Quick Quiz Wiki Markup

Introduction

At its simplest, a quiz starts with a title and is followed by any number of questions. Each question is followed by any number of answers. Here is a simple quiz:

```
~~
CS100 Quiz: Do you like me?

?? Do you like your introductory CS class?
    &&^ yes
    &&  no, but it's not the classs, it's me
    &&  maybe so
    &&  wut? I'm in a CS class?
```

The `~~` (two tildes) at the beginning of the first line signifies a title. All quizzes must start with a title. The `??` (question marks) at the beginning of the third line signifies the start of a question. The subsequent lines starting with `&&` (ampersands) signify possible answers to the previous question. This quiz might render as:

CS100 Quiz: Do you like me?

1. Do you like your introductory CS class?
   (A) wut? I'm in a CS class?
   (B) no, but it's not the classs, it's me
   (C) yes
   (D) maybe so

Note that the answers have been randomized.

Structure tags

Tags are used to structure the parts of the quiz. Structural tags must be at the beginning of a line, not counting whitespace. For example, the tag ?? marks the beginning of a question. The text that follows the question marks is considered part of the question until some other structural tag is encountered.

Here is a table of the character tags and their meanings when they begin a line:

tag	*meaning*
`~~`	title
`%%`	section heading
`**`	add some notes before a question or section
`??`	start a question
`&&`	start an answer
`&&^`	start a correct answer
`{{`	a sequence of related questions, one will be randomly selected
`{{3`	like `{{` but three questions will be randomly selected, for example
`}}`	end of a sequence of related questions
`::`	include a file (like `::more.stuff`)
`^^`	generic end tag

In addition to signifying the start of a structure, a structure tag also signifies the end of the previous structure. The generic end tag is used when you wish to end a structure, such as a question, before the next structural tag is encountered.

Structure tags must begin a line; otherwise they are interpreted as text or formatting tags. To supress interpretation of a structure tag, escape the tag with a backslash For example, to start a line with what looks like a title tag, but not have it interpreted as a title tag, you would do something like this:

```
?? Does
\??
start a question?
    && yes
    && no
    && maybe so
```

This question would render as:

2. Does ?? start a question?
   (A) maybe so
   (B) yes
   (C) no

Quick Quiz Grammar

A quiz takes three forms. The first we have seen, a title followed by one or more questions. Another form is a title followed by a one or more sections. Each section starts with a section header, as in:

```
~~ Title

%% Section header 1

?? Question 1

?? Question 2

%% Section header 2

?? Question 3

?? Question 4
```

Section headers appear in the HTML renderings of the quiz, but are removed from the PDF versions (which are intended to be assessment vehicles). The final form of a quiz is simply some marked up text, with no questions:

```
**
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla tincidunt
turpis et quam ornare vestibulum. Proin et dolor sem. Integer metus
metus, bibendum vel iaculis a, pulvinar at dolor. Sed nulla lacus,
faucibus ut dolor et, auctor sollicitudin tortor.

Mauris mi nulla, lobortis fringilla cursus sit amet, volutpat sit amet
enim. Proin a lorem non tellus molestie vehicula et sed purus. Aenean
rutrum imperdiet viverra. Sed pharetra neque lectus, a sodales magna
hendrerit sit amet.
```

You can also intersperse notes and questions. Notes appear in both the HTML and PDF renderings.

```
~~ Title

%% Section header 1

** Notes for this section

?? Question 1

?? Question 2

%% Section header 2

** Notes for question 3

?? Question 3
```

```
** Notes for question 4

?? Question 4
```

Any where a question can go, you can place a sequence of questions from which some number of them will be randomly selected. For example:

```
 {{2
 ?? Question A

 ?? Question B

 ?? Question C
 }}
```

that two of the three questions would be selected when rendered as a PDF assessment tool. If the number following the opening braces is omitted, one questions will be randomly selected.
You can include marked-up text from another file with the double colon structure tag. Here are some examples:

```
::base.q                      !! base.q should be in the current directory
::../intro.q                  !! intro.q should be in the parent directory
::/home/me/concepts/header.q  !! header.q should be in the specified directory
```

Note that an exclamation mark, except in verbatim sections is the comment character for the system. There should be no spaces between the :: and the start of the filename or path.

Blank Lines

Blank lines within a structure are (mostly) honored. The first blank in the mark up line causes a line feed. Two blank lines in a row is rendered as a single blank line. Three blank lines in a row in the markup render as two blank lines, and so on. If you wish to have a large number of blank lines, use the vertical space command (for example, `;;9` at the beginning of a line for nine blank lines) instead. Note: blank lines are ignored between structure commands. Thus:

```
~~ A Good Quiz
?? Do you like this question?
```

renders the same as:

```
~~ A Good Quiz
~~



?? Do you like this question?
```

whereas:

```
~~ A Good Quiz



?? Do you like this question?
```

renders a blank line between the title and the question.

Formatting tags

Within a structure, one can use tags to markup text for formatting purposes:

tag	*meaning*
`''`	start (and end) italics
`'''`	start (and end) bold
`'''''`	start (and end) bold italics
`@@`	start (and end) verbatim monospaced (typewriter) font
`,,`	suppress a space
`;;`	add a blank line
`;;3`	add three blank lines, for example
`$`	start (and end) an equation
`*`	signifies an unordered list item, when at the beginning of a line
`#`	signifies an ordered list item, when at the beginning of a line
`{{`	start a pick-list, when not at the beginning of a line
`}}`	end a pick-list, when not at the beginning of a line
`+++`	start (and end) a section that is processed by a drawing program

To place a word or phrase in italics/bold/bold-italics/typewriter font, do something like this:

```
''I'' '''really''' '''''like''''' @@CS@@!
```

which renders as: *I* **really** ***like*** `CS`!

The text between @@ tags is verbatim, with a couple of exceptions discussed below. If you want multiple lines of verbatim text, such when showing some program code, use @@ tags; the @@ tags should appear by themselves on a line, with no spaces prior or after. Here's an example code listing:

```
@@
(define (fact n)
    (if (= n 0) 1 (* n (fact (- n 1))))
    )
@@
```

This renders as:

```
(define (fact n)
    (if (= n 0) 1 (* n (fact (- n 1))))
    )
```

Verbatim text is *mostly* verbatim. To embed @@ within a verbatim section, which would otherwise end the verbatim section, escape it with a backslash: \@@. The two *at* symbols will show up in the verbatim section; the backslash won't. The other exception is a pick list, which is discussed in a later section.

The Wiki system attempts to honor the spacing in your marked up text. However, mulitple spaces in a row are ignored (just like HTML and LaTeX). If the system omits a single space where it should not, you can add the space with the `<sp/>` tag. If the system emits a space where it should not, you can remove the space with `,,` markup. Here's an example:

```
Try '''bold''',,''italics''!
```

which renders as: Try **bold***italics*! You can force multiple spaces in a row by using blank spaces with `@@` tags or by using the special character `&nbsp;`.

You can add blank lines manually or you can create a bunch of them with the double semicoln. For example, the tag `;;13`, which needs to be on a line by itself to work, generates 13 blank lines.

Mathematical equations

Mathematical equations are delineated by dollar signs and use LaTeX notation. The system uses the *MathToWeb* package to convert LaTeX mathematical notation to HTML5 notation. Here is an example of an question using an equation:

```
?? '''T''' or '''F''': The Pythagorean Equation has the form:
$a^2 = b^2 + c^2$.
```

This would render as:

3. **T** or **F**: The Pythagorean Equation has the form: $a^2 = b^2 + c^2$.

Basic equations should work fine, but see the note about pick lists as exponents in the pick list section of this document. If you have other problems getting an equation to work, you can find the MathToWeb users guide at: `http://www.mathtoweb.com/cgi-bin/mathtow`

Lists

To add an unordered (itemized) list, use asterisks, ending the list with a double circumflex:

```
?? Which emotions do you feel most often when programming?
    * frustration
    * jubilation
    * ennui
    ^^
Choose two.
    &&  frustration and jubilation
    &&  frustration and ennui
    &&  jubilation and ennui
    &&  I refuse to choose!
```

This would render as:

4. Which emotions do you feel most often when programming?

   - frustration
   - jubilation
   - ennui

   Choose two.

   (A) jubilation and ennui

   (B) frustration and ennui

   (C) frustration and jubilation

   (D) I refuse to choose!

To add an ordered (enumerated) list, use sharp signs, again ending the list with a double circumflex:

```
?? Which emotions do you feel most often when programming?
    # frustration
    # jubilation
    # ennui
    ^^
Choose two.
    &&  ''i'' and ''ii''
    &&  ''i'' and ''iii''
    &&  ''ii'' and ''iii''
    &&  I can't choose just two!
```

This would render as:

5. Which emotions do you feel most often when programming?

   (i) frustration

   (ii) jubilation

   (iii) ennui

   Choose two.

   (A) I can’t choose just two!

   (B) *ii* and *iii*

   (C) *i* and *ii*

   (D) *i* and *iii*

Ordered lists are rendered with lower-case roman numerals. Note, you cannot nest lists.

Pick lists

A pick list is a method to add alternatives to a question. Consider the question:

```
?? Is {{.red.green}} your favorite color?
```

Here, the question has two versions, one which asks if red is your favorite color and one which green is queried. A pick list starts with {{, followed by the separation character. In the example, the separation character is a period (full stop). The separation character can be varied; choose a character that does not appear in any of the alternatives. As a default, the example would get rendered as:

6. Is red your favorite color?

However, there is a switch that allows the alternative to be expressed. You can request that alternatives be generated when you render the quizzes into PDFs. You will be sent, by email, two version of the PDF, one with alternative changes highlighted and one suitable to be given as an assessment. If the pick list occurs in a verbatim listing, the entire listing will be highlighted. If the pick list occurs in an equation, the entire equation will be highlighted. Pick lists also work in drawings, but the changes are not highlighted as of this moment.

You can place a pick list in verbatim text and in equations. If you wish to place a literal {{ or }} in those regions, you must escape the braces, as in:

```
 @@
 function f(x) { while (x > 0) { print x; --x; \}}
 @@
```

or

```
$ \frac{1}{\sqrt{2\}} $
```

Of course, the latter could be safely entered as:

```
$ \frac{1}{\sqrt{2} } $
```

Both of these render as $\frac{1}{\sqrt{2}}$. Do not place any formatting markup within the pick list. To italicize the alternatives, for example, do this:

```
?? Is ''{{.red.green}}'' your favorite color?
```

not this:

```
?? Is {{.''red''.''green''}} your favorite color?
```

It appears *MathToWeb* has at least one glitch: you cannot use a pick list as an exponent. Thus:

```
$n^{{.2.3}}$
```

fails to render as n^2 or n^3. A work-around is to use a set of braces to enclose the pick list:

```
$n^{ {{.2.3}} }$
```

The space after the intial open brace and before the final closing brace is required. This correctly renders as: n^2

Drawings and Images

Here is an example use of the drawing markup:

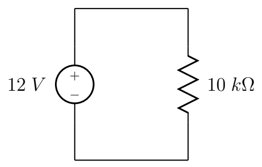
```
?? How many volts drop across the resistor?
+++
    (include "drawEE.scm")
    (battery up 12 volts polarity)
    (wire right 3)
    (resistor down 10 kiloohms)
    (wire left 3)
+++
```

What happens here is that the lines between the +++ markers specify a Scam program (*Scam is a Scheme-like programming language*). This program is run, generating a PNG (for the HTML rendering) and a PIC (for the LaTeX/PDF) rendering. The drawing program is specified with the include:

```
(include "drawEE.scm")
```

This particular drawing program uses a turtle graphics style of drawing; the example renders as

7. How many volts drop across the resistor?



There are other drawing programs available, include a CS package (email *lusth@cs.ua.edu for details*).

To include an arbitrary image, use the `<image>` tag. For example, the specification:

```
<indent>
<image>qq-small.jpg</image>
</indent>
```

renders as:

You should not place images in-line. Instead, set them off as a separate paragraph. Only .jpg and .png images are supported. Scaling of images is not supported at this time.

XML-style markup

Here is a list of XML-style markup tags understood by the system:

tag	*meaning*
`<line>`	draw a horizontal line, end with `</line>`
`<center>`	center text and images, end with `</center>`
`<indent>`	indent text and images, end with `</indent>`
`<bigger>`	use a bigger font, end with `</bigger>`
`<smaller>`	use a smaller font, end with `</smaller>`
`<only>`	conditional content, end with `</only>`
`<link>`	add a verbatim hypertext link, end with `</link>`
`<url>`	add text linked to a url, end with `</url>`

Here are some examples of the above tags:

```
<line>100</line>
<center>I feel like I'm in the middle!</center>
<indent>I feel like moving in a little.</indent>
<bigger>I want to be big</bigger>
<bigger><bigger>I want to be huge</bigger></bigger>
<smaller>I want to be smaller than normal</smaller>
<smaller><bigger>I want to be no different</bigger></smaller>
<only latex>This will appear only in the PDF</only> [missing text in html and forms]
<only latex html form>This will appear everywhere</only>
<link>http://beastie.cs.ua.edu/concepts</link>
<url http://beastie.cs.ua.edu/concepts>Concept Inventories!</url>
```

These render as:

I feel like I'm in the middle!

I feel like moving in a little.

I want to be big
I want to be huge
I want to be smaller than normal
I want to be no different
This will appear only in the PDF [missing text in html and forms]
This will appear everywhere
`http://beastie.cs.ua.edu/concepts`
Concept Inventories!

Note that the `<indent>` tag generates space above and below, so is mostly suitable for whole paragraphs and tables.

In addition, because some symbols have special meaning to both HTML renderers and LaTeX, you should use these XML tags instead of the raw characters in non-verbatim parts of a document:

tag	*character*
`<sharp/>`	`#`
`<dollar/>`	`$`
`<underscore/>`	`_`
`<percent/>`	`%`
`<ampersand/>`	`&`
`<tilde/>`	`~`
`<caret/>`	`^`
`<obrace/>`	`{`
`<cbrace/>`	`}`
`<backslash/>`	`\`
`<lt/>`	`<`
`<gt/>`	`<`

Within a verbatim section, you should **not** use these tags, as they will appear literally. So, do this:

```
@@
make the font bigger with <bigger>...</bigger>
@@
```

not this:

```
@@
make the font bigger with <lt/>bigger<gt/>...<lt/>/bigger<gt/>
@@
```

You can use some special HTML symbols, the named ones and the numeric (decimal) ones. For example, the Euro currency symbol `&euro;` renders as €, while the symbol `&#8251;` renders as ※. You can find a table of decimal HTML symbols at `http://www.tamasoft.co.jp/en/general-info/unicode-decimal.html`. The wiki system adds characters in the vein of `&Abar;` and `&cbar;` which render as $\bar{A}$ and $\bar{c}$, respectively. Note that the bars are at the same height regardless of case.

Not every unicode symbol can currently be rendered in the PDF. Instead, you will see a box with a question mark inside (or maybe a penguin). Please check carefully when using numeric symbols.

Tables

One starts and ends a table with the `{|` and `|}` tags, respectively. Within a table, one starts and ends a row with a vertical bar, separating the columns with a double vertical bar. If one wishes to underline a particular row, the `--` tag, on a line by itself, indicates that the previous row is to be underlined. Note that this type of underlining only shows up in the PDF. All tables are set off by a line above the table and a line below.

Here is an example of a centered two column table with an additional header row, set off in italics, and underlined in the PDF:

```
<center>
```

```
{| lc
| ''symbol'' || ''rendering'' |
--
| @@&ndash;@@ || &ndash; |
| @@&mdash;@@ || &mdash; |
|}
</center>
```

The `lc` after the `{|` indicates how the two columns are to be rendered. In that token, an *l* indicates left justification, a *c* indicates centered, and an *r* indicates right justification. The position of the letter indicates the column to be formatted. So, in the example, the first column is to be left justified, while the second is to be centered. If this token is missing, then left justification is assumed for all columns.

The above specification renders as:

symbol	*rendering*
`&ndash;`	–
`&mdash;`	—