Hi Everyone,

Welcome to MATH2504, I hope you're all looking forward to learning a lot over the coming semester about programming, which in our case will be within the context of Julia. Inevitably, you will likely have numerous questions throughout this course that you can ask here on the Ed Discussion Board. Asking questions of your peers—and, in our case, the teaching staff—is something done by even professional programmers on sites like StackOverflow and Discourse. But, not all questions are created equal! The ability to ask good questions is in fact an important skill that you will develop during this course. So, what makes a question good? Here, I will summarise some of the most important points from other guides (see, for instance, this and this). Also, please take a look at Yoni's example question.

You should try and follow these guidelines wherever possible; we might ask you to modify/improve your question in the interest of providing a better answer for you and other students.

1. **Search** for your question (and similar questions) before posting to avoid creating duplicates. It is helpful if you can provide links to similar questions, course material, or other resources to provide context.
2. **Avoid posting private questions** because this means that only you get to benefit from the answer; it is likely that other students have the same question! You can always use the anonymous posting feature.
3. **Create a good title** that accurately summaries your question or issue. Ideally, someone scanning the sidebar should be able to figure out the general idea of your question. For instance, "What is the difference between `bitstring` and `string`?" is a better title than simply "Confusion about strings".
4. **Present your code using code snippets** (see the toolbar) to ensure that it has syntax highlighting and can be easily copied. You should also try and make sure that your code is neatly styled. **Do not post your code as a screenshot** since this makes it more difficult to answer your question.
5. **Present your math using LaTeX** (see the toolbar) to make it pretty and easy to read.
6. **Try to present a Minimal Working Example (MWE)** that can be easily replicated by anyone and does not include too much unnecessary material. This also means that you should mention what packages you are using (if any). Try to, where possible, "simplify your code to the smallest piece of code that still shows your problem" (quoted from here). See, for instance, this post on StackOverflow.
7. **Include the entire stack trace** (everything printed by Julia alongside an error message) **in a code fence** (wrapped with three backticks, as in ```code```) when asking about unexpected errors. Often, although these can be difficult to read, the clues necessary to answer your question lie somewhere in this stack trace.
8. **Proofread** your question before posting to make sure there aren't any obvious spelling/grammar/formatting/technical mistakes, especially those that might impact the understanding of your question.

I hope that these above points help you throughout the course. Remember that, by making your question easier to answer, you will often get a faster and better quality answer in return.

Best,

Thomas Graham

Coordinating Tutor, MATH2504 2022