fixedWidthCat  Control the output of show methods

Description

fixedWidthCat makes sure that the output of a show method fits on the page by inserting line
breaks into long strings.

numName converts an integer to it’s literal name.

sepInt prints integers with a comma as separator between 1000s

Usage

fixedWidthCat(x, width=getOption("width"))

Arguments

x An R object which is to be shown.

width The number of characters after which lines are to be broken.

Value

A character vector of the output with long lines broken

Author(s)

Florian Hahne

Examples

long <- paste(rep(letters[1:24], 5), sep="", collapse="")
fixedWidthCat(long)
Description

Usage of predefined markup commands for layout of Bioc Case Studies book.

Details

The following markup commands, LaTeX makros and environments are available for controlling the layout and structure of the book:

Ex: environment for exercise chunks.
solution: environment for solutions to the exercises.
\myincfig: makro for figure environments with three parameters: (1) figure filename (2) figure width (3) figure caption
\solfig: makro for figure environments within solution chunks. This is necessary because LaTeX doesn’t allow for floats within minipage environments.
\myref: reference to other labs/chapters. For the book this is a simple wrapper around ref ignoring the second argument, for the labs this command is replaced in the useRlabs.sty file allowing for referencing between the individual documents.
\booklab: macro for conditional text input with two parameters. The first parameter will be used for the book while the second will be used for the labs.

The following makros will automatically create index entries as side effect. Apart from that they do text highlighting as well.

\R: the R language glyph (in sans serif font)
\Rpackage: an R package (in bold font)
\Rclass: an R class (in italics)
\Rmethod: an R method (in small typewriter font)
\Rfunction: an R function (in small typewriter font)

Implicit index terms can be generated using

\indexTerm: with the optional first argument giving the actual term and the second argument giving a string that appears in the text. E.g. indexTerm[tree]{trees} would give you “trees” in the text but create an index for “tree”. Omitting the optional first argument will create an index for the same string that appears in the text.

Some more useful text markup that doesn’t create indices:

\Robject: an R object (in small typewriter font)
\Rfunarg: the argument to an R function (in italics)
\code: typewriter font
\term: whatever \{emph\} is set to
\section*{mySessionInfo}

\textit{Wrapper around sessionInfo}

\begin{description}
\item[Description] This will produce the \LaTeX{} output for the sessionInfo and the references at the end of each lab.
\item[Usage] \func{mySessionInfo}(ref=TRUE)
\item[Arguments]
\begin{itemize}
\item[ref] logical controlling whether to include references
\end{itemize}
\item[Value] This function is called for its side effects
\end{description}

\section*{Author(s)}

Florian Hahne
**parseLibVers**  
*Parse the library versions*

**Description**
This is a helper function to check for valid package versions

**Usage**
```r
parseLibVers()
```

**Value**
Called for its side effects

**Author(s)**
Florian Hahne

---

**requiredPackages**  
*check for missing and outdated packages*

**Description**
Both functions compare the Depends field of the DESCRIPTION of the BiocCaseStudies package. requiredPackages is run before a build of the book. It throws an error if there are any missing or outdated packages. packages2install returns a character vector of packages that need (re)installing.

**Usage**
```r
requiredPackages(load=FALSE)
packages2install()
```

**Arguments**
- `load` Logical. Should all packages be loaded?

**Value**
- `requiredPackages` returns `invisible(NULL)`. The function is called for its side effects.
- `packages2install` returns a character vector that can be passed to the biocLite function from the biocLite.R script.

**Author(s)**
Florian Hahne
resample

Examples

```r
## Not run:
biocLite(packages2install())

## End(Not run)
```

---

resample  

Resample from ALL ExpressionSet and plot

Description

A function to resample samples from each class of an ExpressionSet and plot the results calculated by a function that returns the number of differentially expressed genes between the classes.

Usage

```r
resample(x, selfun, groupsize = 6 * (1:6), nrep = 25)
```

Arguments

- `x`: An ExpressionSet object derived from the ALL data package.
- `selfun`: A function that takes the resampling subset of the ExpressionSet and computes the number of differentially expressed genes between the two classes.
- `groupsize`: The number of samples for each class.
- `nrep`: The number of iterations of resampling procedure.

Value

The function is called for the side effect of producing a plot.

Author(s)

Florian Hahne
Index

*Topic misc*
  fixedWidthCat, 1
  markup, 2
  mySessionInfo, 3
  parseLibVers, 4
  requiredPackages, 4
  resample, 5

booklab (markup), 2

dcol1 (markup), 2
dcol2 (markup), 2
dcol3 (markup), 2

Ex (markup), 2

fixedWidthCat, 1

indexTerm (markup), 2

lcol1 (markup), 2
lcol2 (markup), 2
lcol3 (markup), 2

markup, 2
myincfig (markup), 2
myref (markup), 2
mySessionInfo, 3

numName (fixedWidthCat), 1

packages2install (parseLibVers), 4
parseLibVers, 4

requiredPackages, 4
resample, 5

sepInt (fixedWidthCat), 1
solfig (markup), 2
solution (markup), 2