**Top 50 Functions**

|  |
| --- |
|  |

# **: introduction:**

* **Repository:**

|  |  |  |  |
| --- | --- | --- | --- |
| **repository** | **type** | **description** | **path** |
| **uclib** | **Required** | **This folder contains all common services and functions** | **root/uclib** |
| **pe-gold3** | **Required** | **This is a frontend folder this function need to be run.** | **root/pe-gold3** |
| **pe-gold3-ux** | **Required** | **This folder contains all common CSS and JS.** | **pe-gold3/layout/thems/bootstrap4/ux** |
| **pe-items** | **Required** | **This folder contains items file.** | **pe-gold3/layout/thems/bootstrap4/pe-items** |
| **pe-sim** | **Required** | **This folder contains PHP file of sim area** | **Pe-gold3/sim** |
| **pe-sim/bootstrap3** | **required** | **This folder contains all tpl file and sim area.** | **pe-gold3/layout/thems/bootstrap4/sim** |
| **preEngineAPI** | **optional** | **This folder contains API files.** | **preEngineAPI** |
| **Pe-gold-native** | **Optional** | **Mobile app file.** | **Pe-gold-native** |

# **: functions:**

* **uclib folder:**

1. **isValidEmail (): to check if email valid or not.**

**Param: $email, $ignore\_number=false,**

**Ex:**

**$email =** [**dravinanshu98@gmail.com**](mailto:dravinanshu98@gmail.com)

**if(isValidEmail($email)==true){**

**}**

**return: boolean.**

1. **sanitizeEmail (): it is used to remove all characters except latter’s, digits, and !#$%&”, %,.-?{[]}-@[].**

**Param: $email**

**Ex:**

**$email =** [**dravinanshu98@gmail.com**](mailto:dravinanshu98@gmail.com)

**echo sanitizeEmail($email);**

1. **gen\_uuid (): it is used to generates unique alphanumeric code.**

**Param: $len=8, $salt=”ucertify”**

**Ex:**

**gen\_uuid(5);**

**output: g5Dud.**

1. **dateDiff (): print interval between 2 dates.**

**Param: $interval, $datefrom, $using\_timestamps=false**

**Ex:**

**$dateform = “2023-01-01”;**

**$dateto = “2023-02-01”;**

**$interval = d;**

**dateDiff ($dateform, $dateto, $interval):**

**format: $interval can be.**

**yyy: number of full years.**

**q: number of full quatre.**

**m: number of full months.**

**y: difference between day numbers.**

**(eg: 1st jan 2004 is “1”, the first day 2nd feb 2003 is “33” the default is “-32”)**

**d: number of full days.**

**w: number of full weekdays.**

**ww: number of full weeks.**

**h: number of full hours.**

**n: number of full minutes.**

**s: number of full seconds (default)**

1. **xml2array (): it is used to converts xml to array.**

**Param: $xml.**

**Ex:**

**xml2array($xml)**

1. **array2xml (): it is used to converts array to xml need to pass main tag name.**

**param: $mainTag, $arr,**

**ex:**

**array2xml (array2xml, $arr);**

1. **\_ucd (): it is used to print and exist from the page.**

**Param: $a, $isDie=1, $isPre=1**

**Ex:**

**\_ucd($a,0);**

1. **ucIsset (): ucIsset is used to check set or not.**

**param: $v, $val=false,**

**ex:**

**if(ucIsset($config[‘move\_object’])) {…….}**

**return: boolean,**

1. **array\_merge\_uc (): it is used to merge two arrays also 3rd parameter is for multidimensional array merge.**

**Param: $arr1, $arr2, $recursive=false;**

**Ex:**

**array\_merge\_uc ($arr1, $arr2, true);**

1. **array\_keys\_uc (): it is return array keys array.**

**Param: $arr**

**Ex:**

**array\_keys\_uc ($arr);**

**return: array.**

1. **array\_unique\_uc (): return the unique values (remove duplicate).**

**Param: $arr.**

**Ex:**

**array\_unique\_uc ($arr);**

1. **friendlyDate(): return human readable date.**

**Param: $time.**

**Ex:**

**$time = strtotime(“2023-01-01”);**

**echo friendlyDate();**

**output: 1**

1. **friendlyguid (): it converts some matching char to another char like l, 1,o, 0, Q to, L, L, X, U, X.**

**Param: $guid.**

**Ex:**

**$guid = friendlyguid ();**

**Output: IXoul**

1. **getAttribute (): get attribute from HTML.**

**Param: $strmle, $tag, $attribute, & $i.**

**Ex:**

**$i = 0;**

**$strmle = “<p>this is </p><player type=’video’ subtype=’video\_plus’/>”;**

**$tag = “player”;**

**$attribute = “type”;**

**echo getAttribute ($strmle, $tag, $attribute,$i);**

1. **removecomments (): remove comments from the string.**
2. **redirect\_header (): used to redirect header to messages.**
3. **trimWhiteSpace (): trim white spaces remove CDATA from the xml values.**
4. **removeCDATA ():**
5. **remfiles ():**
6. **showpage (): showpage () method create header footer and all common file automatically when we call.**

**Ex: JS file, CSS file.**

**Examples:**

**Param: $content, $heading=””, $subheading=””, $themepage=””, prepangin.tpl=””, config=array()**

**Showpage($content);**

1. **displaycontent (): displaycontent method () print the page content and exit from the page with the help of ajax call.**

**Ex:**

**Param: $content**

**displaycontent($content)**

1. **getStructure (): getStructure () method is used to print the course structure along with snippets.**

**EX:**

**Param: $coverage, $seq2coverage, $structure\_content, $level= 1, $ddl=false, $onlysnippet=false, $visible=1, $content\_type=”s” $parent\_guid=false.**

**getStructure($coverage, $seq2coverage, $structure\_content);**

1. **getContentsGUIDs (): getContentsGUIDs () function is used to find the any list of course specific GUIDs.**

**Param: $coverage, $seq2coverage, $content\_type=”q,u,f”, $content\_subtype=false, $parent\_guid=false, $level=false, $demo=-1, $visible=1, $testnos=-1**

**Example:**

**getContentsGUIDs($coverage, $seq2coverage, $content\_type=”q”);**

**returns : content guid array.**

1. **get\_course\_info (): get\_course\_info () method is used to print the course all information along with coverage and also store in cache.**

**Param: $course\_code, $class\_code=false, $cache=true, $strore\_in\_session=true;**

**Ex:**

**Get\_course\_info($course\_code);**

**Return: course info (with config and details)**

1. **find\_course\_code (): find\_course\_code () method is used to find the course code.**

**Param: $where, $type=false**

**Ex:**

**Find\_course\_code($where);**

**Return: course data array.**

1. **get\_character\_no (): get\_character\_no method () is used to find the chapter number with guid and coverage and sec2coverage.**

**Param: $guid, $coverage, $seq2coverage**

**Return: chapter no.**

**Ex:**

**get\_character\_no ($guid, $coverage, $seq2coverage**

**):**

1. **guid2seqNo (): guid2seqNo () method is used to get the sequence number of the guid int the course friendly number.**

**Ex:**

**Param: $coverage, $seq2coverage, $g, $type, $level, $visible=1, $chapter\_seq=false**

**guid2seqNo ($coverage, $seq2coverage,0,”s”,”1”);**

**return: sequence number**

1. **seqNo2guid (): seqNo2guid () method is used to get content guid based content no.**

**Param: $coverage, $seq2coverage, $s, $type, $level**

**Return: content guid.**

**Ex:**

**seqNo2guid ($coverage, $seq2coverage, $s, $type, $level):**

1. **checkPermission (): checkPermission () method is used to create permission of array of logged user based on course.**

**Param: $cp=”\*”,$up=”\*”**

**Ex: checkPermission ():**

**Return permission array.**

1. **class clsVisibility (): class clsVisibility () method this method to check content visibility.**

**Param:**

**Ex:**

**$clsobj = new ClsVisibility();**

**$clsobj->isVisible($vdata);**

1. **is\_lab\_mode (): is\_lab\_mode () method is used to check loaded course lab and which type of lab (lab only, lab with courses).**

**Param: $lic\_available = array(), test=false, $isLabOnlyPrep = false**

**Ex:**

**is\_lab\_mode ();**

**return: 0,1,2**

1. **is\_lab\_available (): is\_lab\_available method is used to checks whether lab is available or not in the loaded courses.**

**Param: $test\_stats=array(), $lab\_stats=array()**

**Return: boolean**

**Ex:**

**is\_lab\_available ();**

1. **is\_virtual\_lab\_available (): is\_virtual\_lab\_available () method is used to checks if virtual course available in the loaded course.**

**Param: $test\_stats = array(), $hlab\_stats = array (), $v=0**

**Ex:**

**is\_virtual\_lab\_available ():**

**return: boolean**

1. **is\_live\_lab\_mod (): is\_live\_lab\_mod () is used to check the live lab (ucertify lab, practice lab, simulator lab).**

**Param: $course\_data = array()**

**Ex:**

**is\_live\_lab\_mod ()**

**return: 0,1,2**

1. **is\_video\_only\_mode (): is\_video\_only\_mode () is used to check the course is video only prepkit.**

**Param: $lic\_available**

**Ex:**

**is\_video\_only\_mode ():**

**return: boolean.**

1. **is\_video\_lic\_only (): is\_video\_lic\_only () is sued to checks the video license is available in the loaded course.**

**Param: $lic\_available**

**Ex:**

**is\_video\_lic\_only ()**

**return: boolean.**

1. **extractLicence (): extractLicence () is used to extract product license in the text form.**

**Param: $license, $is\_byte=true, $full=false.**

**Ex:**

**$license = 30;**

**extractLicence (license);**

1. **extractCoursePermission (): extractCoursePermission () is used to converts str to hex value.**

**Param: $permissions, $full=false.**

**Ex:**

**$permission = 1**

**extractCoursePermission ($permission);**

1. **show404 (): it is used to show 404 page error.**

**Ex:**

**Show404 ();**

**Return: void.**

1. **removeOptionalLicense (): it is used to remove Optional License like virtual lab, video, monitoring etc.**

**param: $lic, $montoring=0, $virtual\_lab=0;**

**ex:**

**removeOptionalLicense ($lic);**

**return: license array.**

1. **isConfigOn (): it is used to check if var is set in the config.**

**Param: $config, $configName**

**Ex:**

**isConfigOn ($config);**

**return: boolean.**

1. **arr2Url (): it is used to convert array to URL string.**

**Parameter: $arr, $baseUrl = SITE\_URL**

**Ex:**

**arr2Url ($arr);**

**return: URL string.**

1. **set\_meta\_title (): it is used to assign meta title.**

**Param: $page=””, $title=””, $assign=true**

**Ex:**

**set\_meta\_title (“”, “#course\_name”);**

**return: void / meta array.**

1. **ip\_get (): it is used to get IP address.**

**Ex:**

**ip\_get();**

**Return: string.**

1. **convertByTimezone (): it is used to converts time by time zone.**

**Param: $date, $timezone**

**Ex:**

**convertByTimezone ($date, $timezone);**

**return: stringTime.**

1. **checkPhoneNo (): it is used to check the verify the phone number.**

**Param: $str,**

**Ex:**

**checkPhoneNo (“9876765456”);**

**return: boolean.**

1. **getSessionName (): it is used to get test session name.**

**param: $test=false.**

**Ex:**

**getSessionName ();**

**return: string.**

1. **makeDdlAndSorting (): it is used to create DDL and do sorting.**

**Param: $user\_course\_list, $key, $value.**

**Ex:**

**makeDdlAndSorting ($user\_course\_list, $key, $value);**

1. **FrameworkLogSet (): it is used to save Error log.**

**Param: $data, $error\_type, $user\_guid=””, debug\_trace=false,**

**Ex:**

**FrameworkLogSet ($data, 1);**

1. **array\_reorder (): it is used to reorder an array.**

**Parameter: $array, $oldIndex, $newIndex.**

**Ex:**

**array\_reorder(array, “index1”, “index2”);**

1. **multi\_array\_sort (): it is used to multiple dimensional array sort.**

**Param: &$data, $key, $reverse\_sort=false.**

**Ex:**

**multi\_array\_sort ($arr, $key):**

**return: array**

1. **seconds2human (): it is used to converts time to human readable form.**

**param: $second,**

**ex:**

**seconds2human ($time);**

**output: 10h, 5m, 2s.**

1. **extrcatArray (): it is used to create where array within array.**

**Param: $arr, $Indexes=””, $ignore\_blank=false.**

**Ex:**

**seconds2human ($arr, “couse\_code, user\_guid, name, email”);**

**return: array**

1. **byte2array (): it is used to convert byte to array.**

**Param: $byte, $defarray, $len=1**

**Ex:**

**byte2array (30, $\_product\_permission\_byte);**

**Return: associative array.**

1. **array2byte (): it is used to convert array to byte.**

**Param: $option, $defarray, $len=1**

**Ex:**

**array2byte ($arr, $\_product\_permission\_byte);**

**Return: associative array.**

1. **is\_valid\_XML(): it is used to check valid XML.**

**Param: $xml.**

**Ex:**

**is\_valid\_XML($xml);**

1. **uclsArray (): it is used to check if it is an array.**

**Param: $arr.**

**Ex:**

**uclsArray ($arr);**

**return: boolean.**

1. **datetimeDiff (): it is used to get difference between two dates.**

**Param: $dt1, $dt2.**

**Ex:**

**datetimeDiff ($dt1, $dt2);**