The {(...)} "expression markup" allows for a variety of string and formatting operations to be performed from within markup. Operations defined by this recipe include *substr*, *ftime*, *strlen*, *rand*, *toupper / tolower*, *ucfirst*, *ucwords*, *pagename* and *asspaced*.

substr

The "substr" expression extracts portions of a string. The arguments are

1. the string to be processed. Always quote the string to be processed.
2. the initial position of the substring. Note that the initial position argument is zero-based (i.e., the first character is referenced via a "0").
3. the number of characters to extract

```
 {(substr "PmWiki" 2 3)} {(substr "PmWiki" 2)} {(substr "PmWiki" 0 1)}
{(substr "PmWiki" 0 -3)} {(substr "PmWiki" -3)}
```

```
Wik Wiki P Pm W? iki
```

To obtain the last n characters of a string use `{(substr "string" -n)}`

To truncate the last n characters of a string use `(substr "string" 0 -n)}`

ftime

"Ftime" expressions are used for date and time formatting. The generic form is

```
{(ftime "fmt" "when")}
{(ftime fmt="fmt" when="when")}
```

where *fmt* is a formatting string and *when* is the time to be formatted. The arguments can be in either order and may use the optional "fmt=" and "when=" labels.

Examples:

```
 {(ftime)} {(ftime fmt="%F %H:%M")} {(ftime %Y)} {(ftime fmt=%T)} {(ftime when=tomorrow)} {(ftime fmt="%Y-%m-%d" yesterday)} {(ftime "+1 week" %F)} {(ftime fmt=%D "+1 month")} {(ftime fmt="%a%e %b" when="next week")}
```

```
 28/02/2021 à 10:56 2021-02-28 10:56 2021 10:56:49 01/03/2021 à 00:00 2021-02-27 2021-03-07 03/28/21 dim. 7 mars
```

The *fmt* parameter is whatever is given by "fmt=", the first parameter containing a '%', or else the site's default. The formatting codes are described at http://php.net/strftime. In addition to those, '%F' produces ISO-8601 dates, and '%s' produces Unix timestamps.
Some common formatting strings:

```
%F # ISO-8601 dates "2021-02-28" %s # Unix timestamp "1614506209" %H:%M:%S #
time as hh:mm:ss "10:56:49" %m/%d/%Y # date as mm/dd/yyyy "02/28/2021" "%A, %B
%d, %Y" # in words "dimanche, février 28, 2021"
```

The *when* parameter understands many different date formats. The when parameter is whatever is given by "when=", or whatever parameter remains after determining the format parameter. Some examples:

```
2007-04-11 # ISO-8601 dates 20070411 # dates without hyphens, slashes, or dots
2007-03 # months @1176304315 # Unix timestamps (seconds since 1-Jan-1970 00:00
UTC) now # the current time today # today @ 00:00:00 yesterday # yesterday @
00:00:00 "next Monday" # relative dates "last Thursday" # relative dates "-3
days" # three days ago "+2 weeks" # two weeks from now
```

Note: If you want to convert a Unix timestamp you **must** prefix with the @. Thus, `"{(ftime "%A, %B %d, %Y" @1231116927)}"`.

The *when* parameter uses PHP's strtotime function to convert date strings according to the GNU date input formats; as of this writing it only understands English phrases in date specifications.

The variable $FTimeFmt can be used to override the default date format used by the "ftime" function. The default $F Time Fmt? is `$TimeFmt`.

strlen

The "strlen" expression returns the length of a string. The first argument is the string to be measured.

```
{(strlen "{$:Summary}")}    32
```

rand

The "rand" expression returns a random integer. The first argument is the minimum number to be returned and the second argument is the maximum number to be returned. If called without the optional min, max arguments rand() returns a pseudo-random integer between 0 and RAND_MAX. If you want a random number between 5 and 15 (inclusive), for example, use (rand 5 15).

```
{(rand)} {(rand 1 99)}    1987807289 43
```

toupper / tolower

The "toupper" and "tolower" expressions convert a string into uppercase or lowercase. The first argument is the string to be processed.

```
 {(toupper "{$:Summary}")} {(tolower      STRING AND FORMATTING OPERATIONS
"{$:Summary}")}                          string and formatting operations
```

ucfirst / ucwords

The "ucfirst" expression converts to uppercase the first character of the string, and "ucwords", the first character of each word. The first argument is the string to be processed.

```
 {(ucfirst "{$:Summary}")} {(ucwords       String and formatting operations
"{$:Summary}")}                          String And Formatting Operations
```

pagename

The "pagename" expression builds a pagename from a string. The first argument is the string to be processed.

```
{(pagename "{$:Summary}")}    Pm Wiki.String And Formatting Operations?
```

asspaced

The "asspaced" expression formats wikiwords. The first argument is the string to be processed.

```
        {(asspaced "{$FullName}")}    Pm Wiki.Markup Expressions
```

Nesting expressions

Markup expressions can be nested:

```
            {(tolower (substr "Hello World" 2))}    llo world
```

Notes

- For Pm Wikis? version 2.2.33 or older, the string-processing expressions may not work properly on multibyte UTF-8 characters. Newer versions should work fine.

See also

- Page variables, Page text variables
- Conditional markup
- Cookbook:MarkupExpressionSamples -- custom markup expression samples
- Cookbook:MarkupExprPlus