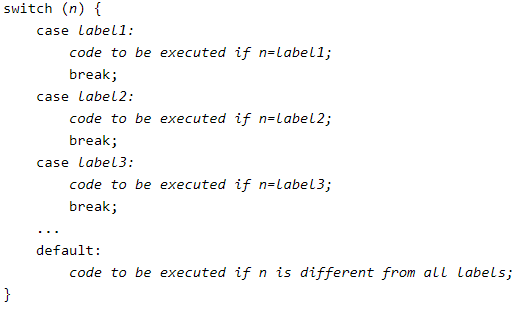
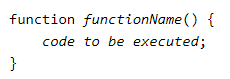
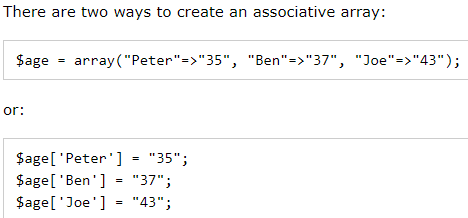
**PHP Basic**

* a php script start with <?php and ends with ?>
* // and # single line comment, /\* \*/ multiple line comment
* in PHP variable start with $ sign
* variable scopes: local, global and static
* outside function variable: global; inside function variable : local; inside function variable with static ahead: static
* inside a function to access global variable: global $x, $y
* Print value in PHP: echo and print
* echo: faster not return value
* print: slower and will return value
* PHP supports: String, Integer, Float, Boolean, Array, Object, Null, Resource
* define(name, value): create a constant (constant are global)
* if(condition){ code}else{code}:if-else statement
* **switch in PHP**
* PHP loops: while, do...while, for, foreach(for in python)
* PHP function:



* declare(strict\_types=1): avoid different type doing operation
* function functionname(var a, var b) : return type{} **：** the ": type" after function will define function return type
* Array: $var = array("0","1","2")
* count($var): get the length of an Array
* sort($var): sort array in Ascending order
* rsort($var): sort array in descending order
* asort($var): sort array according to value
* ksort($var): sort array according to key
* Associative Arrays



* Superglobals: $GLOBALS, $\_SERVER, $\_REQUEST, $\_POST,$\_GET
* Date: d(day), m(month), Y(year), l(day of the week) ex: data("Y-m-d l")
* Time: H(24 hours), h(12 hours), i(minutes), s(seconds),a(am or pm) ex: date("h:i:sa")
* Create a date: mktime(hour, minute, second, month, day, year)
* strtotime(time, now): convert a human readable date string into a Unix timestamp
* include and require : call the php function in the file
* Include always run the script even they cannot find the php file. Require will not execute the script if statement return error
* Cookie: used to identify a user
* setcookie(name, value, expire, path, domain, secure, httponly)
* setcookie(name, value, negative-expire): negative time will delete the cookie
* Sessions us to store information to be use across multi pages
* a session started with the session\_start() function
* session\_unset() and session\_destroy() will remove all global sessions
* Filter: Validating data(determine if the data is in proper form) and Sanitizing data(remove any illegal character form the data)
* filter\_var(): check the variable, the type of check to use
* JSON(JavaScript Object Notation): syntax for storing and exchanging data
* json\_encode(): encode a value to JSON format
* json\_decode(): decode a JSON object into PHP object or an associative array

**String**

* var\_dump: print variable type
* strlen(): return the length of the string
* str\_word\_count(): Count Words in a String
* strrev(): reverse a String
* strpos(string, word): search for word within a string
* str\_replace(text,word,string): replace text inside string with word

**File Handling**

* fopen():open/create a file (more operation that fread())
* fread(): read a file
* fclose(): close a file
* fgets(): read single line
* feof(): check the end of file
* fgetc(): read single character form the file
* fwrite($file,$txt): write to a file
* php.ini configure file setting: to use file upload, need add file\_uploads = On inside php.ini

**OOP**

* Class: Biosphere Object: animals
* Class getter and setter, object input detail value
* $this->name = $name : this keyword refers to current obj
* \_\_destruct() function: php will auto call this function at the end of the script
* \_\_construct() function: php will auto call when you create an object
* public: can be accessed from everywhere (default)
* protected: can be accessed within the class and subclasses
* private: can ONLY accessed within the class
* Inheritance: parent class and child class, child class extends parent class
* Overriding method: child class can have same method in parent class
* final keyword: prevent method overriding
* const: constants cannot be change once it is declared
* Abstract: parent class has a named method but empty inside, need its child class to fill out the tasks
* PHP only supports single inheritance: a child class cna inherit only from one single parent
* trait keyword: used to declare methods that can be used in multiple classes and can have multi-inheritance
* static methods and properties: can be called directly without creating an instance of a class

7.0-7.4 update

* closure: function can be created without a specified name, most important use in callback function
* $closure->call(): Binds and calls the closure
* random\_bytes(int $length) //output string: generates cryptographically secure pseudo-random bytes
* random\_int(int $min, int $max) //output int : generates cryptographically secure pseudo-random integers (random in min and max range)
* error\_clear\_last(viod) //output void : clear the most recent error
* Generator->getReturn(void): get the return value of a generator
* intdiv (int $dividend, int $divisor) //output int : Integer division, return the integer quptient of the division of dividend by divisor
* public static Closure::fromCallable (callable $callable) //output Closure : converts a callable into a closure
* public DOMNodeList::count(void) //output int : get number of nodes in the list
* public DOMNodeMap::count(void) //output int : get number of nodes in the map
* hash\_hmac\_algos(void) //output array: return a list of registered hashing algorithms suitable for hash\_hmac
* public static DateTime::createFromImmutable (DateTimeImmutable $ datetime) //output DateTime : return new DateTime object encapsulating the given DateTimeImmutable object

PHP manages variable memory

* if the argument is passed by value, which means a copy of variable is made which will take more memories
* Zend Engine using a C struct, zval, to store the value of variable. zval.type stores the variable type, zval.value stores the variable value
* In zval, if we cannot find the field to store the name of a variable, then PHP store the name of a variable in a hash table, call symbol\_table and then mapping the name with value
* in Zend Engine:



will do create a variable (name) referring to zval ('henry')

create another variable (fname) referring to zval ('henry')

PHP decrease the number of refcount just not let name not referring to 'henry'

* THIS EXCAMPLE SHOW UNSET CANNOT FREE MEMORY
* PHP have a copy and write mechanism: if PHP going to change a variable. 1. check zval.refcount, if zval.refcount >1 create a new zval and decrease the old zval.refcount by 1. 2. modify the symbol\_table make variables to referring to the new zval (here PHP must allocate new memory) 3. if we unset old variable we can really save some memory
* If two objects point to each other, after you unset() two variables , the object will keep stay and not free in memory. Now need to use gc\_collect\_cycles() to free the circular reference object