The pubQuiz class *

Cynthia Rodriguez
CR.github@pm.me

Version v1.0, 2025-06-27

Abstract

This manual describes how to use the pubQuiz class to generate sheets for running a Pub Quiz or Trivia activity. The package creates simple answer sheets for the participants, and gives the option to generate a script for the host of the event.

1 Examples

1.1 An unscripted example

Quizzard, the renowned Pub Quiz master, has agreed to make the answer sheets for this week's Quiz at the LaTeXpub. A new assistant will be designing the questions this week, so it should be an easy task.

The quiz will have only one round with five standard categories and a bonus game:

1. Beer: 5 questions
2. Spirits: 5 questions
3. Sports: 5 questions
4. BONUS: Darts Game
5. Wines: 5 questions
6. Name the Song: 20 questions

Quizzard wants the Name the Song category to show the answers in 4 columns.

Luckily, Quizzard has been using LaTeXfor a long time, and he has just discovered the `PubQuiz` class, which seems well-suited for the task. The main commands to use for created answer sheets only are:

- `\PubRound`: prints the Team Name space, the Score box (optional: with the points available), and the Round title
- `\PubCat`: prints the answer lines for a category. By default, it prints 5 answer lines in a single column, but it allows for Bonus categories, multiple columns, and other number of questions.

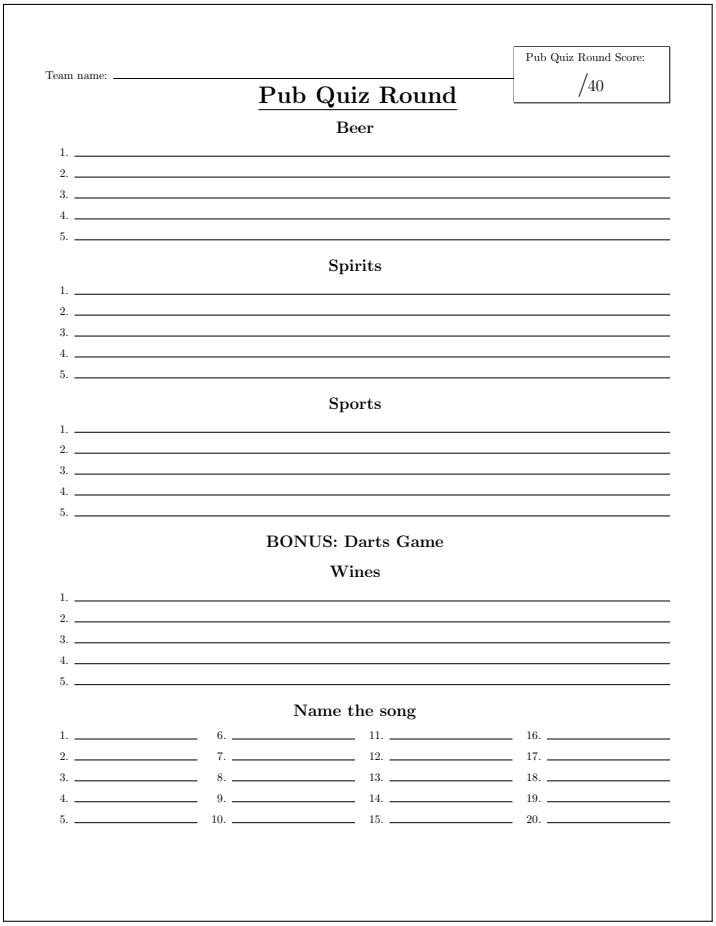
*This document corresponds to pubQuiz v1.0, dated 2025-06-27.

With this information, Quizzard creates their first `pubQuiz` document:

```
\documentclass{pubQuiz}

\begin{document}
  \PubRound{Pub Quiz Round}[40]
  % The total points for the round is optional
    \PubCat{Beer}
    \PubCat{Spirits}
    \PubCat{Sports}
    \PubCat*{BONUS: Darts Game}
    % With the *, we get a bonus category
    \PubCat{Wines}
    \PubCat|4|{Name the song}[20]
    % The number of columns goes between vertical lines
    % the number of questions in square brackets
\end{document}
```

Quizzard looks at the compiled answer sheet, and is happy to have completed their task:



Team name: _____

Pub Quiz Round Score: /40

Pub Quiz Round

Beer

1. _____
2. _____
3. _____
4. _____
5. _____

Spirits

1. _____
2. _____
3. _____
4. _____
5. _____

Sports

1. _____
2. _____
3. _____
4. _____
5. _____

BONUS: Darts Game

Wines

1. _____
2. _____
3. _____
4. _____
5. _____

Name the song

1. _____	6. _____	11. _____	16. _____
2. _____	7. _____	12. _____	17. _____
3. _____	8. _____	13. _____	18. _____
4. _____	9. _____	14. _____	19. _____
5. _____	10. _____	15. _____	20. _____

1.2 A scripted example

Quizzard's assistant has taken a vacation, so Quizzard will have to plan this week's Pub Quiz alone. That means writing a script with questions and answers, as well as the answer sheets. He decides to plan a short trivia, with two categories and a bonus game:

1. Ducks: 5 questions
2. BONUS: Coin Toss
3. Duck songs: 10 questions

Quizzard looks again at the `PubQuiz` class, which he hopes will help him in this task. The main commands/environments to use for scripts and answer sheets are:

- To print a script, one should give the `host` option to the `pubQuiz` class.
- The environment `PubRules` allows for the Pub Quiz rules to be written in the host script.
- The environment `PubInstructions` allows for instructions for the hosts to be written in the host script.
- `\PubRound`: prints the Team Name space, the Score box (optional: with the points available), and the Round title.
- Environment `PubCategory`: prints the category title (with some extra details), questions given as in an `enumerate` environment. The environment allows for Bonus categories, multiple columns, category instructions, and some host notes.
- `\PubQ`: for giving a question and its answer. A dot between question an answer prints inline in the host script.

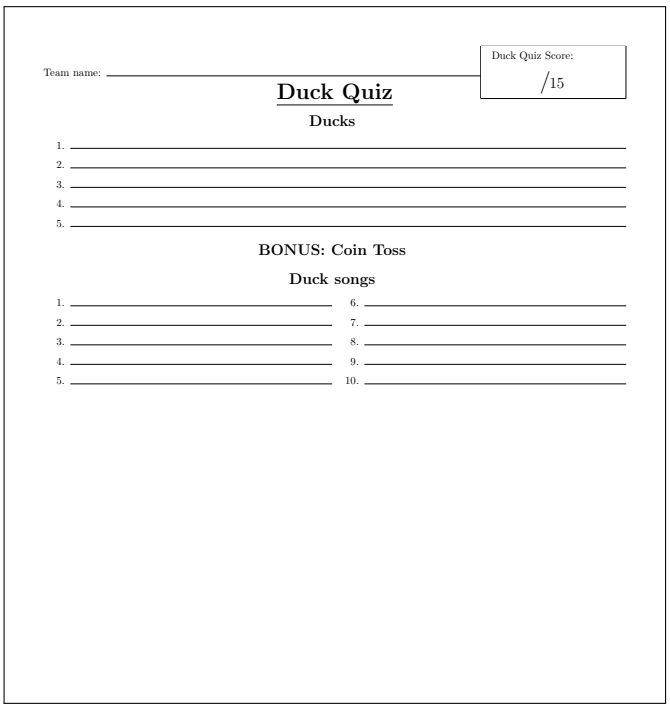
With this information, Quizzard creates their first `pubQuiz` scripted sheet:

```
\documentclass{pubQuiz}

\begin{document}
  \PubRound{Duck Quiz}[15] % Optional total round points
    \begin{PubRules}
      \item Quizzard is ALWAYS right
    \end{PubRules}
    \begin{PubCategory}{Ducks}  % Category name
      <Turn off screens> % Short comments for the host
      \PubQ[When was the rubber ducky invented] % Question
        [Late 19th century] % Answer
      \PubQ[Who made a lithograph called Sitting Ducks]
        [Michael Bedard]
      \PubQ[When was the Duck and cover film made?][1952]
      \PubQ[What is Ducky in the Land before Time?]
        [Saurolophus]
      \PubQ[What is the duckiest search engine?][DuckDuckGo]
    \end{PubCategory}
```

```
    \begin{PubCategory}*     % Bonus category
      {BONUS: Coin Toss}     % Category Name
      [ INSTRUCTIONS: Quizzard will throw a coin.
        Players guess if it will land on heads or tails.
        Players who guess wrong are out of the game.
        The last participant gets a free beer.  ]
    \end{PubCategory}
    \begin{PubInstructions}
      \item Get speakers for the next category
    \end{PubInstructions}
    \begin{PubCategory}|2|     % Number of columns
      {Duck songs}  % Category name
      [We play 10 songs, write the title] % Instructions
      \PubQ[Track 1].[O Pato] % Answer inline on script
      \PubQ[Track 2].[Rubber Duckie]
      \PubQ[Track 3].[Disco Duck]
      \PubQ[Track 4].[I want a new duck]
      \PubQ[Track 5].[Disco Alert]
      \PubQ[Track 6].[Duck the Halls]
      \PubQ[Track 7].[Five Little Ducks]
      \PubQ[Track 8].[Duck Duck Goose]
      \PubQ[Track 9].[Duck Tales]
      \PubQ[Track 10].[Sitting Ducks]
    \end{PubCategory}
\end{document}
```

Quizzard looks at the compiled output and gets this answer sheet



To produce the host script, Quizzard gives the `host` option to the class.

```
\documentclass[host]{pubQuiz}
```

The compiled output then becomes a host script

Duck Quiz

The Rules:

1. Quizzard is ALWAYS right

Ducks

Turn off screens

1. When was the rubber ducky invented
   Late 19th century
2. Who made a lithograph called Sitting Ducks
   Michael Bedard
3. When was the Duck and cover film made?
   1952
4. What is Ducky in the Land before Time?
   Saurolophus
5. What is the duckiest search engine?
   DuckDuckGo

BONUS: Coin Toss

INSTRUCTIONS: Quizzard will throw a coin. Players guess if it will land on heads or tails. Players who guess wrong are out of the game. The last participant gets a free beer.

Instructions:

- Get speakers for the next category

Duck songs

We play 10 songs, write the title

1. Track 1 ..... **O Pato**
2. Track 2 ..... **Rubber Duckie**
3. Track 3 ..... **Disco Duck**
4. Track 4 ..... **I want a new duck**
5. Track 5 ..... **Disco Alert**
6. Track 6 ..... **Duck the Halls**
7. Track 7 ..... **Five Little Ducks**
8. Track 8 ..... **Duck Duck Goose**
9. Track 9 ..... **Duck Tales**
10. Track 10 ..... **Sitting Ducks**

1

1.3 A picture page

As part of the Duck Quiz, Quizzard would like to have a name the duck picture page. The only extra command is

- `\PubQPic`: for giving a picture question and its answer.

With this information, Quizzard creates their first `pubQuiz` picture page:

```
\documentclass[landscape]{pubQuiz}

\usepackage{duckuments} % Random pictures of ducks

\begin{document}
  \PubRound{Picture Page}
    \begin{PubCategory}|5|{Name the Duck} % Category in 5 columns
      \PubQPic{example-image-duck} % Image file
        [Ducky 1] % Answer
      \PubQPic{example-image-duck}[Ducky 2]
      \PubQPic{example-image-duck}[Ducky 3]
      \PubQPic{example-image-duck}[Ducky 4]
      \PubQPic{example-image-duck}[Ducky 5]
    \end{PubCategory}
    \begin{PubCategory}|5|{Meme the Duck}
      \PubQPic{example-image-duck}[Ducky 1]
      \PubQPic{example-image-duck}[Ducky 2]
      \PubQPic{example-image-duck}[Ducky 3]
      \PubQPic{example-image-duck}[Ducky 4]
      \PubQPic{example-image-duck}[Ducky 5]
    \end{PubCategory}
    \begin{PubCategory}|5|{Rubber duckies}
      \PubQPic{example-image-duck}[Ducky 1]
      \PubQPic{example-image-duck}[Ducky 2]
      \PubQPic{example-image-duck}[Ducky 3]
      \PubQPic{example-image-duck}[Ducky 4]
      \PubQPic{example-image-duck}[Ducky 5]
    \end{PubCategory}
    \begin{PubCategory}|5|{Duck placeholders}
      \PubQPic{example-image-duck}[Ducky 1]
      \PubQPic{example-image-duck}[Ducky 2]
      \PubQPic{example-image-duck}[Ducky 3]
      \PubQPic{example-image-duck}[Ducky 4]
      \PubQPic{example-image-duck}[Ducky 5]
    \end{PubCategory}
\end{document}
```

Quizzard looks at the compiled output and gets this answer sheet

Team name: _____

Picture Page Score:

Picture Page

Name the Duck

1. 2. 3. 4. 5.

Meme the Duck

1. 2. 3. 4. 5.

Rubber duckies

1. 2. 3. 4. 5.

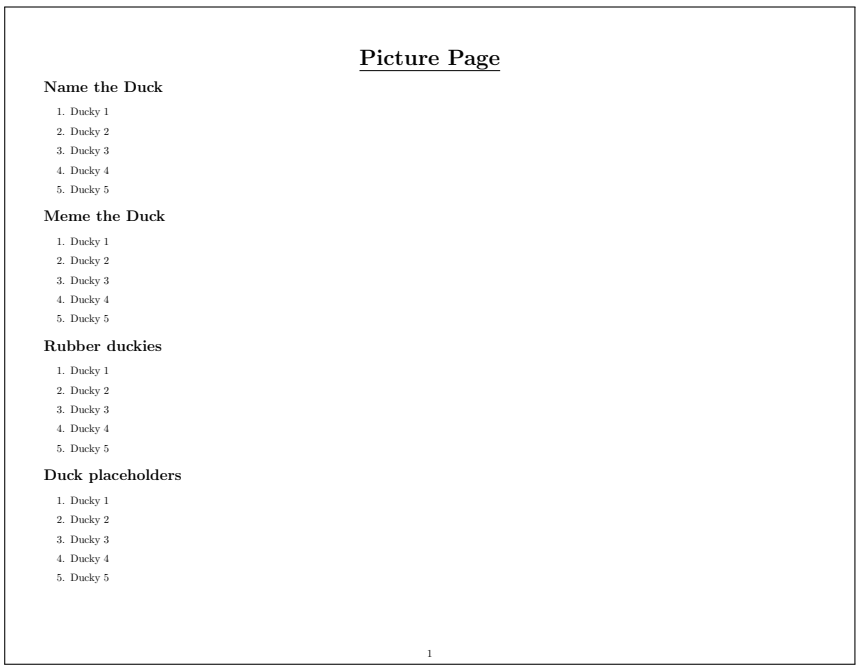
Duck placeholders

1. 2. 3. 4. 5.

To produce the host script, Quizzard gives the `host` option to the class.

```
\documentclass[landscape]{pubQuiz}
```

The compiled output then becomes a host script



Picture Page

Name the Duck

1. Ducky 1
2. Ducky 2
3. Ducky 3
4. Ducky 4
5. Ducky 5

Meme the Duck

1. Ducky 1
2. Ducky 2
3. Ducky 3
4. Ducky 4
5. Ducky 5

Rubber duckies

1. Ducky 1
2. Ducky 2
3. Ducky 3
4. Ducky 4
5. Ducky 5

Duck placeholders

1. Ducky 1
2. Ducky 2
3. Ducky 3
4. Ducky 4
5. Ducky 5

1

2 User Commands

The PubQuiz class can create both answer sheets and host scripts for a Pub Quiz style event. We have made the choice to keep the number of user commands small, and enable functionality via tokens and optional parameters, which we will explain here.

2.1 Class Options

The class inherits all options from the `article` class, and adds three extra options for easier PubQuiz creation:

- `host` and `nohost`:

`host` – The `host` option creates a document for the hosts.
It creates a script with rules, instructions, questions to read (with or without answers) for regular and bonus categories.

`nohost` – The `nohost` option is enabled by default and creates a document for participants.
It creates an answer sheet with space to write team names, a score box, category titles and space for answers, and titles for bonus categories (if this is enabled). Answer spaces are not given for bonus categories.

- `answers` and `noanswers`:

`answers` – The `answers` option is enabled by default and affects only the document for hosts.
It prints the given answers to all questions in the host script.

`noanswers` – The `noanswers` option disables including correct answers in the host document.

- `printbonus` and `noprintbonus`:

`printbonus` – The `printbonus` option is enabled by default and affects only the document for participants.
When it is enabled, the title of bonus categories is printed in the answer sheet.

`noprintbonus` – When the `noprintbonus` option is chosen, bonus categories do not appear in the answer sheets for participants.

2.2 Setting up a Pub Quiz

`\PubRound` The basic set-up command for a Pub Quiz document is

\PubRound{round_name}[max_points]

which takes two parameters

- `round_name`: Name to be printed for the round (mandatory)
- `max_points`: Maximum points in the round (optional)

This command behaves differently in the host script and the answer sheet.

- Host script: Prints only the round title
- Answer sheet: Prints three things on the page:
  1. Line for participants to write their team name
  2. Score box (maximum score included, if given)
  3. Round title

The class also provides two environments for hosts to add instructions in the host script:

`PubRules` (*env.*)
- The environment `PubRules` which creates the document title, prints the headline The Rules. and opens a numbered list. The user writes their rules as in an `enumerate` environment.

`PubInstructions` (*env.*)
- The environment `PubInstructions` prints the headline `Instructions:` and opens a non-numbered list. The user writes their instructions as in an `itemize` environment.

2.3 Creating Categories

The `PubQuiz` class includes three commands to create categories and subcategories in a Pub Quiz document.

`\PubCat` The first one is `\PubCat`, which allows the quick creation of answer sheets, without a host script.

\PubCat * |number_columns| {category_name} [num_questions]

- The star is an optional argument which signals a bonus category. If it is given, no spaces to answer will be printed in the answer sheet.
- The optional parameter `number_columns` allows printing the answer lines in several columns within an answer sheet. The minimum value for `number_columns` is 2.
- The only mandatory parameter is `category_name`, which determines the name to be printed for the category.
- The optional parameter `num_questions` gives the number of questions in the category. This value is set to be 5 by default.

`PubCategory` (*env.*) For scripted pubQuiz documents, the corresponding environment is `PubCategory`. The environment accepts the following paramenters:

- The star is an optional argument which signals a bonus category. If it is given, no spaces to answer will be printed in the answer sheet.
- The optional parameter `number_columns` allows printing the answer lines in several columns within an answer sheet. The minimum value for `number_columns` is 2.
- The only mandatory parameter is `category_name`, which determines the name to be printed for the category.

- The optional parameter `host_notes` allows for short notes on the host script. The notes are printed inline with the category name, using the same format.
- The optional parameter `host_instructions` allows for longer category instructions on the host script. The notes are printed before the questions in the category, using the standard text format.

`PubSubcategory` (*env.*) Scripted `pubQuiz` documents also allow for the inclusion of subcategories with the command `\PubSubcategory`. The environment accepts the following paramenters:

- The star is an optional argument which signals a bonus category. If it is given, no spaces to answer will be printed in the answer sheet.
- The optional parameter `number_columns` allows printing the answer lines in several columns within an answer sheet. The minimum value for `number_columns` is 2.
- The only mandatory parameter is `subcategory_name`, which determines the name to be printed for the category.
- The optional parameter `host_notes` allows for short notes on the host script. The notes are printed inline with the subcategory name, using the same format.
- The optional parameter `host_instructions` allows for longer category instructions on the host script. The notes are printed before the questions in the subcategory, using the standard text format.

Note that the usage and parameters of `\PubSubcategory` are identical to those of `PubCategory`.

2.4 Creating Questions

The `PubQuiz` class gives two commands to include questions (and their answers) in the script.

The command `\PubQ` is used to write text-based questions. In the host script, it prints the question and its answer (if the `printanswers` option is enabled). In the answer sheet, it prints the question number and a line for participants to write their answer on (for non-bonus questions)

\PubQ * {question} . [answer]

- A star is an optional argument which signals a bonus question. If it is given, nothing will be printed in the answer sheet.
- The text for the question is given as the optional parameter `question`. If no text is given, "No question given" will be printed in the host script.
- A dot is an optional argument which indicates that the answer should be written inline with the question, with dots separating them. If it is not given, the answer will be printed in a new line.
- The answer for the question is given as the optional parameter `answer`.

For picture-based questions, the command \PubQPic is given. In the answer sheet, it prints the question number, an image and a line below it for participants to write their answer. In the host script, only the question number and the answer are printed.

\PubQPic {image_file}[answer]

- The path to the image file should be given as the mandatory parameter `image_file`
- The answer for the question is given as the optional parameter `answer`.

3 Implementation

List of required packages for options:

```
\RequirePackage{ifthen}
\RequirePackage{environ}
```

`host` Switch for Host script or answer sheet

`nohost`

```
\newif\ifpubQuiz@Host
 \pubQuiz@Hostfalse
 \DeclareOption{host}{\pubQuiz@Hosttrue}
 \DeclareOption{nohost}{\pubQuiz@Hostfalse}

\newif\ifpubQuiz@Answers
 \pubQuiz@Answerstrue
 \DeclareOption{answers}{\pubQuiz@Answerstrue}
 \DeclareOption{noanswers}{\pubQuiz@Answersfalse}
```

`printbonus` Switch to include (or not include) bonus categories in the answer sheet.

`noprintbonus`

```
\newif\ifpubQuiz@Bonus
 \pubQuiz@Bonustrue
 \DeclareOption{printbonus}{\pubQuiz@Bonustrue}
 \DeclareOption{noprintbonus}{\pubQuiz@Bonusfalse}
```

Inherits from article class, processing any inherited options

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions\relax
\LoadClass{article}
```

List of all other required packages:

```
\RequirePackage[margin=0.5in]{geometry}
\RequirePackage{titlesec}
\RequirePackage[inline]{enumitem}
\RequirePackage{tikz}
\RequirePackage{multicol}
\RequirePackage{mdframed}
\RequirePackage{nicefrac}
\RequirePackage{graphicx}
\RequirePackage{calc}
\RequirePackage[export]{adjustbox}
```

Sets formatting for the pubQuiz documents, which changes the format of section, subsection, and subsubsection as well as some spacing changes to lines and itemized environments, Page numbers are not printed in answer sheets.

```
\titleformat{\section}{\huge\filcenter}{\thesection}{}{}
\titleformat{\subsection}{\Large\filcenter}{\thesubsection}{}{}
\titleformat{\subsubsection}{\large}{\thesubsubsection}{}{}
\titlespacing*{\section}{0cm}{1ex plus 1mm minus 1mm}{1ex plus 1mm minus 1mm}
\titlespacing*{\subsection}{0cm}{2ex plus 1mm minus 1mm}{2ex plus 1mm minus 1mm}
\setlength{\abovedisplayskip}{1em}
\setlist{itemsep=0.5ex}
\ifpubQuiz@Host \else \pagestyle{empty} \fi
```

An internal environment to allow the exclusion of host-specific environments from the answer sheet.

```
\NewEnviron{pubQuiz@Hide}{}
```

An internal boolean to allow the optional use of columns in the answer sheet

```
\newif\ifpubQuiz@Columns
 \pubQuiz@Columnsfalse
\setlength{\multicolsep}{0pt plus 3pt minus 3pt}
\setlength{\columnsep}{5pt}
```

Definition of image size settings from Picture Questions

```
\newlength{\pubQuiz@Pixwidth}
\newlength{\pubQuiz@Pixheight}
\setlength{\pubQuiz@Pixheight}{60bp}
\setlength{\pubQuiz@Pixwidth}{115bp}
```

```
\makeatletter
```

3.1 Round-specific environments and commands

This subsection lists commands and environments to set-up the structure of the trivia script and answer sheets.

`PubRules` (*env.*) The environment `PubRules` allows the script-writer to include the general rules for Trivia. The command creates the title and shows the rules only in the host script. The command has no impact on answer sheets.

The code below the environment allows this environment to be hidden in answer sheets.

```
\NewDocumentEnvironment{PubRules}{
 %\maketitle
 \subsection*{\textsc{The Rules:} \hfill}
 \begin{enumerate}
}{ \end{enumerate}\par }
 \ifpubQuiz@Host \else
   \let\PubRules\pubQuiz@Hide
    \let\endPubRules\endpubQuiz@Hide
 \fi
```

`PubInstructions` (*env.*) The environment `PubInstructions` allows the script-writer to include the general instructions at any point within the Trivia script. The command has no impact on answer sheets.

The code below the environment allows this environment to be hidden in answer sheets.

```
\NewDocumentEnvironment{PubInstructions}{
 \vspace{3mm}
```

```
  \subsection*{\textsc{Instructions:} \hfill}
  \begin{itemize}
 }{ \end{itemize}\par }
 \ifpubQuiz@Host \else
   \let\PubInstructions\pubQuiz@Hide
    \let\endPubInstructions\endpubQuiz@Hide
 \fi
```

`\pubQuiz@ScoreBox` The \pubQuiz@ScoreBox command prints a box for the score of a round. It takes two parameters

\pubQuiz@ScoreBox{round_name}[max_points]

- `round_name`: Name to be printed for the round scored (mandatory)
- `max_points`: Maximum points in the round (optional)

```
 \NewDocumentCommand\pubQuiz@ScoreBox{m o}{
  \begin{mdframed}
   #1 Score:%
   \begin{center}
    {\huge $\IfNoValueTF{#2}{
     \phantom{\nicefrac{}{}}
    }{
     \nicefrac{}{#2}}$
    }
   \end{center}%
  \end{mdframed}%
 }
```

`\pubQuiz@PageHeader` The \pubQuiz@PageHeader command prints the top matter of a page: space for team names, round score box, and round name. It takes two parameters

\pubQuiz@PageHeader {round_name}[max_points]

- `round_name`: Name to be printed for the round scored (mandatory)
- `max_points`: Maximum points in the round (optional)

```
 \NewDocumentCommand\pubQuiz@PageHeader{m o}{
  \ifpubQuiz@Host \else \clearpage%
   \noindent\begin{minipage}{0.75\columnwidth}
    Team name:\enspace\hrulefill
   \end{minipage}%
   \hfill \begin{minipage}{0.25\columnwidth}
    \pubQuiz@ScoreBox{#1}[#2]
   \end{minipage}%
  \fi%
 }
```

`\PubRound` The user-command `\PubRound` prints the top-matter for a round, the content and format changes between the host script (Round title only) and the answer sheet (Team name space, score box, and Round title).)

\PubRound {round_name}[max_points]

- `round_name`: Name to be printed for the round scored (mandatory)
- `max_points`: Maximum points in the round (optional)

```
\NewDocumentCommand\PubRound{mo}{
 \pubQuiz@PageHeader{#1}[#2]
 \ifpubQuiz@Host \else \vspace{-3em} \fi
 \section*{\textbf{\underline{#1}}}
}
```

3.2 Category-specific environments and commands

This subsection lists internal and user environments and commands that may be used to handle Categories and subcategories within a Trivia event.

`\pubQuiz@CategoryTopMatter` The \pubQuiz@CategoryTopMatter command prints the category name. In the host script, it also prints *notes* (in the same format as the title, at the end of the line) and *instructions* (in the regular format, before the enumerate environment). It takes two parameters

\pubQuiz@CategoryTopMatter {category_name}¡notes¿[instructions]

- `category_name`: Name to be printed for the category title (mandatory)
- `notes`: Notes for the host script (optional)
- `instructions`: Instructions for the host script (optional)

```
\NewDocumentCommand\pubQuiz@CategoryTopMatter{m d<> +o}{
 \subsection*{\textbf{#1} \ifpubQuiz@Host \hfill
  \IfNoValueTF{#2}{}{#2}\fi}
 \ifpubQuiz@Host \IfNoValueF{#3}{#3} \fi%
}
```

`\pubQuiz@SubcategoryTopMatter` The \pubQuiz@SubcategoryTopMatter command prints the subcategory name. In the host script, it also prints *notes* (in the same format as the title, at the end of the line) and *instructions* (in the regular format, before the enumerate environment). It takes two parameters

\pubQuiz@SubcategoryTopMatter {category_name}¡notes¿[instructions]

- `subcategory_name`: Name to be printed for the category title (mandatory)
- `notes`: Notes for the host script (optional)
- `instructions`: Instructions for the host script (optional)

```
\NewDocumentCommand\pubQuiz@SubcategoryTopMatter{m d<> +o}{
 \subsubsection*{\textbf{#1} \ifpubQuiz@Host \hfill
  \IfNoValueTF{#2}{}{#2}\fi}
 \ifpubQuiz@Host \IfNoValueF{#3}{#3} \fi%
}
```

`\pubQuiz@CategoryPrint` The \pubQuiz@CategoryPrint receives the body of a pubCategory environment as its only parameter.

\pubQuiz@CategoryPrint {category_environment_body}

If `category_environment_body` is blank, this command prints nothing. Otherwise, it prints `category_environment_body` inside an `enumerate` environment.

```
\NewDocumentCommand\pubQuiz@CategoryPrint{+m}{
 \IfBlankF{#1}{
 \begin{enumerate}[topsep=0px,partopsep=0px]
  #1
 \end{enumerate}}
}
```

`PubCategory` (*env.*) The environment `PubCategory` prints a Category in both the host script and the answer sheets.

\PubCategory * |number_columns| {category_name} ¡host_notes¿ [host_instructions]

- The optional stars indicates a bonus category. If given, the title is printed when the `printbonus` option is active (default setting).
- `number_columns`: the answer spaces will be written in `number_columns` columns (optional).
  If the optional parameter is not given, the lines are printed in a single column. When given, the minimum value for \`number_columns` is 2.
- `category_name`: The name of the category (mandatory)
- `host_notes`: Short notes to be printed in the host script, in-line with the title (and using the title formatting). (optional)
- `host_instructions`: Special category instructions, to be printed in the host-script. (optional)
- The body of the environment is passed automatically as a last parameter.

```
\NewDocumentEnvironment{PubCategory}{s d|| m d<> +o +b}{
 \IfBooleanTF{#1}{
  \ifpubQuiz@Bonus  \pubQuiz@CategoryTopMatter{#3}<#4>[#5]
  \else \ifpubQuiz@Host \pubQuiz@CategoryTopMatter{#3}<#4>[#5] \fi
  \fi
 }{ \pubQuiz@CategoryTopMatter{#3}<#4>[#5] }
}{
 \IfBooleanTF{#1}
  {\ifpubQuiz@Host \pubQuiz@CategoryPrint{#6} \fi}
  { \IfNoValueF{#2}{
    \ifpubQuiz@Host \else
     \begin{multicols}{#2}
     \pubQuiz@Columnstrue
    \fi
   }\pubQuiz@CategoryPrint{#6}
   \IfNoValueF{#2}{
    \ifpubQuiz@Host \else
     \end{multicols}
     \pubQuiz@Columnsfalse
    \fi
```

```
    }
   }\par
}
```

`\PubCat` The user-command \PubCat is intended for answer-sheet only usage.

\PubCat * |number_columns| {category_name} [question_count]

- The optional stars indicates a bonus category. If given, the title is printed when the `printbonus` option is active (default setting).
- `number_columns`: the answer spaces will be written in `number_columns` columns (optional).
  If the optional parameter is not given, the lines are printed in a single column.
- `category_name`: The name of the category (mandatory)
- `question_count`: The number of answer lines to print (optional).
  The default value for `question_count` is 5.

The printing is handled by the standard category environment `\PubCategory` and an internal question-printing command `\pubQuiz@PrintAnswerLines`

```
\NewDocumentCommand\PubCat{s d|| m O{5}}{
 \IfBooleanTF{#1}
  { \begin{PubCategory}*|#2|{#3}
   \end{PubCategory}}
  { \begin{PubCategory}|#2|{#3}
    \pubQuiz@PrintAnswerLines{#4}
   \end{PubCategory}
  }
}
```

`PubSubcategory` (*env.*) The environment `PubSubcategory` prints a Subcategory in both the host script and the answer sheets.

\PubSubcategory * |number_columns| {category_name} ¡host_notes¿ [host_instructions]

- The optional stars indicates a bonus category. If given, the title is printed when the `printbonus` option is active (default setting).
- `number_columns`: the answer spaces will be written in `number_columns` columns (optional).
  If the optional parameter is not given, the lines are printed in a single column. When given, the minimum value for `\number_columns` is 2.
- `category_name`: The name of the category (mandatory)
- `host_notes`: Short notes to be printed in the host script, in-line with the title (and using the title formatting). (optional)
- `host_instructions`: Special category instructions, to be printed in the host-script. (optional)
- The body of the environment is passed automatically as a last parameter.

No quick command is given for subcategories.

```
\NewDocumentEnvironment{PubSubcategory}{s d|| m d<> +o +b}{
 \item[] \vspace{-1em}
 \IfBooleanTF{#1}{
  \ifpubQuiz@Bonus  \pubQuiz@SubcategoryTopMatter{#3}<#4>[#5]
  \else \ifpubQuiz@Host \pubQuiz@SubcategoryTopMatter{#3}<#4>[#5] \fi
  \fi
 }{ \pubQuiz@SubcategoryTopMatter{#3}<#4>[#5] }
}{
 \IfBooleanTF{#1}
  {\ifpubQuiz@Host \pubQuiz@CategoryPrint{#6} \fi}
  { \IfNoValueF{#2}{
    \ifpubQuiz@Host \else
     \begin{multicols}{#2}
     \pubQuiz@Columnstrue
    \fi
   }\pubQuiz@CategoryPrint{#6}
   \IfNoValueF{#2}{
    \ifpubQuiz@Host \else
     \end{multicols}
     \pubQuiz@Columnsfalse
    \fi
   }
  }\par
}
```

3.3 Question-related environments and commands

This subsection covers internal and user commands to handle the scripting and printing of questions (in the host script) and answer lines (in the answer sheet).

`\PubQ` The user-command `\PubQ` prints an answer line in the answer sheet (for non-bonus categories) and the question (with its answer if the `printanswers` switch is active) in the host script.

\PubQ * [question] . [answer]

- The optional star indicates a bonus question (a question that should not appear on the answer sheet).
- The question is given as an optional argument. If no question is given, the text "No question given" will be printed in the host script.
- An optional dot indicates that the answer to the question should be written inline (with dots separating the question and the answer). If not given, the answer appears in its own line.
- The answer is then given as an optional argument. It is not possible to give an answer without first writing the question.

```
\NewDocumentCommand\PubQ{s O{No question given} t. o}{
 \ifpubQuiz@Host
  \item \IfNoValueF{#2}{#2}
  \ifpubQuiz@Answers
```

```
    \IfNoValueF{#4}
     {\IfBooleanTF{#3}{\dotfill}{\\}
     \textbf{#4}}
    \fi
  \else
   \IfBooleanF{#1}{
    \item \hrulefill \par
   }
  \fi
 }
```

\pubQuiz@PrintAnswerLines The \pubQuiz@PrintAnswerLines is intended for answer-sheet only usage.

\pubQuiz@CategoryPrint {question_count}

It receives the number of answer lines to print as mandatory parameter `question_count` and prints `question_count` numbered answer lines in the answer sheet. The printing is handled by the standard question command \PubQ

```
\NewDocumentCommand\pubQuiz@PrintAnswerLines{m}{
 \foreach \i in {1,...,#1}{
  \PubQ
 }
}
```

\PubQ The user-command \PubQPic prints a picture question. In the answer sheet, it prints the image and an answer line below it. In the host script, it only prints the question number and the answer (if it is given and the `printanswers` switch is active).

\PubQPic {image_file}[answer]

- The location of an image file should be given as a mandatory argument.
- The answer to the question is then given as an optional argument.

```
\NewDocumentCommand\PubQPic{m o}{
 \item \ifpubQuiz@Host
   \ifpubQuiz@Answers
     #2
   \fi
 \else  \hfill%
  \includegraphics[
   keepaspectratio=true,
   width=\pubQuiz@Pixwidth,
   height=\pubQuiz@Pixheight,
   valign=t]{#1}
  \hspace*{\fill}

  \vspace{6mm}
  \hrulefill \par
 \fi
}

\makeatother
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

A

`answers` (option) *8*

E

`\endpubQuiz@Hide` 53, 62
environments:
`PubCategory` . . *9*, 107
`PubInstructions` *9*, 55
`PubRules` *9*, 46
`PubSubcategory` *10*, 139

H

`host` (option) *8*, *11*

N

`noanswers` (option) . . . *8*
`nohost` (option) . . *8*, *11*
`noprintbonus` (option) *8*, *11*

O

options:
`answers` *8*
`host` *8*, *11*
`noanswers` *8*
`nohost` *8*, *11*
`noprintbonus` . *8*, *11*
`printbonus` . . . *8*, *11*

P

`printbonus` (option) *8*, *11*
`\PubCat` *9*, 130
`PubCategory` (env.) *9*, 107
`PubInstructions` (env.) *9*, 55
`\PubQ` 163, 179, 182
`\PubQPic` 182
`\pubQuiz@CategoryPrint` 101, 115, 121, 148, 154
`\pubQuiz@CategoryTopMatter` . 91, 109, 110, 112
`\pubQuiz@Hide` . . . 52, 61
`\pubQuiz@PageHeader` 76, 87
`\pubQuiz@PrintAnswerLines` 135, 177
`\pubQuiz@ScoreBox` 64, 82
`\pubQuiz@SubcategoryTopMatter` . 96, 142, 143, 145
`\PubRound` *8*, 86
`\PubRules` 52
`PubRules` (env.) . . . *9*, 46
`PubSubcategory` (env.) *10*, 139