mfirstuc.sty v1.05: uppercasing first letter

Nicola L.C. Talbot

Dickimaw Books
http://www.dickimaw-books.com/

2012-05-21

The glossaries bundle is supplied with the package mfirstuc which provides the command:

\makefirstuc

> `\makefirstuc{`⟨*stuff*⟩`}`

This makes the first object of ⟨*stuff*⟩ uppercase unless ⟨*stuff*⟩ starts with a control sequence followed by a non-empty group, in which case the first object in the group is converted to uppercase. Examples:

- `\makefirstuc{abc}` produces Abc.

- `\makefirstuc{\emph{abc}}` produces *Abc* (\MakeUppercase has been applied to the letter "a" rather than \emph). Note however that

  `\makefirstuc{{\em abc}}`

  produces *ABC* (first object is {\em abc} so equivalent to \MakeUppercase{\em abc}), and

  `{\makefirstuc{\em abc}}`

  produces abc (\em doesn't have an argument therefore first object is \em so equivalent to {\MakeUppercase{\em}abc}).

- `\makefirstuc{{\'a}bc}` produces Ábc.

- `\makefirstuc{\ae bc}` produces Æbc.

- `\makefirstuc{{\ae}bc}` produces Æbc.

- `\makefirstuc{{ä}bc}` produces Äbc.

Note that non-Latin or accented characters appearing at the start of the text must be placed in a group (even if you are using the inputenc package) due to expansion issues.

> In version 1.02 of mfirstuc, a bug fix resulted in a change in output if the first object is a control sequence followed by an empty group. Prior to version 1.02, \makefirstuc{\ae{}bc} produced æBc. However as from version 1.02, it now produces Æbc.

Note also that

```
\newcommand{\abc}{abc}
\makefirstuc{\abc}
```

produces: ABC. This is because the first object in the argument of \makefirstuc is \abc, so it does \MakeUppercase{\abc}. Whereas:

```
\newcommand{\abc}{abc}
\expandafter\makefirstuc\expandafter{\abc}
```

produces: Abc. There is a short cut command which will do this:

\xmakefirstuc   \xmakefirstuc{⟨*stuff*⟩}

This is equivalent to \expandafter\makefirstuc\expandafter{⟨*stuff*⟩}. So

```
\newcommand{\abc}{abc}
\xmakefirstuc{\abc}
```

produces: Abc.

> \xmakefirstuc only performs one level expansion on the first object in its argument. It does not fully expand the entire argument.

If you want to use an alternative command to convert to uppercase, for example \MakeTextUppercase,[1] you can redefine \glsmakefirstuc. For example:

```
\renewcommand{\glsmakefirstuc}[1]{\MakeTextUppercase #1}
```

New to mfirstuc v1.06:

\capitalisewords   \capitalisewords{⟨*text*⟩}

This command apply \makefirstuc to each word in ⟨*text*⟩ where the space character is used as the word separator. Note that it has to be a plain space

---

[1] defined in the textcase package

character, not another form of space, such as ~ or \space. Note that no expansion is performed on ⟨*text*⟩.

\xcapitalisewords

> \xcapitalisewords{⟨*text*⟩}

This is a short cut for \expandafter\capitalisewords\expandafter{⟨*text*⟩}.
Examples:

1. \capitalisewords{a book of rhyme.}

   produces: A Book Of Rhyme.

2. \capitalisewords{a book\space of rhyme.}

   produces: A Book of Rhyme.

3. \newcommand{\mytitle}{a book\space of rhyme.}
   \capitalisewords{\mytitle}

   produces: A BOOK OF RHYME. (No expansion is performed on \mytitle, so ⟨*text*⟩ consists of just one "word".) Compare with next example:

4. \newcommand{\mytitle}{a book\space of rhyme.}
   \xcapitalisewords{\mytitle}

   produces: A Book of Rhyme.