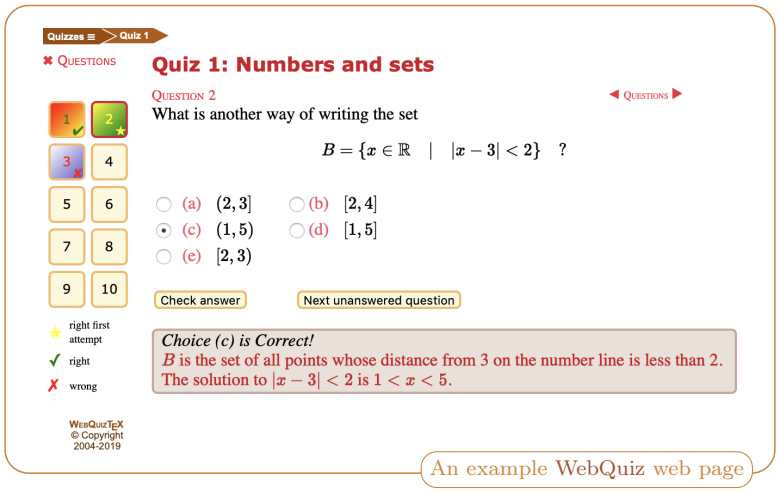
WebQuiz

A LaTeX package for writing online quizzes

Andrew Mathas

Version 5.2

WebQuiz is a LaTeX package for writing online quizzes. It allows the quiz author to concentrate on the content of quizzes, written in standard LaTeX, unencumbered by the technicalities of HTML and Javascript. Online quizzes written using WebQuiz can contain any material that can be written using LaTeX, including text, mathematics, graphics and diagrams.



An example WebQuiz web page

1 Introduction

Online quizzes provide a good way to reinforce learning, especially because they can give "interactive" feedback to the students[1] based on the answers that they give. Unfortunately, in addition to writing the actual quiz content there are significant technical hurdles that need to be overcome when writing an online quiz – and there are additional complications if the quiz involves mathematics or diagrams.

WebQuiz makes it possible to write online quizzes using LaTeX, which is the typesetting language used by mathematicians who use LaTeX to write their research papers, books and teaching materials. In principle, a WebQuiz quiz can contain anything that can be typeset using LaTeX. In practise, the LaTeX is converted to HTML using TeX4ht (and `make4ht`), so the quizzes can contain any LaTeXcommands that are understood by TeX4ht, which is almost everything. In particular, it is possible to use graphics constructed using `pstricks` and `tikz`; see Section 2.3.

WebQuiz supports the following three types of questions:

- Multiple choice questions with a unique correct answer
- Multiple choice questions zero or more correct answers
- Questions with an answer that is supplied by the student.

Each time a student answers a question it is possible to give them feedback, reinforcing their learning when they answer correctly and giving them further hints when they are wrong. This allows the quiz author to give targeted feedback to the student based on their answer.

The online quizzes constructed using WebQuiz can, in principle, contain anything that can be typeset by LaTeX. In particular, they do not need to contain mathematics. In fact, the quizzes do not even have to contain "questions" as it is possible for a WebQuiz "quiz" to contain only `discussion` environments that can be used to revise material, or to introduce new material, for the students; see §2.1c.

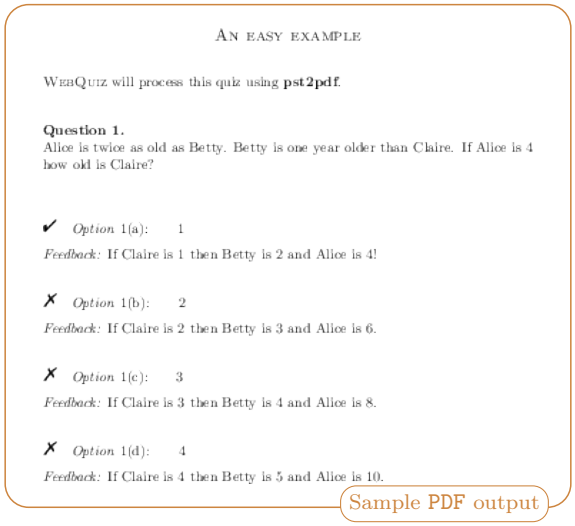
This introduction outlines how to use WebQuiz, however, the impatient reader may want to skip ahead directly to the Chapter 2, where the LaTeX commands used by WebQuiz are described.

The easiest way to explain how WebQuiz works is by example. The following LaTeX file defines a quiz with a single multiple choice question that has four possible answers, each of which has a customised feedback. Giving feedback to the students in each question is optional but the capability of being able to give students feedback on their answer is one of the main pedagogical advantages of online quizzes.

```
\documentclass{webquiz}
\title{An easy example}
\begin{document}
  \begin{question}     % a quiz question
     Alice is twice as old as Betty.
     Betty is one year older than Claire.
     If Alice is $4$ how old is Claire?
     \begin{choice}[columns=2]    % multiple choice question
       \correct   $1$  % first choice - correct answer
       \feedback  If Claire is $1$ then Betty is $2$ and Alice is~$4$!

       \incorrect $2$  % second choice - incorrect
       \feedback  If Claire is $2$ then Betty is $3$ and Alice is~$6$.

       \incorrect $3$  % third choice  - incorrect
       \feedback  If Claire is $3$ then Betty is $4$ and Alice is~$8$.

       \incorrect $4$  % fourth choice - incorrect
       \feedback  If Claire is $4$ then Betty is $5$ and Alice is~$10$.
     \end{choice}
  \end{question}
\end{document}
```

[1] Throughout this manual, "student" means any person taking the online quiz.

Since this is a LaTeX file it can be processed using `pdflatex`, or `latex`, to produce a readable and printable version of the quiz, which can be useful when proofreading. With the example above, the PDF version of the quiz looks like this:



An easy example

WebQuiz will process this quiz using **pst2pdf**.

Question 1.
Alice is twice as old as Betty. Betty is one year older than Claire. If Alice is 4 how old is Claire?

✔ *Option* 1(a): 1
Feedback: If Claire is 1 then Betty is 2 and Alice is 4!

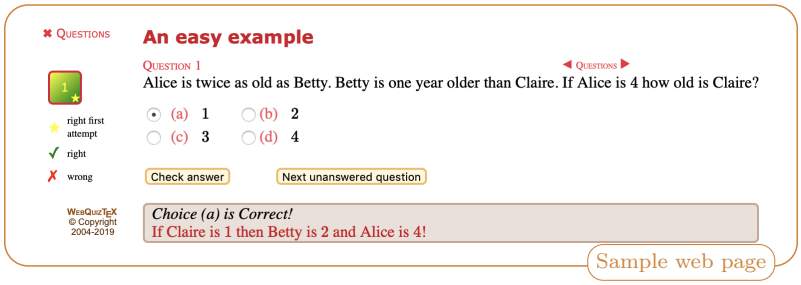
✗ *Option* 1(b): 2
Feedback: If Claire is 2 then Betty is 3 and Alice is 6.

✗ *Option* 1(c): 3
Feedback: If Claire is 3 then Betty is 4 and Alice is 8.

✗ *Option* 1(d): 4
Feedback: If Claire is 4 then Betty is 5 and Alice is 10.

Sample PDF output

Of course, the real reason for using WebQuiz is to create a web page for the quiz, which you do by processing the quiz using the `webquiz` command)instead of, say `pdflatex`). If you do this and open the resulting web page in your favourite browser, after selecting answer (a), you will see a web page like this:



Sample web page

The actual page that you see may be slightly different to this because the appearance of the web page depends partly on your choice of browser.

By default, the online version of the quiz displays one question at a time, with the question buttons serving the dual purpose of navigation between questions and displaying how successful the student was in answering the question. The decorations on the question buttons indicate whether the question has been attempted and, if so, whether it was answered correctly or incorrectly on the first or subsequent attempts. One of the main points of WebQuiz is that (optional) targeted feedback can be given to the student taking the quiz based on their answer.

1.1 What WebQuiz does and does not do

WebQuiz is a tool that makes it possible to write "interactive" online quizzes using LaTeX. To use WebQuiz you only need basic working knowledge of LaTeX. In particular, no familiarity of the underlying CSS, HTML or Javascript is required.

WebQuiz can be used to ask students a series of "quiz" questions. In addition, your online quiz web pages can contain course material using the WebQuiz `discussion` environment; see §2.1c. You can write WebQuiz quizzes that only contain questions, and no `discussion`, quizzes that contain questions and `discussion`, and (pseudo) quizzes that contains only `discussion` and no quiz questions.

By default, the online quizzes display one question (or `discussion` environment) at a time. It is also possible to display all of the quiz questions on a single web page (Section 2.2). One of the key features of WebQuiz is that you can give feedback to the students based on their answers. In this way you can give hints to the students to correct their mistakes and you can reinforce the students' understanding when they are correct. Each question in a quiz, and each quiz itself, can be attempted as many times as the student wants. WebQuiz does not limit the number of times that questions can be attempted.

As described in Section 2.1, WebQuiz supports the following question types:

- Multiple choice questions with a unique correct answer
- Multiple choice questions zero or more correct answers
- Questions that require students to type in an answer. There are several different "comparison" methods available for comparing the students answer where for example, the entered answers can be a "string" or a "number".

Questions can appear in either the same order that they appear in the LaTeX file for the quiz or in a random order that changes each time the quiz page is loaded. For multiple choice questions the order in which the choices appear is always the order that they appear in the LaTeX file for the quiz, even if the questions appear in random order.

WebQuiz supports several different languages and it provides a number of different colour schemes (see Section 4.1 and Appendix A).

WebQuiz quizzes are not timed and they do not have time-limits.

Quizzes made using WebQuiz are intended to be used as a revision resource rather than as an assessment tool. In particular, WebQuiz does not provide a mechanism for recording the marks obtained by the students taking the quiz (Section 2.2). Technically, it probably would not be very hard to record marks but this introduces a significant amount of extra overhead in terms of student authentication and interfacing with a database. In addition, if WebQuiz were used as an assessment tool then there would be additional "security issues" to ensure that the quiz content is secure. Currently, even though the solutions to the quiz questions do not appear in the HTML source code for the quiz pages it is possible to access the answers if you know what you are doing.

The questions in a WebQuiz quiz are static. In particular, WebQuiz quiz questions do not accept variables.

The WebQuiz program was designed to be run from the command-line. To process the file `quiz.tex` using WebQuiz you would type

```
> webquiz quiz          or          > webquiz quiz.tex
```

(Throughout this manual, `>` is used for the command-line prompt.)

It is possible to use WebQuiz from inside editors like TeXShop, but exactly how this is done will depend on the program that you use. In the case of TeXShop you need to define a new *engine* following, for example the instructions at tex.stackexchange.com/questions/376649.

1.2 Credits

WebQuiz was written and developed in the School of Mathematics and Statistics at the University of Sydney. The system is built on LaTeX with the conversion from LaTeX to HTML being done by Eitan Gurari's TeX4ht and `make4ht`.

To write quizzes using WebQuiz it is only necessary to know LaTeX, however, the underlying WebQuiz system actually has three components:

- A LaTeX document class file, `webquiz.cls`, and a TeX4ht configuration file, `webquiz.cfg`, that enables the quiz files to be processed by LaTeX and TeX4ht, respectively.
- A python program, `webquiz`, that translates the LaTeX into XML, using TeX4ht, and then into HTML.
- CSS, HTML and Javascript code controls and style the quiz web pages.

The LaTeX component of WebQuiz was written by Andrew Mathas and the python, CSS and Javascript code was written by Andrew Mathas based on an initial prototype that was written by

Don Taylor in 2001-2. Since 2004 the program has been maintained and developed by Andrew Mathas. Although the program has changed substantially since 2004, Don's idea of using TEX4ht, and some of his code, is still in use. Prior to releasing WebQuiz on `ctan` , the program was known as `MathQuiz`.

Hendrik Suess contributed code to improve session history and suggested the `\qref` command.

Thanks are due to Bob Howlett for general help with CSS and to Michal Hoftich for invaluable technical advice on TEX4ht. Thanks are due to Thomas Cailleteau Michael Palmer and Hendrik Suess for helpful feedback on the package.

2 The WebQuiz document class — LATEX commands

This chapter describes the commands and environments provided by the WebQuiz document class. This assumes that you have already installed and configured WebQuiz. If you have not yet initialised WebQuiz then please follow the instructions in Chapter 3.

All of the code examples given in this and other sections can be found in the `examples` subdirectory of the WebQuiz web directory.[2] Additional examples can be found in the Online manual, which is included, in `PDF` form, as Appendix B.

Zoomed out, the structure of a typical WebQuiz quiz file is a LATEX file of the form:

```
\documentclass{webquiz}
\title{A quiz}% optional, but potentially informative, title
\begin{document}

  \begin{question}% text for first question
  \end{question}

  \begin{question}% text for second question
  \end{question}

  \begin{question}% text for third question
  \end{question}

  ...

\end{document}
```

You should write your quizzes using the editor that you normally use to write LATEX documents. As you write your quiz, say `quiz.tex`, you should use `pdflatex` (or `latex`), in the usual way:

```
> pdflatex quiz
```

This is the easiest way to check that your quiz compiles and to proofread the output, just as if you were writing a normal LATEX document. When you are satisfied with the content of the quiz, then you can convert the quiz to an online quiz using the command

```
> webquiz quiz    or    > webquiz -d quiz  # -d = draft mode = faster!
```

The quiz file, `quiz.tex`, should be in a directory on your web server because WebQuiz creates a number of different files and directories when it converts the file into an online quiz and all of these files are needed to display the quiz on the web.

The reasons for using this workflow are:

- *Every file that you give to WebQuiz must be a valid LATEX file!*
- The `dvi` or `PDF` file produced by LATEX shows all of the information in the quiz *in an easy-to-read format*. That is, the `PDF` file displays the questions, the answers and the feedback that you are giving to the students. In contrast, by design, the online version of the quiz hides most of this information and shows it to the student only when they need to see it.

[2] After you have initialised WebQuiz, you can find the WebQuiz example directory from the command-line using: `webquiz -settings webquiz-www`.

- Typesetting the quiz file with LaTeX is *much faster* than processing it with WebQuiz. In fact, WebQuiz uses `htlatex` to process the quiz file at least *three times* in order to produce an XML file and it is only then that the WebQuiz program kicks in to rewrite this data as an HTML file. (If you use *draft mode* then `htlatex` processes the quiz file only once.)
- If LaTeX produces errors then WebQuiz will produce more errors. Further, *LaTeX error messages are much easier to read and understand than those produced by TeX4ht and WebQuiz.*

This said, WebQuiz does check for more errors in the quiz than LaTeX is (easily) able to do.

The `PDF` version of a quiz does not contain information about the unit, department or institution, which can be used in the breadcrumbs.

The next sections describe the commands and environments provided by WebQuiz for typesetting quizzes as well as the document-class options for the package. If you plan to use `pstricks` or `tikz` then you should read Section 2.3, which describes how to use graphics in a WebQuiz quiz. Section 2.4 describes a work-around for using (some?) LaTeX features that have not been configured for use with TeX4ht.

2.1 Question environments

The WebQuiz document class defines the following four environments:

question
: Each quiz question needs to be inside `question` environment

choice
: Typesets multiple choice questions, with one or more correct answers

discussion
: Includes (optional) discussion, or revision, material at the start of the quiz web page

quizindex
: Writes an index file for the quizzes in a "unit of study" and generates drop-drop menus in each quiz for the unit

This section describes these environments and gives examples of their use.

2.1a Question environments and the \answer macro

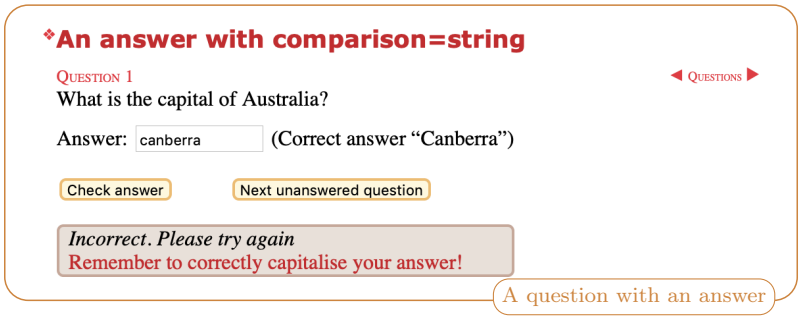
Each quiz question must be placed inside a `question` environment. Typically, a quiz has several questions, each wrapped in its own `question` environment. For brevity, most of the examples in this chapter have only one question. See the Online manual (Appendix B), in the WebQuiz web directory for a more complete quiz file.

This manual describes the WebQuiz commands, often by example. The following code-block generates a quiz with one question for which the student has to enter the answer. This answer is then compared with the correct answer as a string, which it must match exactly.

```
\documentclass[hidesidemenu]{webquiz}
\title{An answer with comparison=string}
\begin{document}
  \begin{question}     % a quiz question
     What is the capital of Australia?
     \answer[string]{Canberra} (Correct answer ``Canberra'')
     \whenRight  Yes, Sydney is the capital of NSW and Melbourne is the
     capital of Victoria
     \whenWrong Remember to correctly capitalise your answer!
  \end{question}
\end{document}
```

The *optional* macros `\whenRight` and `\whenWrong` are used to give the student additional feedback, when they are right, or further hints etc for approaching the question, when they are wrong. This feedback is displayed on the quiz page only when the student checks their answer.

The web page created by the code above, when an incorrect answer of “canberra”, instead of “Canberra” is given, looks something like the following:



A question with an answer

This example shows one way of using the `\answer` macro, which asks for the student to type in an answer to the question. The general syntax of this macro is:

```
\answer[comparison type]{correct answer}
```

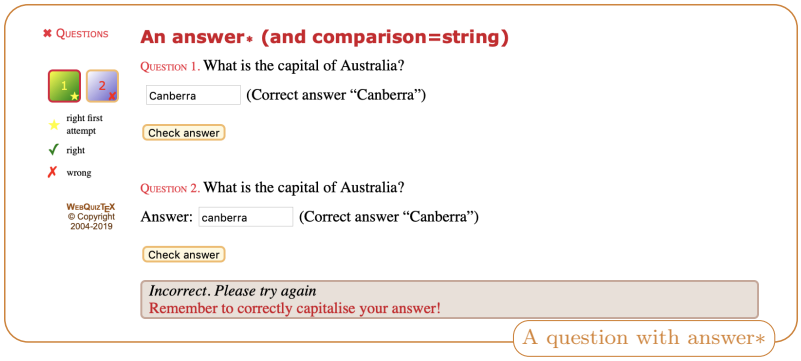
where the optional *comparison type* is one of:

```
complex   integer   lowercase   number   string
```

with `string` being the default. In addition, there is a $*$-variant of the answer macro that does not print the word “Answer” (or equivalent in other languages) before the input box. The syntax for the `\answer*` command is identical to that for the `\answer` except, of course, that there is a $*$:

```
\answer*[comparison type]{correct answer}
```

As `string` is the default answer comparison method, if we instead use `\answer*{Canberra}` in the last example then the quiz page generated by WebQuiz looks like:



A question with answer*

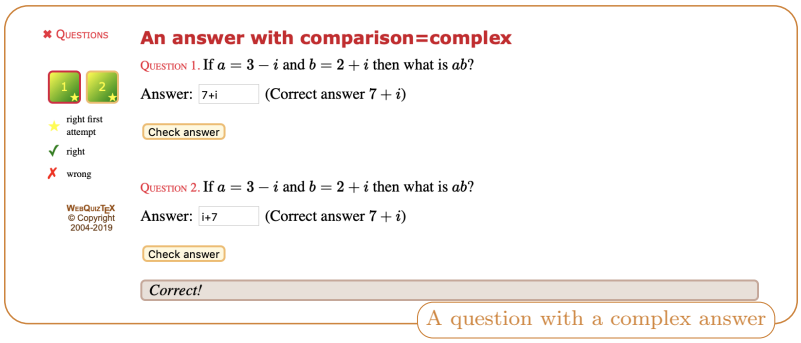
Notice that the word `Answer` no longer appears in front of the answer box. Of course, if you use the unstarred version of the `\answer`-macro together with the document class option `language=xxx` to change the default language (Section 2.2), then the appropriate translation of `Answer` will appear on the web page.

We now give a description of the other comparison types for the `\answer` and `\answer*` macros together with code examples and screenshots.

complex comparison The answers are compared as complex numbers: the answer is marked as correct if it has the same real and imaginary parts.

```
\documentclass[onepage]{webquiz}
\title{An answer with comparison=complex}
\begin{document}
  \begin{question}     % a quiz question
     If $a=3-i$ and $b=2+i$ then what is $ab$?
     \answer[complex]{7+i} (Correct answer $7+i$)
```

```
  \end{question}
  \begin{question}     % a quiz question
     If $a=3-i$ and $b=2+i$ then what is $ab$?
     \answer[complex]{7+i} (Correct answer $7+i$)
  \end{question}
\end{document}
```

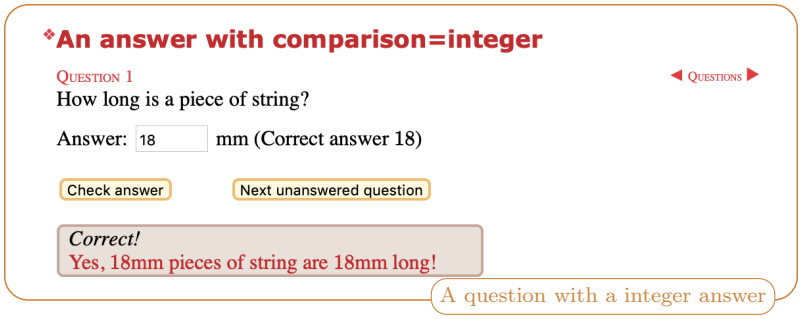


A question with a complex answer

Observe that the correct answer is given in the quiz file as $7 + i$ and that WebQuiz accepts $i + 7$ as the correct answer.

integer comparison The answers are compared as integers. If the correct answer was 18 and a student entered 36/2 then their answer would be marked wrong.

```
\documentclass[hidesidemenu]{webquiz}
\title{An answer with comparison=integer}
\begin{document}
  \begin{question}     % a quiz question
     How long is a piece of string?
     \answer[integer]{18} mm (Correct answer $18$)
     \whenRight Yes, $18$mm pieces of string are $18$mm long!
     \whenWrong Incorrect! Is your ruler working?
  \end{question}
\end{document}
```

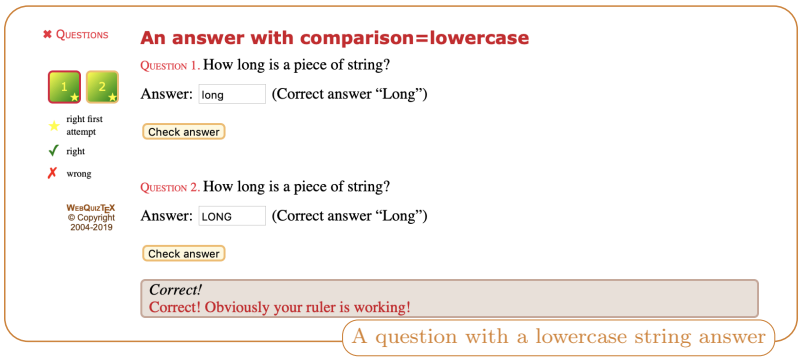


A question with a integer answer

lowercase comparison The quiz answer and the students' answer are both converted to lower case and then compared as strings.

```
\documentclass[onepage]{webquiz}
\title{An answer with comparison=lowercase}
\begin{document}
  \begin{question}     % a quiz question
     How long is a piece of string?
     \answer[lowercase]{Long} (Correct answer ``Long'')
     \whenRight Correct! Obviously your ruler is working!
     \whenWrong Incorrect! Is your ruler working?
```

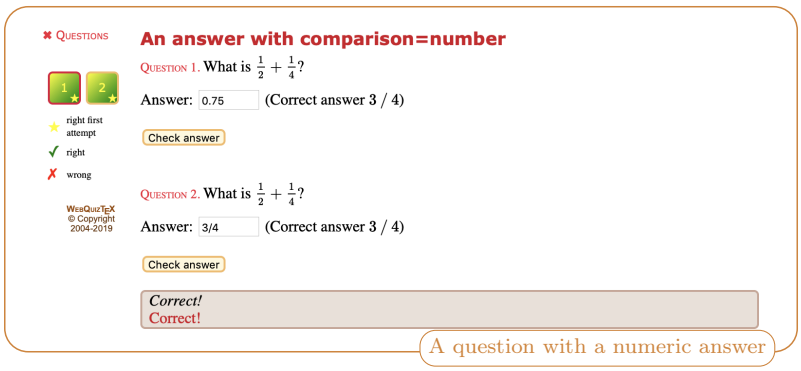
```
  \end{question}
  \begin{question}     % a quiz question
     How long is a piece of string?
     \answer[lowercase]{lonG} (Correct answer ‘‘Long’’)
     \whenRight Correct! Obviously your ruler is working!
     \whenWrong Incorrect! Is your ruler working?
  \end{question}
\end{document}
```



A question with a lowercase string answer

number comparison The quiz answer and the students' answer are compared as numbers.

```
\documentclass[onepage]{webquiz}
\title{An answer with comparison=number}
\begin{document}
  \begin{question}     % a quiz question
      What is $\frac12+\frac14$?
     \answer[number]{3/4} (Correct answer $3/4$)
     \whenRight Correct!
     \whenWrong Incorrect! Is your ruler working?
  \end{question}
  \begin{question}     % a quiz question
      What is $\frac12+\frac14$?
     \answer[number]{3/4} (Correct answer $3/4$)
     \whenRight Correct!
     \whenWrong Incorrect! Is your ruler working?
  \end{question}
\end{document}
```



A question with a numeric answer

Notice that the answer is given in the LaTeX file as 3/4 and that the equivalent fraction, 0.75, is accepted as the correct answer.

string comparison This is the default, so `\answer{word}` and `\answer[string]{word}` are equivalent. The student's answer is marked correct if and only if it agrees exactly with the quiz answer.

2.1b Multiple choice questions

The multiple choice options for a quiz question need to be placed inside a `choice` environment – and every `choice` environment needs to be inside a `question` environment.

The `choice` environment accepts two optional arguments, can appear in any order:

- The word `single` (default, and can be omitted) or `multiple`, which indicates whether the quiz has a *single* correct answer or whether 0 or more of the answers are correct, respectively.
- The number of *columns* in which to typeset the choices. This is specified as `columns=n`, where n is a non-negative integer. By default, the choices appear in $n = 1$ columns (`columns=1`). *Use `columns=n` with care when $n > 1$ as multiple columns do not always display well on mobile devices.*

The key difference between these two types of `choice` questions is that a `single`-choice environment uses radio boxes, so it is only possible to select *one correct answer*. In a `multiple`-choice environment checkboxes are used, so that is possible to select *zero or more correct answers*.

A `choice` environment modelled on the standard LaTeX list environments (*enumerate*, *itemize*, *description*, ...), except that instead of using the `\item` command to separate the items the `choice` environment uses `\correct` `\incorrect` , which indicate correct and incorrect answers respectively. In addition, after each `\correct` or `\incorrect` you can, optionally, use `\feedback` to give feedback to the student taking the quiz. Like `\whenRight` and `\whenWrong` this feedback is displayed only when the student checks their answer.

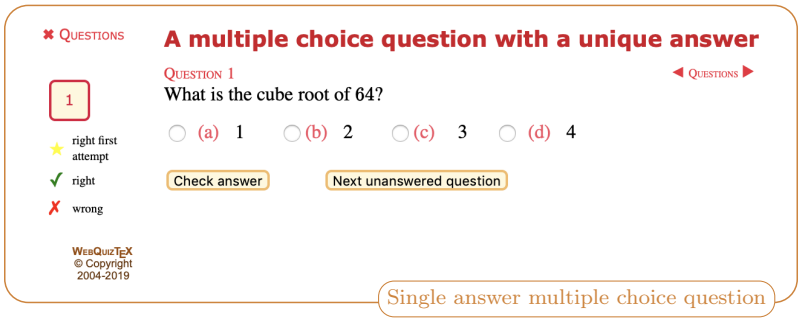
Here is an example of a `single`-choice question with a unique answer:

```
\documentclass{webquiz}
\title{A multiple choice question with a unique answer}
\begin{document}
  \begin{question}     % a quiz question
     What is the cube root of $64$?
     \begin{choice}[columns=4]% unique answer rendered in four columns
       \incorrect 1 \feedback No,  $1^3=1$
       \incorrect 2 \feedback No,  $2^3=8$
       \incorrect 3 \feedback No,  $3^3=27$
       \correct   4 \feedback Yes, $4^3=64$
     \end{choice}
  \end{question}
\end{document}
```

Note that `single` is the default, so this could also be written as

```
    \begin{choice}[columns=3, single]
      ...
    \end{choice}
```

It is not necessary to put the `\feedback` lines on the same line as the `\incorrect` and `\incorrect`; this is done only to make the example more compact. This results in the following web page:

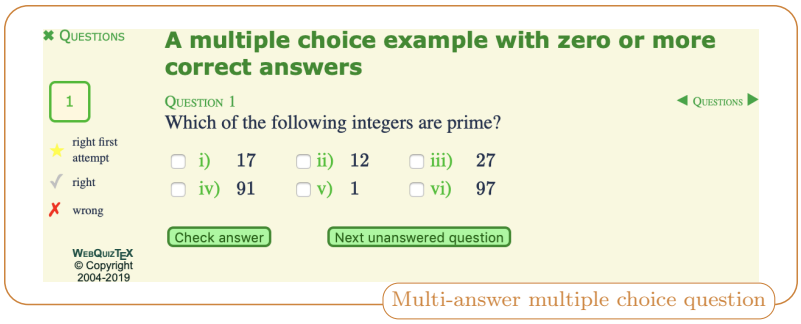


Here is an example of a multiple choice question that has two correct answers:

```
\documentclass[theme=vibrant]{webquiz}
```

```
\title{A multiple choice example with zero or more correct answers}
\renewcommand\thechoice{\roman{choice})}
\begin{document}
  \begin{question}     % a quiz question
     Which of the following integers are prime?
     % multiple (0 or more) correct answers rendered in three columns
     \begin{choice}[multiple, columns=3]
       \correct   $17$ \feedback Yes, the only divisors of $17$ are $1$
                  and $17$
       \incorrect $12$ \feedback No, $12 = 3\times 4$
       \incorrect $27$ \feedback No, $27=3^3$
       \incorrect $91$ \feedback No, $91 = 3\times 17$
       \incorrect $1$ \feedback One is not a prime number because it
                  is invertible
       \correct   $97$ \feedback Yes, the only divisors of $97$ are $1$
                  and $97$
     \end{choice}
  \end{question}
\end{document}
```

Notice that this example uses the document-class option `theme=vibrant`, which changes the WebQuiz colour theme; see Section 2.2.



Multi-answer multiple choice question

When the optional argument `multiple` to the `choice` environment is used, as above, then the question is marked correct if and only if *all* of the correct choices, and none of the incorrect choices, are selected. If the student's selections are not completely correct then are given feedback that is randomly selected from amongst their wrong choices (that is, the feedback is randomly selected from the set of `\correct` choices that were not selected and the `\incorrect` choices that were selected).

Finally, observe that the multiple choice items in the screenshot above are labelled by roman numerals. The items in a `choice` environment are labelled by a standard LaTeX counter, that is also called `choice`. Redefining the LaTeX macro `\thechoice` changes how the corresponding question choices are labelled in the online quiz. For example, to label the items in a `choice` environment by A), B), C) ... add the line:

```
    \renewcommand\thechoice{\Alph{choice})}
```

to the preamble of the LaTeX file for your quiz.

2.1c Discussion environments

In addition to asking questions, it is possible to display revision, or discussion, material on the quiz web page using the `discussion` environment. All discussion material is displayed on the quiz page *before* the questions in the quiz and the (short) titles for the discussion items appear before the quiz questions in the menu on the left-hand side of the page — it is not possible to interleave discussion items and questions in the side menu.

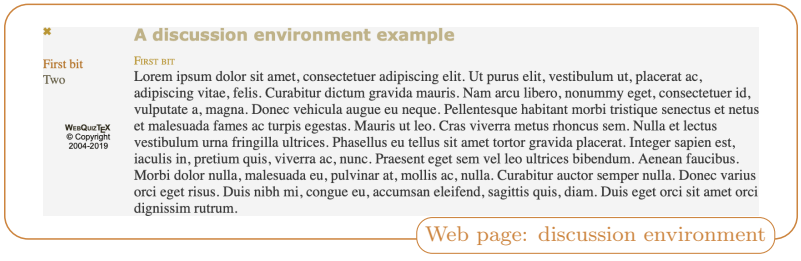
Each quiz can have zero or more discussion environments. These environments can, in principle, contain arbitrary LaTeX code. The syntax for the `discussion` environment is:

```
\begin{dicussion}[short heading][heading]
  ...
\end{dicussion}
```

The *short heading* is used in the side-menu. Both the *heading* and *short-heading* are optional, both defaulting to `Discussion`. If only one heading is given then this sets both the *short heading* and *heading* for the discussion item. For example, running the following LaTeX file, which uses `theme=muted`, through WebQuiz

```
\documentclass[theme=muted]{webquiz}
\usepackage{lipsum}% for some random text
\title{A discussion environment example}
\begin{document}
  \begin{discussion}[First bit]
     \lipsum[1]
  \end{discussion}
  \begin{discussion}[Two][Second bit]
     \lipsum[2]
  \end{discussion}
\end{document}
```

produces the quiz page:



Web page: discussion environment

As with the questions, only one `discussion` environment is displayed on the quiz web page at a time (unless the document-class option `onepage` is used). It is possible to have quizzes that contain only `discussion` environments, with no questions, and quizzes that contain only `question` environments, with no discussion.

If you have a mixture of `discussion` and `question` environments then it is useful to be able to add thinks between them. LaTeX provides the `\label` and `\ref` commands for cross-referencing, so WebQuiz builds on this idea and provides the three commands `\dref`, `\qref` and `\Qref` to reference `discussion` and `question` environments. The syntax for these commands is as follows:

```
\dref[optional text]{LaTeX label}      % inserts discussion button
\dref*[optional text]{LaTeX label}     % inserts discussion link
\qref[optional text]{LaTeX label}      % inserts question button by label
\qref*[optional text]{LaTeX label}     % inserts question link by label
\Qref[optional text]{question number}  % inserts question button by number
\Qref*[optional text]{question number} % inserts question link by number
```

In each case the text in the link or button defaults to either the short-title, for discussion environments, or the question number for questions. The "optional text" is used instead whenever it is supplied. These commands can be used anywhere in a quiz, including `discussion`, `question` and `choice` environments and inside feedback text for the students written using `\feedback`, `\whenRight` and `\whenWrong`.

The macros `\dref` and `\qref` use a standard LaTeX *label*, defined using the `\label` command, to insert a button or a link. In contrast, `\Qref` uses the actual *question number*, so it is not necessary to define a `\label` for a question when using `\Qref`.

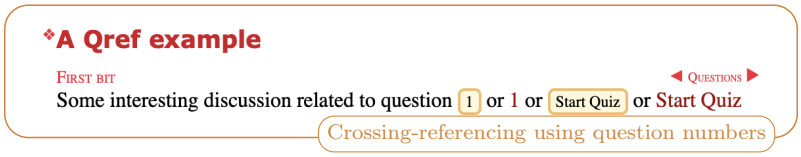
Even though the macros `\qref` and `\Qref` are quite similar they serve different functions when the `randomorder` document class option is used (see Section 2.2). In this case we do not know ahead of time the question numbers that will be used in the quiz. So if `q:one` is the label for the

first question in the LaTeX file then `\qref{q:one}` will insert a button to this question *but* this will almost certainly not be Question 1. On the other hand, we can create a "Start quiz" button, for example, that will open Question 1 on any quiz, using `\Qref[Start quiz]{1}`.

Here is an example that shows how `\Qref` works:

```
\documentclass[hidesidemenu]{webquiz}
\title{A Qref example}
\begin{document}
  \begin{discussion}[First bit]\label{d:one}
    Some interesting discussion related to question \Qref{1} or \Qref*{1}
    or \Qref[Start Quiz]{1} or \Qref*[Start Quiz]{1}
  \end{discussion}
  \begin{question}\label{q:one}
     An interesting question relating to discussion \dref{d:one}\answer{1}
  \end{question}
\end{document}
```

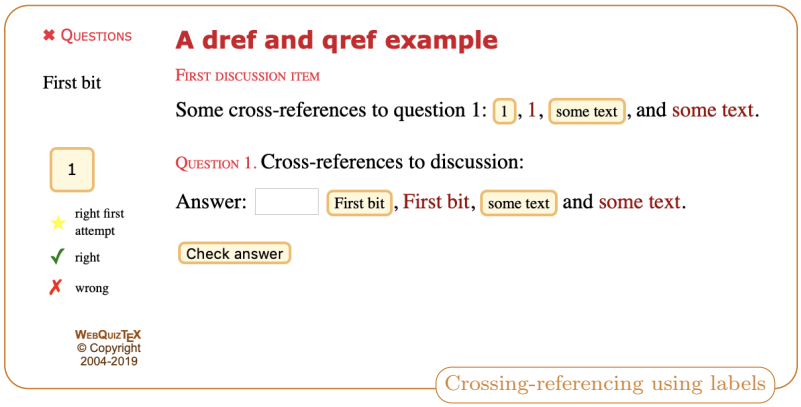
which produces the online quiz:



Similarly, here is an example showing how `\dref` and `qref` are used:

```
\documentclass[onepage]{webquiz}
\title{A dref and qref example}
\begin{document}
  \begin{discussion}[First bit][First discussion item]\label{d:one}
    Some cross-references to question 1: \qref{q:one}, \qref*{q:one},
    \qref[some text]{q:one}, and \qref*[some text]{q:one}.
  \end{discussion}
  \begin{question}\label{q:one}
    Cross-references to discussion: \answer{1} \dref{d:one}, \dref*{d:one},
    \dref[some text]{d:one} and \dref*[some text]{d:one}.
  \end{question}
\end{document}
```

This code produces the online quiz:



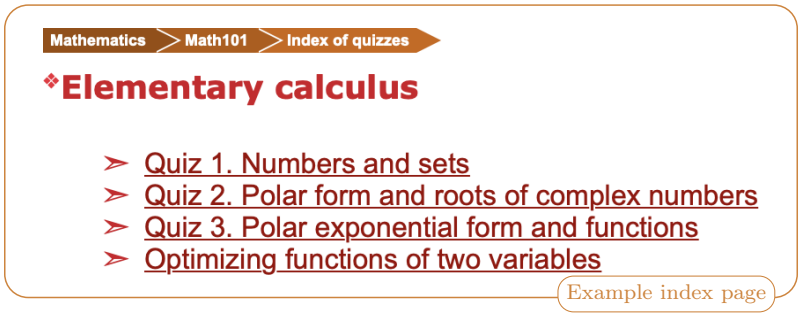
When using the `randomorder` document class option (Section 2.2), "optional text" should always be given when using `\qref`. This is because the question number that is displayed by default will always be the question number *in the LaTeX file* rather than the question number in the online quiz.

2.1d Index pages for quizzes

Most quizzes occur in sets that cover related material, such as for a particular unit of study. The `quizindex` environment creates an index web page for related sets of quizzes. The LaTeX files for all of these quizzes must be in the same directory, or folder, on the web server. The index web page is a WebQuiz file of the form:

```
\documentclass[hidesidemenu]{webquiz}
\BreadCrumbs{ Mathematics / | Math101 /math101 | Index of quizzes }
\title{Elementary calculus}
\begin{document}
  \begin{quizindex}
    \quiz{Numbers and sets}
    \quiz{Polar form and roots of complex numbers}
    \quiz{Polar exponential form and functions}
    \quiz*{Optimizing functions of two variables}
  \end{quizindex}
\end{document}
```

which generates a web page that looks like:



Example index page

As the next section describes, index files are also used to automatically add a drop-down menu that contains the quiz-index to the breadcrumbs on all of the quiz web pages. This drop-down menu provides an easy way to navigate between all of the quizzes for a particular unit of study.

As the example above shows, the entries in the `quizindex` are given using the `\quiz` or the `\quiz*` command. The syntax for these commands is

```
\quiz[URL for quiz]{Title for quiz}
\quiz*[URL for quiz]{Title for quiz}
```

The `\quiz` macro automatically inserts the quiz numbers into the index listing. The `\quiz*` command is identical *except* that it does not add `Quiz 1.`, `Quiz 2` etc to the index listing. By default, the URLs for the quizzes in the index are assumed to be of the form `quiz1.html`, `quiz2.html`, `quiz3.html`, ... These URLs can be changed using the optional argument of the `\quiz` and `\quiz*` commands. For example,

```
\quiz[realquiz.html]{This is the real quiz}
```

would create an item in a quiz index that links to the web page `realquiz.html`.

Index pages in other languages are produced in exactly the same way. For example,

```
\documentclass[hidesidemenu,language=czech]{webquiz}
\usepackage[czech]{babel}
\usepackage[T1]{fontenc}
\Department{Matematika}\DepartmentURL{/}
\BreadCrumbs{ department | Mat101 /mat101 | breadcrumb }
\BreadCrumb{ Index kvízů }
\title{Elementární počet}
\begin{document}
  \begin{quizindex}
```

```
    \quiz{Čísla a množiny}
    \quiz{Polární forma a kořeny složitých čísel}
    \quiz{Polar exponenciální forma a funkce}
    \quiz*{Optimalizace funkcí dvou proměnných}
  \end{quizindex}
\end{document}
```

produces the web page:

A Czech index

The next section describes the `\BreadCrumb` and `\BreadCrumbs` commands.

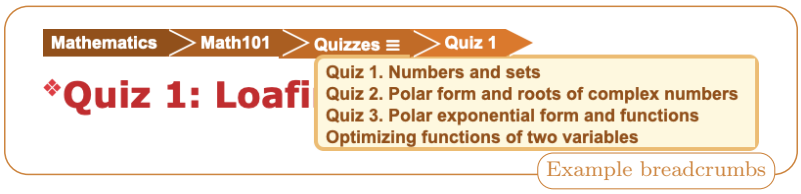
At most one file in each directory should contain a `quizindex` environment. This is because WebQuiz creates the file Javascript file `quizindex.js` whenever it compiles a `quizindex` environment. This file contains Javascript commands for displaying the quiz index.

2.1e Breadcrumbs

WebQuiz provides a straightforward way to place navigation breadcrumbs at the top of the quiz web page. By default the breadcrumbs are disabled. If you have a `\BreadCrumbs` command like:

```
\BreadCrumbs{Mathematics /|Math1001 /u/Math1001|quizindex|title}
```

in your LaTeX file then WebQuiz will add a corresponding strip of breadcrumbs, or navigation links, to the top of your quiz page:



Example breadcrumbs

The drop-down menu is normally hidden, appearing only after the ≡ "button" on the web page is clicked.

Usually, most of the breadcrumbs are navigation links to other web pages. In the example above:

- The first "crumb' inside the `\BreadCrumbs` command is `Mathematics /`. This inserts the text `Mathematics` together with a (relative) URL to `/`, the root directory for the web server, which is often the correct URL for the department (or the institution)
- The `Math101 /u/math101` inserts the text `Math101` as the second breadcrumb with URL `/u/math101`.
- The `quizindex` inserts the text `Quizzes` together with the symbol ≡, which opens a drop-down menu that contains the list of quizzes in the quiz index for the unit. This is described in more detail below.
- Finally the `title` in the breadcrumbs inserts, as text, the part of the title *before the first colon* in the title, where the title is given by `\title{...}`.

The breadcrumbs for the quiz web page can be either be configured quiz-by-quiz, using the `\BreadCrumbs` macro , as above, or by setting default `breadcrumbs` in the `webquizrc` file (Section 4.2) using the command-line option

```
> webquiz --edit-settings
```

as described in Section 4.1. Breadcrumbs are disabled by default.

The breadcrumbs inside the \BreadCrumbs{...} command are given as a "|-separated list". For example, quite reasonable breadcrumbs are given by:

```
\BreadCrumbs{ department | unitcode | quizindex | title }
```

To make this the default set of breadcrumbs use webquiz --edit-settings to set breadcrumbs in the webquizrc file (Section 4.2) to:

```
department | unitcode | quizindex | title
```

More generally, the breadcrumbs can be specified as:

```
\BreadCrumbs{ crumb1 | crumb2 | crumb3 | crumb4 | ... }
```

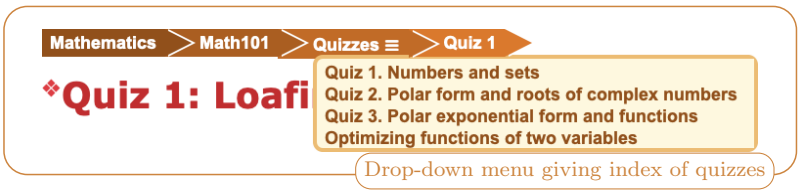
In principle, there can be arbitrarily many crumbs in your breadcrumbs but, in practice, five is more than enough. The crumbs inside a \BreadCrumbs command have the following meanings:

breadcrumb
: The breadcrumb for the current quiz, which is set using the \BreadCrumb macro. This breadcrumb is purely descriptive, with no hyperlink being added: only the text given by \BreadCrumb appears.

department
: This expands to a link to your department, where the department text is set using the macro \Department and its URL is set by \DepartmentURL.

institution
: This expands to a link to your institution, where the institution text is set using \Institution and its URL is set by \InstitutionURL. The institution also appears in the side-menu above the WebQuiz copyright notice.

quizindex
: This expands to Quizzes, which is a link to the index page for your unit, as defined by \QuizzesURL, which is described below. In addition, if the directory contains a BashCode|quizindex.js| file, which is created by the quizindex environment (see §2.1d), then the symbol ≡ will appear, giving access to a drop-down menu to the index page, looking something like this:



Drop-down menu giving index of quizzes

Such drop-down menus are automatically added to quiz web pages that have a quizindex breadcrumb| as soon as an quiz page that contained a WebQuiz quizindex environment has been compiled in the current directory.

For those interested in how this is done, whenever WebQuiz compiles a quizindex environment it creates a Javascript file quizindex.js. Every quiz file includes this Javascript file, if it exists, and this allows it to display a drop-down menu for the quiz index.

Title
: This expands to the full title of the quiz page, without a hyperlink, as given by the \title.

title
: This expands to the part of title of the quiz page, without a hyperlink, that occurs *before* the first colon in the title of the quiz. For example, if the title is given as

```
\title{Quiz 1: The wonders of life}
```

then "Quiz 1" will be added to the list of breadcrumbs. If the title does not contain a colon then the full title is printed.

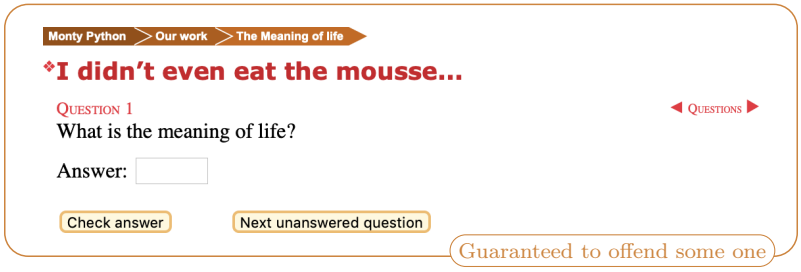
unitcode
: This expands to a link to the unit code, where the unit code text is set using `\UnitCode`, and its URL is set by `\UnitURL`

unitname
: This expands to a link to the unit name, where the unit name text is set using `\UnitName`, and its URL is set by `\UnitURL`

In addition, each *crumb* in a breadcrumb, except for the "magic crumbs" listed above, is allowed to be arbitrary text — although, non-ascii characters may cause problems. In this case, the last "word" in the crumb is treated as a URL if it either beings with a forward slash, `/`, or if it begins with `http`. For example, the code:

```
\documentclass[hidesidemenu]{webquiz}
\BreadCrumbs{ Monty Python http://www.montypython.com/
            | Our work http://www.montypython.com/ourwork
            | The Meaning of life http://www.montypython.com/film_Monty\%20
                Python's\%20The\%20Meaning\%20of\%20Life\%20(1983)/17
}
\title{I didn't even eat the mousse...}
\begin{document}
  \begin{question}
    What is the meaning of life?
    \answer{42}
  \end{question}
\end{document}
```

results in the following breadcrumbs:

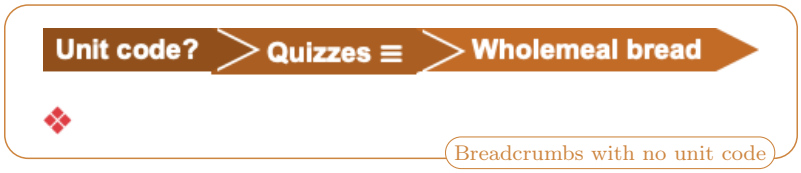


Notice that it is necessary to correctly escape spaces etc in URLs that are specified this way. Similarly, all of the characters in the breadcrumbs should be ascii characters as unicode is likely to cause encoding issues (compare with the Czech index given in §2.1d).

If any part of a "magic" breadcrumb has not been defined then it is printed with a question mark on the web page. For example, the quiz file

```
\documentclass[hidesidemenu]{webquiz}
\BreadCrumb{Wholemeal bread}
\BreadCrumbs{unitcode|quizindex|breadcrumb}
\begin{document}\end{document}
```

does not define the unit code, so it results in the "questionable" first breadcrumb:



Here is the list of WebQuiz macros that we be used to set the values of the "magic" breadcrumbs inside a `\BreadCrumbs`. Note that default values for many of these "crumbs" can be given in the `webquizrc` file (Section 4.2).

\BreadCrumb

The `\BreadCrumb` command sets the `breadcrumb` variable in the `\BreadCrumbs`. The primary use for this is when you have default breadcrumbs in the `webquizrc` file (Section 4.2) like

```
breadcrumbs = department | unitcode | breadcrumb
```

Using `\BreadCrumb` allows you to set the last crumb to some meaningful text that describes the quiz.

\Department

The `\Department` command sets the name of the `department`. As described earlier in this section, by default, the `department` is the first item in the breadcrumbs that appear at the top of the web page.

The default department can be set in the `webquizrc` file (Section 4.2) using `webquiz --edit-settings`.

Default value: `''` (i.e. the empty string)

\DepartmentURL

The `\DepartmentURL` command sets URL for the department. As described earlier in this section, by default the department URL is the link in the first breadcrumb on each web page.

The default department URL can be set in the `webquizrc` file (Section 4.2) using `webquiz --edit-settings`.

Default value: `/`

\Institution

The `\Institution` command sets the institution, or university. The `institution` appears below the question buttons in the left-hand navigation menu that appears on every quiz web page (provided that the screen size is not too small). As described earlier in this section, the institution can be used in the web page breadcrumbs.

The default institution can be set in the `webquizrc` file (Section 4.2) using `webquiz --edit-settings`.

Default value: `''` (i.e. the empty string)

\InstitutionURL

The `\InstitutionURL` command sets the institution URL. This is used as the link for the `institution` in the left-hand navigation menu that appears on every quiz page. As described earlier in this section, the institution URL can be used in the web page breadcrumbs.

The default institution URL can be set in the `webquizrc` file (Section 4.2) using `webquiz --edit-settings`.

Default value: `/`

\QuizzesURL

The `\QuizzesURL` command sets the URL for the suite of quizzes attached to this unit of study. As described earlier in this section, this can be used in the breadcrumb at the top of the quiz web page.

Default value: `UnitURL/Quizzes`, where `UnitURL` is set using `\UnitURL`

\UnitCode

The `\UnitCode` command sets the unit of study code for the unit that the quiz is part of.

\UnitName

The `\UnitName` command sets the name of the unit of study for the unit that the quiz is attached to.

\UnitURL

The \UnitURL command sets the URL for the unit of study code for the unit that the quiz is attached to.

It makes sense to set defaults for \BreadCrumbs \Department, \DepartmentURL, \Institution and \InstitutionURL in the webquizrc file (Section 4.2). After doing this, a typical WebQuiz file might look like this:

```
\documentclass{webquiz}
\UnitName{Fundamental stuff}
\UnitCode{Stuff101}
\UnitURL{/courses/stuff101}

\title{Stuffing: the art of taxidermy}

\begin{document}
  \begin{question}
    first question...
  \end{question}
  \begin{question}
    second question...
  \end{question}
\end{document}
```

If you have many quizzes for many different units then a better approach is to create a LaTeX "package", say ourunits.sty, to set these variables:

```
% ourunits.sty - set unit names, codes and URLs
\DeclareOption{stuff101}{
  \UnitName{Fundamental stuff}
  \UnitCode{Stuff101}
  \UnitURL{/courses/stuff101}
}
\DeclareOption{stuff102}{
  \UnitName{Fundamental stuff too}
  \UnitCode{Stuff102}
  \UnitURL{/courses/stuff102}
}
\ProcessOptions
\endinput
```

Then you can replace the opening lines of the quiz file with \usepackage[stuff101]{ourunits}. Of course, you could also set the default \Department, \DepartmentURL, \Institution and \InstitutionURL in such a style file as well.

2.2 WebQuiz document class options

The WebQuiz document class supports the following options:

```
fixedorder, hidesidemenu, language, onepage, pst2pdf,
separatepages, showsidemenu, theme, tikz
```

This section describes all of these document-class options except for tikz and pst2pdf, which are discussed in Section 2.3. Many of the document-class options below occur in pairs and defaults for many of these can be set in the webquizrc file (Section 4.2) file. The settings given in the LaTeX file for a quiz will always override the default settings in the webquizrc file.

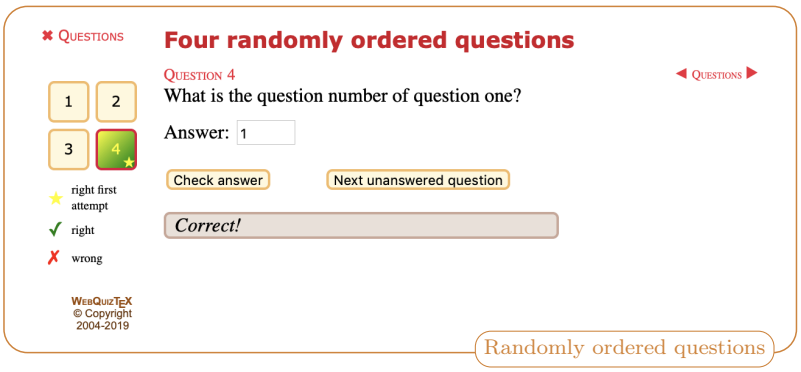
fixedorder, randomorder

By default, the questions in the quiz are displayed in a fixedorder for all students who take the quiz. This order is the order that the question appear in the LaTeX file for the quiz. That is, the first question in the online quiz is the first question appearing in the LaTeXfile, the

second question in the quiz is the second question in the LaTeX file, and so on. With the randomorder option the questions in the quiz are displayed in a random order that changes each time that the quiz is run. With this option, the online quiz questions are generally in a different order for every student. For example, the code:

```
\documentclass[randomorder, showsidemenu]{webquiz}
\title{Four randomly ordered questions}
\begin{document}
  \begin{question}What is the question number of question one?  \answer{1}\end{
      question}
  \begin{question}What is the question number of question two?  \answer{2}\end{
      question}
  \begin{question}What is the question number of question three?\answer{3}\end{
      question}
  \begin{question}What is the question number of question four? \answer{4}\end{
      question}
\end{document}
```

produces the quiz with randomly arranged quiz questions, such as:



(So the first question in the LaTeX file is being displayed as the fourth online quiz question. The next time the page is loaded, such as for the next student, the question order will change again.) When using the randomorder document-class option only the questions appear in random order. If the quiz contains multiple choice questions then the choices are *not* randomly permuted. That is, the choices always appear in the order that they are written in the LaTeX file.

hidesidemenu, sidemenu

If the hidesidemenu option is set then the side menu on the left-hand side of the quiz web page will not be displayed when the quiz first loads. By default, the side menu appears unless the screen size is too small, such as on a mobile phone. Many examples of the hidesidemenu and showsidemenu class options can be found above.

The display of the side menu can is also toggled by clicking on the ✖ and ❖ symbols. The side-menu automatically disappears for devices with narrow screens, such as mobile phones.

language=<lang>

Set the language used by the *web pages* constructed by WebQuiz. The following languages are currently supported by WebQuiz:

> Czech, English, French, German, Greek, Italian, Japanese, Mandarin, Russian, Spanish and Swedish

The languages files are used to print the various buttons and text that is generated on the web pages constructed by WebQuiz. The language option does not affect the DVI or PDF versions of the quiz and it does not load language packages like babel or polyglossia.

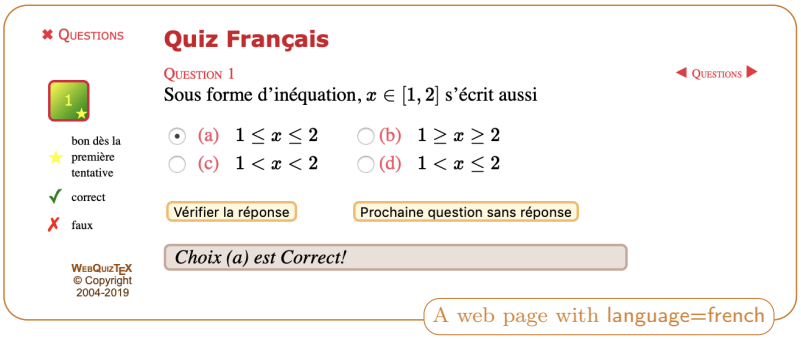
The language keyword can be in upper or lower case, with the result that either (but not both!) of the following two lines set the quiz language to German:

```
\documentclass[language=German]{webquiz}
\documentclass[language=german]{webquiz}
```

Typical usage of the `language` option is the following:

```
\documentclass[language=french]{webquiz}
\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\title{Quiz Français}
\begin{document}
  \begin{question}
    Sous forme 'dinéquation, $x\in[1,2]$ 'sécrit aussi
    \begin{choice}[columns=2]
      \correct $1\le x\le 2$
      \incorrect $1\ge x\ge 2$
        \feedback 'Lordre croissant est-il respecté?
      \incorrect $1<x<2$
        \feedback 'Lintervalle est-il ouvert ou fermé?
      \incorrect $1<x\le 2$
        \feedback Les bornes sont-elles incluses ou exclues?
    \end{choice}
  \end{question}
\end{document}
```

This produces a web page like this:



As a general rule, LaTeX and TeX4ht do not cope well with unicode characters, so if your quiz contains (a lot of) unicode characters then we recommend using LuaLaTeX or XeLaTeX, which corresponds to the WebQuiz command-line options `webquiz -x` or `webquiz -l`, respectively. The default TeX engine can be set in the `webquizrc` file (Section 4.2).

The language files were created largely using google translate so they may well need fine-tuning[3]. You can use `kpsewhich` to look at the language files, which all have names of the form `webquiz-xxx.lang`, where `xxx` is the name of the language in lower case. For example, the *English* language file, which is the default, can be found using the command:

```
> kpsewhich webquiz-english.lang
```

The file `webquiz-english.lang` contains the following:

```
answer             = Answer
check answer       = Check answer
choice             = Choice {} is
correct            = Correct!
false              = false
incorrect          = Incorrect.
multiple incorrect = For example, choice {} should be
next question      = Next question
```

[3]The word "Copyright" in the left-hand side-margin is not translated but perhaps it should be.

```
next question      = Next unanswered question
one mistake        = There is at least one mistake.
previous question  = Previous unanswered question
question           = Question
questions          = Questions
quiz               = Quiz
quizzes            = Quizzes
side menu cross    = wrong
side menu star     = right first attempt
side menu tick     = right
true               = true
try again          = Please try again
```

In these files, the material to the left of the equals signs are effectively variables, and so they should never be changed, or deleted, whereas anything to the right of the equals signs is the text that will appear on the WebQuiz web pages. *The pairs of braces,* `{}`, *in the language files must be present because in the online quizzes they expand to expressions like* `(a)`, `(b)`...

To add WebQuiz support for a new language, say language `xxx`, copy any WebQuiz language file to a new file `webquiz-xxx.lang` and then translate all of the words to the right of the equals signs.[4] WebQuiz will be able to find the new language file as long as it appears in the LaTeX search path[5]. Once the new language file `webquiz-xxx.lang` is in the LaTeX search path it can be used by WebQuiz using `language=xxx` as a document-class option:

```
\documentclass[language=xxx]{webquiz}
\begin{document}
     ...quiz code here...
\end{document}
```

Please submit any new language files, or corrections to existing language files, as a *new issue* at: github.com/AndrewAtLarge/WebQuiz/issues.

onepage, separatepages

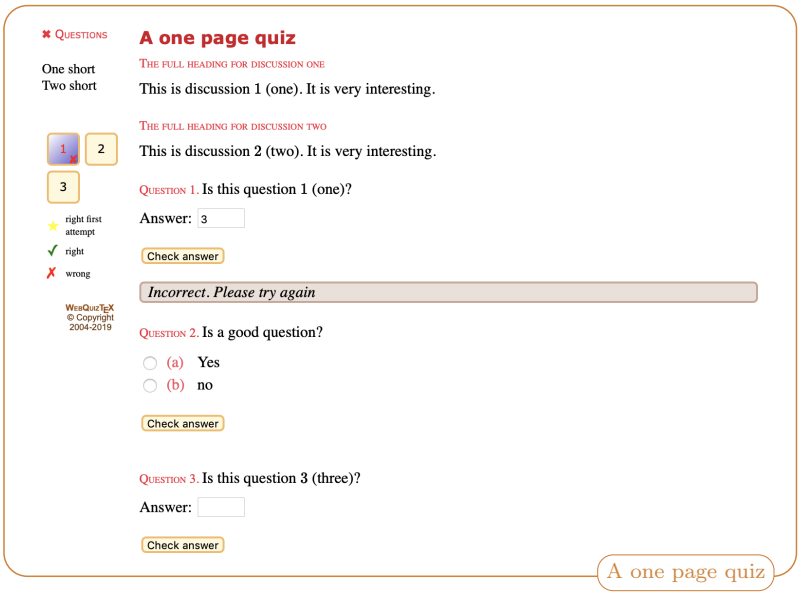
By default, only one question, or one discussion environment, is displayed by the quiz at any time. As `separatepages` is the default, every example so far is of this form. With the document-class option `onepage` all questions, and discussion environments, are displayed at the same time on a single web page. So, for example, the code:

```
\documentclass[onepage]{webquiz}
\title{A one page quiz}
\begin{document}
  \begin{discussion}[One short][The full heading for discussion one]
        This is discussion $1$ (one). It is very interesting.
  \end{discussion}
  \begin{discussion}[Two short][The full heading for discussion two]
     This is discussion $2$ (two). It is very interesting.
  \end{discussion}
  \begin{question} Is this question $1$ (one)? \answer{1}\end{question}
  \begin{question} Is a good question?
    \begin{choice}\correct Yes \incorrect no\end{choice}
  \end{question}
  \begin{question} Is this question $3$ (three)? \answer{3}\end{question}
\end{document}
```

produces the quiz page:

[4] WebQuiz assumes that all language names are in lower case so `xxx`, and not `XXX`, should be used.

[5] To help LaTeX/WebQuiz fined your language file you may need to run a program like `mktexlsr` using an administrators account



theme

WebQuiz has a small number of different themes for setting the colours on the quiz web pages. The theme can be set as an option to the document class or in the `webquizrc` file (Section 4.2). Most, but not all, of the examples so far have used the `default` theme. WebQuiz currently supports the following themes:

> blue, darkblue, darkred, default, earthy, fresh, light, lively, muted, sleek, spring and vibrant

Example screenshots of all WebQuiz themes can be found in Appendix A.

2.3 Including graphics and using pstricks and tikz

It is also possible to include complicated diagrams in WebQuiz quizzes using packages like `tikz` and `pstricks`. *As there have been several recent updates to these packages it is advisable to install the latest version of both of these packages, as well as the packages `make4ht`, `pgf` and TEX4ht.* In fact, it is recommended that you update all installed TEX packages.

By far the easiest way to include images when using WebQuiz is by adding the following lines to your document preamble:

```
\usepackage[dvipdfmx]{graphicx}
\DeclareGraphicsExtensions{.png}
```

to your document preamble. You need to use `\DeclareGraphicsExtensions` to tell WebQuiz the different types of images you are using, so the code above works for `png` images. More generally, you can use a comma separated list of extensions, such as:

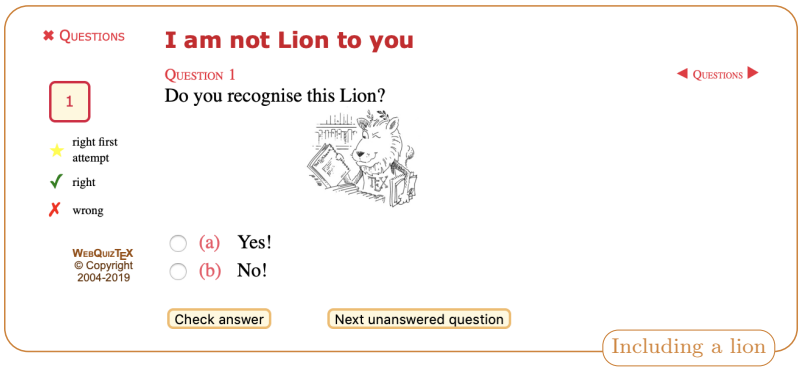
```
\DeclareGraphicsExtensions{.png, .jpg, .gif}
```

The option `dvipdfmx` to `graphicx` is only necessary if you want to be able to rescale images. For example, the code:

```
\documentclass{webquiz}
\usepackage[dvipdfmx]{graphicx}
\DeclareGraphicsExtensions{.jpg}
\title{I am not Lion to you}
\begin{document}
  \begin{question}
    Do you recognise this Lion?
    \begin{center}
```

```
        \includegraphics[height=30mm]{ctanLion.jpg}
    \end{center}
    \begin{choice}
      \correct Yes!  \feedback This is the \TeX{} lion!
      \incorrect No! \feedback You are legally required to admire this lion!
    \end{choice}
  \end{question}
\end{document}
```

produces the quiz page:

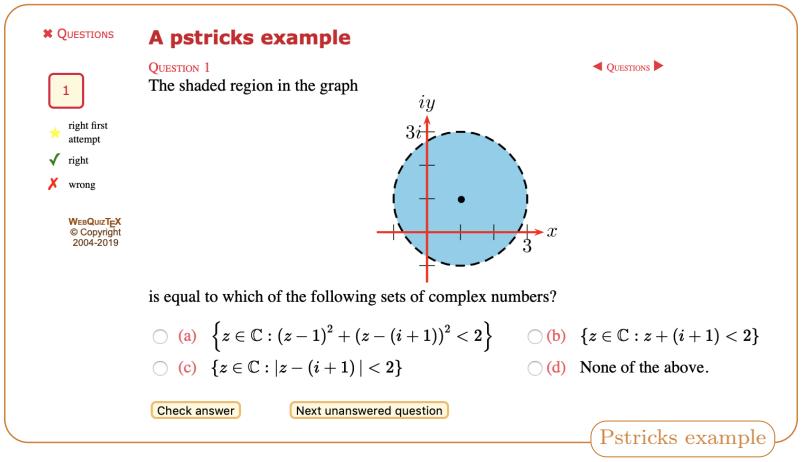


Note that WebQuiz assumes that all images are `SVG` images by default so it is necessary to give the full filename in any `\includegraphics` command.

Using `pstricks` is often just as easy, such as the following code that works out of the box:

```
\documentclass[svgnames]{webquiz}
\usepackage{pst-all}
\newcommand\C{\mathbb{C}}
\title{A pstricks example}
\begin{document}
  \begin{question}
    The shaded region in the graph
    \begin{center}
      \begin{pspicture}[unit=0.6cm](-3,-1.5)(3,4)
        \pscircle[linewidth=1pt,linestyle=dashed,
                  fillcolor=SkyBlue,fillstyle=solid](1,1){2}
        \psaxes[linecolor=red,linewidth=1pt,labels=none]
                 {->}(0,0)(-1.5,-1.5)(3.5,3.5)
        \rput(3.75,0){$x$}  \rput(0,3.85){$iy$}
        \rput(3,-0.4){3}    \rput(-0.4,3){3$i$}
        \psdots(1,1)
      \end{pspicture}
    \end{center}
    is equal to which of the following sets of complex numbers?
    \begin{choice}[columns=2]
      \incorrect $\{z \in \C : (z-1)^{2}+(z-(i+1))^{2}<2\}$
      \incorrect $\{z \in \C : z+(i+1)<2\}$
      \correct   $\{z \in \C : |z-(i+1)|<2\}$
      \incorrect None of the above.
    \end{choice}
  \end{question}
\end{document}
```

to produce the web page:



Even though tikz and pstricks can be used in WebQuiz quizzes, both of these packages sometimes have problems. WebQuiz tries to solve some these problems for you if you use the pst2pdf or tikz document-class options, which we now describe.

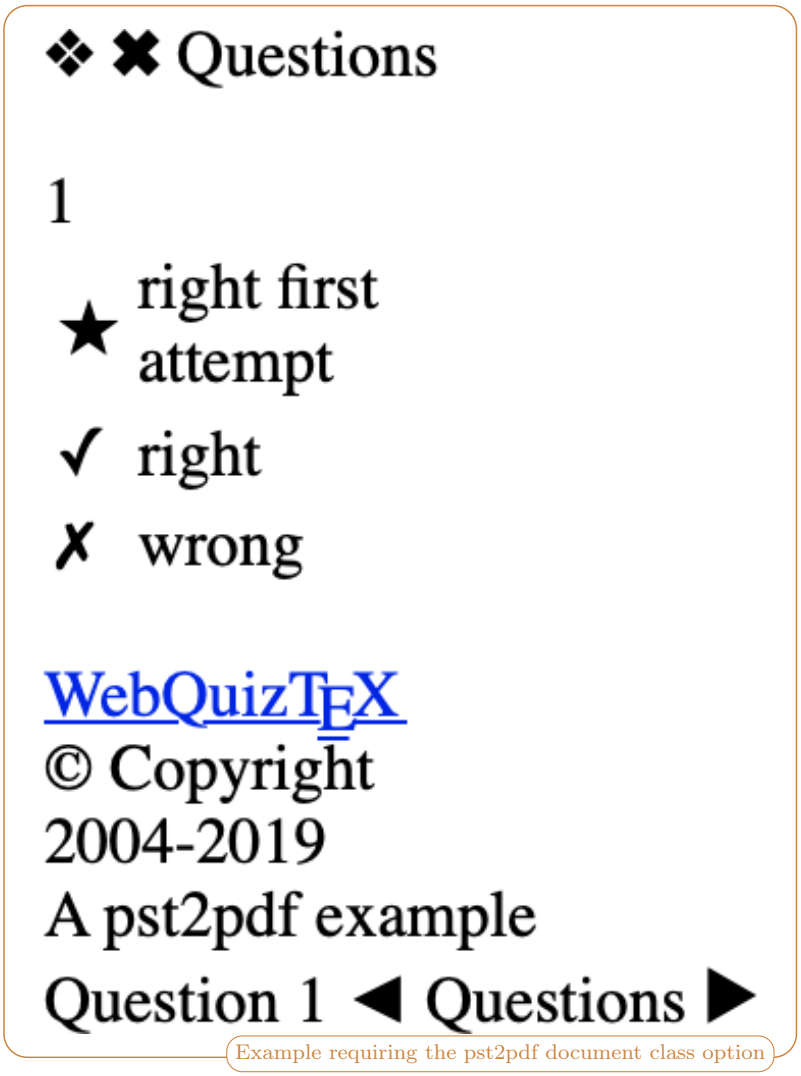
pst2pdf

For the most part, pstricks drawings display correctly. When they do fail they can sometimes be salvaged using pst2pdf. Applying pst2pdf to a WebQuiz quiz is not completely straightforward, so WebQuiz provides the document-class option pst2pdf to automatically apply pst2pdf as part of the quiz web page build process. If your pstricks drawings do not display correctly it is worthwhile to see if pst2pdf fixes the problems.

For example, the following quiz compiles only with the pst2pdf document-class option:

```
\documentclass[pst2pdf]{webquiz}
\usepackage{pst-all,pst-3dplot}
\title{A pst2pdf example}
\begin{document}
  \begin{question}     % a quiz question
      Which of the equations below could describe the following surface?

        \begin{pspicture*}(-4,-4)(4,4)
          \psplotThreeD[linecolor=blue, plotstyle=curve, drawStyle=yLines,
                      yPlotpoints=30, xPlotpoints=30, linewidth=1pt](-4,1)(-4,1){
                        x dup mul y dup mul add 1.01 exp}
          \pstThreeDCoor[linewidth=1pt, xMin=-4,xMax=4,yMin=-4,yMax=4,zMin=-2,zMax
              =6]
        \end{pspicture*}

      \begin{choice}
        \incorrect \( z=\log(x^2+y^2) \)
        \correct   \( z=e^{x^2+y^2} \)
        \incorrect \( z=1-e^{x^2+y^2} \)
        \incorrect \( z=\dfrac1{x^2+y^2}\)
      \end{choice}
  \end{question}
\end{document}
```

to produce the quiz:



Example requiring the pst2pdf document class option

The position of the image adjusts with the screen size and it does, in fact, display well on a mobile device. WebQuiz is not able to display this image without the `pst2pdf` document-class option.

Unfortunately, `pst2pdf` can fail silently without giving any warnings. If you plan to use the `pst2pdf` document-class option then you should first check to make that the `pst2pdf` package and executable is properly installed. According to the `pst2pdf` manual:

> `pst2pdf` needs Ghostscript (>=9.14), perl (>=5.18), pdf2svg, pdftoppm and pdftops (from poppler-utils or xpdf-utils) to process a file using `pst2pdf`.

If using `pst2pdf` does not produce an image then, rather than `pst2pdf` not working, the problem might be that you have not installed all of the programs that `pst2pdf` relies upon, so look in your log files for error messages and check that all of the programs listed above are correctly installed, with the specified version numbers. See also Section 3.4.

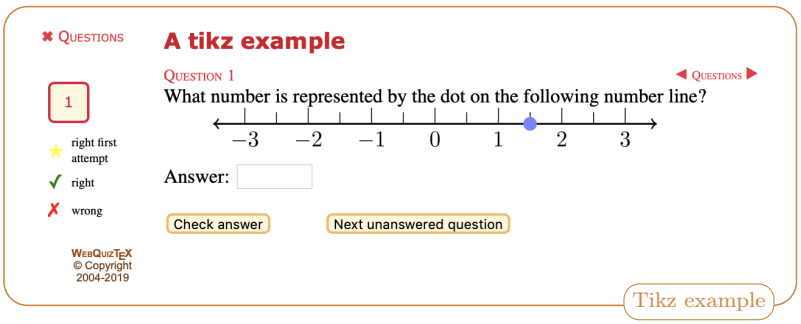
tikz

Giving this class option both loads the `tikz` package and it fixes several issues with PGF that prevent it from working with TEX4ht. It is important that you use the WebQuiz `tikz` document-class option, and not `\usepackage{tikz}`, because WebQuiz loads slightly different configuration files for `tikz` that are optimised for use with TEX4ht. *Please note that for*

tikz you need to use TEXLive 2018 with all packages updated. Thanks are due to Michal Hoftich for the enormous amount of effort that he has put into making TEX4ht and tikz more compatible. As an example, the quiz file

```
\documentclass[tikz]{webquiz}
\usepackage{tikz}
\title{A tikz example}
\begin{document}
  \begin{question}     % a quiz question
     What number is represented by the dot on the following number line?
     \begin{center}
       \begin{tikzpicture}
          \foreach \x in {-3,...,3} { \draw(\x,0.25) --(\x,0)node[below]{$\x$}; }
          \foreach \x in {-2.5,...,2.5} { \draw(\x,0.18) --(\x,0); }
          \draw[thick,<->](-3.5,0)--(3.5,0);
          \filldraw[blue!50!white](1.5,0) circle (1mm);
       \end{tikzpicture}
     \end{center}
     \answer[number]{1.5} % inserts an answer box and specifies the answer as 1.5
     \whenRight Correct!
     \whenWrong Incorrect!
  \end{question}
\end{document}
```

produces the online quiz:



Most people use either pstricks or tikz. A quiz that tries to use both pstricks and tikz will probably not compile.

2.4 Configuring commands and environments for TEX4ht

The underlying engine used by WebQuiz is TEX4ht so, because TEX4ht is not able process all LATEX code, there is LATEX code that WebQuiz is not able to cope with. This said, TEX4ht is able to display *most* LATEX code and WebQuiz has been used to write literally thousands of quiz questions so it is likely that you will be able to typeset what you want in your online quizzes. In particular, as discussed in Section 2.3, it is possible to use tikz and pstricks in WebQuiz quizzes.

WebQuiz uses TEX4ht to convert the quiz content from LATEX to HTML. If TEX4ht has not been configured to for some of the commands or environments that you are using then they may not display correctly in your online quizzes. The "correct" way to fix such problems is to write appropriate TEX4ht configuration commands, however, this can be tricky to do — especially if you are not familiar with the inner workings of TEX and TEX4ht.

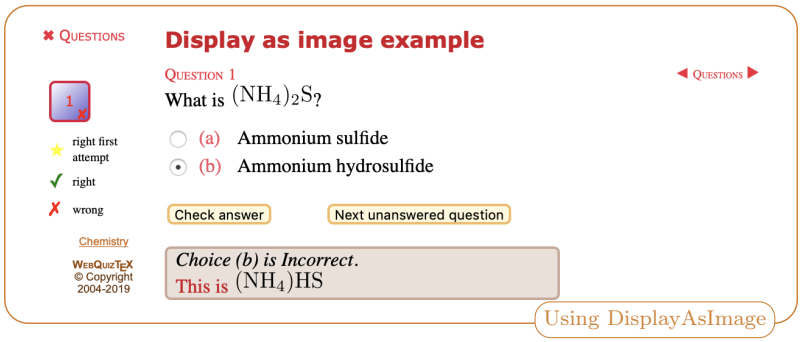
As a workaround, WebQuiz provides the command \DisplayAsImage that, in effect, tells TEX4ht to treat your command as an image when it creates the web page. This is an easy work-around that often produces good results – and it is much easier than writing your own TEX4ht configuration commands.

For example, the mhchem package is a powerful package that defines a macro \ce for writing chemical symbols but, unfortunately, the \ce macro has not (yet) been configured to work with TEX4ht, which means that this command does not work very well when used in WebQuiz quizzes.

For example, the following code:

```
\documentclass{webquiz}
\usepackage[version=4]{mhchem}
\Department{Chemistry}
\UnitCode{Chem101}\UnitURL{/chem101}
\title{Display as image example}
\begin{document}
  \DisplayAsImage[style="vertical-align: middle;"]{ce}
  \begin{question}
    What is \ce{(NH4)2S}?
    \begin{choice}[columns=1]
      \correct Ammonium sulfide
      \incorrect Ammonium hydrosulfide
      \feedback This is \ce{(NH4)HS}
    \end{choice}
  \end{question}
\end{document}
```

shows that if you add the line \DisplayAsImage{ce} to your quiz after \begin{document} then it is possible to use \ce in your quizzes:



As the example code shows, \DisplayAsImage accepts an optional argument that can be used to fine-tune the placement of the image on the quiz web page using CSS. For those interested in the technical details, the definition of \DisplayAsImage is:

```
\RequirePackage{etoolbox}
\renewcommand\DisplayAsImage[2][]{%
  \csletcs{real:#2}{#2}%
  \NewConfigure{#2}{2}
  \csdef{#2}##1{\Picture+[#1]{}\csuse{real:#2}{##1}\EndPicture}
  \Configure{#2}{\Picture+[#1]{}}{\EndPicture}
}
```

3 System requirements, installation and configuration

WebQuiz takes a LATEX file and translates it into a functional web page. To use WebQuiz the quiz author only needs to know how to use LATEX *and* to have all of the programs used by WebQuiz installed. Fortunately, most of the system requirements will already be installed on a system with an up-to-date installation of TEXLive, however, some tweaking may still be necessary.

Behind the scenes, WebQuiz uses TEX4ht, python, Javascript and several other tools to construct and operate the online quizzes. The WebQuiz program has three main components:

- LATEX files (a class file and TEX4ht configuration files)

- A Python3 program that uses TEX4ht to convert LATEX files into web pages
- Web files (Javascript, CSS and online documentation)

Of course, to use the online quizzes created by WebQuiz you need a web server. To use WebQuiz all of these files need to be in appropriate places. Fortunately, `ctan` takes care of most of this but the web-related files still need to be put onto your web server.

WebQuiz has been tested extensively on Linux and Mac operating systems. Several people have used WebQuiz on windows computers, but I have not tested the program on a windows computer myself.

3.1 System requirements

In order to work WebQuiz needs the following programs to be installed on your system:

- An up-to-date LATEX distribution, such as that provided by TEXLive. In particular, you need to have TEX4ht and `make4ht` installed.
  Unfortunately, the version of `make4ht` that was released with TEXLive 2018 had some bugs and there have been many recent changes to `make4ht` and `pstricks`, so it is strongly recommended that you update all packages from `ctan` before you try and use WebQuiz.
- Python 3. As of writing python 3.7.2 is available.
- Javascript
- If your quizzes use `pstricks`, or if you want to compile the Online manual (Appendix B), then you need to ensure that Ghostscript and `dvisvgm` are installed and properly configured; see Section 3.4 for more details.
- A web server. As detailed in Section 3.3, you will need to install the web components of WebQuiz.

3.2 Installing WebQuiz

WebQuiz will be included as a package in TEXLive 2019 and, once a packaging issue is sorted out, it will be included in MiKTEX. If WebQuiz is available as a package from either of these distributions then it can be installed in the usual way.

Alternatively, it is possible to download the WebQuiz zipfile from `ctan` . To install WebQuiz from the zipfile you should:

1. Download the WebQuiz zipfile from `ctan` .
2. Unzip the file
3. Change directory of the `webquiz/scripts` subdirectory from the zipfile
4. From an administrators account, or using `sudo` on a unix-like system, run he command:

```
> webquiz --tex-install
```

   On windows, use:

```
> webquiz.bat --tex-install
```

This will install the different components of WebQuiz to their "standard" locations in he TEX search tree. If you are using MiKTEX you may need to open the MiKTEX console, go to the Tasks menu and then rebuild the TeX filename database.

Once the WebQuiz package is installed you should initialise thew package, following the steps given in Section 3.3.

3.3 Initialising WebQuiz

WebQuiz is a tool for creating online quizzes and in order for it to work efficiently various files (javascript and cascading style sheets) need to put onto your web server. WebQuiz has an *initialisation* routine for installing the web components of the program. In fact, until WebQuiz has been initialised it will ask you if you to run the initialisation routine every time you use WebQuiz. You can reinitialise WebQuiz at any time using the command-line option:

```
> webquiz --initialise
```

WebQuiz will actually work without being initialised, however, any quiz web pages that are created before initialisation will be emblazoned with a message reminding you to initialise WebQuiz.

The location of the files on the web server depends both on the operating system that is running on your computer and how your web server has been configured. It is essential that the WebQuiz files are installed in a directory that is accessible from the web. It does not matter if they are put into a user web directory or into a system web directory. If in doubt please consult your system administrator.

Common locations for the *web root* of the server are:

Operating system	Root of web server
Mac OSX	`/Library/WebServer/Documents`
Linux	`/var/www/html`
Windows	`C:\inetpub\wwwroot`

WebQuiz needs to copy several files into a subdirectory of this *web root*. When you run

```
> webquiz --initialise
```

you will be prompted for the following:

- The location of the WebQuiz web directory, which needs to be a directory that is visible to your web server
- The relative URL for this directory, which tells your web browser where to find these files

For example, on my system the web root for our web server is `/usr/local/httpd/` and the WebQuiz web files are in the directory `/usr/local/httpd/UoS/WebQuiz`. So, I set:

WebQuiz web directory = `/usr/local/httpd/UoS/WebQuiz`
WebQuiz relative URL = `/UoS/WebQuiz`

Once the initialisation step is complete, WebQuiz is ready to use although you, possibly from an administrators account or using `sudo`, may also want to run:

```
> webquiz --edit-settings
```

This will talk you the process of setting system defaults for the quizzes that, for example, specify the name and URL for your department and institution as well as the default language and theme used for the quizzes. If in doubt about any of the option press return to accept the default. See Section 4.2 for more details.

You can test your WebQuiz installation by compiling the example files from the WebQuiz manual. You can find these files in the directory `web_root/doc/examples`, where the `web_root` is the directory where you just installed the WebQuiz web files. If Ghostscript and `dvisvgm` are installed and properly configured (see Section 2.3) then you should also be able to compile the Online manual (Appendix B) using WebQuiz.

To install the WebQuiz files for general use on your system, or to save system wide settings, you need to run the initialisation command `webquiz --initialise` *using an administrator account or using* `sudo` *on a UNIX or Mac OSX system.*

If you have already saved a *user* `webquizrc` file (Section 4.2) file then to change the *system* `webquizrc` file you will use need to use the `--rcfile` command-line option:

```
> webquiz --rcfile /path/to/system/rc-file --initialise
```

Remark To remove all WebQuiz files from your web server use:

```
> webquiz --uninstall
```

3.4 Graphics and dvisvgm

WebQuiz, via TEX4ht, uses dvisvgm to convert certain images to Scalable Vector Graphics (SVG). This is done using his is done using dvisvgm. At first sight this is OK because dvisvgm is included in TEXLive and MiKTEX, however, dvisvgm uses Ghostscript and this needs to be correctly configured and, as outlined in FAQ, dvisvgm needs to know where to find the Ghostscript libraries. For example, to get dvisvgm to work on my system I needed to add the line

```
export LIBGS=/usr/local/lib/libgs.dylib
```

to my .bashrc file. To see whether you need to do something similar on your system you need to look at the output from the following two commands:

```
> dvisvgm -h
> dvisvgm -l
```

There are three possibilities:

- the -h output does not contain -libgs and the -l output contains ps: ghostscript was linked at build time, so everything should work
- the -h output contains -libgs and the -l output does not contain ps: gpostscript support is enabled but ghostscript is not linked. You need to locate the ghostscript library libgs.so or libgs.dylib on your system and set the LIBGS environment variable, or equivalent, accordingly
- the -h output does not contain -libgs and the -l output does not contain ps: dvisvgm was not built with postscript support. In this case, WebQuiz will not be able to process svg images. You need to reinstall dvisvgm with ghostscript support.

4 The WebQuiz program

The WebQuiz program was designed to be run from the command-line. To produce an online quiz from a WebQuiz LATEX file quiz.tex type:

```
> webquiz quiz          or          > webquiz quiz.tex
```

One feature of WebQuiz is that you can process more than one quiz file at a time. For example, if you have quiz files quiz1.tex, ..., quiz9.tex in the current directory then, on a UNIX system, you can rebuild the web pages for all of these quizzes using the single command:

```
> webquiz quiz[1-9].tex
```

This is useful if some generic aspect of all of the quizzes has changed, such as the theme, the language, thebreadcrumbs or a department URL. In fact, one would probably use

```
> webquiz -dqq quiz*.tex
```

because the webquiz --qq command-line option suppresses almost all of the output produced by LATEX and TEX4ht and -d, which is *draft mode*, is faster and it is probably sufficient if the quizzes were compiled recently. The next section discusses the WebQuiz command-line options.

4.1 Command-line options

Typing webquiz -h, or webquiz --help on the command-line gives the following summary of the WebQuiz command-line options:

```
usage: webquiz [-h] [-q] [-d] [-s] [--latex | -l | -x] [-r RCFILE]
               [-i | -e | --settings [SETTINGS]]
               [quiz_file [quiz_file ...]]

A LaTeX package for writing online quizzes
```

```
positional arguments:
  quiz_file             latex quiz files

optional arguments:
  -h, --help            show this help message and exit
  -q, --quiet           Suppress tex4ht messages (also -qq etc)
  -d, --draft           Use make4ht draft mode
  -s, --shell-escape    Shell escape for tex4ht/make4ht
  --latex               Use latex to compile document with make4ht (default)
  -l, --lua             Use lualatex to compile the quiz
  -x, --xelatex         Use xelatex to compile the quiz
  -r RCFILE, --rcfile RCFILE
                        Specify location of the webquiz rc-file
  -i, --initialise, --initialize
                        Install web components of webquiz
  -e, --edit-settings   Edit default settings for webquiz
  --settings [SETTINGS]
                        List default settings for webquiz
```

The command-line options are listed on separate lines above to improve readability. In practice, the different options need to be one the same line, although they can appear in any order.

We describe the different options, grouping them according to functionality.

-h, –help
list the command-line options and exit

-q, -qq, --quiet
Suppress LaTeX and TeX4ht error messages: `-q` is quiet and `-qq` is very quiet. If you use `webquiz -qq texfile.tex` then almost no output will be printed by WebQuiz when it is processing your quiz file. Be warned, however, that both of these options can make it harder to find and fix errors, so using the `-q` and `-qq` options is recommended if your file is known to compile.

TeX options

-d, --draft
draft mode. The LaTeXfile is processed only once by `htlatex`, which gives a much faster compilation time but cross references etc may not be completely up to date. This is equivalent to using: `make4ht --mode draft`

-s,--shell-escape
Shell escape for LaTeX/htLaTeX/make4ht

--latex
Use LaTeX to compile the quiz (the default)

-l,--lua
Use luaLaTeX to compile the quiz

-x, --xelatex
Use xeLaTeX to compile the quiz

Settings and configuration

-r [RCFILE], --rcfile [RCFILE]
Specify the location of the `webquizrc` file (Section 4.2) file to use. This setting is only necessary if you want to override the default `webquizrc` file.

-i, --initialise
Initialise files and settings for webquiz. See Section 3.3 for more details.

--edit-settings

Edit the webquiz settings in the WebQuiz rc-file. See Section 4.2 for more details.

If you do not have permission to write to the system rc-file, which is in the WebQuiz scripts directory, then you will be given the option of saving the WebQuiz settings to an rc-file in your home directory or another file of your choice. If you want to save the settings to a particular rc-file use the `--rcfile` *option. If you want to change the system* `webquizrc` *file (Section 4.2) then use a administrator account or, on a unix-like system,* use `sudo webquiz --edit-settings`.

--settings [SETTING]

List system settings for webquiz stored in `webquizrc` file (Section 4.2), the (*run-time configuration file*). Optionally, a single `SETTING` can be given in which case the value of only that setting is returned. If `SETTING` is omitted then the list of current settings are printed. Use `SETTING=verbose` for a more verbose listing of the settings and `SETTING=help` for a summary of the settings.

The rc-file can be edited by hand, however, it is recommended that you instead use

```
webquiz --edit-settings
```

Typical settings returned by this command look like:

```
------------------------------------------------------------------------
WebQuiz rc-file: /usr/local/texlive/texmf-local/tex/latex/webquiz/webquizrc
------------------------------------------------------------------------
breadcrumbs       =
department        =
department-url    = /
engine            = latex
hide-side-menu    = false
institution       =
institution-url   = /
language          = english
one-page          = false
random-order      = false
theme             = default
version           = 5.2
webquiz-url       = /WebQuiz
webquiz-www       = /Library/WebServer/Documents/WebQuiz
make4ht           =
mathjax           = https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js
webquiz-layout    = webquiz_layout
------------------------------------------------------------------------
```

See Section 4.2 for more details.

Advanced command-line options

Change these settings with care: an incorrect value for these settings can stop WebQuiz from working.

--make4ht MAKE4HT-OPTIONS

Options to be passed to `make4ht` when converting the LaTeX to XML. This option is equivalent to setting the `make4ht` in the `webquizrc` file (Section 4.2); see Section 4.2. At least under UNIX, multiple arguments should be enclosed in quotes. For example, to give `make4ht` a custom mk4 file (note that `myquiz.mk4` is included, if it exists), you would use

```
> webquiz --make4ht "-e file.mk4" myquiz.tex
```

The `make4ht` command-line option will be required only in rare instances.

--uninstall
: Remove all WebQuiz files from your web server directory. This command only removes files that WebQuiz may have installed on your web server. It does *not* remove WebQuiz from your LaTeX distribution.

--webquiz-layout WEBQUIZ-LAYOUT
: Name of (local) python code that controls the layout of quiz web page. This option is equivalent to setting the `webquiz-layout` in the `webquizrc` file (Section 4.2). See Section 4.3 for more details.

Other options

The following command-line options are mainly useful mainly for code developers.

--version
: Prints the WebQuiz version number and exit (currently 5.2)

--tex-install
: Use this command-line option if you are installing the LaTeX components of WebQuiz from the `ctan` zip-filei (see Section 3.2). It will install the different LaTeX components of WebQuiz into `TEXMFMAIN`. If you installed WebQuiz as part of a TeX distribution, such as TeXLive, then you do not need to use this option.

--tex-uninstall
: Use this command-line option to remove the LaTeX components of WebQuiz from your system. Only use this option if you installed WebQuiz using the `ctan` zip-file.

--debugging
: Displays extra debugging information when compiling and prevents WebQuiz from deleting the many intermediary files that are created when building the quiz web pages.

4.2 WebQuiz settings and the webquizrc file

WebQuiz stores the following default sets in `webquizrc` file, a *run-time configuration file*:

breadcrumbs	breadcrumbs at the top of quiz page (§2.1e)
department	name of department (§2.1e)
department-url	url for department (§2.1e)
engine	default tex engine used to compile web pages (Section 2.2)
hide-side-menu	do not display the side menu at start of quiz (Section 2.2)
institution	institution or university (§2.1e)
institution-url	url for institution or university (§2.1e)
language	default language used on web pages (Section 2.2)
one-page	display questions on one page (Section 2.2)
random-order	randomly order the quiz questions (Section 2.2)
theme	default colour theme used on web pages (Section 2.2)
version	webquiz version number for webquizrc settings (Section 4.2)
webquiz-url	relative url for webquiz web directory (Section 3.3)
webquiz-www	full path to webquiz web directory (Section 3.3)
make4ht	build file for make4ht (Section 3.3)
mathjax	url for mathjax (Section 3.3)
webquiz-layout	name of python module that formats the quizzes (Section 4.3)

The last three options are *advanced options* that you should change with care.

The default values of all of these settings can be overridden in the LaTeX file for the quiz, or with the command-line options. The default values can be changed at any time using

```
webquiz --edit-settings
```

When changing the settings WebQuiz tries to explain what it is doing at each step. If you are unsure what a particular setting does then *press return* to accept the default value — the default value is printed inside square brackets as `[default]`. In particular, when you first start using WebQuiz it is recommended that you accept the default values for all of the advanced options because it is very unlikely that you will need to change these initially.

The first line of output from `webquiz --settings` gives the location of the rc-file being used. The system rc-file, `webquizrc` file, is saved in the `tex/latex/webquiz` subdirectory of the `TEXMFLOCAL` directory. A typically location for this file is

```
/usr/local/texlive/texmf-local/tex/latex/webquiz/webquizrc
```

By default, the WebQuiz settings are saved here so that you do not need to reinitialise WebQuiz whenever you update your TEX distribution. If you do not have permission to write to this directory then you will be asked if you would like to save the rc-file somewhere else. The location of the user rc-file is:

- `~/.dotfiles/config/webquizrc` if the directory `~/.dotfiles/config` exists,
- `~/.config/webquizrc` if the directory `~/.config` exists,
- `~/.webquizrc` otherwise.

Each time WebQuiz is run it reads the system and user rc-files, if they exist. When using `webquiz --edit-settings` you will be promoted for a different installation location if you do not have permission to write to the specified rc-file. To use a particular `webquizrc` file use the `--rcfile` command-line option:

```
> webquiz --rcfile <full path to rc-file> ...
```

If you save the settings to a non-standard location then you will need to use the command-line option `webquiz --rcfile RCFILE` to access these settings.

To describe the WebQuiz defaults settings we consider them by category.

Institution settings

department
Sets the default department name. This can be overridden in the LaTeX file using `\Department` in §2.1e

department-url
Sets the URL for the department. This can be overridden in the LaTeX file using `\DepartmentURL` in §2.1e

institution
Sets the default institution name. This can be overridden in the LaTeX file using `\Institution` in §2.1e

institution-url
Sets the URL for the institution.This can be overridden in the LaTeX file using `\InstitutionURL` in §2.1e

Formatting options

breadcrumbs
Sets the default breadcrumbs at the top of quiz page. The default breadcrumbs can be overwritten in the quiz file using the `\BreadCrumbs` command. See §2.1e for more details.

engine
Sets the default TEX engine to be used when compiling the quiz. By default, `latex` is used, with the two other possibilities being `lua` and `xelatex`, for LuaLaTeX and XeLaTeX respectively. Behind the scenes, the two choices correspond to invoking `make4ht` with the `--lua` and `--xelatex` options, respectively. The `engine` setting in the `webquizrc` file can be overridden by the webquizrc `--latex`, `--lua` and `--xelatex`.

language
Sets the default language for the WebQuiz web pages. This can be overridden in the quiz file by using the document class `language` option: `language=xxx` See Section 2.2.

theme
Sets the default colour theme for the WebQuiz web pages. This can be overridden in the quiz file by using the document class `theme` option: `theme=xxx`. See Section 2.2.

Advanced options

make4ht
Options for `make4ht`. Rather than using TEX4ht directly, WebQuiz uses `make4ht` to convert the LATEX file to XML. Use this option to customise how `make4ht` is called. See the `make4ht manual` for more information.

mathjax
WebQuiz web pages use mathjax to render the mathematics on the quiz web pages. By default this is done by loading `mathjax` from

```
https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js
```

Fetching `mathjax` from an external site can cause a delay when the quiz web pages are loaded. This setting in the rc-file allows you to change where `mathjax` is loaded from. For example, if you install `mathjax` on your web server then you would replace this will the corresponding relative URL.

webquiz-layout
Sets the python file that writes the HTML file for the quiz. Most people will not need this option. The next subsection describes how to do this in more detail.

Configuration settings

Use `webquiz --initialise` to change these settings.

webquiz-url
This is the relative URL for WebQuiz web directory

webquiz-www
This is the full path to the WebQuiz web directory. The Online manual (Appendix B) and other example code can be found in the `docs` subdirectory. If you use `bash`, then the command

```
> cd $(webquiz --settings webquiz-www)/docs
```

will take you to the WebQuiz online `docs/` directory.

4.3 Changing the layout of the WebQuiz web pages

This is an advanced WebQuiz feature that most people will not need. To change the layout of the quiz web pages created by WebQuiz requires working knowledge of HTML and python.

The construction of the online quizzes is controlled by the python file `webquiz_standard.py`. If you want to change the structure of the web pages for the quizzes then the easiest way to do this is to make a copy of `webquiz_standard.py`, say `webquiz_myformat.py`, and then edit this file directly. This will require working knowledge of python and HTML. To give you an idea of what needs to be done, the python file `webquiz_standard` contains a single function `write_web_page` that returns the HTML for the page as a string using the following:

```
quiz_page = r'''<!DOCTYPE HTML>
<html lang="en">
<head>
  <title> {title} </title>
  {htmlpreamble}
</head>
<body>
  {no_script}{breadcrumbs}
  <div class="quiz-page">
    {side_menu}
    <div class="quiz-questions">
      {quiz_header}
      {quiz_questions}
    </div>
  </div>
  {webquiz_init}
</body>
</html>
'''
```

By changing this output you can change the layout of the quizzes produced by WebQuiz. For example, by adding code to the `<head>...</head>` section of `quiz_page` it is easy to include new CSS code and by modifying `<body>....</body>` you can change the layout of the page. More sophisticated versions of `webquiz_standard.py`, where you change the underlying python code, are possible. At the University of Sydney we have a custom version of `webquiz_standard.py` that calls our content management system and, in this way, embeds the quiz web page inside a web page that used the official "branding" required by our university.

When experimenting with a new layout can run WebQuiz using the command:

```
> webquiz --webquiz-layout webquiz_myformat quizfile.tex
```

Once the new layout is finalised you can make it the default layout by setting `webquiz-layout` equal to `webquiz_my_format` in the `webquizrc` file (Section 4.2) using `webquiz --edit-settings`.

If you do make modifications to any of these files then, by the WebQuiz Licensing agreement, you are required to create a new version of this file that has a *different name*. Doing this will also make it easier for you to integrate your changes with any future releases of WebQuiz.

4.4 Bugs, issues and feature requests

Please report any bugs, issues or feature requests using the *issue* tracker at

github.com/AndrewAtLarge/WebQuiz/issues.

When reporting a bug please provide a *minimal working example* that clearly demonstrates your problem. This should be a compilable LaTeXfile that looks something like the following:

```
\documentclass{webquiz}
\begin{document}
     ** insert problematic code here **
\end{document}
```

Bug reports that do not have a minimal working example can be hard to reproduce in which case it is not possible to fix them. Before submitting a bug export please first compile your quiz using (pdf)latex to check to see if your problem is an issue with LaTeXor with WebQuiz. If you can, please also test to see if your code compiles using TeX4ht directly.

Appendices

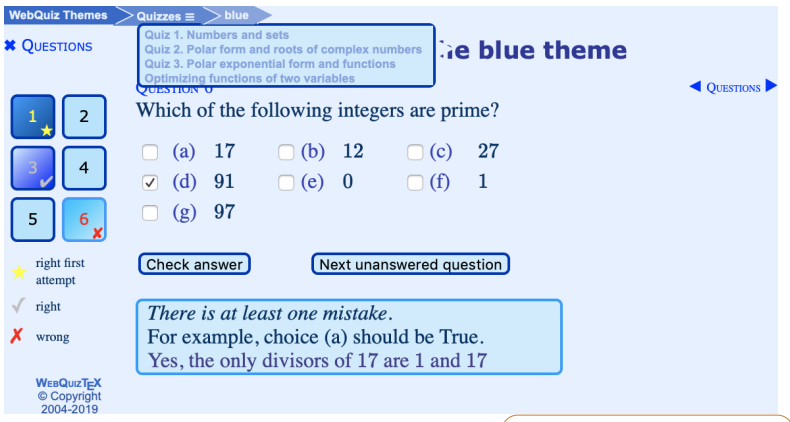
A Catalogue of web page themes

WebQuiz comes with different themes for changing the colour scheme of the online quizzes, which can be set using the `theme` document-class option or in the `webquizrc` file (Section 4.2); see Section 2.2 for more details. Themes are easy to construct in principle although finding colours that work well together can be tricky in practice so, as a result, there are some themes that I would not personally recommend!

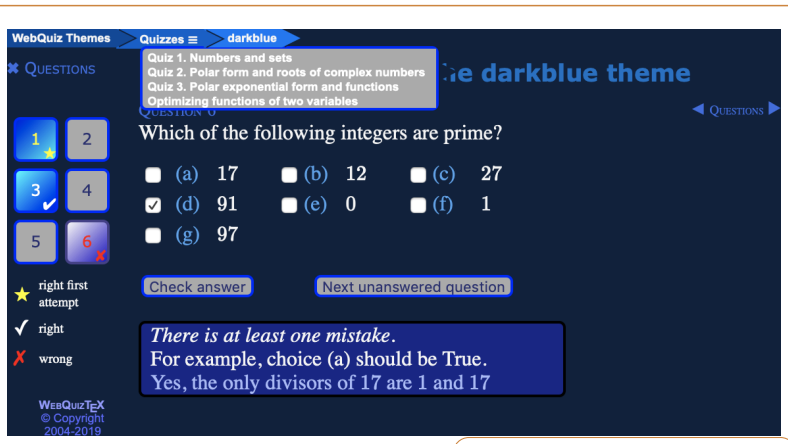
New themes, and modifications to existing themes, can be submitted as an *issue* at:

github.com/AndrewAtLarge/WebQuiz/issues/issues.

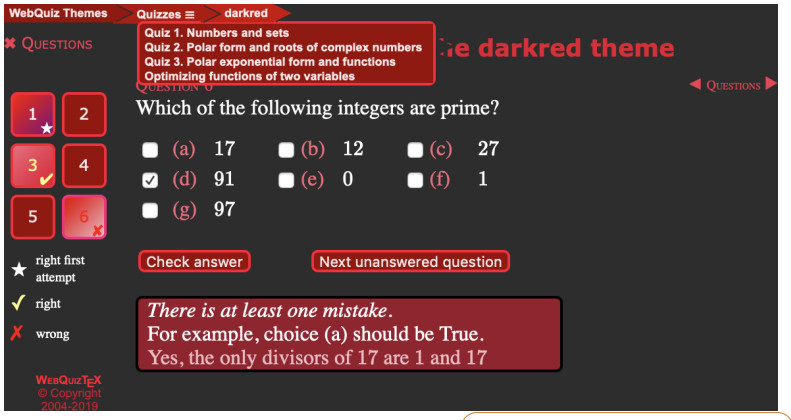
Where possible these will be incorporated into a future release of the package, although there is a potential technical issue here in that the underlying CSS files are written using sass.



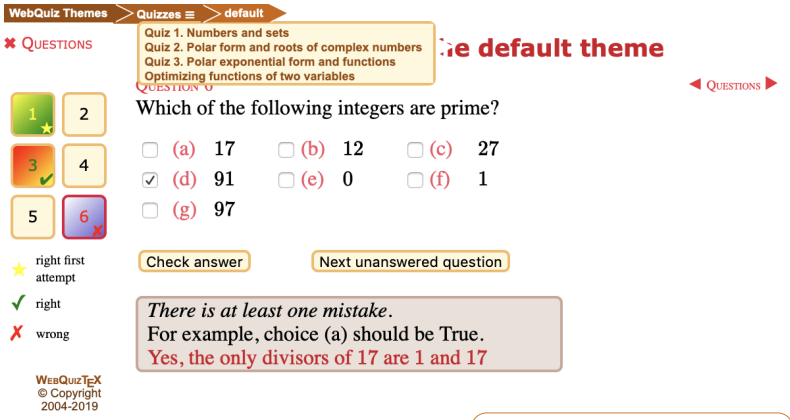
Example of the **blue** theme



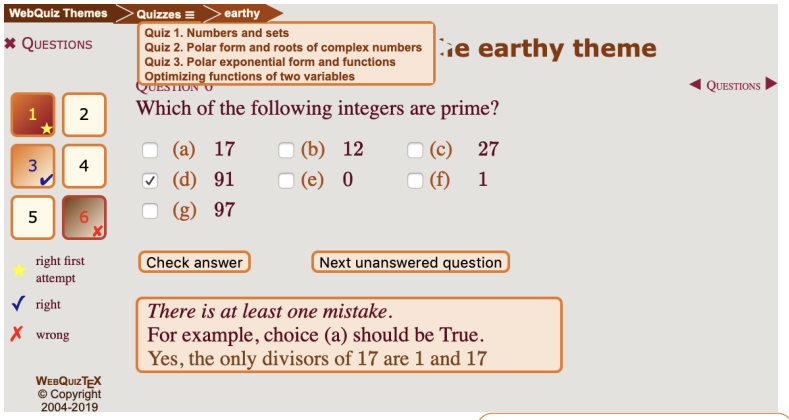
Example of the **darkblue** theme



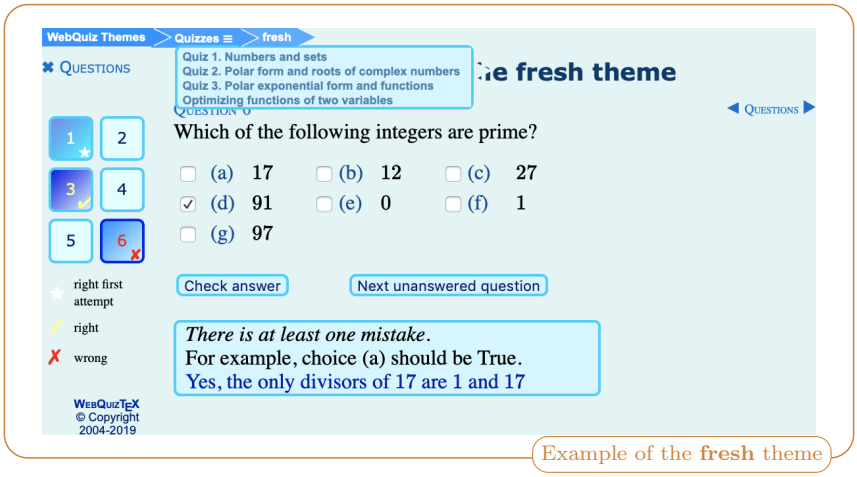
Example of the **darkred** theme



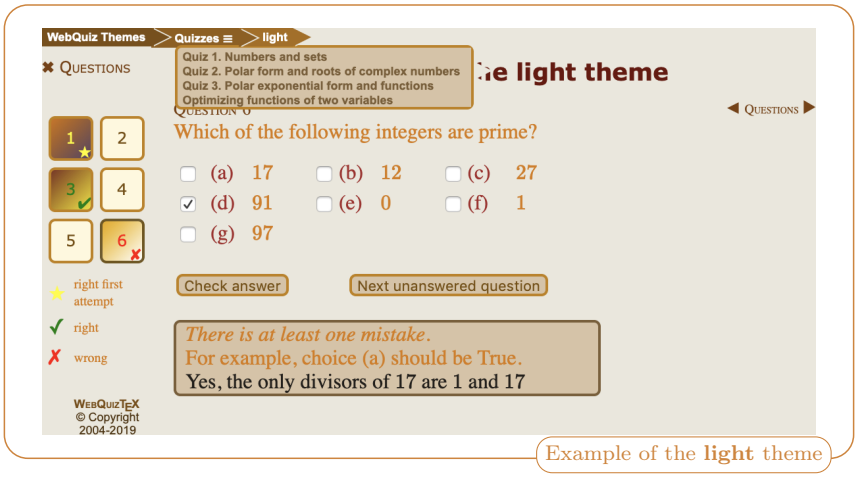
Example of the **default** theme



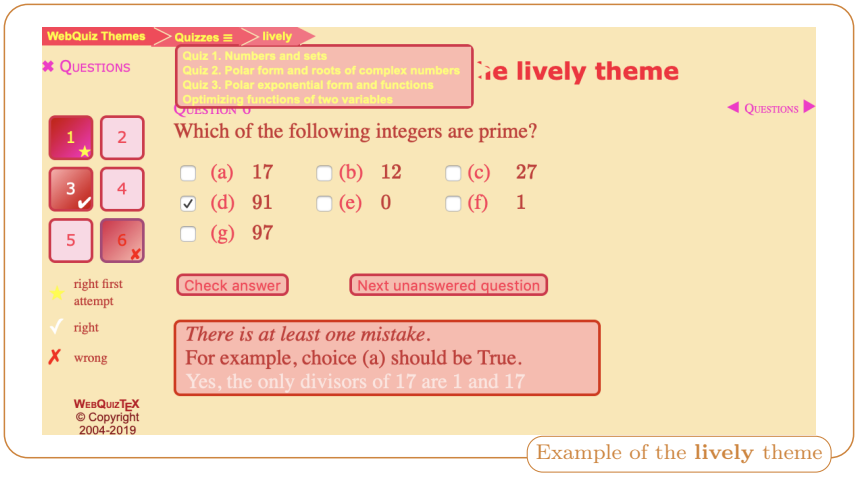
Example of the **earthy** theme



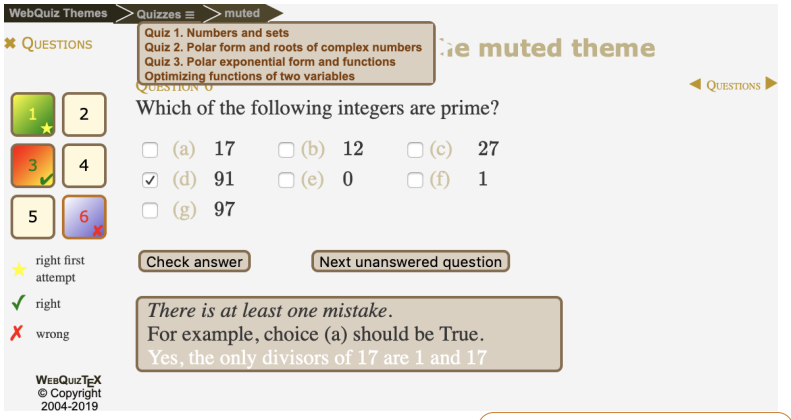
Example of the **fresh** theme



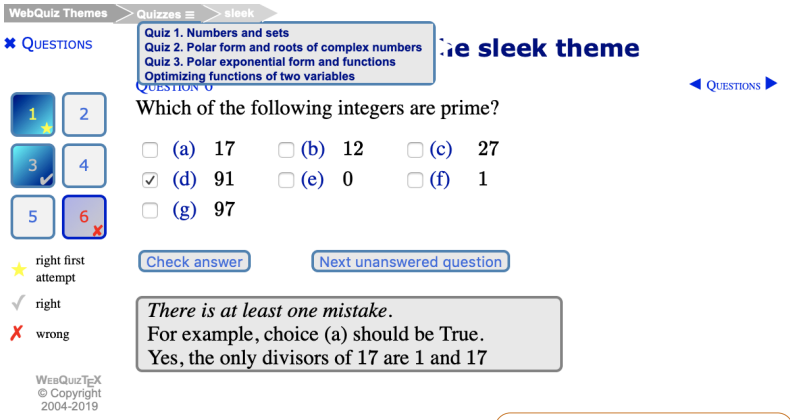
Example of the **light** theme



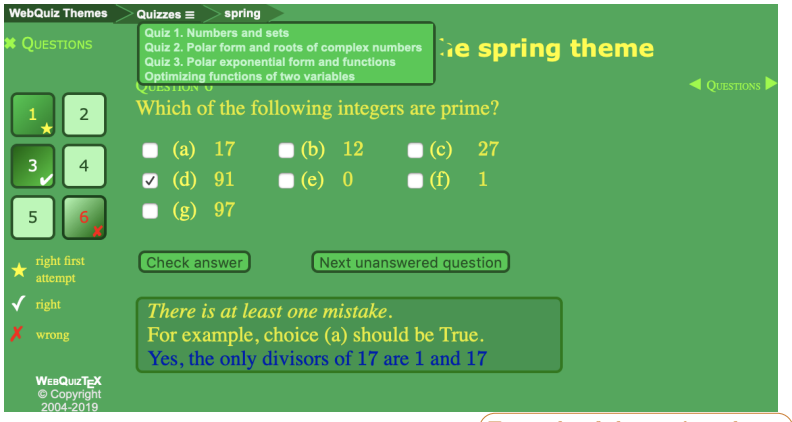
Example of the **lively** theme



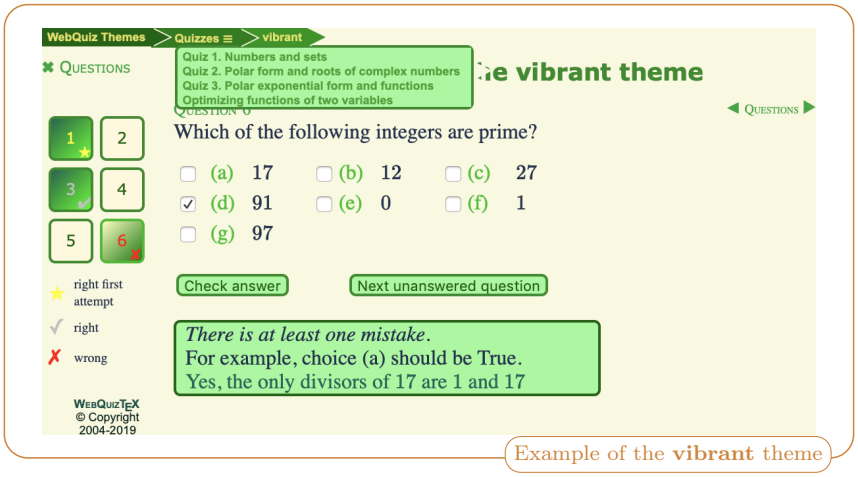
Example of the **muted** theme



Example of the **sleek** theme



Example of the **spring** theme



Example of the **vibrant** theme

B The online WebQuiz manual

WebQuiz has an Online manual that is a LaTeXfile written with the `webquiz` document class. The conversion of the manual from LaTeX to HTML is done by WebQuiz. The `PDF` version of this manual is included here as an easy reference. The source file for the Online manual is included in the documentation of WebQuiz to allow you to create a local version of the Online manual. Look for the file `webquiz-online-manual.tex` in the `webquiz-www/docs` directory; see Section 4.2.

The online manual can either be compiled as a PDF file (see below), or using WebQuiz to produce an online version of the manual. The Online manual was written for "internal use" when WebQuiz was first written in 2004. WebQuiz has evolved quite a lot since then. There is some overlap between the Online manual and previous sections, however, the Online manual only describes how to typeset questions and it does not cover some of the more recent features of WebQuiz, such as the document class options, or how to use the program. If there are any discrepancies between the Online manual and the earlier sections of this manual then the Online manual should be discounted.

The Online manual has diagrams that are drawn using `pstricks` and, as a result, to create a `PDF` version of the Online manual use `latex webquiz-online-manual` to create a `dvi` file. The `dvi` file can be converted to `PDF` using `dvipdf`. The online manual needs to be compiled using `latex` rather than `pdflatex`, which will generate errors.

WebQuiz: Web Quizzes using LaTeX

WebQuiz will process this quiz using **pst2pdf**.

Introduction (Introduction) WebQuiz is a LaTeX package for creating *interactive* web quizzes. The idea is that you write the quiz using LaTeX and that WebQuiz creates the web page from this file. Anything that you can write using LaTeX will be converted to HTML by WebQuiz. This includes, for example, mathematics and graphics written using `pstricks`. WebQuiz supports three different types of quiz questions:

1. Multiple choice questions with a *unique* correct answer. (See button to to question 1)
2. Multiple choice questions with *several* (or no) correct answers. (See button to to question 2)
3. Questions that require the student to enter an answer, which can then be compared with the correct answer in several different ways. (See button to to question 3)

The use of WebQuiz is described in the next section. Later sections describe how each of the WebQuiz environments are used.

The `discussion` environment in WebQuiz can also be used to write Web Pages like this one (The pages you are reading here were written using WebQuiz.)

Basic Usage (Basic Usage) Once you have a WebQuiz file, you can run it through LaTeX, in the usual way, to produce a readable version of your quiz. When you are happy with the quiz, use WebQuiz to create the HTML version. Note that the printable version of the quiz does *not* look like the web page; rather, it contains all of the information in an easily readable layout.

If, for example, your quiz file is called *quiz1.tex* then you can use the following commands:

```
> latex quiz1          % latex a quiz file
> pdflatex quiz1       % a PDF versio of the quiz
> xdvi quiz1           % view the quiz using xdvi
> dvips quiz1          % print the quiz
> webquiz quiz1        % converts the quiz to html
```

Converting the quiz to html can take quite a long time, particularly if a large number of images need to be created.

WebQuiz files (WebQuiz files)

The basic structure of a WebQuiz file is as follows:

```
\documentclass{webquiz}

\title{Quiz 1: Complex numbers}

\UnitCode{MATH1001}
\UnitName{Differential Calculus}
\UnitURL{/u/UG/JM/MATH1001/}
\QuizzesURL{/u/UG/JM/MATH1001/Quizzes/}

\begin{document}

\begin{discussion}[short heading][optional heading]
    . . . % optional discussion
\end{discussion}

\begin{question} % question 1
    . . .
\end{question}

\begin{question} % question 2
    . . .
\end{question}
.
.
\end{document}
```

In the preamble of the LaTeX file you can specify the unit code, the name of the unit of study, the location of the homepage for the unit and the index file for the quizzes for this unit; this is done using the commands `\UnitCode`, `\UnitName`, `\UnitURL` and `\QuizzesURL` respectively. If the command `\QuizzesURL` is omitted then the URL for the quiz index file is set to `\UnitURL`/Quizzes.

The title of the quiz can be set in the preamble using the `\title` command. Note that the `\title` command *must* appear before the `\begin{document}` command. As in any LaTeX document, the preamble can define macros and load other LaTeX packages the usual way .

By using the `discussion` environment you can summarise the material for the students or add introductory material for the quiz. For example, `discussion` environments can be used to recall that main concepts being covered by the quiz or to give references to the lecture notes for the unit. The syntax for the

`discussion` environment is as follows:

```
\begin{discussion}[optional short heading][optional heading]
  . . .
\end{discussion}
```

Anything you like (text, mathematics, ...) can go inside discussion environments. The *optional heading*, which defaults to Discussion", is used both as the section heading on the web page and as the heading in the side-menu on left hand side of the page. If a *short heading* is also given then it is used in the side-menu. The quiz can contain zero or more discussion items (and zero or more quiz questions).

Questions are set inside a `question` environment. The text is followed by the answers.

WebQuiz supports three types of questions:

- Multiple choice questions with *precisely one* correct answer;
- Multiple choice questions with *zero or more* correct answers;
- Questions that require the students to enter an answer. Five difference comparison methods are available.

With each of these types of questions you can (and should) provide feedback to the students depending on whether their answer is correct or incorrect. Below we describe how to produce these different types of questions.

Question 1 (Question 1) The syntax for a multiple choice question having *precisely one* correct answer is as follows:

```
\begin{question}
  . . .question text
  \begin{choice}
    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response

    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response
    .
    .
    .
  \end{choice}
\end{question}
```

The different choices in a multiple choice question must be inside a `choice` environment. This environment behaves like a standard LaTeX list environment except that instead of using `\item` for list item you use:

- \correct for a correct choice
- \incorrect for an incorrect choice
- \feedback to give feedback to the student when they select this choice

At most one \feedback response should be given for each \correct and each \incorrect response. The \feedback commands are optional; however, it is recommended that you use them because targeted feedback to the students based on their responses can be beneficial.

For example, the code below, when run through WebQuiz, produces button to to question 1 in the online manual quiz.

```
\begin{question}
  The shaded region in the graph
  \begin{center}
    \begin{pspicture}(-3,-1.5)(3,4)
      \pscircle[linewidth=1pt,linestyle=dashed,%
                fillcolor=SkyBlue,fillstyle=solid](1,1){2}
      \psaxes[linecolor=red,linewidth=1pt,labels=none]%
      {->}(0,0)(-1.5,-1.5)(3.5,3.5)
      \rput(3.75,0){$x$}
      \rput(0,3.85){$iy$}
      \rput(3,-0.4){3}
      \rput(-0.4,3){3$i$}
      \psdots(1,1)
    \end{pspicture}
  \end{center}
  is equal to which of the following sets of complex numbers?
  \begin{choice}
    \incorrect $\{z \in \C : (z-1)^{2}+(z-(i+1))^{2}<2\}$
    \feedback  The equation of a circle in the complex plane with
    centre $a+ib$ and radius $r$ is
    \begin{displaymath}
      \{z\in\C : |z-(a+ib)|<r \}.
    \end{displaymath}

    \incorrect $\{z \in \C : z+(i+1)<2\}$
    \feedback  You want the set of points whose \textit{distance}
    from $1+i$ is less than $2$.

    \correct   $\{z \in \C : |z-(i+1)|<2\}$
    \feedback  The graph shows the set of complex numbers whose
    distance from $1+i$ is less than $2$.

    \incorrect $\{z \in \C : |z-2|<|i+1-2|\}$
    \feedback  As $|i+1-2|=\sqrt 2$, this is the set of complex
    numbers whose distance from $2$ is less than
    $\sqrt 2$.
```

```
    \incorrect None of the above.
    \feedback The graph shows the set of complex numbers whose
    distance from the centre of the circle is less than $2$.
  \end{choice}
\end{question}
```

Question 2 (Question 2) To allow multiple (or no) correct answer we add `multiple` as an optional argument to the `choice` environment:

```
\begin{question}
  . . .question text. . .
  \begin{choice}[multiple]
    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response

    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response
    .
    .
    .
  \end{choice}
\end{question}
```

The only difference to the previous case is that zero or more `\correct` commands can appear.

For example, button to to question 2 below was typed into WebQuiz using the following commands:

```
\begin{question}
  Which of the following numbers are prime?
  \begin{choice}[multiple]
    \incorrect 1
    \feedback  By definition, a prime is a number greater than 1
    whose only factors are 1 and itself.

    \correct   19
    \feedback  The only factors of 19 are 1 and itself.

    \incorrect 6
    \feedback  2 is a factor of 6

    \correct   23
    \feedback  The only factors of 23 are 1 and itself.

    \correct   191
    \feedback  The only factors of 191 are 1 and itself.
  \end{choice}
\end{question}
```

Question 3 (Question 3)

By default, the `choice` environments puts the multiple choice options into one column format. Sometimes the options look better when listed in two or more columns, however, this should be used sparingly as multiple columns do not always display well if the quiz is viewed on a mobile device. By using the `columns` key word in a `choice` environment you can specify the number of columns in the HTML version of the quiz.

```
\begin{question}
  . . .question text. . .
  \begin{choice}[multiple, columns=n]     . . . n columns
    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response

    \(in)correct . . . text for (in)correct option
    \feedback    . . . feedback on response
    .
    .
    .
  \end{choice}
\end{question}
```

If the optional argument `[multiple]` is not present, then the question admits precisely one correct answer.

For example, button to to question 3 below was typed into WebQuiz using the following commands:

```
\begin{question}
  What are suitable parametric equations for this plane curve?
  \begin{center}
    \psset{unit=.6cm}
    \begin{pspicture}(-2.5,-0.5)(5,5.5)
      \psaxes[linecolor=red,linewidth=1pt,labels=none]%
      {->}(0,0)(-2.5,-1.5)(5,5)
      \psellipse[linecolor=SkyBlue,linewidth=2pt](1,2)(3,2)
    \end{pspicture}
  \end{center}

  \begin{choice}[columns=1]
    \incorrect $x=2\cos t + 1$, $y=3\sin t + 2$
    \feedback This is an ellipse with centre $(1,2)$ and with axes of
    length $4$ in the $x$-direction and $6$ in the $y$-direction.
    \begin{center}
      \psset{unit=.6cm}
      \begin{pspicture}(-2.5,-0.5)(5,5.5)
        \psaxes[linecolor=red,linewidth=1pt,labels=none]%
        {->}(0,0)(-2.5,-1.5)(5,5)
        \parametricplot[linecolor=SkyBlue,linewidth=2pt]{0}{360}%
```

```
        {t cos 2 mul 1 add t sin 3 mul 2 add}
      \end{pspicture}
    \end{center}

    \correct $x=3\cos t + 1$, $y=2\sin t + 2$
    \feedback The curve is an ellipse centre (1,2) with axes length 6
    in the $x$ direction and 4 in the $y$ direction.

    \incorrect $x=3\cos t - 1$, $y=2\sin t - 2$
    \feedback This is an ellipse with centre $(-1,-2)$ and with axes
    of length $6$ in the $x$-direction and $4$ in the $y$-direction.
    \begin{center}
      \psset{unit=.6cm}
      \begin{pspicture}(-5,-4)(1,2)
        \psaxes[linecolor=red,linewidth=1pt,labels=none]%
        {<-}(0,0)(-4.5,-5.5)(1,2)
        \parametricplot[linecolor=SkyBlue,linewidth=2pt]{0}{360}%
        {t cos 3 mul 1 sub t sin 2 mul 2 sub}
      \end{pspicture}
    \end{center}

    \incorrect $x=2\cos t - 1$, $y=3\sin t - 2$
    \feedback This is an ellipse with centre $(-1,-2)$ and with axes
    of length $4$ in the $x$-direction and $6$ in the $y$-direction.
    \begin{center}
      \psset{unit=.6cm}
      \begin{pspicture}(-4,-5)(1,2)
        \psaxes[linecolor=red,linewidth=1pt,labels=none]%
        {<-}(0,0)(-4.5,-5.5)(1,2)
        \parametricplot[linecolor=SkyBlue,linewidth=2pt]{0}{360}%
        { t cos 2 mul 1 sub t sin 3 mul 2 sub}
      \end{pspicture}
    \end{center}
  \end{choice}
\end{question}
```

Question 4 (Question 4) The final type of question that WebQuiz supports is a question that requires an answer, which can be a number or a string. The answer is typeset using the `\answer` macro. The `\answer` macro takes two arguments: an optional comparison method, which defaults to `string`, and the correct answer for the question:

```
\answer[comparison method]{correct answer}
```

Feedback for correct and incorrect answers is given using the macros `\whenRight` and `\whenWrong`, respectively. The structure of questions with `\answer`'s is as follows:

```
\begin{question}
  . . .question text. . .
  \answer[*][complex|integer|lowercase|number|string]{actual answer}
  \whenRight . . . feedback when right (optional)
  \whenWrong . . . feedback when wrong (optional)
\end{question}
```

See the WebQuiz manual for details of the different comparison types. For example, button to to question 4 below was typed into WebQuiz using the following commands:

```
\begin{question}
  If the vectors $\vect{a}$ (of magnitude 8 units) and $\vect{b}$
  (of magnitude 3 units) are perpendicular, what is the value
  of
  \begin{displaymath}
    |\vect{a} -2\vect{b}|~?
  \end{displaymath}
  (Hint: Draw a diagram!)

  \answer[number]{10}
  \whenRight The vectors $\vect{a}$, \(-2\vect{b}\), and
  $\vect{a} - 2\vect{b}$ form the sides of a right-angled
  triangle, with sides of length $8$ and $6$ and
  hypotenuse of length $|\vect{a} -2\vect{b}|$. Therefore
  by Pythagoras' Theorem,
  \(|\vect{a} -2\vect{b}|=\sqrt{8^2+6^2}=10\).

  \whenWrong Draw a diagram and then use Pythagoras' theorem.
\end{question}
```

Index Files (Index Files) WebQuiz also provides a mechanism for creating a web page containing an index of all quizzes for a given Unit of Study. This is done with a WebQuiz file that contains a `quizindex` environment. The syntax for this environment is as follows:

```
\begin{quizindex}
  \quiz[url1]{title for quiz 1}
  \quiz[url2]{title for quiz 2}
  . . .
\end{quizindex}
```

If no *URL* is given as an optional argument to `\quiz` then WebQuiz sets the url(s) to `quiz1.html`, `quiz2.html` and so on.

Credits (Credits) WebQuiz was written and developed in the School of Mathematics and Statistics at the University of Sydney. The system is built on LaTeX with the conver-

sion from LaTeX to HTML using Eitan Gurari's TeX4ht, and Michal Hoftich's make4ht.

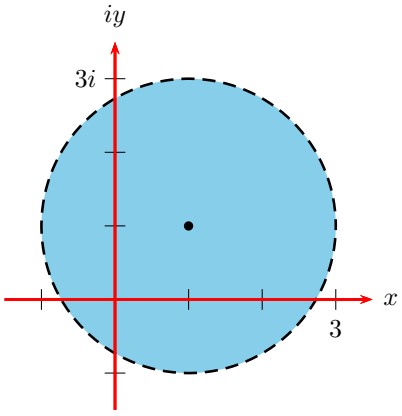
To write quizzes using WebQuiz it is only necessary to know LaTeX, however, the WebQuiz system has three components:

- A LaTeX document class file, `webquiz.cls`, and a TeX4ht configuration file, `webquitexz.cfg`, that enable the quiz files to be processed by LaTeX and TeX4ht, respectively.
- A python program, `webquiz`, that translates the XML file that is produced by TeX4ht into HTML.
- Some Javascript and CSS that controls the quiz web page.

The LaTeX component of WebQuiz was written by Andrew Mathas and the python, CSS and Javascript code was written by Andrew Mathas (and Don Taylor), based on an initial prototype of Don Taylor's from 2001. Since 2004 the program has been maintained and developed by Andrew Mathas. Although the program has changed substantially since 2004, Don's idea of using TeX4ht, and some of his code, is still in use.

Thanks are due to Bob Howlett for general help with CSS and, for Version 5, to Michal Hoftich for technical advice.

Question 1.
The shaded region in the graph



is equal to which of the following sets of complex numbers?

✗ *Option* 1(a): $\{z \in \mathbb{C} : (z-1)^2 + (z-(i+1))^2 < 2\}$

Feedback: The equation of a circle in the complex plane with centre $a + ib$ and radius r is

$$\{z \in \mathbb{C} : |z - (a + ib)| < r\}.$$

✗ *Option* 1(b): $\{z \in \mathbb{C} : z + (i + 1) < 2\}$

Feedback: You want the set of points whose *distance* from $1 + i$ is less than 2.

✔ *Option* 1(c): $\{z \in \mathbb{C} : |z - (i + 1)| < 2\}$

Feedback: The graph shows the set of complex numbers whose distance from $1 + i$ is less than 2.

✗ *Option* 1(d): $\{z \in \mathbb{C} : |z - 2| < |i + 1 - 2|\}$

Feedback: As $|i+1-2| = \sqrt{2}$, this is the set of complex numbers whose distance from 2 is less than $\sqrt{2}$.

✗ *Option* 1(e): None of the above.

Feedback: The graph shows the set of complex numbers whose distance from the centre of the circle is less than 2.

Question 2.
Which of the following numbers are prime?

✗ *Option* 2(a): 1

Feedback: By definition, a prime is a number greater than 1 whose only factors are 1 and itself.

✔ *Option* 2(b): 19

Feedback: The only factors of 19 are 1 and itself.

✗ *Option* 2(c): 6

Feedback: 2 is a factor of 6

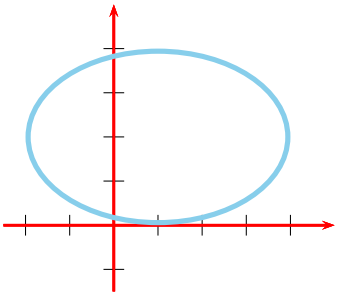
✔ *Option* 2(d): 23

Feedback: The only factors of 23 are 1 and itself.

✔ *Option* 2(e): 191

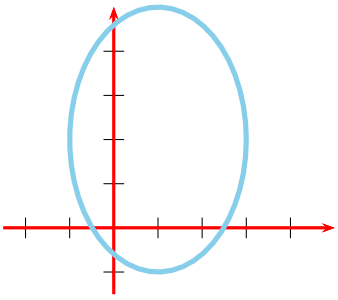
Feedback: The only factors of 191 are 1 and itself.

Question 3.
What are suitable parametric equations for this plane curve?



✗ *Option* 3(a): $x = 2\cos t + 1,\ y = 3\sin t + 2$

Feedback: This is an ellipse with centre $(1, 2)$ and with axes of length 4 in the x-direction and 6 in the y-direction.

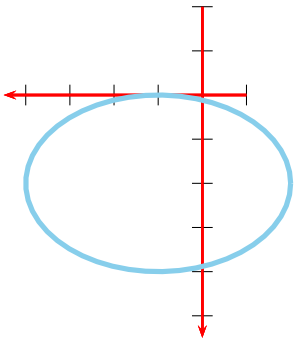


✔ *Option* 3(b): $x = 3\cos t + 1,\ y = 2\sin t + 2$

Feedback: The curve is an ellipse centre (1,2) with axes length 6 in the x direction and 4 in the y direction.

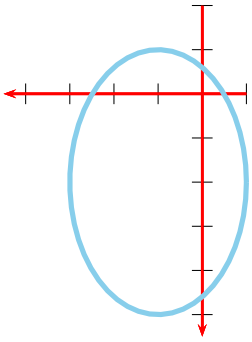
✗ *Option* 3(c): $x = 3\cos t - 1,\ y = 2\sin t - 2$

Feedback: This is an ellipse with centre $(-1, -2)$ and with axes of length 6 in the x-direction and 4 in the y-direction.



✗ *Option* 3(d): $x = 2\cos t - 1,\ y = 3\sin t - 2$

Feedback: This is an ellipse with centre $(-1, -2)$ and with axes of length 4 in the x-direction and 6 in the y-direction.



Question 4.
If the vectors **a** (of magnitude 8 units) and **b** (of magnitude 3 units) are perpendicular, what is the value of $|\mathbf{a} - 2\mathbf{b}|$? (Hint: Draw a diagram!)

Answer (number comparison): 10	units

Feedback when right: The vectors **a**, $-2\mathbf{b}$, and $\mathbf{a} - 2\mathbf{b}$ form the sides of a right-angled triangle, with sides of length 8 and 6 and hypotenuse of length $|\mathbf{a} - 2\mathbf{b}|$. Therefore by Pythagoras' Theorem, $|\mathbf{a} - 2\mathbf{b}| = \sqrt{8^2 + 6^2} = 10$.

Feedback when wrong: Draw a diagram and then use Pythagoras' theorem.

C Licence

Authors	Andrew Mathas
Description	A LaTeX package for writing online quizzes
Maintainer	Andrew Mathas
System requirements	Python3 and LaTeX, including TeX4ht and make4ht
Licence	GNU General Public License, Version 3, 29 June 2007
Release date	2019/02/08
Repository	github.com/AndrewAtLarge/WebQuiz/issues

Index

This is an index only for the main WebQuiz manual. It does not index the Online manual (Appendix B).