match_parens find mismatches of various brackets and quotes

doc generated from the script with gendoc — ruby script, version=1.43

Synopsis

```
match_parens [filename]
```

Options:

`-p,--pairs=S`	set matching pairs to S (default: `{}[]()”””“‘‘‘’’’`)
`-n,--number=N`	set number of mismatching characters shown to N (default: 10)
`-l,--latex`	convert ‘‘...’’ to “...” before testing
`-V,--version`	print version and exit
`-h,--help`	print short help information and exit
`--test`	do an internal test and exit

Description

Mismatches of parentheses, braces, (angle) brackets, especially in TeX sources which may be rich in those, may be difficult to trace. This little script helps you by writing your text to standard output, after adding a left margin to your text, which will normally be almost empty, but will clearly show up to 10 mismatches. (Just try me on myself to see that the parenthesis starting this sentence will not appear to be matched at the end of the file. If you look at me in the vim editor, then select this paragraph and try the command: `:!%`.

By default, the following pairs are tested:

`()`	round brackets or parentheses
`{}`	curly brackets or braces
`[]`	square brackets
`<>`	angle brackets (within html text only)
`””`	ASCII double quotes
`“”`	Unicode double quotation marks
`’’`	ASCII single quotes
`‘’`	Unicode single quotation marks

The exit value of the script is 0 when there are no mismatches, 1 otherwise.

Angle brackets are only looked for inside HTML text, where HTML is supposed to start with `<html>` or `=begin␣rdoc` and to end with `</html>` or `=end`.

Options

`-p,--pairs=S`
: Set matching pairs to S (default: `{}[]()”””“‘‘‘’’’`). For example, if you want to look for mismatching ASCII single quotes *only*, use `--pairs=”’’”`. Or, if you want to match braces and guillemets only, use `-p␣«»`. Note that if html is detected in your text, `<>` is automatically added to the pairs list. So by default, `<...>` is tested only in html, but you can test that in other text by specifying the `<>` pair in the `--pairs` option.

`-n,--number=N`
: Set number of mismatching characters shown to N. By default, only the first 10 are shown.

`-l,--latex`
: convert ‘‘...’’ to “...”‘ before testing.

`-V,--version`

print this script's version and exit.

`-h,--help`
: print short help information and exit.

`--test`
: do an internal test and exit. Note that if, with the `--pairs` option, you specify an other pairs list than the default, the test will probably fail, but you can still see the effects of your pairs list on the test data.

Examples

Suppose we have two files, `good` and `bad`, containing these texts:

```
good:                           bad:
This is a (simple) test         This is a (simple test
without mismatches              containing mismatches
```

then here are some usage examples. First a simple test on these files:

```
$ match_parens good
      1 |       | This is a (simple) test
      2 |       | without mismatches
$ echo $?
0
$ match_parens bad
      1 | (     | This is a (simple test
      2 | (     | containing mismatches
$ echo $?
1
```

Just report if there are mismatches:

```
$ match_parens good >/dev/null && echo fine || echo problems
fine
$ match_parens bad >/dev/null && echo fine || echo problems
problems
```

Report all tex files with mismatches in the current directory:

```
$ for i in *.tex; do match_parens $i >/dev/null || echo $i; done
```

Matches must be in correct order:

```
$ echo -e "This is a ([simple)] test\n" | match_parens
      1 ([)]    This is a ([simple)] test
      2 ([)]
```

Changes

Changes with respect to version 1.41:

- test on more quote characters (single, double, ASCII, Unicode)
- option for changing the character pairs to be tested
- conversion of latex' "..." is now an option
- built-in test option `--test`

Author and copyright

`Author` Wybo Dekker
`Email` wybo@dekkerdocumenten.nl