 장예서

컴퓨터 공학  
주 업무 : 코드리뷰, 질의응답  
사용 언어: python3  
Github : yesyeseo  
Baekjoon : redemption

## 커리큘럼의 분리

- 우선 초급까지는 꼭 저희와 함께 가는거예요..!
- 브루트 포스가 끝난 후, 1차 수료 기간(aka 철회 기간)이 있을 예정...😓
- 초급 커리큘럼이 끝난 후, 중급을 계속 들으지 말지 선택!
- 연속 수강 선택자가 적을 경우, 중급 커리큘럼 때 재모집할 가능성도 있음

## 튜티 사이의 그룹 형성!

- 1차 수료 기간이 끝난 후, 그룹을 형성해드릴 예정입니다!
- 그룹 간, 서로 과제 풀이도 공유하고 질문 하는 등 자유롭게 활용 가능
- 그룹 별 과제를 많이 풀수록 소소한 상품이...?!

## 실시간 ZOOM 수업

- 실시간 ZOOM 수업
- 수업 전, Notice 레포의 리드미를 통해 라이브 코딩 문제와 과제 문제를 미리 공개 (여기 과제 문제 **힌트**도 있어요!)
- 모든 **라이브 코딩**과 **샘플 코드** 및 **코드리뷰 가이드**는 Notice 레포에 제공!

# 02

## C vs C++



## Hello World!

```
#include <stdio.h>

int main() {
    printf("Hello World!\n");
}
```

C

```
#include <iostream>

using namespace std;

int main() {
    cout << "Hello World!" << endl;
}
```

C++

## 입출력

```
#include <stdio.h>

int main() {
    int a;
    scanf("%d", &a);
    printf("int a = %d\n", a);
}
```

C

```
#include <iostream>

using namespace std;

int main() {
    int a;
    cin >> a;
    cout << "int a = " << a << endl;
}
```

C++

## Boolean

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool is_boolean = true;
}
```

C

```
#include <iostream>

using namespace std;

int main() {
    bool is_boolean = true;
}
```

C++

## String

```
#include <stdio.h>

int main() {
    char str[10] = "Hello";
    printf("%s\n", str);
}
```

C

```
#include <iostream>

using namespace std;

int main() {
    string str = "Hello";
    cout << str << endl;
}
```

C++

## Struct

```
#include <stdio.h>

typedef struct info {
    char name[20];
    int age, height, weight;
} info;

int main() {
    info i1;
}
```

C

```
#include <iostream>

using namespace std;

struct info {
    string name;
    int age, height, weight;
}

int main() {
    info i1;
}
```

C++

## 동적할당

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int size = 5;
    int *arr = (int*)malloc(sizeof(int) * size);
}
```

C

```
#include <iostream>

using namespace std;

int main() {
    int size = 5;
    int *arr = new int[size];
}
```

C++

## STL(Standard Template Library)

- 표준 C++ 라이브러리
- PS에 자주 사용되는 자료구조와 알고리즘 제공  
→ vector, stack, queue, sort, etc.

## Time Complexity

- 알고리즘의 성능을 분석하는 방법 중 하나
- 일반적으로 최악의 경우를 가정하고 계산
- Big-O Notation 으로 주로 표기
- 시간 복잡도를 고려하지 않고 작성한 코드는 시간초과를 유발할 수 있음



```
int a = 1;
```

$O(1)$

```
int a = 0;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        a += (i + j);
    }
}
```

$O(n^2)$

```
int a = 0;
for (int i = 0; i < N; i++) {
    a += i;
}
```

$O(n)$

```
int a = 0;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        for (int k = 0; k < N; k++) {
            a += (i + j + k);
        }
    }
}
```

$O(n^3)$

```
int a = 0;
while (N > 0) {
    a += N;
    N /= 2;
}
```

$O(\log n)$

```
void function(int n) {
    if (n == 0)
        return;
    function(n - 1);
    function(n - 1);
}
```

$O(2^n)$

```
void function(int n) {
    for(int i = 0; i < n; i++) {
        function(n - 1);
    }
}
```

$O(n!)$

정렬!!

$O(n \log n)$

03

# Python3 필수문법

## lambda 함수

- `lambda <인자>:<리턴 값>`
- 함수를 간결하게 정의하기 위해 사용합니다.
- 이름이 없이 생성되므로 **익명 함수**라고도 불립니다.
- 이름을 줄 수도 있지만, 주로 **일회성**으로 사용합니다.
- 여러 번 쓰이는 함수를 작성할 때는 **기존 방식**으로 함수를 정의하세요.

```
1 # 기존 함수 작성 방식
2 def func1(x):
3     return 2*x
4
5 # 람다 함수 작성 방식
6 lambda x:2*x
7
8 # 물론 원하면 이름을 줄 수도 있습니다.
9 func2 = lambda x:2*x
```

## input() vs sys.stdin.readline()

- sys.stdin.readline()이 속도면에서 우수하다!
- 입력이 많은 문제의 경우, input() 함수 사용시 시간초과가 발생할 수 있다.
- PS에서는 sys.stdin.readline()을 사용하자.

### 차이점

```
1 msg = input()
2 print(msg)
```



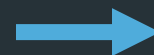
```
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
input() 함수는 개행문자를 제거합니다.
input() 함수는 개행문자를 제거합니다.
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
```

```
1 import sys
2 input = sys.stdin.readline
3 msg = input()
4 print(msg)
```



```
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
sys.stdin.readline()은 개행문자를 제거하지 않습니다.
sys.stdin.readline()은 개행문자를 제거하지 않습니다.
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
```

```
1 import sys
2 input = sys.stdin.readline
3 msg = input().rstrip()
4 print(msg)
```



```
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
rstrip()을 통해 개행문자를 제거합니다.
rstrip()을 통해 개행문자를 제거합니다.
Q /mnt/c/U/J/Doc/Al/Altube-2-preparation/03월
```

## map()

**map**(*function, iterable, ...*)

Return an iterator that applies *function* to every item of *iterable*, yielding the results. If additional *iterable* arguments are passed, *function* must take that many arguments and is applied to the items from all iterables in parallel. With multiple iterables, the iterator stops when the shortest iterable is exhausted. For cases where the function inputs are already arranged into argument tuples, see `itertools.starmap()`.

- 반복이 가능한 객체 ex. list, tuple, dictionary
  - 즉, map() 함수는 두번째 인자로 받은 객체의 원소를 하나씩 첫번째 인자로 받은 함수에 넣고, 함수의 리턴 값을 iterator 형태로 돌려줍니다.
  - 이 값을 받기 위해서는 `list()`, `tuple()` 등으로 감싸주거나, 직접 변수에 할당해야 합니다.

```
1 map(lambda x:x**2, range(5))
[2] ✓ 0.4s
... <map at 0x7f0b9c036ac0>
```

```
1 list(map(lambda x:x**2, range(5)))
✓ 0.9s
[0, 1, 4, 9, 16]
```

- 앞서 설명한 `input()`과 `sys.stdin.readline()` 모두 입력을 한 줄씩, string으로 받는 함수
  - 따라서, 공백으로 구분되는 두 숫자를 입력 받기 위해서는 `split()`을 통해 공백으로 구분한 리스트를, map으로 `int()` 함수를 적용해 정수 타입으로 바꿔서 n과 m에 대입합니다.

예제 입력 1 복사

7 3

```
1 n, m = map(int, input().split())
2 print(n, m)
```

# 04

## 백준 사용법

BAEKJOON>  
ONLINE JUDGE

회원가입

로그인

문제

문제집

대회

채점 현황

랭킹

게시판

그룹

블로그

강의

Q

2022 ICT멘토링 프로젝트 참여대학생 모집

ICT멘토링 취업백신으로  
부스터샷맞고 취업성공까지  
1차, 2차, n차까지 지원

ICT멘토링 백신으로

ICT취업  
그린패스

Baekjoon Online Judge

프로그래밍 문제를 풀고 온라인으로 채점받을 수 있는 곳입니다.



BAEKJOON

ONLINE JUDGE

bsa0322

설정

로그아웃

83

★

문제

문제집

대회

채점 현황

랭킹

게시판

그룹

블로그

강의

Q

2022 ICT멘토링 프로젝트 참여대학생 모집

ICT멘토링 취업백신으로  
부스터샷맞고 취업성공까지  
1차, 2차, n차까지 지원

ICT멘토링 백신으로

ICT취업  
그린패스

Baekjoon Online Judge

프로그래밍 문제를 풀고 온라인으로 채점받을 수 있는 곳입니다.

[정보 수정](#)

[비밀번호 변경](#)

**[학교/회사 정보](#)**

[계정 연동](#)

[링크 설정](#)

solved.ac

[보기 설정](#)

[알림 설정](#)

[테마 설정](#)

[언어 설정](#)

[소스 코드 공개 설정](#)

[문의하기 / 탈퇴하기](#)

### 학교/회사 정보

학교/회사	이메일	삭제
이화여자대학교	bsa0322@ewhain.net	<a href="#">삭제</a>

검색이 되지 않는 학교/회사나, 학교/회사 이메일이 잘못된 경우, 초/중/고등학교, 회사 추가 요청, 이메일이 없는 경우 모두 문의하기를 이용해주세요. (학교/회사 도움말)

#### 학교/회사 정보 추가

학교/회사 이름	<div>이름</div>	학교	주소	선택
학교/회사 이메일	<div>이메일</div>			
<div>추가</div>				

2022-1 SW학부 원스탑 튜터 알튜비튜

정보 수정
비밀번호 변경
학교/회사 정보
계정 연동
링크 설정

solved.ac

보기 설정
알림 설정
테마 설정
언어 설정
소스 코드 공개 설정

문의하기 / 탈퇴하기

solved.ac



solved.ac는 BOJ 유저 [shiftpsh](#)님이 만든 서비스입니다.

이 서비스는 Baekjoon Online Judge의 문제의 난이도를 보고, 유저의 티어를 볼 수 있습니다.

다음 데이터를 solved.ac에 제공해 bsa0322님의 티어를 계산 및 공개하려면 사용하기를 눌러주세요.

- 아이디
- 모든 채점 결과

제출한 소스 코드는 solved.ac에 **제공하지 않습니다.**

사용 중

그만 사용하기

정보 수정

비밀번호 변경

학교/회사 정보

계정 연동

링크 설정

solved.ac

보기 설정

알림 설정

테마 설정

언어 설정

소스 코드 공개 설정

문의하기 / 탈퇴하기

보기 설정

알고리즘 분류

☒ 보기

☐ 보지 않기

알고리즘 분류 언어

☒ 한국어

☐ 영어

문제 언어

☒ 한국어

☐ 영어

☐ 원문

문제 메모

☒ 메모 보기

☐ 메모 보지 않기

예제 공백 하이라이트

☐ 보기

☒ 보지 않기

solved.ac 티어

☒ 보기

☐

성공

 인 경우만 보기

☐

성공

부분 성공

 인 경우만 보기

☐ 보지 않기

solved.ac 티어 이름  
(문제 제목 하단)

☐ 보기

☒ 보지 않기

런타임 에러 이유

☒ 보기

☐ 보지 않기

문제 폰트

☒ Sans-serif

☐ Serif

☐ Cursive

☐ Monospace

변경

AC solved.ac 문제 기여 랭킹 정보

BOJ로 로그인

SOLVED AC

우리 모두가 만들어가는  
알고리즘 문제해결 학습의 이정표

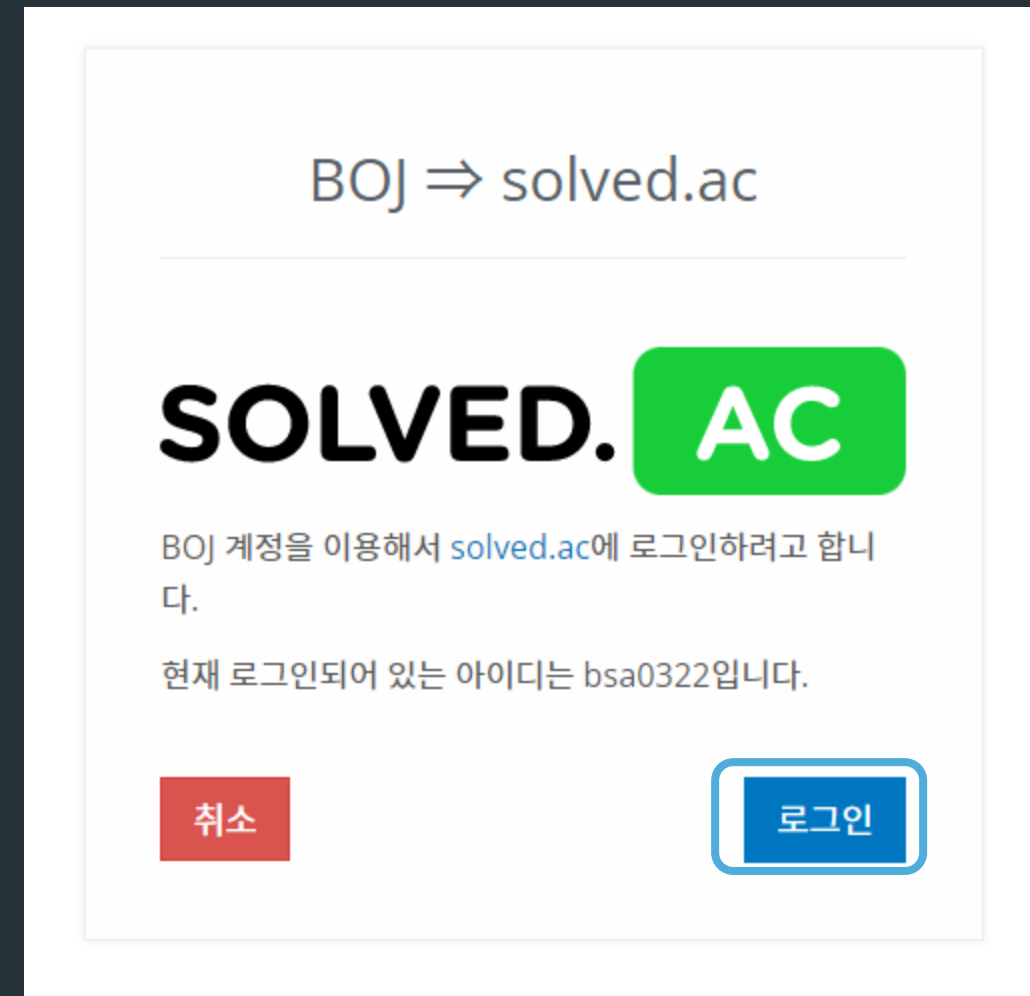
solved.ac는 Baekjoon Online Judge 문제들에 태그와 난이도를 붙이는 커뮤니티 프로젝트입니다.  
현재 **12,630**개 문제에 난이도 정보를 제공하고 있습니다.

길라잡이 →

문제 둘러보기 →

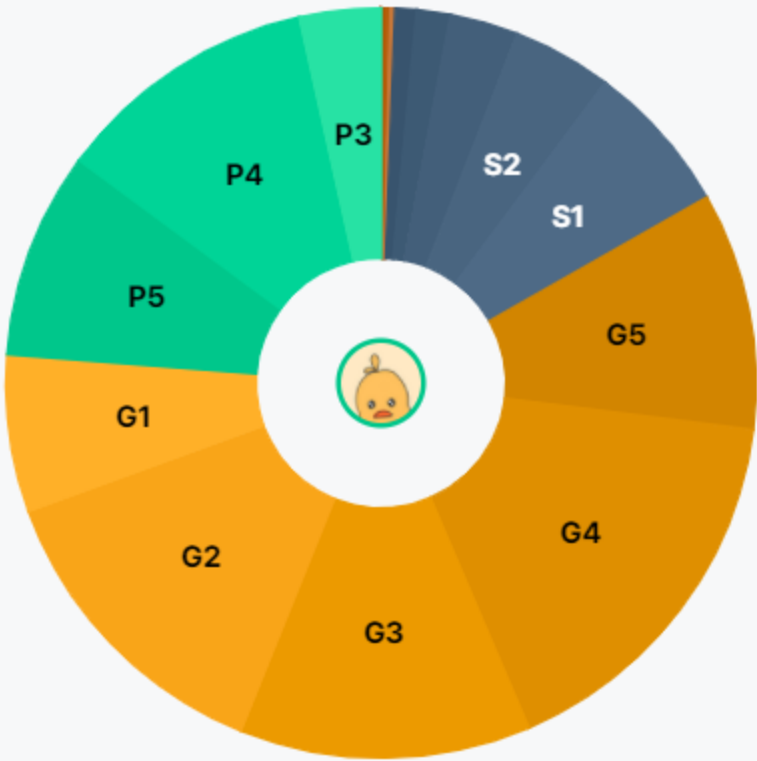
확장 프로그램은 지원이 종료되었습니다. 이제 BOJ에서 난이도를 확인할 수 있습니다.  
BOJ 로그인 후, "설정 > 문제 보기 설정 > solved.ac 티어"에서 난이도 표시를 켜고 끌 수 있습니다.

BOJ로 로그인



난이도 분포

경험치 11,649,087



레벨	문제		EXP	
BRONZE	61	12.6%	70,992	0.6%
SILVER	237	49.0%	1,881,150	16.1%
GOLD	174	36.0%	6,926,476	59.5%
PLATINUM	12	2.5%	2,770,149	23.8%
DIAMOND	0	0.0%	0	0.0%
RUBY	0	0.0%	0	0.0%

5 1000번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

🔗 난이도 기여

강의▼

질문 검색

A+B

성공

다국어

☆

한국어 ▼

시간 제한

메모리 제한

제출

정답

맞힌 사람

정답 비율

2 초

128 MB

549301

230047

164834

42.490%

## 문제

두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

## 입력

첫째 줄에 A와 B가 주어진다. ( $0 < A, B < 10$ )

## 출력

첫째 줄에 A+B를 출력한다.



# 백준 문제 제출 방법

5 1000번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

난이도 기여

강의

질문 검색

A+B

언어

C++17

언어 설정

언어

Python 3

or

언어

PyPy3

소스 코드 공개

☐ 공개

☐ 비공개

☒ 맞았을 때만 공개

소스 코드

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int a, b;
7     cin >> a >> b;
8     cout << a + b;
9 }
```

제출

간혹 python3로는 시간 안에 풀리지 않는 문제가 있습니다. 이 경우 언어를 PyPy3로 바꾸어 제출해보세요!

# 백준 문제 제출 방법



39778160	bsa0322	1000	맞았습니다!!	2020 KB	0 ms	C++17 / 수정	108 B	7초 전	
----------	---------	------	---------	---------	------	------------	-------	------	--

연습 이름	우승	준우승	시작	종료	상태
12 Dijkstra			2020년 8월 27일 20:00	2020년 9월 3일 23:59	종료
11 Priority Queue			2020년 8월 24일 18:01	2020년 8월 31일 23:59	종료
10 (C++) Map, Set / (Python) Dictionary, Set			2020년 8월 20일 18:00	2020년 8월 27일 23:59	종료
09 Tree			2020년 8월 17일 17:30	2020년 8월 24일 23:59	종료
08 DFS/BFS			2020년 8월 13일 18:23	2020년 8월 20일 23:59	종료
07 Greedy			2020년 8월 10일 18:00	2020년 8월 17일 23:59	종료
06 Queue, Deque			2020년 8월 6일 17:30	2020년 8월 13일 23:59	종료
05 Stack			2020년 8월 3일 16:00	2020년 8월 10일 23:59	종료
04 Binary Search			2020년 7월 30일 18:00	2020년 8월 6일 23:59	종료
03 Sorting			2020년 7월 23일 18:00	2020년 7월 30일 23:59	종료
02 Backtracking			2020년 7월 16일 18:02	2020년 7월 23일 23:59	종료
01.5 Permutation and Combination (C++)			2020년 7월 13일 14:00	2020년 7월 22일 23:59	종료
01.5 Permutation and Combination (Python)			2020년 7월 13일 14:00	2020년 7월 22일 23:59	종료
01 Brute-force (+Time complexity)			2020년 7월 9일 18:00	2020년 7월 16일 17:59	종료

A - 블랙잭	순위	아이디	A	B	C	D	E	F	G	H	I	
B - 분해합	1		1 / 14	2 / 46	1 / 39	1 / 46	1 / 53	1 / 66	1 / 77	1 / 88	1 / 96	9 / 525
C - 기숙사 바닥	2		1 / 23	3 / 79	1 / 72	1 / 78	2 / 115	2 / 81	3 / 154	1 / 146	1 / 153	9 / 901
D - 거꾸로 구구단	3		1 / 185	1 / 55	3 / 258	1 / 164	1 / 224	1 / 126	1 / 122	1 / 133	1 / 57	9 / 1324
E - Doubles	4		1 / 87	1 / 104	1 / 113	1 / 140	1 / 145	1 / 203	1 / 273	4 / 307	1 / 263	9 / 1635
F - 체스판 다시 칠하기	5		1 / 174	1 / 177	1 / 202	1 / 229	1 / 249	1 / 178	1 / 179	1 / 179	1 / 178	9 / 1745
G - 영화감독 숨	6		1 / 147	1 / 148	2 / 172	1 / 362	1 / 247	1 / 245	1 / 345	1 / 340	1 / 313	9 / 2319
H - 덩치	7		1 / 372	1 / 393	1 / 411	2 / 459	6 / 586	1 / 489	1 / 488	1 / 454	1 / 446	9 / 4098
I - 한수	8		1 / 238	4 / 310	2 / 1261	2 / 350	1 / 378	1 / 283	1 / 2548	1 / 388	1 / 416	9 / 6172
	9		1 / 248	2 / 269	2 / 276	2 / 380	1 / 336	1 / 305	1 / 306	1 / 335	1 / 8776	9 / 11231
	10		1 / 159	2 / 219	1 / 226	1 / 262	1 / 177	2 / 5734	1 / 5316	1 / 174	1 / 162	9 / 12429

## 백준

- 다양한 알고리즘 문제 풀이
- 그룹을 활용해 과제 문제, 스코어보드 등 확인

## Solved.ac

- 문제 난이도 확인 가능
- 자신의 알고리즘 별 문제 풀이 등의 능력치 확인 가능
- 라이벌 콘텐츠 등의 재미요소

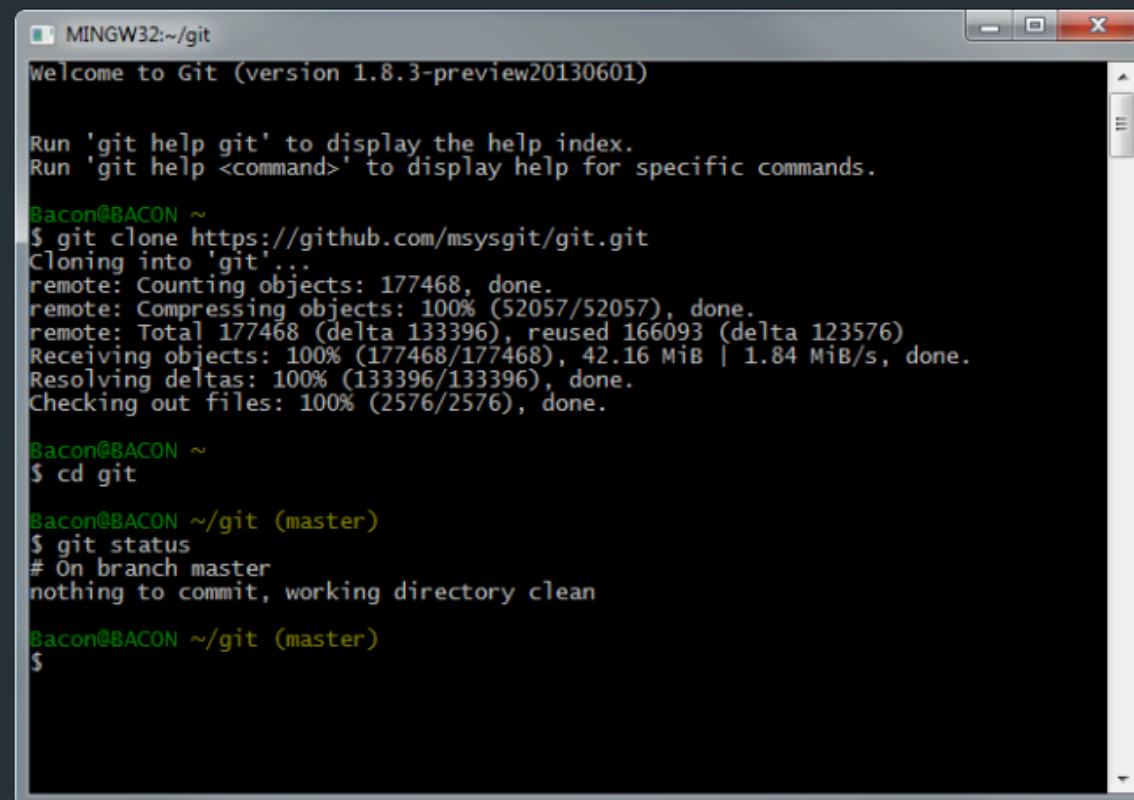
# 05

## 깃허브 사용법

## Git과 GitHub는 다르다

- Git      작업이 겹치지 않도록 프로젝트의 변경을 관리하는 버전관리 소프트웨어
- GitHub    이러한 Git을 사용하여 원격 저장소에서 관리하도록 만든 사이트가 GitHub!

## Git과 GitHub를 쉽게 사용하도록 해주는 소프트웨어



```
MINGW32:~/git
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

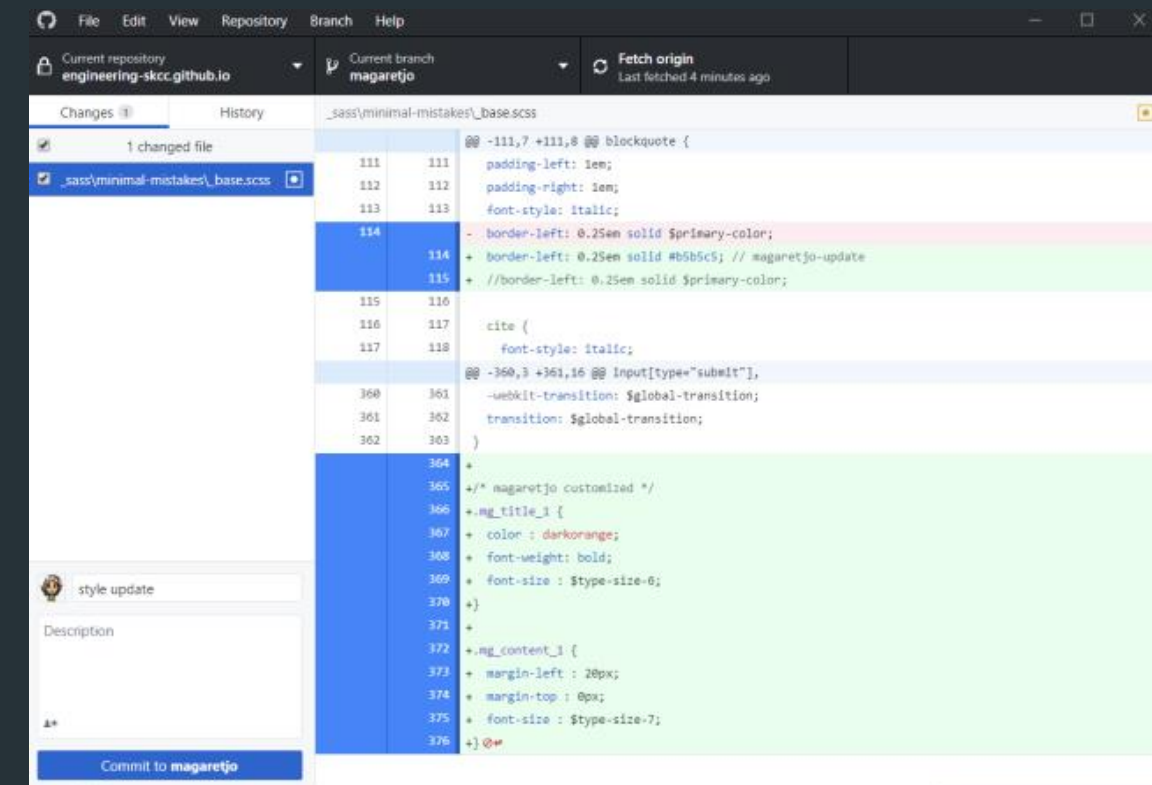
Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

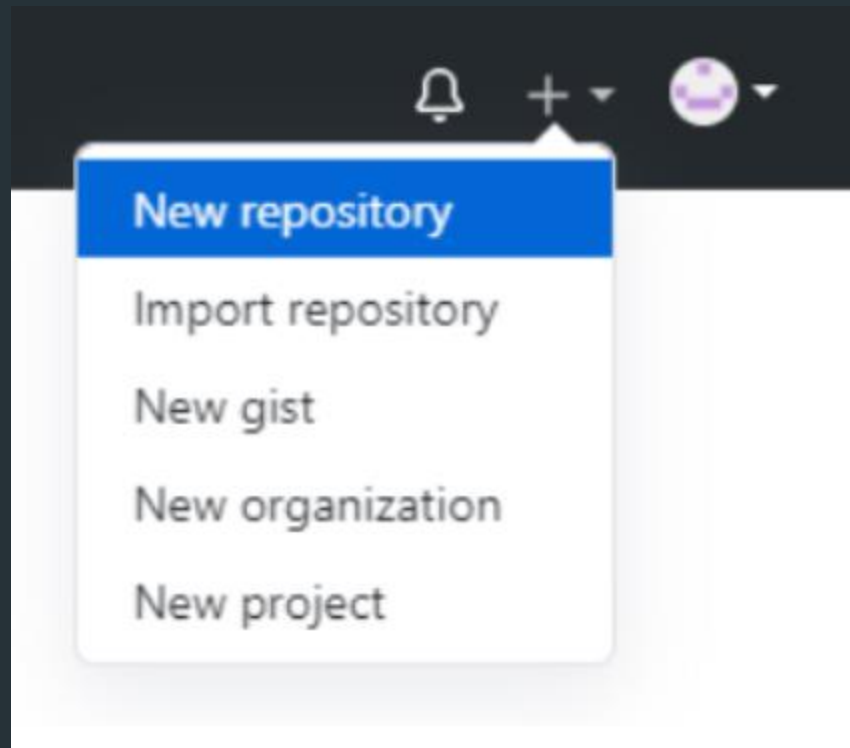
Git bash



GitHub Desktop



## 레파지토리 생성



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

**Repository template**  
Start your repository with a template repository's contents.

---

**Owner \***  **Repository name \***

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

**Description (optional)**

---

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

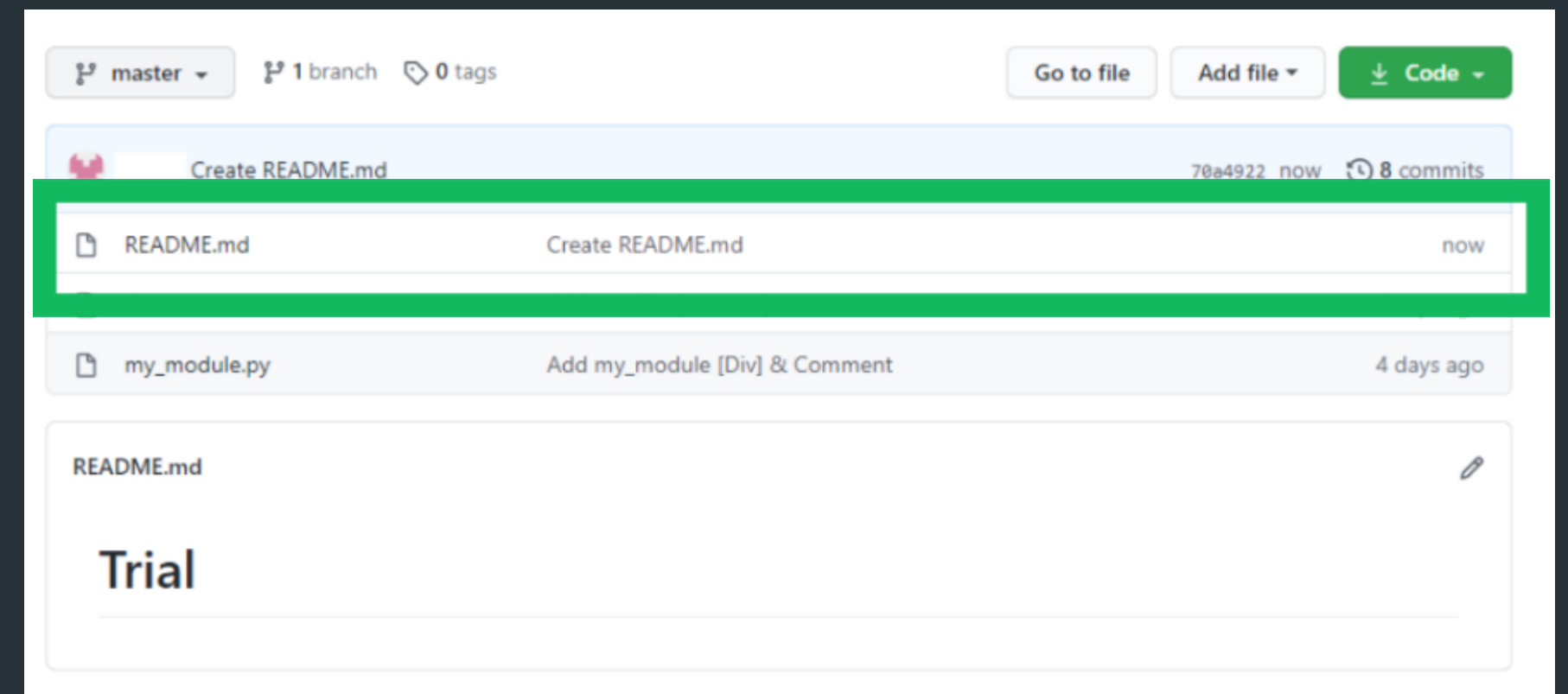
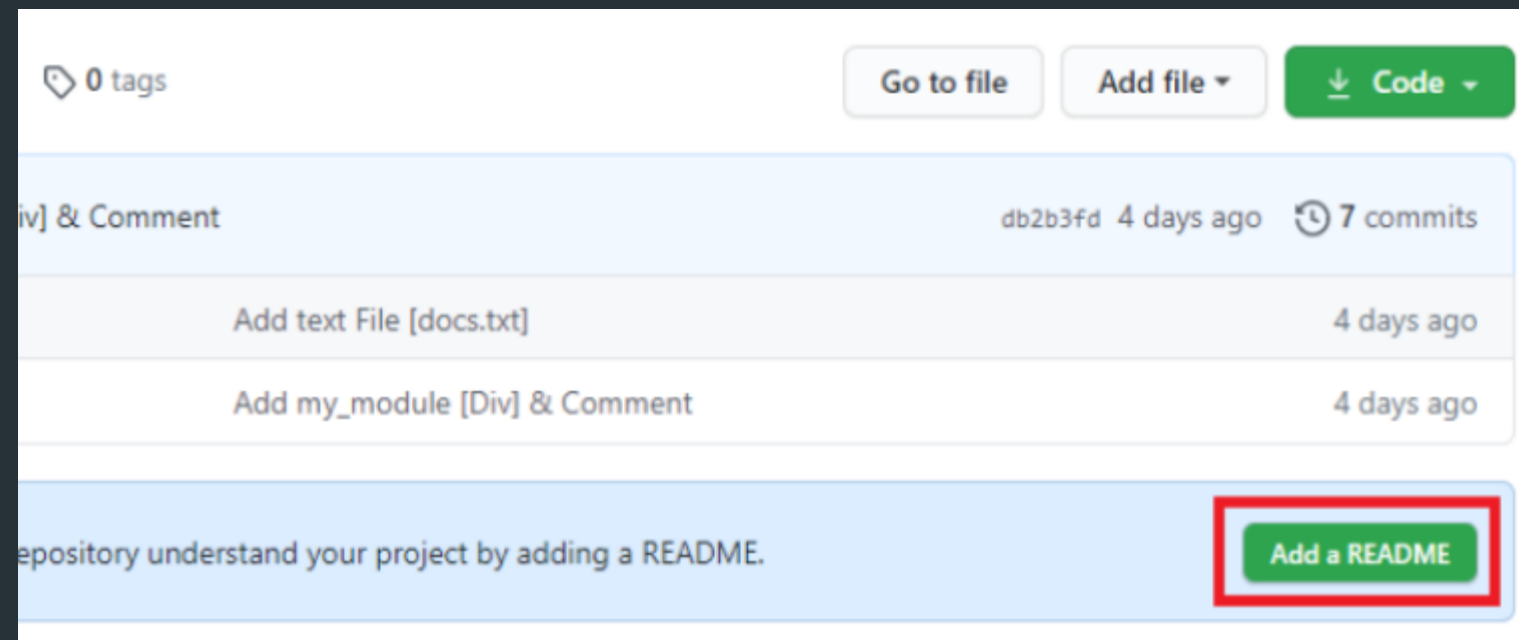
☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

---

## 리드미 작성

- 마크다운 문법을 익히자@



git add

- 새로운 파일 추가 (ex) git add 1000.app / git add . (변경 사항이 있는 모든 파일)

git commit

- 추가한 파일을 커밋 (ex) git commit -m "커밋 메시지 "

git push

- 커밋내역을 원격저장소에 push

git fetch

- 원격 저장소의 branc와 commi들을 로컬 저장소와 동기화

git pull

- 원격 저장소의 변경사항을 작업 공간에 가져오기

git clone

- 프로젝트를 복제 (ex) git clone <프로젝트 주소>

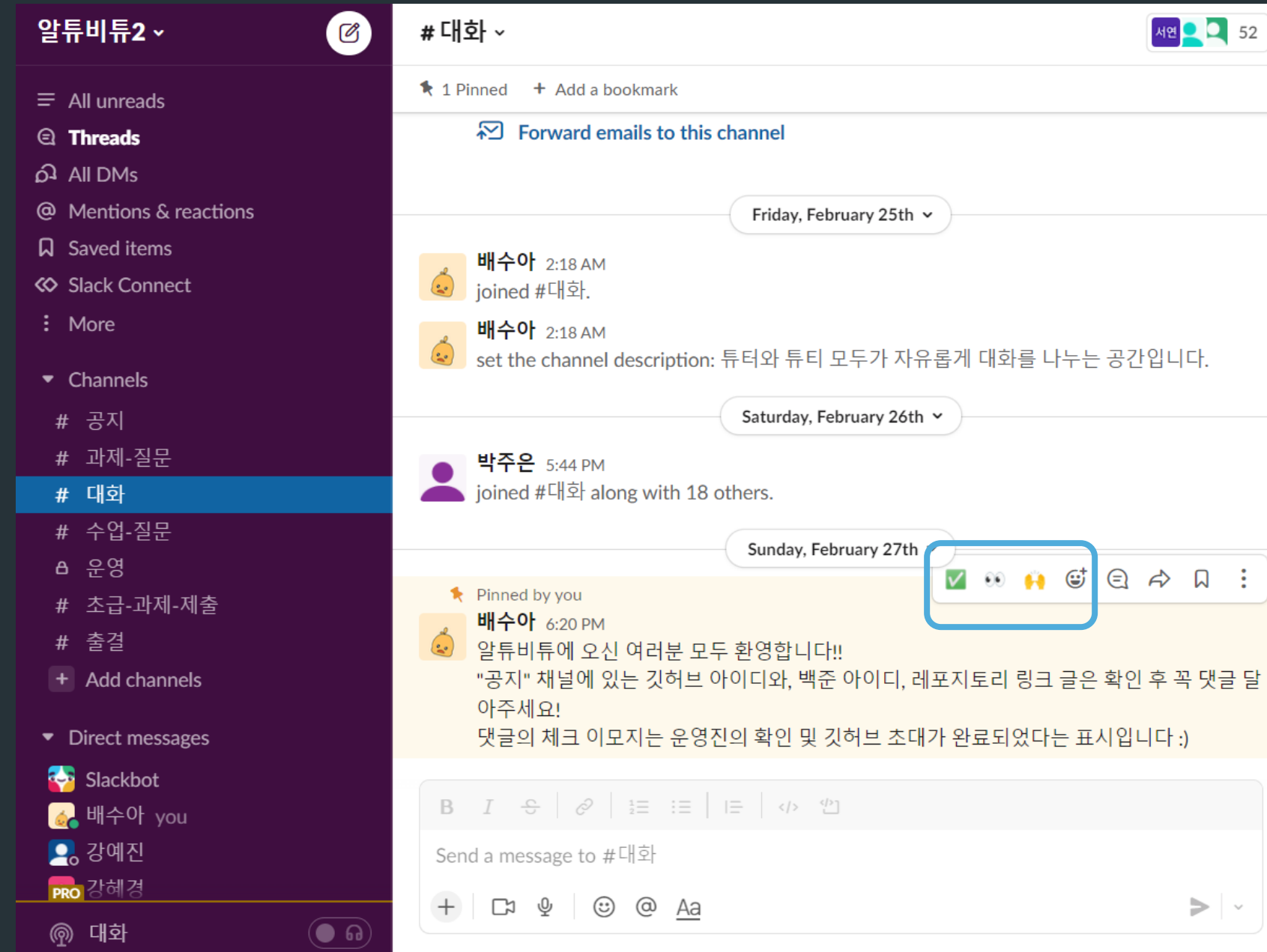
git checkout

- 해당 브랜치로 이동 (ex) git checkout <브랜치 이름>

# 06

## 슬랙 사용법

## 이모티콘 사용



알튜비뷰2 ▾

- All unreads
- Threads
- All DMs
- Mentions & reactions
- Saved items
- Slack Connect
- More
- Channels
  - # 공지
  - # 과제-질문
  - # 대화
  - # 수업-질문
  - # 운영
  - # 초급-과제-제출
  - # 출결
  - + Add channels
- Direct messages
  - Slackbot
  - 배수아 you
  - 강예진
  - 강혜경
- 대화

#대화 ▾ 52

1 Pinned + Add a bookmark

Forward emails to this channel

Friday, February 25th ▾

배수아 2:18 AM  
joined #대화.

배수아 2:18 AM  
set the channel description: 튜터와 튜티 모두가 자유롭게 대화를 나누는 공간입니다.

Saturday, February 26th ▾

박주은 5:44 PM  
joined #대화 along with 18 others.

Sunday, February 27th ▾

Pinned by you

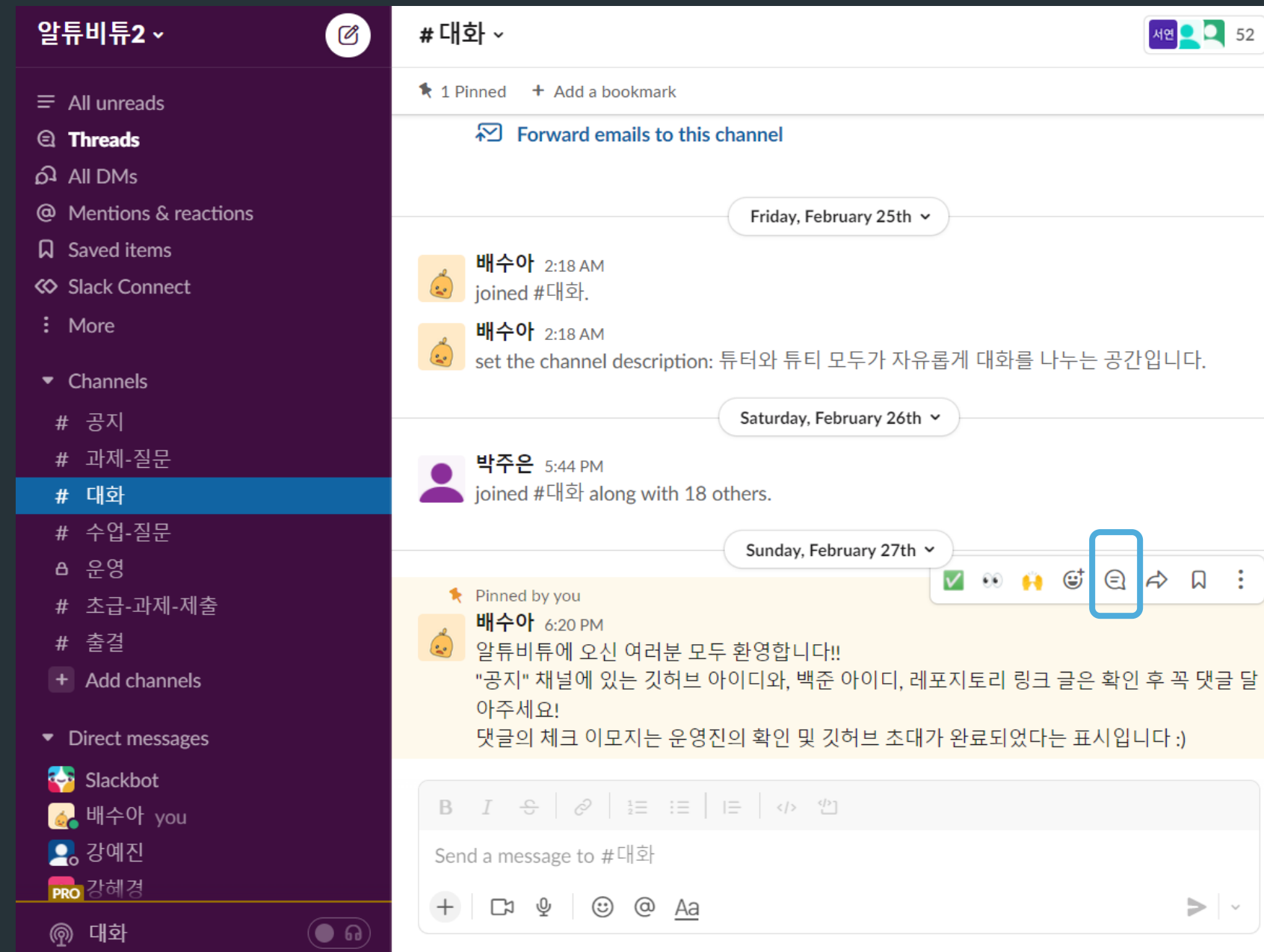
배수아 6:20 PM  
알튜비뷰에 오신 여러분 모두 환영합니다!!  
"공지" 채널에 있는 깃허브 아이디와, 백준 아이디, 레포지토리 링크 글은 확인 후 꼭 댓글 달아주세요!  
댓글의 체크 이모지는 운영진의 확인 및 깃허브 초대 완료되었다는 표시입니다 :)

B I [emojis] [link] [list] [code] [quote] [code]

Send a message to #대화

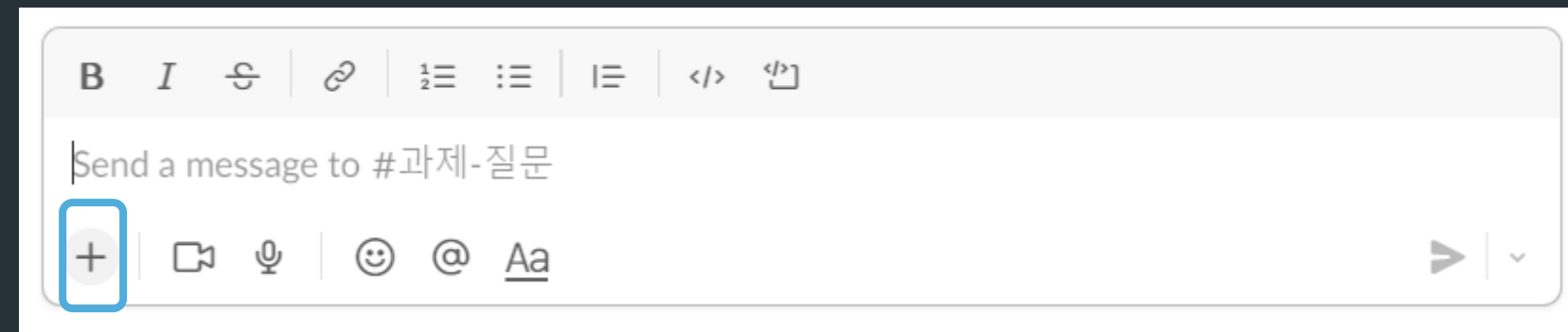
+ [attach] [voice] [emojis] [mention] [text] [send]

## 메시지에 thread 달기

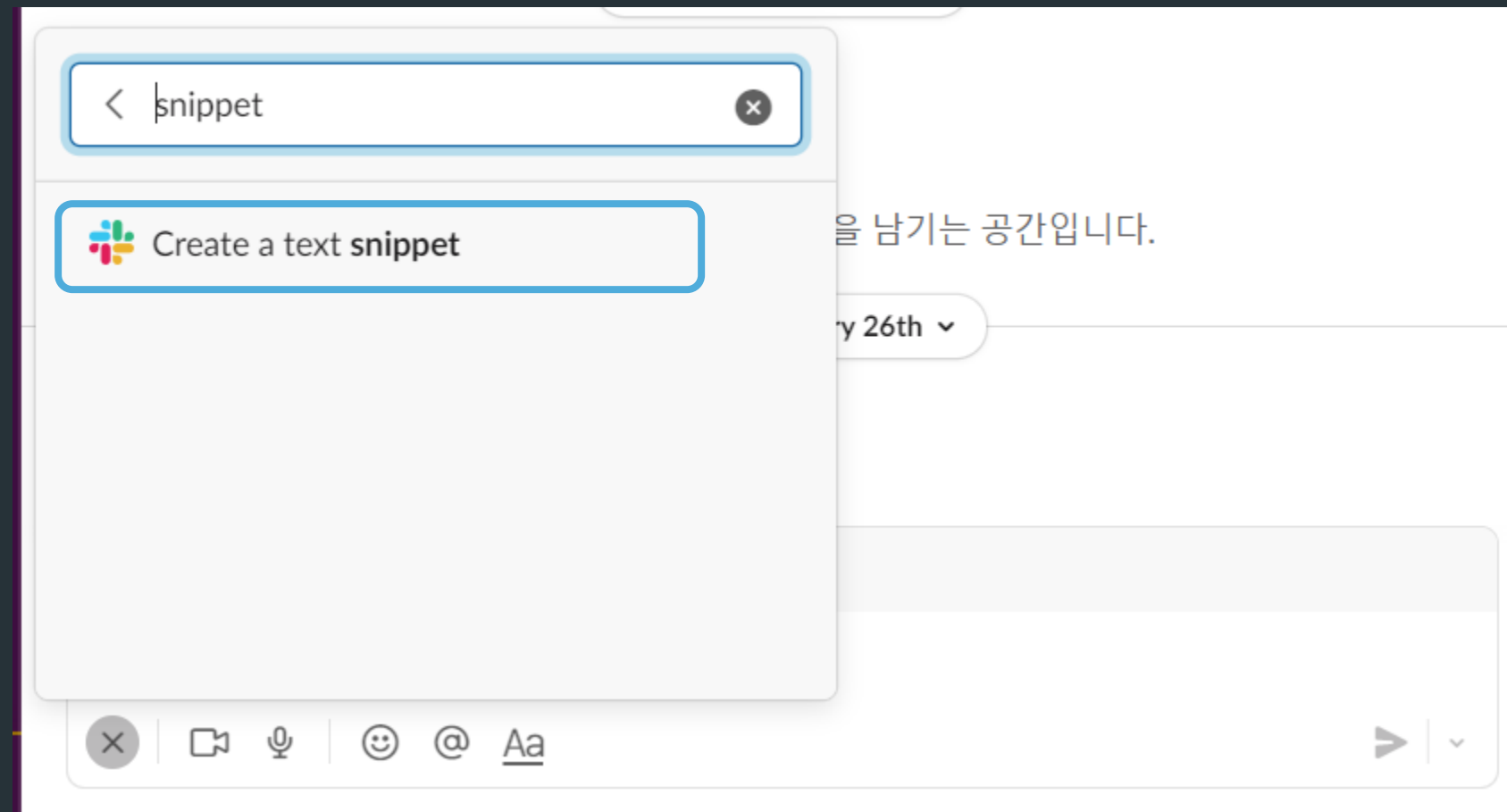


The screenshot shows the Slack interface for a channel named "#대화". The left sidebar lists various channels and direct messages, with "#대화" selected. The main area displays a message from "배수아" (Baesooa) at 2:18 AM, which has been pinned. The message content is: "알튜비튜에 오신 여러분 모두 환영합니다!! '공지' 채널에 있는 깃허브 아이디와, 백준 아이디, 레포지토리 링크 글은 확인 후 꼭 댓글 달아주세요! 댓글의 체크 이미지는 운영진의 확인 및 깃허브 초대가 완료되었다는 표시입니다 :)". The message is highlighted in yellow. Below the message, there is a row of icons for actions: a green checkmark, a speech bubble, a thumbs up, a thumbs down, a thread icon (a speech bubble with a tail), a share icon, a bookmark icon, and a more options icon (three dots). The thread icon is highlighted with a blue box. At the bottom of the screen, there is a text input field for sending a message to "#대화".

## 코드 올리기



## 코드 올리기





## 코드 올리기

### Create snippet

Title (optional)  
1000.cpp

Type  
C++

Content  

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     cout << "Hello World!";
7 }
```

☐ Wrap Press Esc to stop editing.

**B** *I*

Add a message, if you'd like.

@ Aa

☒ Share this file

# 과제-질문

Create Snippet

## 질문 형식

[문제 번호]  
질문 내용  
(관련 링크)  
1000.cpp ▼

1 코드 블록 (선택)

## 과제

[주제]  
질문 내용  
(관련 링크)

## 수업

- 위 형식에 맞춰서 “과제-질문” 채널과 “수업-질문” 채널에 각각 올려주시면 튜터가 thread로 답변해드리겠습니다!

# 07

## 규칙

## 출결 규칙

- 튜터링 시작 전날 자정까지 “출결” 채널에 결석 사유 메시지 남기기 → 사유결석
- 무단으로 결석 2번 시, 진행 의사를 여쭙볼 예정입니다...
- 무단으로 결석 3번 시, 튜터링 제외
- 사유결석 3번은 무단결석 1번으로 카운트!

## 레파지토리 생성 후 조직으로 소유자 변경

- 개인 저장소의 [Settings / Danger Zone / Transfer ownership](#) 메뉴에서 [Transfer](#) 를 누른다.
- 현재 저장소의 owner를 [Altu-Bitu-2](#) 로 변경한다.
- 컨펌을 위해 현재 저장소 이름을 한번 더 적어준다.

## 기한

- 일주일!!
- 일주일 후 튜터링 전날 낮 12시까지 제출 → 코드리뷰 O
- 일주일 후 튜터링 전날 밤 12시까지 제출 → 코드리뷰 X
- 과제 5개를 다 채우지 못했다면...? → 우선 제출하고, 못한 개수만큼 추가제출

## 추가제출이란?

- 샘플코드의 모든 라인에 주석을 달아서 제출하는 것!
- 기한: 일주일 후 튜터링 당일 밤 12시까지

# 조금 헛갈리죠...?

## 3월 4일에 나간 과제라면!!

- 3월 10일 낮 12시까지 제출 → 코드리뷰 O
- 3월 10일 밤 12시까지 제출 → 코드리뷰 X
- 3월 11일 밤 12시까지 추가제출

## 코드 작성법

- 변수: 스네이크 표기법
- 함수: 카멜 표기법
- 상수: 대문자

```
int hello_world; //변수  
int helloWorld() //함수  
const int HELLO_WORLD //상수
```



## 클린 코드 작성법

- 전역 변수 최대한 사용 ✕
- 함수의 용도는 명확하게, 가능한 메인은 짧게
- 인덴테이션은 최대 3-depth 까지만 (삼중 for문까지 가능)
- 구현이 까다로운 부분은 주석으로 보충 설명

## 폴더 구조

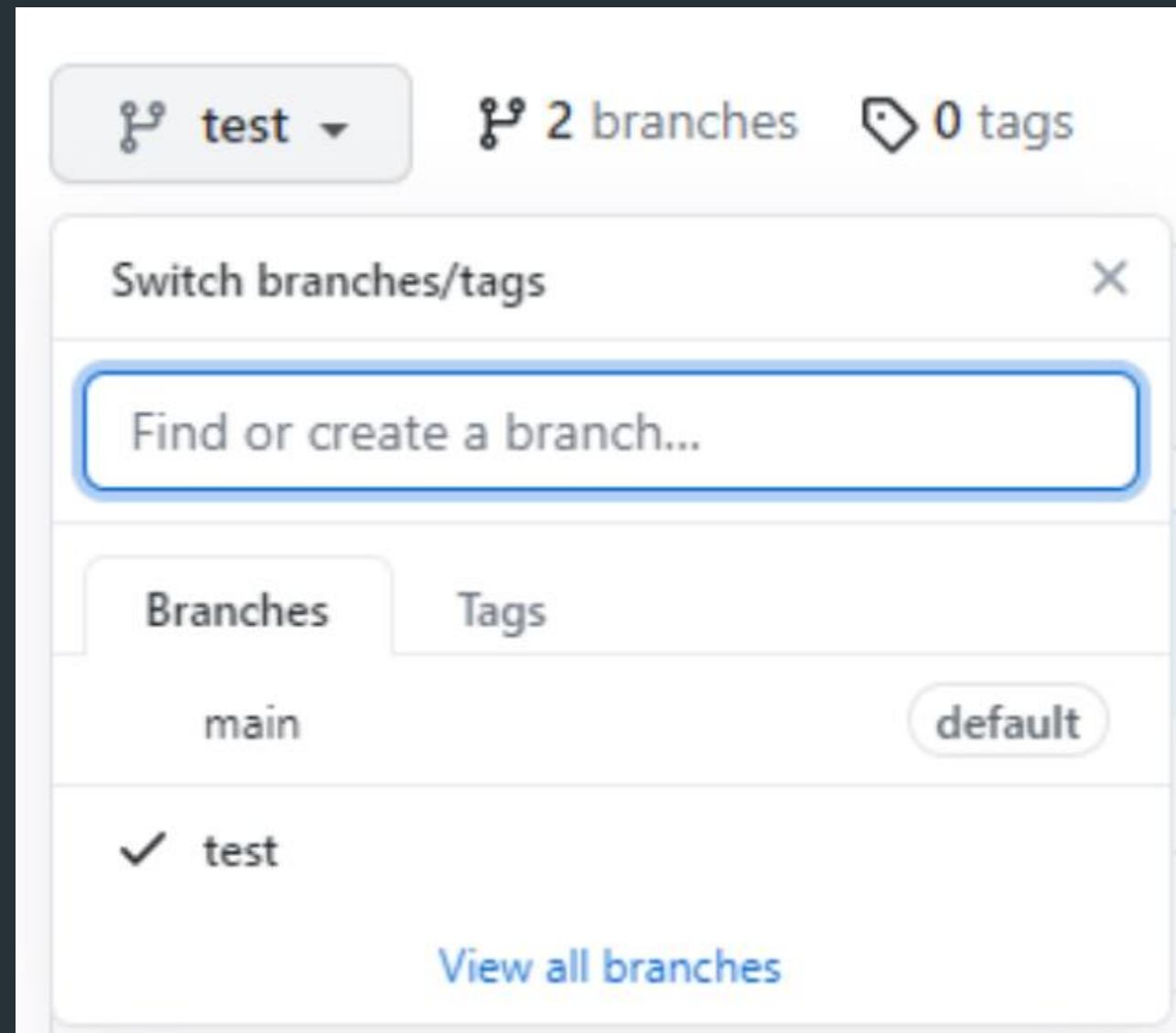
```
03월 04일 - 정렬 <!--(튜터링 날짜) - (알고리즘명)-->  
└─ 1886.cpp <!--(문제 번호).cpp-->
```

## 커밋 메시지




- 제출: [ <알고리즘 명> ] <제출 날짜>
  - (ex). [정렬] 3월 4일
- 수정 시: [ <알고리즘 명> ] <제출 날짜> - Update
  - (ex). [정렬] 3월 5일 - Update
- 풀고 있는 중: [ <알고리즘 명> ] <제출 날짜> - ing
  - (ex). [정렬] 3월 6일 - ing

## 쿠폰

- 주어진 선택 과제 5개와 필수 구현 2개를 모두 푼 경우 → 쿠폰 1개 (튜터들이 기록)
- 쿠폰 혜택: 과제 문제 1개 안 풀기 (대신, 한 수업당 한 개만 사용 가능!!)









branch를 생성해주세요!  
이름은 자유롭게!

 test ▾  2 branches  0 tags


[Go to file](#) [Add file ▾](#) [Code ▾](#)

This branch is 1 commit ahead of main. [Contribute ▾](#)

 bsa0322 test 70a2adf 16 seconds ago  11 commits

 test	test	15 seconds ago
 .gitignore	Update .gitignore	7 days ago
 LICENSE.md	create LICENSE.md	6 days ago
 README.md	Update README.md	6 days ago

해당 **branch**에 소스코드 올린 후 **커밋푸쉬**하기  
절대 **main**에 곧바로 올리지 말아주세요!!! 코드 리뷰를 할 수 없어요.

 test had recent pushes less than a minute ago

Compare & pull request

“Compare & pull request” 버튼을 눌러 pr 활성화

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: test ✓ Able to merge. These branches can be automatically merged.

test

Write Preview

과제 제출합니다~! 그런데 1000을 브루트 포스로 풀리긴 했는데 비효율적인 거 같아서요 더 효율적인 방법이 있을까요?

기존 제출: 1000, 1001  
추가 제출: 1002

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

“Create pull request” 버튼으로 pr 날리기!  
만약 튜터가 특별히 보셨으면 하는 부분이나 질문이 있다면 write에 적어주세요!

기존 제출: 1000, 1001  
추가 제출: 1002

추가 제출과 분리하기 위해 기존 제출 과제와 추가 제출 과제가 무엇인지  
write에 꼭 명시해주세요!





배수아 10:48 PM

3월 4일 배수아 <https://github.com/Altu-Bitu-2/Notice/pull/1>

Pr 생성 후, 슬랙 “**초급-과제-제출**” 채널에 다음과 같이 남겨주시면, 튜터가 차례대로 확인을 해드릴 예정입니다!  
이후 튜터의 **머지해도 된다는 코멘트**를 얻으면 main에 머지해주세요!

기존 제출: <튜터링 날짜> <이름> <pr 링크>  
추가 제출: <튜터링 날짜> <이름> 추가 제출 <pr 링크>

## 튜터의 코멘트는 이런 뜻이에요

**p1** 꼭 반영해주세요: 코드가 잠재적인 버그 가능성을 내포하고 있을 때

**p2** 고려해 주세요: 좀 더 효율적인 코드로 작성할 수 있을 때

**p3** 사소한 의견: 가독성 면에서 수정할 만한 부분이 있을 때

- P1을 제외하고는 튜터의 코멘트를 받은 후, 수정 없이 머지 가능합니다~!

# 알튜비튜2

## OT



감사합니다!