Make Comments
Stand Out

David Golden
Senior Engineer, MongoDB

YAPC::NA 2015

Inspired by

Your Syntax Highlighter is Wrong

by James Fisher

"A comment may be used to amplify the importance of something that may otherwise seem inconsequential"

— *Clean* Code by Robert Martin

"A comment may be used to
amplify the importance
of something that may
otherwise seem inconsequential"

— *Clean* Code by Robert Martin

"A comment may be used to amplify the importance of something that may otherwise **seem inconsequential**"

— *Clean* Code by Robert Martin

Most syntax highlighters get this wrong

```
    my $path = shift;
    Carp::croak("Path::Tiny paths require defined, positive-length parts")
      unless 1 + @_ == grep { defined && length } $path, @_;

    # non-temp Path::Tiny objects are effectively immutable and can be reused
    if ( !@_ && ref($path) eq __PACKAGE__ && !$path->[TEMP] ) {
        return $path;
    }

    # stringify initial path
    $path = "$path";

    # expand relative volume paths on windows; put trailing slash on UNC root
    if ( IS_WIN32() ) {
        $path = _win32_vol( $path, $1 ) if $path =~ m{^($DRV_VOL)(?:$NOTSLASH|$)};
        $path .= "/" if $path =~ m{^$UNC_VOL$};
    }

    # concatenate more arguments (stringifies any objects, too)
    if (@_) {
        $path .= ( _is_root($path) ? "" : "/" ) . join( "/", @_ );
    }

    # canonicalize paths
    my $cpath = $path = File::Spec->canonpath($path); # ugh, but probably worth it
    $path =~ tr[\\][/] if IS_WIN32();                 # unix convention enforced
    $path .= "/" if IS_WIN32() && $path =~ m{^$UNC_VOL$}; # canonpath strips it

    # hack to make splitpath give us a basename; root paths must always have
    # a trailing slash, but other paths must not
    if ( _is_root($path) ) {
        $path =~ s{/?$}{/};
    }
    else {
        $path =~ s{/$}{};
```

```
my $path = shift;
Carp::croak("Path::Tiny paths require defined, positive-length parts")
  unless 1 + @_ == grep { defined && length } $path, @_;

# non-temp Path::Tiny objects are effectively immutable and can be reused
if ( !@_ && ref($path) eq __PACKAGE__ && !$path->[TEMP] ) {
    return $path;
}

# stringify initial path
$path = "$path";

# expand relative volume paths on windows; put trailing slash on UNC root
if ( IS_WIN32() ) {
    $path = _win32_vol( $path, $1 ) if $path =~ m{^($DRV_VOL)(?:$NOTSLASH|$)};
    $path .= "/" if $path =~ m{^$UNC_VOL$};
}

# concatenate more arguments (stringifies any objects, too)
if (@_) {
    $path .= ( _is_root($path) ? "" : "/" ) . join( "/", @_ );
}

# canonicalize paths
my $cpath = $path = File::Spec->canonpath($path); # ugh, but probably worth it
$path =~ tr[\\][/] if IS_WIN32();                 # unix convention enforced
$path .= "/" if IS_WIN32() && $path =~ m{^$UNC_VOL$}; # canonpath strips it

# hack to make splitpath give us a basename; root paths must always have
# a trailing slash, but other paths must not
if ( _is_root($path) ) {
    $path =~ s{/?$}{/};
}
else {
    $path =~ s{/$}{};
}
```

```
    my $path = shift;
    Carp::croak("Path::Tiny paths require defined, positive-length parts")
      unless 1 + @_ == grep { defined && length } $path, @_;

    # non-temp Path::Tiny objects are effectively immutable and can be reused
    if ( !@_ && ref($path) eq __PACKAGE__ && !$path->[TEMP] ) {
        return $path;
    }

    # stringify initial path
    $path = "$path";

    # expand relative volume paths on windows; put trailing slash on UNC root
    if ( IS_WIN32() ) {
        $path = _win32_vol( $path, $1 ) if $path =~ m{^($DRV_VOL)(?:$NOTSLASH|$)};
        $path .= "/" if $path =~ m{^$UNC_VOL$};
    }

    # concatenate more arguments (stringifies any objects, too)
    if (@_) {
        $path .= ( _is_root($path) ? "" : "/" ) . join( "/", @_ );
    }

    # canonicalize paths
    my $cpath = $path = File::Spec->canonpath($path); # ugh, but probably worth it
    $path =~ tr[\\][/] if IS_WIN32();                 # unix convention enforced
    $path .= "/" if IS_WIN32() && $path =~ m{^$UNC_VOL$}; # canonpath strips it

    # hack to make splitpath give us a basename; root paths must always have
    # a trailing slash, but other paths must not
    if ( _is_root($path) ) {
        $path =~ s{/?$}{/};
    }
    else {
        $path =~ s{/$}{};
    }
```

Excess comments

stand out, too

```perl
#!/usr/bin/env perl
use v5.10;
use strict;
use warnings;

# first define some variables from @ARGV
my ( $x, $y ) = @ARGV;

# Check that they are numbers and throw an exception if they
# aren't both numbers
die "Args must be numbers" if 2 != grep { /^\d+$/ } $x, $y;

# subtract the bigger number from the smaller one to manually
# generate the absolute value
my $abs = $x > $y ? $x - $y : $y - $x;

# print out the result
say "Absolute value is $abs";
```

```
    my $path = shift;
    Carp::croak("Path::Tiny paths require defined, positive-length parts")
      unless 1 + @_ == grep { defined && length } $path, @_;

    # non-temp Path::Tiny objects are effectively immutable and can be reused
    if ( !@_ && ref($path) eq __PACKAGE__ && !$path->[TEMP] ) {
        return $path;
    }

    # stringify initial path
    $path = "$path";

    # expand relative volume paths on windows; put trailing slash on UNC root
    if ( IS_WIN32() ) {
        $path = _win32_vol( $path, $1 ) if $path =~ m{^($DRV_VOL)(?:$NOTSLASH|$)};
        $path .= "/" if $path =~ m{^$UNC_VOL$};
    }

    # concatenate more arguments (stringifies any objects, too)
    if (@_) {
        $path .= ( _is_root($path) ? "" : "/" ) . join( "/", @_ );
    }

    # canonicalize paths
    my $cpath = $path = File::Spec->canonpath($path); # ugh, but probably worth it
    $path =~ tr[\\][/] if IS_WIN32();                 # unix convention enforced
    $path .= "/" if IS_WIN32() && $path =~ m{^$UNC_VOL$}; # canonpath strips it

    # hack to make splitpath give us a basename; root paths must always have
    # a trailing slash, but other paths must not
    if ( _is_root($path) ) {
        $path =~ s{/?$}{/};
    }
    else {
        $path =~ s{/$}{};
    }
```

```
sub path {
    my $path = shift;
    Carp::croak("Path::Tiny paths require defined, positive-length parts")
      unless 1 + @_ == grep { defined && length } $path, @_;

    # non-temp Path::Tiny objects are effectively immutable and can be reused
    if ( !@_ && ref($path) eq __PACKAGE__ && !$path->[TEMP] ) {
        return $path;
    }

    # stringify objects
    $path = "$path";

    # expand relative volume paths on windows; put trailing slash on UNC root
    if ( IS_WIN32() ) {
        $path = _win32_vol( $path, $1 ) if $path =~ m{^($DRV_VOL)(?:$NOTSLASH|$)};
        $path .= "/" if $path =~ m{^$UNC_VOL$};
    }

    # concatenations stringifies objects, too
    if (@_) {
        $path .= ( _is_root($path) ? "" : "/" ) . join( "/", @_ );
    }

    # canonicalize, but with unix slashes and put back trailing volume slash
    my $cpath = $path = File::Spec->canonpath($path);
    $path =~ tr[\\][/] if IS_WIN32();
    $path .= "/" if IS_WIN32() && $path =~ m{^$UNC_VOL$};

    # root paths must always have a trailing slash, but other paths must not
    if ( _is_root($path) ) {
        $path =~ s{/?$}{/};
    }
    else {
        $path =~ s{/$}{};
    }
```

For better comments, make them stand out!