Line Categories 2/lcats

Purpose

We are going to need to identify lines of source code as falling into 18 different categories – the start of a definition, a piece of a comment, and so on. In this section we define constants to enumerate these categories, and provide a debugging routine to show the classification we are using on the web we've just read.

Definitions

¶1. The line categories are enumerated as follows:

define `$NO_LCAT 0` *none set as yet*
define `$COMMENT_BODY_LCAT 1`
define `$MACRO_DEFINITION_LCAT 2`
define `$BAR_LCAT 3`
define `$INDEX_ENTRY_LCAT 4`
define `$PURPOSE_LCAT 5`
define `$INTERFACE_LCAT 6`
define `$GRAMMAR_LCAT 7`
define `$DEFINITIONS_LCAT 8`
define `$PARAGRAPH_START_LCAT 9`
define `$BEGIN_VERBATIM_LCAT 10`
define `$TEXT_EXTRACT_LCAT 11`
define `$BEGIN_DEFINITION_LCAT 12`
define `$GRAMMAR_BODY_LCAT 13`
define `$INTERFACE_BODY_LCAT 14`
define `$CODE_BODY_LCAT 15`
define `$CONT_DEFINITION_LCAT 19`
define `$SOURCE_DISPLAY_LCAT 16`
define `$TOGGLE_WEAVING_LCAT 17`
define `$COMMAND_LCAT 18`

---

§**1.** The scanner is intended for debugging `inweb`, and simply shows the main result of reading in and parsing the web:

```
sub scan_line_categories {
    my $sigil = $_[0];
    my $confine_to = -1;
    my $sn;
    my $i;
    for ($sn=0; $sn<$no_sections; $sn++) {
        if ($section_sigil[$sn] eq $sigil) {
            $confine_to = $sn;
        }
    }
    for ($i=0; $i<$no_lines; $i++) {
        if (($confine_to >= 0) && ($confine_to != $line_sec[$i])) { next; }
        print sprintf("%04d  %16s  %s\n",
            $i, category_name($line_category[$i]), $line_text[$i]);
    }
}
```

§**2.** And the little routine which prints category names to `stdout`:

```
sub category_name {
    my $cat = $_[0];
    if ($cat == $COMMENT_BODY_LCAT) { return "COMMENT_BODY"; }
    elsif ($cat == $MACRO_DEFINITION_LCAT) { return "MACRO_DEFINITION"; }
    elsif ($cat == $BAR_LCAT) { return "BAR"; }
    elsif ($cat == $INDEX_ENTRY_LCAT) { return "INDEX_ENTRY"; }
    elsif ($cat == $PURPOSE_LCAT) { return "PURPOSE"; }
    elsif ($cat == $INTERFACE_LCAT) { return "INTERFACE"; }
    elsif ($cat == $GRAMMAR_LCAT) { return "GRAMMAR"; }
    elsif ($cat == $DEFINITIONS_LCAT) { return "DEFINITIONS"; }
    elsif ($cat == $PARAGRAPH_START_LCAT) { return "PARAGRAPH_START"; }
    elsif ($cat == $BEGIN_VERBATIM_LCAT) { return "BEGIN_CODE"; }
    elsif ($cat == $TEXT_EXTRACT_LCAT) { return "TEXT_EXTRACT"; }
    elsif ($cat == $BEGIN_DEFINITION_LCAT) { return "BEGIN_DEFINITION"; }
    elsif ($cat == $GRAMMAR_BODY_LCAT) { return "GRAMMAR_BODY"; }
    elsif ($cat == $INTERFACE_BODY_LCAT) { return "INTERFACE_BODY"; }
    elsif ($cat == $CODE_BODY_LCAT) { return "CODE_BODY"; }
    elsif ($cat == $SOURCE_DISPLAY_LCAT) { return "SOURCE_DISPLAY"; }
    elsif ($cat == $TOGGLE_WEAVING_LCAT) { return "TOGGLE_WEAVING"; }
    elsif ($cat == $COMMAND_LCAT) { return "COMMAND"; }
    elsif ($cat == $CONT_DEFINITION_LCAT) { return "CONT_DEFINITION"; }
    else { return "? cat $cat"; }
}
```