Code Review

What is code review?

contributor's code is **reviewed** by their **peers** to

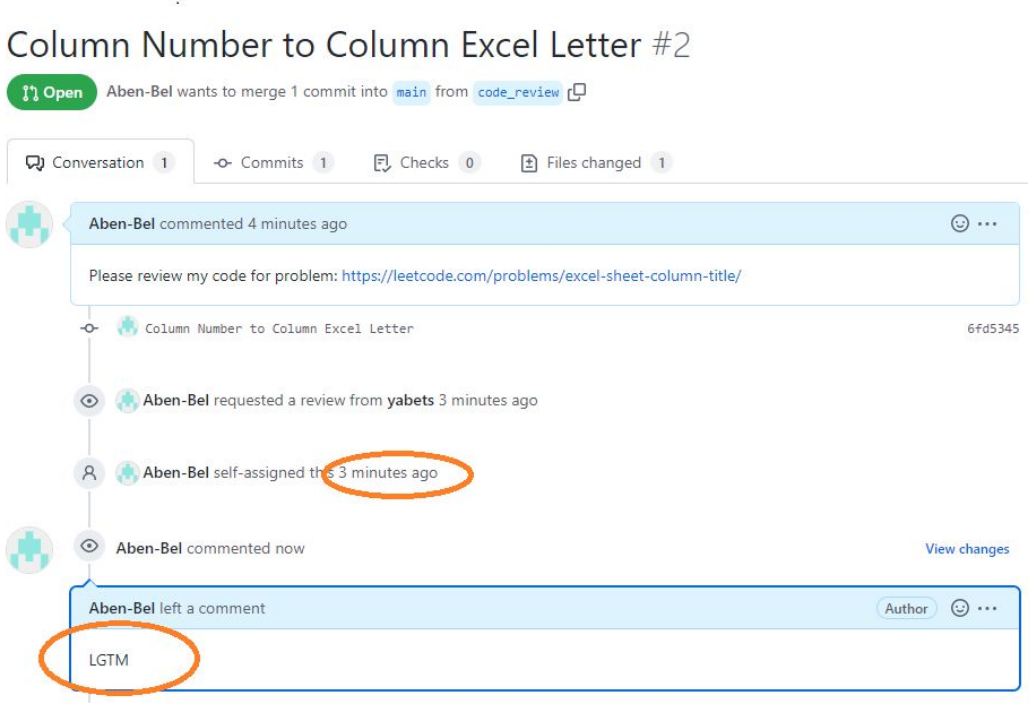
- Improve code quality,
- Avoid logical errors,
- Avoid requirement misses,
- Check style consistency
- Learn from each other.

Why code review is important?

- Humans make many **mistakes**, and code reviews can help catch those mistakes
- Creates an environment where everyone **learns and grows together**
- Getting a code review is getting **immediate feedback**; we learn best with immediate feedback
- Serve as an **accountability** system that ensures the code is written not only works but is also readable

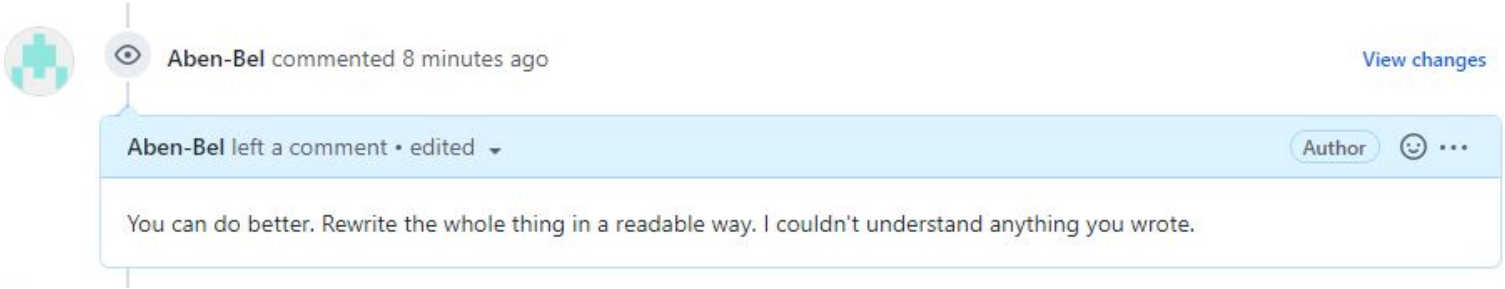
Signs of a bad code review

LGTM(looks good to me) [after 5 mins of review request]



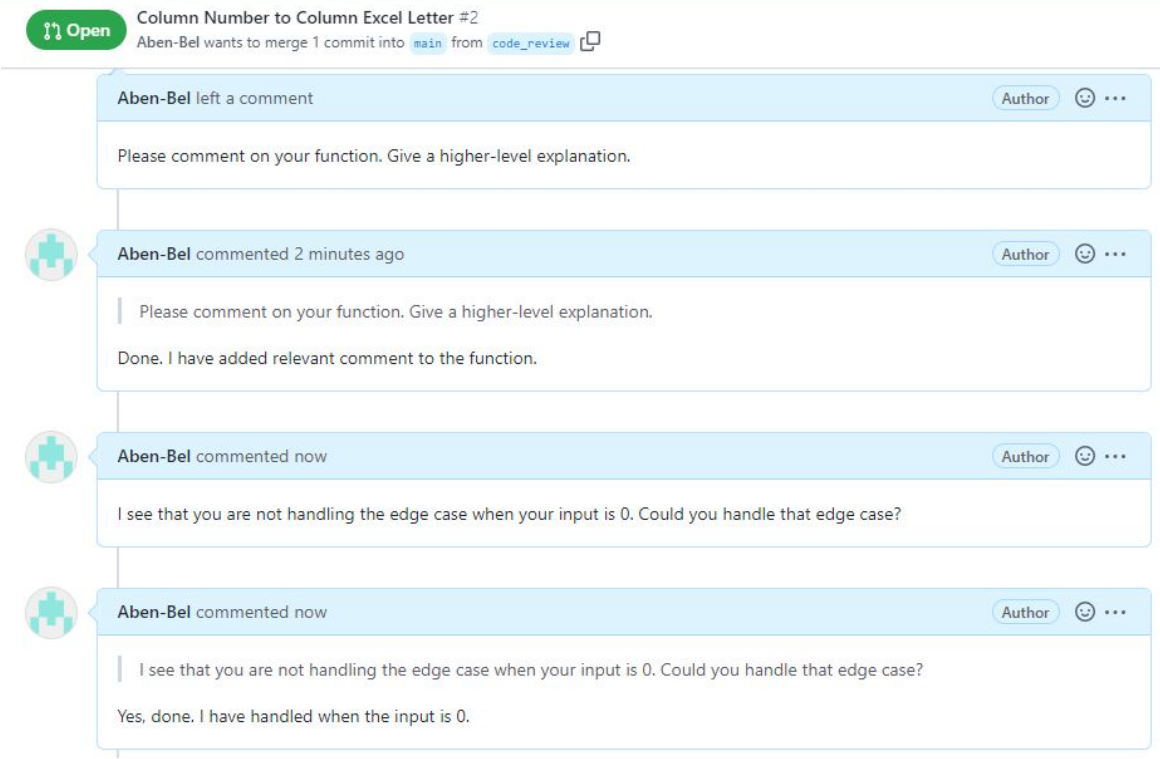
Signs of a bad code review

Not empathic (reviewer is rude)

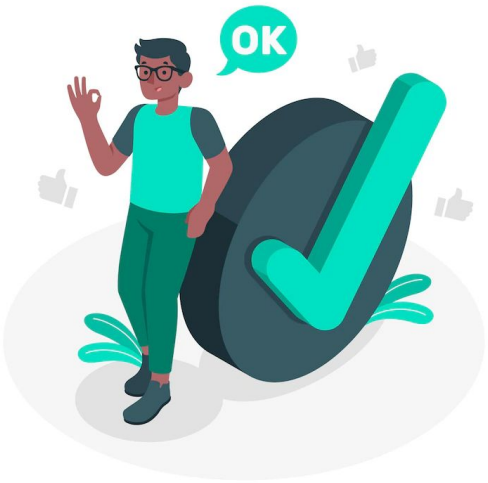


Signs of a bad code review

Chained back and forth between reviewer and reviewee



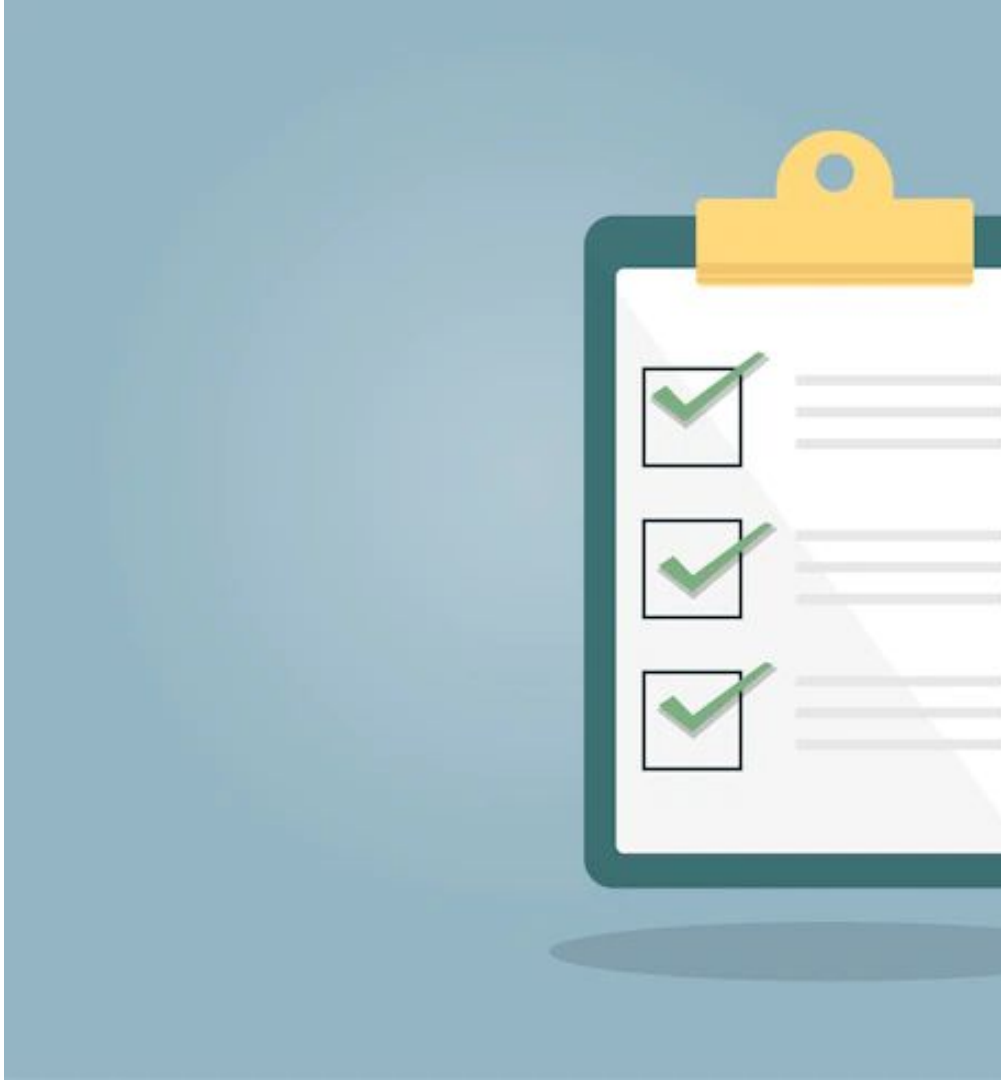
Sign of good Code Review



- Responsibility and ownership
- Code Review are done timely
- Be kind and thorough
- Your code review have these things
  - The code isn’t more complex than it needs to be
  - The author used clear names for everything.
  - Comments are clear and useful, and mostly explain why instead of what.
  - The code conforms to our style guides(best coding practices lecture)

What to review?

- **Readability**
  - Explicit than implicit
  - Intent is obvious
  - Naming
  - Conventions
- **Edge cases**
  - Is there better way to handle edge cases?
- **Time and space complexity**
  - Can it be improved?



How to review code?

- Are you the right person to review the task?
  - If not, do your research on the problem
- Set aside time
- Get the context
  - What is the task?

- Read the code to **understand**
  - What is it doing?
  - What are the other ways to solve the problem?
  - What are the time and space complexity of the solution provided?

- Write **comments**
  - Take notes as you go
  - Each comment should have a reason(why?)
  - After you finish, rewrite your comments
  - Leave some positive comments
  - You can provide resources for the reader to follow you
  - Comment the code(don’t comment the person)

Let’s do an example

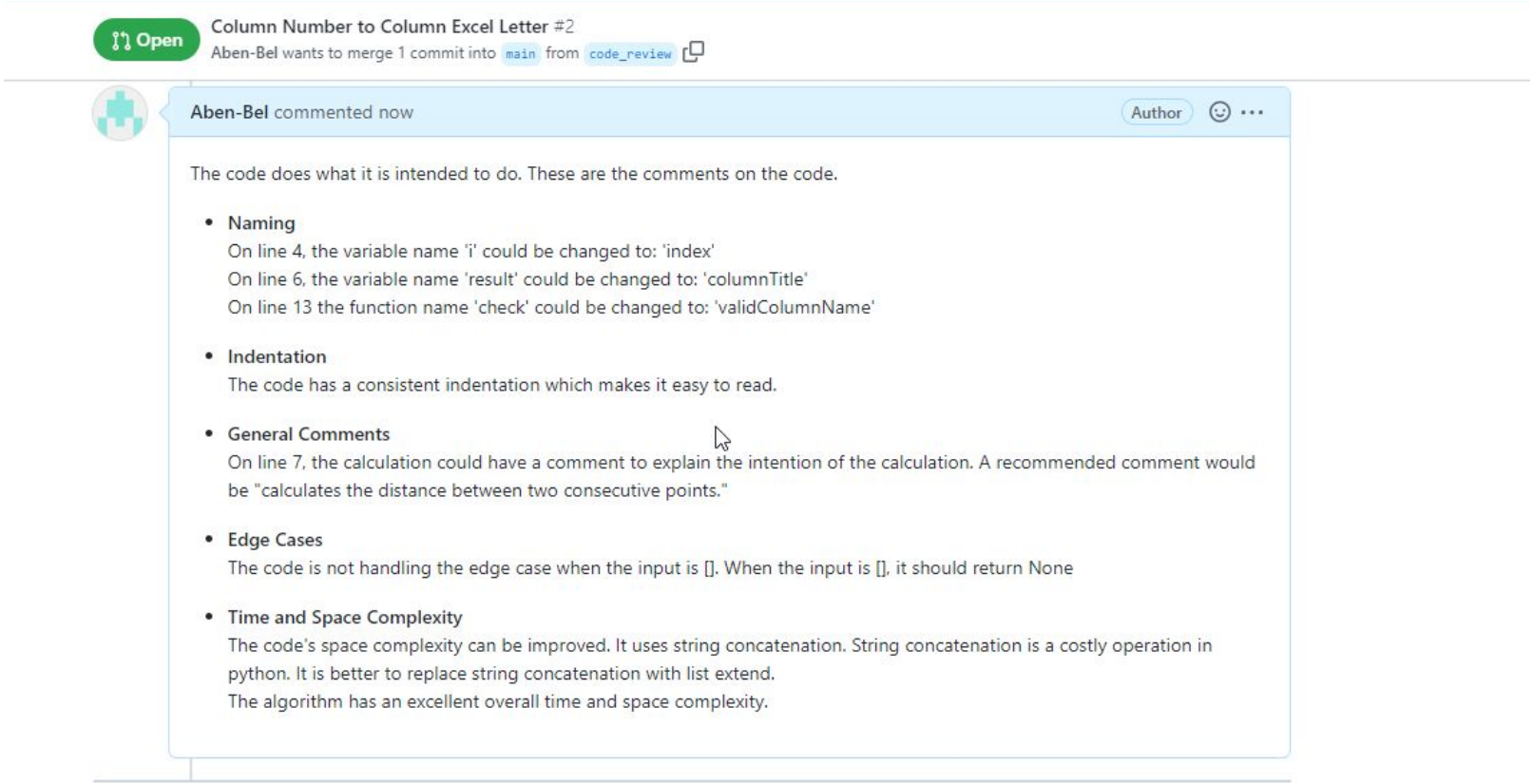
Example 1(Excel Sheet Column Title)

```
class Solution:
    def convertToTitle(self, columnNumber: int) -> str:
        result = ""
        while columnNumber > 0:
            remainder = (columnNumber-1) % 26

            result = chr(remainder + 65) + result
            columnNumber = (columnNumber-1)//26

        return result
```

- Are we the right person to review this code?
- Is it time to review this code?
- What is the problem? What does the author of this code learning?
- What is the code doing?
- Is there other way to solve the problem?
- What are the time and space complexity?

Let’s add another example

Example 2: Report



Open

Column Number to Column Excel Letter #2

Aben-Bel wants to merge 1 commit into `main` from `code_review`

Aben-Bel commented now — Author

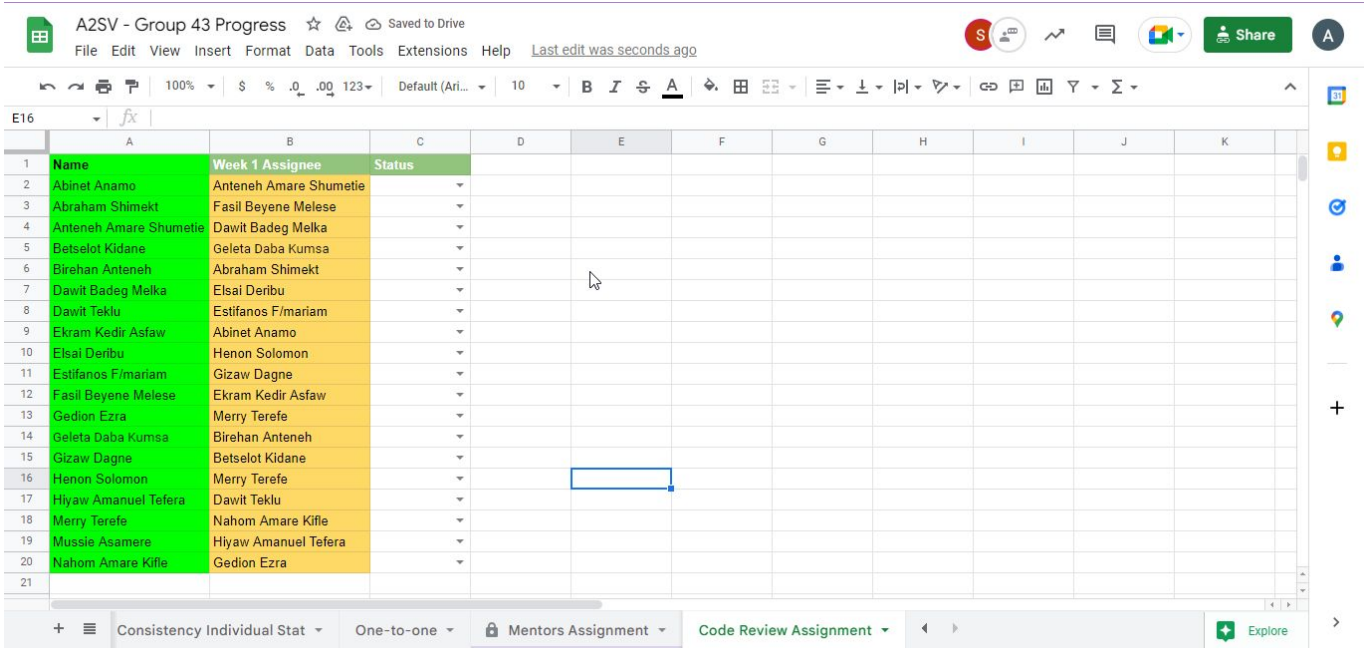
The code does what it is intended to do. These are the comments on the code.

- **Naming**
  On line 4, the variable name 'i' could be changed to: 'index'
  On line 6, the variable name 'result' could be changed to: 'columnTitle'
  On line 13 the function name 'check' could be changed to: 'validColumnName'
- **Indentation**
  The code has a consistent indentation which makes it easy to read.
- **General Comments**
  On line 7, the calculation could have a comment to explain the intention of the calculation. A recommended comment would be "calculates the distance between two consecutive points."
- **Edge Cases**
  The code is not handling the edge case when the input is []. When the input is [], it should return None
- **Time and Space Complexity**
  The code's space complexity can be improved. It uses string concatenation. String concatenation is a costly operation in python. It is better to replace string concatenation with list extend.
  The algorithm has an excellent overall time and space complexity.

Review the code below (link)

```
n = int(input())
li = input().split()
for i in range(len(li)):
    li[i] = int(li[i])

size = len(li)
c = 0
for i in range(1,size):
    flag1 = [li[j] < li[i] for j in range(i-1,-1,-1)]
    flag2 = [li[j] > li[i] for j in range(i-1,-1,-1)]
    if all(flag1):
        c += 1
    elif  all(flag2):
        c += 1
print(c)
```

The new system of reviewing peer codes



A2SV - Group 43 Progress

Saved to Drive

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

Share

E16

	A	B	C
1	Name	Week 1 Assignee	Status
2	Abinet Anamo	Anteneh Amare Shumetie	
3	Abraham Shimekt	Fasil Beyene Melese	
4	Anteneh Amare Shumetie	Dawit Badeg Melka	
5	Betselot Kidane	Geleta Daba Kumsa	
6	Birehan Anteneh	Abraham Shimekt	
7	Dawit Badeg Melka	Elsai Deribu	
8	Dawit Teklu	Estifanos F/mariam	
9	Ekram Kedir Asfaw	Abinet Anamo	
10	Elsai Deribu	Henon Solomon	
11	Estifanos F/mariam	Gizaw Dagne	
12	Fasil Beyene Melese	Ekram Kedir Asfaw	
13	Gedion Ezra	Merry Terefe	
14	Geleta Daba Kumsa	Birehan Anteneh	
15	Gizaw Dagne	Betselot Kidane	
16	Henon Solomon	Merry Terefe	
17	Hiyaw Amanuel Tefera	Dawit Teklu	
18	Merry Terefe	Nahom Amare Kifle	
19	Mussie Asamere	Hiyaw Amanuel Tefera	
20	Nahom Amare Kifle	Gedion Ezra	
21			

Consistency Individual Stat | One-to-one | Mentors Assignment | Code Review Assignment | Explore

Guidelines

- Check google sheet code review sheet
- Check who you are assigned to
- Review the person and fill the fields in the sheet


“If you cannot do great things, do small things in a great way.”
Napoleon Hill