Professional Diploma in Commercial Web Design

Lesson 7a

Object - TV

By Raymond Tsang in Fevaworks

t-raymond.tsang@fevaworks.com

Objective

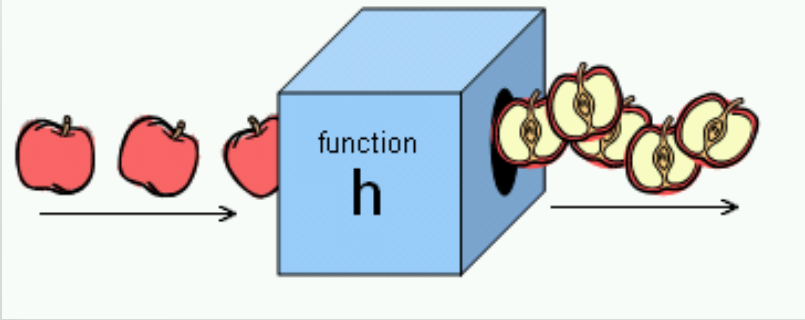
- Understand object in real life example
- Create TV object

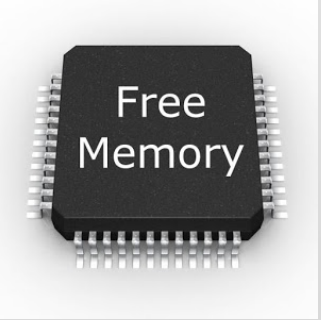
What is object

- 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.

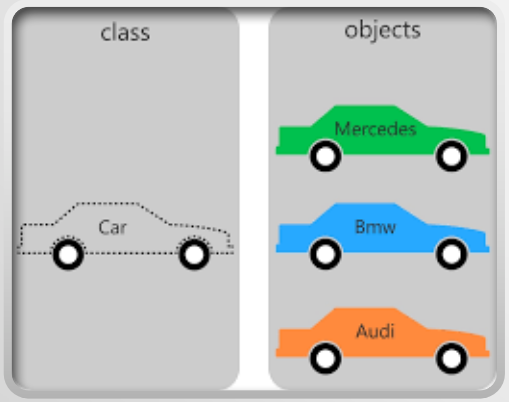
Why do we need object

- 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
- Free memory





Concept



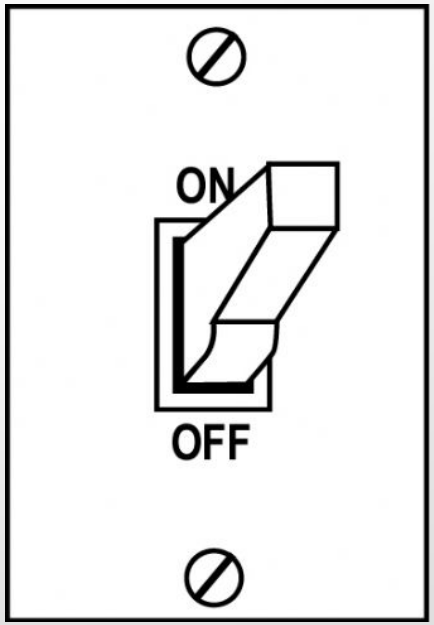
- Class – The blueprints for an object and the actual code that defines the properties and methods
- Object – running instances of a class that contain all the internal data and state information needed for your application to function
- $object=new Class();
- tv01.php

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



Method

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

Classwork

- Add a new volumeUp and volumeDown method in TV class
- echo “volume up
”;
- tv04.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;
- tv05.php

Constructor

- Initialization in a class is called constructor. It is defined by creating a method that named function _ _construct(){}
- We need to know the size before we make a TV.
- tv06.php

Constructor

- How to add 2 variables to constructor?
- We also need to know the brand name.
- tv07.php

Classwork

- Now Philips ask you to create a new TV for 32”.
- Test getName, turnOn, turnOff, volumeUp, volumeDown, getPlug methods
- Tell me the size of Philips.
- tv08.php

Classwork

- Your boss want to add a new “color” property for each TV.
- Before you create a new TV, you need to tell me what is the TV color.
- Sony is “black” and Philips is “silver”.
- Show us each TV color
- tv09.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 {
- }
- $child_object=new ChildClass();

Inheritance

- class LCD extends TV{
- }
- $samsung=new LCD();
- tv11.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.
- tv12.php

Classwork

- Add dolbyOff method
- $samsung->dolbyOff();
- tv13.php

Parent method

- You may use methods from Parent class
- parent::turnOff()
- tv14.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();
- tv15.php

Static method

- Different class can has some static method name but different function
- LCD::insurance();
- tv16.php

Public, Private, Protected

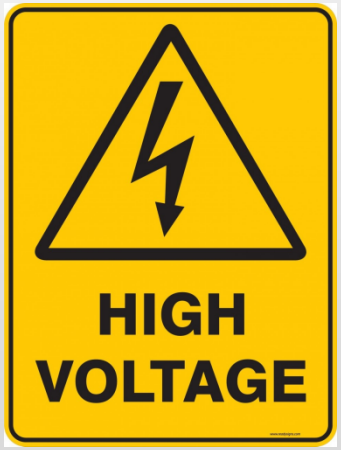
- Class properties must be defined as
  - Public
  - Private
  - Protected

public

- By default, all class members are public . If properties declared using var, the property will be defined as public.
- You can change public variable at anytime.
- public $plug="UK";
- $sony->plug="CN";
- tv17.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=110; (Error)
- tv18.php



Change private variable

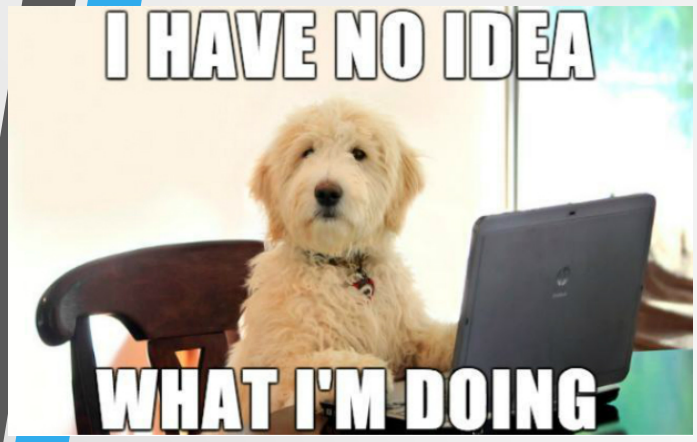
- Encapsulation – the ability of an object to protect access to its internal data
- getter and setter
- We need a setter to change the private variable
- tv19.php

Validate private variable

- Since we can validate the value before we change the private variable, private variable are more secure than public variable.
- tv20.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. But unlike private, protected methods and properties are visible from a parent class.
- SetMethod from child
- tv21.php



Concept 2

- 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)
- Abstract – similar to Interface, but it can contacts method content

Interface

- An interface is similar to a class except that it cannot contain code. An interface can define method names and arguments, but not the contents of the methods. Any classes implementing an interface must implement all methods defined by the interface. A class can implement multiple interfaces.
- Delete any method will cause error.
- tv22.php

Abstract

- An abstract class is a mix between an interface and a class. It can define functionality as well as interface (in the form of abstract methods). Classes extending an abstract class must implement all of the abstract methods defined in the abstract class.
- tv23.php

Concept 3

- 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”)

Example

- Search 農曆 php
- https://gist.github.com/eagleon/1702129
- cal.php

Example

- Search php pdf
- http://www.fpdf.org/
- pdf.php

References

- http://newaurora.pixnet.net/blog/post/190301967-static%E8%AE%8A%E6%95%B8%E3%80%81public%E8%AE%8A%E6%95%B8%E3%80%81private%E8%AE%8A%E6%95%B8%E3%80%81protected%E8%AE%8A

QUESTIONS