

## NAME

Pod::Simple::PullParserStartToken -- start-tokens from Pod::Simple::PullParser

## SYNOPSIS

(See *Pod::Simple::PullParser*)

## DESCRIPTION

When you do `$parser->get_token` on a *Pod::Simple::PullParser* object, you might get an object of this class.

This is a subclass of *Pod::Simple::PullParserToken* and inherits all its methods, and adds these methods:

`$token->tagname`

This returns the tagname for this start-token object. For example, parsing a `"=head1 ..."` line will give you a start-token with the tagname of "head1", token(s) for its content, and then an end-token with the tagname of "head1".

`$token->tagname(somestring)`

This changes the tagname for this start-token object. You probably won't need to do this.

`$token->tag(...)`

A shortcut for `$token->tagname(...)`

`$token->is_tag(somestring)` or `$token->is_tagname(somestring)`

These are shortcuts for `$token->tag() eq somestring`

`$token->attr(attrname)`

This returns the value of the *attrname* attribute for this start-token object, or undef.

For example, parsing a `L<Foo/"Bar">` link will produce a start-token with a "to" attribute with the value "Foo", a "type" attribute with the value "pod", and a "section" attribute with the value "Bar".

`$token->attr(attrname, newvalue)`

This sets the *attrname* attribute for this start-token object to *newvalue*. You probably won't need to do this.

`$token->attr_hash`

This returns the hashref that is the attribute set for this start-token. This is useful if (for example) you want to ask what all the attributes are -- you can just do `keys %{$token->attr_hash}`

You're unlikely to ever need to construct an object of this class for yourself, but if you want to, call `Pod::Simple::PullParserStartToken->new( tagname, attrhash )`

## SEE ALSO

*Pod::Simple::PullParserToken*, *Pod::Simple*, *Pod::Simple::Subclassing*

## COPYRIGHT AND DISCLAIMERS

Copyright (c) 2002 Sean M. Burke. All rights reserved.

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

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

**AUTHOR**Sean M. Burke [sburke@cpan.org](mailto:sburke@cpan.org)