AcroTEX.Net

The dljslib package
English number format

D. P. Story

This article documents and discusses the option, `useEnNums`, for the dljlib package designed to work with numbers in the English format. The English number format is a format that uses the period (.) as the decimal mark (separator) for a decimal number. The word “English” is used as is a general designator for all countries that use this number format.

This `useEnNums` option requires recent versions of both `exerquiz` (2021/04/12) and dljslib (2021/04/04).

AeB Blog April 16, 2021

dpstory@acrotex.net
Version 2.0

1. Introduction

This article concerns entering English formatted decimal numbers into a text field created by `\RespBoxMath`. Acrobat/Adobe Reader do support English format numbers, but this is a general purpose feature and does not integrate well into the `exerquiz` way of doing things; after the number is entered, the number is submitted to the JavaScript engine for analysis to determine if it is the correct answer.

The purpose of the `useEnNums` is to support a English decimal number. The period (.) is supported as the decimal separator, but the thousands separator (,) is not supported.

2. Discussion and documentation

The first example below, illustrates the default behavior of `exerquiz`, which natively supports a decimal separator of period (.). The `useEnNums` option of `dljslib` is not needed for this example or for the example following it.

(Ans: `4.2`)

```
\RespBoxMath{4.2}{1}{.00001}{[0,1]}
```

The response `4.2` is correct, but as any number decimal places are accepted in this form, `4.200000001` is correct as well.

Non-number expressions are valid in `\RespBoxMath` (these non-numbers expressions can be declared variables and mathematical functions that are supported; for example,

(Ans: `17.88 x^(0.5)`)

```
\RespBoxMath{17.88 x^(0.5)}{3}{.00001}{[0,1]}
```

By the way, the coefficient and exponent can correctly be expressed as a fraction.

We now discuss the circumstances under which the `useEnNums` option of the `dljslib` packages come into play.

Inputting numerical values only. For questions (`\RespBoxMath`) that require a numerical input only, commands defined by the `useEnNums` option can filter user input to verify input is a number (in the English decimal format).

- **Force the use of (decimal) numbers only.** The `\numEn` filter requires a number in the English number format; this is a general number requirement; integers and decimal numbers are accepted, but rational numbers and other non-numeric expressions are not allowed.

  (Ans: `4.2`)

```
\RespBoxMath{4.2}{1}{.00001}
  {[0,1]}[\numEn{\MsgEni}]
```

Correct answer is `4.2`, the number `4.0000001` is acceptable and correct; `42/10` is not acceptable input. The critical code is,

`[\numEn{\MsgEni}]` (placement shown underlined above)

which is inserted after the interval specification. The first argument of `\numEn` is a message if the user does not meet the input requirement. The definition of `\MsgEni` is as follows:

```
\flJSStr*[noquotes]{\MsgEni}{"English decimal notation is
expected, for example: 12.3456."}
```

- **Require exactly *n* decimal places.** The `\rndNumEnReq` filter requires a decimal number (using English notation) with a specified number of decimal places; for example,

  (Ans: `4.20`)

  ```
  \RespBoxMath{4.2}{1}{.00001}
    {[0,1]}[\rndNumEnReq{2}{\MsgEnii}]
  ```

  Here, the correct response `4.20`, whereas, `4.2` is not correct (not two decimal places). The critical code is,

  `[\rndNumEnReq{2}{\MsgEnii}]` (placement shown underlined above)

  which is inserted after the interval specification. The first argument of `\rndNumEnReq` is the number of decimal places required; the second argument is a message if the user does not meet the input requirement. The definition of `\MsgEnii` is as follows:

  ```
  \flJSStr*[noquotes]{\MsgEnii}{"A decimal number is required,
  rounded to two decimal places, for example: 12.34"}
  ```

  If you require three decimal places, then define your own alert message based on this example:

  ```
  \flJSStr*[noquotes]{\MsgEniii}{"A decimal number is required,
  rounded to three decimal places, for example: 12.345"}
  ```

- **Require at most *n* decimal places.** The `\rndNumEnOpt` requires a number, in English decimal notation, with at most *n* decimal places.

  (Ans: `4.2`)

  ```
  \RespBoxMath{4.2}{1}{.00001}
    {[0,1]}[\rndNumEnOpt{2}{\MsgEnii}]
  ```

Here, the user needs to input `4.2` or `4.20` for a correct response. The critical code is,

`[\rndNumEnOpt{2}{\MsgEnii}]` (placement shown underlined)

which is inserted after the interval specification. The first argument of `\rndNumEnOpt` is the number of decimal places required; the second argument is a message if the user does not meet the input requirement. The definition of `\MsgEnii` is given above. Compare the response of the field to entering `4.2`, `4.20`, and `4.200`. A decimal place is not required in this form; for example, `4` is a valid input, but is a wrong answer.

3. Comprehensive Examples

We conclude this article a quiz that incorporates the ideas present earlier.

Start Answer each of the following, passing is 100%.

1. Differentiate: $4.78x^{0.4} =$ □ (Ans: `4.78x^(0.4)`)

2. Enter the number 4.2 in English format:

   □ (Ans: 4.2)

3. Enter the number 4.237 in English format, round to two digits:

   □ (Ans: 4.24)

4. Enter the number 3.14159 in English format, round to three digits:

   □ (Ans: 3.142)

End Score: Correct

Ans: □