

Kaufoption für CMBasic mit Symbolbefehl

(Stand 26.01.2018)

Offene Probleme:

- Nach Versand, kann wieder auf Bestellformular zurückgegangen werden und erneut versendet werden! Das **Quittungsmail** kann so **unendlich mal versendet werden**
- Mail von **Bestellform kann Umlaute enthalten, die nicht von allen Browsern richtig angezeigt werden.**
- Abbruchknopf bei Bestellformular noch einfügen.
- Wie können 2 – 3 Kaufknöpfe eingefügt werden? Symbolbefehl mit Nr versehen (KAUFEN1 bis KAUFEN3 und mehrere bot_kaufen-Programme würde gehen! -> **Kann umgangen werden mit vorheriger Auswahl in Liste oder bildlicher Darstellung und Kaufknopf bei Objektdetails.**

Vorstudien

Idee und Wünsche an die Kaufoption:

- Mittels Symbolbefehl kann einem Gegenstand (Bild) ein Kauf-Icon beigelegt werden.
- Dieses Kauf-Icon kann pro Verkaufsseite nur einmal verwendet werden können.
- Dieses Kauf-Icon soll bei Anklicken automatisch den Gegenstandsname, eine ID und den Preis zu einem Bestellformular weitergeben.
- Das Bestellformular soll die Möglichkeit zum Stornieren anbieten
- Das Bestellformular soll die erfasste Bestellung inkl. einsetzbaren Versandkosten per Mail an den Besteller (Bestellkopie) und eine Kopie an den Webseitenbesitzer senden.
- Bei der Bestellung müssen Personalien, Telefon und E-Mail erfasst und der Bestellung beigelegt werden.
- Bei Bestellung sollte der Symbolbefehl herausgelöscht werden, bzw. das Formular gelöscht sein. Es wäre auch denkbar bei bereits verkauften Objekten einen Hinweis zuzuschalten.
- Direktes Bezahlen ist nicht vorgesehen.

Aufbau und Name des Symbolbefehls:

Aufbau ähnlich wie andere Symbolbefehle (Beispiele: {sitemap} / {blog,nnn} / {galerie,galerieX} / {PWZUGANG} etc.):

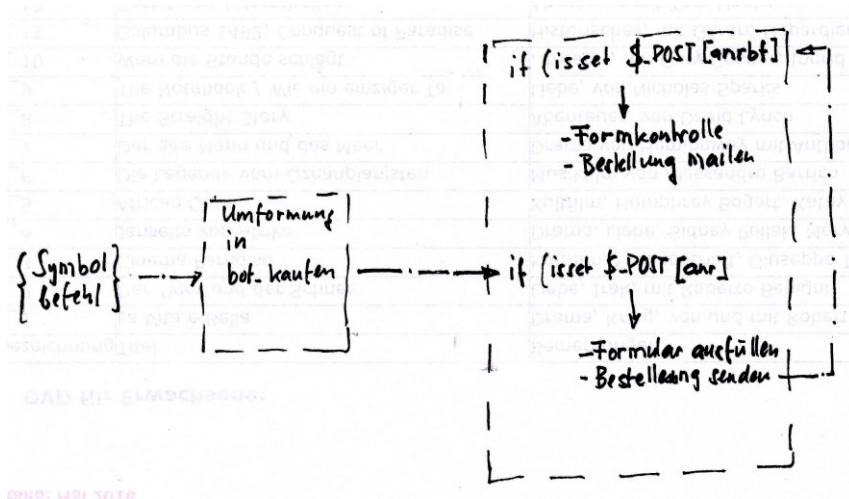
{KAUFEN, Artikelnummer, Artikelbezeichnung, Preis}

Informationsabläufe im CMBasic

In der Datei **contentshow.inc.php** wird eine zusätzliche Zeile zur Einbindung des Symbolbefehls eingefügt. Diese Zeile ist aber nur funktionsfähig, wenn vorher in der unter dem Verzeichnis „edit“ abgelegten Konfigurationsdatei **config.inc.php** der Symbolbefehl freigegeben worden ist.

Nun wird der erkannte Symbolbefehl zur Datei **bot_kaufen.inc.php** weitergeleitet, wo er (wie im nachfolgenden Bild gezeigt) in seine Attribute zerlegt und in einen html-String umgewandelt wird, der an Stelle des Symbolbefehls in den Seitencode (\$acticon) eingefügt wird. Dieser html-String enthält ein entsprechend den Attributen vorbereitetes unsichtbares Formular mit einem anklickbaren Kaufen-Knopf mit welchem das Bestellformular angesteuert wird.

Das Bestellformular zeigt zuoberst die drei Attribute des Kaufen-Knopfes (Artikelnummer, Bezeichnung, Preis) und kann nun durch den Käufer mit seinem Namen, seiner Adresse, seiner Mailadresse und seiner Telefonnummer ergänzt werden. Die vollständig und korrekt ausgefüllte Bestellung wird mit einem Klick auf einen Bestell-Knopf in eine Mailquittung umgewandelt welche an die Mailadressen des Käufers und des Webmasters versandt wird.



Variablenfestlegungen:

Symbolbefehl	<u>bot_kaufen</u>	<u>bestellform</u>	<u>bestellform</u>	<u>bestellung</u>	<u>bestellung</u>
	Form-Ausgang	Eingang	Form-Ausgang	Eingang	Druck/Mail
Artikelnummer	anrk	\$anr	anrbf	\$anrbe	\$anrbe
Bezeichnung	bezk	\$bez	bezbfbf	\$bezbe	\$bezbe
Preis	preisk	\$preis	preisbf	\$preisbe	\$preisbe
		<u>Vorname</u>	vornbf	\$vornbe	\$vornbe
		<u>Nachname</u>	nachnbf	\$nachnbe	\$nachnbe
		<u>Strasse</u>	strbf	\$strbe	\$strbe
		<u>Ort</u>	ortbf	\$ortbe	\$ortbe
		<u>PLZ</u>	plzbf	\$plzbe_k	\$plzbe
		<u>Land</u>	landbf	\$landbe_k	\$landbe
		<u>Mail</u>	mailbf	\$mailbe_k	\$mailbe

Eintrag auf Seite / von Hand eingefügt / ungeprüft / getrimmt / voll geprüft

Entwicklung des Moduls „bot_kaufen.inc.php“:

```

<?php
defined('CMBASIC') or die();
/* Analysiert und ersetzt den Symbolbefehl durch "kaufen"-Knopf.
Gibt Attribute weiter */
$muster_syntax = "|{KAUFEN,URL,Titel,Preis}|";
$suchmuster = "|{((KAUFEN),(. *?))}|si";
preg_match_all($suchmuster,$acticon,$kauffund);
foreach($kauffund[1] as $kaufpara_komp) {
    $kaufpara = explode(", " , $kaufpara_komp);
    if (!empty($kaufpara[1])) {
        $anrk = $kaufpara[1];
        $kaufadd = "$kaufpara[1]";
    }
    if (!empty($kaufpara[2])) {
        $bezk = $kaufpara[2];
    }
}
  
```

```

    $kaufadd .= ",$kaufpara[2]";
}
if (!empty($kaufpara[3])) {
    $spreisk = $kaufpara[3];
    $kaufadd .= ",$kaufpara[3]";
}
}
}
/* Erstellen des hinter dem Kauf-Knopf versteckten Kauf-Formulars */
$kaufcode_0 = <<<KAUFEN
<form action='skripts/content/bestellform.php' name="abfrage"
method='POST'>
<input type='hidden' name='anr' value='{ $anrk }'
maxlength='5'>
<input type='hidden' name='bez' value='{ $bez }' maxlength='25'>
<input type='hidden' name='preis' value = '{ $spreisk }' maxlength='5'>
<input type='submit' name='abfrage' value='kaufen'></td>
</form>
KAUFEN;
/* Aendern des Kaufcodes zum Einfügen */
$kaufcode = "$kaufcode_0";
/* Einfügen des fertigen Kaufcodes an Stelle des Symbolbefehls */
$muster2 = "|{KAUFEN,$kaufadd}|";
$acticon = preg_replace($muster2, $kaufcode, $acticon);
?>

```

Entwicklung des „bestellform.php“:

```
<?php
```

```

/* Überprüft POST von bestellform.php und lässt bei richtiger
Meldung die Bestellung aus */

if ((isset($_POST['anrbf'])) && (isset($_POST['bezbef'])) &&
( isset($_POST['preisbf']) && (isset($_POST['vornbf']))
&& (isset($_POST['nachnbf'])) && (isset($_POST['strbf'])) &&
( isset($_POST['ortbf'])) && (isset($_POST['plzbf'])) &&
( isset($_POST['landbf'])) && (isset($_POST['mailbf'])) &&
( isset($_POST['telbf'])) ) ) {

/* Interne Variablenwerte für Bestelldokument */
$anrbe = $_POST['anrbf'];
$bezbe = $_POST['bezbef'];
$preisbe = $_POST['preisbf'];

/* extern eingegebene Variablenwerte für Bestelldokument */
$vornbe_k = isset($_POST['vornbf']) ?
htmlspecialchars(trim($_POST['vornbf'])):"";
$muster_vorn = "/^[^0-9]{4,25}$/";
if (preg_match($muster_vorn, $vornbe_k) == 0) {
    $vornbe = "";
}
}
}

```

```

    }else{
        $vornbe = $vornbe_k;
    }
    $nachnbe_k = isset($_POST['nachnbf']) ?
    htmlspecialchars(trim($_POST['nachnbf'])):"";
    $muster_nachn = "/^[^0-9]{4,25}$/";
    if (preg_match($muster_nachn, $nachnbe_k) == 0) {
        $nachnbe = "";
    }else{
        $nachnbe = $nachnbe_k;
    }
    $strbe_k = isset($_POST['strbf']) ?
    htmlspecialchars(trim($_POST['strbf'])):"";
    $muster_str = "/^[^$!]{4,25}$/";
    if (preg_match($muster_str, $strbe_k) == 0) {
        $strbe = "";
    }else{
        $strbe = $strbe_k;
    }
    $ortbe_k = isset($_POST['ortbf']) ?
    htmlspecialchars(trim($_POST['ortbf'])):"";
    $muster_ort = "/^[^0-9]{4,25}$/";
    if (preg_match($muster_ort, $ortbe_k) == 0) {
        $ortbe = "";
    }else{
        $ortbe = $ortbe_k;
    }
    $plzbe_k = isset($_POST['plzbf']) ?
    htmlspecialchars(trim($_POST['plzbf'])):"";
    $muster_plz = "/^[A-Z0-9- ]{4,8}$/";
    if (preg_match($muster_plz, $plzbe_k) == 0) {
        $plzbe = "";
    }else{
        $plzbe = $plzbe_k;
    }
    $landbe_k = isset($_POST['landbf']) ?
    htmlspecialchars(trim($_POST['landbf'])):"";
    $muster_land = "/^[A-Za-z]{3,15}$/";
    if (preg_match($muster_land, $landbe_k) == 0) {
        $landbe = "";
    }else{
        $landbe = $landbe_k;
    }
    $mailbe_k = isset($_POST['mailbf']) ?
    htmlspecialchars(trim($_POST['mailbf'])):"";
    $muster_mail = "/^[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+\.[a-zA-Z]{2,4}$/";
    if (preg_match($muster_mail, $mailbe_k) == 0) {
        $mailbe = "";
    }
    else{
        $mailbe = $mailbe_k;
    }
}

```

```

$stelbe_k = isset($_POST['telbf']) ?
htmlspecialchars(trim($_POST['telbf'])):"";
$muster_tel = "/^[0-9- ]{3,15}$/";
if (preg_match($muster_tel, $stelbe_k) == 0) {
    $stelbe = "";
} else {
    $stelbe = $stelbe_k;
}
$website = "http://www.zawadi.ch";
$headers = "From: webmaster@zawadi.ch\r\n"."Bcc:
webmaster@zawadi.ch";
$empfaenger = $mailbe;
$betreff = "Ihre Bestellung";
$mailbody = "Artikelbezeichnung: {$bezbe}
Artikelnummer: {$anrbe}
Preis: {$preisbe} Franken

```

Ihre Adressangaben:

```

{$vornbe} {$nachnbe}
{$strbe}
{$plzbe} {$ortbe}
{$landbe}
{$mailbe}
{$stelbe}

```

Hallo {\$vornbe} {\$nachnbe}. Besten Dank fuer die Bestellung!
Ich werde umgehend mit ihnen Kontakt aufnehmen um die Bezahlung und
Uebergabe der Ware zu regeln.

Mit freundlichem Gruss
der Webmaster";

```

if ((empty($vornbe) || empty($nachnbe) || empty($strbe) ||
empty($plzbe) || empty($ortbe) || empty($landbe) ||
empty($mailbe) || empty($stelbe)) {
echo "<p><h1>Fehlerhafte oder fehlende Eingabe!</h1>Bitte mit
Retourknopf (<=>) zurueck und Eingabe nach Richtigstellen erneut
senden</p>";
} else {
@mail($empfaenger, $betreff, $mailbody, $headers);
?>

```

```

<!DOCTYPE HTML>
<html lang="de">
<html>
<head>
<meta http-equiv="">
<title>Bestellung</title>
<link rel="stylesheet" type="text/css"
href="../../../../backend/css/backend.css">
</head>

<body>

```

```

<h3>Ihre Bestellung</h3>
<p>Artikelbezeichnung: <?php echo $bezbe ; ?> <br>
Artikelnummer: <?php echo $anrbe ; ?> <br>
Preis: <?php echo $preisbe ; ?> Franken <br><br>
Adressangaben: <br>
<?php echo $vornbe ; ?> <?php echo $nachnbe ; ?> <br>
<?php echo $strbe ; ?> <br>
<?php echo $plzbe ; ?> <?php echo $ortbe ; ?> <br>
<?php echo $landbe ; ?> <br>
<?php echo $mailbe ; ?> <br>
<?php echo $stelbe ; ?> <br>
</p>
<p>Besten Dank fuer Ihre Bestellung!<br>
Sie werden umgehend ueber Mail eine Kopie ihrer Bestellung und Infos
ueber das weitere Vorgehen erhalten.</p>
<p>-> zur <a title="zur Website" href=<?php echo $website ; ?>
>Website</a></p>
</body>

```

```
</html>
```

```
<?php
```

```
}
```

```
} else {
```

```
/* Überprüft POST von bot_kaufen.inc.php und zeigt das auszufüllende
Bestellformular an */
```

```
if ((isset($_POST['anr'])) && (isset($_POST['bez'])) &&
(isset($_POST['preis']))) {
```

```
/* Variablen für Bestellformular */
```

```
$anr = $_POST['anr'];
```

```
$bez = $_POST['bez'];
```

```
$preis = $_POST['preis'];
```

```
?>
```

```
<!DOCTYPE HTML>
```

```
<html lang="de">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="">
```

```
<title>Bestellformular</title>
```

```
<link rel="stylesheet" type="text/css"
href="../../../backend/css/backend.css">
```

```
</head>
```

```
<body>
```

```
<h1>Bestellformular</h1>
```

```
<form action='' method='post' name='bestform'>
```

```
<table width='450px' border='0'>
```

```
<tbody>
```

```
<tr align='left' valign='middle'>
```

```
<td width='140px' align='left'>Artikelnummer</td>
```

```

<td width='20px'></td>
<td width='250px' align='left'><?php echo $sanr ; ?></td>
<td width='20px'><input type='hidden' name='anrbf' value= <?php echo
$sanr ; ?> maxlength='15'></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Artikelbezeichnung</td>
<td width='20px'></td>
<td width='250px' align='left'><?php echo $bez ; ?></td>
<td width='20px'><input type='hidden' name='bezbf' value= <?php echo
$bez ; ?> maxlength='15'></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Preis in SFr</td>
<td width='20px'></td>
<td width='250px' align='left'><?php echo $preis ; ?></td>
<td width='20px'><input type='hidden' name='preisbf' value= <?php
echo $preis ; ?> maxlength='15'></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Vorname</td>
<td width='20px'></td>
<td><input type='text' name='vornbf' size='30' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Nachname</td>
<td width='20px'></td>
<td><input type='text' name='nachnbf' size='30' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Strasse & Nr</td>
<td width='20px'></td>
<td><input type='text' name='strbf' size='35' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Ort</td>
<td width='20px'></td>
<td><input type='text' name='ortbf' size='30' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Postleitzahl</td>
<td width='20px'></td>
<td><input type='text' name='plzbf' size='8' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Land</td>
<td width='20px'></td>
<td><input type='text' name='landbf' size='35' /></td>
</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Mailadresse</td>
<td width='20px'></td>
<td><input type='text' name='mailbf' size='35' /></td>

```



```

</tr>
<tr align='left' valign='middle'>
<td width='140px' align='left'>Telefon/td>
<td width='20px'></td>
<td><input type='text' name='telbf' size='35' /></td>
</tr>
<tr>
<td width='140px'></td>
<td width='20px'></td>
<td><input type='submit' value='bestellen' /></td>
</tr>
</tbody>
</table>
</form>
<p><br />Versenden Sie bitte die vollstaendig ausgefuellte
Bestellung mit dem Knopf 'bestellen'</p>
</body>

</html>
<?php
}
}
?>

```

Beschreibung für's Handbuch

Einbinden einer Verkaufsmöglichkeit

Wenn erwünscht, lässt sich pro Seite unter oder neben einem Gegenstand ein Kauf-Knopf einfügen, welcher bei Anklicken ein ausfüllbares Bestellformular für diesen Gegenstand einblendet. Es ist zu beachten, dass nur ein einziger Kauf-Knopf pro Seite möglich ist! Durch versenden des vollständig ausgefüllten Bestellformulars wird dem Käufer und dem Verkäufer ein Mail mit den Bestelldaten übermittelt.

Der dazu verwendete Symbolbefehl lautet:

{KAUFEN, Artikelnummer, Artikelbezeichnung, Preis}

Das erste fixe Attribut (KAUFEN) dieses Befehls zeigt dem System an, dass an diesem Platz einen Kauf-Knopf einblenden soll. Mit dem zweiten Attribut (Artikelnummer) wird die Artikelnummer des Gegenstandes festgelegt, mit dem dritten Attribut (Bezeichnung) wird der Gegenstand Umschrieben und mit dem vierten Attribut (Preis) wird der geforderte Preis angegeben

Für die Artikelnummer sind maximal 15 Zeichen vorgesehen. Es können Buchstaben (exkl. Sonderzeichen), Zahlen und Sonderzeichen (exkl. Komma und \$) verwendet werden. Leerschläge werden nicht akzeptiert können aber mit Underscore ungangen werden.

Unter Artikelbezeichnung wird der Gegenstand verbal umschrieben. Dies kann ein einfacher Name oder eine Beschreibung aus mehreren Wörtern sein. Bei mehreren Wörtern müssen die Lücken zwischen den Wörtern weggelassen oder durch ein Underscore-Zeichen ersetzt werden. Umlaute sind erlaubt.

Mit dem Attribut **Preis** wird der Wert in Franken eingetragen. Es sind Preise bis 99'999 Franken vorgesehen. Eintrag nur in Zahlen ohne Einheit!

Der Symbolbefehl wird im Dokument durch einen grauen „Kaufen-Knopf“ ersetzt. Pro Seite ist nur ein einziger Symbolbefehl möglich.

Beim Anklicken des Kaufen-Knopfes wird ein Bestellformular eingeblendet. Die im Symbolbefehl enthaltenen artikelrelevanten Informationen werden hier automatisch in den ersten 3 Zeilen aufgelistet. Die restlichen Informationen wie Adresse, Mail, Telefon müssen vom Käufer selber in die vorgegebenen Felder eingefügt werden.

Ist das Bestellformular fertig ausgefüllt, kann die Bestellung mit einem Klick auf den Knopf „bestellen“ ausgelöst werden. Die Bestellung wird dann in Form eines Mails an den Besteller und als Blindkopie zum Verantwortlichen der Webseite verschickt.

Die Kauf-Option muss vor der Verwendung eingerichtet und freigegeben werden. Dazu müssen die die beiden Dateien bot_kaufen.inc.php und bestellform.php neu eingefügt, die beiden Dateien contentshow.inc.php und edit/config.inc.php angepasst und das bestellform.php mit der Websiteadresse und dem Mail des Verantwortlichen versehen werden.

Einrichten des Symbolbefehls auf der Website:

Datei **bot_kaufen.inc.php** unter Verzeichnis minibot/ ablegen

Datei **bestellform.php** unter Verzeichnis skripts/content/ ablegen

Datei **contentshow.inc.php** mit folgendem Einschub ergänzen (ca. Zeile 584 ff):

```
if (defined('KAUFEN') && KAUFEN == TRUE){  
    include_once 'minibot/bot_kaufen.inc.php';  
}
```

Datei **edit/config.inc.php** mit folgendem Einschub ergänzen:

```
define('KAUFEN', TRUE); // Bestellformularzugang freigeben
```

Datei **bestellform.php** bezüglich Eigenadresse (z.B. \$website = http://www.projekte.zawadi.ch / \$headers = "From: webmaster@zawadi.ch" / "Bcc: webmaster@zawadi.ch") in den Variablen für Bestellung (Zeilen 49ff) anpassen.

Symbolbefehl **{KAUFEN,Artikelnummer,Artikelbezeichnung,Preis}** auf Seite einfügen

Test der entwickelten Application KAUFEN

Funktionstest in Arbeitsumgebung

Ein Funktionstest zeigte, dass der Symbolbefehl {KAUFEN} grundsätzlich funktioniert. Beim Test wurden folgende Eigenheiten festgestellt:

- **Pro Seite ist nur ein einziger Kaufknopf einrichtbar** (Grund: Variablen werden beim Symbolbefehl nicht differenziert behandelt!)

Überprüfung des Einrichtverhaltens

Eingeben der Artikelnummer:

- Beim Einrichten des Symbolbefehls können für Artikelnummer mehr als 15 Zeichen eingegeben werden. Es sind Buchstaben, Zahlen und Kombinationen davon möglich. Komma und Sonderzeichen \$ gehen nicht, hingegen funktionieren die Umlaute und die Sonderzeichen #.-_ . Leerschläge werden nicht akzeptiert, Umgehung mit Underscore. Alle diese Zeichen wurden bis zum Bestätigungsmail weitergegeben. Test erfolgte mit **#.A.37ü-1234567890**

Eingeben der Artikelbezeichnung:

- Beim Einrichten des Symbolbefehls können für Bezeichnung mehr als 15 Zeichen in Form von Zahlen-, Buchstaben- oder gemischten-Gruppen verwendet werden. . Leerschläge werden nicht akzeptiert, Umgehung mit Underscore. Komma und das Sonderzeichen \$ gehen nicht, hingegen funktionieren die Umlaute und Sonderzeichen #.-_ . Alle diese Zeichen wurden bis zum Bestätigungsmail weitergegeben. Test erfolgte mit **#.Wort1_“Wört2“-Wort3**

Eingeben des Preises:

- Beim Einrichten des Symbolbefehls können für Preis mehr als 15 Zeichen eingegeben in Form von Zahlen-, Buchstaben- oder gemischten-Gruppen verwendet werden. Leerschläge werden nicht akzeptiert, Umgehung mit Underscore. werden. Komma und das Sonderzeichen \$ gehen nicht, hingegen funktionieren die Umlaute und Sonderzeichen #.-_ . Alle diese Zeichen wurden bis zum Bestätigungsmail weitergegeben. Test erfolgte mit **#.99'999“Franken“**

Position des Kaufknopes:

- Die Positionierung des Symbolbefehls ist irgendwo auf der Seite möglich, **kann aber nicht immer sehr genau positioniert werden.**
- normalerweise wird nach Text immer mit einer neue Zeile begonnen
- Kaufknopf in Tabelle zusammen mit Text (vorher oder nachher) gibt unlösbare Darstellungsprobleme.
- Der Kaufknopf kann mittels Tabelle mit 2 Kolonnen auf gleiche Zeile wie Text gebracht werden.
- Ein grauer Kaufbutton wird gezeigt. Unklar wie er durch etwas anderes ersetzt werden kann.

Ausfüllen des Bestellformulars:

- Die Kontrolle der Texteingabe inkl. Begrenzung auf Maximallänge funktioniert

Versand der Bestellung:

- Versand ist nur mit vollständig ausgefülltem Formular möglich

- Nach Versand, kann wieder auf Bestellformular zurückgegangen werden und erneut versendet werden! Das **Quittungsmail** kann so **unendlich oft versendet werden**
- Mail von **Bestellform kann Umlaute enthalten die aber nicht von allen Browsern richtig angezeigt** werden. Mail auf MailManager und iPad zeigen die Umlaute richtig. Thunderbird hingegen zeigt Mails mit Fragezeichen, da er den falschen Code benutzt. Grund ist bis jetzt unklar. Das Mailformular überträgt auf jedenfall den Code nicht. Wird er eventuell aus dem HTML der Site abgeleitet? Prüfung mit Thunderbird zeigte:

<u>Ursprung des Mails</u>	<u>Umlautanzeige</u>	<u>erkannte Textcodierung</u>
Bestellform	falsch	unicode
SBB-Mail	richtig	unicode
Bethli	falsch	unicode
Bethli	richtig	westlich
Grizetti	richtig	westlich

Vorschläge zum eliminieren des Mehrfachversand-Problems

Variante mit Zeitstempel

Die Variante mit Zeitstempelvergleich funktioniert nicht, da unklar ist wie viel Zeit der Besteller zum Ausfüllen des Formulars benötigt.

Variante mit Löschen der Mailadresse

Die Mailadresse wird extern abgelegt (z.B. im selben Verzeichnis wie die Datei bestellform.php) und beim Aussenden des Mails von dort bezogen. Unmittelbar nach Versenden des Mails wird der Inhalt im externen Speichers gelöscht, so dass er für eine weitere Verbindung nicht mehr zur Verfügung steht. Diese **Variante wird vermutlich zu Fehlermeldungen führen**, da eine der Variablen fehlt.

Variante mit Löschen des Formularinhaltes

Der **Formularinhalt wird unmittelbar nach Aussenden gelöscht**. Es kann zwar nach Rücksprung erneut ausgefüllt und versendet werden erfordert aber grossen Aufwand und schreckt ab. Zudem lässt sich eine Bestellung solange der Bestellknopf noch sichtbar ist jederzeit wiederholen.

Vorschläge zur Darstellung von Verkaufsseiten

Auswahlmöglichkeiten

Hier wird das Produkt mit der Maus ausgewählt!

mit Auswahlliste:

- kleines antikes Teleskopfernrohr 12 x 28
- antikes Teleskopfernrohr Negretti 30x60 (ca. 1900)
- Photokamera Rollei 35 (ca. 1940)
- antike Influenzmaschine Radiguet (ca. 1885) / restauriert!
- Occasions Insel-Solaranlage 120 Wh
- antikes Taschen-Amperemeter

mit einzelnen Auswahlbildern:



mit Auswahlbilderzusammenstellung



Produktseite mit Details und „Kaufknopf“

Das ausgewählte Produkt wird mit Doppelklick aktiviert. Nun zeigt sich eine Produktseite mit allen Details (Größen, Preis, etc. und einem „kaufen“ - Knopf

[Home](#) > [Antiquitäten & Occasionen](#) > antikes Amperemeter

ANTIQUES TASCHEN-AMPÈREMETER

Taschen-Ampèremeter aus den Anfängen des letzten Jahrhunderts (geschätztes Alter: 80 Jahre). Technisch gesehen ist es ein analoges Spulen-Anzeigesystem welches in eine Taschenuhr eingebaut wurde und zur Messung des Verbraucherstromes bis maximal 6 A diente.



Das Taschen-Ampèremeter ist in einem guten Zustand. Wert ca. 25 bis 50 CHF.

Verkaufspreis: Fr. 50.-

[kaufen](#)

Bestellformular

Nach Klick auf den Knopf „kaufen“ erscheint ein Bestellformular zum Ausfüllen das mittels „bestellen“ verschickt werden kann.

Bestellformular

Bild Nr	55
Bildbezeichnung	Amperemeter
Preis in Fr	50
Vorname	<input type="text" value="Walo"/>
Nachname	<input type="text" value="Zach"/>
Strasse & Nr	<input type="text" value="Unterdorfstrasse 53"/>
Ort	<input type="text" value="Ostermundigen"/>
Postleitzahl	<input type="text" value="3072"/>
Land	<input type="text" value="Schweiz"/>
Mailadresse	<input type="text" value="zawadi@bluewin.ch"/>
Telefon	<input type="text" value="031 932 05 75"/>
	<input type="text" value="031 932 05 75"/>

Versenden Sie bitte die vollstaendig ausgefuellte Bestellung mit dem Knopf 'bestellen'

Bestätigung der Bestellung

Nach Ausfüllen des Bestellformulars und Klick auf den Knopf „bestellen“ erscheint eine Bestätigung der Bestellung, die auch per Mail an den Besteller und den Webseitenverantwortlichen verschickt wird.

- **Ihre Bestellung**

Artikelbezeichnung: Amperemeter
Artikelnummer: 55
Preis: 50 Franken

Adressangaben:
Walo Zach
Unterdorfstrasse 53
3072 Ostermundigen
Schweiz
zawadi@bluewin.ch
031 932 05 75

Besten Dank fuer Ihre Bestellung!
Sie werden umgehend ueber Mail eine Kopie ihrer Bestellung und Infos ueber das weitere Vorgehen erhalten.

-> zur [Website](#)

Hilfsmittel / Spez. PHP-Funktionen

1. Überprüfen auf Mindesteingabe in Formularen:

```
if ((empty($_POST['FeldA'])) OR (empty($_POST['FeldB'])) OR
(empty($_POST['FeldC'])) OR (empty($_POST['FeldD']))) {
echo „<p><h1>Eingabe nicht vollständig!</h1>Bitte mit Retourknopf
zurück (<=> und Eingabe vervollständigen</p>“;
}
```

2. Überprüfen der Text-Formulareingaben mit Schablonen:

Bemerkung: Umlaute äöü und Spezialzeichen éèâô lassen sich nicht mit diesen Schablonen prüfen (auch nicht mit utf-Umlauten!). In solchen Fällen muss mit invertierten Schablonen gearbeitet werden.

```
$inh_p = $_POST['inh']; // prov. Inhalt
$muster_inh = "/^[A-Z]{1,1}$/"; // Prüfschablone
if (preg_match($muster_inh, $inh_p) == 0) { //Prüfanweisung
    $inh = ""; // leer wenn Inhalt nicht akzeptiert
}
else{
    $inh = $inh_p; // akzeptierter Inhalt wird weitergegeben
}
```

Prüfschablonen-Vorschläge:

Schablone für Text ohne Umlaute: `"/^[A-Za-z0-9-_.*]{3,25}$/"`
Invertierte Schablone für Text mit Umlauten : `"/^[^0-9]{3,25}$/"`
Schablone für Artikel-Nummer : `"/^[A-Za-z0-9-_.]{2,8}$/"`
Schablone für Postleitzahl: `"/^[A-Z0-9-]{4,8}$/"`
Schablone für Mailadressen: `"/^[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+\.[a-zA-Z]{2,4}$/"`

Erklärung dazu: geschweifte Klammer {} hinter eckiger [] zeigt minimale und maximale Länge des Ausdrucks in der eckigen Klammer an. `\/^` bedeutet Vorlage-Beginn und `$/` bedeutet Vorlage-Ende. Das Pluszeichen fügt die verschiedenen Teile zusammen. Ein Zeichen unmittelbar vor der Klammer bedeutet, dass dieses Zeichen einmal vorkommen muss. Bei Schablonen mit Umlauten wird die max. Länge pro Umlaut um 1 Character reduziert.

3. Überprüfen der Mail-Formulareingabe mit Schablone

```
$mail_p = $_POST['mail']; // prov. Mailadresse
$muster_mail = "/^[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+\.[a-zA-Z]{2,4}$/";
if (preg_match($muster_mail, $mail_p) == 0) { //Prüfanweisung
    $mail = ""; // leer wenn Inhalt nicht akzeptiert
}
else{
    $mail_1 = $mail_p; // akzeptierter Inhalt wird weitergegeben
}
```


4. Leerzeichenreduktion & Entfernen von html-Tags < und > aus Formulareingaben in ternärer Form (ternärer Operator):

```
$inh_p = isset($_POST['inh']) ?  
htmlspecialchars(trim($_POST['inh'])):"";
```

Erklärung: Ausgabe = (Ausdruck1) ? (Ausdruck2) : (Ausdruck3)
Wenn Ausdruck1 = true ist erscheint Ausdruck2 am Ausgang d.h.
modifizierter inh

Wenn Ausdruck1 = false ist erscheint Ausdruck3 am Ausgang d.h.
leerer String

trim() Putzt unnötige, doppelte Leerzeichen heraus und optimiert so String

htmlspecialchars() wandelt < und > in Escape-Codes um und verhindert so Hackzugriffe!

5 Zeichenauswechslung (ersetzt im String alle im Array festgelgten Werte):

```
$string = ' . . . . . '  
$array = array( 'ä' => 'ae', 'ü' => 'ue', 'ß' => 'ss')  
$neuerstring = strtr ( $string, $array );
```

6 Umwandlung Grossbuchstaben in Kleinbuchstaben:

```
$name = 'xyz';  
$array=('ä'=>'ae','ö'=>'oe');  
$neuername = strtr($name,$array)
```

Nicht funktionierende if-Prüfung im Programm:

```
if ((empty($vornbe) || empty($nachnbe) || empty($strbe) || empty($plzbe) ||  
empty($ortbe) || empty($landbe)) {
```

```
echo "$vornbe$nachnbe$strbe$plzbe$ortbe$landbe";
```

- Die Eingaben werden angezeigt, sie sind daher nicht leer!
- Wenn eine Eingabe leer wäre würde auch nichts angezeigt
- Warum wird diese Schleife trotzdem freigegeben?

```
echo "<p><h1>Fehlerhafte oder fehlende Eingabe!</h1>Bitte mit Retourknopf (<=>) zurueck  
und Eingabe nach Richtigstellen erneut senden</p>";
```

```
}else{  
@mail($empfaenger, $betreff, $mailbody, $headers);  
?>
```

Anzeige im Browser:

HansjörgLüdiCH 3072MühlheimSchweizzawa1@bluewin.ch031 932 05 75

Fehlerhafte oder fehlende Eingabe!

Bitte mit Retourknopf (<=>) zurueck und Eingabe nach Richtigstellen erneut senden

Erkenntnisse / weiteres Vorgehen:

isset()	ist Variable gesetzt?	Wahrheitswert = True, wenn gesetzt (Wert unwichtig, kann auch leer oder 0 sein)
isempty()	ist Inhalt der Variablen leer?	Wahrheitswert = True, wenn Inhalt leer ist oder den Wert 0 hat
!isempty()	ist Inhalt der Variablen nicht leer?	Wahrheitswert = True, wenn Inhalt nicht leer ist (Wert 0 wird als „nicht leer“ erkannt)

Die if-Schleife funktioniert nicht richtig, wenn die Variablen vorher auf äöü überprüft wurden! Bei Verzicht auf dieses Filter funktioniert alles korrekt.

Da der **Fehlergrund nicht eruierbar** ist, wurde auf **Inversprüfung der Variablen** gewechselt.

Invers-Schablone für Text mit Umlauten : `"/^[^äöü0-9]{3,25}$/"`

In diesem Fall müssen keine Umlaute überprüft werden, womit das Problem umgangen werden kann. Auf diese Weise können zudem auch noch die Maximal- und Minimallängen überprüft werden. Aber **Achtung: Pro durchgelassenem Umlaut wird die max. Länge pro Umlaut um 1 Character reduziert!** Es sieht so aus wie Umlaute vor der Prüfung in Normbuchstaben umgesetzt werden