# Adaptive Quiz – Solution Documentation

## Usage (for devs)

In your HTML, place an element to use as a container which for the Adaptive Quiz. Use any id you want.

<section id="astronomy"></section>

Place these scripts into your HTML page, at the end of the body, but before the </body> tag. Use the id you used for the container element. Use the relative location of your JSON file to load the data. And point the script’s “src” attribute to the relative location of the JavaScript file.

<script src="js/adaptive\_quiz.js"></script>

<script>

loadAdaptiveQuiz("data/astronomy\_quiz\_data.json", "astronomy");

</script>

## Quiz Language Term Options

Language terms change the text inside of certain quiz components like buttons and progress areas. This gives buttons a set of default text, like “Next”, or “Restart Quiz”.

You can associate different language terms for each language code (as in “en-us”, “fr-ca”, etc…). This allows all buttons and other elements to be translated automatically when the code is detected.

### Where do language terms come from?

The quiz has three possible sources for language terms. The reason to have all three options is to give maximum flexibility in developing quizzes which might require special language terms.

1. The JSON quiz data file
2. Language files
3. The default English built-in language terms

Option #1 is useful for rare situations where you need to override a specific quiz’s button names or progress text.

For most situations, option #2 is used. The page’s language is automatically detected. The associated language file has terms translated to the detected language. This can allow users to have a consistent experience across a language, rather than a one-off for one single quiz as in option #1.

If both option #1 and option #2 fail, then default English language terms are used.

### How is the language term source chosen?

The language data chosen follows this procedure. It uses whichever language data it can find first, going in this order:

1. Is there a “langTerms” property in the quiz data JSON file?
   * If yes, then use it. Otherwise, go to #2.
2. Is there a file under the “lang” folder which has a name that matches the HTML page’s language code?
   * For example, if your page’s language code is “en-us”, then the Adaptive Quiz will look for “en-us.txt”.
   * If it exists, then use the language terms in it. Otherwise go to #3.
3. Since no external language data could be found, the default English data is used.

### List of Language Terms and Default Text

|  |  |
| --- | --- |
| Name | English Default Text |
| Next Button | Next |
| Previous Button | Previous |
| Restart Button | Restart Quiz |
| Check Answer Button | Check Answer |
| Hint Button | <span class=\"icon hint\" aria-label=\"Toggle Hint\"></span> |
| Strike Zone Text  (aria-valuetext) | You have #{numRemainingToDowngrade} remaining before going to the lower level. |
| Progress Bar Text (aria-valuetext) | You have #{numRemainingToUpgrade} remaining before progressing to the next level. |

## Quiz JSON options

|  |  |
| --- | --- |
| JSON Name | Additional Info |
| quizHeading | Quiz Heading  This is what appears at the top of the quiz.  Can be text or HTML. Since HTML is allowed, you can choose to use an h1 tag or any other HTML tags within the header.  If false, no heading is shown on the quiz. |
| feedbackType | Feedback Type  The type of feedback to present to the users. Currently only “continuous” is supported. This means feedback is presented on each question as soon as the user completes the question. |
| showPoolName | Show Pool Name  Show the name of the current question pool within the quiz. Question pool names are defined in the question pool options.  (Pool options are in the 3rd table) |
| showProgressBar | Show Progress Bar  Shows a progress bar which increases in size as the user approaches a “positive” end to the question pool. The more you get correct, the bigger the progress bar.  If “numQuestionsCorrectToEnd” is an enabled end condition, and “useForProgressBar” is enabled, then the progress bar will show overall progress.  Otherwise, then pool progress is shown.  When changing pools, the progress bar will restart. |
| showStrikes | Show Strikes  Shows as many “x” symbols as there are “numIncorrectBeforeDowngrade” for the question pool. Each “x” represents a strike. In other words, the more you get wrong, the more strikes you get.  If “numQuestionsIncorrectToEnd” is an enabled end condition, and “useForStrikes” is enabled, then the progress bar will show overall progress.  Otherwise, then pool strikes are shown.  When changing pools, the strikes will restart. |
| colouredFeedback | Coloured Feedback  If this is set to *true***,** then feedback for incorrect answers will be in a red and feedback for correct answers will be in a green box.  If this is set to *false*, then feedback will be in a gray box. |
| preQuizHtml | Pre-quiz HTML content  Displays some HTML content before the quiz questions are presented. It’s HTML so it supports images, text, video, or whatever you want to put in here. |
| postQuizHtml | Post-quiz HTML content  Displays some HTML content after the quiz ends. It’s HTML so it supports images, text, video, or whatever you want to put in here. |
| endConditions | End Conditions  The conditions required to cause the quiz to end. Multiple end conditions can be applied.  End conditions are described in the next table. |
| questionPools | Question Pools  A list of question pools which each contain different questions. |

## End conditions (endConditions object)

|  |  |
| --- | --- |
| JSON Name | Additional Info |
| preface | End Condition Preface  Some HTML content that is displayed before the end conditions are listed.  If *false*, no preface will be displayed. |
| numQuestionsCorrectToEnd | If the quiz-taker answers *X* number of questions correctly (score = 1), then the quiz ends.  If false, then this end condition does not apply.  Options:   * Show (True or False) * Best practice is true * Text (html) * The text to show if “Show” is true. * E.g. “You need to answer 5 questions correctly overall to reach the end of the quiz.” |
| numQuestionsIncorrectToEnd | If the quiz-taker answers *X* number of questions incorrectly (score < 1), then the quiz ends.  If false, then this end condition does not apply. |
| noRemainingQuestionsAtCurrentPool | If true, then the quiz ends when the user has exhausted all the questions within their question pool. If the user happens to move up to the next level at this time, then the quiz does not end, it goes to the next level. But if you go back down to a level where you have exhausted all options then the quiz ends.  If false, then this end condition does not apply. And instead of ending when there are no questions in the pool, the questions are shuffled back into the queue, repeating forever.  So if you want this to be false, make sure at least one of repeatCorrect or repeatIncorrect is true in your pools. If they are both false, then the quiz ends, regardless of if “noRemainingQuestionsAtCurrentPool” is false. Because there are just no questions left. (No jumping to other pools even if they have anything left) |

## Question pools (questionPools object)

|  |  |
| --- | --- |
| JSON Name | Additional Info |
| poolName | Question Pool Name  The name of the question pool. Can be text. Some examples include “Level 1”, or “Types of Cats”, or “Stage 5”  This is really only useful if “showPoolName” is true. |
| poolId | Question Pool ID  A unique identifier for the question pool.  This is used so you can refer to a pool when upgrading/downgrading pools.  Can be a number or text. |
| poolLevel | The Level of the Question Pool  This is used to determine the highest level the user achieved which can be displayed using a replace string. |
| downgradePoolId | ID of Question Pool to switch to, if the number of incorrect questions limit is reached. |
| upgradePoolId | ID of Question Pool to switch to, if the number of incorrect questions limit is reached. |
| numIncorrectBeforeDowngrade | Number of Incorrect Answers before switching to the question pool with id = “DowngradePoolId”  The quiz will switch to another pool if the value for “numIncorrectBeforeDowngrade” is reached. The pool it switches to is defined by the downgradePoolId. |
| numCorrectBeforeUpgrade | Number of Correct Answers before switching to the question pool with id = “UpgradePoolId”  The quiz will switch to another pool if the value for “numCorrectBeforeUpgrade” is reached. The pool it switches to is defined by the upgradePoolId. |
| repeatWrong | Repeat questions that were answered incorrectly.  If true, then questions that the user gets wrong will be reappear within the question pool.  If both repeatCorrect and repeatIncorrect are false, then you only get questions you have not seen before. And if you run out, then the quiz ends. |
| repeatCorrect | Repeat questions that were answered correctly.  If true, then questions that the user gets right will reappear in the question pool after the rest of the questions are done.  They only reappear in the pool if you answer the questions you got wrong first. (If repeatIncorrect is true.) And if you run out the quiz ends. |
| feedbackOnUpgradeHtml | Feedback text  This can be HTML, so you can put whatever you want for feedback to a specific question pool.  Pool feedback will appear on its own page.  Appears whenever you UPGRADE pools.  If this is “false” then no feedback. |
| feedbackOnDowngradeHtml | Feedback text  This can be HTML, so you can put whatever you want for feedback to a specific question pool.  Pool feedback will appear on its own page.  Appears whenever you DOWNGRADE pools.  If this is “*false*” then no feedback. |
| randomizeQuestionOrder | Randomize Question Order  If *true*, the order of questions will be shuffled when the user starts the question pool.  If *false*, the order in the JSON config file will be respected. |
| questions | Questions  Contains the list of questions. Question format is described in the next table. |
| langTerms | Language Terms which would override all other language terms  Only use this if you want to override the default language terms of all languages. Otherwise, edit the language file in the “lang” folder.  Possible JSON Example:  "langTerms": {  "nextButton": "Next",  "previousButton": "Previous",  "restartButton": "Restart Quiz",  "checkAnswerButton": "Check Answer",  "hintButton": "<span class=\"icon hint\" aria-label=\"Toggle Hint\"></span>",  "strikeZoneText": "You have #{numRemainingToDowngrade} remaining before going to the lower level.",  "progressBarText": "You have #{numRemainingToUpgrade} remaining before progressing to the next level."  } |

## Questions (questions object)

|  |  |
| --- | --- |
| JSON Name | Additional Info |
| questionType | Question Type  Can be:   * Multiple Choice (MC, True False, TF) * Multi-select (MS, All that Apply, ATA) * Matching (Match, ord, ordering, sorting, sort) * Fill in the Blank (FITB)   “All That Apply” and “True False” are aliases for Multi-select and multiple choice as described above.  Any alias, shorthand or full text is valid. And they are not case sensitive. |
| questionHtml | Question HTML Content  Can contain images, video, etc. because it’s any HTML you want to use. |
| hintHtml | Hint HTML Content  Can contain images, video, etc. because it’s any HTML you want to use. |
| randomizeOptions  [MC, MS, Matching] | Randomize Options  Randomizes the options when the question is first presented.  Not applicable to Fill in the Blank type questions. |
| feedbackForPerfect  [MS, Matching, FITB] | Feedback for a perfect score  Can be text or HTML.  Feedback appears after checking the answer. |
| feedbackForImperfect  [MS, Matching, FITB] | Feedback for a wrong or partially correct answer  Can be text or HTML.  Feedback appears after checking the answer. |
| caseSensitive  [Fill in the Blank] | Case Sensitivity toggle for Fill in the Blank questions  If set to *false*, the text input will be case sensitive when checking the answer.  If set to *true*, the letter case is checked with the answer. |
| blankSize  [Fill in the Blank] | Blank Size, for Fill in the Blank questions  Sets the “size” attribute for the input box. The size here is referring to the horizontal length of the box.  (The input box replaces the “#{blank}” keyword within the questionHtml.)  Should be a number. |
| blankMaxLength  [Fill in the Blank] | Blank Max Length for Fill in the Blank questions  The max number of characters allowed in the input box. Sets the “max-length” attribute for the input box.  (The input box replaces the “#{blank}” keyword within the questionHtml.)  Should be a number. |
| answers  [Fill in the Blank] | List of Possible Answers for Fill in the Blank questions  This should contain the list of possible answers that would be considered correct for the user to input in the blank.  If the user inputs the same text, then they’ve answered correctly. Any different text will be counted as incorrect.  Example format for the answers JSON…  questionHtml: “The season after summer is… #{blank}”,  answers: [“autumn”, “fall”] |
| options  [MC, MS, Matching] | Options  A list of options to show as answers for the question. |

## Options (options object)

|  |  |
| --- | --- |
| JSON Name | Additional Info |
| score  [MC, MS] | Option score  The score for the selected option. Can be 1 or 0. If the property is missing, then the option gets a score of 0.  Use score=1 to denote the “correct” answer. Multiple Choice questions should only have one correct answer. Multi-Select questions can have any number of answers. |
| optionText  [MC, MS, Matching] | Option text  The text to display for the option. Can be HTML.  In a Multiple Choice and Multi-Select question, this is one of the possible things the user can select.  In a Matching question, this is the property to match select box options to. |
| feedback  [MC] | Feedback for specific option  Can be text or HTML (including pictures or video).  Displayed to user when the feedback is shown and the user selected this specific option. |
| matchingText  [Matching] | Matching text for a specific option  In a matching question, the matching text will appear in a selector box beside each optionText. This can be a number or text.  matchingText that is duplicated in the same question will be combined into one option in the selector box.  Only works in Matching. |

## Replace Strings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Replace String | | Valid Components/Pages | | Description | |
| #{quizName} | | * Feedback text * Pre-upgrade page * Pre-downgrade page * Post-quiz page | | The name of the quiz. | |
| #{currentPool} | | * Feedback text * Pre-upgrade page * Pre-downgrade page | | The current question pool’s name. | |
| #{upgradePool} | | * Feedback text * Pre-upgrade page * Pre-downgrade page | | The name of the question pool you upgrade to. | |
| #{downgradePool} | | * Feedback text * Pre-upgrade page * Pre-downgrade page | | The name of the question pool you downgrade to. | |
| #{numRemainingToUpgrade} | | * Feedback text * Pre-upgrade page * Pre-downgrade page | | Number of remaining questions you need to answer correctly to upgrade to the next question pool. | |
| #{numRemainingToDowngrade} | | * Feedback text * Pre-upgrade page * Pre-downgrade page | | Number of incorrect answers you need to make before you get sent to the downgrade question pool. | |
| #{endingLevel} | | * Post-quiz page | | The level you have at the end, just before the postQuizHtml page appears. | |
| #{highestLevelAchieved} | | * Post-quiz page | | The name of the question pool you completed that has the highest level value (“poolLevel” property in JSON.) | |