Parsing XML / HTML

Elizabeth Mattijsen

Eerste Nederlandse Perl Workshop

5 maart 2004

Wat wil je bereiken?

- Wil je gegevens uit HTML halen om er verder iets mee te doen?
- Wil je de HTML netjes maken?
- Wil je er XML van maken?
- Waarschijnlijk nog wel iets anders... ;-)
- EIMDEMOHTD (TIMTOWTDI) !

Tekst uit HTML halen

- Tekst, bijvoorbeeld voor een zoekmachine:

```
lynx -dump URL >filename
```

* Quick 'n' Dirty oplossing
* Niet direct Perl nodig
* Eventueel in combinatie met open()

```
open( my $handle,"lynx -dump URL |" );
local $/; # slurp mode
my $text = <$handle>;
```

Links uit HTML halen

- Daar zijn een heleboel modules voor op CPAN!
- HTML::LinkExtor
- HTML::SimpleLinkExtor
- HTML::LinkExtractor
- en nog veel meer algemene modules die hiervoor gebruikt / misbruikt kunnen worden!

HTML::LinkExtractor

```
use HTML::LinkExtractor;
my $LX = HTML::LinkExtractor->new;
$LX->parse( 'programma.html' );

my @link = map {$_->{'href'}}
           grep {$_->{'tag'} eq 'a'}
           @{$LX->links};

print " $_\n" foreach @link;
__END__
 /
 /aanmelden.html
 /overnachten.html
 /contact.html
 /route.html
 /logo/
 http://www.yapc.org
```

En nu de images!

```
use HTML::LinkExtractor;
my $LX = HTML::LinkExtractor->new;
$LX->parse( 'programma.html' );

my @link = map {$_->{'src'}}
           grep {$_->{'tag'} eq 'img'}
           @{$LX->links};

print " $_\n" foreach @link;
__END__
 /perl.jpg
 /logo.png
```

EIMDEMOHTD!

```
use HTML::TokeParser;

my $p = HTML::TokeParser->new( 'programma.html' );

while (my $token = $p->get_tag( 'a' )){
    next unless my $url = $token->[1]->{ 'href' };
    print " $url\n";
}
__END__
 /
 /aanmelden.html
 /overnachten.html
 /contact.html
 /route.html
 /logo/
 http://www.yapc.org
```

Teksten met Perl modules

- HTML::TokeParser is ook algemener te gebruiken:

```
use HTML::TokeParser;

my $p =
 HTML::TokeParser->new( 'programma.html' );

while (my $token = $p->get_tag) {
    next unless my $text = $p->get_text;
    next unless $text =~ m#\w#;
    print " $text";
}
__END__
 Perl Workshop 2004
```

de XML aanpak!

- XML::LibXML is een snelle XML parser die gebaseerd is op de GNOME "libxml" library.

```
use XML::LibXML;
my $libxml = XML::LibXML->new;
$libxml->recover( 1 );

my $dom = $libxml->parse_html_file( 'programma.html' );

foreach ($dom->findnodes( '//a' )) {
    print " ",$_->getAttribute( 'href' ),"\n";
}
__END__
 /
 /aanmelden.html
 /overnachten.html
 /contact.html
 /route.html
 /logo/
 http://www.yapc.org
```

Alle tekst met XML

```
use XML::LibXML;

my $libxml = XML::LibXML->new;
$libxml->recover( 1 );

my $dom =
 $libxml->parse_html_file('programma.html');

print $dom->textContent;
__END__
Perl Workshop 2004; Programma
```

XML van HTML maken

```
use XML::LibXML;

my $libxml = XML::LibXML->new;
$libxml->recover( 1 );

my $dom = $libxml->parse_html_file('programma.html');

print $dom->toString;
__END__
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN" "http://www.w3.org/TR/REC-html40/
loose.dtd">
<html>
 <head>
  <title>Perl Workshop 2004; Programma</title>
```

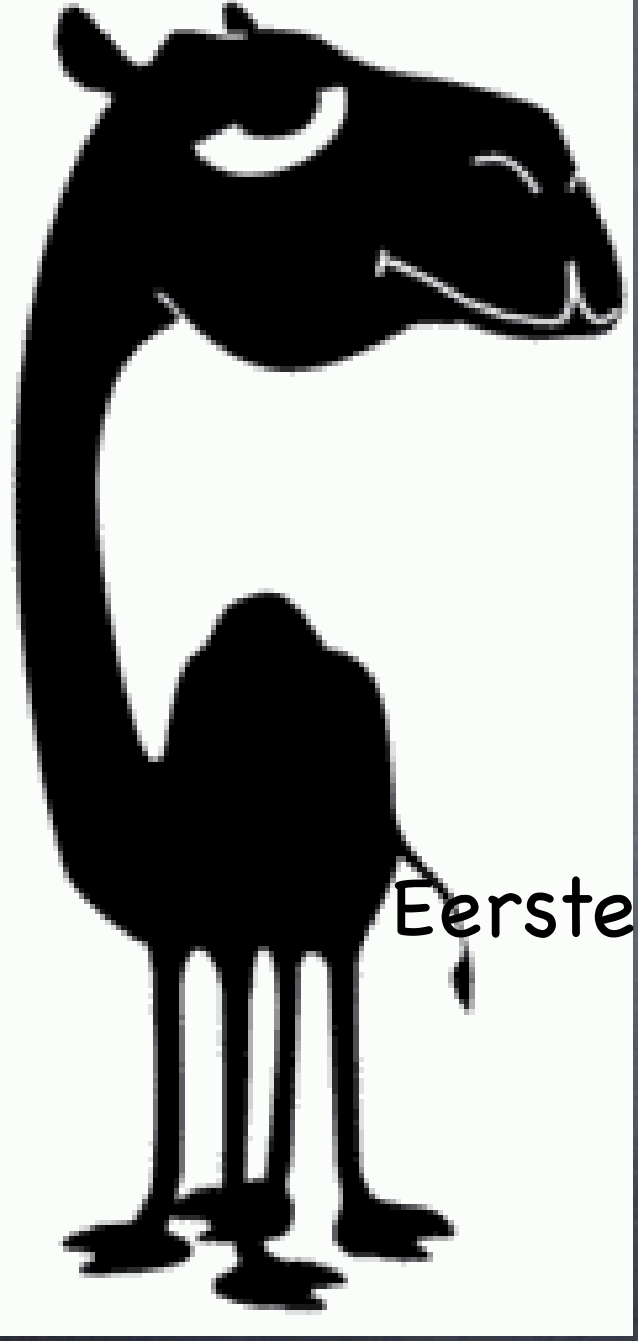
Een kleine Benchmark

```
use Benchmark qw(:all);
use HTML::LinkExtractor;
use HTML::TokeParser;
use XML::LibXML;

my $libxml = XML::LibXML->new;
$libxml->recover( 1 );

cmpthese( -2,{
 tokeparser => sub {
    my $p = HTML::TokeParser->new( 'programma.html' );
    my @url;
    while (my $token = $p->get_tag( 'a' )) {
        push @url, $token->[1]->{ 'href' };
    } },
 libxml => sub {
    my $dom = $libxml->parse_html_file( 'programma.html' );
    my @url = map {$_->getAttribute( 'href' )} $dom->findnodes('//a');
    },
 linkextractor => sub {
    $LX->parse( 'programma.html' );
    my @url = map {$_->{'src'}} grep {$_->{'tag'} eq 'img'} @{$LX->links};
 },
} );
__END__
```

	Rate	linkextractor	tokeparser	libxml
linkextractor	5.85/s	--	-88%	-95%
tokeparser	49.5/s	746%	--	-57%
libxml	116/s	1878%	134%	--



Vragen?

Elizabeth Mattijsen

Eerste Nederlandse Perl Workshop

5 maart 2004

Dank voor uw aandacht en medewerking!