The xspace package*

David Carlisle

1997/10/13

Abstract

\xspace should be used at the end of a macro designed to be used mainly in text. It adds a space unless the macro is followed by certain punctuation characters.

1 Introduction

After `\newcommand{\gb}{Great Britain\xspace}`
`\gb is a very nice place to live.`
Great Britain is a very nice place to live.
`\gb, a small island off the coast of France.`
Great Britain, a small island off the coast of France.

`\xspace` saves the user from having to type `\␣` or `{}` after most occurrences of a macro name in text. However if either of these constructions follows `\xspace`, a space is not added by `\xspace`. This means that it is safe to add `\xspace` to the end of an existing macro without making too many changes in your document.

Sometimes `\xspace` may make the wrong decision, and add a space when it is not required. In these cases follow the macro with `{}`, as this has the effect of suppressing the space.

Note that this package must be loaded *after* any language (or other) packages that make punctuation characters 'active'.

2 The Macros

```
⟨*package⟩
```

`\xspace` `\xspace` just looks ahead, and then calls `\@xspace`.

```
\DeclareRobustCommand\xspace{\futurelet\@let@token\@xspace}
```

`\@xspace` If the next token is one of a specified list of characters, do nothing, otherwise add a space. If you often use a different punctuation character, add the appropriate line (do not forget the `\fi` at the end!)

```
\def\@xspace{%
  \ifx\@let@token\bgroup\else
  \ifx\@let@token\egroup\else
  \ifx\@let@token\/\else
```

*This file has version number v1.06, last revised 1997/10/13.

```
\ifx\@let@token\ \else
\ifx\@let@token~\else
\ifx\@let@token.\else
\ifx\@let@token!\else
\ifx\@let@token,\else
\ifx\@let@token:\else
\ifx\@let@token;\else
\ifx\@let@token?\else
\ifx\@let@token/\else
\ifx\@let@token'\else
\ifx\@let@token)\else
\ifx\@let@token-\else
\ifx\@let@token\@xobeysp\else
\ifx\@let@token\space\else
\ifx\@let@token\@sptoken\else
 \space
```

When I first wrote this there were about half a dozen `\fi` here, and I thought that was bad enough.

```
  \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}
⟨/package⟩
```