

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Ben Ramsey</title>
    <link>http://benramsey.com</link>
    <description>PHP and Other Techno-babble</description>
    <pubDate>Sun, 28 May 2006 16:58:04 +0000</pubDate>
    <item>
      <title>Praesent et massa</title>
      <link>http://benramsey.com/archives/praesent-et-massa/</link>
      <pubDate>Sat, 27 May 2006 12:39:23 +0000</pubDate>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
        felis. Sed quis ligula. Praesent et massa. Nunc varius fermentum leo.
      </description>
      <enclosure url="http://benramsey.com/podcast1.mp3" length="12216320"
        type="audio/mpeg" />
    </item>
    <item>
      <title>Lorem ipsum dolor sit amet</title>
      <link>http://benramsey.com/archives/lorem-ipsum/</link>
      <pubDate>Sun, 28 May 2006 16:24:36 +0000</pubDate>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
        felis. Sed quis ligula. Praesent et massa. Nunc varius fermentum leo.
      </description>
      <enclosure url="http://benramsey.com/podcast2.mp3" length="12216320"
        type="audio/mpeg" />
    </item>
  </channel>
</rss>
```

Mastering XML with SimpleXML

Introduction: Ben Ramsey

— [PHP Security Consortium, Zend Certified Engineer

— [Author of articles in PHP Magazine & php | architect

— [Currently writing Professional PHP Security (Wrox) and a mystery book (php | architect nanobooks)

— [Contributed to the PHP Manual SimpleXML docs

— [Engineer for Art & Logic

What is RSS?

— [Really Simple Syndication

— [XML-based format for describing syndicated content

— [Intended to distribute regularly updated content

— [Owned by Harvard; not an open standard

— [Most widely adopted syndication format

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Ben Ramsey</title>
    <link>http://benramsey.com</link>
    <description>PHP and Other Techno-babble</description>
    <pubDate>Sun, 28 May 2006 16:58:04 +0000</pubDate>
    <item>
      <title>Praesent et massa</title>
      <link>http://benramsey.com/archives/praesent-et-massa/</link>
      <pubDate>Sat, 27 May 2006 12:39:23 +0000</pubDate>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
        felis. Sed quis ligula. Praesent et massa. Nunc varius fermentum leo.
      </description>
      <enclosure url="http://benramsey.com/podcast1.mp3" length="12216320"
        type="audio/mpeg" />
    </item>
    <item>
      <title>Lorem ipsum dolor sit amet</title>
      <link>http://benramsey.com/archives/lorem-ipsum/</link>
      <pubDate>Sun, 28 May 2006 16:24:36 +0000</pubDate>
      <description>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
        felis. Sed quis ligula. Praesent et massa. Nunc varius fermentum leo.
      </description>
      <enclosure url="http://benramsey.com/podcast2.mp3" length="12216320"
        type="audio/mpeg" />
    </item>
  </channel>
</rss>
```

Parsing XML with PHP

— [Before PHP 5, there was DOMXML

— [Yuck!

— [PHP 5 introduces SimpleXML

SimpleXML API

— [`__construct()`

— [`addAttribute()`

— [`addChild()`

— [`asXML()`

— [`attributes()`

SimpleXML API

— [children()

— [getDocNamespaces()

— [getName()

— [getNamespaces()

— [xpath()

Getting individual nodes

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

echo $rss->channel->title . "\n";
echo $rss->channel->link . "\n";
echo $rss->channel->item[0]->title;
?>
```

Looping through nodes

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

foreach ($rss->channel->item as $item)
{
    echo $item->title . " <" . $item->link . ">\n";
}
?>
```

Finding nodes with XPath

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

$result = $rss->xpath('//item/title');

foreach ($result as $title)
{
    echo $title . "\n";
}
?>
```

Getting children and names

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

foreach ($rss->channel->children() as $child)
{
    echo $child->getName() . "\n";
}
?>
```

Handling attributes

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

foreach ($rss->channel->item as $item)
{
    foreach ($item->enclosure->attributes()
        as $key => $value)
    {
        echo "{$key} => {$value}\n";
    }
}
?>
```

Handling attributes

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

foreach ($rss->channel->item as $item)
{
    echo "url => ".$item->enclosure['url']."\n";
    echo "length => ".$item->enclosure['length']."\n";
    echo "type => ".$item->enclosure['type']."\n\n";
}
?>
```

Adding new children

```
<?php
$rss = new SimpleXMLElement('feed.rss.xml',
    NULL, TRUE);

$item = $rss->channel->addChild('item');

$item->addChild('title', 'My New Post');
$item->addChild('link',
    'http://example.org/archives/my-new-post/');
$item->addChild('pubDate', date(DATE_RFC2822));
$item->addChild('description',
    'This is what my post is about.');
```

?>

Adding attributes

```
<?php
/* Continued from previous example */

$enclosure = $item->addChild('enclosure');

$enclosure->addAttribute('url',
    'http://example.org/podcast3.mpg');
$enclosure->addAttribute('length', '12216320');
$enclosure->addAttribute('type', 'audio/mpeg');

header('Content-type: text/xml');
echo $rss->asXML();
?>
```

SimpleXML is that simple

— [It's easy

— [It's simple

— [It's a piece of cake

— [Why use DOM again?

For more information:

[<http://www.php.net/SimpleXML>

[<http://www.rssboard.org/rss-specification>

[<http://www.w3.org/TR/xml11/>

[Questions?

Project Questions?

Database connection

— [Zend_Db

— [Documentation at:

<http://framework.zend.com/manual/en/zend.db.html>

— [Several different database drivers, including PDO for MySQL, PostgreSQL, SQLite, and MSSQL, as well as MySQLi and Oracle

What kind of application?

— [Meeting check-in application (for a start)

— [Database design?

— [Application design?

Database design

— [What kind of data do we want to collect?

Database design

People

Id

FirstName

LastName

Email

Timestamp

Application design

— [For now, let's assume that the application will list all checked-in members at:

<http://example.org/members/list>

— [We need a `MembersController` class

— [This class will need a `listAction()` method

Connecting to a database

```
<?php
/* in the bootstrap index.php file */
/* Set up Database object */
$params = array (
    'host'          => '127.0.0.1',
    'username'     => 'dbuser',
    'password'     => 'dbpass',
    'dbname'       => 'atlantaphp'
);
$db = Zend_Db::factory('pdoMySQL', $params);
Zend::register('db', $db);
?>
```

Fetching from the database

```
<?php
/* in your action method */
$db = Zend::registry('db');

$select = $db->select();
$select->from('People', '*')
    ->order('LastName, FirstName ASC');

$result = $db->fetchAll($select);
?>
```

Questions?