

```
<code php> <? /* * @author KAD * Скрипт парсинга v.5 * * - Добавлено кэширование страниц
сайта-донора * - Функции для конфигурации вынесены вверх файла * - Экспорт теперь
осуществляется в виде отдельных файлов */
```

```
@set_time_limit(90000); require_once('bootstrap.php'); @session_start(); Идентификатор сайта
$site_id = 2;
```

```
URL донора $url = "http://videoglaz.ru"; Количество элементов за шаг $stepCnt = 50; $curr =
"руб.";
```

```
Кодировки (если пустые, то строка парсинга обрабатывается стандартным способом)
$encryption_from = "CP1251"; $encryption_from = ""; $encryption_to = "UTF-8";
```

```
Массив соответствий для доп. свойства prop-69 где 69 это id свойства $arr_conf = array(
```

```
"shop_groups_value",
"shop_groups_value",
"shop_groups_value",
"shop_groups_value",
"shop_groups_value",

"shop_items_catalog_marking",
"shop_items_catalog_name",
"shop_items_catalog_text",
"shop_items_catalog_description",
"shop_producers_list_value",
"shop_items_catalog_image",
```

```
);
```

```
=====
=====
```

Функции для заточки под сайт

```
=====
===== /* * создает лист
ссылок каталога */ function createParseList() { global $url, $delim; $out = ""; $groupsList = array(
'/catalog.php?id=1419' => 'Безопасность здоровья',
```

```
    '/catalog.php?id=118' => 'Системы охранно-пожарной сигнализации',
    '/catalog.php?id=816' => 'Системы видеонаблюдения',
    '/catalog.php?id=1160' => 'Модули порошкового пожаротушения',
    '/videoregistrator-avtomobilnyi' => 'Автомобильные видеорегистраторы',
    '/catalog.php?id=1193' => 'Комплекты беспроводной GSM-сигнализации',
    '/catalog.php?id=1189' => 'Комплекты охранной сигнализации (проводные)',
    '/catalog.php?id=299' => 'Беспроводная GSM-сигнализация',
    '/catalog.php?id=717' => 'Радиоканальные системы',
    '/catalog.php?id=729' => 'Системы охраны периметра',
    '/catalog.php?id=316' => 'Извещатели охранные для помещений',
    '/catalog.php?id=531' => 'Поворотные Wi-Fi-камеры'
```

[illegible]

подгруппы";

```
$groups4Data->clear();  
unset($groups4Data);  
}
```

```
0);  
$items4data = $data4->find('.goodtable',  
if(isset($items4data))  
{  
    $pref = $group . "~" . $group1 .  
    "~" . $group2 . "~" . $group3 . "~" . $group4 . "~";  
    $parseList[] = array( 'href' =>  
$href4, 'pref' => $pref);  
    //echo "<br/>есть блок товаров в  
groups4Data" . $url . $href4;  
    $items4data->clear();  
    unset($items4data);  
}  
$data4->clear();  
unset($data4);  
}  
$groups3Data->clear();  
unset($groups3Data);  
}
```

```
$items3data = $data3->find('.goodtable', 0);  
if(isset($items3data))  
{  
    $pref = $group . "~" . $group1 . "~" .  
$group2 . "~" . $group3 . "~" . "~";  
    $parseList[] = array( 'href' => $href3,  
'pref' => $pref);  
    //echo "<br/>есть блок товаров в groups3Data" .  
$url . $href3;  
    $items3data->clear();  
    unset($items3data);  
}  
  
$data3->clear();  
unset($data3);  
}  
$groups2Data->clear();  
unset($groups2Data);  
}
```

```
$items2data = $data2->find('.goodtable', 0);  
if(isset($items2data))  
{  
    $pref = $group . "~" . $group1 . "~" . $group2 .  
    "~" . "~" . "~";  
    $parseList[] = array( 'href' => $href2, 'pref' =>
```

```

$pref);
                                //echo "<br/>есть блок товаров в groups2Data" . $url .
$href2;\
                                $items2data->clear();
                                unset($items2data);
                                }

```

```

                                $data2->clear();
                                unset($data2);
                                }
                                $groups1Data->clear();
                                unset($groups1Data);
                                }

```

```

$items1data = $data1->find('.goodtable', 0);
if(isset($items1data))
{
    $pref = $group . "~" . $group1 . "~" . "~" . "~" . "~";
    $parseList[] = array( 'href' => $href1, 'pref' => $pref);
    //echo "<br/>есть блок товаров в groups1Data" . $url . $href1;
    $items1data->clear();
    unset($items1data);
}

```

```

    $data1->clear();
    unset($data1);
}
$groupsData->clear();
unset($groupsData);
}

```

```

$itemsData = $data->find('.goodtable', 0);
if(isset($itemsData))
{
    $pref = $group . "~" . "~" . "~" . "~" . "~";
    $parseList[] = array( 'href' => $key, 'pref' => $pref);
    //echo "<br/>есть блок товаров в groupsData" . $url . $key;
    $itemsData->clear();
    unset($itemsData);
}

```

```

    $data->clear();
    unset($data);
}
echo "Парслист готов: " . count($parseList) . " страниц";
uLog("Parse list created!");
return array('parselist' => $parseList, 'out' => $out);

```

```

}

```

/* * парсим со страницы каталога товаров, когда в подразделе произвольное число страниц */

```
function loadPage($pref, $href, $start) {
```

```
    global $url, $delim, $stepCnt, $imgDir;
```

```
    $i = 0;
    uLog("Loading page... {$pref} -> {$url}{$href}");
    $cOut = "";
    $aItems = array();
    $errcnt = 0;
    $end = false;
    $finPage = (isset($_SESSION['finPage']))?$_SESSION['finPage']:0;
    $page_url = $url . $href;
    if ($start != 0)
    {
        $page = $start+1;
        $page_url .= "&page=" . $page;
    }
    echo $page_url."<br>";
    //$page_url = 'http://zoo-galereya.ru/category/shlejki-dlja-sobak/';
    $cData = loadCachePage($page_url);
    if($start == 0)
    {
        $pagination = $cData->find('td.nav div.pages', 0);
        if(isset($pagination))
        {
            $pagination = $pagination->find('a');
```

```

            $out = $pagination[count($pagination)-2]->plaintext;
            $out = str_replace("\n", "", str_replace("\r", "",
str_replace(';', ' ', $out)));
```

```

            $finPage = intval($out);
            $_SESSION['finPage'] = $finPage;
        }
        else
        {
            $_SESSION['finPage'] = 0;
        }
    }
    $block = $cData->find("table.goodtable", 0);
    //$rows = $block->find("tr");
    //берем со всей страницы блоки tr
    $rows = $cData->find("tr");
    foreach ($rows as $row)
    {
        // Разбор товаров
        $itemHref = getHref('a', $row);
        //если блок tr содержит ссылку с good то это товар
        $strpos = strpos($itemHref,"good.php?id=");
        if ($itemHref==' ' || $itemHref=='#' || $strpos===FALSE){
            continue;
```

```

    }
    // идем на страницу товара
    $itemHref = $url."/".$itemHref;
    uLog("Try to load first time... {$itemHref}");
    $iData = loadItem($itemHref, $pref);
    if (!$iData) //исправляет проблему с быстрой загрузкой ошибочной страницы
    {
        usleep(0.5 * 1000000);
        uLog("Try to load again... {$itemHref}");
        $iData = loadItem($itemHref, $pref);
    }
    if (!$iData)
    {
        $errcnt++;
        error("Error: {$itemHref}");
    } else
    {
        $cOut .= $iData;
    }
}

```

```

$cData->clear();
unset($cData);

if($finPage != 0 && $page < $finPage)
{
    $end = false;
}
else
{
    $end = true;
}

```

```

/*if ($start*$stepCnt >= count($rows))
{
    $end = true;
}*/

$return_arr = array('out' => $cOut, 'end' => $end, 'err' => $errcnt);
return $return_arr;

```

```

}

```

```

/* парсим со страницы товара */ function loadItem($href, $pref/*, $aProps*/) {

```

```

global $imgDir, $curr, $url;
$pData = loadCachePage( $href );

// Тут была проблема, если убрать трим, то все работает
//if ($pData && trim($pData) != "")
if ($pData)

```

```

{
    $pData = $pData->find('.context', 0);
    //имя
    $name = $pData->find('h1.pageHeader', 0);
    if(isset($name))
    {
        $name = $name->innertext;
        $name = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsaquo;', ' ', str_replace('~', '-', $name))))));
    }
    else
    {
        //echo "не нашли блок с именем";
        $name = '';
    }
}

```

echo "
Название: ".\$name; краткое описание

```

$description = '';
$descr_mass = $pData->find('p');
foreach ($descr_mass as $descr_elem) {
    if (strpos($descr_elem->prev_sibling(), "Описание")) {
        $descr = $descr_elem;
    }
}
if(isset($descr))
{
    $elements = $descr->find('h2');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('a');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('table');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('div');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $descr = $descr->innertext;
    $descr = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',

```

```

'.', str_replace('&rsquo;', ' ', str_replace('~', '-', $descr))))));
}
else
{
    //echo "не нашли блок с кратким описанием";
    $descr = '';
}

```

*echo "
Краткое описание: ".\$descr; полное описание*

```

$text_mass = $pData->find('ul');
foreach ($text_mass as $text_elem) {
    if (strstr($text_elem->prev_sibling(), "Технические характеристики")) {
        $text = $text_elem;
    }
}
if(isset($text))
{
    $elements = $text->find('p');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $text = $text->innertext;
    $text = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsquo;', ' ', str_replace('~', '-', $text))))));
}
else
{
    $text = '';
}

```

*echo "
Полное описание: ".\$text; артикул*

```

$art = $pData->find('.sertext', 0);
if(isset($art))
{
    $art = $art->innertext;
    $art = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsquo;', ' ', str_replace('~', '-', $art))))));
    $art = substr($art, 11);
}
else
{
    //echo "не нашли блок с артикулом";
    $art = '';
}

```

*echo "
Артикул: ".\$art; производитель*


```

$producer = $pData->find('td.desc div a', 0);
if(isset($art))
{
    $producer = $producer->innertext;
    $producer = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsaquo;', ' ', str_replace('~', '-', $producer))))))));
}
else
{
    //echo "не нашли блок с производителем";
    $producer = '';
}

```

`echo "<br>Производитель: ".$producer; изображение`

```

$img_href = $pData->find('a.fancybox img',0);
$img_href = $img_href->getAttribute('src');
if(isset($img_href))
{
    $img = saveImg($img_href);
}
else
{
    //echo "не нашли блок с производителем";
    $img='';
}

```

`echo "<br>Изображение: ".$img; дополнительные свойства`

```

$property_mass = $pData->find('table.bbt');
foreach ($property_mass as $property_elem)
{
    if (strstr($property_elem->prev_sibling(), 'Параметры'))
    {
        $property = $property_elem;
    }
}
if(isset($property))
{
    $property_setes = $property->find('tr');
    foreach ($property_setes as $property_set)
    {
        $id = $property_set->find('td',0)->innertext;
        $val = $property_set->find('b',0)->innertext;
        if ($val != "-"){
            setPropertiesValues($id, $val, $art);
        }
    }
}
}

```

```

else
{
    $property = '';
}

```

```

// Модификации
$sMod = "";
uLog("Item loaded!<br>");
$name = trim($name);
$pData->Clear();
unset($pData);

```

```

    return $pref . csvStr(array($art, $name, $text, $descr, $producer,
$img)) . $sMod;
} else
{
    return false;
}
exit();

```

```

}

```

/ Извлекаем информацию о группах, для размещения в конце файла */ function loadGroup(\$pref, \$cData) {*

```

$groupDescr = getHTML("div.category-top-text p", $cData, 1);

uLog("Group info loaded!");

return $pref . csvStr(array('', '', '', '', '', '', '', '', '', '', '',
$groupDescr));

```

```

}

```

```

=====
=====

```

Функции движка

```

=====
===== Разделитель для
CSV $delim = "~";

```

*Директория изображений \$imgDir = 'parse/imgs/'; Директория файлов \$fileDir = 'parse';
Директория кэширования \$cacheDir = 'parsecache'; Имя временного файла \$tmpltemName =
CMS_FOLDER . \$fileDir . DIRECTORY_SEPARATOR . "export"; Имя лог файла \$logFile = "log.txt"; Имя
файла ошибок \$errorFile = "errors.txt"; Имя конфиг файла \$configFile = CMS_FOLDER . \$fileDir .
DIRECTORY_SEPARATOR . "config.inf"; \$log = ""; \$errors = ""; define('CURRENT_SITE', \$site_id);*

```

$oSite = Core_Entity::factory('Site', CURRENT_SITE); Core::initConstants($oSite); include
'simple_html_dom.php'; $start = Core_Array::getPost('start'); if (Core_Array::getPost('cancel')) { if
(isset($_SESSION)) { parseReset(); } } if (isset($_GET['step'])) { $_SESSION['currentStep'] =
$_GET['step']; $_SESSION['itemStart'] = 0; unset($_SESSION['altems']); } if (isset($_GET['items'])) {
$_SESSION['itemscount'] = $_GET['items']; } ?> <!DOCTYPE html PUBLIC "-//W3CDTD XHTML 1.0
StrictEN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <head> <meta content="text/html;
charset=UTF-8" http-equiv="Content-Type"></meta> </head> <form name="parse"
method='POST' action='/parse.php'> <?

```

```

if (!$start) {

```

```

    if (isset($_SESSION['parseList']))
    {?>
        <input type='submit' name='cancel' value ='Сбросить'><br>
        <input type="submit" name="start" value="Продолжить парсинг"/>
    {?} else {?>
        <input type="submit" name="start" value="Начать парсинг"/>
    {?
    }

```

```

}

```

```

function parseReset() {

```

```

    global $imgDir, $fileDir;

```

```

    unset($_SESSION['parseList']);
    unset($_SESSION['currentStep']);
    unset($_SESSION['aItems']);
    unset($_SESSION['properties']);
    unset($_SESSION['config']);
    unset($_SESSION['finPage']);
    unset($_SESSION['itemStart']);

    //clearDir($fileDir . DIRECTORY_SEPARATOR);

    $_SESSION['producer'] = "";

```

```

}

```

```

function delim($str = ) { global $delim; return $str . $delim; } function
endl($str = ) {

```

```

    return $str . "\n";

```

```

}

```

```

function csvStr(array $array) {

```

```

    global $arr_conf;

```

```

$out = "";
foreach ($arr_conf as $key => $value)
{
    $out .= delim(isset($array[$key]) ? $array[$key] : "");
}

return endl($out);

```

```

}

```

```

/* * получаем html элемента */ function getHTML($pattern, $data, $num = 0) {

```

```

if ($out = $data->find($pattern, $num))
{
    $out = $out->innertext;
    //$out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' '
    ', str_replace('"', '\"', $out))));
    $out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' ',
    str_replace('&nbsp;', ' ', $out))));

```

```

    return $out;
}

```

```

}

```

```

/* * получаем значение элемента */ function getValue($pattern, $data, $num = 0) {

```

```

if ($out = $data->find($pattern, $num))
{
    $out = $out->plaintext;
    //$out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' '
    ', str_replace('"', '\"', $out))));
    $out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' ',
    $out)));
    return $out;
}

```

```

}

```

```

/* * получаем ссылку элемента */ function getHref($pattern, $data, $num = 0) {

```

```

if ($out = $data->find($pattern, $num))
{
    $out = $out->href;
    $out = str_replace("\n", "", str_replace("\r", "", str_replace('&',
    '&', $out)));
    return $out;
}

```

```

}

```

```
/* * получаем source элемента */ function getSrc($pattern, $data) {
```

```
if ($out = $data->find($pattern,0))
{
    $out = $out->src;
    $out = str_replace("\n", "", str_replace("\r", "", $out));
```

```
    return $out;
}
```

```
}
```

```
function saveImg($img_h, $img_l = false) {
```

```
global $imgDir, $url;
```

```
    $img = '';
    //uLog( $url . $img_h );

    if ($img_h)
    {
        $img = substr_replace($img_h, ' ', strrpos($img_h, '/') , 1);
        $img = basename($img);
        // Если изображение не было скачено ранее
        if (!is_file(CMS_FOLDER . $imgDir . $img))
        {
            $oImage = @file_get_contents( $url . $img_h);
            if (!$oImage && $img_l)
            {
                $oImage = @file_get_contents( $url . $img_l);
            }
            if ($oImage)
            {
                file_put_contents( CMS_FOLDER . $imgDir . $img , $oImage);
            }
        }
    }
    return $img;
```

```
}
```

```
/* Обработчики доп. свойств ===== */ lib $aProperties = array(); $aPropertiesIdentified = array();
$aPropertiesValues = array(); Поместить значение в массив, если такого нет function insert($arr,
$value) {
```

```
if (!in_array($value, $arr))
{
    $arr[] = $value;
}
return $arr;
```

```
}
```

user Добавляем доп. свойства и значения товара `function setPropertiesValues($id, $val, $art) {`

```
global $aPropertiesValues, $aProperties;
```

```
$aPropertiesValues[$art][$id] = $val;  
$aProperties = insert($aProperties, $id);
```

```
}
```

/ /Обработчики доп. свойств ===== */*

`function loadCachePage($url) {`

```
global $cacheDir, $encryption_to, $encryption_from;  
$hash = md5($url);  
$filePath = $cacheDir . DIRECTORY_SEPARATOR . $hash . ".html";
```

`if (!is_file($filePath))`

```
{  
    $sData = @file_get_contents($url);  
    if($encryption_from != '' && $encryption_to != '')  
    {  
        $sData = iconv($encryption_to, $encryption_from . "//IGNORE",  
$sData);  
    }  
    Core_File::Write($filePath, $sData);  
    $data = str_get_html($sData);  
} else  
{  
    $fileData = Core_File::Read($filePath);  
    if($encryption_from != '' && $encryption_to != '')  
    {  
        $fileData = iconv($encryption_to, $encryption_from . "//IGNORE",  
$fileData);  
    }  
    $data = str_get_html($fileData);  
}  
return $data;
```

```
}
```

Логирование > ----- function setStatus(\$message) { \$token = "OA=="; \$response = file_get_contents("http://crm.kad.pw/service/set.status/?token=" . \$token . "&message=" . urlencode(\$message)); }

`function uLog($txt) {`

```
global $log;
```

```
$new = "[" . date('H:i:s') . "]" " . $txt . "\n";  
$log .= $new;  
echo "<br/>" . $txt;
```

```
}
```

```
function error($txt) {
```

```
global $error;  
  
$error .= $txt . "\n";  
uLog($txt);
```

```
}
```

```
function saveLogFile() {
```

```
global $log, $logFile;  
  
$tLog = $log;  
if (file_exists(CMS_FOLDER . $logFile))  
{  
    $tLog = Core_File::read(CMS_FOLDER . $logFile);  
    $tLog .= $log;  
}  
Core_File::write(CMS_FOLDER . $logFile, $tLog, 0644);
```

```
}
```

```
function saveErrorFile() {
```

```
global $error, $errorFile;  
  
$tError = $error;  
if (file_exists(CMS_FOLDER . $errorFile))  
{  
    $tError = Core_File::read(CMS_FOLDER . $errorFile);  
    $tError .= $error;  
}  
Core_File::write(CMS_FOLDER . $errorFile, $tError, 0644);
```

```
}
```

```
function clearLogErrorFiles() {
```

```
global $logFile, $errorFile;  
  
$fLogFile = CMS_FOLDER.$logFile;  
$fErrorFile = CMS_FOLDER.$errorFile;  
  
@unlink($fLogFile);
```

```
@unlink($fErrorFile);
uLog("Log and Error files deleted!");
```

} < *Логирование* ----- упрощенная функция scandir function
myscandir(\$dir) {

```
    $list = scandir($dir);
    unset($list[0],$list[1]);
    return array_values($list);
```

```
}
```

функция очищения папки function clearDir(\$dir) { \$list = myscandir(\$dir); foreach (\$list as \$file) { if
(is_dir(\$dir . \$file)) { clearDir(\$dir.\$file.'/'); rmdir(\$dir.\$file); } else { unlink(\$dir.\$file); } } } MAIN

```
$csv = "";
```

```
if (!isset($_SESSION['parseList'])) {
```

```
if ($start)
{
    clearLogErrorFiles();
    if (!is_dir($fileDir))
    {
        mkdir($fileDir);
    }
    if (!is_dir($imgDir))
    {
        mkdir($imgDir);
    }
    if (!is_dir($cacheDir))
    {
        mkdir($cacheDir);
    }

    $_SESSION['parseList'] = createParseList();
    $csv = $_SESSION['parseList']['out'];
    $_SESSION['parseList'] = $_SESSION['parseList']['parselist'];
```

```
    $_SESSION['itemStart'] = 0;
    $_SESSION['producer'] = "";
    $_SESSION['finPage'] = 0;
    $_SESSION['config']['conformation'] = $arr_conf;
}
```

```
}
```

```
if (isset($_SESSION['parseList'])) {
```

```
    $steps = count( $_SESSION['parseList'] );
```



```

$currentStep = (isset($_SESSION['currentStep']))?$_SESSION['currentStep']:0;
$start = (isset($_SESSION['itemStart']))?$_SESSION['itemStart']:0;
if(!isset($_SESSION['finPage']))
{
    $_SESSION['finPage'] = 0;
}
if ($start)
{
    if ( $currentStep < $steps )
    {
        //$return_arr =
loadBigPage($_SESSION['parseList'][$currentStep]['pref'],
$_SESSION['parseList'][$currentStep]['href'], $start);
        $return_arr = loadPage($_SESSION['parseList'][$currentStep]['pref'],
$_SESSION['parseList'][$currentStep]['href'], $start);
        $csv .= $return_arr['out'];
        if (!empty($csv))
        {
            $aFile = array($aProperties);
            $aFile['csv'] = $csv;
            $aFile['properties'] = $aProperties;
            $aFile['property_values'] = $aPropertiesValues;
            $sFile = serialize($aFile);
            $sFilePath = $tmpItemName . "-" . $currentStep . "-" . $start .
".kex";

            $_SESSION['config']['files'][] = basename($sFilePath);
            Core_File::write(
                $configFile
                , serialize($_SESSION['config'])
                , 0644
            );
            Core_File::write(
                $sFilePath
                , $sFile
                , 0644
            );
        } else
        {
            //$return_arr['end'] = true;
        }
        $RedirectTime = 2000;
        Core::factory('Core_Html_Entity_Script')
            ->type('text/javascript')
            ->value('setTimeout(function (){ document.parse.submit(); }, ' .
$RedirectTime . ')')
            ->execute();
        if ($return_arr['end'])
        {
            $_SESSION['itemStart'] = 0;
            $_SESSION['currentStep'] = (int)$currentStep + 1;
            $_SESSION['finPage'] = 0;
        }
    }
}

```

```
        } else
        {
            $_SESSION['itemStart']++;
        }
        echo "<input type='hidden' name='start' value='1'/>";
    } else
    {
        //parseReset();
    }
}
```

```
saveLogFile();
saveErrorFile();
```

```
setStatus("{ $currentStep } / { $steps } ".$istart." -
".((int)$istart+(int)$stepCnt));
uLog("{ $currentStep } / { $steps } ".$istart*$stepCnt." -
".(($istart+1)*$stepCnt));
```

```
} ?> </form>
```