wordChoice

In-line multiple Choice dropdown box.

How word choice works

The intended point of word choice is to make an "in-line" version of multiple-choice, although in practice it can be a little more flexible and useful. Note that it generates a dropdown menu instead of a list of choices, to allow for in-line formatting.

How to generate word choice answers

To generate word choice answers, you use the wordchoice command:

```
\wordChoice{
    \choice{$($}
    \choice[correct]{$[$}
    \choice{$)$}
    \choice{$]$}
    }
```

Which generates: $((/ [\checkmark /)/])$

Notice that this is formatted almost identically to the multipleChoice environment - the main difference being that it is a command rather than an environment to ensure that you don't force a line break (since wordchoice is intended to be an in-line version of the multiple choice answer type). Also note that, as long as you don't use the line break latex command: \\or include a double line break yourself (enter twice) in the code, the wordChoice will command itself won't force a line break, allowing you to chain together a number of these to create a sequence of dropdown boxes that allow you to make more elaborate answers. See the tile on letting students input intervals as answers in this documentation for an example.

How word choice validates an answer

Simply. Word choice allows the student to click on (only) one answer choice, and if that answer is marked as correct (has the "[correct]" optional argument),

---

Learning outcomes:

then they get credit for the problem, otherwise it is marked wrong.

Note that, this means if you have multiple choices flagged as correct, then if a student selects *any* answer that is marked correct, they will get the answer marked correct - which also locks the problem, so they won't be able to check other options for correctness.

Optional Arguments

Word Choice itself (currently) has no optional arguments. The choice command (currently) has only one key-value pair, which is the flag for whether the choice is correct or not. Technically the key is 'correct' and the value is 'correct' or 'incorrect', but the key defaults to 'incorrect' and simply using the key flips it to correct. In other words using `\choice[correct]{Correct!}` is the same as using `\choice[correct=true]{Correct!}`, and both of those will mark the given choice as the correct answer. Likewise `\choice{False!}` and `\choice[correct=false]{False!}` are equivalent, and both result in the answer being incorrect.

Potential Pitfalls and Problems

Only one answer can be selected

Since only one answer can be selected then having multiple 'correct' answers in the list means a student can only select one of them. Since Xronos locks the answer once it is correct, the only way to test other answers for correctness is to "erase work" and start over with a different selection. Since student's won't do this in practice, and since they expect that only one answer being selectable means there is only one correct answer - this can lead to the assumption that since they got "the" correct answer, then all other answers **must be** incorrect, which may result in questions as to why the other (actually correct) answers aren't correct - or even worse, the blanket assumption that they must be incorrect without any questions or opportunities to correct that misunderstanding.

There are a finite number of options

Since Xronos allows unlimited attempts at answering (this is a feature we intend to make controllable in some way in the future, but currently all answers get unlimited attempts), a clever-enough-to-be-dangerous student will realize they can just spam the answer until they find the right one and move on without even reading the options. There isn't much to do about this, but it is important to keep in mind when designing problems and deciding on answer-type variety.

Randomization is 'kind of' pointless

Although there are ways to randomize the content of the problems and answers (see the randomization section), it is important to note that LaTeX level content *cannot* be randomized currently (this is going to be changed in the future edition) which means, although the content of the choice command might be randomized, *which* multiple choice option is correct *cannot* be randomized - something that the students will pick up on mighty quick. For review this isn't *as bad* since students are at least aware enough that they are trying to understand how the correct answer is the correct answer - so knowing "the right answer is the third choice" is *less* important... but it's still important.