Useful PHP classes and collections

Dates and Intervals, DOMDocument, Manipulating XML, Strings in PHP

Mario Peshev

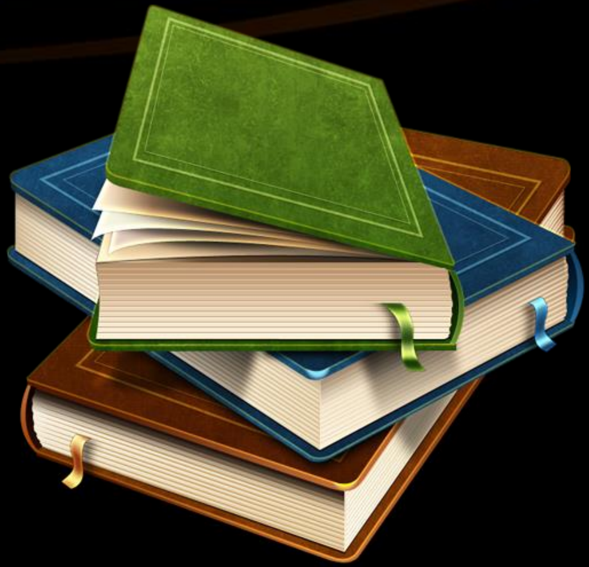
Technical Trainer

http://peshev.net

Software University

http://softuni.bg

Table of Contents



DateTime Class

DateTime Class

- In-build from PHP 5.2.x ->
- Used to manipulate Dates and Times
- Can be used to obtain current Date/Time
- Functions depend on the locale settings of the server
- No need of installation

php.ini configuration

- You can configure some settings in php.ini in order to change the behavior of some DateTime functions

Name	Description
date.timezone	The default timezone (used by all date/time functions)
date.default_latitude	The default latitude (used by date_sunrise() and date_sunset())
date.default_longitude	The default longitude (used by date_sunrise() and date_sunset())
date.sunrise_zenith	The default sunrise zenith (used by date_sunrise() and date_sunset())
date.sunset_zenith	The default sunset zenith (used by date_sunrise() and date_sunset())

Using DateTime

- Initialized like an object

```
$date = new DateTime('2014-07-15');
```

- Can take various constructors

http://php.net/manual/en/datetime.formats.php

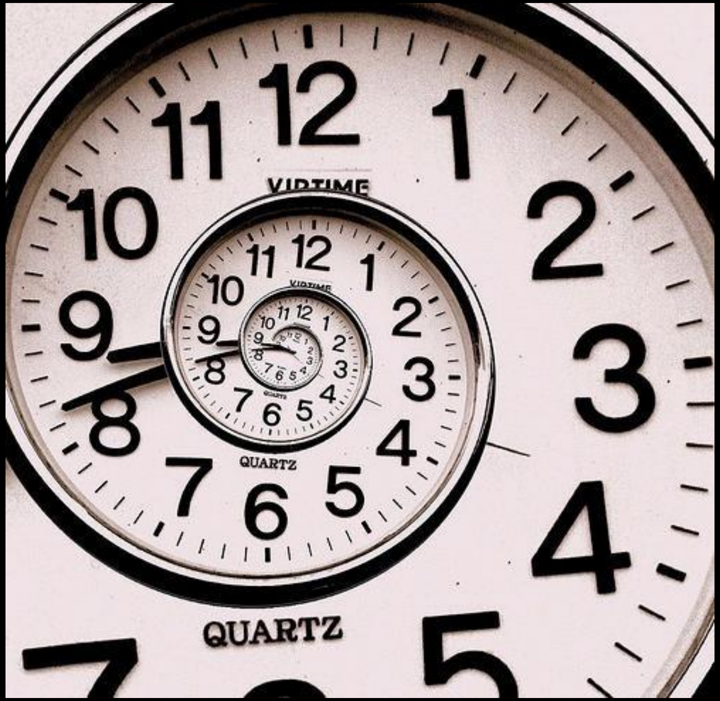
DateInterval Class

DateInterval Class

- A date interval stores either a fixed amount of time (in years, months, days, hours etc) or a relative time string in the format that DateTime's constructor supports
- The format starts with the letter *P*, for "period." Each duration period is represented by an integer value followed by a period designator. If the duration contains time elements, that portion of the specification is preceded by the letter *T*.

```
//2 Years, 4 Days, 6 Hours and 8 Minutes
$interval = new DateInterval('P2Y4DT6H8M');
```

DateTime Methods



DateTime Methods

SOFTWARE UNIVERSITY FOUNDATION

- format() – formats the DateTime
- add() – adds a period of time to the object

```
$date = new DateTime('2014-08-16');
$date->add(new DateInterval('P10D')); //adds 10 days
echo $date->format('Y-m-d') . "\n";
```

- sub() – subtracts a period of time from the object
- setdate() – sets a date new value for the object

DateTime Methods (2)

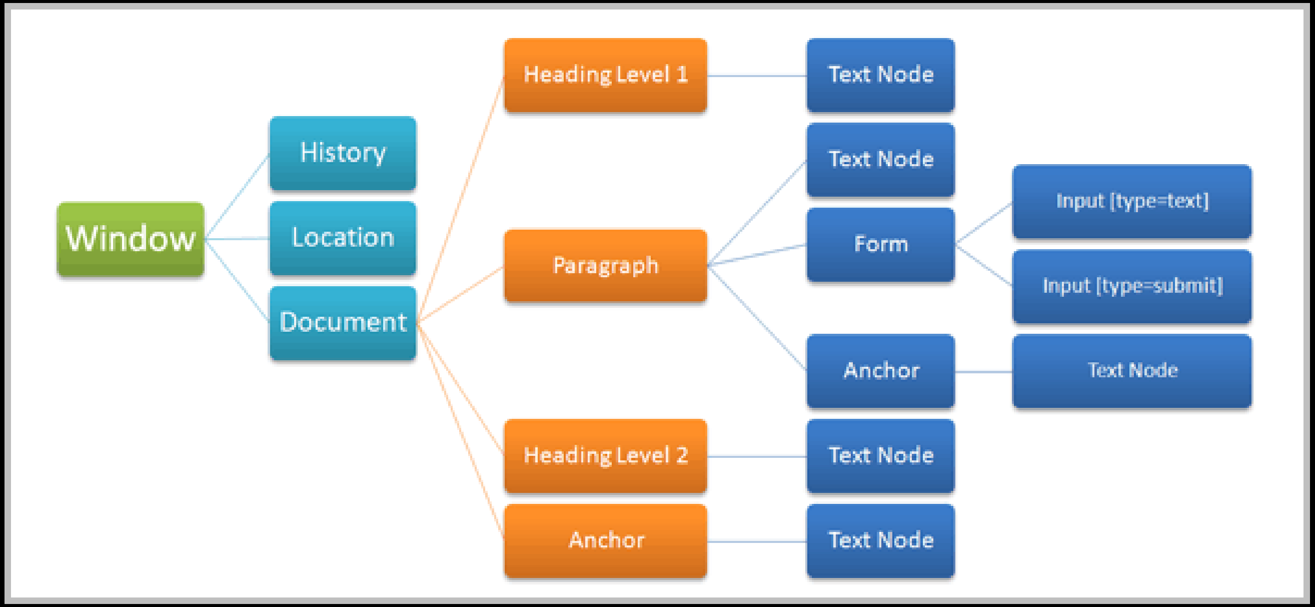
- settime() – sets a new time value for the object
- setTimezone () – sets a specific Timezone for the object

```
$date->setTimezone(new DateTimeZone('Pacific/Chatham'));
```

Dates and Intervals

Live demo

DOMDocument Class



What is DOM?

- The **Document Object Model** (**DOM**) is a cross-platform and language-independent *convention* for representing and interacting with objects in HTML, XHTML and XML documents
- Objects in the DOM tree may be addressed and manipulated by using methods on the objects
- The public interface of a DOM is specified in its application programming interface (API)

DOMDocument

- Setting an appropriate content type so the browser recognizes that we want to use xml:

```
header("content-type: application/xml; charset=UTF-8");
```

- Creating a DOMDocument Object:

```
$xml = new DOMDocument("1.0", "ISO-8859-15");
```

DOMDocument (2)

- Creating some elements:

```
$xml_album = $xml->createElement("Album");
$xml_track = $xml->createElement("Track", "The ninth symphony");
```

- Setting some attributes

```
$xml_track->setAttribute("length", "0:01:15");
$xml_track->setAttribute("bitrate", "64kb/s");
$xml_track->setAttribute("channels", "2");
```

- Creating another element to simulate sublevels:

```
$xml_note = $xml->createElement("Note", "The last symphony by  Beethoven");
```

DOMDocument (3)

- Appending the elements:

```
$xml_track->appendChild($xml_note);
$xml_album->appendChild($xml_track);
$xml->appendChild($xml_album);
```

- Parsing the XML:

```
print $xml->saveXML();
```

DOMDocument

Live demo

XMLReader Class

XMLReader

- The XMLReader extension is an XML Pull parser.
- The reader acts as a cursor going forward on the document stream and stopping at each node on the way
- It is important to note that internally, libxml uses the UTF-8 encoding and as such, the encoding of the retrieved contents will always be in UTF-8 encoding
- No need of installation

XMLReader (2)

- Initializing XMLReader and opening a file:

```
$reader = new XMLReader();
$reader->open('data.xml');
```

- After that we can read the next Node from the XML and get its attributes:

```
$xml->read();
echo $xml->name; //gets the node’s name
echo $xml->getAttribute(‘someAttribute'); //gets an attribute
$xml->next(); //gets the next node
```

XMLReader

Live demo

XMLWriter Class

XMLWriter

- Represents a writer that provides a non-cached, forward-only means of generating streams or files containing XML data
- This extension can be used in an object oriented style or a procedural one
- No need of installation

XMLWriter

- Initializing a writer and setting output:

```
$writer = new XMLWriter();
$writer->openURI('php://output');
$writer->startDocument('1.0','UTF-8');
```

- Creating elements and attributes:

```
$writer->startElement('color');
$writer->writeAttribute('color', 'A6A6A6');
$writer->endElement();
```

- Ending and flushing:

```
$writer->endDocument();
$writer->flush();
```

XMLWriter

Live demo

Strings in PHP

Strings

- Single quoted strings - display things almost completely "as is." Variables and most escape sequences will not be interpreted.
- The exception is that to display a literal single quote, you can escape it with a back slash \', and to display a back slash, you can escape it with another backslash \\.
- Single quoted strings are parsed.

Strings (2)

- `Double quoted strings` - display a host of escaped characters (including some regexes), and variables in the strings will be evaluated
- An important point here is that you can use curly braces to isolate the name of the variable you want evaluated
- For example let's say you have the variable `$type` and you what to `echo "The $types are"` That will look for the variable `$types`
- To get around this use `echo "The {$type}s are".` You can put the left brace before or after the dollar sign

Strings (3)

- Heredoc strings - string syntax works like double quoted strings
- It starts with <<<. After this operator, an identifier is provided, then a newline
- The string itself follows, and then the same identifier again to close the quotation
- You don't need to escape quotes in this syntax

Strings (4)

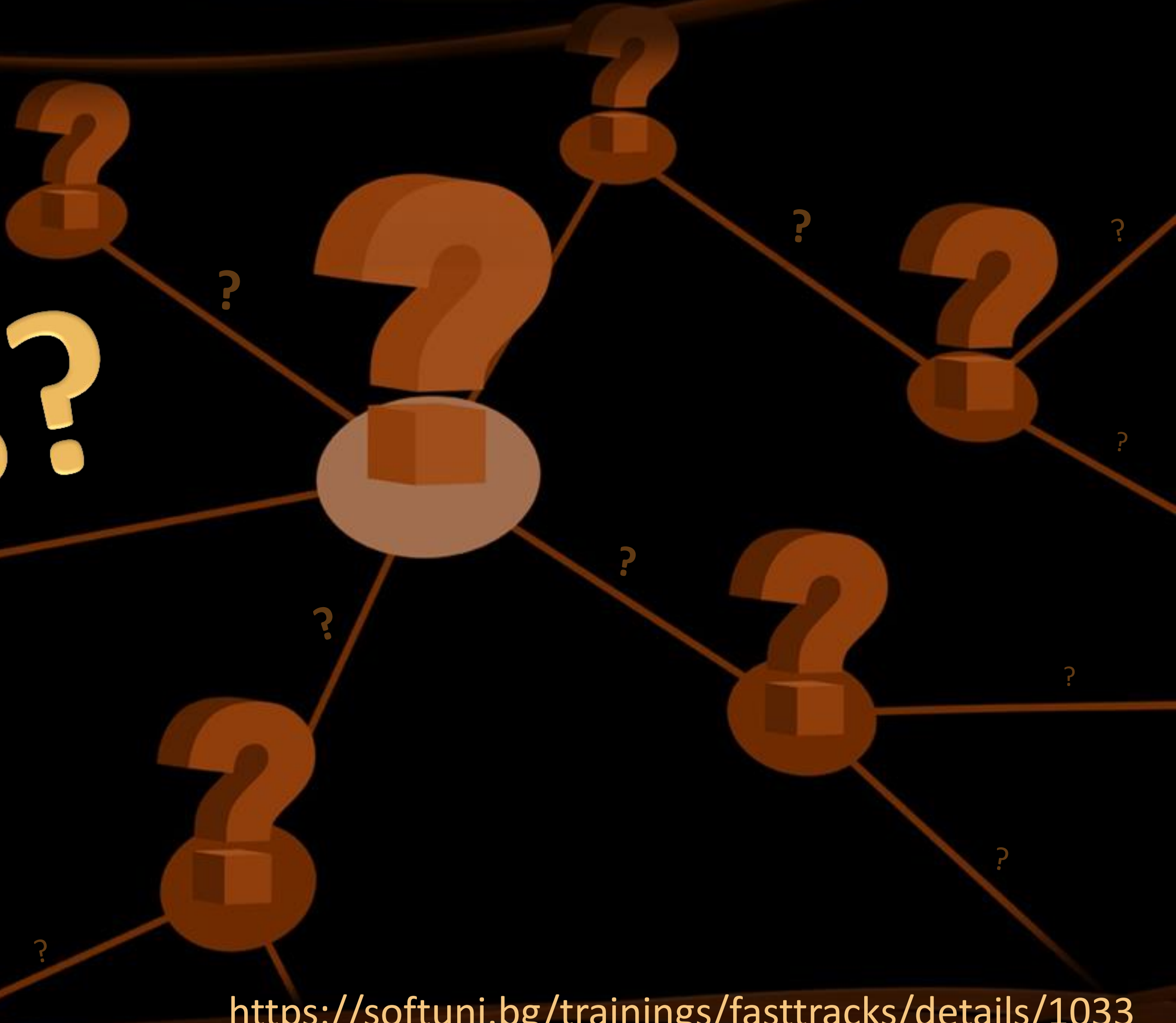
- Nowdoc strings - (since PHP 5.3.0) string syntax works essentially like single quoted strings
- The difference is that not even single quotes or backslashes have to be escaped.
- A nowdoc is identified with the same <<< sequence used for heredocs, but the identifier which follows is enclosed in single quotes, e.g. <<<'EOT‘
- No parsing is done in nowdoc.

Summary

- You can use DateTime and DateInterval classes to manipulate dates and intervals
- You can manipulate DOM and XML
- There are various types of Strings in PHP

PHP & MySQL

Questions?



https://softuni.bg/trainings/fasttracks/details/1033

License

Free Trainings @ Software University

- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University @ YouTube
  - youtube.com/SoftwareUniversity
- Software University Forums – forum.softuni.bg