Scalar Type Declarations and Anonymous Classes

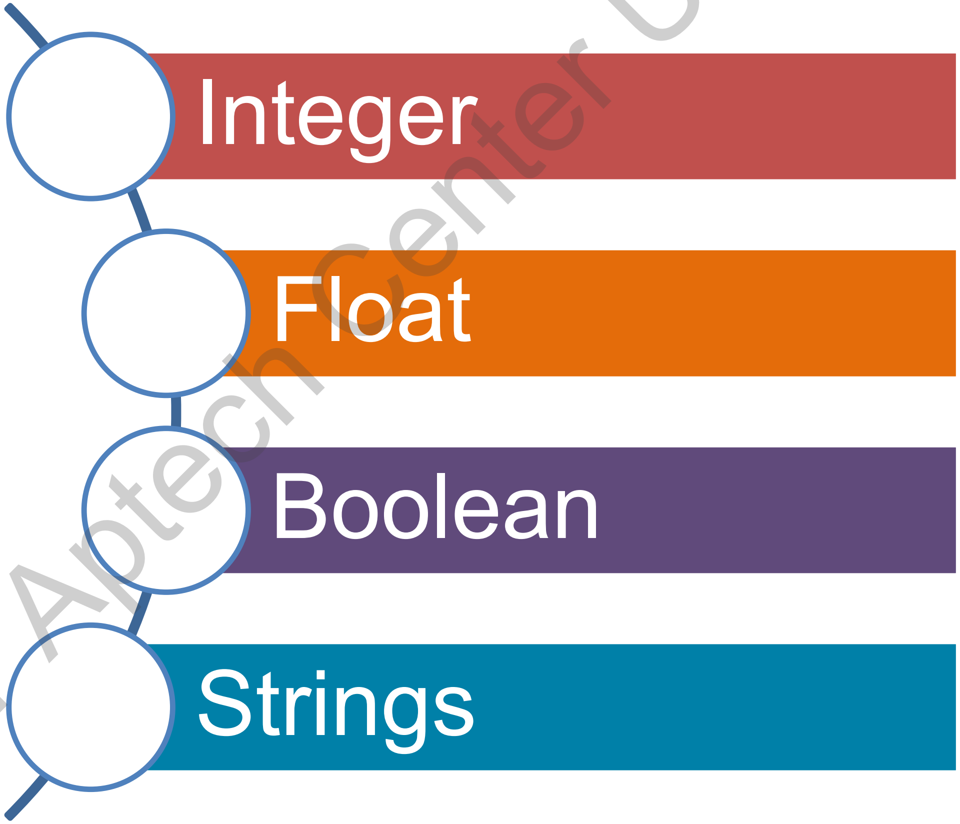
Session 7

Objectives

- *Explain scalar type declarations.*
- *Describe usage of scalar type declarations in PHP programs.*
- *Explain scalar type hinting.*
- *Explain anonymous classes.*
- *Describe usage of anonymous classes in PHP programs.*

Scalar Data Type

- The data types that hold single data type are known as scalar data types.
- The data types can be:



Type Declarations in PHP

- Refers to specifying the data type of a parameter in a function
- Also referred to as type hinting
- Provides hints to a function
- Enforces the input the parameter data type, that can be either strict or coercive

Explicit Declarations of Scalar Types

- Declaring a new function

Syntax

```
function function_name(type, p1)
```

Where,

- **m** – is the dividend
- **n** – is the divisor

Declaring and Using Scalar Types 1-10

- Displaying the value of a parameter in `int` data type

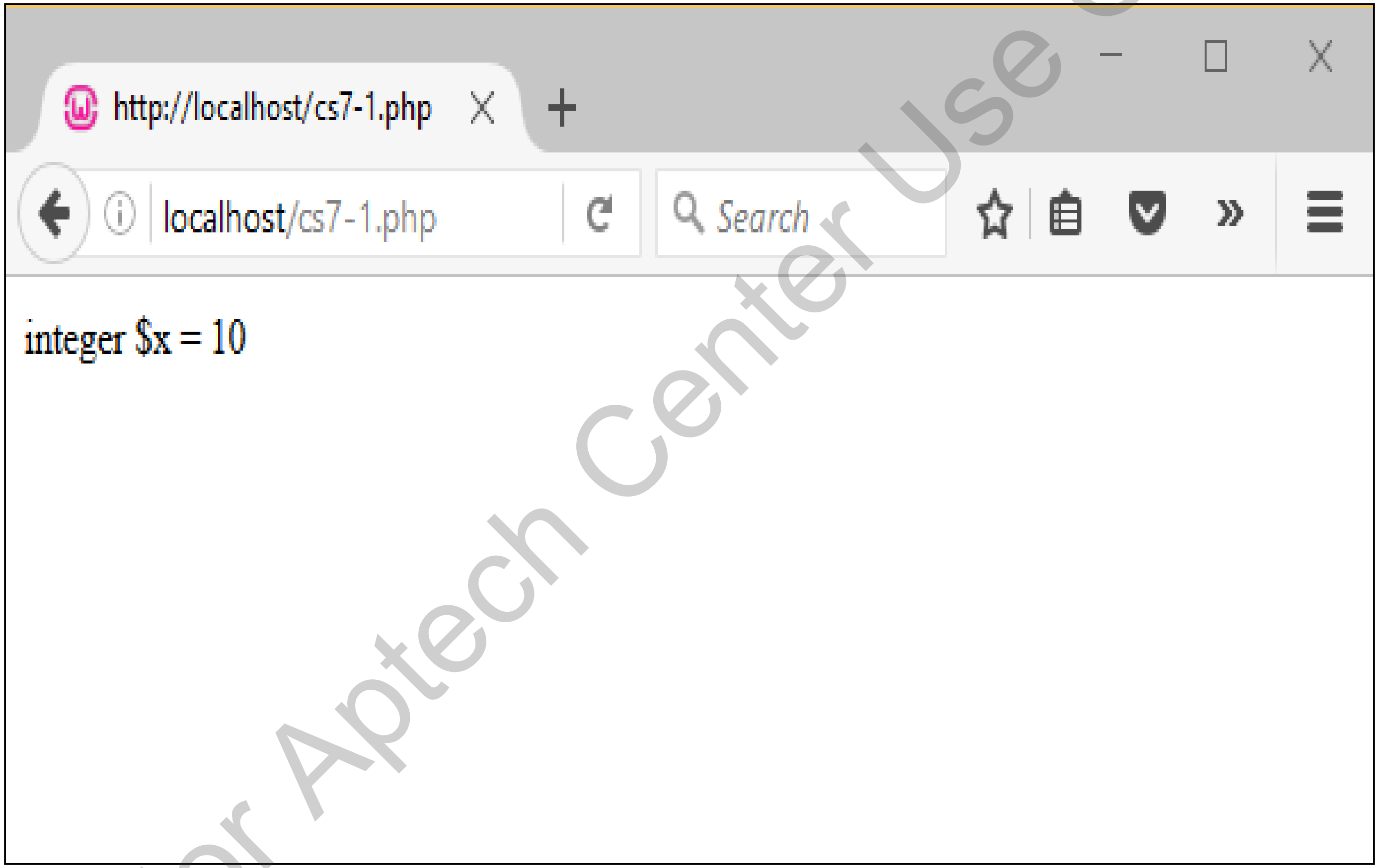
Snippet

```
<?php
function test1(int $x){
echo 'integer $x = '.$x;
}
test1(10.124);
//output: integer $x = 10
?>
```

- `test1` - is the function name
- `int` – is the data type
- `$x` – is the parameter

Declaring and Using Scalar Types

Displays the following output:



```
integer $x = 10
```

Declaring and Using Scalar Types 3-10

- Displaying the value of a parameter in `float` data type

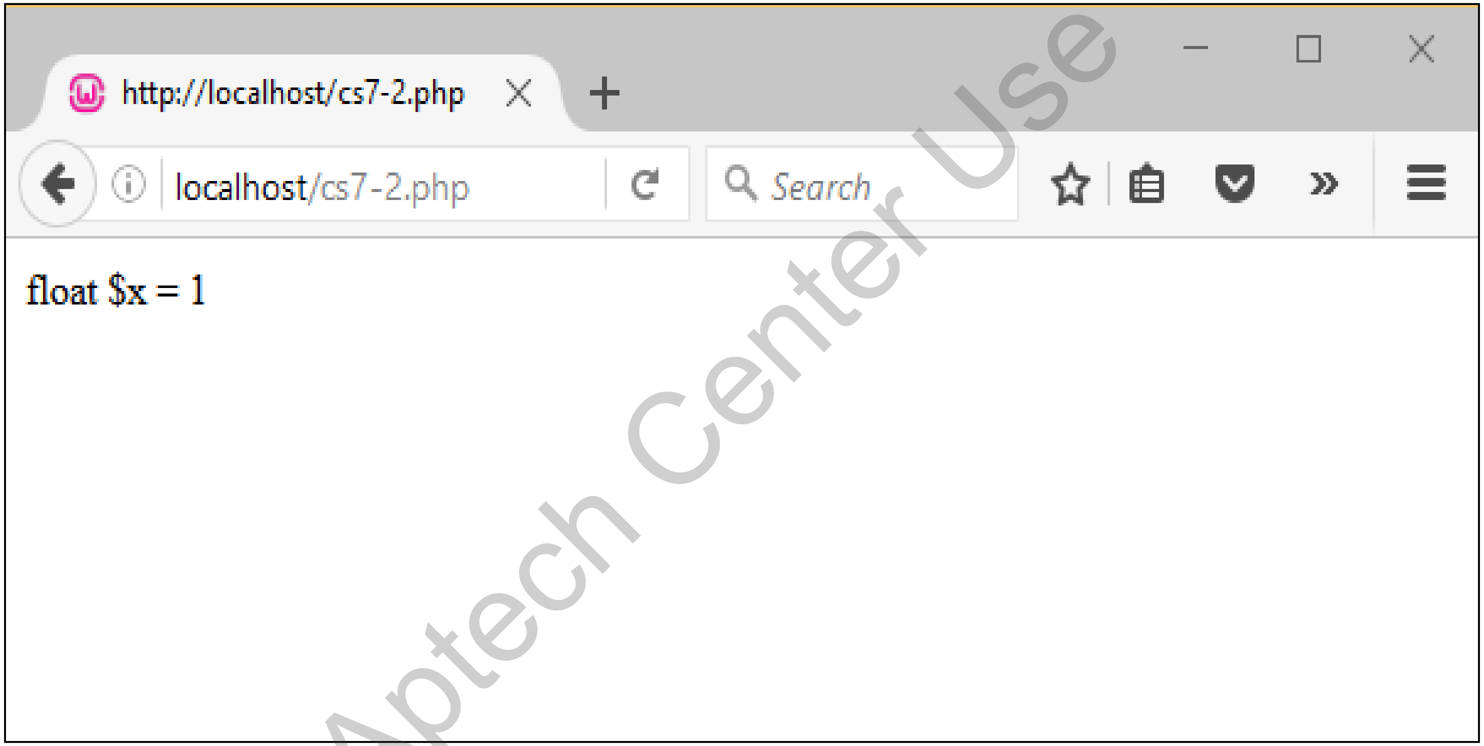
Snippet

```
<HTML>
<BODY>
<?php
function test1(float $x){
echo 'float $x = '.$x;
}
test1(true);
 ?>
 </BODY>
 </HTML>
```

- `test1` – is the function name
- `float` – is the data type
- `$x` – is the parameter

Declaring and Using Scalar Types

Displays the following output:



```
float $x = 1
```

Declaring and Using Scalar Types

- Displaying the value of a parameter in `boolean` data type

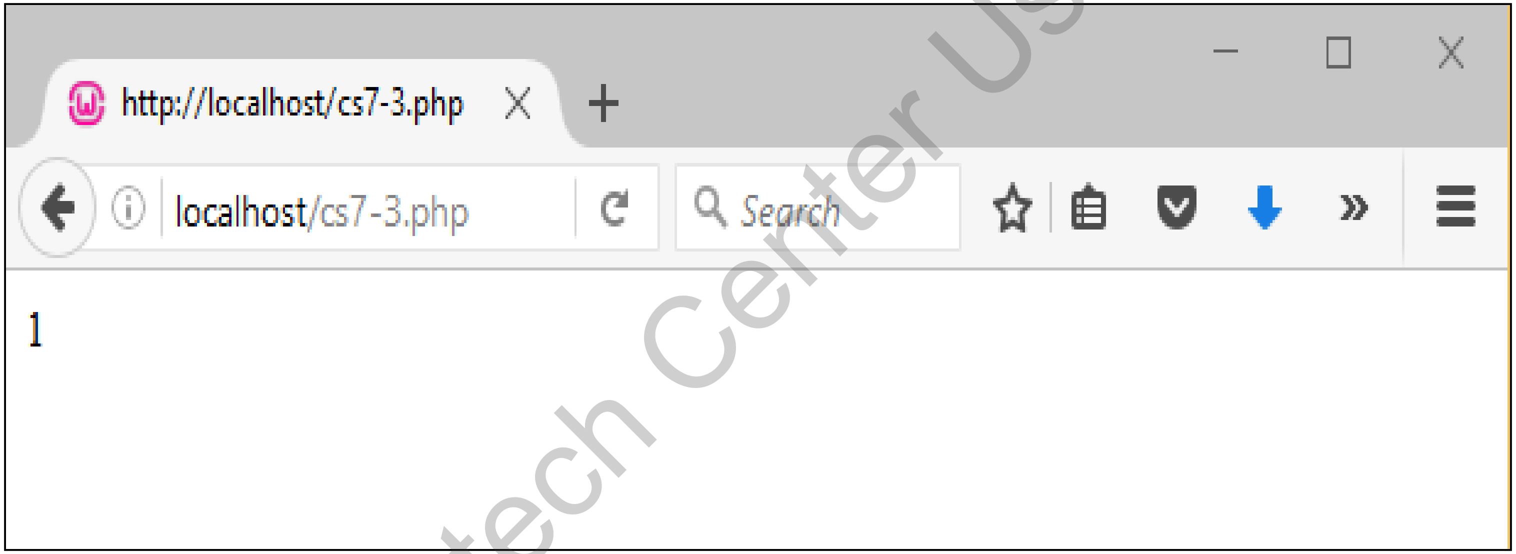
Snippet

```
<HTML>
<BODY>
<?php
 function test1(bool $a){
 echo $a;
  }
 test1(10.34);//
?>
</BODY>
</HTML>
```

- `test1` - is the function name
- `bool` – is the data type
- `$a` – is the parameter

Declaring and Using Scalar Types

Displays the following output:



```
1
```

Declaring and Using Scalar Types 7-10

- Displaying the value of a parameter in `string` data type

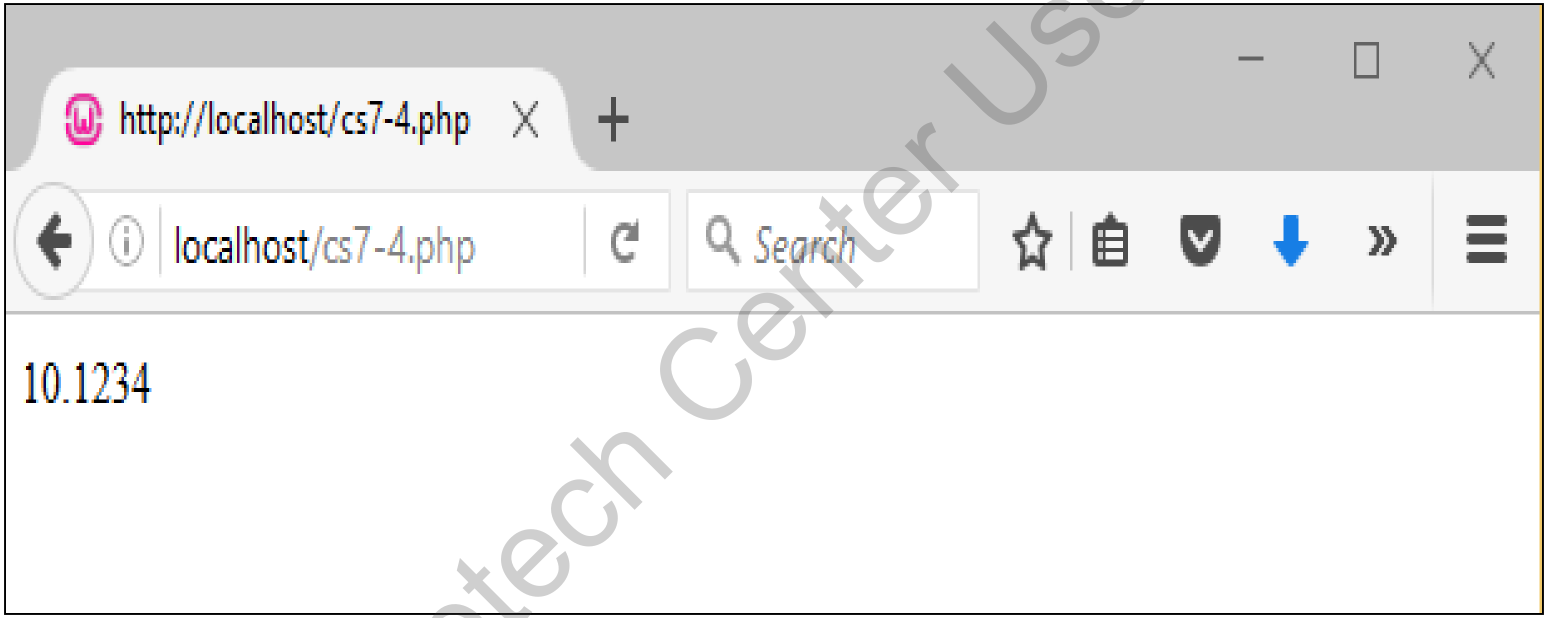
Snippet

```
<HTML>
<BODY>
<?php
 function test1(string $a){
 echo $a;
 }
 test1(10.1234);
?>
</BODY>
</HTML>
```

- `test1` – is the function nam
- `string` – is the data type
- `$a` – is the parameter

Declaring and Using Scalar Types

Displays the following output:



```
10.1234
```

Declaring and Using Scalar Types 9-10

- Demonstrating strict type declaration

Snippet

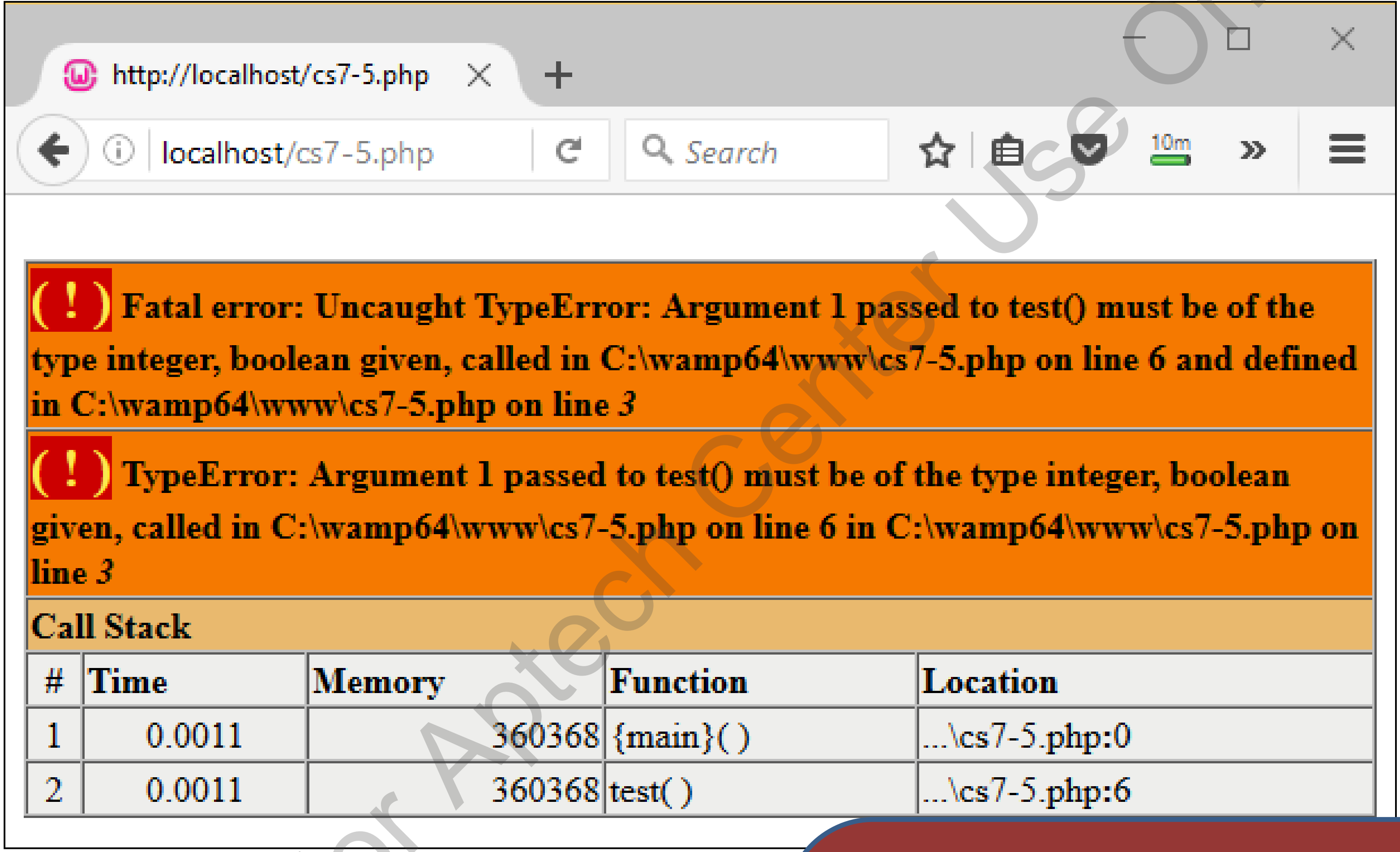
```
<?php declare(strict_types=1);
 function test1(int $a){
 echo $a;
 }
 test1(true);
?>
```

The `declare` statement explicitly declares the scalar type, that are strictly checked.

The `function test1` is the function name that has an `int` data type.

The `echo` statement prints the value of the argument.

Displays the following output:



(!) Fatal error: Uncaught TypeError: Argument 1 passed to test() must be of the type integer, boolean given, called in C:\wamp64\www\cs7-5.php on line 6 and defined in C:\wamp64\www\cs7-5.php on line *3*

(!) TypeError: Argument 1 passed to test() must be of the type integer, boolean given, called in C:\wamp64\www\cs7-5.php on line 6 in C:\wamp64\www\cs7-5.php on line *3*

Call Stack

#	Time	Memory	Function	Location
1	0.0011	360368	{main}( )	...\cs7-5.php:0
2	0.0011	360368	test( )	...\cs7-5.php:6

The program fails to execute and terminates prematurely because the argument types does not match the parameter type.

Behavior of Weak Type Checking and Type Conversion 1-2

- Weak type conversion is enforced using the `declare(strict_types=0)` statement
- Rules to be considered while using weak type checking:

Calls to a built-in PHP function or to an extension have the same behavior as earlier versions

Weak type new scalar type declarations are same as that of built-in PHP function or to an extension

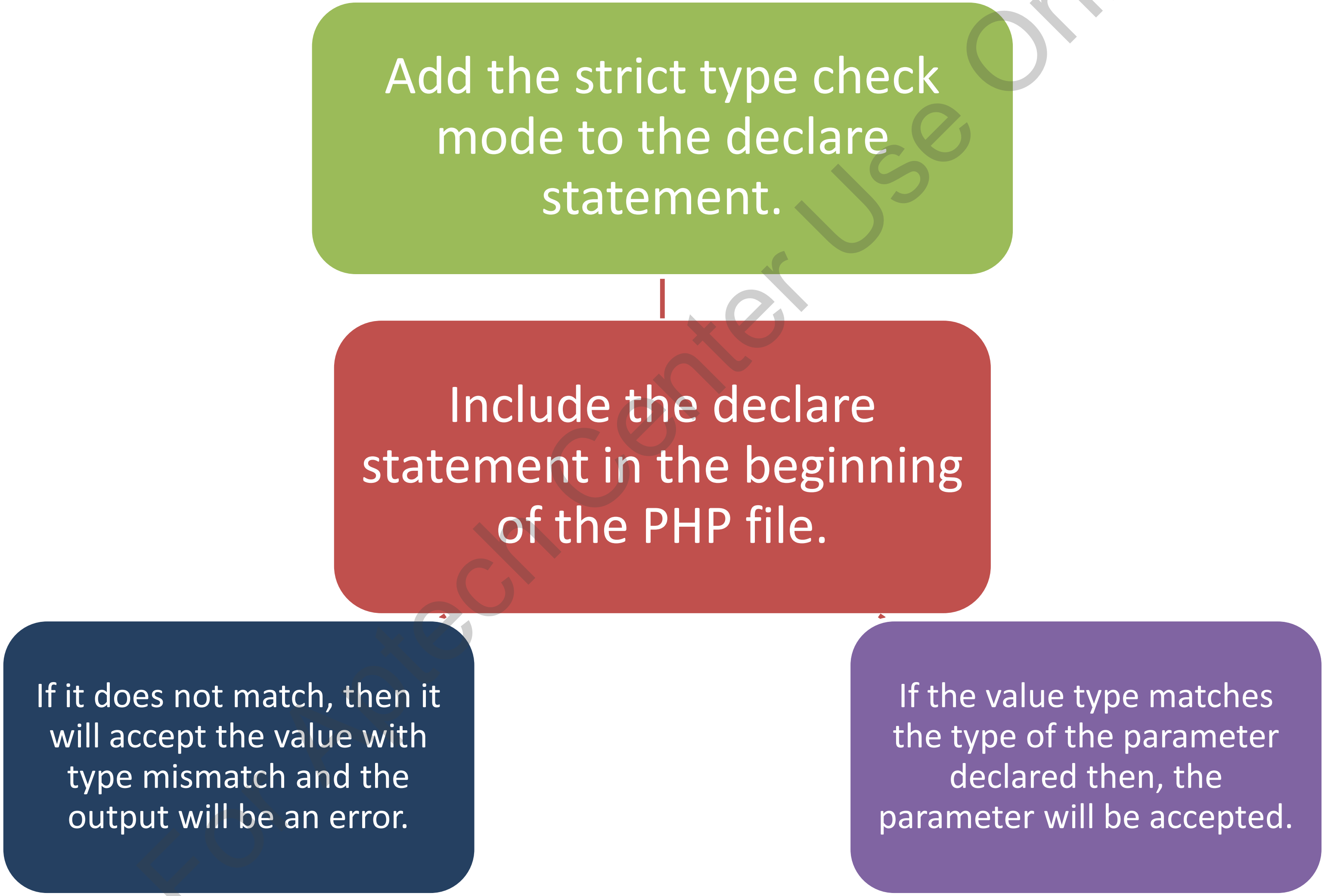
NULL is not excepted unless it is a parameter and is explicitly given a default value

Behavior of Weak Type Checking and Type Conversion 2-2

- An implicit scalar conversion in a weak mode

Type Declaration	int	float	string	bool
int	yes	yes	yes	yes
float	yes	yes	yes	yes
string	yes	yes	yes	yes
bool	yes	yes	yes	yes

Behavior of Strict Type Checking



Add the strict type check mode to the declare statement.

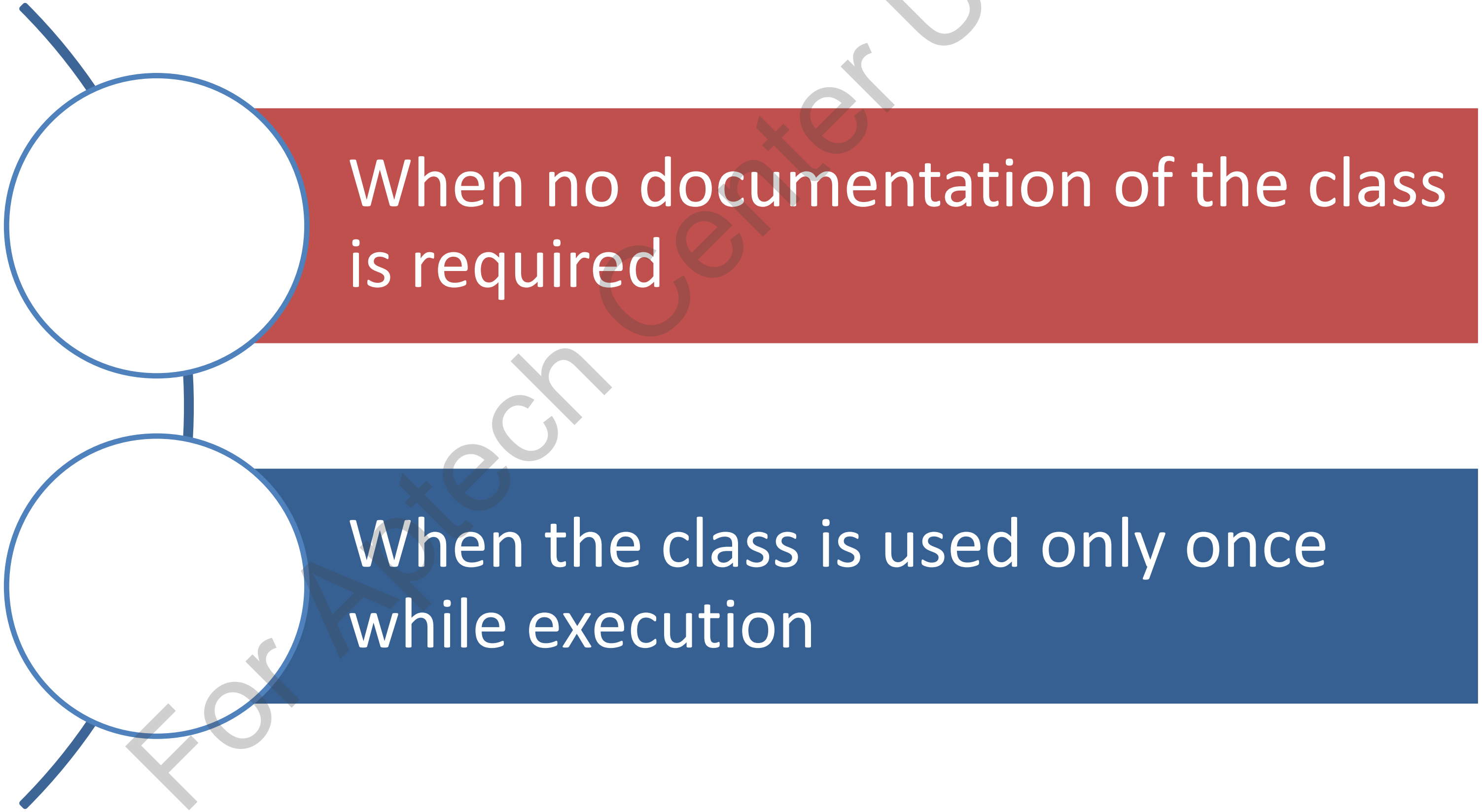
Include the declare statement in the beginning of the PHP file.

If it does not match, then it will accept the value with type mismatch and the output will be an error.

If the value type matches the type of the parameter declared then, the parameter will be accepted.

Anonymous Class

- Is a class that does not have a name
- Can be used in the following scenarios:



When no documentation of the class is required

When the class is used only once while execution

Creating Anonymous Classes 1-5

- Demonstrating creation of a named class

Step 1

Use the keyword `class` before the class name

Step 2

Enclose the property and method definitions within curly braces

```
class class_Name {

 // defined properties and methods

};

$object = new class_Name( 'arguments' );
```

Creating Anonymous Classes

Step 1	Create a public variable.
Step 2	Assign values to the public variables.
Step 3	Create a function within the anonymous class without any arguments.
Step 4	Create a second function within the anonymous class with one argument.

The first function will return a message to the user whereas the second function will return the value of the argument.

Creating Anonymous Classes 3-5

Snippet

```
<?php
// anonymous_class.php
// PHP 7

$anon_class_obj = new class{
 public $greeting = 'hello';
 public $Id = 754;
 const SETT = 'some configuration';

 public function getValue()
 {
 // do some operation
 return 'some returned value';
 }

 public function getValueWithArg($str1)
 {
 // do some operation
 return 'returned value is '.$str1;
 }
 };
```

`$greeting`, `$Id`, and `SETT` are members of the anonymous class.

`'hello', 754`, and `'some configuration'` are the values assigned to these members.

`getValue` and `getValueWithArgument` are the functions within the class.

The `getValue function` will display the message `‘Some Returned Value’`

The `getValueWithArgument` function will display the value of its argument.

Creating Anonymous Classes

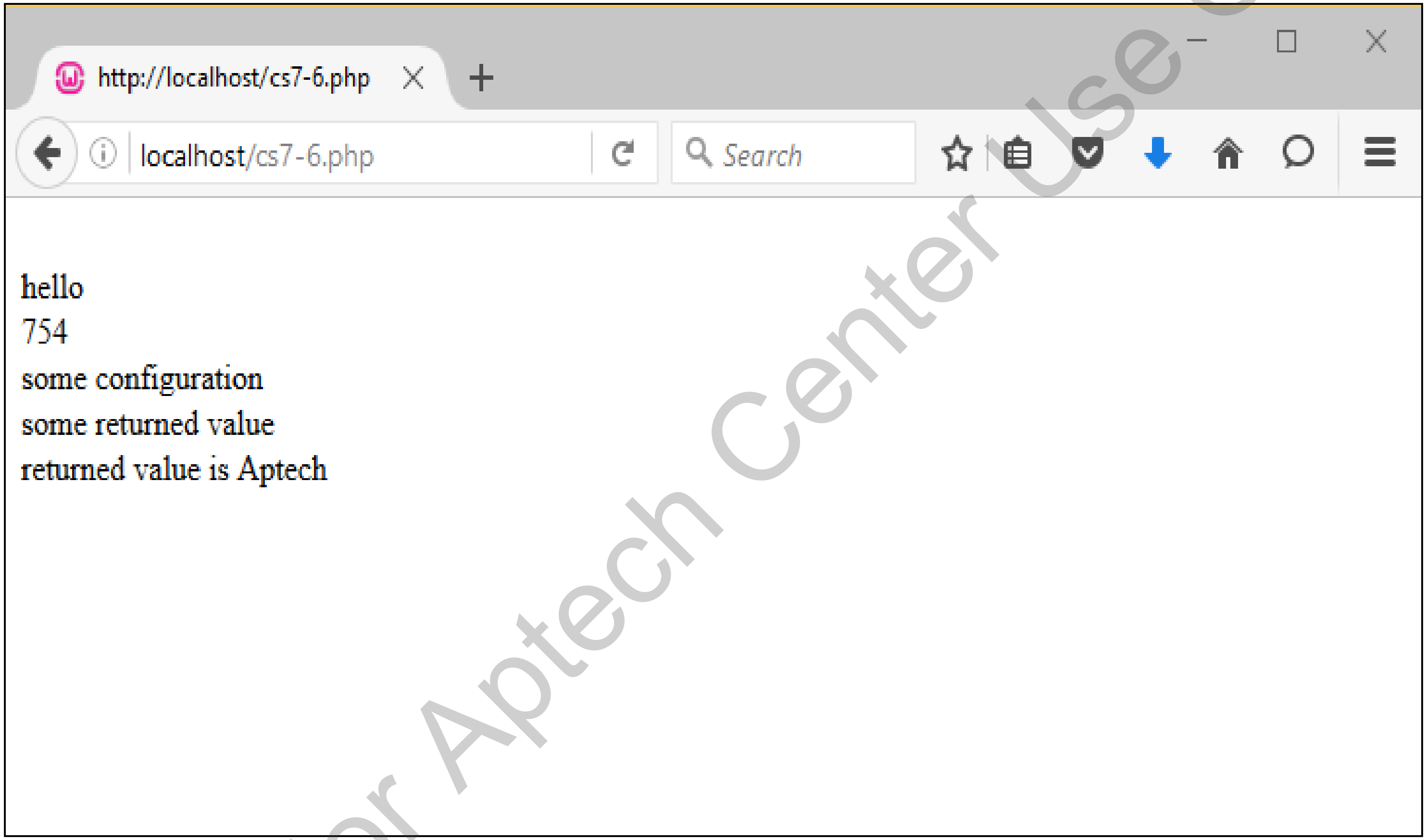
Snippet

```
echo '</br>';

echo $anon_class_obj->greeting;
echo '</br>';
echo $anon_class_obj->Id;
echo '</br>';
echo $anon_class_obj::SETT;
echo '</br>';

echo $anon_class_obj->getValue();
echo '</br>';

echo $anon_class_obj->getValueWithArg('Aptech');
echo '</br>';

echo '</br>';
?>
```

The `echo` statements will display the values of the members and functions of the anonymous class.

Creating Anonymous Classes

Displays the following output:



```
hello
754
some configuration
some returned value
returned value is Aptech
```

Summary

1-2

- Scalar data types are those types that hold single values. These data types can either be integer, string, Boolean, or float.
- Type declaration refers to process of specifying the data type of the parameter when passing it to a function. They are also known as type hinting.
- Scalar type hints help design reliable PHP code.
- To enable strict type checking, use the declare statement to declare strict_types directive. Any type mismatch with function arguments results in an error.

Summary

- An anonymous class is a class that is defined without a name.

- To create anonymous class, combine new class ($constructor, $args) followed by a standard class definition.