

NAME

HTML::AsSubs - functions that construct a HTML syntax tree

VERSION

This document describes version 5.03 of HTML::AsSubs, released September 22, 2012 as part of HTML-Tree.

SYNOPSIS

```
use HTML::AsSubs;
$h = body(
  h1("This is the heading"),
  p("This is the first paragraph which contains a ",
    a({href=>'link.html'}, "link"),
    " and an ",
    img({src=>'img.gif', alt=>'image'}),
    ". "
  ),
);
print $h->as_HTML;
```

DESCRIPTION

This module exports functions that can be used to construct various HTML elements. The functions are named after the tags of the corresponding HTML element and are all written in lower case. If the first argument is a hash reference then it will be used to initialize the attributes of this element. The remaining arguments are regarded as content.

For a similar idea (i.e., it's another case where the syntax tree of the Perl source mirrors the syntax tree of the HTML produced), see HTML::Element's `new_from_lo1` method.

For what I now think is a cleaner implementation of this same idea, see the excellent module `XML::Generator` which is what I suggest for actual real-life use. (I suggest this over [HTML::AsSubs](#) and over `CGI.pm`'s HTML-making functions.)

ACKNOWLEDGEMENT

This module was inspired by the following message:

```
Date: Tue, 4 Oct 1994 16:11:30 +0100
Subject: Wow! I have a large lightbulb above my head!
```

Take a moment to consider these lines:

```
%OVERLOAD=( '' => sub { join("", @{$_[0]}) } );

sub html { my($type)=shift; bless ["<$type>", @_, "</$type>"]; }
```

:-) I *love* Perl 5! Thankyou Larry and Ilya.

Regards,
Tim Bunce.

p.s. If you didn't get it, think about recursive data types: `html(html())`
p.p.s. I'll turn this into a much more practical example in a day or two.
p.p.p.s. It's a pity that overloads are not inherited. Is this a bug?

BUGS

The exported `link()` function overrides the builtin `link()` function. The exported `tr()` function must be called using `&tr(...)` syntax because it clashes with the builtin `tr/./././` operator.

SEE ALSO

[HTML::Element](#), [XML::Generator](#)

`html head title base link meta isindex nextid script style body h1 h2 h3 h4 h5 h6 p pre div blockquote a img br hr ol ul dir menu li dl dt dd dfn cite code em kbd samp strong var address span b i u tt center font big small strike sub sup table tr td th caption form input select option textarea object applet param map area frame frameset noframe`

A bunch of methods for creating tags.

Private Functions

`_elem()`

The `_elem()` function is wrapped by all the `html 'tag'` functions. It takes a tag-name, optional hashref of attributes and a list of content as parameters.

AUTHOR

Current maintainers:

- Christopher J. Madsen <perl AT cjmweb.net>
- Jeff Fearn <jfearn AT cpan.org>

Original HTML-Tree author:

- Gisle Aas

Former maintainers:

- Sean M. Burke
- Andy Lester
- Pete Krawczyk <petek AT cpan.org>

You can follow or contribute to HTML-Tree's development at <http://github.com/madsen/HTML-Tree>.

COPYRIGHT AND LICENSE

Copyright 1995-1998 Gisle Aas, 1999-2004 Sean M. Burke, 2005 Andy Lester, 2006 Pete Krawczyk, 2010 Jeff Fearn, 2012 Christopher J. Madsen.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

The programs in this library are distributed in the hope that they will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.