

Understanding the Impact of Smart and Not-Smart Questions on StackOverflow

Not-Smart Question

Question: [The third option in my function is not working](#)

Summary:

The question revolves around a C++ code snippet where the third option in an `if` function is not working as expected. The user provides a code snippet but does not include a clear explanation of the issue they are facing. The tags for the question include C++,

Analysis:

This question lacks clarity and the description of the problem is very vague. The user does not specify what the "third option" in their `if` function is or what the expected behavior is. They fail to mention specific errors or behaviors that they observed, making it difficult for others to find the problem. Additionally, the user does not describe what troubleshooting steps they have already tried.

By not providing enough detail or context, it makes it more difficult for potential responders to understand the issue. This vagueness forces others to guess the problem or run the code themselves to identify the issue.

Result:

As of the time of writing, this question has not received any responses and has been downvoted. This outcome is indicative of how lack of detail and clarity can lead to poor engagement and ineffective problem-solving.

Smart Question

Question: [How to remove "http://", "https://", or "www." from string in pandas \[closed\]](#)

Summary:

This user is asking how to remove specific prefixes like ("http://", "https://", or "www.") from a column of URLs in a pandas DataFrame. The user clearly outlines the task, providing examples of input URLs and the expected output. There was no code provided in this question.

The user tagged python, panda, and replace

Analysis:

The user was clear in their question on what their task was, removing URL prefixes, and provided examples of the input and expected output. They specified the exact prefixes they wanted to remove and included a pandas DataFrame context. This aligned well with Raymonds recommendation to provide a concise and detailed problem description

Result:

At the time of writing this essay, there are 2 upvotes and 2 responses to the question, one of which has been verified and has 4 upvotes. This positive outcome demonstrates how a well-formed question can lead to efficient and effective help from the community.

Conclusion

The comparison between the two questions highlights the importance of asking smart questions. A smart question is characterized by clear problem definition, sufficient context, and specific details about the issue and any troubleshooting efforts. This approach not only helps in receiving accurate and timely assistance but also respects the time and effort of those who contribute to the community.

On the other hand, a question lacking clarity and detail can result in poor engagement and unhelpful responses. By understanding and applying the principles of smart questioning, software engineers can enhance their communication skills and improve their problem-solving effectiveness.

When I checked back on the "not smart" question, I saw that people started responding. But the lack of details in the original question made it harder for others to help quickly. If this had been a more complex and time consuming question, there may have been no responses for days. This shows why it's really important to ask smart questions from the start. Giving clear and detailed information can make it easier for others to understand and solve the problem faster.

In summary, asking smart questions is crucial for getting good help and making the open-source community work better. It helps speed up finding solutions and makes interactions more effective. This experience shows how important it is to be clear and specific when asking for help.