A Software Tool for Typesetting Quizzes

Hans Petter Langtangen[1,2]

[1]Center for Biomedical Computing, Simula Research Laboratory
[2]Department of Informatics, University of Oslo

Jun 22, 2014

Abstract

This note describes a system for writing quizzes (multiple-choice questions) in a compact text format with support for mathematics, computer code, and flexible formatting. The text format can be automatically translated to HTML and to a Python data structure. The HTML can be rendered directly in web pages, and the data structure can (in the future) be fed to online quiz/survey services a la Kahoot, JotForm, and Google forms.

1 A pure text quiz

1.1 Question and choices

Here is a typical quiz with a question and four alternative answers:

Question: What is the capital of Norway?

A. Helsinki
B. Drammen
C. Oslo
D. Denmark

Answer: C.

The above quiz can be specified by the compact text

```
!bquiz
Q: What is the capital of Norway?
Cw: Helsinki
Cw: Drammen
Cr: Oslo
Cw: Denmark
!equiz
```

Syntax:

- The quiz is specified between `!bquiz` ("begin quiz") and `!equiz` ("end quiz") tags.
- The question starts right after `Q:`.
- Wrong choices are specified with `Cw:`, and right choices with `Cr:`.
- The instructions `!bquiz`, `!equiz`, `Q:`, `Cw:`, and `Cr:` must appear at the beginning of a line.
- The text of (e.g.) a wrong answer (`Cr:`) lasts up to the next instruction (like `Cw:`, `Cr:`, `!equiz`).
- One can have as many wrong and right choices as desired.

Multiple correct choices are possible. Here is an example with three right choices:

```
!bquiz
Q: Which of the following cities are capitals?
Cw: Sidney
Cr: Kigali
Cw: Bonn
Cr: Bern
Cr: Ottawa
Cw: New York
!equiz
```

Question: Which of the following cities are capitals?

A. Sidney
B. Kigali
C. Bonn
D. Bern
E. Ottawa
F. New York

Answer: B, D, E.

1.2 Explanations of choices

Sometimes it is desireable to give some explanation why certain choices are wrong (or right). If you click on the symbols in the quiz below, explanations will be shown for Choices 1 and 2.

Question: What is the capital of Norway?

A. Helsinki
B. Drammen
C. Oslo
D. Denmark

Answer: C.

The corresponding typesetting applies `E: ...` after the choice to give an explanation of that choice.

```
!bquiz
Q: What is the capital of Norway?

Cw: Helsinki
E: Helsinki is the capital of Finland.

Cw: Drammen
E: Drammen is a small city close to Oslo.

Cr: Oslo

Cw: Denmark
!equiz
```

This time we also added some spaces for enhanced reading of the text.

1.3 Typesetting

It is easy to use *emphasize font*, **boldface**, color, hyperlinks, etc., since the text in the specification of questions, choices, and explanations are rendered according to the DocOnce markup language. For simple typesetting (emphasize, boldface, links) DocOnce also accepts Markdown syntax. Here is an example:

Question: This is a very famous quote:

> *Premature optimization is the root of all evil.*

This quote is attributed to

A. Geroge W. Bush
B. Donald Knuth
C. Ole-Johan Dahl

Answer: B.

The corresponding specification reads

```
!bquiz
Q: Here is a famous quote:

!bquote
```

```
*Premature optimization is the root of all evil.*
!equote
This quote is attributed to

Cw: Geroge W. Bush

Cr: Donald Knuth
E: According to "Wikiquote": "http://en.wikiquote.org/wiki/Donald_K...
Donald Knuth wrote this statement in *Structured Programming with
Goto Statements*. Computing Surveys, _6_:4,
pp. 261-301, _1974_.

Cw: Ole-Johan Dahl
E: Ole-Johan Dahl was a famous Norwegian professor of computer
science and together with Kristen Nygaard the inventor of
object-oriented programming, but he is not the man behind this
quote.
!equiz
```

1.4 Specification in HTML

Instead of using the compact text specification with DocOnce formatting, one can use a more verbose syntax and specify everything in HTML. The previous example then reads

```
<!-- --- begin quiz --- -->
<!-- --- begin quiz question --- -->
Here is a famous quote:

<blockquote>
    <em>Premature optimization is the root of all evil.</em>
</blockquote>

This quote is attributed to
<!-- --- end quiz question --- -->

<!-- --- begin quiz choice 1 (wrong) --- -->
Geroge W. Bush
<!-- --- end quiz choice 1 (wrong) --- -->

<!-- --- begin quiz choice 2 (right) --- -->
Donald Knuth
<!-- --- end quiz choice 2 (right) --- -->

<!-- --- begin explanation of choice 2 --- -->
According to
<a href="http://en.wikiquote.org/wiki/Donald_Knuth"
 target="_self">Wikiquote</a>,
Donald Knuth wrote this statement in <em>Structured Programming wi...
Goto Statements</em>. Computing Surveys, <b>6</b>:4,
pp. 261-301, <b>1974</b>.
<!-- --- end explanation of choice 2 --- -->

<!-- --- begin quiz choice 3 (wrong) --- -->
Ole-Johan Dahl
<!-- --- end quiz choice 3 (wrong) --- -->

<!-- --- begin explanation of choice 3 --- -->
Ole-Johan Dahl was a famous Norwegian professor of computer scienc...
and together with Kristen Nygaard the inventor of object-oriented
```

```
programming, but he is not the man behind this quote.
<!-- --- end explanation of choice 3 --- -->
<!-- --- end quiz --- -->
```

This syntax applies begin-end comments to mark the start and end of the question, the choices, and the explanations.

Warning.

The HTML specification of a quiz is not a meaningful HTML code for displaying the quiz in a browser, it is just an application of the HTML language to specify information and have full control of the typesetting details. Some program must interpret the HTML above and typset questions, choices, and explanations adequately.

2 Typesetting of mathematics and computer code

2.1 Mathematics

Mathematical typesetting follows a restricted LaTeX syntax. Inline formulas appear inside dollar signs, while separate equations appear inside `!bt` ("begin TeX") and `!et` ("end TeX") tags.

Simple example.

Question: Compute the result of $a + b$ in the case $a = 2$ and $b = 2$.

A. 5.
B. 4.
C. The computation does not make sense when a and b are given without units.

Answer: B.

The source code for defining the above quiz reads

```
!bquiz
Q: Compute the result of $a+b$ in the case $a=2$ and $b=2$.

Cw: 5.

E: Good attempt, especially when referring to the following story.

!bquote
An anthropologist was asking a primitive tribesman about arithmetic...
When the anthropologist asked, *What does two and two make?* the
tribesman replied, *Five.* Asked to explain, the tribesman said, *I...
have a rope with two knots, and another rope with two knots, and I
join the ropes together, then I have five knots.*
```

```
!equote

Cr: 4.
E: Seems trivial, but once upon a time...

FIGURE: [fig/1p1, width=180 frac=0.3]

Cw: The computation does not make sense when $a$ and $b$ are given ...
units.

E: It is indeed possible to add pure numbers without any units.
!equiz
```

Only a subset of LaTeX equation environment is supported!

To make sure blocks with equations come out correctly in different output formats (LaTeX, HTML, Sphinx, Markdown), only four types of standard LaTeX equation environments should be used:

1. Single equation without number: `\[ ... \]` or `equation*` environment
2. Single equation with number: `equation` environment
3. Multiple, aligned equations without numbers: `align*` environment
4. Multiple, aligned equations with numbers: `align` environment

This means that one has to stay away from `eqnarray`, `alignat`, and other common LaTeX equation environments. However, inside an equation, standard LaTeX math typesetting works (like `\alpha`, `\mbox{...}`, etc.).

A more complicated example.

Question: The equation

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

is famous in physics. Select the wrong assertion(s):
['gradient', 'divergence', 'curl', 'vector calculus']

A. The equation tells that the net outflow of something with velocity $\boldsymbol{u}$ in region is zero.
B. The equation tells that the vector field $\boldsymbol{u}$ is divergence free.
C. The equation implies that there exists a vector potential $\boldsymbol{A}$ such that $\boldsymbol{u} = \nabla \times \boldsymbol{A}$.
D. The equation implies $\nabla \times \boldsymbol{u} = 0$.
E. The equation implies that $\boldsymbol{u}$ must be a constant vector field.

Answer: D, E.

The corresponding code needed to define this quiz is listed below. Note a new line: `K: gradient; divergence; ...`. This construction allows specification of a set of keywords separated by semi-colon. The feature is handy when automatically selecting quizzes from a large database. There is no output of the keywords in typeset quizzes.

```
!bquiz
Q: The equation

!bt
\begin{equation}
\nabla\cdot\bm{u} = 0
label{cont:eq}
\end{equation}
!et
is famous in physics. Select the wrong assertion(s):

K: gradient; divergence; curl; vector calculus

Cw: The equation tells that the net outflow of something with
velocity $\bm{u}$ in region is zero.
E: This is right: integrating (ref{cont:eq}) over an arbitrary doma...
$\Omega$ and using Gauss' divergence theorem, we get the surface in...

!bt
\[ \int_{\partial\Omega}\bm{u}\cdot\bm{n}dS=0,\]
!et
where $\bm{n}$ is an outward unit normal on the boundary $\partial\...
The quantity $\bm{u}\cdot\bm{n}dS$ is the outflow of volume per
time unit if $\bm{u}$ is velocity.

Cw: The equation tells that the vector field $\bm{u}$ is divergence...
E: Yes, *divergence free* is often used as synonym for *zero diverg...
and $\nabla\cdot\bm{u}$ is the divergence of a vector field $\bm{u}...

Cw: The equation implies that there exists a vector potential $\bm{...
such that $\bm{u}=\nabla\times\bm{A}$.
E: Yes, this is an important result in vector calculus that is much
used in electromagnetics.

Cr: The equation implies $\nabla\times\bm{u}=0$.
E: No, only if $\bm{u}=\nabla\phi$, for some scalar potential $\phi...
we have $\nabla\times\bm{u}=0$.

Cr: The equation implies that $\bm{u}$ must be a constant vector fi...
E: No, it is the *sum* of derivatives of different components of $\...
that is zero. Only in one dimension, where $\bm{u}=u_x\bm{i}$
and consequently $\nabla\cdot\bm{u}=du/dx$, the vector field must b...
!equiz
```

2.2 Code

Inline computer code (variables, expressions, statements) are normally typeset with a monospace font, and this is enabled by enclosing the code in backticks. Blocks of computer code are typeset with `!bc` ("begin code") and `!ec` ("end code") tags. One can specify the computer language as part of the `!bc` tag: `!bc LX`, where `L` is the language (`py` for Python, `m` for Matlab, `cpp` for C++, for instance) and `X` can be `pro` for a complete executable program or `cod` for a code snippet (cannot be executed without additional statements)[1]. The quizzes below demonstrate the syntax.

Question: We want to create a Python list object of length `n` where each element is `0`. Is the following code then what we need?

```
import numpy
mylist = numpy.zeros(n)
```

A. Yes.
B. Yes, provided we write `np` instead of `numpy`:

```
import numpy as np
mylist = np.zeros(n)
```

C. No.

Answer: C.

```
!bquiz
Q: We want to create a Python list object of length ‘n‘ where each
element is ‘0‘. Is the following code then what we need?

!bc pycod
import numpy
mylist = numpy.zeros(n)
!ec

Cw: Yes.

E: Not exactly: ‘numpy.zeros‘ creates an array of zeros, not a list...

Cw: Yes, provided we write ‘np‘ instead of ‘numpy‘:

!bc pycod
import numpy as np
mylist = np.zeros(n)
!ec

E: No, this is fully equivalent to the original code, so ‘mylist‘ b...
an array, not a list.

Cr: No.
E: One would need to do ‘mylist = [0]*n‘ or ‘numpy.zeros(n).tolist(...
!equiz
```

2.3 Example: Putting it all together

[1]‘X‘ can also be `hide` for code that is not supposed to be shown, but possibly required to execute other snippets in an interactive document (that allows code to be edited and executed by the reader). This is currently being implemented in DocOnce’s support for Runestone Interactive books (using the Sphinx format).

Question:

```
from math import sin

def D(u, t, dt=1E-5):
    return (u(t + dt) - u(t - dt))/(2*dt)

def u(t):
    "A quadratic function."
    return t^2

print D(u, t=4),
print D(lambda x: return 2*x, 2)
```

The purpose of this program is to differentiate the two mathematical functions

$$\begin{aligned} u(t) &= t^2, \\ f(x) &= 2x. \end{aligned}$$

Determine which of the following assertions that is **wrong**.

A. In Python, the syntax for t^2 is `t**2`, not `t^2`, so the `u` function contains an error.
B. The string in the `u` function is a valid doc string.
C. The output from the program is on a single line, despite two `print` statements.
D. One cannot use `u` both inside the `D` function and in the outer calling code (the main program).
E. The call `D(lambda x: return 2*x, 2)` is equivalent to defining

```
def f(x):
    return 2*x
```

and then calling `D(f, 2)`.
F. There is danger of integer division in the `D` function.
G. The `D` function computes an approximate derivative of the function `u(t)`.
H. Both calls to `D` results in the exact derivative, provided we replace `t^2` by `t**2`.

Answer: D.

3 The Python data structure for quizzes

When a DocOnce file `mydoc.do.txt` containing quizzes (and other types of text, like this document) is translated to some format by the `doconce format` command, a Python list of all the quizzes is created and written to `.mydoc.quiz`. Each list element represents one quiz as a dictionary. The list corresponding to the quizzes in the current document starts with

```
[{'choices': [[u'wrong', u'Helsinki'],
              [u'wrong', u'Drammen'],
              [u'right', u'Oslo'],
              [u'wrong', u'Denmark']],
  'no': 1,
  'question': u'What is the capital of Norway?'},
```

A more complicated quiz with specification of prefix for the question and one choice (see example above in the question admonition) has explanations in the list for each choice, as well as two more keys (`choice prefix` and `question prefix`):

```
 {'choice prefix': [u'Answer:', None, None, None],
  'choices': [
    [u'wrong',
     u'Stockholm',
     u"Stockholm is the capital of Sweden, Norway's neighboring cou...
    [u'wrong',
     u'Bergen',
     u'Some people from Bergen may claim so... It is just the secon...
    [u'right', u'Oslo'],
    [u'wrong', u'Denmark']],
  'no': 5,
  'question': u'What is the capital of Norway?',
  'question prefix': u''},
```

The text in the Python list-dictionary data structure is ready-made for being displayed in HTML. Here is an example involving mathematics (MathJax syntax):

```
 {'choices': [
   [u'wrong',
    u'5.',
    u'Good attempt, especially when referring to the following stor...
\n\n<p>\n<blockquote>\n    An anthropologist was asking a primitive...
out arithmetic.\n    When the anthropologist asked, <em>What does t...
ke?</em> the\n    tribesman replied, <em>Five.</em> Asked to explai...
man said, <em>If I\n    have a rope with two knots, and another rop...
ots, and I\n    join the ropes together, then I have five knots.</e...
ote>'],
   [u'right',
    u'4.',
    u'Seems trivial, but once upon a time...\n\n<p>\n<center><p><im...
   [u'wrong',
    u'The computation does not make sense when \\( a \\) and \\( b ...
    u'It is indeed possible to add pure numbers without any units.'...
  'no': 6,
  'question': u'Compute the result of \\( a+b \\) in the case \\( a...
```

Computer code gets typeset by nice colors by default (using the Pygments package):

```
{'choices':
   [[u'wrong',
     u'Yes.',
     u'Not exactly: <code>numpy.zeros</code> creates an array of ze...
, not a list.'],
    [u'wrong',
     u'Yes, provided we write <code>np</code> instead of <code>nump...
     u'No, this is fully equivalent to the original code, so <code>...
    [u'right',
     u'No.',
     u'One would need to do <code>mylist = [0]*n</code> or <code>nu...
 'no': 8,
 'question': u'We want to create a Python list object of length <c...
```

The keys in the dictionaries in this data structure are

- `question`: the text of the question. This key is always present.
- `no`: the number of the quiz (starts at 1). This key is always present.
- `keywords`: a list of specified keywords for the quiz.
- `choices`: list of all the choices as 2- or 3-lists. First element is `right` or `wrong`, second is the text of the choice, the optional third element is the explanation (if it was specified).
- `choice prefix`: list of the prefix specified for each choice. `None` implies the default prefix (depends on the format, see the documentation of the `--quiz_choice_prefix=` option by typing `doconce format --help`). This key is absent if there are no specifications of such a prefix.
- `question prefix`: The prefix specified for the question, if different from the default value `Question:`.
- `new page`: Headline for a new page of quizzes - indicates that a new page is to be made.

4 Interpreting the quiz format

The program `doconce` must be run to interpret the quiz format, translate it to LaTeX, HTML, or other formats, and produce the Python representation of the collection of quizzes in a document. Let the name of the document with quiz specifications and optional text be `myquiz.do.txt`. Documents to play around with are `pure_quiz.do.txt` or the present document.

4.1 HTML

Plain HTML. A simple HTML rendering of quizzes is done by

```
Terminal> doconce format html myquiz
```

The result `myquiz.html` can be loaded into a browser.

Bootstrap HTML. A more sophisticated rendering is offered if a Bootstrap HTML style is adopted,

```
Terminal> doconce format html myquiz --html_style=bootstrap
```

Other variations of this class of styles are `bootswatch`, and `bootswatch_X`, where `X` is the name of a Bootswatch style, e.g., `journal`.

The Bootstrap-based styles have a striking red color for inline computer code, which can be avoided by giving the option `html_code_style=inherit`. See the documentation of DocOnce and Bootstrap for details and examples.

By default, there is a horizontal rule enclosing each quiz in the HTML format. This can be turned off by the option `--quiz_horizontal_rule=off`.

Multiple page HTML document. By inserting `!split` before headings in a DocOnce document, one can run

```
Terminal> doconce split_html myquiz.html
```

to create a multi-page document with navigation arrows. Bootstrap styles will also have a *Contents* pull-down menu to the right in the top bar. Without the `split_html` command (and `!split` instructions in the document), all the text will be in one HTML page.

Setting the question and choice prefix. The prefix before a question is `Question:` and before a choice is `Choice X`, where `X` is a number. To set a different prefix, see the `--quiz_question_prefix` and `--quiz_choice_prefix` commands when running `doconce format --help`.

4.2 LaTeX

The standard command to produce a LaTeX document is

```
Terminal> doconce format pdflatex myquiz \
          --without_answers --without_solutions
```

If desired, the correct answer to each quiz can be included by omitting `--without_answers`. Omitting `--without_solutions` will print the right answers and the available explanations below the quiz.

The next steps are

```
Terminal> doconce ptex2tex myquiz envir=minted
Terminal> pdflatex -shell-escape myquiz
```

to produce a PDF `myquiz.pdf`. The `doconce ptex2tex` step enables a lot of different typesetting of blocks of computer code. Here, we use the `minted` style, which applies the Pygments package. The default (no `envir` specification) is a plain LaTeX verbatim environment. Even more possibilities for typesetting computer code exist if one runs the `ptex2tex` program (and not `doconce ptex2tex`).