

Reflection Capabilities

Introduction

- ❖ The PHP Reflection API includes various methods and objects we can use to inspect PHP code during runtime. This capability can be very useful especially for reverse engineering, generating documentation and determining whether a specific functionality is available or not.

Introduction

- ❖ The `Reflector` interface is implemented by all reflection classes. The `Reflector` interface defines the shared functionality between all reflection classes.
It includes the definition for two methods: `_toString()` and `export()`.
- ❖ The `ReflectionException` extends `Exception` and is thrown by the various functions the reflection classes include.

The Reflection Classes

❖ The reflection classes include the following:

ReflectionFunction

ReflectionParameter

ReflectionClass

ReflectionObject

ReflectionMethod

ReflectionProperty

ReflectionExtension

The ReflectionFunction Class

- ❖ This class represents a function. For every function in our PHP script we can get a ReflectionFunction object that represents it.

Once we get a ReflectionFunction object we can call various methods on it and get detailed information about the function it represents.

The ReflectionFunction Class

```
<?php

function doSomething()
{
    echo "<B>Do Something</B><BR>";
}

function sayHello($ob = "Students")
{
    echo "<B>Hello " + $ob + "</B>";
}

function sayGoodMorning($ob = "Students")
{
    echo "<B>Good Morning " + $ob + "</B>";
}

$funcs = get_defined_functions();
```



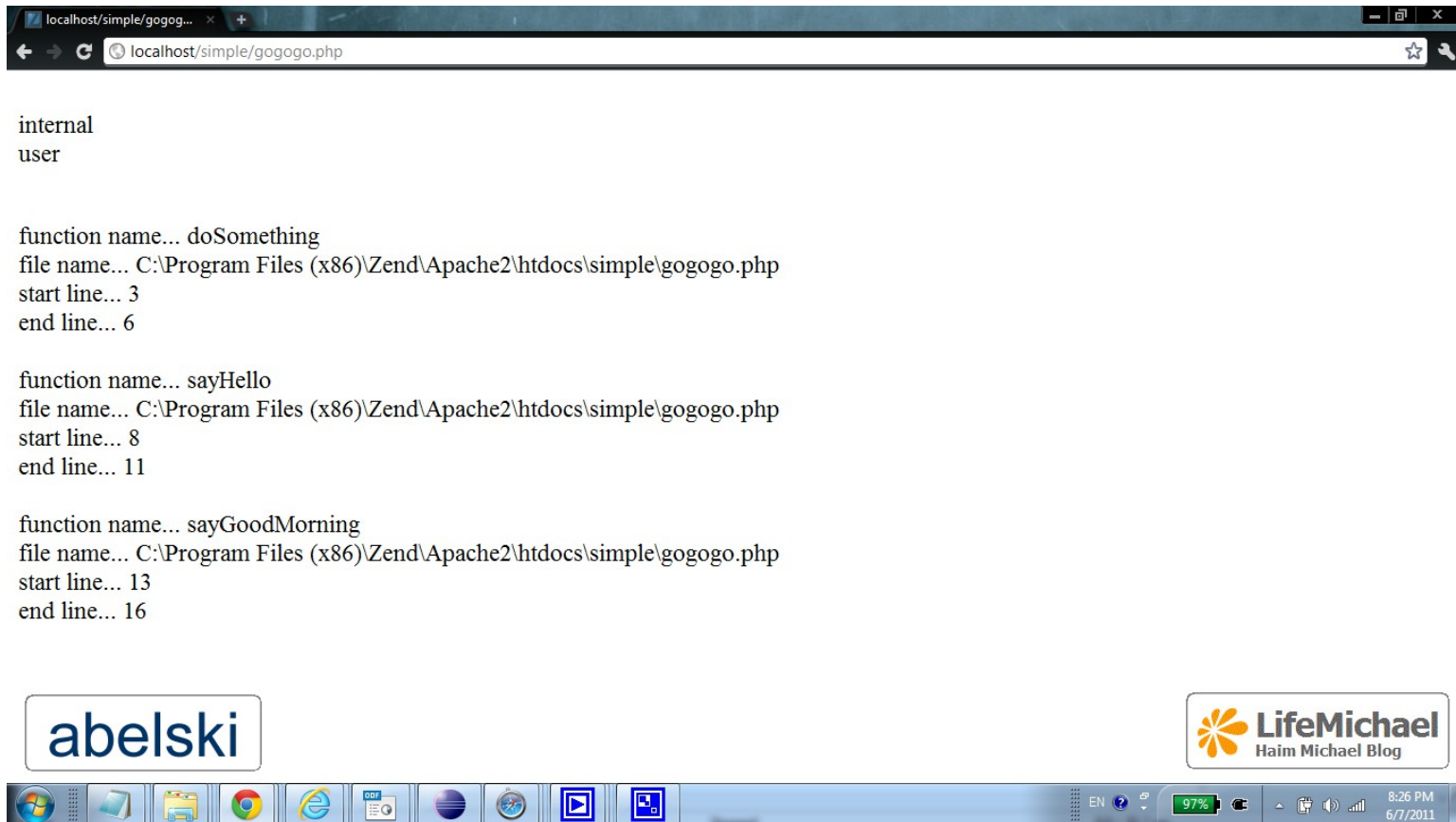
The ReflectionFunction Class

```
foreach($funcs as $k=>$v)
{
    echo "<br>$k";
}

echo "<p>";

foreach($funcs['user'] as $function)
{
    try
    {
        $obj = new ReflectionFunction($function);
        echo "<BR>function name... ".$obj->getName();
        echo "<BR>file name... ".$obj->getFileName();
        echo "<BR>start line... ".$obj->getStartLine();
        echo "<BR>end line... ".$obj->getEndLine();
        echo "<BR>";
    }
    catch(ReflectionException $exception)
    {
        echo "<BR><B>Exception Caught</B><BR>";
    }
}
?>
```

Sample



The screenshot shows a web browser window with the address bar containing 'localhost/simple/gogogo.php'. The main content area displays three PHP error messages:

```
internal  
user  
  
function name... doSomething  
file name... C:\Program Files (x86)\Zend\Apache2\htdocs\simple\gogogo.php  
start line... 3  
end line... 6  
  
function name... sayHello  
file name... C:\Program Files (x86)\Zend\Apache2\htdocs\simple\gogogo.php  
start line... 8  
end line... 11  
  
function name... sayGoodMorning  
file name... C:\Program Files (x86)\Zend\Apache2\htdocs\simple\gogogo.php  
start line... 13  
end line... 16
```

At the bottom of the browser window, there are two logos: 'abelski' on the left and 'LifeMichael Haim Michael Blog' on the right. Below the browser window is a Windows taskbar with various application icons and system tray information including '97%' battery, '8:26 PM', and '6/7/2011'.

The ReflectionParameter Class

❖ Objects of this class represent parameters.

Once we get a ReflectionParameter object we can call various methods on it and get detailed information about the parameter it represents.

The ReflectionParameter Class

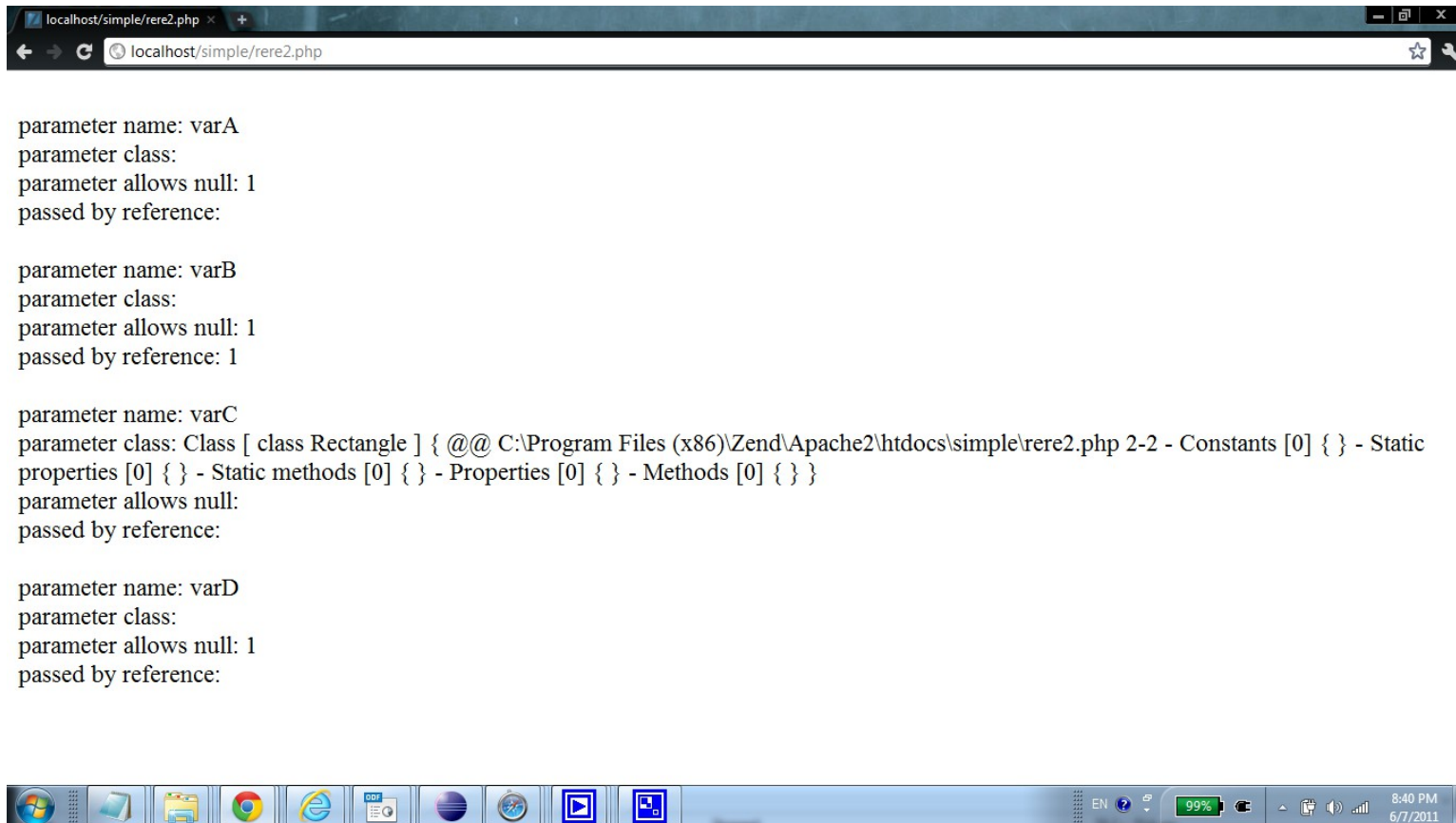
```
<?php
class Rectangle {}

function doSomething($varA, &$varB, Rectangle $varC, $varD=12)
{
    echo "<br>something<br>";
}

$obj = new ReflectionFunction("doSomething");

foreach ($obj->getParameters() as $index => $param)
{
    echo "<br>parameter name: ".$param->getName();
    echo "<br>parameter class: ".$param->getClass();
    echo "<br>parameter allows null: ".$param->allowsNull();
    echo "<br>passed by reference: ".$param->isPassedByReference();
    echo "<br>";
}
?>
```

The ReflectionParameter Class



The ReflectionClass Class

- ❖ Objects of this class represent specific classes.

Once we get a ReflectionClass object we can call various methods on it and get detailed information about the class it represents.

The ReflectionClass Class

```
<?php
```

```
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW,$valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
}
```



The ReflectionClass Class

```
function setHeight($val)
{
    if($val>0)
    {
        $this->height = $val;
    }
}

$obj = new ReflectionClass("Rectangle");

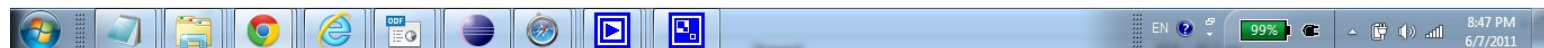
echo "<br>\$obj->isAbstract()=".$obj->isAbstract();
echo "<br>\$obj->isFinal()=".$obj->isFinal();
echo "<br>\$obj->isInterface()=".$obj->isInterface();
echo "<br>\$obj->getName()=".$obj->getName();
echo "<br>\$obj->getStartLine()=".$obj->getStartLine();
echo "<br>\$obj->getEndLine()=".$obj->getEndLine();

?>
```

The ReflectionClass Class



```
$ob->isAbstract()=  
$ob->isFinal()=  
$ob->isInterface()=  
$ob->getName()=Rectangle  
$ob->getStartLine()=3  
$ob->getEndLine()=31
```



The ReflectionMethod Class

- ❖ Objects of this class represent specific methods.

Once we get a ReflectionMethod object we can call various methods on it and get detailed information about the method it represents.

The ReflectionMethod Class

```
<?php
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW, $valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
}
```



The ReflectionMethod Class

```
function setHeight($val)
{
    if($val>0)
    {
        $this->height = $val;
    }
}

$obj = new ReflectionMethod('Rectangle', 'setHeight');

echo "<br>\$obj->isAbstract()=" . $obj->isAbstract();
echo "<br>\$obj->isFinal()=" . $obj->isFinal();
echo "<br>\$obj->isPublic()=" . $obj->isPublic();
echo "<br>\$obj->isPrivate()=" . $obj->isPrivate();
echo "<br>\$obj->getName()=" . $obj->getName();
echo "<br>\$obj->getStartLine()=" . $obj->getStartLine();
echo "<br>\$obj->getEndLine()=" . $obj->getEndLine();

?>
```

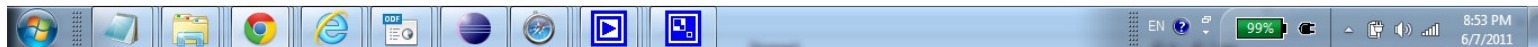
The ReflectionMethod Class



```
$ob->isAbstract()=  
$ob->isFinal()=  
$ob->isPublic()=1  
$ob->isPrivate()=  
$ob->getName()=setHeight  
$ob->getStartLine()=22  
$ob->getEndLine()=28
```

abelski

 LifeMichael
Haim Michael Blog



The ReflectionProperty Class

❖ Objects of this class represents specific a property.

Once we get a ReflectionProperty object we can call various methods on it and get detailed information about the property it represents.

The ReflectionProperty Class

```
<?php
class Rectangle
{
    private $width;
    private $height;
    function Rectangle($valW,$valH)
    {
        $this->setHeight($valH);
        $this->setWidth($valW);
    }
    function area()
    {
        return $this->width * $this->height;
    }
    function setWidth($val)
    {
        if($val>0)
        {
            $this->width = $val;
        }
    }
}
```

The ReflectionProperty Class

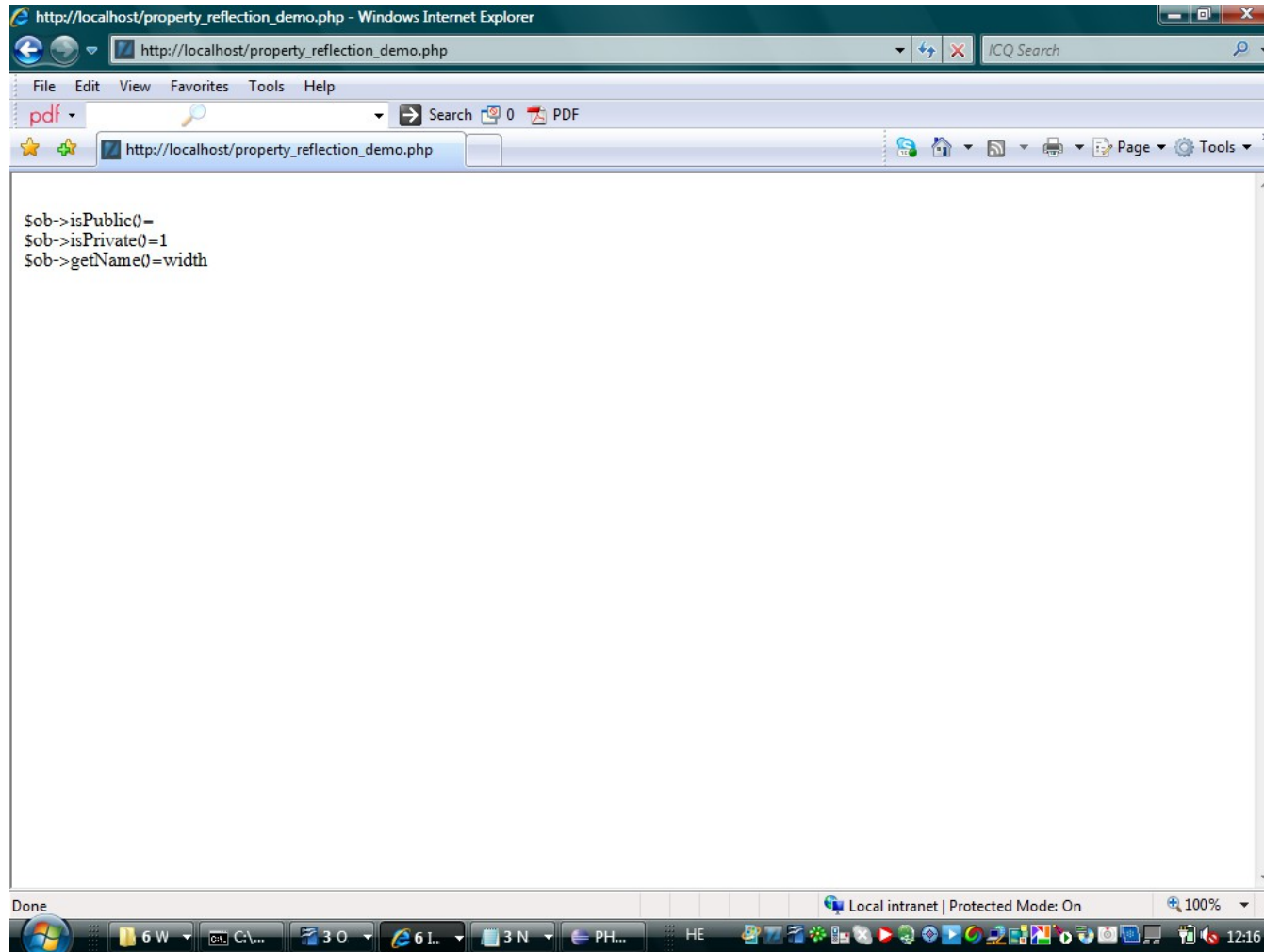
```
function setHeight($val)
{
    if($val>0)
    {
        $this->height = $val;
    }
}

$obj = new ReflectionProperty('Rectangle','width');

echo "<br>\$obj->isPublic()=" . $obj->isPublic();
echo "<br>\$obj->isPrivate()=" . $obj->isPrivate();
echo "<br>\$obj->getName()=" . $obj->getName();

?>
```

The ReflectionProperty Class



The ReflectionExtension Class

- ❖ Objects of this class represent extensions installed on our PHP execution environment.

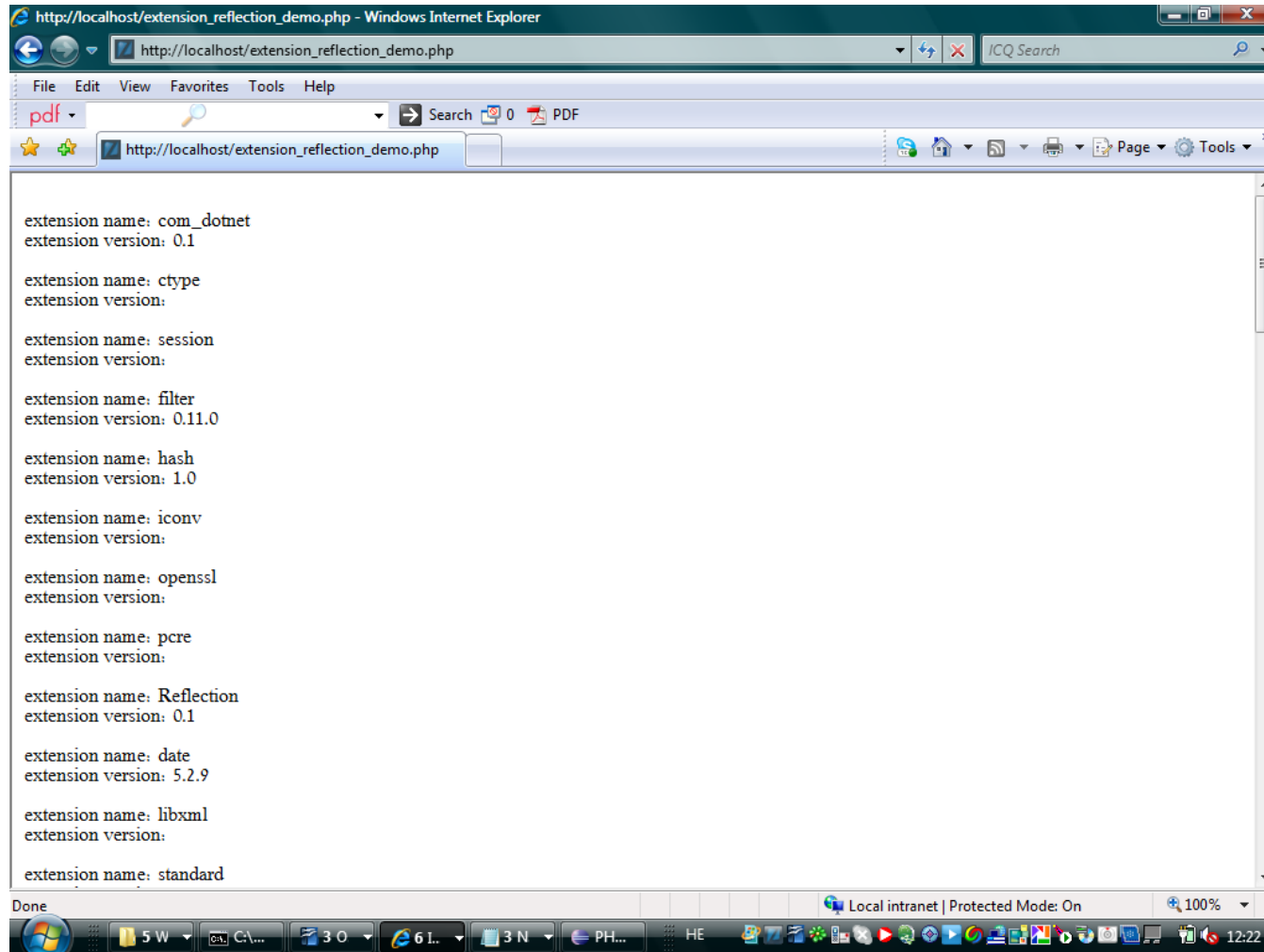
Once we get a ReflectionExtension object we can call various methods on it and get detailed information about the extension it represents.

The ReflectionExtension Class

```
<?php
$extensions = get_loaded_extensions();

foreach($extensions as $key => $value)
{
    $obj = new ReflectionExtension($value);
    echo "<br>extension name: ".$value;
    echo "<br>extension version: ".$obj->getVersion();
    echo "<br>";
}
?>
```

The ReflectionExtension Class



```
extension name: com_dotnet
extension version: 0.1

extension name: ctype
extension version:

extension name: session
extension version:

extension name: filter
extension version: 0.11.0

extension name: hash
extension version: 1.0

extension name: iconv
extension version:

extension name: openssl
extension version:

extension name: pcre
extension version:

extension name: Reflection
extension version: 0.1

extension name: date
extension version: 5.2.9

extension name: libxml
extension version:

extension name: standard
```

The ReflectionExtension Class

- ❖ Objects of this class represent extensions installed on our PHP execution environment.

Once we get a ReflectionExtension object we can call various methods on it and get detailed information about the extension it represents.

The `::class` Keyword

- ❖ When adding `::class` to a class name we will get the fully qualified name of that class.

The ::class Keyword

```
<?php
namespace com\lifemichael\samples;
class Rectangle{}
echo Rectangle::class;
?>
```



The ::class Keyword

