

## Contents

<b>About</b>	<b>1</b>
<b>For the User</b>	<b>2</b>
Questions . . . . .	2
On Numbering . . . . .	2
Question . . . . .	2
Problem . . . . .	2
Exercise . . . . .	3
Exploration . . . . .	3
Answers . . . . .	3
Short Answer . . . . .	3
Multiple Choice . . . . .	4
Free Response . . . . .	5
Shuffle . . . . .	5
Grading . . . . .	6
<b>For the Developer</b>	<b>6</b>
Question Environments . . . . .	6
To add a new form of question . . . . .	6
The problemEnvironConfig command . . . . .	7
On Numbering . . . . .	7
Shuffle . . . . .	7
Answer Environments . . . . .	8
<b>About this Documentation</b>	<b>8</b>
Who to ask . . . . .	8
Information Still Needed . . . . .	8

## About

The below is documentation on how to use (and create) environments for questions and responses. For an excellent interactive example page for most of these

concepts, go to here: <https://ximera.osu.edu/course/kisonecat/ximeraSample/sample>

To contribute to this documentation and/or see who else is contributing reference the last section.

## For the User

This content is for the person writing the latex code. It contains information regarding the environments currently implemented. If you wish to add more (or tweak the code) see the "For the developer" section below.

### Questions

This is for the various "question" environments. They are currently all the same, but have separate sections in case this changes.

#### On Numbering

Numbering for all question environment types are still numbered sequentially (hence, if you compile this file, you can see that you get Question 1 then Problem 2, then Exercise 3, and finally Exploration 4).

The main exception to this is the shuffle environment, which has different behavior depending on if the file type is a handout or not. See Shuffle for specifics.

#### Question

The Question environment is typed as an environment and has the prompt "Question" as it's title.

**Question 1** *Here is a question. There is currently just a question here, but nothing else.* `test`

---

#### Problem

The Problem environment is typed as an environment and has the prompt "Problem" as it's title.

**Problem 2** *Here is a question. There is currently just a question here, but nothing else.* `test`

---

### Exercise

The exercise environment is typed as an environment and has the prompt "Exercise" as its title.

**Exercise 3** *Here is a question. There is currently just a question here, but nothing else.* `test`

---

### Exploration

The exploration environment is typed as an environment and has the prompt "Exploration" as its title.

**Exploration 4** *Here is a question. There is currently just a question here, but nothing else.* `test`

---

### Answers

#### Short Answer

In order to have a short answer you can use the command "answer" with a single argument and an optional argument. The required argument is the desired answer, whereas the optional argument is tolerance for numeric problems.

Thus we can use the following

**Problem 5** *What is  $2 + 2$ ?* `4`

---

#### Optional Arguments:

- "tolerance":  
For the engineers among us we can use the tolerance factor;

**Problem 6** *What is  $\pi$ ?* `3.14`

---

The above is 3.14 with a tolerance of 0.14. This means that we can take the answer 3, but we can't take the answer 2.99 or 3.29. Note that, if we simply wrote  $\pi$  it would still work.

- "given"

An optional argument to the "answer" is "given". In handout mode, the "given" optional argument will actually fill in the answer, rather than leaving it blank. Thus the command;

**Question 7** *The answer to  $2 + 2$  is*   
given

### Multiple Choice

To answer with multiple choice, we can use the multipleChoice environment. Keep in mind that (currently) there is no way to put a hard limit on number of guesses, but there are some metrics that could (theoretically) take number of guesses into account.

You can use an optional argument to mark any given answer correct. Note that this means that multipleChoice can be used for the "choose all that are correct" style questions.

For example:

**Problem 8** *Which of these is the right answer?*

**Multiple Choice:**

- (a) *This one?*
- (b) *Maybe this one?*
- (c) *Definitely this one ✓*

**Problem 9** *Which of these is the right answer? (Mark all that are correct)*

**Multiple Choice:**

- (a) *Definitely not this one?*
- (b) *Maybe this one? Not so much.*
- (c) *Definitely this one. ✓*
- (d) *And this one. ✓*
- (e) *Probably this one too. ✓*

### Free Response

There is a free response answer style that saves the content verbatim into the database (somewhere) but does not parse or grade in any manner. Thus it must be looked over individually by an instructor/grader to assign grades.

Needs more details added (by someone that has actually used this).

### Shuffle

The shuffle environment will take a list of question environment types and shuffle their order and serve up the optional argument number of them (1 by default). If the optional number is higher than the number of questions, shuffle will automatically list all the questions (without blank questions).

So for example, the below has no argument, and so it will take one of the question environments within it at random and display it. Recompile several times to see the change.

**Problem 10.1** *This is my first problem!*

---

In contrast, the below shuffle has an optional argument of 3, so it will choose 3 questions and order them randomly.

**Exercise 11.1** *This is my second problem!*

---

**Problem 11.2** *This is my first problem!*

---

**Exploration 11.3** *This is my third problem!*

---

Note that shuffle is taking place at the latex level. This means that, although when compiled by xake the ordering will be random, the ordering will **not** be random by student. Thus each student will see identical assignments.

#### Numbering:

Notice that the numbering may change depending on if the document is labeled as a "handout". Without the "handout" tag on the documentclass, shuffle uses a sub-numbering system. Thus for the second shuffle environment it is question 10, and since there are 3 elements, they are 10.1, 10.2, 10.3.

**Known issues:**

- (a) The shuffle environment without handout mode will always use sub numbering, even if there is only 1 question (see the first shuffle environment). This may or may not be a bug depending on how we eventually decide to implement the "give me another" feature. For instance, the extra decimal is a good indicator that another is available, even if it is not immediately displayed.

## Grading

To be added... hopefully.

## For the Developer

Below is more detailed information on how the environments are coded and how to add new versions in some cases.

## Question Environments

The question environments are built using an underlying command called `problemEnvironConfig` and the `NewEnviron` package.

### To add a new form of question

In order to add a new form of a question environment, simply define the question using the `problemEnvironConfig` command. For example you can use; (Commented out for compile-ability)

This would form a new environment called 'newquestion' which would display "New-Question" as it's label (in place of something like Problem).

You should even be able to do this in a document directly (although you may want to at least place it in a university-specific style file).

**New-Question 12** *This is a newquestion style question! It was just made!*

---

Notice that the numbering and formatting are the same as other problems. This will run the risk that this is in latex only. It may need additional configuration on the tex-4ht and/or javascript levels.

The optional argument is a backward compatibility feature from the original problem, exercise, question, and exploration.

### **The problemEnvironConfig command**

Currently the problemEnvironConfig command uses the previously used begin/end config commands from before the switch to NewEnviron. This could likely be removed if desired and rewritten into one large config command.

The first part of the command is designed to account for the shuffle situation;

This defines a global command based on which problem in shuffle, and which shuffle, the problem is in. The global command ContentProblem (with concatenated numerals) contains the entire body of the environment (which is the BODY command), and the global problem ContentProblemType (with concatenated numerals) contains the intended problem environment (problem, question, exploration, etc). Then it steps the shuffle@trackProblem (which keeps track of which problem in that specific shuffle you are on). The shuffle environment's end stage will actually shuffle and display the problems.

If it is not in shuffle, the problemEnvironConfig just outputs the problem as it has been typed.

## **On Numbering**

The numbering for problems is somewhat complex as it keeps track of a number of aspects of the problems. The counters of interest are: "problem" and "problemType". "problem" tracks the problem on the page, and "problemType" tracks subproblem numbers, like problems within shuffle. This will likely be supplanted, as this is mostly legacy code from before the different problem environments were universalized into one general routine.

## **Shuffle**

See above for how the problem environments are recorded to allow shuffle. The shuffle environment itself has two stages;

In the begin section of shuffle, it records and sets useful counter info. The counter shuffle@maxCount records the number of questions we want out (we step this for the inequality logic checks). The counter shuffle@count keeps track of which shuffle environment we are on, inside the activity. e.g. "This is the 3rd shuffle on the page". This is used to keep different shuffles' content separate.

The second phase (the end section) of shuffle permutes a vector of numbers (from 1 to the number of problems), and then pulls the number of desired problems in order from the permuted vector. So if the permuted vector permutes from 1-10, to:

## *Question and Answer environments*

1, 2, 5, 10, 9, 8, 3, 7, 4, 6

And we wanted 3 problems, we would pull problems 1, 2, and 5 (where 1, 2, and 5 are the problems in their original order).

## **Answer Environments**

## **About this Documentation**

### **Who to ask**

The question environment and shuffle info in this file largely added by:  
Jason Nowell, JNowell@ufl.edu

### **Information Still Needed**

Most of the answer environment info is still needed. Only skeletal information is currently included.