shawacademy

Diploma in Web Development – Part II

PHP Development – Week 3

Error Handling & Advanced Development

Presented by:
Julian Quirke
Web Development Educator

Week 2 Recap

Cookies & PHP Sessions

- What is a Cookie?
- Creating PHP Sessions
- PHP Session Demo

- Summary
- Q&A

Today's Lesson

Error Handling & Advanced Development

- Class Member Visibility
- Abstract Classes & Interfaces
- Error Handling in PHP

- Summary
- Q&A

Let's Begin!

Class Member Visibility

shawacademy

Visibility

Julian Quirke
Web Development Educator

ADVANCED WEB DEVELOPMENT

Class Member Visibility

Visibility

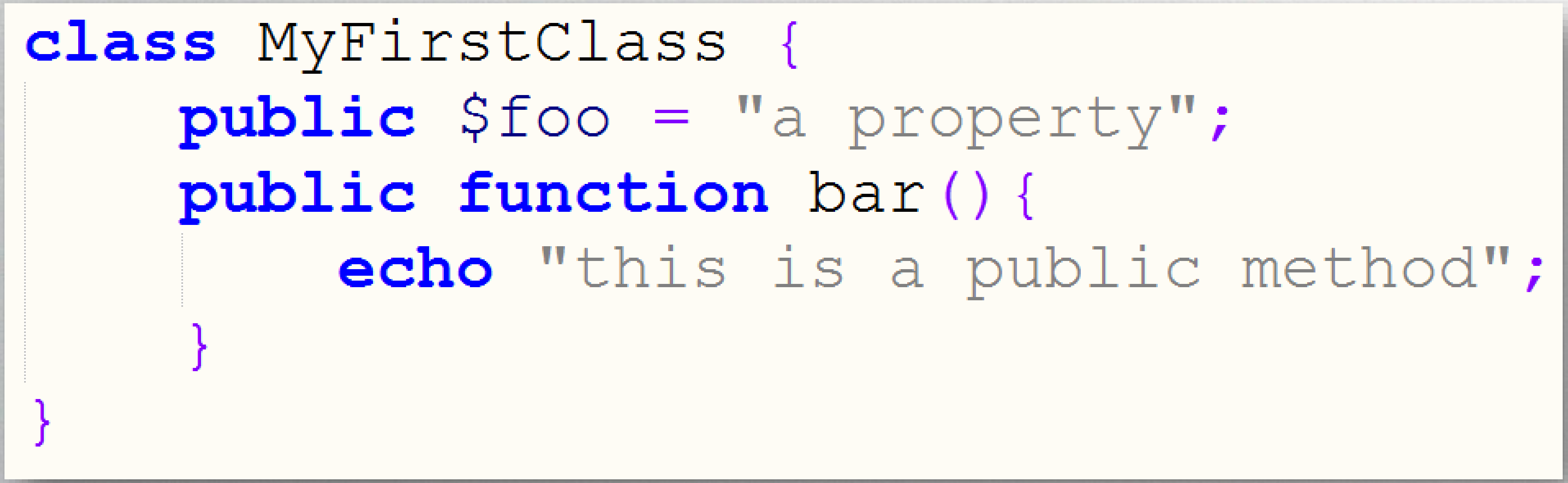
is the accessibility of a member outside of an object's scope

Plays an important role in encapsulation of objects

Class Member Visibility

Levels of Visibility: Public

- Use the “**public**” keyword to declare public visibility
- Accessible from anywhere that has a reference to the object



```php
class MyFirstClass {
    public $foo = "a property";
    public function bar(){
        echo "this is a public method";
    }
}
```

Class Member Visibility

Levels of Visibility: Protected

- Declared using the "**protected**" keyword
- Accessible from within class and child classes
- **Cannot** be accessed from outside object!
- Useful for internally managed information

Class Member Visibility

shawacademy

Levels of Visibility: Protected

```
class MyFirstClass {
    protected $foo = "a property";
    public function getFoo(){
        echo $this->foo;
    }
}
$myInstance = new MyFirstClass();
echo $myInstance->foo; //error
$myInstance->getFoo(); //this works just fine
```

Class Member Visibility

Levels of Visibility: Private

- Declared using the "**private**" keyword
- Accessible from within object scope **only**
- **Cannot** be accessed from global scope **or** child classes!
- Useful for internal information

Abstract Classes & Interfaces

shawacademy

Abstract Class

Abstract Classes & Interfaces

Abstract Class

is a blueprint for an object template that cannot be instantiated directly

Can be declared explicitly, or implicitly (when the class contains an abstract method)

Abstract Classes & Interfaces

shawacademy

Abstract Method

Abstract Classes & Interfaces

Abstract Method

is a method that cannot be called directly, but *must* be implemented by classes that inherit from the containing abstract class

Abstract Classes & Interfaces

Why build an abstract class?

- Allows us to build templates for objects that will have similar functionality
- We can optionally leave the actual implementation of a method to the inheriting class

Abstract Classes & Interfaces

In PHP

Use the "abstract" keyword to declare an abstract class or method:

```
abstract class myClass{
    //This class can't be instantiated
    abstract public function myMethod(){
        //This method is simply a placeholder and can't be called
    }
}
```

Abstract Classes & Interfaces

Interface

Abstract Classes & Interfaces

shawacademy

Interface

is a description of the actions that an object instance can perform

Only describe public method for classes that implement one

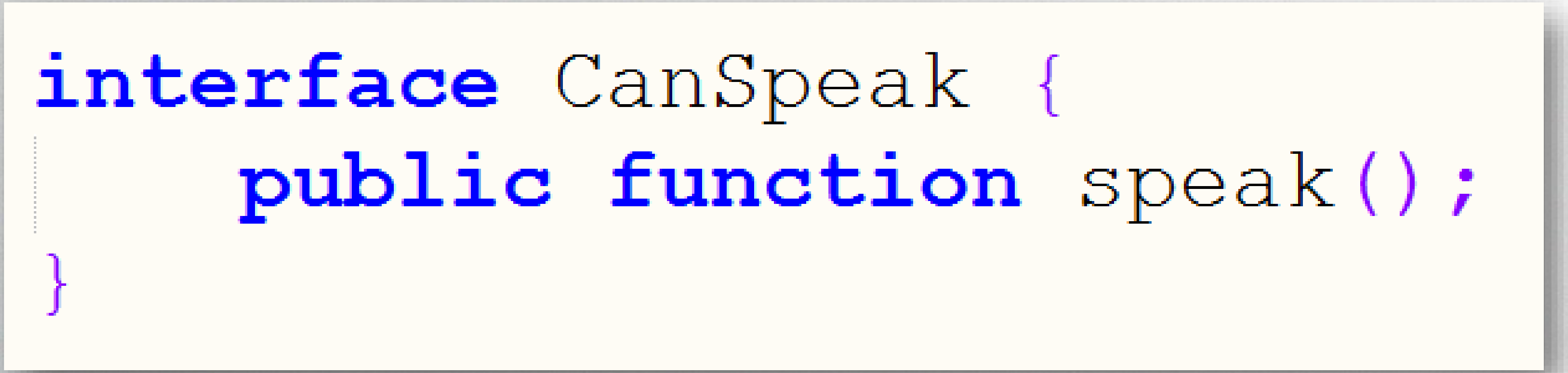
Abstract Classes & Interfaces

Why build an interface?

- Allows us to describe the expected actions of a class or set of classes
- We do *not* need to know how the methods are implemented
- E.g. How a dog speaks is different from how a person does, but both can inherit an abstract "speak()" method from the creature abstract class

Abstract Classes & Interfaces
In PHP

Use the "interface" keyword to declare an interface:



```php
interface CanSpeak {
    public function speak();
}
```

Abstract Classes & Interfaces

In PHP

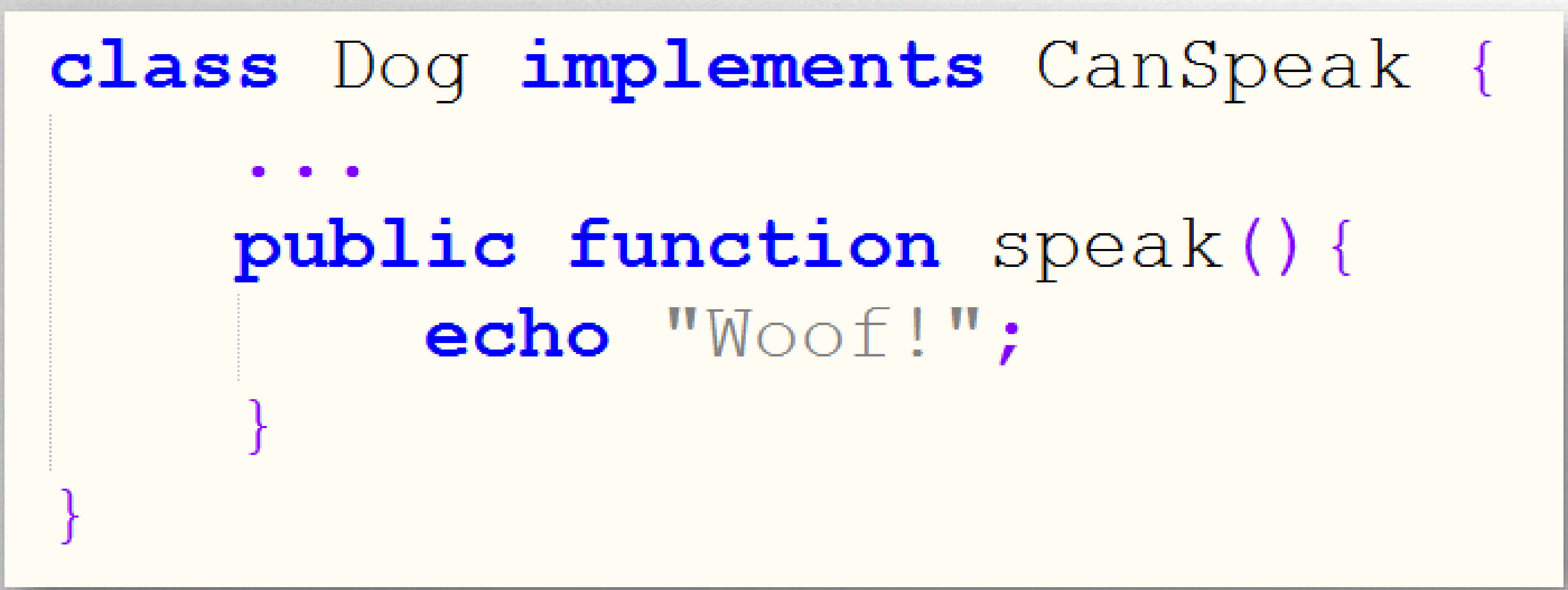
Use the “implements” keyword to implement an interface in a class:

```
class Person implements CanSpeak {
    ...
    public function speak(){
        echo "Hello";
    }
}
```

Abstract Classes & Interfaces

In PHP

Similarly:



```
class Dog implements CanSpeak {
    ...
    public function speak(){
        echo "Woof!";
    }
}
```

Abstract Classes & Interfaces

Abstract Classes vs Interfaces

Abstract Class	Interface
Is part of the modelled class hierarchy	Describes a the API for the classes that implement it
Child classes share behaviour with abstract parent (just like with regular parent-child class relationships)	Does not represent behaviour or any other aspect of the implementing class
Child class inherits at most one abstract class	Implementing class implements as many interfaces as appropriate to it

Abstract Classes & Interfaces

Abstract Classes & Interfaces

Error Handling in PHP

shawacademy

Error Handling

Error Handling in PHP

Error Handling

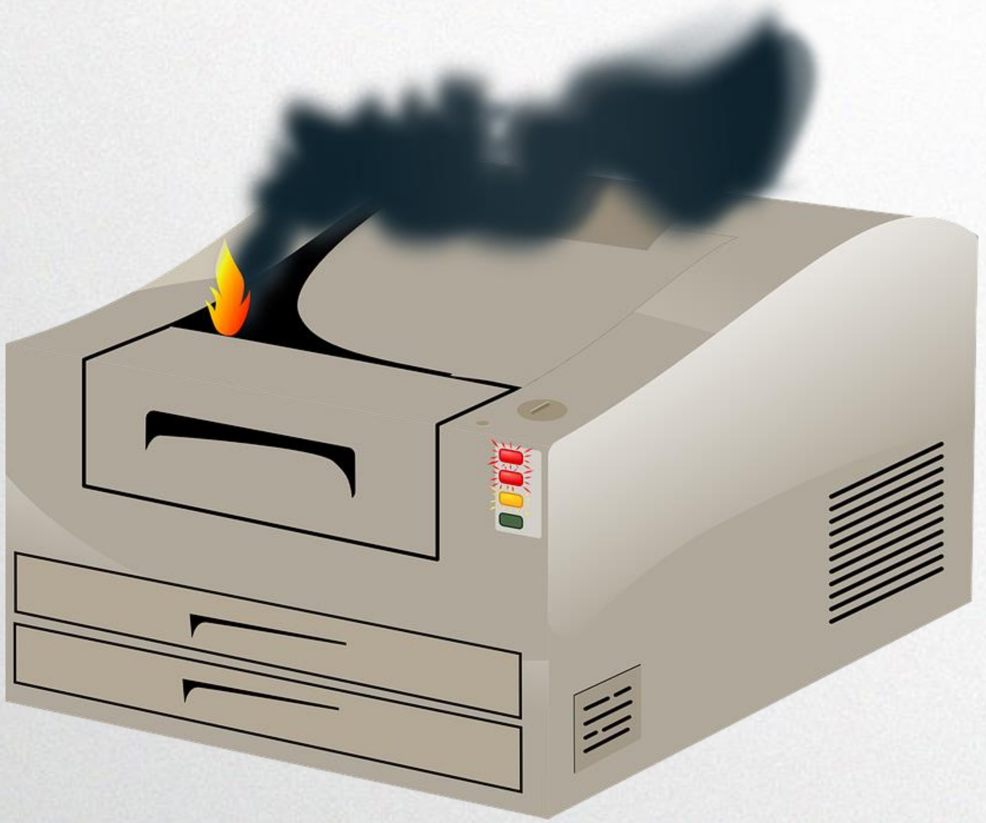
is the process of reducing the impact of run-time errors during the execution of an application

Error Handling in PHP

shawacademy

Sources of Errors

- Programming errors
- Invalid Input Data
  - Human error
  - Hacking attempts
- Transfer errors
  - electrical noise
  - hardware

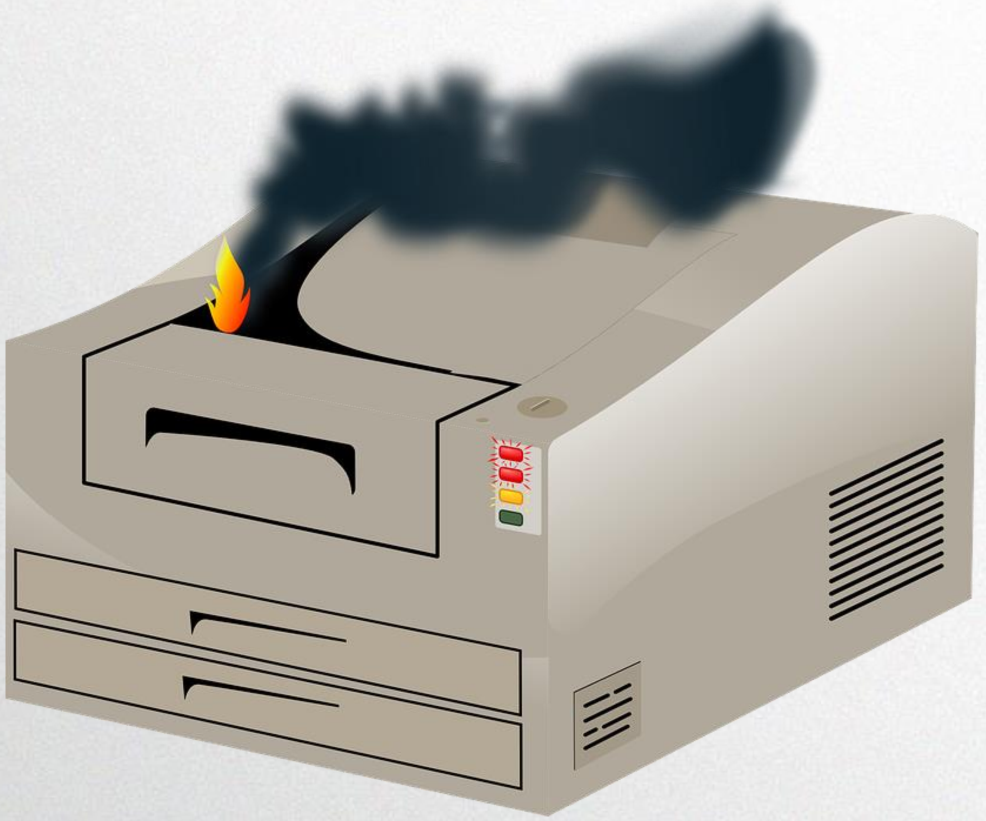


Error Handling in PHP

How to Error Handle?

1. Use “if” conditional loop
   - ➢ Check datatype
   - ➢ Check string contents
   - ➢ etc

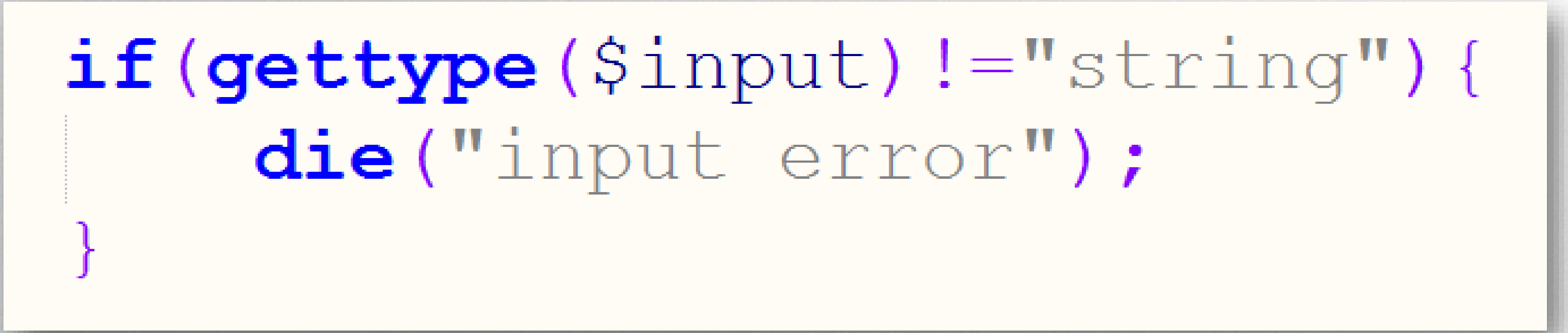
2. Perform a PHP error handle if true



Error Handling in PHP

Simple Error Handling

Ensure that “$input” is of datatype string:



```
if(gettype($input)!="string"){
      die("input error");
}
```

die() function is the default error handle function for PHP

1. Script is stopped
2. Argument is echoed to default error log

Next Week

shawacademy

PHP Development Semester

- ➢ The next session is "PHP & Security"
  - Encrypted Data & HTTPS
  - Data Validation with Hashing
  - Storing Passwords Securely

- ➢ Recordings are available within 24 hours after the live webinar
  - ➢ Go to www.shawacademy.com and then the Top Right Corner – Members Area

Q&A

shawacademy

Next Lesson is

Error Handling & Advanced Development

- Learn advanced processed used in building encapsulated, self-monitoring classes & applications
- You will understand the value of managing member visibility & the purpose of custom error handling

www.shawacademy.com

www.facebook.com/shawacademy

www.twitter.com/shawacademy

support@shawacademy.com