

**XML::Liberal**

**Tatsuhiko Miyagawa**  
**miyagawa@gmail.com**

**Six Apart, Ltd. / Shibuya Perl Mongers**  
**YAPC::NA 2006 Chicago**

**Quiz:**  
**What's wrong  
with following XMLs?**

```
<?xml version="1.0"?>  
<heroes>Larry & Damian</heroes>
```

```
<?xml version="1.0"?>  
<heroes>Larry &amp; Damian</heroes>
```

```
<?xml version="1.0"?>  
<shout>  
I &hearts; Perl &amp; Ruby!  
</shout>
```

```
<?xml version="1.0"?>  
<shout>  
I &#x2665; Perl &amp; Ruby!  
</shout>
```

```
<?xml version="1.0"?>
<html>
<body>
<a href=foo.html>Foo foo!</a>
</body>
</html>
```

```
<?xml version="1.0"?>
<html>
<body>
<a href="foo.html">Foo foo!</a>
</body>
</html>
```

```
<?xml version="1.0"?>
<html>
<body>
<a href="/search?q=YAPC+NA&hl=en">Search for YAPC</a>
</body>
</html>
```

```
<?xml version="1.0"?>
<html>
<body>
<a href="/search?q=YAPC+NA&hl=en">Search for
YAPC</a>
</body>
</html>
```

```
<?xml version="1.0"?>
<rss version="2.0">
<channel>
<item>
  <title>YAPC is love</title>
  <content:encoded>Here I am!</content:encoded>
</item>
</channel>
</rss>
```

```
<?xml version="1.0"?>
<rss version="2.0
  xmlns:content="http://purl.org/rss/1.0/modules/content/">
<channel>
<item>
  <title>YAPC is love</title>
  <content:encoded>Here I am!</content:encoded>
</item>
</channel>
</rss>
```

```
<?xml version="1.0" encoding="utf-8"?>
<search>
<result>
  <excerpt>YAPC 最高! ...</excerpt>
</result>
</search>
```

```
<?xml version="1.0" encoding="utf-8"?>
<search>
<result>
  <excerpt>YAPC 最高! □...</excerpt>
</result>
</search>
```

**If you're  
a feed publisher,  
Make sure  
your feed is valid.**

**If you're a feed consumer,  
Not being able  
to parse broken feeds  
might make your users unhappy.**

# As we saw in the Quiz

**Detecting XML errors  
Is easy and simple  
(sometimes).**

**Browsers deal  
With it.**

**Human can  
parse it.**

# Why not XML parsers doing the same thing?

# **XML::Liberal**

## **Does it for you.**

# Demo

# Usage #1

## If you use XML::LibXML

```
use strict;  
use XML::LibXML;  
  
my $parser = XML::LibXML->new;  
my $doc = $parser->parse_string($xml);
```

```
use strict;  
use XML::Liberal;  
  
my $parser = XML::Liberal->new("LibXML");  
my $doc = $parser->parse_string($xml);
```

**Note: There's NO  
performance drawback  
If the data is valid**

`$parser->max_fallback(3)`

# Usage #2

**If you use XML::LibXML  
Without control to  
change the code**

```
use strict;
use XML::Liberal;

XML::Liberal->globally_override('LibXML');

# later on ...
my $parser = XML::LibXML->new;
```

# Usage #3

## If you don't use LibXML

```
use strict;  
use XML::FooParser;  
  
$doc = parse_xml($xml);
```

```
use strict;
use XML::FooParser;
use XML::Liberal;

eval { $doc = parse_xml($xml) };
if ($?) {
    my $p = XML::Liberal->new('LibXML');
    my $d = $p->parse_string($xml);
    $xml = $d->toString(1);
    $doc = parse_xml($xml);
}
```

**XML::Liberal**  
**Came out of Plagger**  
**<http://plagger.org/>**

**cpan XML::Liberal**

# Thank you