

## Re: transforming german characters

**Source:** <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2004-08/0475.html>

---

**From:** steve\_f (me\_at\_example.com)

**Date:** 08/07/04

Date: Fri, 06 Aug 2004 20:46:30 -0400

Thank you John, this is really useful. Just to start, I must always remind myself if I am doing something too many times to generalize.

>John W. Krahn wrote:

[ snip – my statement of problem ]

>>

>> *I wrote the follow which works well, but looks*

>> *pretty bad I think.*

>

>*It doesn't look too bad, I've seen worse. :-)*

>

I was able to brute force my way through it ;-)

>

>> *so again this is a style question...*

>> *can anyone suggest a cleaner approach? TIA*

>

>*The usual idiom is to use a hash for the search and replace tables.*

>

yes, I see and it is very good...changes the whole approach

>

>> *sub transform\_characters {*

>> *my @input = @\_;*

>> *my @output;*

>> *for my \$string ( @input ) {*

>> *push @output, \$string;*

>> *if ( \$string =~ /\xDF/ ) {*

>> *\$string =~ s/\xDF/ss/g;*

>

>*Using a match followed by a substitution is a usual beginner mistake.*

>*You only need the substitution.*

>

> *if ( \$string =~ s/\xDF/ss/g ) {*

>

ahh...ok...that's good to learn

[ snip code ]

```
>
> Using a hash you could write that as:
>
> my %set1 = (
>   "\xDF" => 'ss',
> );
> # Use a character class because all keys are single characters
> # If keys are multiple characters use alternation instead
```

can you explain this a bit further? I'm not quite sure what you mean by alternation, but I really only looked up the escaped values for this particular problem.

```
> my $key1 = '[' . join( " ", keys %set1 ) . ']';
```

also here I start to get really lost...ok, you are loading into a scalar the keys as one long string...joining them with no space between... with two brackets so

```
$key1 = [\xDF]
$key2 = [\xC4\xD6\xDC\xE4\xF6\xFC]
correct?
```

I see you use it down below in this substitution but it is a bit hard for me to understand:

```
if ( $string =~ s/($key1)/$set1{$1}/og )
```

well, if you have the time please give me a bit more clarification on this because I haven't seen it before.

```
>
> my %set2 = (
>   "\xC4" => 'Ae',
>   "\xD6" => 'Oe',
>   "\xDC" => 'Ue',
>   "\xE4" => 'ae',
>   "\xF6" => 'oe',
>   "\xFC" => 'ue',
> );
> my $key2 = '[' . join( " ", keys %set2 ) . ']';
>
> sub transform_characters {
>   my @input = @_;
>   my @output;
>   for my $string ( @input ) {
>     push @output, $string;
```

comp.lang.perl.misc: Re: transforming german characters

```
> if( $string =~ s/($key1)/$set1{$1}/og ) {  
> push @output, $string;  
> if( $string =~ s/($key2)/$set2{$1}/og ) {  
> push @output, $string;  
> }  
> next;  
> }  
> if( $string =~ s/($key2)/$set2{$1}/og ) {  
> push @output, $string;  
> }  
> }  
> return @output;  
> }  
>  
>  
>  
>John
```

Thanks again John.

Steve