Professional Diploma in Commercial Web Design

Lesson 27a

Object - TV

By Raymond Tsang in Fevaworks

t-raymond.tsang@fevaworks.com

Objective

- Understand object in real life example
- Create TV object

What is object

- You can turn-on/ turn-off a TV
- If Sony is a TV, then you can turn-on/turn-off a Sony.
- TV is an object, turn-on/turn-off is a function.
- Everything you say is a TV, you can turn-on/turn-off.

Why do we need object

- We want to reuse functions, also want to reuse variables.
- Everything is stored in “class”
- You do not need to know how it works inside, but you need to know what to input and what is the output

Concept

- Class – The blueprints for an object and the actual code that defines the properties and methods
- Object – running instances of a class that contain all the internal data and state information needed for your application to function
- $object=new Class();
- tv01.php

Create an object and instance

- class Classname{} – define object
- $object_var=new classname(); - create object

Method

- Function in an object is called method.
- Methods can be created more than one
- tv02.php

Classwork

- Add a new turnOff method in TV class
- Turn off $sony
- tv03.php

Property

- Variable in an object is called property
- Properties can be declared with var $var;
- Set variable to the class by $this->var = value;
- tv04.php

Constructor

- Initialization in a class is called constructor. It is defined by creating a method that named function _ _constructor(){}
- tv05.php

Classwork

- Now Philips ask you to create a new TV for 32".
- Test turnOn, turnOff, getPlug methods
- Tell me the size of Philips.
- tv06.php

Classwork

- Your boss want to add a new “color” property for each TV.
- Both Sony is “black” and Philips is “silver”.
- Show us each TV color
- tv07.php

Inheritance

- Inheritance is based around the concept of parent classes and child classes
- When you create a child class, it inherits all the properties and methods of the parent. The child class can then include additional properties and methods, thereby extending the functionality of the parent class.
- Dragonball example
- EG. LCD is a kind of TV. LCD can do everything TV can do.

Inheritance

- class ChildClass extends ParentClass {
- }
- $samsung=new LCD();
- tvo8.php

Over parent

- Child class can have new methods which parent class does not have.
- EG. New iPhone has new function that more than old iPhone.
- tv09.php

Classwork

- Add dolbyOff method
- tv10.php

Parent method

- You may use methods from Parent class
- parent::turnOff()
- tv11.php

Parent property

- You may save/use property from Parent class
- tv12.php

public

- Class properties must be defined as public, private, or protected.
- By default, all class members are public . If properties declared using var, the property will be defined as public.
- public $plug="UK";
- public function timeroff(){}
- tv13.php

public

- You can change the value of any public property.
- $sony->plug="CN";
- tv14.php

private

- Access is limited to the declaring class only. No external access whatsoever is allowed.
- It is a good practice to protect from outsider giving invalid value. Always check input value before setting the new value.
- $sony->voltage=120; (Error)
- tv15.php

protected

- To access a parent method or property from a child class
- Like the private keyword, protected methods and properties are available only to the class that created them.
- tv16.php

protected

- But unlike private, protected methods and properties are visible from a parent class.
- SetMethod from child
- tv17.php

Static property

- Static means the method or variable is accessible through the class definition and not just through objects
- public static $madeIn="China";
- TV::$madeIn
- tv18.php

Static method

- Static means the method or variable is accessible through the class definition and not just through objects
- public static function insurance(){}
- TV::insurance();
- tv19.php

Concept 2

- Polymorphism – allows a class to be defined as being a member of more than one category of classes (EG. a car is "a thing with engine" and also "a thing with wheels")
- Interfaces – a way of specifying that an object is capable of doing something without actually defining how it is to be done (EG. a dog and a human are "things that walk" but they are different)
- Encapsulation – the ability of an object to protect access to its internal data

QUESTIONS