

“Wild” Question Modification Checklist

Enter...If You Dare!

This checklist will guide you through the basic things that you will need to do in most cases if you are modifying a problem. It's divided up into sections based on the different sections that are present in most .pg files.

Preamble

- Change the description, subject, date, institution, author, keywords, etc. in the .pg file to match the question. This is important, because it makes it easier to search for your problems in the WeBWorK library.
- Check that the first uncommented line in the file is `DOCUMENT()` ;
- Make sure that the `loadMacros` function is at minimum loading the `PGstandard.pl`, `MathObjects.pl`, and `PGML.pl` macros.
- Load any additional macros needed for the problem. Make sure that each macro name is enclosed in double quotes and followed by a comma.
- After the `loadMacros` function, check for the line `TEXT(beginproblem())` ;

Setup

- Set the Context to be whichever is necessary for the problem. If there are any variables that will be used in the problem, add these to the Context now. Make sure to set their types!
- Initialize all MathObjects that you will be using in the problem. This includes constant values, formulas, random values, etc.
- Change the Context as needed to define different MathObjects for the problem. Remember to change the Context back to the ones containing your variables if you need to use them! However, you do not need to change the Context to access MathObjects; they store the Context they were created in, so an Interval MathObject will still be treated as an Interval, even if the Context is switched to Complex.
- Initialize the MathObject that will be used to contain the answer.

Main Text

- Use the `BEGIN_TEXT / BEGIN_PGML` and `END_TEXT / END_PGML` tags to surround the problem text. Enter any text, equations, etc. that you need to describe the problem.
- Add an answer blank. If using PG, enter `ans_rule(int)`, where `int` is some integer that determines how many characters wide the answer blank is. If using PGML, use

[_____], where each underscore represents one character of width for the answer blank.

Answer Evaluation

- ❑ Determine if you want to give students feedback on the correctness of their answers if they don't get the entire question correct. Enter `$showPartialCorrectAnswers = (1 or 0, depending on if you want to give feedback or not, respectively.)`
- ❑ Add an answer checker. See the [Answers section](#) for more information on how to do this. In general, though, this will just be `ANS($answer->cmp());`.

Solution

- ❑ Add your solution text (if necessary). Remember to use the `BEGIN_SOLUTION / BEGIN_PGML_SOLUTION` and `END_SOLUTION / END_PGML_SOLUTION` tags to surround the text.

Closing

- ❑ Add a file comment if necessary. In many cases, this will just be something to tell people what “add-ins” are used in the question (e.g. PGML, GeoGebra, etc.). Enter `COMMENT('comment');`
- ❑ Check that the last uncommented line in the file is `ENDDOCUMENT();`