

# Udacity Student's Quick Forum Guide

[Quick Summary](#)

[Searching and Posting in the Forums](#)

[Has your question already been answered?](#)

[If not, create a new topic!](#)

[Asking Programming Questions](#)

[Formatting Your Posts](#)

[Provide as much context up front as possible](#)

[References](#)

Here at Udacity we love to help you learn, and our goal is for each student to become an independent learner and problem solver. While we strive to support learning in the forums through conversation and discussion, the forums also serve as a place for students to get technical help and to develop problem solving skills.

This is a guide to help you quickly orient to our forums. You will learn how to:

1. Search and post in the appropriate forum category
2. Format your posts effectively
3. Ask good questions

Here is the direct link to your forums: <https://discussions.udacity.com>

## Quick Summary

*Here are the major steps you should take to ensure your post can be efficiently answered by our staff.*

1. Search your question first in the forums. Someone might have had a similar problem.
2. Next, create a new topic in the appropriate category, for example, "Project 1" if you have a question pertaining to Project 1 in your Nanodegree.
3. Write out the problem and ask specifically what you need help with. Include any relevant links, code, possible solutions that you've looked at or tried.
  - a. If your question is in regard to a classroom quiz or video include a link to the quiz or video page.
  - b. If your subject area is programming, refer to this section for more details on how to post a [good programming question](#)



4. Write a short descriptive title that summarizes your problem
5. Post! Avoid posting the same question in multiple categories. This helps our staff keep track of which posts need attention.

*Some things to keep in mind when answering other students' questions.*

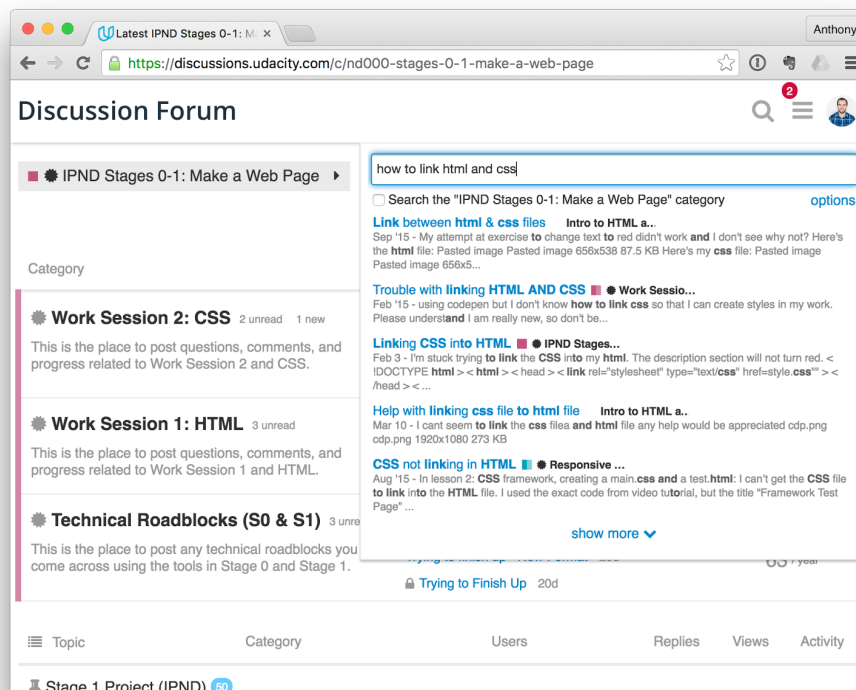
1. Try to provide answers that *lead the poster in the correct direction* rather than giving them a complete solution. Revealing your own problem-solving process can help students improve theirs. You can help students by:
  - a. Reframing the question into one they can more easily search on to find relevant information.
  - b. Providing a link to a resource in the Udacity classroom, Udacity forums, MDN, StackOverflow or elsewhere that discusses the issue. Sources that provide examples are especially helpful.
2. Never post a project solution on the forums unless specifically asked to do so in the project instructions.

# Detailed Steps

## Searching and Posting in the Forums

### Has your question already been answered?

The Udacity forums are filled with helpful information! You can search the forums by clicking on the search icon (magnifying glass) in the upper right hand corner of the forums homepage. Type some keywords about your question and read some of the posts that come up. If you can't find an answer, then you can create a topic for your question! (see the next section)



### If not, create a new topic!

- Find the right category in which to post.**  
Take a few moments to find the relevant category and subcategory in which to post your question. If they exist, subcategories will be organized by supporting course lesson or by broader topics within the course. Finding the right category will ensure that the right viewers see your topic, and also allow other students to find the topic later.
- Create a topic and introduce the problem with a brief description.**  
To provide context for your question, make sure to give a brief, problem focused description of the

problem you are having before posting anything else. Here you can include links to any other topics that you think might be related, or other topics with solutions that did not work.

3. **Write a title that summarizes your problem.**

Once you have found the right category for your topic and carefully crafted the body, it is time to write a title! The title should be written last so you can provide a focused description for the body that you have just written. For examples of good titles and bad, see the [StackOverflow how-to-ask](#) page.

4. **Post! But only once for a given question.**

Once you post, you can sit back and wait for others to comment. If you are enrolled in a Nanodegree program, our staff will try to respond within 24 hours, although the typical response time is much faster. Note that posting several closely related topics will make it more difficult for Coaches, Mentors and other students to determine which questions have been addressed and which still need to be answered.

## Asking Programming Questions

If you're studying a programming field such as Java, Python, Android, etc, follow these additional guidelines to help scope your problem more effectively. This will dramatically increase your chances of getting the answers to your question quickly.

### Formatting Your Posts

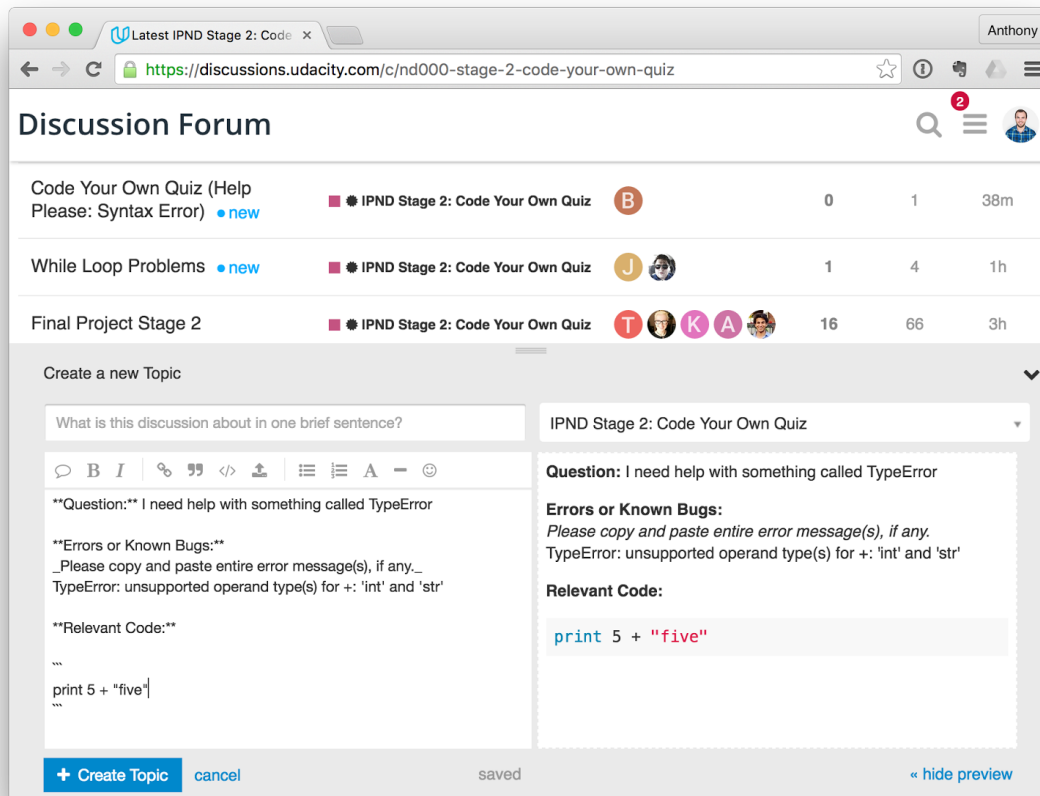
To help others understand your code, it's important to format your post in a way that enhances its readability. You can use [Markdown](#) syntax to style text or, more importantly, to format your code. Formatting your code with Markdown will ensure that indentation stays intact, and will also add syntax highlighting to your post. To enable this do the following:

```
...
```

Paste your code here

```
...
```

That character is the backtick, which is found in the top left of most keyboards (note it is **not** the apostrophe). On the right you'll see a preview of your post. If your code still doesn't look formatted correctly, try adding a newline before the opening and after the closing backticks, and ensure there is no unnecessary whitespace after the backticks. Some forums may have this structure ready for you to use in a new post already!



## Provide as much context up front as possible

1. **Include all code needed to understand the problem.** Coaches, Mentors, and other students will often run your code locally while debugging. If only a small snippet of code is provided, this may restrict others to just a visual inspection of the code. Note that if you do post sensitive code (e.g. project or homework code), it is a good idea to [use a spoiler tag](#) either in the title of your topic or spoiler tags in the body.
2. **Narrow the code down to a minimal working example.** While the previous point encouraged you to provide enough code to understand the question, it is also possible to provide too much code. You can help others by providing only the bare minimum amount of code needed to understand the problem you are having. By doing this, you will also be developing your ability to locate and focus on the root of a problem, which will help you to grow in your problem solving ability. For more information, see [the StackOverflow page on minimal working examples](#).
3. **Include the full traceback of any errors you are seeing.** This is crucial for helping others to understand where the problem lies. The traceback should include any errors thrown, the line numbers of your code on which these error occurred, and usually a description or code snippet.
4. **Comment your code.** While you might be familiar with the minute details of your code after having spent hours working on it, the code will be completely unfamiliar to others looking at it for the first

time. Including comments throughout the code explaining why each part of the code is there will help others get up to speed quickly.

## References

- [StackOverflow: How-to-ask](#)
- [StackOverflow: minimal working examples](#)
- [Writing the perfect question](#)
- [How do I ask and answer homework questions?](#)
- [How to debug small programs](#)
- [Meta discussions on asking questions](#)
- [How to ask questions the smart way](#)