```
$foo{$a,$b,$c}
```

it really means

```
$foo{join($;, $a, $b, $c)}
```

But don't put

```
@foo{$a,$b,$c}      # a slice--note the @
```

which means

```
($foo{$a},$foo{$b},$foo{$c})
```

Default is "\034", the same as SUBSEP in **awk**. If your keys contain binary data there might not be any safe value for `$;`. (Mnemonic: comma (the syntactic subscript separator) is a semi-semicolon. Yeah, I know, it's pretty lame, but `$,` is already taken for something more important.)

Consider using "real" multidimensional arrays as described in *perllol*.

$ #

The output format for printed numbers. This variable is a half-hearted attempt to emulate **awk**'s OFMT variable. There are times, however, when **awk** and Perl have differing notions of what counts as numeric. The initial value is "%.*n*g", where *n* is the value of the macro DBL_DIG from your system's *float.h*. This is different from **awk**'s default OFMT setting of "%.6g", so you need to set `$#` explicitly to get **awk**'s value. (Mnemonic: # is the number sign.)

Use of `$#` is deprecated.

HANDLE->format_page_number(EXPR)

$ FORMAT_PAGE_NUMBER

$ %

The current page number of the currently selected output channel. Used with formats. (Mnemonic: % is page number in **nroff**.)

HANDLE->format_lines_per_page(EXPR)

$ FORMAT_LINES_PER_PAGE

$ =

The current page length (printable lines) of the currently selected output channel. Default is 60. Used with formats. (Mnemonic: = has horizontal lines.)

HANDLE->format_lines_left(EXPR)

$ FORMAT_LINES_LEFT

$ -

The number of lines left on the page of the currently selected output channel. Used with formats. (Mnemonic: lines_on_page - lines_printed.)

@LAST_MATCH_START