
HTM2COB 1.02

User's Guide

HTML to COBOL converter

László Erdős, email: lex-x@freenet.de
Facebook: <https://www.facebook.com/wortfee>

HTM2COB (and this document) is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

HTM2COB (and this document) is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with HTM2COB (and this document). If not, see <http://www.gnu.org/licenses>

Summary of Changes

Date	Edition	Change Description
2019.05.01	v01.00	Initial release of this document.
2019.12.30	v01.01	Project p001_wortfee added.
2020.01.05	v01.02	Update documentation for samples 0029 – 0039.

Table of Contents

1.	Introduction.....	5
1.1	What does it do?.....	5
1.2	Installation and Configuration.....	5
1.2.1	Directory structure.....	5
1.2.2	Compile and run.....	5
1.3	Test.....	6
1.4	Requirements.....	6
1.5	Features of the generated CGI server.....	6
2.	Usage.....	7
2.1	Command line options of HTM2COB.....	7
2.2	A detailed example.....	7
2.2.1	Input file name convention.....	7
2.2.2	Embedding COBOL code in the HTML file.....	7
2.2.3	Run the HTM2COB converter.....	8
2.3	The parameter file.....	9
2.4	Built-in modules.....	9
2.4.1	HTM2COB-POST.....	9
2.4.2	HTM2COB-SPEC-CHARS.....	9
2.4.3	HTM2COB-ENV.....	9
2.4.4	HTM2COB-POST-MULTI.....	9
2.4.5	HTM2COB-SESSION-ID.....	10
2.4.6	HTM2COB-SESSION.....	10
2.4.7	HTM2COB-SET-COOKIE.....	10
2.4.8	HTM2COB-GET-COOKIE.....	11
2.4.9	HTM2COB-UPD-SESS-STR.....	11
2.4.10	imgcaptcha.....	11
2.4.11	imgscale.....	11
2.5	Some security hints.....	11
3.	Examples.....	14
3.1	Samples.....	14
3.1.1	0001_hello_get.....	14
3.1.2	0002_hello_post.....	16
3.1.3	0003_hello_form_get.....	17
3.1.4	0004_hello_form_post.....	19
3.1.5	0005_embedding.....	20
3.1.6	0006_comment.....	21
3.1.7	0007_variables.....	22
3.1.8	0008_copyfile.....	24
3.1.9	0009_section.....	25
3.1.10	0010_call.....	28
3.1.11	0011_form_elements.....	30
3.1.12	0012_env_vars.....	32
3.1.13	0013_img_upload.....	34
3.1.14	0014_img_upload_scale.....	36
3.1.15	0015_multiple_menu.....	38
3.1.16	0016_js_alert.....	39

3.1.17 0017_more_button.....	41
3.1.18 0018_link_submit.....	43
3.1.19 0019_menu.....	44
3.1.20 0020_sess_id.....	46
3.1.21 0021_sess_var.....	47
3.1.22 0022_sess_counter.....	48
3.1.23 0023_sess_login.....	50
3.1.24 0024_sess_captcha.....	54
3.1.25 0025_sess_cart.....	58
3.1.26 0026_sess_regen_login.....	65
3.1.27 0027_sess_regen_cart.....	71
3.1.28 0028_ck_visit.....	78
3.1.29 0029_ck_save_data.....	80
3.1.30 0030_ck_param.....	83
3.1.31 0031_ck_sess_login.....	85
3.1.32 0032_ck_sess_captcha.....	90
3.1.33 0033_ck_sess_cart.....	94
3.1.34 0034_ck_sess_regen_login.....	101
3.1.35 0035_ck_sess_regen_cart.....	106
3.1.36 0036_ajax_hello.....	113
3.1.37 0037_ajax_query_str.....	115
3.1.38 0038_jq_form_validate.....	117
3.1.39 0039_jq_jcarousel.....	121
3.2 Projects.....	123
3.3 p001_wortfee.....	123
3.3.1 What does it do?.....	123
3.3.2 Directory structure.....	123
3.3.3 How to install.....	124
3.3.4 Test.....	124
3.3.5 Screenshots.....	125
4. Bibliography and references.....	133

1. Introduction

1.1 What does it do?

With the HTM2COB tool you can convert an HTML file into COBOL source code. The generated COBOL program is a CGI server program. Into the HTML input you can additionally embed COBOL codes with a simple syntax. After generating the COBOL program, you can immediately compile and install it onto your web server.

1.2 Installation and Configuration

1.2.1 Directory structure

htm2cob	→ main directory
htm2cob/readme.txt	→ readme file
htm2cob/doc/HTM2COB_v01_0.odt	→ this document in LibreOffice format
htm2cob/doc/HTM2COB_v01_0.pdf	→ this document in PDF
htm2cob/examples/projects	→ project examples (this is a TO DO)
htm2cob/examples/samples	→ sample examples (see below)
htm2cob/source/htm2cob.cob	→ the HTM2COB converter program
htm2cob/source/htm2cob0.cpy	→ template: WORKING-STORAGE
htm2cob/source/htm2cob1.cpy	→ template: PROCEDURE DIVISION
htm2cob/source/htm2cob2.cpy	→ template: CGI fields / values
htm2cob/source/htm2cob3.cpy	→ template: CGI environment variables
htm2cob/source/htm2cob4.cpy	→ template: decode UTF-8 chars
htm2cob/source/htm2cob5.cpy	→ template: replace HTML special chars
htm2cob/source/htm2cob6.cpy	→ template: session ID with SHA3-512
htm2cob/source/htm2cob7.cpy	→ template: session file handling
htm2cob/source/htm2cob8.cpy	→ template: create cookie string
htm2cob/source/Makefile	→ make file for the converter

In the samples folder are some examples. All examples have a separated make file and a screenshot directory.

1.2.2 Compile and run

First compile the HTM2COB converter under htm2cob/source/ and after it the examples. The examples use the converter. The provided make files are for cygwin. For other environments you have to do some small changes in the make files. For example you have to customize the target directories for your web server.

1.3 Test

This program was developed and tested using:

- Windows 10 (64 bit) running on a HP laptop
- GnuCOBOL 3.1-dev.0, built on Aug 17 2019
- cygwin (64 bit)
- Firefox Quantum 66.0.3 (64-Bit)

1.4 Requirements

- To run the examples you need a web server.
- If you are using GnuCOBOL with BDB (Berkeley DB) you have to set the environment variable DB_HOME to where your session files are saved.
- HTM2COB includes the cobsha3 GnuCOBOL contribution. The cobsha3 GnuCOBOL contribution was developed and tested in a little-endian environment. There are some issues with big-endian, because of redefines and BINARY-DOUBLE. **Very important:** Before you use HTM2COB, you have to run the test for cobsha3 program in your environment, and then you have to check the results!

1.5 Features of the generated CGI server

- HTTP request methods: GET, POST, POST with image upload.
- Debug flag for testing.
- It uses utf-8.
- Definable max data length parameters.
- Image upload options: BMP, GIF, JPG, PNG, TIF.
- Session ID generation with SHA3-512.
- Session file handling. Read / Write additional data for the session ID.
- Time out parameters for the session ID.
- Cookie string creation with formatted time stamp.
- Setting and getting cookies.

2. Usage

2.1 Command line options of HTM2COB

```
htm2cob <input HTML file> [-m] [-v]
htm2cob converts a HTML file in a CGI COBOL program
Options:
<input HTML file> This is a HTML file with embedded COBOL
-m, -module It converts only the lines in <BODY>
-v, -verbose Verbose mode
```

2.2 A detailed example

2.2.1 Input file name convention

In addition to the HTML file, the converter also needs a parameter file, which is a COBOL copy file. Both file names have a special format. The input HTML file has the suffix "_cob.html" and the parameter file has the suffix "_param.cpy". The prefix must be the same. The converter will change the suffix "_cob.html" to ".cob" in the generated output file.

Example:

Input files:	0004_hello_form_post_cob.html
	0004_hello_form_post_param.cpy
Output file:	0004_hello_form_post.cob

2.2.2 Embedding COBOL code in the HTML file

There are only two identifiers for embedding the COBOL code in HTML. One for the variable definitions, and one for program codes.

Defining variables: **<?cob-ws ... ?>**

Defining program codes: **<?cob ... ?>**

It does not matter where you define the variables, the converter will place them all in the WORKING-STORAGE SECTION.

An example for variable definition:

```
<?cob-ws
*> You can define variables in body or outside of body also.
01 fname                      pic x(100).
01 lname                       pic x(100).
01 htm-fname                   pic x(100).
01 htm-lname                   pic x(100).
?>
```

An example for program code:

```
<?cob display "Hello GnuCOBOL world!" end-display ?>
```

Everything up to "</ html>" from the HTML file is placed in the main section in the generated output file. You can define your own sections after the "</ html>". These sections are placed after the main section in the generated output file.

2.2.3 Run the HTM2COB converter

```
$ make
../../../../source/htm2cob 0004_hello_form_post_cob.html -v
Input  file: 0004_hello_form_post_cob.html
Output file: 0004_hello_form_post.cob
PROGRAM-ID. 0004_hello_form_post.
```

Count		Line		Col		Tag	
1		3		1		<HTML>	
2		9		1		<BODY>	
3		11		1		<?COB	
4		16		1		?>	
5		19		1		<?COB	
6		19		51		?>	
7		22		13		<?COB	
8		22		55		?>	
9		23		13		<?COB	
10		23		55		?>	
11		26		1		</BODY>	
12		27		1		</HTML>	
13		29		1		<?COB-WS	
14		35		1		?>	

Check tags...

```
cobc -x -free 0004_hello_form_post.cob -o 0004_hello_form_post
cp 0004_hello_form_post /srv/www/cgi-bin/0004_hello_form_post
cp 0004_hello_form_post_test.html
/srv/www/htdocs/0004_hello_form_post_test.html
```

The converter always checks the tags. With the verbose parameter “-v” you can see the begin and end tags in the output list.

There are cases when you need multiple end BODY and end HTML, e.g. after a possible abort because of errors. The converter can not handle these cases, so you have to code them differently.

You can see this in the example 0023_sess_login_2_cob.html:

```
*>      slash with concatenate, because of the htm2cob preprocessor
DISPLAY "</" "body>" "</" "html>"
```

2.3 The parameter file

The parameter file has the suffix "_param.cpy". It is well commented, and always has the same structure. It is important that you set the parameters carefully. It is security relevant. Set the right request method, number of input fields, memory and other options, what you really need in your server.

2.4 Built-in modules

There are some built-in modules in the generated COBOL program. Here is a short list, but you can see the usage of these modules in the examples later.

2.4.1 HTM2COB-POST

The CGI field names and values will be saved in a memory table and in a variable. With this module you can read the value for a given name. (You can also access the HTM2COB-TABLE and the HTM2COB-DATA-VALUE from your program, if you want. See the generated COBOL code.)

Example:

```
call "HTM2COB-POST" using "firstname", fname end-call
```

2.4.2 HTM2COB-SPEC-CHARS

This module replaces HTML special chars. You have to call it always before writing something back to the client.

Example:

```
call "HTM2COB-SPEC-CHARS" using fname, htm-fname end-call
```

2.4.3 HTM2COB-ENV

This module reads an environment variable.

Example:

```
CALL "HTM2COB-ENV" USING LNK-HTM2COB-ENV END-CALL
```

2.4.4 HTM2COB-POST-MULTI

With a “multiple selection menu” you can get the same CGI variable name multiple times in the internal table. With this module you can read these. (See

later in example 0015_multiple_menu.)

Example:

```
*> use ; as a separator char
   call "HTM2COB-POST-MULTI" using "pizza", ";", ws-pizza end-call
```

2.4.5 HTM2COB-SESSION-ID

This module generates a session ID with the SHA3-512 algorithm. As input the REMOTE-ADDR and HTTP-USER-AGENT will be used. (See below in example 0020_sess_id.)

Example:

```
CALL "HTM2COB-SESSION-ID" USING LNK-HTM2COB-SESSION-ID END-CALL
```

2.4.6 HTM2COB-SESSION

This module writes or deletes the session data in a file. You can call it directly or thru Sections. (See later in example 0021_sess_var, and in other session examples.)

Example:

```
CALL "HTM2COB-SESSION" USING LNK-HTM2COB-SESSION END-CALL
```

Example with Sections:

```
*> start or resume session
   PERFORM HTM2COB-SESS-START

*> delete old sessions
   PERFORM HTM2COB-SESS-DEL-OLD

*> delete a session
   PERFORM HTM2COB-SESS-DESTROY

*> regenerate a session (create a new session, copy old session data in
*> new session, delete old session)
   PERFORM HTM2COB-SESS-REGENERATE

*> set session var
   MOVE "mydog" TO HTM2COB-SESS-VAR-NAME
   MOVE "Morgo" TO HTM2COB-SESS-VAR-VALUE
   PERFORM HTM2COB-SESS-SET

*> get session var
   MOVE "mydog" TO HTM2COB-SESS-VAR-NAME
   PERFORM HTM2COB-SESS-GET

*> del session var
   MOVE "mydog" TO HTM2COB-SESS-VAR-NAME
   PERFORM HTM2COB-SESS-DEL
```

2.4.7 HTM2COB-SET-COOKIE

This module creates a cookie string. (See later in example 0026_cookie_visit and

in other cookie examples.)

Example:

```
CALL "HTM2COB-SET-COOKIE" USING LNK-HTM2COB-SET-COOKIE END-CALL
```

2.4.8 HTM2COB-GET-COOKIE

This module reads a saved cookie value from the HTM2COB-HTTP-COOKIE field. (See later in example 0026_cookie_visit and in other cookie examples.)

Example:

```
CALL "HTM2COB-GET-COOKIE" USING LNK-HTM2COB-GET-COOKIE END-CALL
```

2.4.9 HTM2COB-UPD-SESS-STR

This module updates SESSIONID in the HTM2COB-HTTP-COOKIE field.

2.4.10 imgcaptcha

This is a separate C module. It creates a PNG image captcha file. (See later in example 0024_sess_captcha.)

Example:

```
CALL "imgcaptcha" USING BY REFERENCE WS-CAPTCHA-FILE-NAME
      , BY REFERENCE WS-CAPTCHA-CHAR-TYPE
      , BY REFERENCE WS-CAPTCHA-TEXT
      RETURNING WS-RETVAL
END-CALL
```

2.4.11 imgscale

This is a separate C module. It converts and scales an image. (See below in example 0014_img_upload_scale.)

Example:

```
CALL "imgscale" USING BY REFERENCE WS-IMG-IN
      , BY REFERENCE WS-IMG-IN-TYPE
      , BY REFERENCE WS-IMG-OUT
      , BY REFERENCE WS-IMG-OUT-TYPE
      , BY REFERENCE WS-WIDTH-OUT
      RETURNING WS-RETVAL
END-CALL
```

2.5 Some security hints

Web development has to deal a lot more than traditional programming with security risks. Your application will be accessible to the whole world, and this includes hackers. **You have to stay up to date with potential security risks.** Read literature on secure web development. If you have a lack of knowledge in this field then don't change things that you don't know, the effects on the privacy could be severe.

- Always use HTTPS and not HTTP on your production web server. Hypertext Transfer Protocol Secure (HTTPS) is an extension of the

Hypertext Transfer Protocol (HTTP) for secure, encrypted communication over a computer network: <https://en.wikipedia.org/wiki/HTTPS>. Free SSL certificates can be acquired from the Let's Encrypt initiative.

- Never enable warning or error messages on your production web server since they could aid hackers or show parts of your program that have to remain unknown. Set the flag HTM2COB-DEBUG-FLAG on V-HTM2COB-DEBUG-NO and check all "display" in your COBOL code.
- It is important for security reasons that you set the parameters carefully. Set the right request method, number of fields, bytes of maximum memory and other options, depending on your needs in your server program. For example, if you do not need a file upload, or session handling, then do not allow these. Always set the maximum allowed size for fields and upload files. For images set the allowed image type.
- If possible, do not use the GET request method. GET requests include all required data in the browser URL which could leave sensitive data exposed to hackers. Instead of GET, you should use POST.
- You have to validate every data the client returns. A client side validation is not enough.
- Make sure that your application handles cases where POST or GET data is expected but not returned by your user (or a hacker).
- Use the "HTM2COB-SPEC-CHARS" filter before you write a text back. If you get something from the client, and you write it back later, then you should filter it to prevent cross site scripting (https://en.wikipedia.org/wiki/Cross-site_scripting).
- When you set a cookie, you have to specify the domain and subdomain of your application.
- Always use cookies with the "HttpOnly" parameter. This prevents javascript functions from manipulating the cookies you set.
- Use session-cookies, and do not save any other data in cookies, only the session-ID. Data stored in cookies is not secure.
- Cookie data from the client must also be validated, escaped and filtered. If you only store the session-id in the cookie then a hacker might have a harder time.
- Replace all Line-Feed and Carriage-Return chars in your cookie text before you set a cookie. (You should not use any special chars in cookie text, but maybe a client sends you special chars in a text, that you want to use later in cookie.)
- There are many ways to track users, see <https://en.wikipedia.org/wiki/Evercookie>
- **SQL injections are not filtered in this program!!** See https://en.wikipedia.org/wiki/SQL_injection . You have to use embedded SQL or Prepared SQL Statements.

- You MUST NOT store private files under the htdocs folders. These folders are directly accessible from the web. Examples of what NOT to store here: Session files, Data files for your applications, files uploaded by users and so on.
- You should prevent users from loading files higher directories than the one you are working in. Ensure that “..” expressions are filtered, else hackers might even steal your server’s passwd file by simply accessing “/../../etc/passwd”
- You have to set the proper permissions on each directory for access control.
- If you implement a search feature, then make sure that it filters out files that should not be accessible to your user.
- It’s best to regenerate the session-id every time the user visits a new page. If however you have lots of users this might be too much of burden on your server, then you can choose to generate a new session-id only after each n-th visit of your user. The minimum count of session-id generation you can do is one, when the user logs in. If you don’t regenerate the session-id, then hackers could use Session hijacking to log in: https://en.wikipedia.org/wiki/Session_hijacking
- The example programs here store the session files in the /tmp folder. **This is forbidden on any production system!** Session files are sensitive and should be private, the /tmp folder is accessible to any user of your server!
- Every separate application should use it’s own session files. The parameter files specify where the session files of the application are located.
- In case of unsuccessful logins, the login form must not give details about which of the entries was incorrect. This is also true for the user registration (account creation) form.
- Choose a sensible minimum length requirement for the passwords, and verify it on the server side. Require users to use lower and upper case characters, numbers and special characters.
- Do not store user input data from the outside world in any log files. Hackers could get entry by submitting special entries.
- Use salting and peppering for your login forms:
[https://en.wikipedia.org/wiki/Salt_\(cryptography\)](https://en.wikipedia.org/wiki/Salt_(cryptography))
[https://en.wikipedia.org/wiki/Pepper_\(cryptography\)](https://en.wikipedia.org/wiki/Pepper_(cryptography))

Never save your user’s password in clear text!! This includes not saving it in a file or in a database. Even the RAM is sensitive to attacks, so make sure to initialize the variable where the user’s clear-text password is stored after you generated the hash for it.

You **have** to use a hashing algorithm to ensure the security of your users.

3. Examples

3.1 Samples

3.1.1 0001_hello_get

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Hello GnuCOBOL world!</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-Type" content="text/css" />
</head>
<body>
This is a GET without values.<br><br>
<?cob display "Hello GnuCOBOL world!" end-display ?>
</body>
</html>

```

HTML with embedded COBOL

Get request method without values.

```

*> ***** begin user defined content HTM2COB-MAIN SECTION *****

DISPLAY "<!DOCTYPE HTML PUBLIC ""//W3C//DTD HTML 4.01 Transitional//EN"" END-DISPLAY
DISPLAY "  ""http://www.w3.org/TR/html4/loose.dtd"" END-DISPLAY
DISPLAY "<html>" END-DISPLAY
DISPLAY "<head>" END-DISPLAY
DISPLAY "<title>Hello GnuCOBOL world!</title>" END-DISPLAY
DISPLAY "<meta http-equiv=""content-type"" content=""text/html; charset=utf-8"" />" END-DISPLAY
DISPLAY "<meta http-equiv=""Content-Type"" content=""text/css"" />" END-DISPLAY
DISPLAY "</head>" END-DISPLAY
DISPLAY "<body>" END-DISPLAY
DISPLAY "This is a GET without values.<br><br>" END-DISPLAY

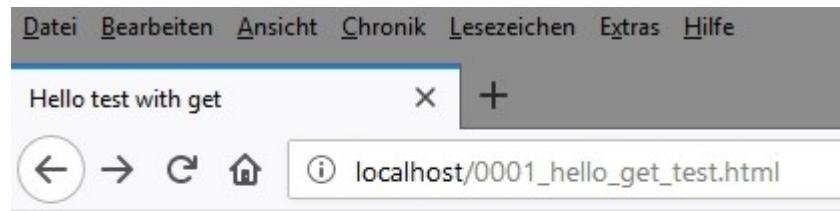
display "Hello GnuCOBOL world!" end-display

DISPLAY "</body>" END-DISPLAY
DISPLAY "</html>" END-DISPLAY

*> ***** end user defined content  HTM2COB-MAIN SECTION *****

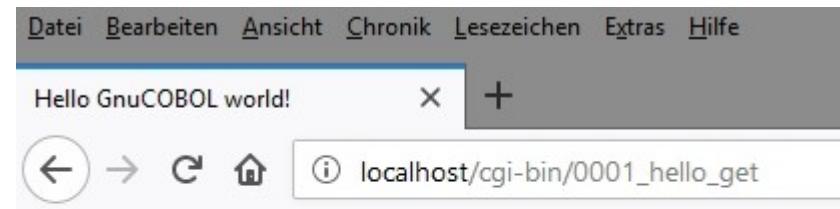
```

Generated COBOL code



Hello test with get

0001_hello_get_test



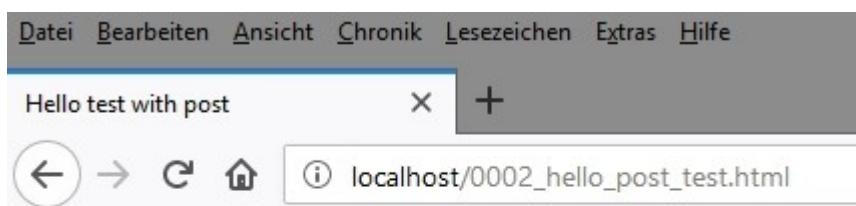
This is a GET without values.

Hello GnuCOBOL world!

0001_hello_get

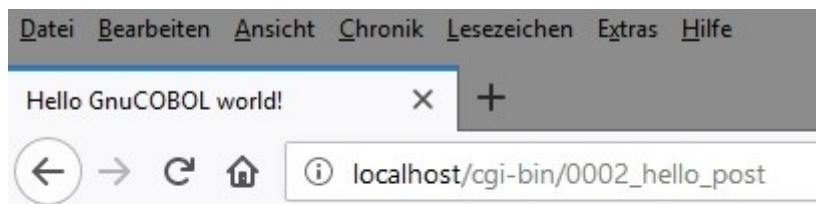
3.1.2 0002_hello_post

Post request method without values.



Hello test with post

0002_hello_post_test



This is a POST without values.

Hello GnuCOBOL world!

0002_hello_post

3.1.3 0003_hello_form_get

GET request method with values. If it possible, do not use the GET request method. **GET requests include all data in the browser URL**. Instead of GET, you should use POST because of security reasons.

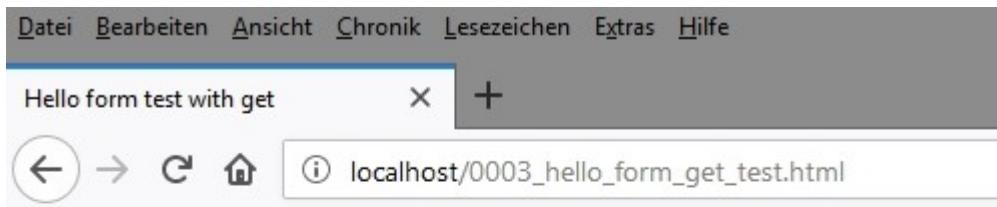
This is also an example of usage of the modules "HTM2COB-POST" and "HTM2COB-SPEC-CHARS". You have to call "HTM2COB-SPEC-CHARS" always, before writing something back to the client.

```
<?cob
    call "HTM2COB-POST" using "firstname", fname end-call
    call "HTM2COB-SPEC-CHARS" using fname, htm-fname end-call
    call "HTM2COB-POST" using "lastname", lname end-call
    call "HTM2COB-SPEC-CHARS" using lname, htm-lname end-call
?>
This is a GET with values.<br><br>

<?cob display "Hello GnuCOBOL world!" end-display ?>

<p>
First name: <?cob display trim(htm-fname) end-display ?><br>
Last name : <?cob display trim(htm-lname) end-display ?><br>
</p>
```

HTML with embedded COBOL



0003_hello_form_get_test



This is a GET with values.

Hello GnuCOBOL world!

First name: László
Last name : Erdős

0003_hello_form_get

3.1.4 0004_hello_form_post

POST request method with values.

This also is an example of using the modules "HTM2COB-POST" and "HTM2COB-SPEC-CHARS". You have to call "HTM2COB-SPEC-CHARS" always, before writing something back to the client.

The screenshot shows a web browser window with the following details:

- Header Bar:** Datei, Bearbeiten, Ansicht, Chronik, Lesezeichen, Extras, Hilfe.
- Title Bar:** Hello form test with post, with a close (X) and a plus (+) button.
- Address Bar:** localhost/0004_hello_form_post_test.html
- Content Area:**

Hello form test with post

First name:

Last name:

Buttons: Send, Zurücksetzen

0004_hello_form_post_test

The screenshot shows a web browser window with the following details:

- Header Bar:** Datei, Bearbeiten, Ansicht, Chronik, Lesezeichen, Extras, Hilfe.
- Title Bar:** Hello GnuCOBOL world!, with a close (X) and a plus (+) button.
- Address Bar:** localhost/cgi-bin/0004_hello_form_post
- Content Area:**

This is a POST with values.

Hello GnuCOBOL world!

First name: László
Last name : Erdős

0004_hello_form_post

3.1.5 0005_embedding

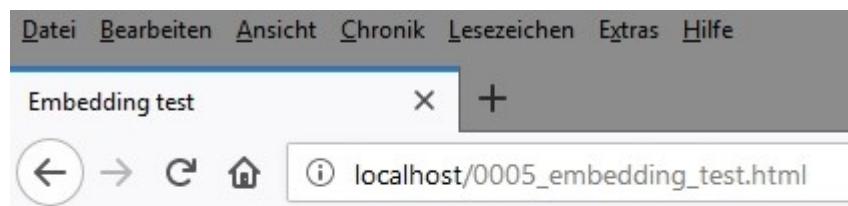
Example of embedding COBOL lines in HTML lines.

```

<body>
  The first line in HTML<br/>
  <%cob display "The second line in COBOL<br/>" %>
  The third line in HTML<br/>
<%cob
  display "The fourth line in COBOL<br/>"
  display "The fifth line in COBOL"
%>
</body>

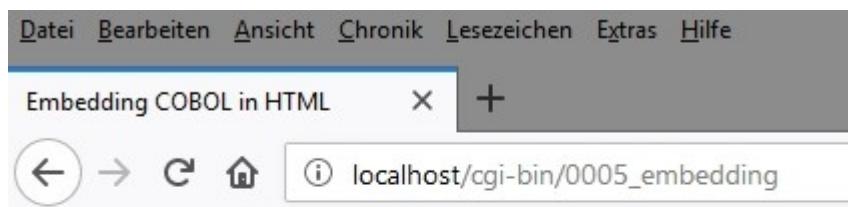
```

HTML with embedded COBOL



Embedding test

0005_embedding_test



The first line in HTML
 The second line in COBOL
 The third line in HTML
 The fourth line in COBOL
 The fifth line in COBOL

0005_embedding

3.1.6 0006_comment

Example of COBOL comments in the embedded COBOL.

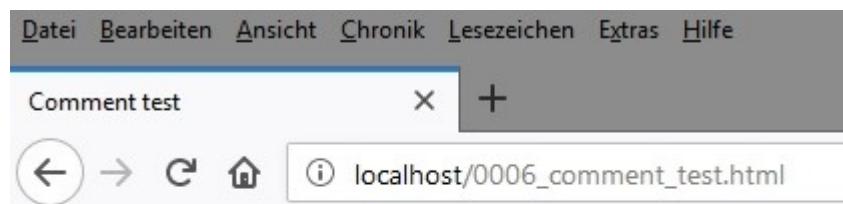
```
<body>
<?cob-ws
*> This is a COBOL comment in column 1.
 01 text-1 pic x(30) value "Welcome in Hotel Moon!".
 01 price  pic 9(3).
 01 price-disp pic zz9. *> comment to end of line
?>

<?cob-ws*> This is a COBOL comment in column 1. ?>

<?cob display "This is the start of the program.<br>" *> comment to end of line ?>
<?cob
*> This is a COBOL comment in column 1.
  display "This is the end of the program.<br>"
    *> comment to end of line
?>

<?cob*> This is a COBOL comment in column 1. ?>
</body>
```

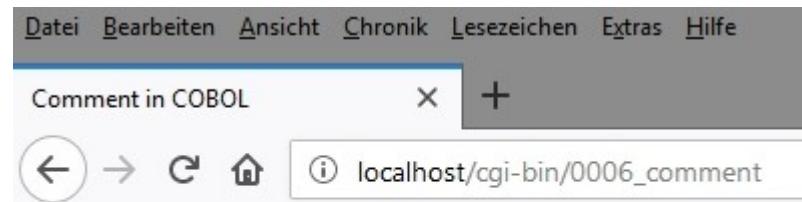
HTML with embedded COBOL



Comment test

[Send](#)

0006_comment_test



This is the start of the program.
This is the end of the program.

0006_comment

3.1.7 0007_variables

You can define your variables everywhere with "<?cob-ws", they will be copied in WORKING-STORAGE SECTION.

```

<body>
  <?cob-ws
    *> You can define your variables everywhere,
    *> they will be copied in WORKING-STORAGE SECTION.
    01 text-1 pic x(30) value "Welcome in Hotel Triton!".
    01 price  pic 9(3).
    01 price-disp pic zz9.
  ?>

  <?cob compute price = room + breakfast
    move price to price-disp ?>

  <?cob-ws 01 room pic 9(3) value 50. ?>

  <h1><font color="#FF0000"><?cob display text-1 ?></font></h1>
  <p><h2>
    Price with breakfast: <?cob display price-disp ?> Euro
  </h2></p>
  <hr color="#00FF00">

  <?cob-ws 01 breakfast pic 9(3) value 8. ?>

</body>

```

HTML with embedded COBOL

```

*> ***** begin user defined content WORKING-STORAGE SECTION *****

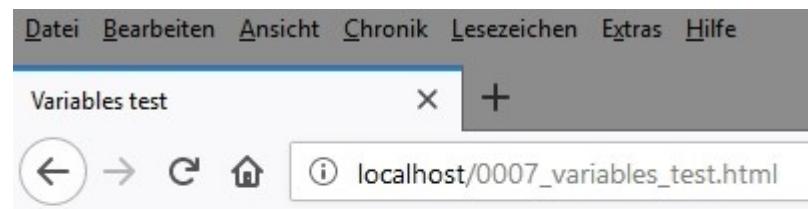
*> You can define your variables everywhere,
*> they will be copied in WORKING-STORAGE SECTION.
  01 text-1 pic x(30) value "Welcome in Hotel Triton!".
  01 price  pic 9(3).
  01 price-disp pic zz9.

  01 room pic 9(3) value 50.
  01 breakfast pic 9(3) value 8.

*> ***** end user defined content      WORKING-STORAGE SECTION *****

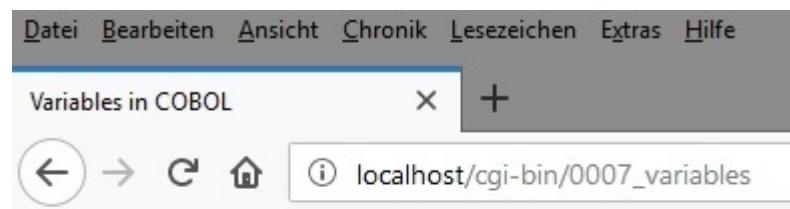
```

Generated COBOL code



Variables test

0007_variables_test



Welcome in Hotel Triton!

Price with breakfast: 58 Euro

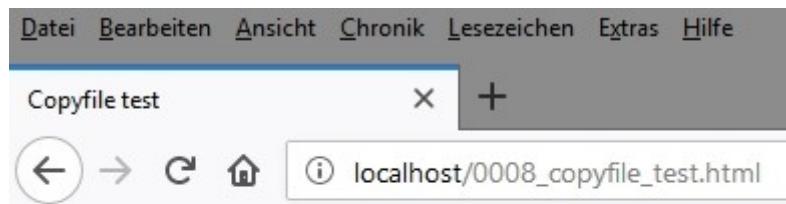
0007_variables

3.1.8 0008_copyfile

You can use copy files.

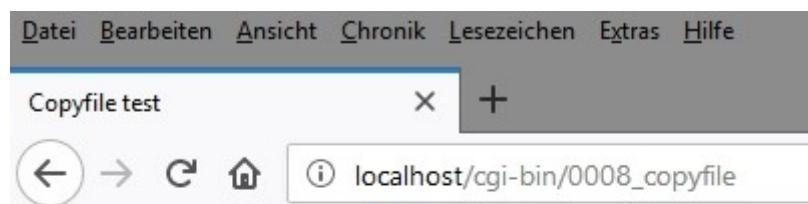
```
<?cob-ws
 *> you can use copy files also
    copy "cobcopy.cpy".
?>
```

HTML with embedded COBOL



Copyfile test

0008_copyfile_test



Welcome in Hotel Moon!

Price with breakfast: 42 Euro

0008_copyfile

3.1.9 0009_section

COBOL sections after the HTML body will be copied after the HTM2COB-MAIN section in the generated COBOL code.

```

<?cob
*> COBOL sections after html body will be copied after the HTM2COB-MAIN
*> section in the generated COBOL code.

LIST-BOOK SECTION.

PERFORM VARYING WS-IND FROM 1 BY 1
      UNTIL WS-IND > C-MAX-IND

      INITIALIZE WS-TAB-FIELDS
      UNSTRING WS-BOOKS-LINE (WS-IND)  DELIMITED BY ";""
          INTO WS-SALES
              WS-TITLE
              WS-AUTHOR

      DISPLAY
          "<tr>"
          "<td>"WS-IND"</td>"
          "<td>"WS-SALES"</td>"
          "<td>"WS-TITLE"</td>"
          "<td>"WS-AUTHOR"</td>"
          "</tr>"
      END-DISPLAY
      END-PERFORM

      EXIT SECTION .
?>

```

Section outside of HTML body.

```

/*> COBOL sections after html body will be copied after the HTM2COB-MAIN
/*> section in the generated COBOL code.

LIST-BOOK SECTION.

PERFORM VARYING WS-IND FROM 1 BY 1
      UNTIL WS-IND > C-MAX-IND

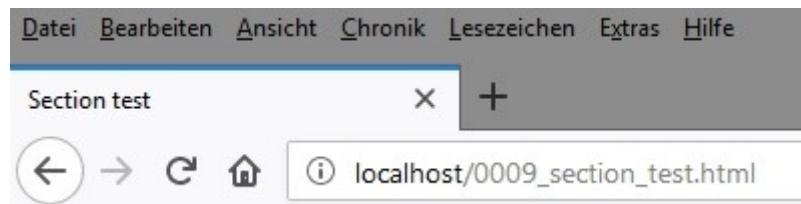
INITIALIZE WS-TAB-FIELDS
UNSTRING WS-BOOKS-LINE(WS-IND) DELIMITED BY ";" 
      INTO WS-SALES
      WS-TITLE
      WS-AUTHOR

DISPLAY
      "<tr>" 
      "<td>"WS-IND"</td>" 
      "<td>"WS-SALES"</td>" 
      "<td>"WS-TITLE"</td>" 
      "<td>"WS-AUTHOR"</td>" 
      "</tr>" 
END-DISPLAY
END-PERFORM

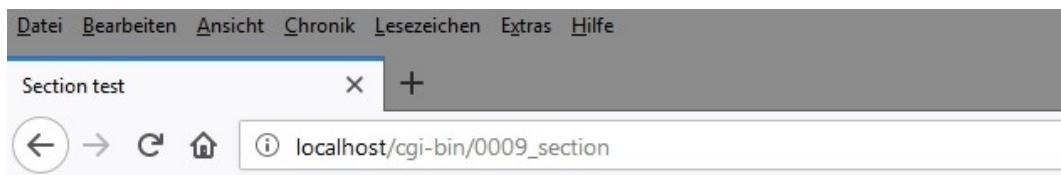
EXIT SECTION .

```

Generated COBOL code



0009_section_test



Top 10 Most Read Books In The World

Nr.	Approximate sales	Title	Author
01	3.9 Billion Copies	The Bible	---
02	820 Million Copies	Quotations from the Works of Mao Tse-tung	By Mao Tse-tung
03	400 Million Copies	Harry Potter	By J. K. Rowling
04	103 Million Copies	Lord of the Rings	By J. R. R. Tolkien
05	65 Million Copies	The Alchemist	By Paulo Coelho
06	57 Million Copies	The Da Vinci Code	By Dan Brown
07	43 Million Copies	Twilight - The Saga	By Stephenie Meyer
08	33 Million Copies	Gone With the Wind	By Margaret Mitchell
09	30 Million Copies	Think and Grow Rich	By Napoleon Hill
10	27 Million Copies	Diary of Anne Frank	By Anne Frank

0009_section

3.1.10 0010_call

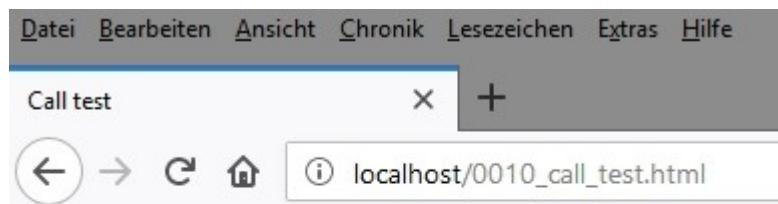
Example for calling COBOL modules.

```
<?cob
  DISPLAY "<h2>Start main</h2>"

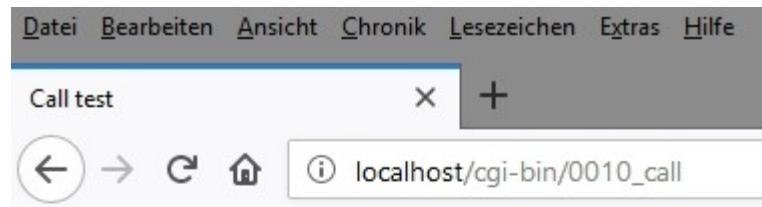
  SET SUB-PROG-1 TO TRUE
  CALL SUB-PROG-NAME END-CALL

  SET SUB-PROG-2 TO TRUE
  CALL SUB-PROG-NAME USING TEST-DATA
                           TEST-DATA-LEN
  END-CALL
- ?>
```

COBOL call



0010_call_test



Dynamic CALL test

Start main

Start 0010_sub1

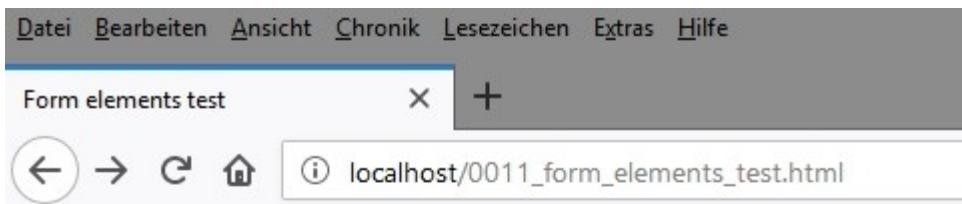
Start 0010_sub2

LNK-DATA: abcde12345
LNK-DATA-LEN: +0000000010

0010_call

3.1.11 0011_form_elements

HTML form elements with Chinese characters, a demonstration of the utf-8 support.



Form elements test

first name: 拉斯洛

surname: 鄂爾多斯

gender: male
 female

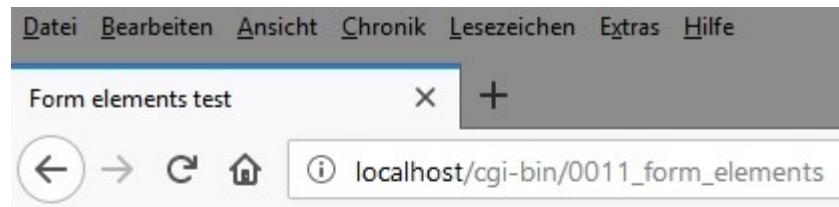
skills: COBOL
 C/C++
 java
 ASM

your city: New York ▾

comment: 側經意責家方家開討店暖育田序載社軛線宇。

hidden data:

0011_form_elements_test



0011_form_elements

3.1.12 0012_env_vars

This example prints the CGI Environment Variables. It is also an example of using the module "HTM2COB-ENV"

```

<?cob
LIST-ENV-VARS SECTION.

PERFORM VARYING WS-IND FROM 1 BY 1
      UNTIL WS-IND > C-MAX-IND

INITIALIZE WS-TAB-FIELDS
UNSTRING WS-ENV-LINE(WS-IND) DELIMITED BY ";""
INTO WS-ENV-NAME
      WS-ENV-DESC

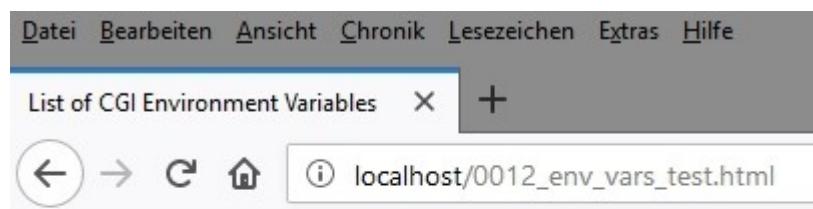
MOVE FUNCTION TRIM(WS-ENV-NAME) TO LNK-ENV-NAME OF LNK-HTM2COB-ENV
CALL "HTM2COB-ENV" USING LNK-HTM2COB-ENV END-CALL
MOVE FUNCTION TRIM(LNK-ENV-VALUE OF LNK-HTM2COB-ENV) TO WS-ENV-TXT

DISPLAY
      "<tr>"
      "<td>WS-IND</td>"
      "<td>WS-ENV-NAME</td>"
      "<td>WS-ENV-DESC</td>"
      "<td>WS-ENV-TXT</td>"
      "</tr>"
END-DISPLAY
END-PERFORM

EXIT SECTION .
?>

```

Call "HTM2COB-ENV"



List of CGI Environment Variables

0012_env_vars_test

List of CGI Environment Variables

Nr.	Env. Var	Desc.	Value
0001	DOCUMENT_ROOT	The root directory of your server	/srv/www/htdocs
0002	HTTP_COOKIE	The visitor's cookie, if one is set	
0003	HTTP_HOST	The hostname of the page being attempted	localhost
0004	HTTP_REFERER	The URL of the page that called your program	http://localhost/0012_env_vars_test.html
0005	HTTP_USER_AGENT	The browser type of the visitor	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
0006	HTTPS	on, if the program is being called through a secure server	
0007	PATH	The system path your server is running under	usr/local/bin:/usr/bin:/cygdrive/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/cygdrive/c/ProgramData/Oracle/Java/javapath:/cygdrive/c/Windows/System32:/cygdrive/c/Windows:/cygdrive/c/Windows/System32/WindowsPo
0008	QUERY_STRING	The query string	
0009	REMOTE_ADDR	The IP address of the visitor	127.0.0.1
0010	REMOTE_HOST	The hostname of the visitor (if your server has reverse-name-lookups on, otherwise this is the IP address again)	
0011	REMOTE_PORT	The port the visitor is connected to on the web server	56946
0012	REMOTE_USER	The visitor's username (for .htaccess-protected pages)	
0013	REQUEST_METHOD	GET or POST	POST
0014	CONTENT_LENGTH	length of content	0
0015	CONTENT_TYPE	type of content	application/x-www-form-urlencoded
0016	REQUEST_URI	The interpreted pathname of the requested document or CGI (relative to the document root)	/cgi-bin/0012_env_vars
0017	SCRIPT_FILENAME	The full pathname of the current CGI	/srv/www/cgi-bin/0012_env_vars
0018	SCRIPT_NAME	The interpreted pathname of the current CGI (relative to the document root)	/cgi-bin/0012_env_vars
0019	SERVER_ADMIN	The email address for your server's webmaster	you@example.com
0020	SERVER_NAME	Your server's fully qualified domain name (e.g. www.name.com)	localhost
0021	SERVER_PORT	The port number your server is listening on	80
0022	SERVER_SOFTWARE	The server software you're using (e.g. Apache 1.3)	Apache/2.4.29 (Unix)

0012_env_vars

3.1.13 0013_img_upload

Image upload example. You have to set the file path for the uploaded files, and set the allowed file types for image uploads in the parameter file.
0013_img_upload_param.cpy.

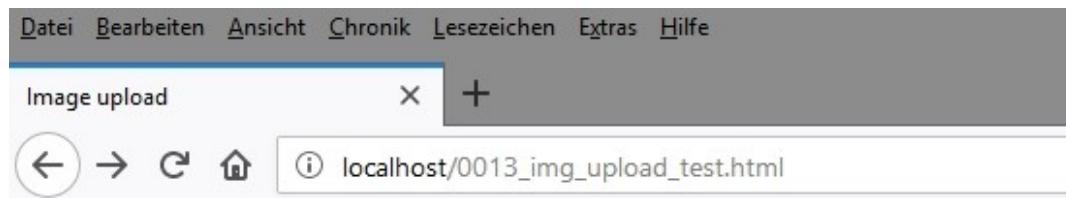


Image upload

For image upload you have to use enctype="multipart/form-data" in form request!

Image description1:	Vegan csabai kolbasz
upload1:	Durchsuchen... csabai_kolbasz.jpg
Image description2:	Trapper, the original jeans
upload2:	Durchsuchen... Trapper.jpg
<input type="button" value="Send"/> <input type="button" value="Zurücksetzen"/>	

0013_img_upload_test

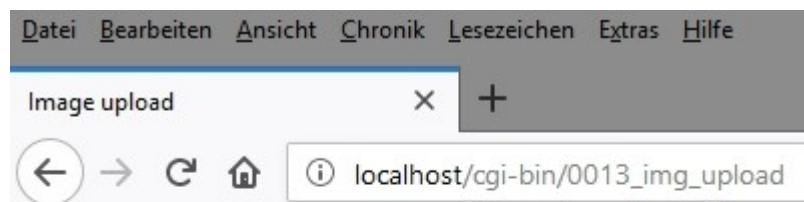
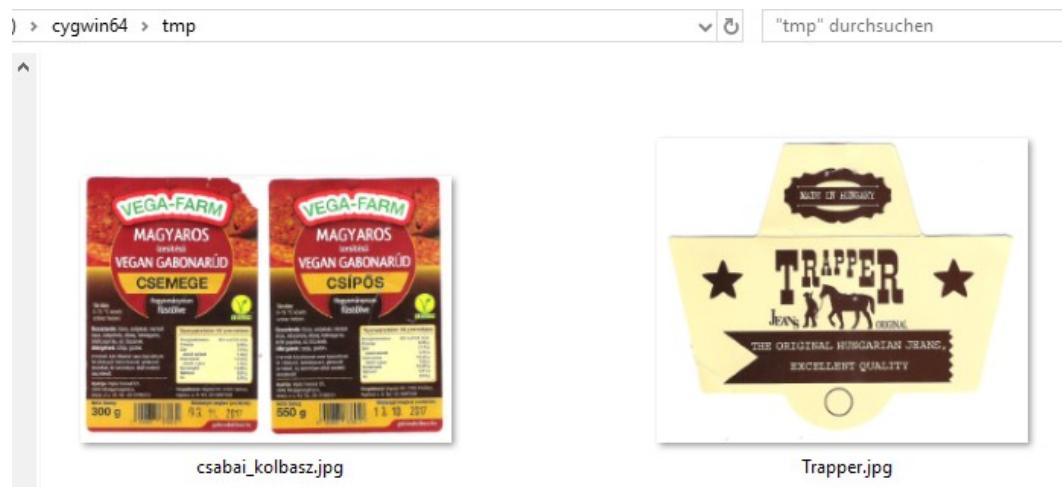


Image upload

Image description1: Vegan csabai kolbasz
Image upload1: csabai_kolbasz.jpg

Image description2: Trapper, the original jeans
Image upload2: Trapper.jpg

0013_img_upload



The uploaded images under /tmp

3.1.14 0014_img_upload_scale

Image upload, conversion, scaling and copy example. You have to set the file path for the uploaded file, and allowed file types for image uploads in the parameter file: 0014_img_upload_scale_param.cpy.

After upload the imgscale.c module will be used. With this module you can convert, scale and copy the uploaded image. The new image will be shown on the response page.

```

<body>
<?cob PERFORM GET-VALUES
        PERFORM CREATE-OUT-NAME
        PERFORM IMGSCALE-AND-COPY
?>

<h2>Image upload and scale</h2>
<p>
Image description: <?cob display trim(htm-filedesc) end-display ?><br>
Image upload: <?cob display trim(htm-upload) end-display ?><br>
<br>
Scaled Image: <?cob display trim(WS-OUT-NAME) end-display ?><br>
<?cob display "<img src='.. /img /"trim(WS-OUT-NAME) "'>" end-display ?><br>
</p>
</body>

```

Use of imgscale.c module

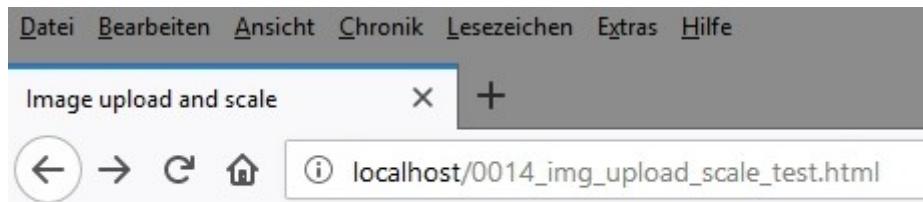


Image upload and scale

Image description: <input type="text" value="Cyborg test"/>
PNG upload (max. 750 Kb): <input type="text" value="Durchsuchen..."/> <input type="text" value="cyborg.png"/>
<input type="button" value="Send"/> <input type="button" value="Zurücksetzen"/>

0014_img_upload_scale_test

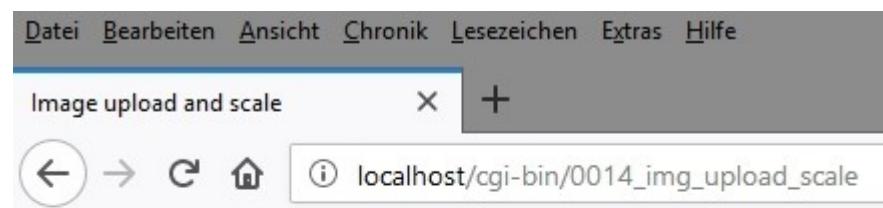


Image upload and scale

Image description: Cyborg test
Image upload: cyborg.png

Scaled Image: cyborg.jpg



0014_img_upload_scale

3.1.15 0015_multiple_menu

Multiple selection menu example. It is also an example of using the module "HTM2COB-POST-MULTI".

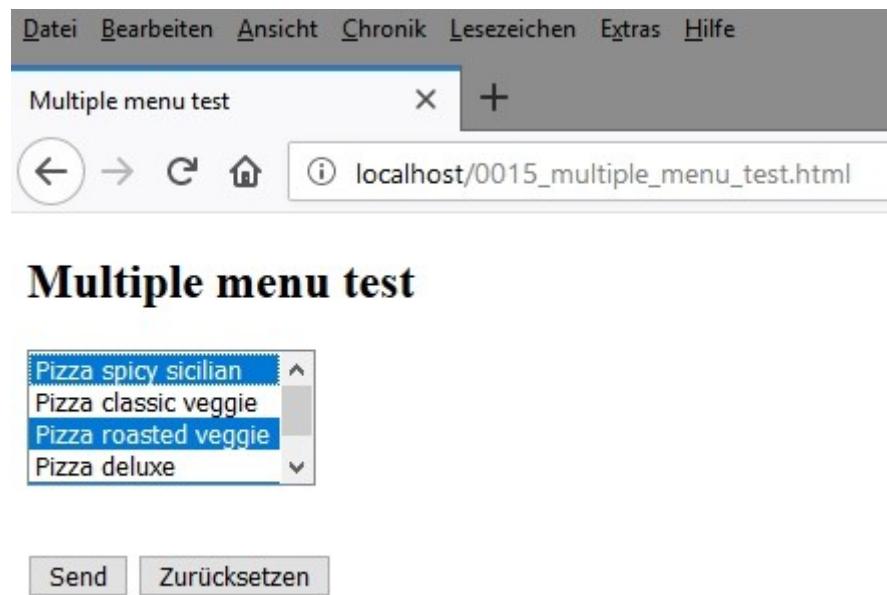
```

<?cob
 *> use ; as a separator char
    call "HTM2COB-POST-MULTI" using "pizza", ";", ws-pizza end-call
    call "HTM2COB-SPEC-CHARS" using ws-pizza, htm-pizza end-call
?>

<p>
Your selected pizzas: <?cob display trim(htm-pizza) end-display ?><br>
(separator char: ";" )<br>
</p>

```

HTML with embedded COBOL



0015_multiple_menu_test



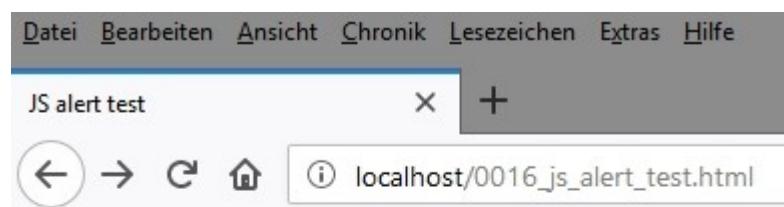
0015_multiple_menu

3.1.16 0016_js_alert

An example of Javascript: using onSubmit to validate inputted text.

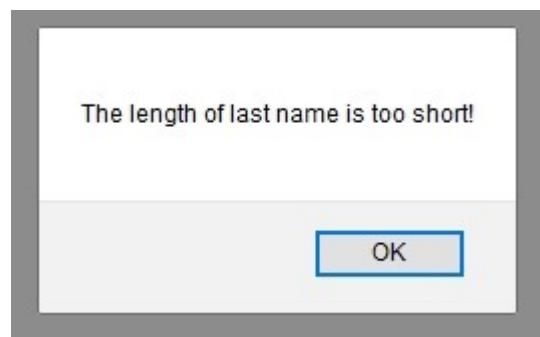
```
<h2>JS alert test</h2>
<p>
<form name="formname" onSubmit="return length_check(); " action="/cgi-bin/0016_
<table border="0" cellpadding="0" cellspacing="4">
<tr>
<td align="right">First name (2-20 chars):</td>
<td><input name="firstname" type="text" size="20" maxlength="20"></td>
</tr>
```

HTML form with onSubmit



First name (2-20 chars):	<input type="text" value="László"/>
Last name (2-20 chars):	<input type="text" value="E"/>
<input type="button" value="Send"/> <input type="button" value="Zurücksetzen"/>	

0016_js_alert_test



0016_js_alert_test_message

3.1.17 0017_more_button

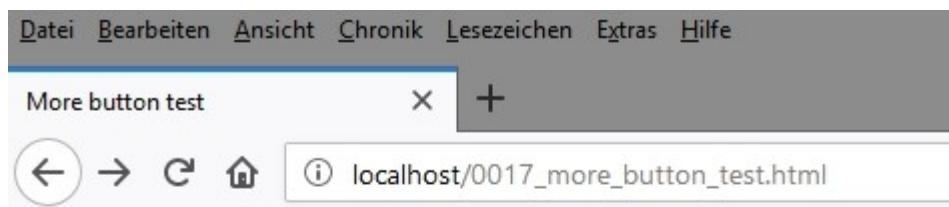
This is an example for more buttons in a form.

```

<tr>
  <td align="left">Editor</td>
  <td><input name="pw_editor" type="password" size="20" maxlength="20"></td>
  <td><input type="button" value="Login" onClick="user_login('Editor')"></td>
</tr>
<tr>
  <td align="left">Admin</td>
  <td><input name="pw_admin" type="password" size="20" maxlength="20"></td>
  <td><input type="button" value="Login" onClick="user_login('Admin')"></td>
</tr>
<tr>
  <td align="left">SuperAdmin</td>
  <td><input name="pw_super_admin" type="password" size="20" maxlength="20"></td>
  <td><input type="button" value="Login" onClick="user_login('SuperAdmin')"></td>
</tr>

```

onClick with user_login() java script function



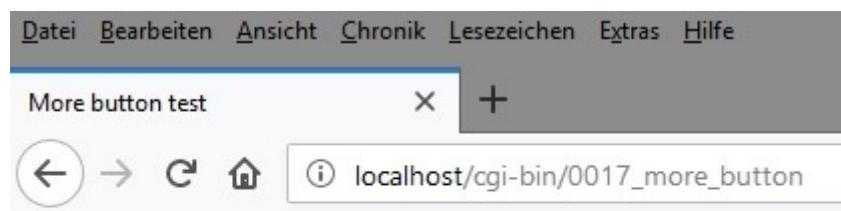
More button test

Name	Password	Registered users
Editor		Login
Admin	*****	Login
SuperAdmin		Login

(Passwords: Editor/qwerty1, Admin/qwerty2, SuperAdmin/qwerty3)

[Login as guest](#) [Zurücksetzen](#)

0017_more_button_test



More button test

Display information for Admin...

Display information for Guest...

0017_more_button

3.1.18 0018_link_submit

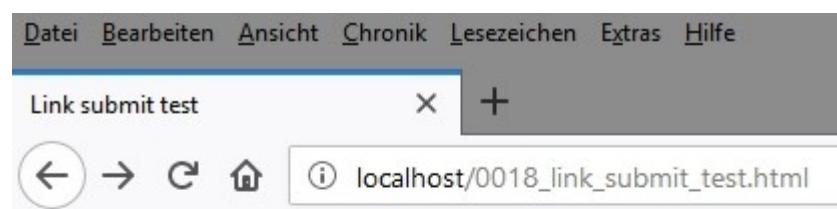
This is an example for a link submit.

```

</form>
</p>
-<p>
<a href="javascript:document.hello_form.submit();">Click for hello!</a>
-</p>

```

Hyperlink submit



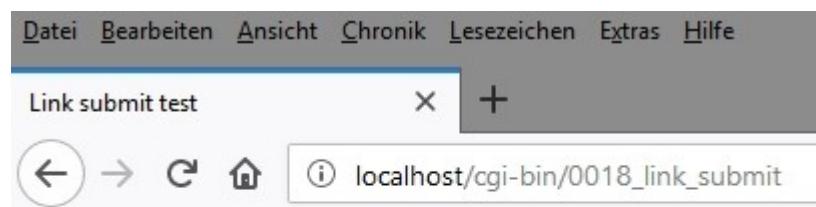
Link submit test

First name: László

Last name: Erdős

[Click for hello!](#)

0018_link_submit_test



Link submit test

Hello László Erdős!

0018_link_submit

3.1.19 0019_menu

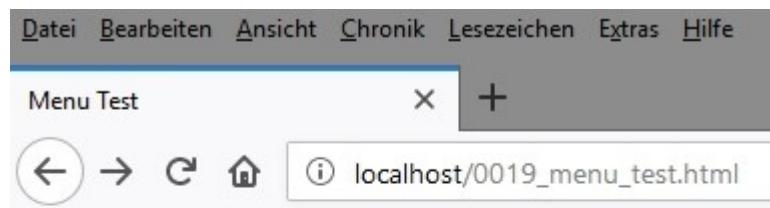
This is an example for an ` ` CSS menu.

```

<ul id="menu">
    <li><a href="/cgi-bin/0019_menu_1">Menu1</a></li>
    <li><a href="/cgi-bin/0019_menu_2">Menu2</a>
        <ul>
            <li><a href="/cgi-bin/0019_submenu_21">SubMenu21</a></li>
            <li><a href="/cgi-bin/0019_submenu_22">SubMenu22</a></li>
            <li><a href="/cgi-bin/0019_submenu_23">SubMenu23</a></li>
        </ul>
    </li>
    <li><a href="/cgi-bin/0019_menu_3">Menu3</a>
</ul>

```

` ` menu definition

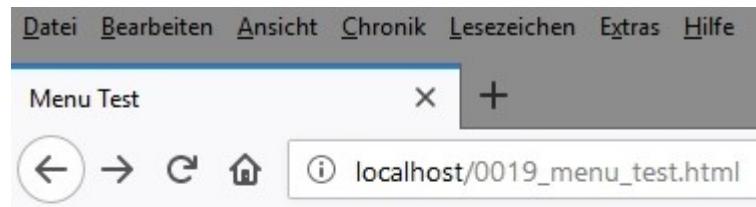


Menu Test

Menu1	Menu2	Menu3
-------	-------	-------

Menu Test page...

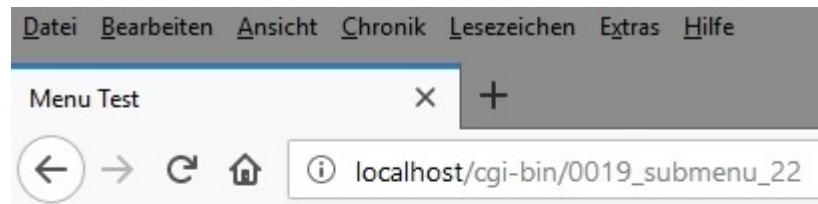
0019_menu_test



Menu Test

Menu1	Menu2	Menu3
	SubMenu21	
Menu Test	SubMenu22	
	SubMenu23	

0019_menu_test_select



Menu Test

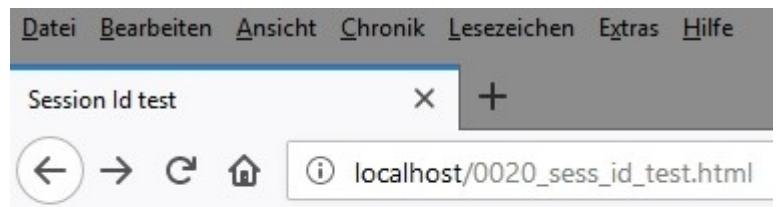
Menu1	Menu2	Menu3
-------	-------	-------

Sub-Menu22 page...

0019_submenu_22

3.1.20 0020_sess_id

This example uses the "HTM2COB-SESSION-ID" module to generate 5 session-ids.



Session Id test

Send

0020_sess_id_test

```

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Session Id test x +
← → ⌂ ⌂ localhost/cgi-bin/0020_sess_id ...
... ☆ Suchen ↴

Input IP addr.: ::1

Input user agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0

Call: +0000000001
SESSION-ID-HEX:
580C3F02696043829616722D5B52F63DF491C0BA42DB6E87D9D7AB443E9026120E17A2484AC132F4FE41321436FF962DDA7A13F602E74B8C77FF3D67F6CEAC6D
USER-AGENT-HASH-HEX:
3FE6BD58DE60F7E37F954BD80A68614A520035171D230A9E9001BDA90A1D95350E07FA0607E33A8785C103F94951F99CFEA28B0FE973ABC535DB4ED9E95C84F3

Call: +0000000002
SESSION-ID-HEX:
C030900BBCBD4A7544F43D73278595CA0397B6B2F96F1EFC89E476C7402139DC5EF8FAC13F846789C563003AB911D51B112D5296C8BBDE0FD9B281B3353C2406
USER-AGENT-HASH-HEX:
3FE6BD58DE60F7E37F954BD80A68614A520035171D230A9E9001BDA90A1D95350E07FA0607E33A8785C103F94951F99CFEA28B0FE973ABC535DB4ED9E95C84F3

Call: +0000000003
SESSION-ID-HEX:
621C290F67E3C638E148437D67788C13E3C9C855D4B2C898EDC45BBE10A44C2D1FFD64DA97090C7AA13365791362CA85D04E5AF687B6785A3FD08B8D0C7F8D3C
USER-AGENT-HASH-HEX:
3FE6BD58DE60F7E37F954BD80A68614A520035171D230A9E9001BDA90A1D95350E07FA0607E33A8785C103F94951F99CFEA28B0FE973ABC535DB4ED9E95C84F3

Call: +0000000004
SESSION-ID-HEX:
136312E08FC39890C62CA8594BDD13EDB970BCFEE70572EFA1C0AD58A5E84A845B9C4AD8CA770E2F1130B6C499C4B19CFD52689E06F304C80104A61AE936E54F
USER-AGENT-HASH-HEX:
3FE6BD58DE60F7E37F954BD80A68614A520035171D230A9E9001BDA90A1D95350E07FA0607E33A8785C103F94951F99CFEA28B0FE973ABC535DB4ED9E95C84F3

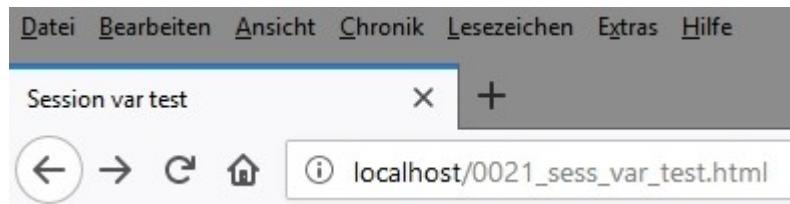
Call: +0000000005
SESSION-ID-HEX:
1B23EEC0EBCB0A709F9BB5F020FE2D11F2355F5291FBE8FEF7962FE4A43200FE1180E5BBDEE92F76E69EB1DE8828AE1803F68F8373675FB7C354C28F30DC587
USER-AGENT-HASH-HEX:
3FE6BD58DE60F7E37F954BD80A68614A520035171D230A9E9001BDA90A1D95350E07FA0607E33A8785C103F94951F99CFEA28B0FE973ABC535DB4ED9E95C84F3

```

0020_sess_id

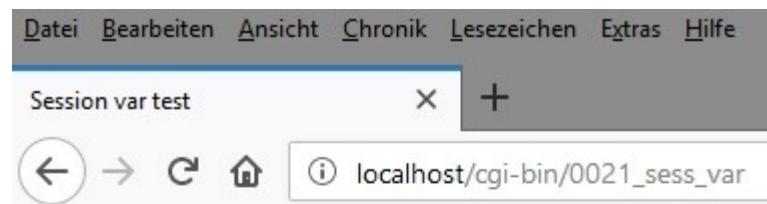
3.1.21 0021_sess_var

This example uses some session relevant sections: "HTM2COB-SESS-START", "HTM2COB-SESS-SET", "HTM2COB-SESS-GET" and "HTM2COB-SESS-DEL". Session data will be written in the session files: "/tmp/0021_session.dat" and "/tmp/0021_session_var.dat".



Session var test

0021_sess_var_test



Step 1: start session

Step 2: set session var

HTM2COB-SESS-VAR-NAME: mydog
HTM2COB-SESS-VAR-VALUE: Mrgo

Step 3: get session var

HTM2COB-SESS-VAR-NAME: mydog
HTM2COB-SESS-VAR-VALUE: Mrgo

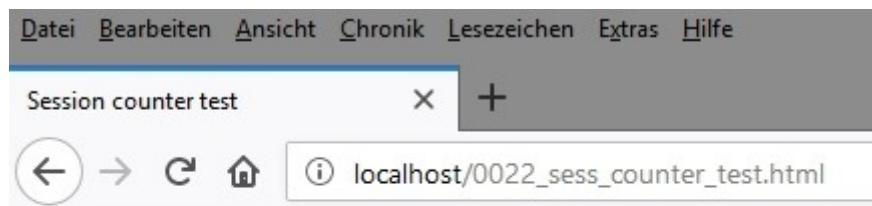
Step 4: del session var

HTM2COB-SESS-VAR-NAME: mydog

0021_sess_var

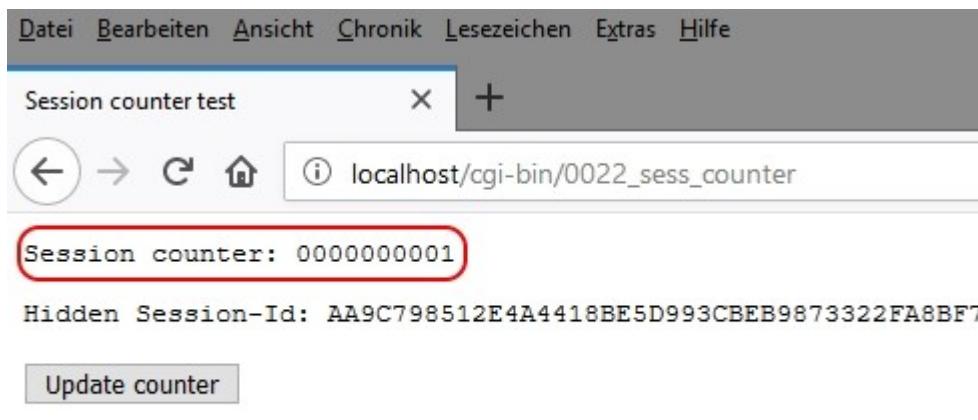
3.1.22 0022_sess_counter

This is an example for a counter. The counter value will be saved in the session file, and it will be updated after every call.



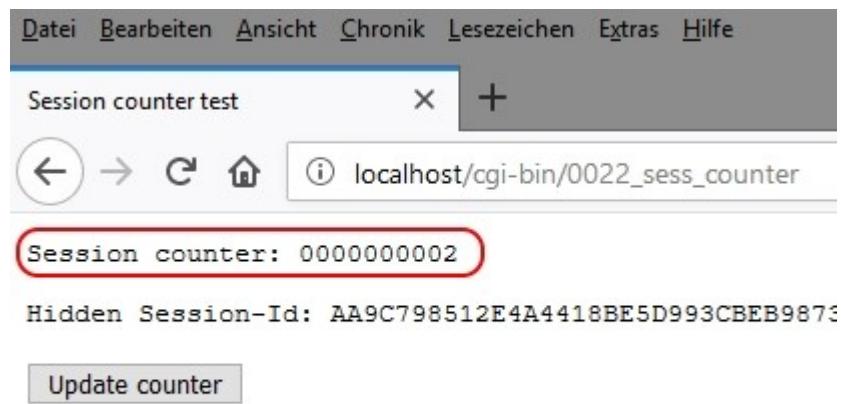
Session counter test

0022_sess_counter_test



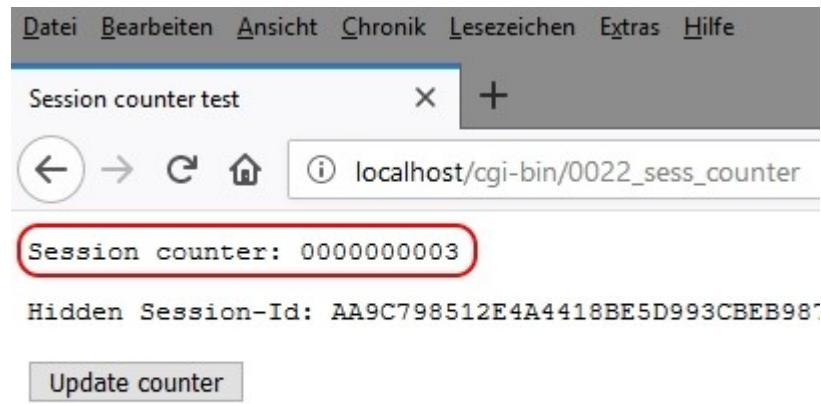
[Session counter Start page](#)

0022_sess_counter_1



[Session counter Start page](#)

0022_sess_counter_2

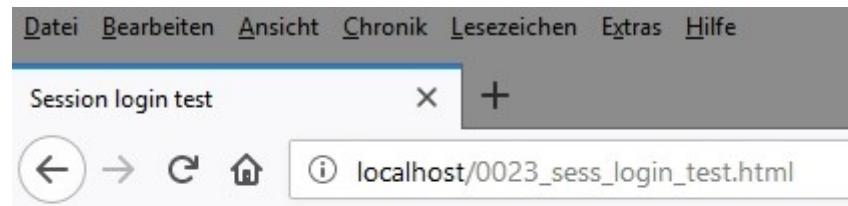


[Session counter Start page](#)

0022_sess_counter_3

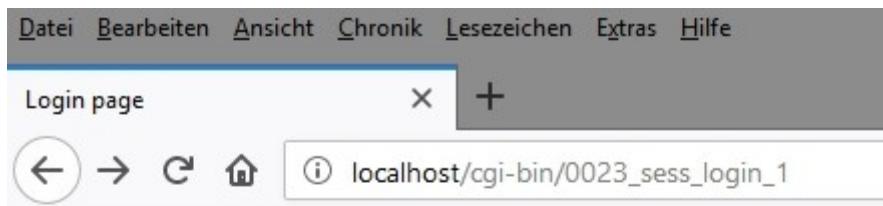
3.1.23 0023_sess_login

This is a simple example for session tracking with login. The user name will be saved in the session file after successful login. The existence of this user name will be checked on the following pages. If the user visits the login page later, then a new session-id will be created.



Session login test

0023_sess_login_test



HTM2COB-HIDDEN-SESSION-ID: DF7258233D32CFAE7F392EAABE4

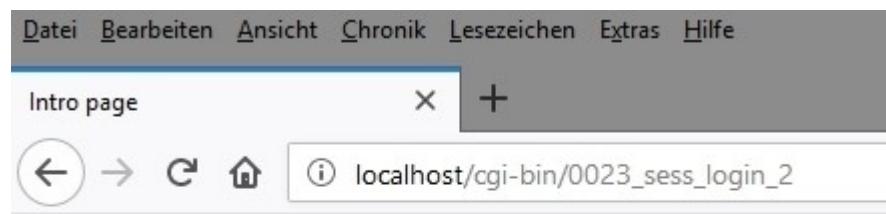
Login page

Name:	<input type="text" value="user1"/>
Password:	<input type="password" value="*****"/>

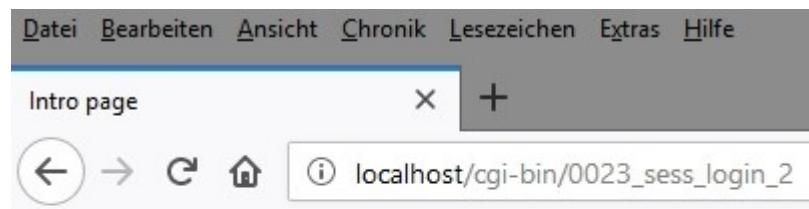
Valid names and passwords: "admin/qwerty" or "user1/pw123".

[Session login start page](#)

0023_sess_login_1



0023_sess_login_2_unsuccessful



Intro page

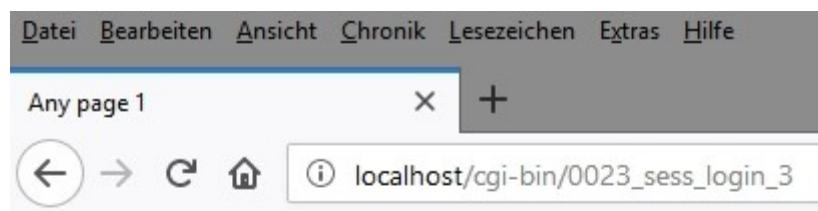
Hello user1

[Any page 1](#)

[Logoff](#)

[Session login start page](#)

0023_sess_login_2



HTM2COB-HIDDEN-SESSION-ID: 8A643FA31EFFBE301CECA99

Any page 1

Hello user1

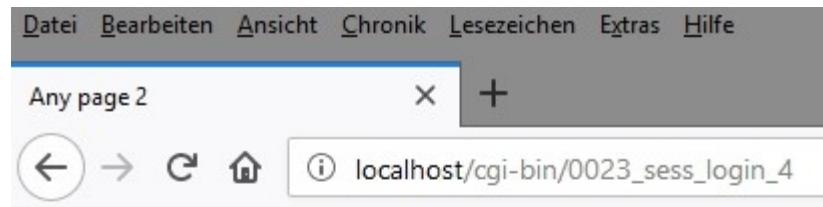
[Any page 2](#)

[Intro page](#)

[Logoff](#)

[Session login start page](#)

0023_sess_login_3



HTM2COB-HIDDEN-SESSION-ID: 8A643FA31EFFBE301CECA99

Any page 2

Hello user1

[Any page 1](#)

[Intro page](#)

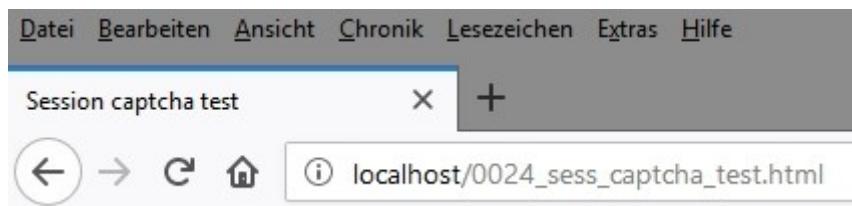
[Logoff](#)

[Session login start page](#)

[0023_sess_login_4](#)

3.1.24 0024_sess_captcha

This is a captcha example. The text of the image captcha will be saved in the session. This text will be checked at submit. It is also an example of using the module "imgcaptcha".



Session captcha test

0024_sess_captcha_test

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Session captcha test X +

← → ⌂ ⌂ localhost/cgi-bin/0024_sess_captcha_1

Write your comment

Your name: Hit Girl

Comment: Real programmers don't write specs. Users should consider themselves lucky to get any programs at all and take what they get.

Real programmers don't comment their code. If it was hard to write, it should be hard to read.

Real programmers never work 9 to 5. If any real programmers are around at 9 am, it's because they were up all night.

 CAPTCHA number: ↑

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

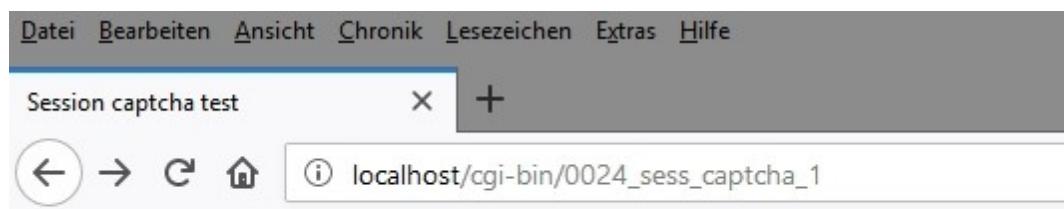
Session captcha test X +

← → ⟳ 🏠 ⓘ localhost/cgi-bin/0024_sess_captcha_2

Captcha verification failed, please try again.

Session captcha test page

0024_sess_captcha_2_unsuccessful



Write your comment

Your name: Hit Girl

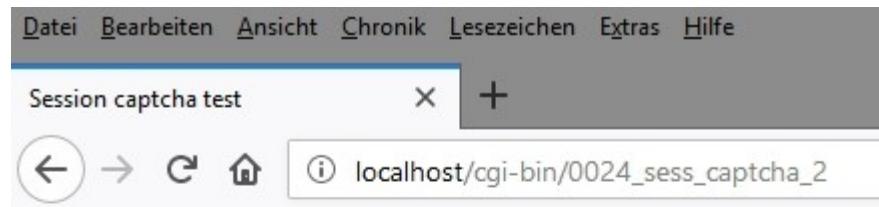
Comment: Real programmers don't write specs. Users should consider themselves lucky to get any programs at all and take what they get.
Real programmers don't comment their code. If it was hard to write, it should be hard to read.
Real programmers never work 9 to 5. If any real programmers are around at 9 am, it's because they were up all night.



CAPTCHA number: 282388

[Session captcha test start page](#)

0024_sess_captcha_1



Your comment was accepted!

Hello Hit Girl

Your comment: Real programmers don't write specs. Users should
was hard to write, it should be hard to read. Real programmers nev

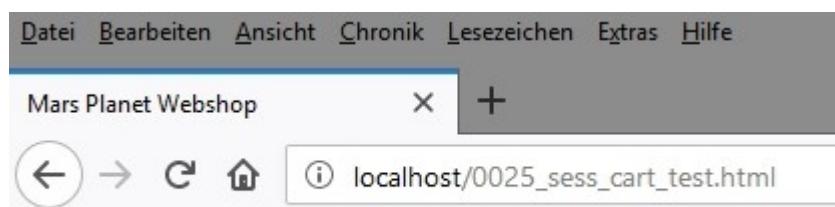
[Session captcha test page](#)

[Session captcha test start page](#)

[0024_sess_captcha_2](#)

3.1.25 0025_sess_cart

This is a shopping cart example. You can add or delete items from several categories to the shopping cart. These things are stored in the session file. At the end you can do a checkout and pay.

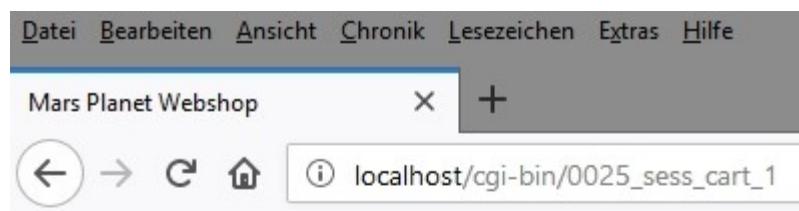


Welcome to the Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

[Start](#)

0025_sess_cart_test



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9CFB4C

Mars Planet Webshop

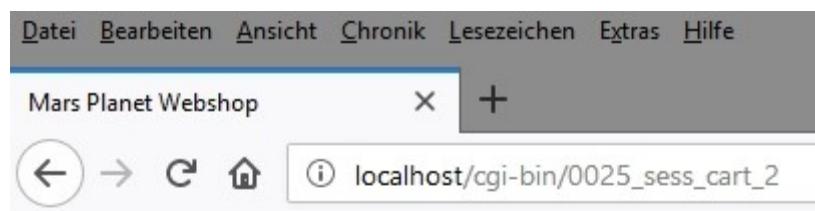
Sorry, there is no return ticket back to Earth...

Choose a category

Nr.	Category	Select Category
1	Vital items	Go to Category...
2	Robots and Mars-Rovers	Go to Category...
3	Real estate	Go to Category...
4	Travel	Go to Category...

[Mars Planet Webshop start page](#)

0025_sess_cart_1



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9CFB401

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

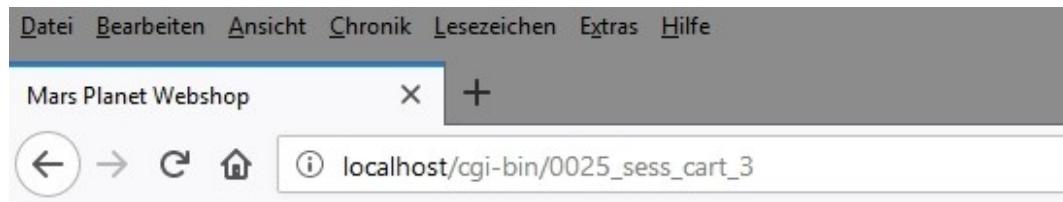
Nr.	Item	Price (\$)	Quantity
1	Oxygen (1 liter)	0.01	0100
2	Water (1 liter)	0.50	0002
3	Potato (1 Kg)	2.50	0001
4	Mars beer (0,33 liter)	3.99	0003
5	Mars VitalMix (0,33 liter)	9.99	0004

[Add to Cart](#) [Zurücksetzen](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0025_sess_cart_2_select_1



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9CFB407A6271D1C034BA89E6

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	100	1.00
Cat1-Item2	Water (1 liter)	0.50	2	1.00
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	3	11.97
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	4	39.96

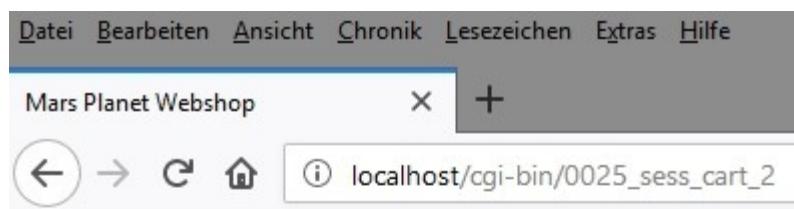
Total Price (\$): 56.43

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0025_sess_cart_3_selected_1



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9CFB4C

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

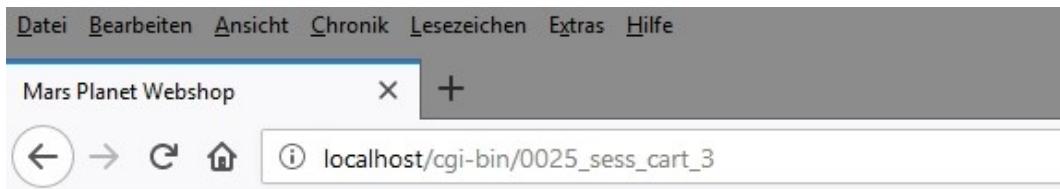
Nr.	Item	Price (\$)	Quantity
1	Universal Mars Robot	600.00	0001
2	Cherry 2000 Female Android	2,000.00	0001
3	Mars Rover Speed	15,000.00	0001
4	Mars Rover Off Road 4x4	25,000.00	0000
5	Mars Flying Saucer	80,000.00	0000

[Add to Cart](#) [Zurücksetzen](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0025_sess_cart_2_select_2



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9C9FB407A6271D1C034BA89E68CFA

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	100	1.00
Cat1-Item2	Water (1 liter)	0.50	2	1.00
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	3	11.97
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	4	39.96
Cat2-Item1	Universal Mars Robot	600.00	1	600.00
Cat2-Item2	Cherry 2000 Female Android	2,000.00	1	2,000.00
Cat2-Item3	Mars Rover Speed	15,000.00	1	15,000.00

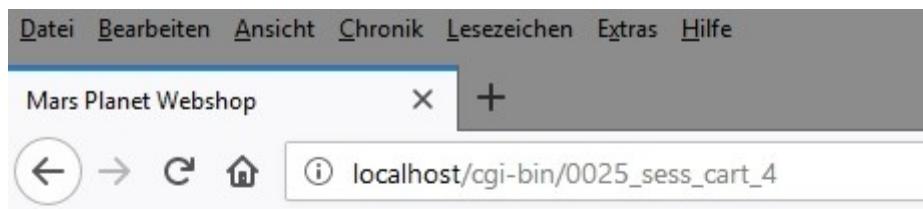
Total Price (\$): 17,656.43

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0025_sess_cart_3_selected_2



HTM2COB-HIDDEN-SESSION-ID: 7E9D289CDF83A30E9CFB407A6271D1

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Checkout

Total Price (\$): 17,656.43

Please transfer the total amount to the given bank account. IBAN: ...

[Mars Planet Webshop start page](#)

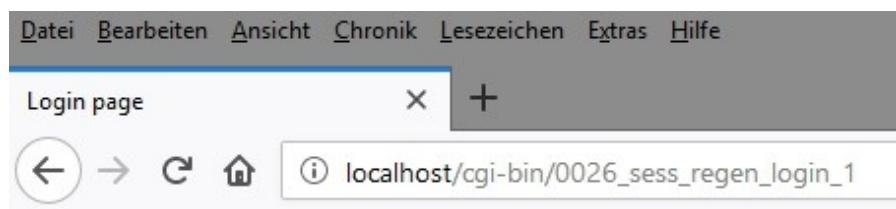
0025_sess_cart_4

3.1.26 0026_sess_regen_login

This example is similar to the example 0023_sess_login. But here the session-id is regenerated after every request. It is also an example for the usage of section HTM2COB-SESS-REGENERATE.

The user name will be saved in the session file after successful login. The existence of this user name will be checked on the following pages. If the user visits the login page later, then a new session-id will be created.





HTM2COB-HIDDEN-SESSION-ID: D8173BAF4159EF5493F105D5B54B7

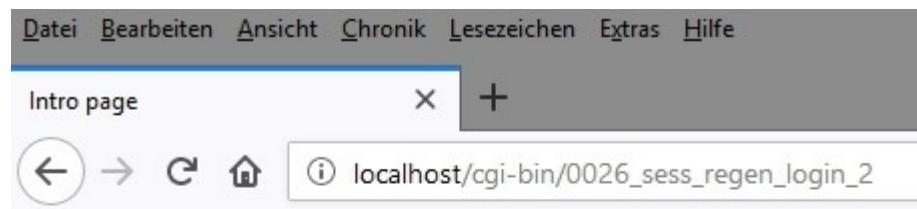
Login page

Name:	admin
Password:	*****

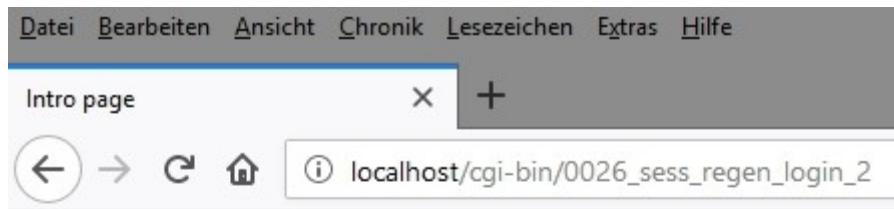
Valid names and passwords: "admin/qwerty" or "user1/pw123".

[Session regenerate login start page](#)

0026_sess_regen_login_1



0026_sess_regen_login_2_unsuccessful



HTM2COB-HIDDEN-SESSION-ID: 10B0639334140B58C645CB7C8FA

after regenerate session

HTM2COB-HIDDEN-SESSION-ID: 252923B491CA0246DEC27DCB7D2

Intro page

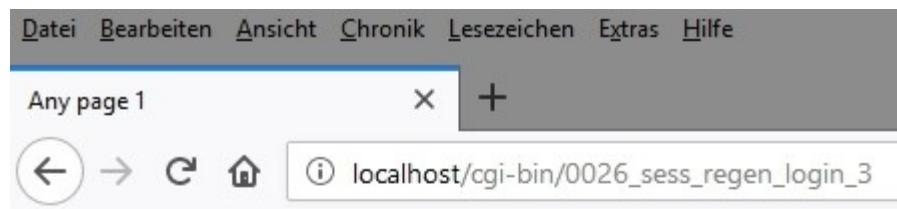
Hello admin

Any page 1

Logoff

[Session regenerate login start page](#)

0026_sess_regen_login_2



after regenerate session

HTM2COB-HIDDEN-SESSION-ID: CEBCEC29916DF38A8F4CACE5C9E4

Any page 1

Hello admin

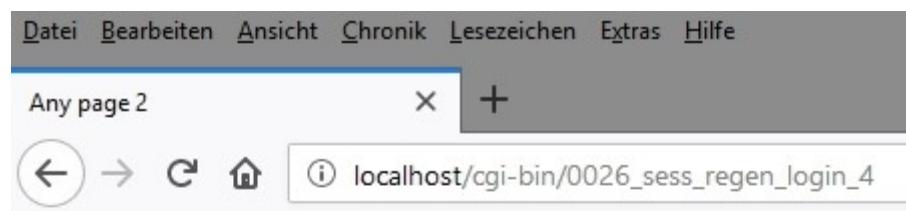
Any page 2

Intro page

Logoff

Session regenerate login start page

0026_sess_regen_login_3



after regenerate session

HTM2COB-HIDDEN-SESSION-ID: 3E352BDB3E18001E681ADF84A95B

Any page 2

Hello admin

Any page 1

Intro page

Logoff

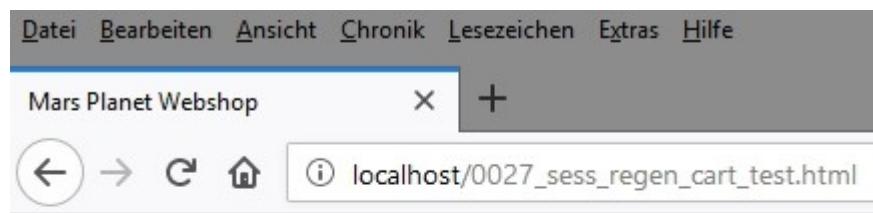
[Session regenerate login start page](#)

[0026_sess_regen_login_4](#)

3.1.27 0027_sess_regen_cart

This example is similar to the example 0025_sess_cart. But here the session-id is regenerated after every request. It is also an example for the usage of section HTM2COB-SESS-REGENERATE.

This is a shopping cart example. You can add or delete items from several categories to the shopping cart. These things are stored in the session file. At the end you can do a checkout and pay.

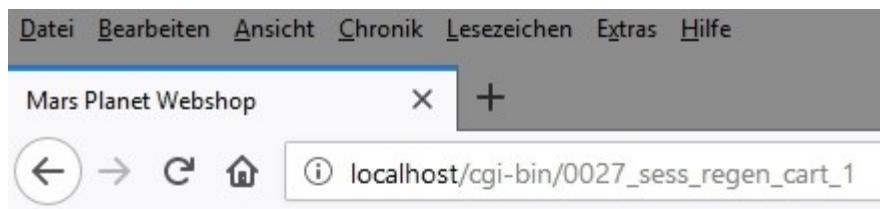


Welcome to the Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

[Start](#)

0027_sess_regen_cart_test



HTM2COB-HIDDEN-SESSION-ID: 482B33EC555F9E463372262F012

after regenerate session

HTM2COB-HIDDEN-SESSION-ID: ABC0C9482F8DA483D2493E03E3E

Mars Planet Webshop

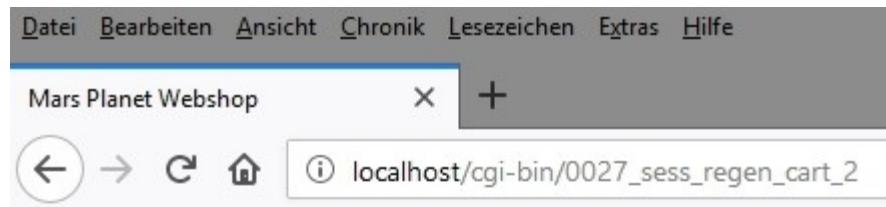
Sorry, there is no return ticket back to Earth...

Choose a category

Nr.	Category	Select Category
1	Vital items	Go to Category...
2	Robots and Mars-Rovers	Go to Category...
3	Real estate	Go to Category...
4	Travel	Go to Category...

[Mars Planet Webshop start page](#)

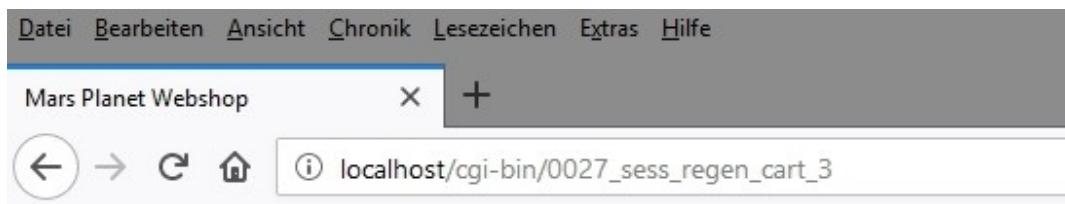
0027_sess_regen_cart_1



Mars Planet Webshop Categories

[Mars Planet Webshop start page](#)

0027_sess_regen_cart_2_select_1



Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	20	0.20
Cat1-Item2	Water (1 liter)	0.50	3	1.50
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	2	7.98
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	4	39.96

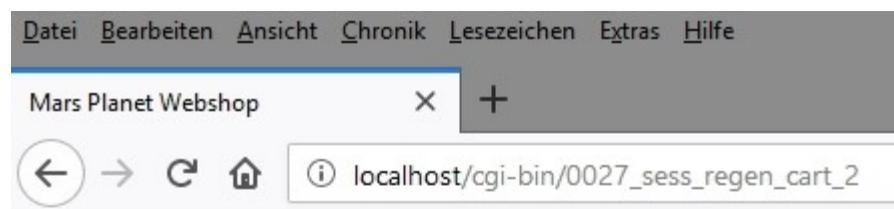
Total Price (\$): 52.14

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0027_sess_regen_cart_3_selected_1



HTM2COB-HIDDEN-SESSION-ID: 69F150AC40831A39252A9538113C

after regenerate session

HTM2COB-HIDDEN-SESSION-ID: 01B84414E7BE8E2F16E4ED26E13C

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

Nr.	Item	Price (\$)	Quantity
1	Universal Mars Robot	600.00	0001
2	Cherry 2000 Female Android	2,000.00	0001
3	Mars Rover Speed	15,000.00	0000
4	Mars Rover Off Road 4x4	25,000.00	0000
5	Mars Flying Saucer	80,000.00	0001

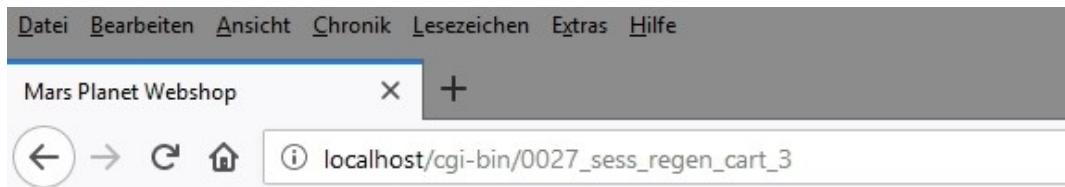
Add to Cart

Zurücksetzen

Mars Planet Webshop Categories

[Mars Planet Webshop start page](#)

0027_sess_regen_cart_2_select_2



HTM2COB-HIDDEN-SESSION-ID: 669823D4AD72EEB9BF739F5F5535E9EC2D87A43E610C

after regenerate session

HTM2COB-HIDDEN-SESSION-ID: 8E5960337D6306ADCC5EF0DF2F9E58791C6B6BD48A01

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	20	0.20
Cat1-Item2	Water (1 liter)	0.50	3	1.50
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	2	7.98
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	4	39.96
Cat2-Item1	Universal Mars Robot	600.00	1	600.00
Cat2-Item2	Cherry 2000 Female Android	2,000.00	1	2,000.00
Cat2-Item5	Mars Flying Saucer	80,000.00	1	80,000.00

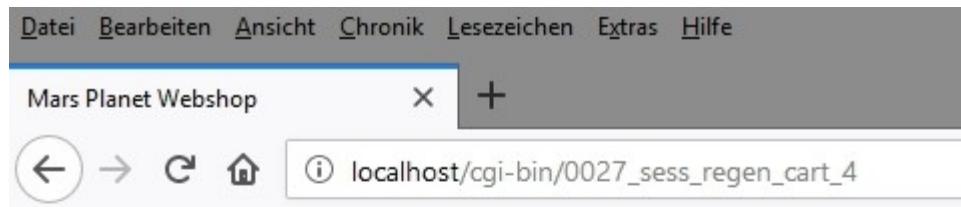
Total Price (\$): 82,652.14

Checkout

Mars Planet Webshop Categories

[Mars Planet Webshop start page](#)

0027_sess_regen_cart_3_selected_2



HTM2COB-HIDDEN-SESSION-ID: 8E5960337D6306ADCC5EF0DF2F9E5879

after regenerate session

HTM2COB-HIDDEN-SESSION-ID: F5029519878E0460737E4A09F80E3D19

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Checkout

Total Price (\$): 82,652.14

Please transfer the total amount to the given bank account. IBAN: ...

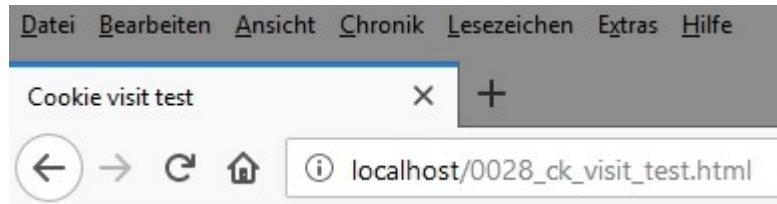
[Mars Planet Webshop start page](#)

0027_sess_regen_cart_4

3.1.28 0028_ck_visit

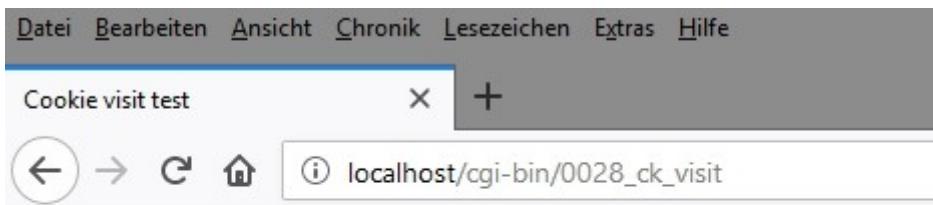
This is a cookie visitor test. The cookie does not exist at first try, and after the user has visited the page it will be set. At the second try the cookie will be successfully read, and the visitor be recognized. For cookies you have to set some parameters in the parameter file 0028_ck_visit_param.cpy, and in the HTML file you have to implement the section HTM2COB-SET-COOKIE.

```
*>-----
*> Max length of the HTM2COB-HTTP-COOKIE field. This has the value from
*> the HTTP_COOKIE environment variable.
[ 78 HTM2COB-HTTP-COOKIE-MAX-LEN      VALUE 10000.
[*> If it set to "YES", then the section HTM2COB-SET-COOKIE will be used. That
[*> means, you have to implement this section in your HTML file.
>>DEFINE USE-HTM2COB-SET-COOKIE-SECTION AS "YES"
```



Cookie visit test

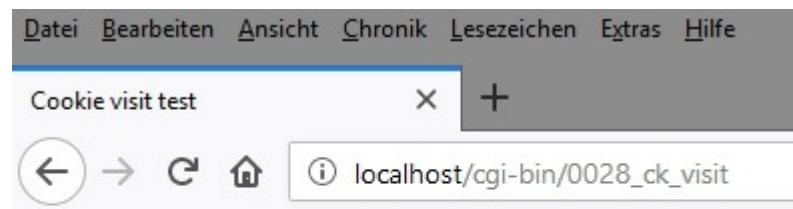
0028_ck_visit_test



Cookie visit test

You have not been here yet, or you can not save cookies. Try again!

0028_ck_visit_try_1



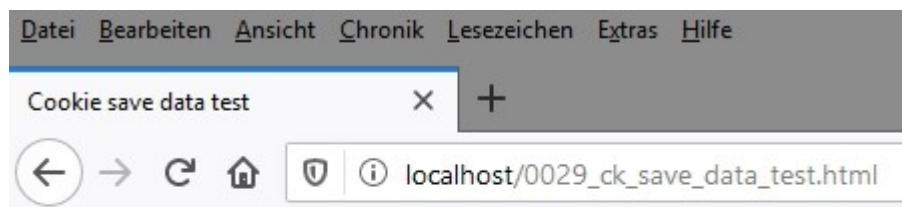
Cookie visit test

You have been here before.

0028_ck_visit_try_2

3.1.29 0029_ck_save_data

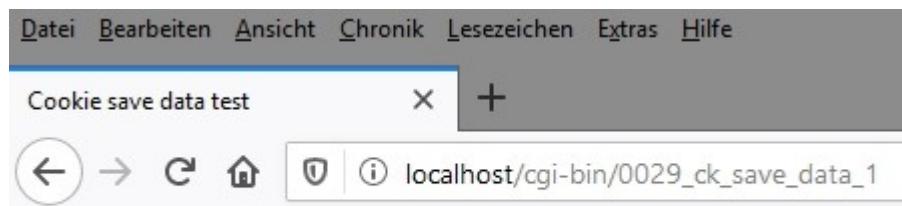
This is a cookie save data test. The cookie does not exist at first try, and the input fields remain empty. After the user has visited the page the cookie data will be set. At the second try the cookie will be successfully read, the visitor be recognized, and the input fields are pre-filled automatically.



Cookie save data test

Send

0029_ck_save_data_test



Cookie save data test

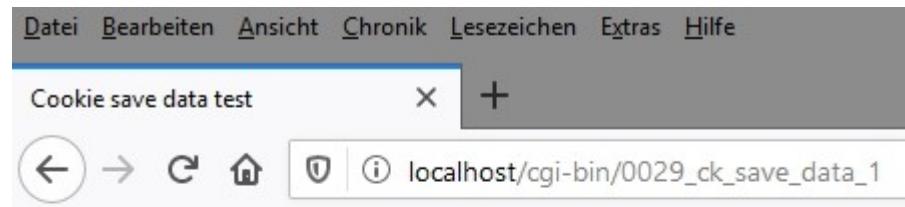
First name:

Last name:

Send

Zurücksetzen

0029_ck_save_data_1_try_1

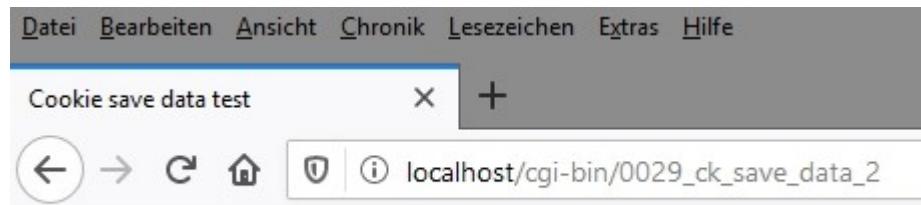


Cookie save data test

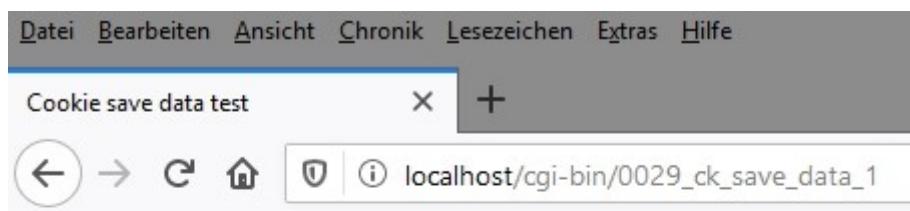
First name: László
Last name: Erdős

Send Zurücksetzen

0029_ck_save_data_1_fill_data



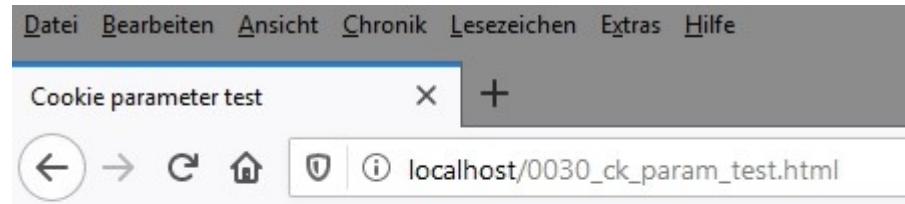
0029_ck_save_data_2



0029_ck_save_data_1_try_2

3.1.30 0030_ck_param

This is a cookie parameter test. Different cookie parameter combinations are set in section HTM2COB-SET-COOKIE. Then the cookie values are read. Before your site is online, you should try these parameter combinations. In addition to name, value and expires, you should always set Domain, Path and HttpOnly.



Cookie parameter test

0030_ck_param_test

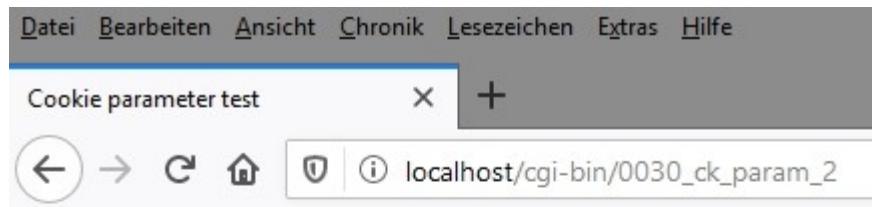


Cookie parameter test

Write cookies...

```
Set-Cookie: NAME01=VALUE01
Set-Cookie: NAME02=VALUE02; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800
Set-Cookie: NAME03=VALUE03; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost
Set-Cookie: NAME04=VALUE04; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/
Set-Cookie: NAME05=VALUE05; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; HttpOnly
Set-Cookie: NAME06=VALUE06; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; HttpOnly; SameSite=Lax
Set-Cookie: NAME07=VALUE07; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; HttpOnly; SameSite=Strict
Set-Cookie: NAME08=VALUE08; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; Secure; HttpOnly
Set-Cookie: NAME09=VALUE09; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; Secure; HttpOnly; SameSite=Lax
Set-Cookie: NAME10=VALUE10; Expires=Thu, 02 Jan 2020 18:29:30 GMT; Max-Age=1800; Domain=localhost; Path=/; Secure; HttpOnly; SameSite=Strict
```

0030_ck_param_1

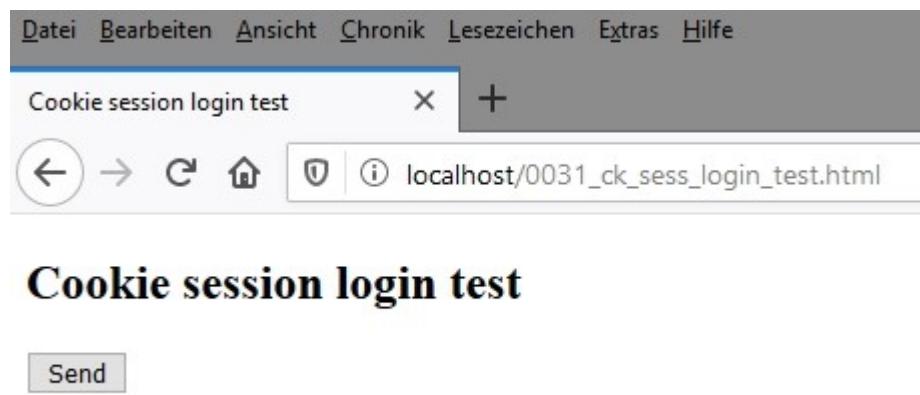


0030_ck_param_2

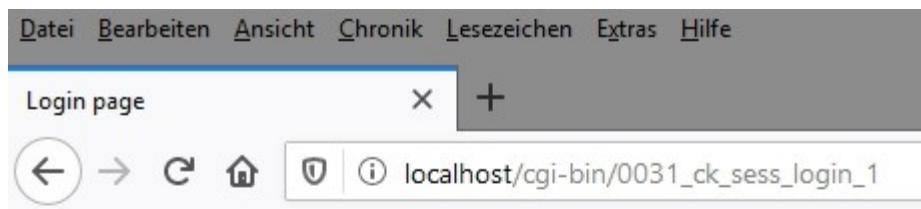
3.1.31 0031_ck_sess_login

This example is the same as example 0023_sess_login, but here the session-id is read from the cookie.

This is a simple example for session tracking with login. The user name will be saved in the session file after successful login. The existence of this user name will be checked on the following pages. If the user visits the login page later, then a new session-id will be created.



0031_ck_sess_login_test



HTM2COB-HIDDEN-SESSION-ID: FB8A6C47D99DEBAB6562978E5F34F1

Login page

The session id is written on this page in the cookie.

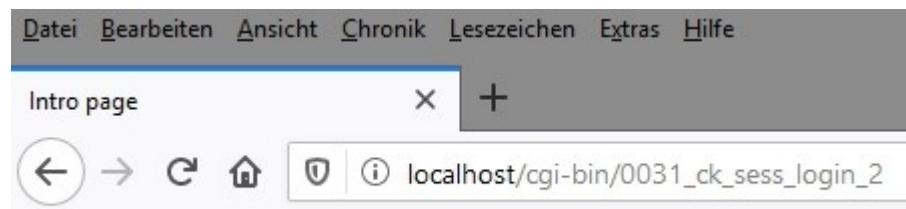
Name:	user1
Password:	*****

Valid names and passwords: "admin/qwerty" or "user1/pw123".

Submit Zurücksetzen

[Cookie session login start page](#)

0031_ck_sess_login_1

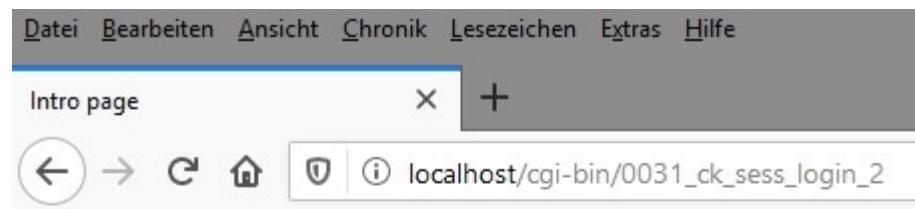


HTM2COB-HIDDEN-SESSION-ID: 500B61F75AA3134B7F14D3ECC4B9

No access

Login page

0031_ck_sess_login_2_unsuccessful



HTM2COB-HIDDEN-SESSION-ID: 2349246A58DB1A3595BC365D5E442

Intro page

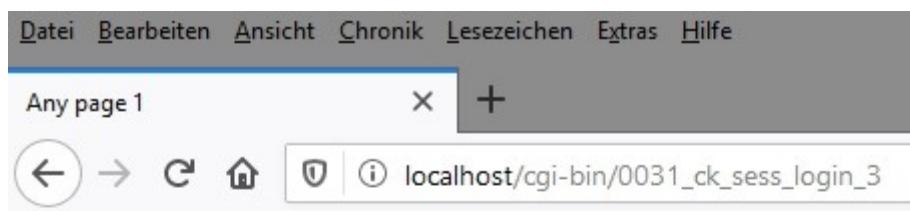
Hello user1

Any page 1

Logoff

[Cookie session login start page](#)

0031_ck_sess_login_2



HTM2COB-HIDDEN-SESSION-ID: 2349246A58DB1A3595BC365D5E44

Any page 1

Hello user1

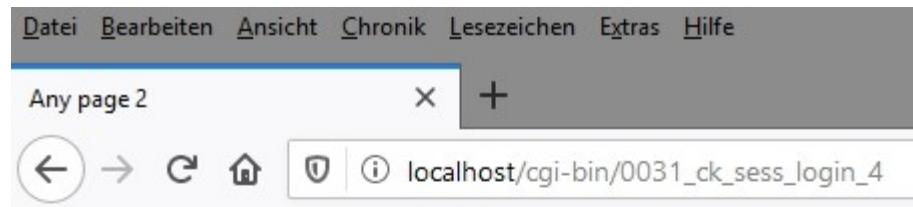
Any page 2

Intro page

Logoff

[Cookie session login start page](#)

0031_ck_sess_login_3



HTM2COB-HIDDEN-SESSION-ID: 2349246A58DB1A3595BC365D5E44:

Any page 2

Hello user1

Any page 1

Intro page

Logoff

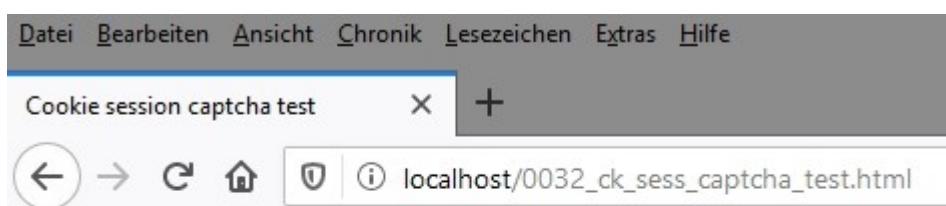
[Cookie session login start page](#)

0031_ck_sess_login_4

3.1.32 0032_ck_sess_captcha

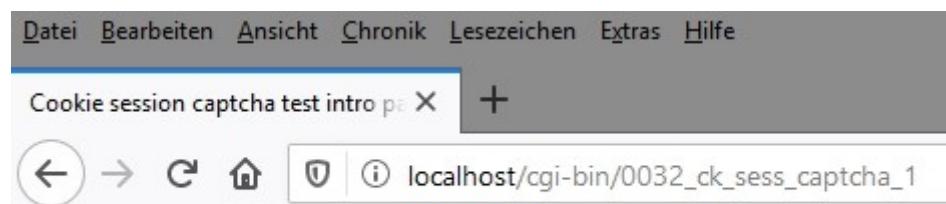
This example is the same as example 0024_sess_captcha, but here the session-id is read from the cookie.

This is a captcha example. The text of the image captcha will be saved in the session. This text will be checked at submit. It is also an example of using the module "imgcaptcha".



Cookie session captcha test

0032_ck_sess_captcha_test



Cookie session captcha test intro page

The session id is written on this page in the cookie.

[Cookie session captcha test start page](#)

0032_ck_sess_captcha_1

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Cookie session captcha test X +

← → ⟳ ⌂ ⚠ ⓘ localhost/cgi-bin/0032_ck_sess_captcha_2

Write your comment

Your name: Hit Girl

Comment: Real programmers never work 9 to 5. If any real programmers are around at 9 am, it's because they were up all night.

 CAPTCHA number:

0032 ck sess captcha 2 wrong

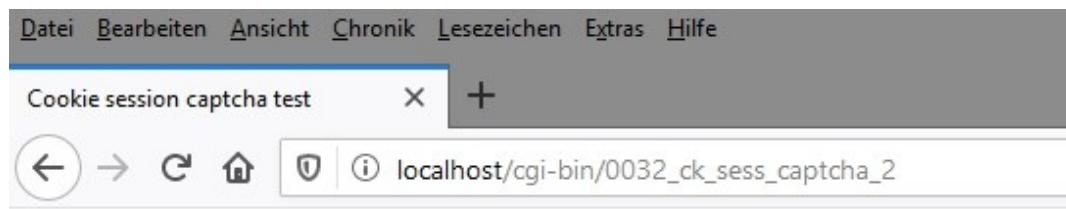
0032 ck sess captcha 2 wrong

A screenshot of a web browser window. The title bar contains the menu items: Datei, Bearbeiten, Ansicht, Chronik, Lesezeichen, Extras, and Hilfe. Below the title bar, the main content area shows a single tab labeled "Session captcha test". To the right of the tab are standard window control buttons: a close button (X) and a new tab button (+). At the bottom of the window, there is a navigation bar with icons for back, forward, refresh, and home, followed by a URL bar containing the address "localhost/cgi-bin/0032_ck_sess_captcha_3".

Captcha verification failed, please try again.

Session captcha test page

0032_ck_sess_captcha_3_unsuccessful



Write your comment

Your name:

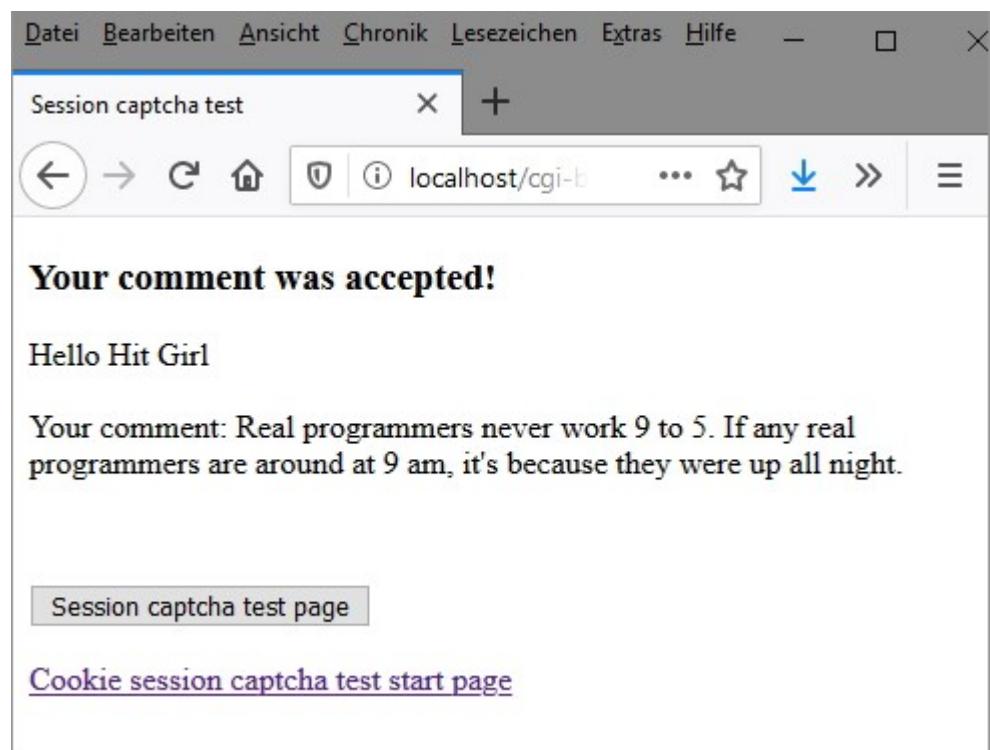
Comment:



CAPTCHA number:

[Cookie session captcha test start page](#)

0032_ck_sess_captcha_2

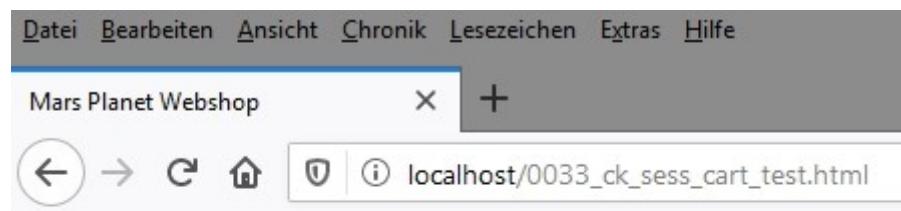


0032_ck_sess_captcha_3

3.1.33 0033_ck_sess_cart

This example is the same as example 0025_sess_cart, but here the session-id is read from the cookie.

This is a shopping cart example. You can add or delete items from several categories to the shopping cart. These things are stored in the session file. At the end you can do a checkout and pay.

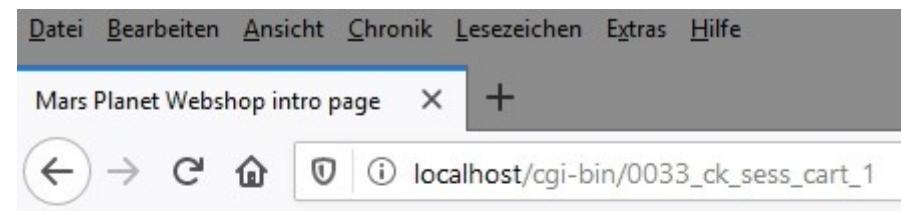


Welcome to the Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

[Start](#)

[0033_ck_sess_cart_test](#)



Mars Planet Webshop intro page

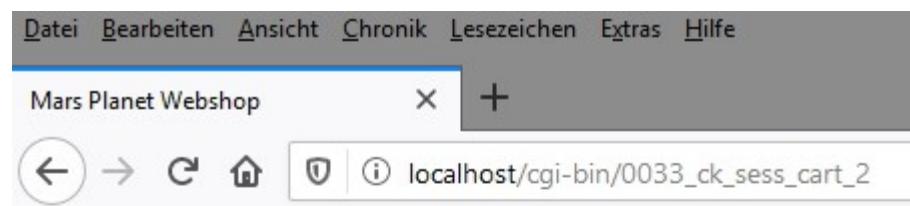
Sorry, there is no return ticket back to Earth...

The session id is written on this page in the cookie.

[Start](#)

[Mars Planet Webshop start page](#)

[0033_ck_sess_cart_1](#)



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE556FF

Mars Planet Webshop

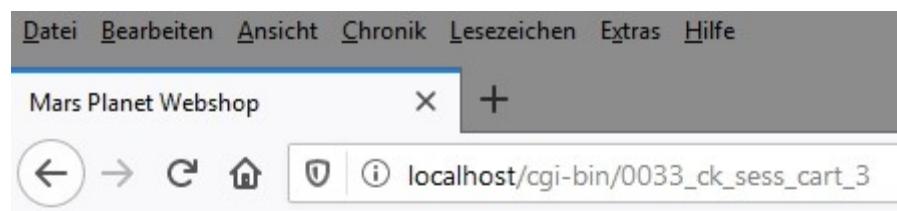
Sorry, there is no return ticket back to Earth...

Choose a category

Nr.	Category	Select Category
1	Vital items	Go to Category...
2	Robots and Mars-Rovers	Go to Category...
3	Real estate	Go to Category...
4	Travel	Go to Category...

[Mars Planet Webshop start page](#)

0033_ck_sess_cart_2



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE5561

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

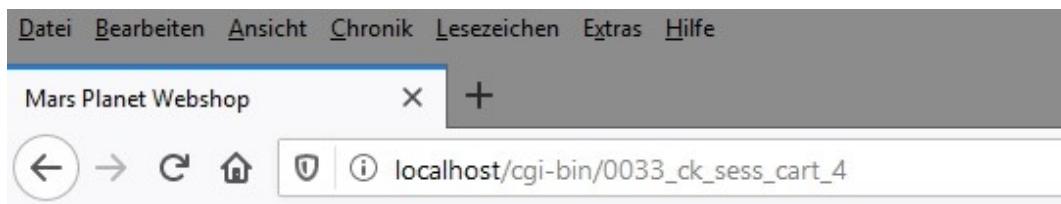
Nr.	Item	Price (\$)	Quantity
1	Oxygen (1 liter)	0.01	0010
2	Water (1 liter)	0.50	0001
3	Potato (1 Kg)	2.50	0001
4	Mars beer (0,33 liter)	3.99	0011
5	Mars VitalMix (0,33 liter)	9.99	0002

[Add to Cart](#) [Zurücksetzen](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0033_ck_sess_cart_3_select_1



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE556FE9C06F5531F1

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	10	0.10
Cat1-Item2	Water (1 liter)	0.50	1	0.50
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	11	43.89
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	2	19.98

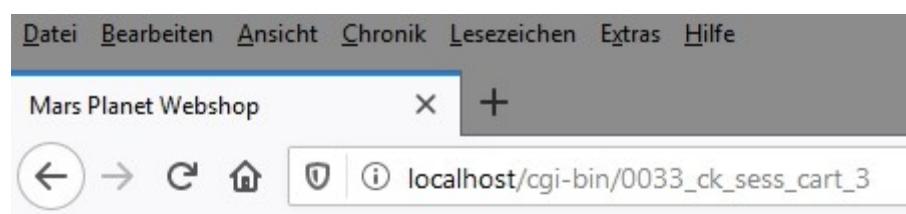
Total Price (\$): 66.97

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0033_ck_sess_cart_4_selected_1



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE556F

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

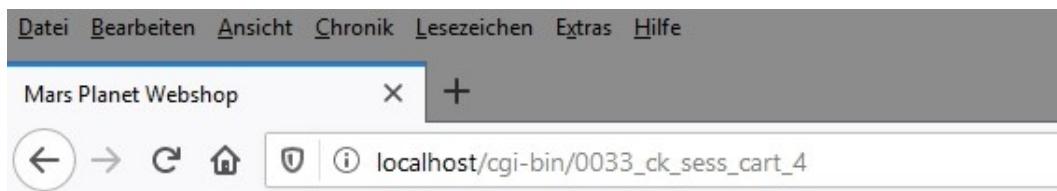
Nr.	Item	Price (\$)	Quantity
1	Universal Mars Robot	600.00	0005
2	Cherry 2000 Female Android	2,000.00	0002
3	Mars Rover Speed	15,000.00	0001
4	Mars Rover Off Road 4x4	25,000.00	0002
5	Mars Flying Saucer	80,000.00	0001

[Add to Cart](#) [Zurücksetzen](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0033_ck_sess_cart_3_select_2



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE556FE9C06F5531F1636

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat1-Item1	Oxygen (1 liter)	0.01	10	0.10
Cat1-Item2	Water (1 liter)	0.50	1	0.50
Cat1-Item3	Potato (1 Kg)	2.50	1	2.50
Cat1-Item4	Mars beer (0,33 liter)	3.99	11	43.89
Cat1-Item5	Mars VitalMix (0,33 liter)	9.99	2	19.98
Cat2-Item1	Universal Mars Robot	600.00	5	3,000.00
Cat2-Item2	Cherry 2000 Female Android	2,000.00	2	4,000.00
Cat2-Item3	Mars Rover Speed	15,000.00	1	15,000.00
Cat2-Item4	Mars Rover Off Road 4x4	25,000.00	2	50,000.00
Cat2-Item5	Mars Flying Saucer	80,000.00	1	80,000.00

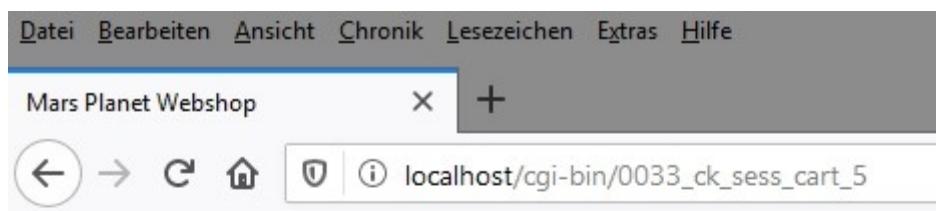
Total Price (\$): 152,066.97

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0033_ck_sess_cart_4_selected_2



HTM2COB-HIDDEN-SESSION-ID: DFE320ACEBB21EDEBAB266BE556FE91

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Checkout

Total Price (\$): 152,066.97

Please transfer the total amount to the given bank account. IBAN: ...

[Mars Planet Webshop start page](#)

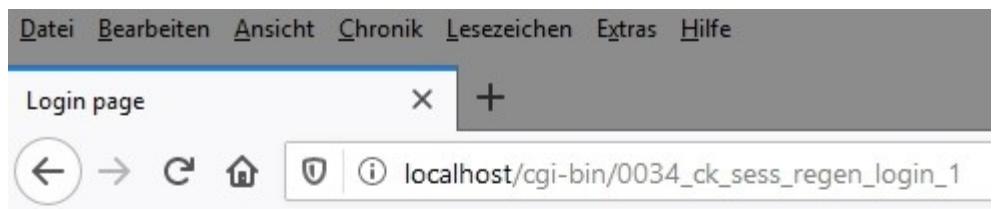
0033_ck_sess_cart_5

3.1.34 0034_ck_sess_regen_login

This example is similar to the example 0031_ck_sess_login (and 0026_sess_regen_login). But here the session-id is regenerated after every request. It is also an example for the usage of section HTM2COB-SESS-REGENERATE.

The user name will be saved in the session file after successful login. The existence of this user name will be checked on the following pages. If the user visits the login page later, then a new session-id will be created.





Login page

The session id is written on this page in the cookie.

Name:

Password:

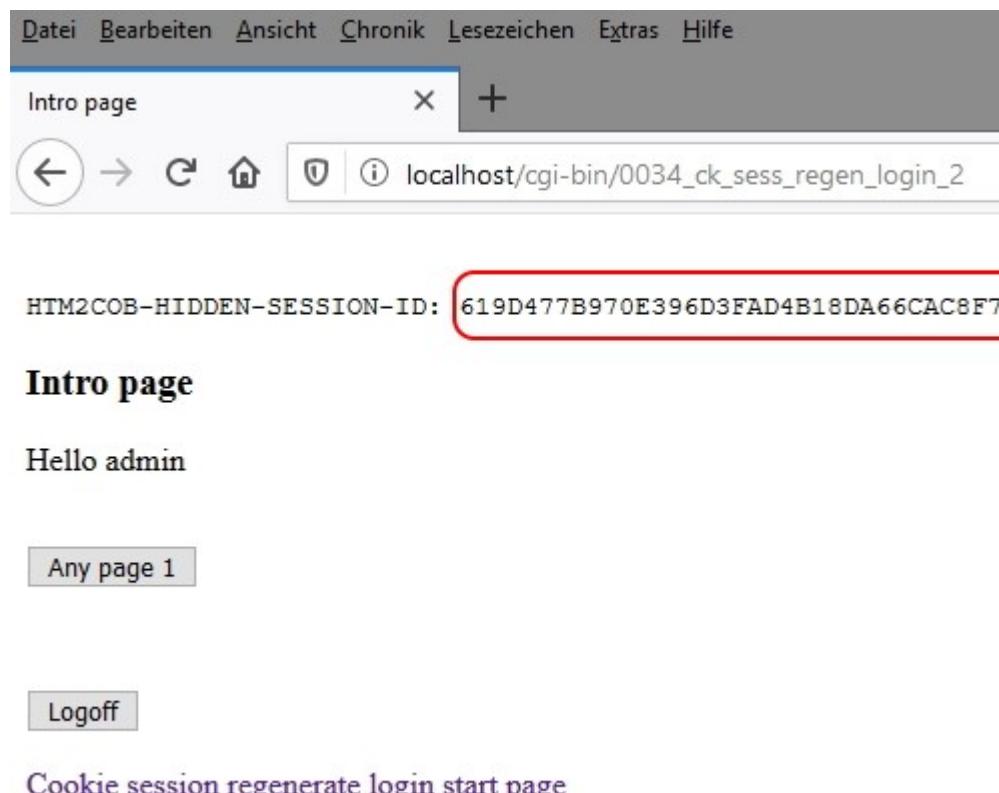
Valid names and passwords: "admin/qwerty" or "user1/pw123".

[Cookie session regenerate login start page](#)

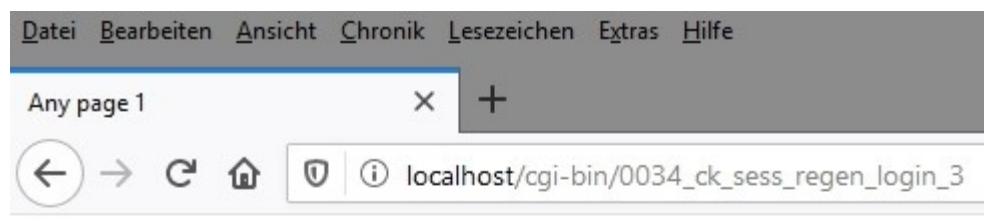
0034_ck_sess_regen_login_1



0034_ck_sess_regen_login_2_unsuccessful



0034_ck_sess_regen_login_2



Any page 1

Hello admin

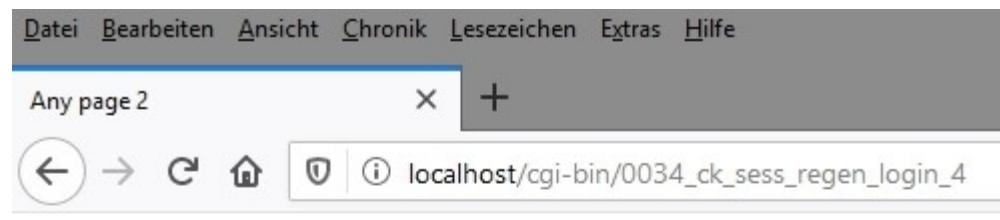
Any page 2

Intro page

Logoff

[Cookie session regenerate login start page](#)

0034_ck_sess_regen_login_3



Any page 2

Hello admin

Any page 1

Intro page

Logoff

[Cookie session regenerate login start page](#)

0034_ck_sess_regen_login_4

3.1.35 0035_ck_sess_regen_cart

This example is similar to the example 0033_ck_sess_cart (and 0027_sess_regen_cart). But here the session-id is regenerated after every request. It is also an example for the usage of section HTM2COB-SESS-REGENERATE.

This is a shopping cart example. You can add or delete items from several categories to the shopping cart. These things are stored in the session file. At the end you can do a checkout and pay.

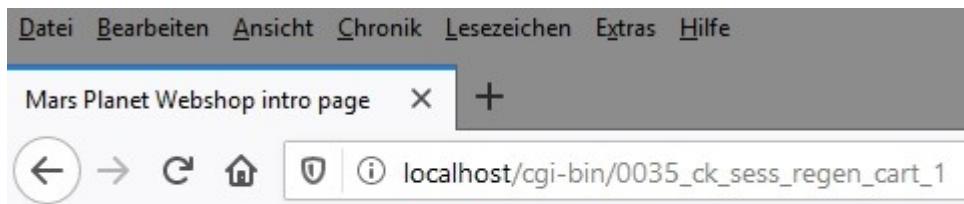


Welcome to the Mars Planet Webshop (Cookie session regenerate test)

Sorry, there is no return ticket back to Earth...

[Start](#)

[0035_ck_sess_regen_cart_test](#)



Mars Planet Webshop intro page

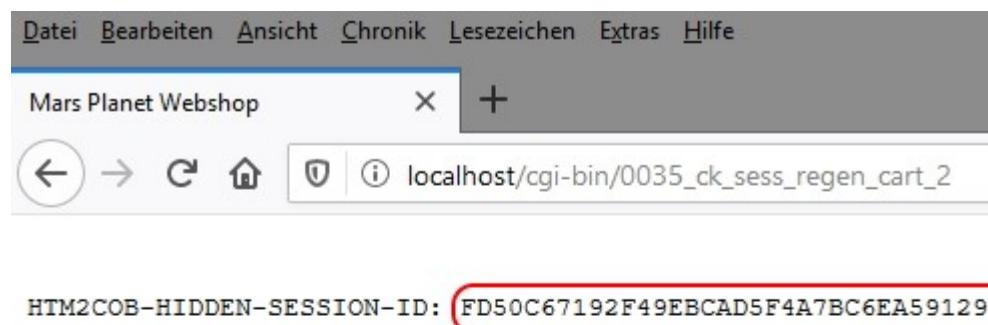
Sorry, there is no return ticket back to Earth...

The session id is written on this page in the cookie.

[Start](#)

[Mars Planet Webshop start page](#)

[0035_ck_sess_regen_cart_1](#)



Mars Planet Webshop

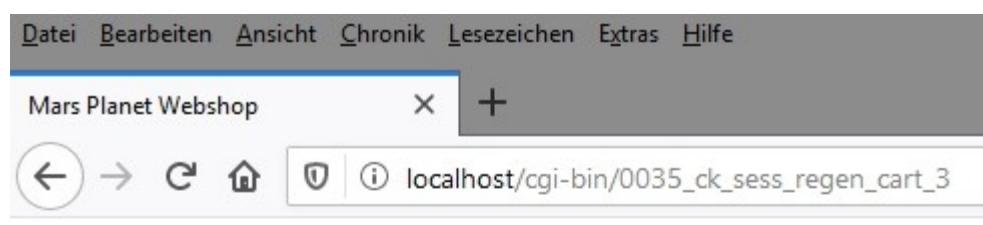
Sorry, there is no return ticket back to Earth...

Choose a category

Nr.	Category	Select Category
1	Vital items	Go to Category...
2	Robots and Mars-Rovers	Go to Category...
3	Real estate	Go to Category...
4	Travel	Go to Category...

[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_2



HTM2COB-HIDDEN-SESSION-ID: **6C10A8134AB651FD7B86D00CC2BE056FE**

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

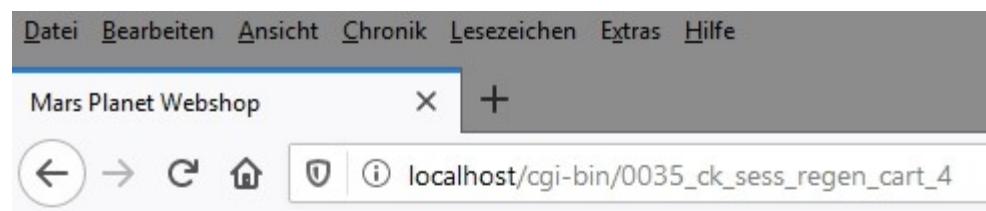
Enter quantities for items

Nr.	Item	Price (\$)	Quantity
1	Mars Igloo	50,000.00	0001
2	Mars Farm	200,000.00	0001
3	Mars Flat in the City	600,000.00	0000
4	Mars Villa	2,500,000.00	0000

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_3_select_1



HTM2COB-HIDDEN-SESSION-ID: **OBFE19E24F0578DBC933A4B8C479BAC85**

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Cart Contents

Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat3-Item1	Mars Iglu	50,000.00	1	50,000.00
Cat3-Item2	Mars Farm	200,000.00	1	200,000.00

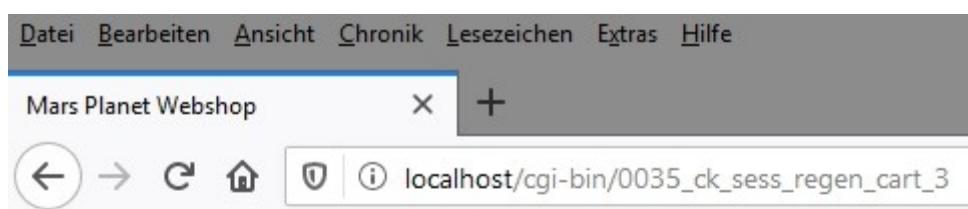
Total Price (\$): 250,000.00

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_4_selected_1



HTM2COB-HIDDEN-SESSION-ID: 7610DA5CE75AEB3DFD1400091A7D6CE9

Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Enter quantities for items

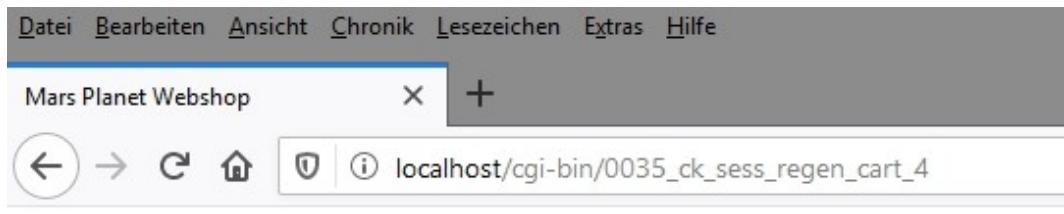
Nr.	Item	Price (\$)	Quantity
1	Travel to the north pole	499.00	0001
2	Travel to the south pole	499.00	0001
3	Mars Round trip	2,000.00	0001

[Add to Cart](#) [Zurücksetzen](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_3_select_2



Item-Nr.	Item	Price (\$)	Quantity	Subtotal Price (\$)
Cat3-Item1	Mars Iglu	50,000.00	1	50,000.00
Cat3-Item2	Mars Farm	200,000.00	1	200,000.00
Cat4-Item1	Travel to the north pole	499.00	1	499.00
Cat4-Item2	Travel to the south pole	499.00	1	499.00
Cat4-Item3	Mars Round trip	2,000.00	1	2,000.00

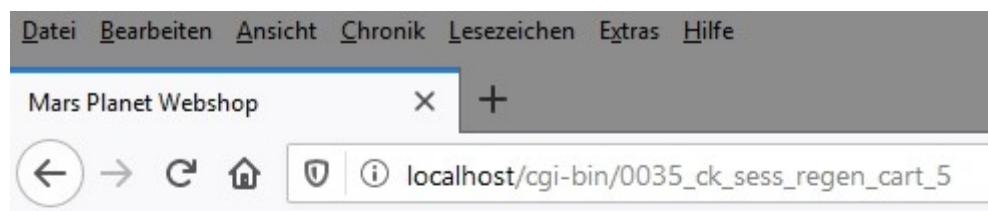
Total Price (\$): 252,998.00

[Checkout](#)

[Mars Planet Webshop Categories](#)

[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_4_selected_2



Mars Planet Webshop

Sorry, there is no return ticket back to Earth...

Checkout

Total Price (\$): 252,998.00

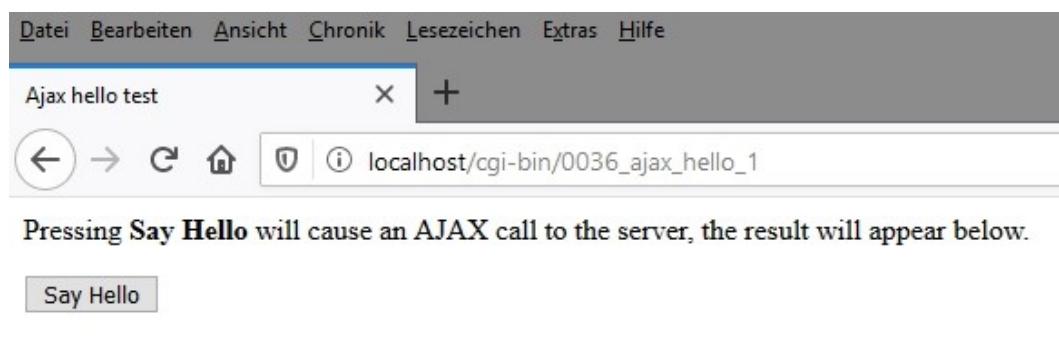
Please transfer the total amount to the given bank account. IBAN: ...

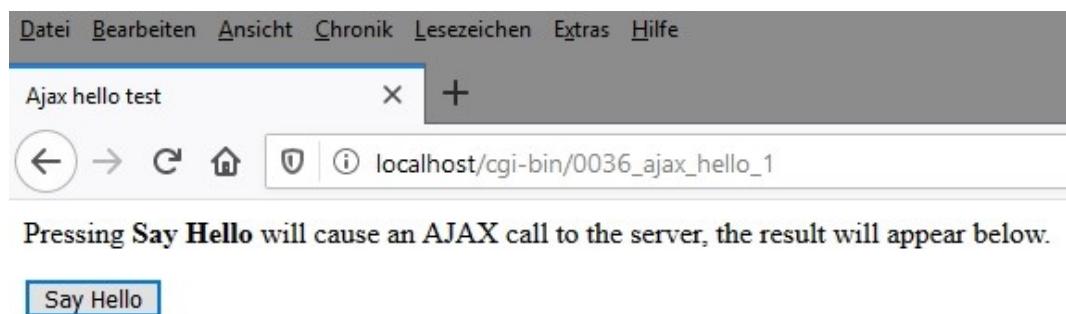
[Mars Planet Webshop start page](#)

0035_ck_sess_regen_cart_5

3.1.36 0036_ajax_hello

This is a simple AJAX call test, it uses the xmlhttpPost() function. After call, the <div id="result"></div> will be replaced with the result.
 The input file 0036_ajax_hello_2_cob.html must be converted with the parameter "-m". That means that HTM2COB converts only the lines between <BODY> and </BODY>. (See the Makefile.)





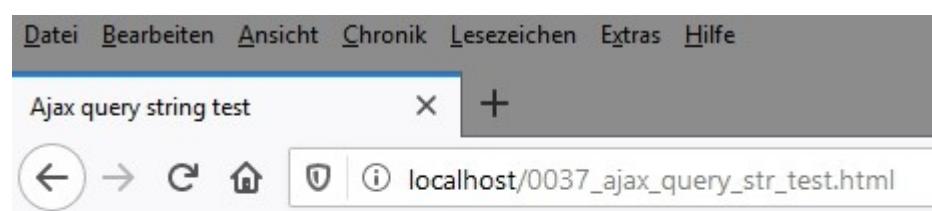
Hello Ajax GnuCOBOL world!

0036_ajax_hello_1_after_click

3.1.37 0037_ajax_query_str

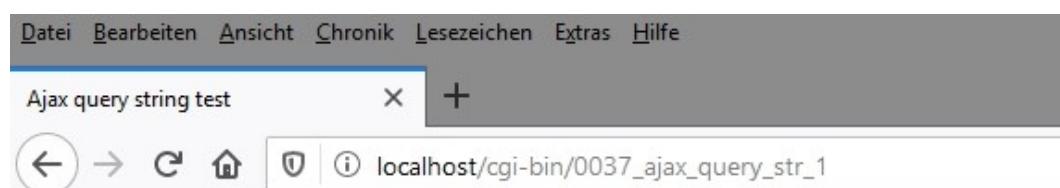
This is a simple AJAX call test, it uses the xmlhttpPost() function. The two form parameters will be sent in a query string to the server. After call, the <div id="result"></div> will be replaced with the result.

The input file 0037_ajax_query_str_2_cob.html must be converted with the parameter “-m”. That means that HTM2COB converts only the lines between <BODY> and </BODY>. (See the Makefile.)



Ajax query string test

0037_ajax_query_str_test



First name:

Last name:

0037_ajax_query_str_1

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Ajax query string test X +

← → ⟳ ⌂ 🛡 ⓘ localhost/cgi-bin/0037_ajax_query_str_1

Pressing **Say Hello** will cause an AJAX call to the server, the result will appear below.

First name:

Last name:

Say Hello

Hello László Erdős!

0037_ajax_query_str_1_after_click

3.1.38 0038_jq_form_validate

It is a simple jQuery test. There are several thousand jQuery plugins, and you can use them with the HTM2COB tool. This example is for an HTML form validation plugin. But don't forget that you should also validate the data on the server side.



Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Form validation test X +

← → ⌂ ⌃ ⌄ localhost/cgi-bin/0038_jq_form_validate_1

Form validation test

first name: This field is required.

surname: This field is required.

gender: male
 female

skills: COBOL
 C/C++
 java
 ASM

your city:

comment:

E-mail: This field is required.

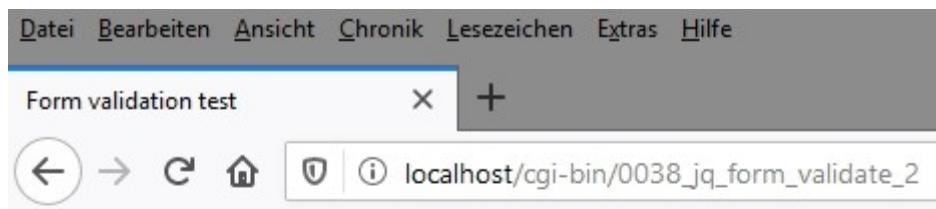
URL:

hidden data:

Send data Reset form

0038_jq_form_validate_1_empty

0038_jq_form_validate_1



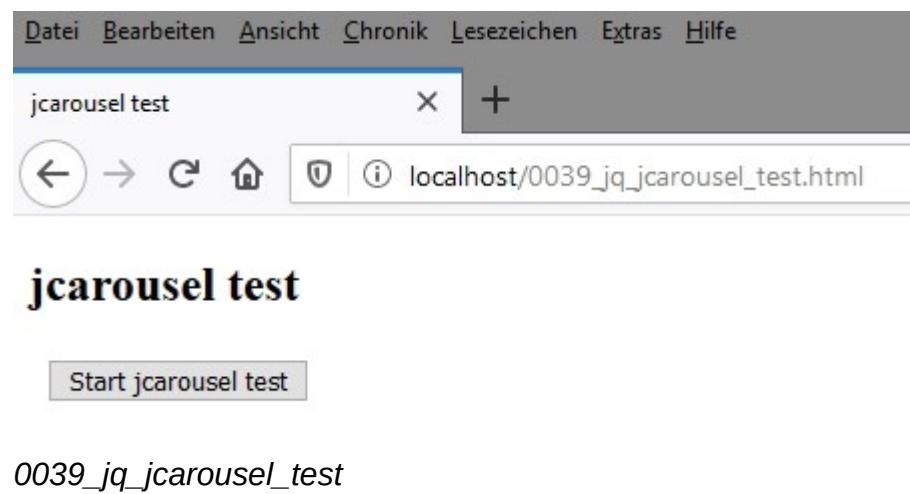
Form validation test

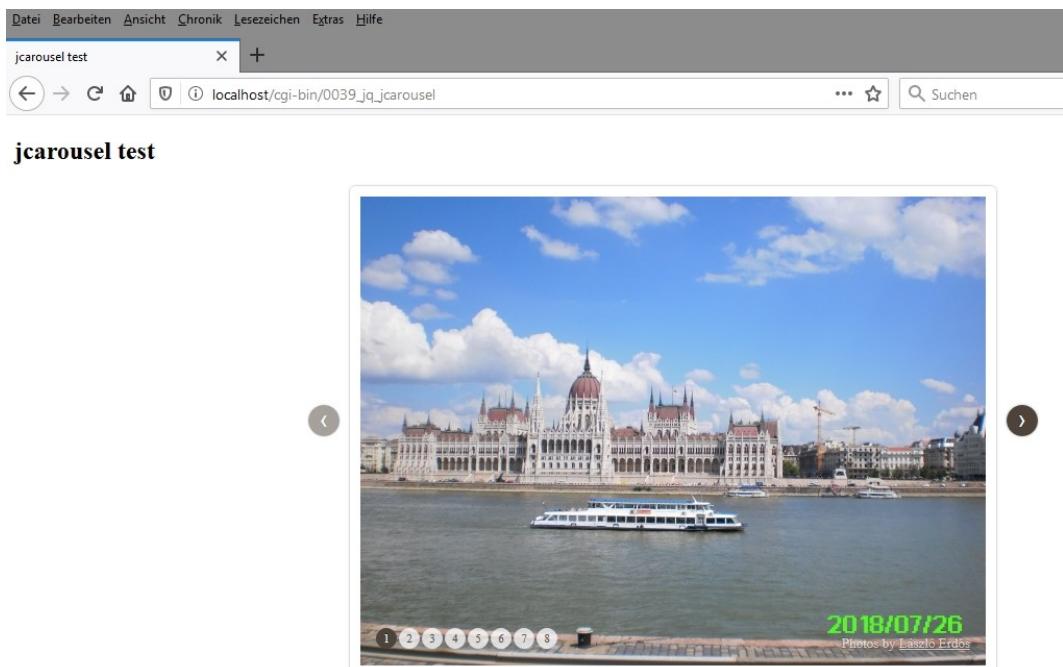
First name : László
Surname : Erdős
Gender : m
Skill-1 : cobol
Skill-2 : cpp
Skill-3 : java
Skill-4 : asm
Your city : Budapest
Comment : I am looking for a new job.
E-mail : lex-x@freenet.de
URL :
Hidden data: 1234567890

0038_jq_form_validate_2

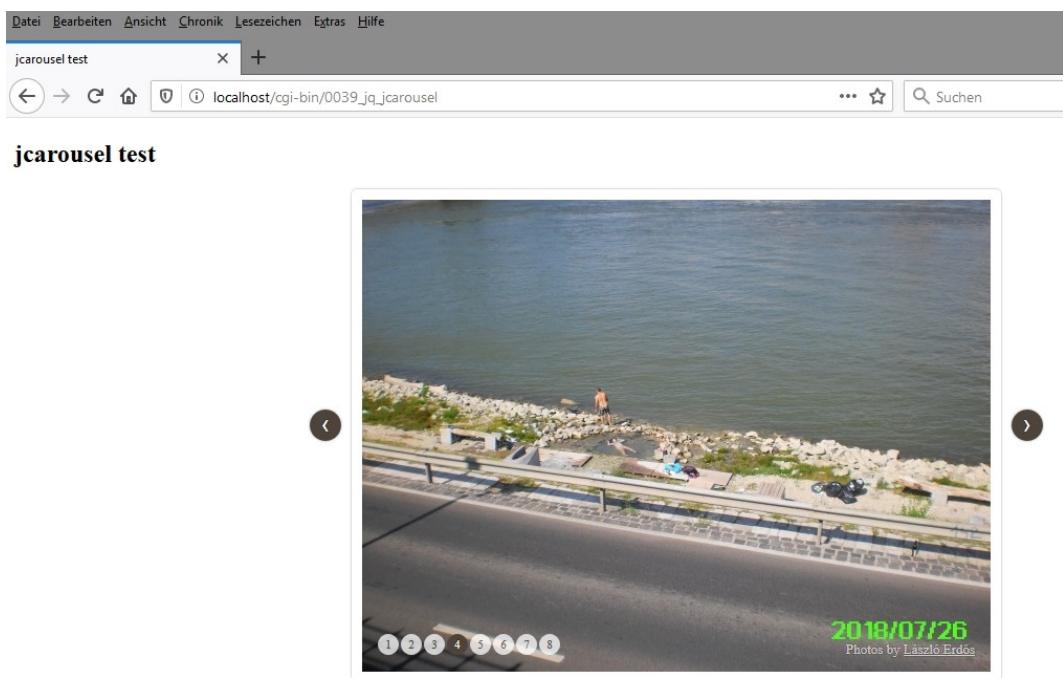
3.1.39 0039_jq_jcarousel

It is a simple jQuery test. There are several thousand jQuery plugins, and you can use them with the HTM2COB tool. This example is for a jCarousel plugin, with 8 images.





0039_jq_jcarousel_image_1



0039_jq_jcarousel_image_4

3.2 Projects

3.3 p001_wortfee

3.3.1 What does it do?

Wort-Fee (in English Word-Fairy) helps you learn the most commonly used words in English, German and Hungarian. This service is free of charge and you can use it as a "guest" without registration with "Guest Login".

After selecting the language and level (number of words), you will be asked a question in the first selected language and six possible answers in the second selected language. One of the six answers is correct. The Word-Fairy shows you immediately whether your answer was correct or not. The correct answers are counted and you must answer each word correctly at least five times to complete a level.

The first language you choose should be the one you are learning. The words are queried in this language. The second language chosen must be a known language, the possible answers are given in this language. For some words there are pictures that will help you answer.

If there is no response for more than 15 minutes, you have to log in again. In this case you lose the previous result. When you have successfully completed a level, you will be added to the guest list.

3.3.2 Directory structure

p001_wortfee	→ main directory
p001_wortfee/css	→ CSS files
p001_wortfee/DB/PostgreSQL	→ PostgreSQL scripts, and CSV input file
p001_wortfee/htm2cob	→ inputs for HTM2COB, and PostgreSQL modules
p001_wortfee/html	→ pure HTML files
p001_wortfee/img	→ images for the site
p001_wortfee/img_wortfee	→ images for the words
p001_wortfee/js	→ jQuery and javascript
p001_wortfee/screenshots	→ screenshots

Every directory has a separate make file.

3.3.3 How to install

- The provided make files are for cygwin and Debian. For other environments you have to do some small changes in the make files. For example you have to customize the target directories for your web server.
- You have to create a PostgreSQL database (with UTF8 encoding), and the SQL tables. See the file psql_create_table.sql. After it you have to import the words from the CSV file into the WF_DICTIONARY table.
- Change the PGUSER.cpy file. Update your DB name and DB user for the SQL access.
- Change all *_param.cpy files. Update the values at HTM2COB-DB-HOME, HTM2COB-SESSION-FILE and HTM2COB-SESSION-VAR-FILE.
- Change the domain name in the section HTM2COB-SET-COOKIE in the file p001_wortfee_login_cob.html: MOVE "wortfee.net" TO LNK-DOMAIN-VALUE.

3.3.4 Test

This program was developed and tested using:

- Windows 10 (64 bit) running on a HP laptop
- cygwin (64 bit)
- GnuCOBOL 3.1-dev.0, built on Aug 17 2019
- Open-COBOL-ESQL-1.2 pre-compiler
<https://github.com/opensourcecobol/Open-COBOL-ESQL>
- psql (PostgreSQL) 10.4
- Firefox 71 (64-Bit)

There is a live demo on a Debian server: <https://wortfee.net>

- Debian version 8.11
- GnuCOBOL 3.1-dev.0, built on Sep 21 2019
- Open-COBOL-ESQL-1.2 pre-compiler
<https://github.com/opensourcecobol/Open-COBOL-ESQL>
- psql (PostgreSQL) 9.4.25

3.3.5 Screenshots

The screenshot shows the homepage of Wortfee.net. At the top, there is a navigation bar with links for "Home", "Guest Login", and "Guest List". Below the navigation bar is a large, semi-transparent watermark-like image of a woman with wings, labeled "Wortfee.net" in red. The main content area features a collage of various photographs including landscapes, animals (a pig, a cat), objects (a teacup, a book), and a painting. Below the collage, the word "Wortfee.net" is displayed in a large, serif font. Underneath, a subtitle reads: *This is a free tool to build your foreign language vocabulary.* At the bottom of the page, there is another navigation bar with links for "English", "German", and "Hungarian". The "English" link is highlighted with a blue background and white text. To the right of the English link is the Union Jack flag. A descriptive text box below the language links states: "Wort-Fee (in English Word-Fairy) helps you learn the most commonly used words in English, German and Hungarian. This service is free of charge".

p001_wortfee_index

The screenshot shows the Wortfee.net website. At the top, there is a black navigation bar with two white buttons: "Home" and "Guest List". Below the navigation bar, the Wortfee.net logo is displayed in a large, bold, sans-serif font. A descriptive subtitle follows: "This is a free tool to build your foreign language vocabulary." Below this, a note states: "A higher level also contains all words from the levels below it." The main content area features a light gray rounded rectangle containing a login form. The form has a title "Login as guest" at the top. It includes three input fields: "Nickname" with the value "László", a dropdown menu set to "English - German", and another dropdown menu set to "Level 1 (0280 words)". At the bottom of the form is a blue "Login" button. At the very bottom of the page, within a dark footer bar, are links for "Impressum", "Datenschutz", and "Privacy".

Home Guest List

Wortfee.net

This is a free tool to build your foreign language vocabulary.

A higher level also contains all words from the levels below it.

Login as guest

Nickname

László

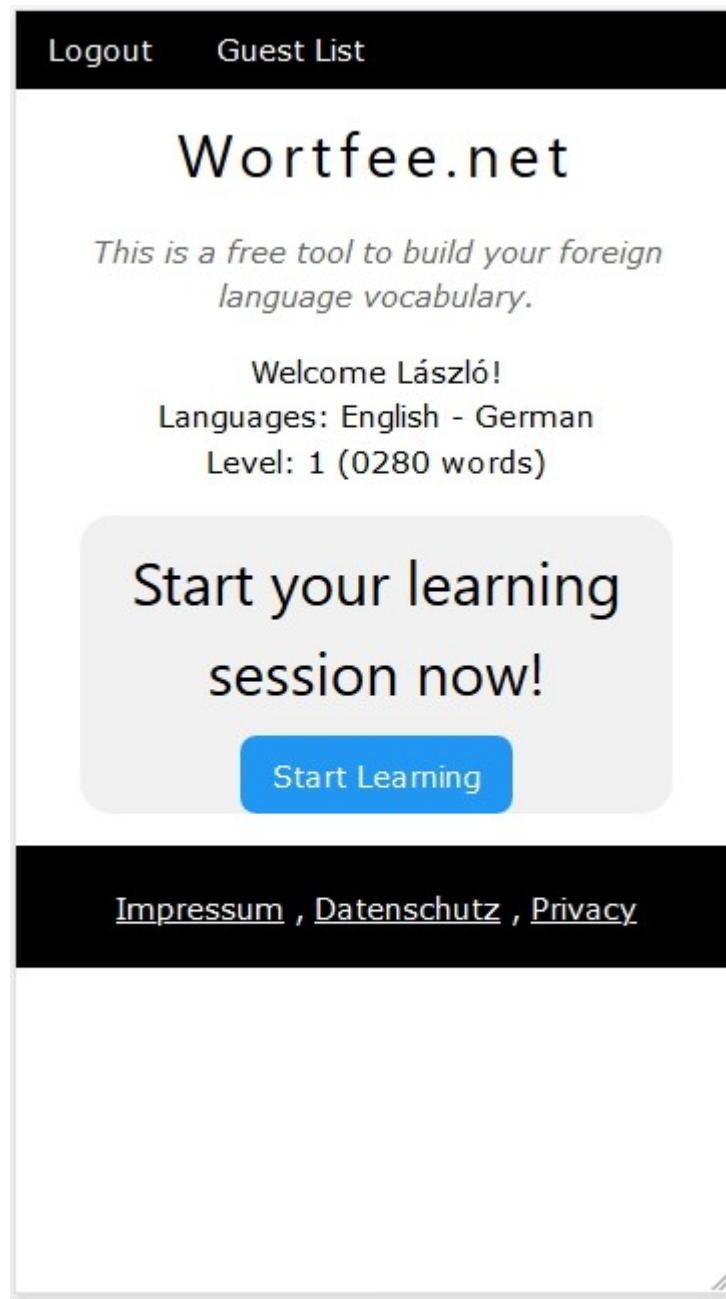
English - German

Level 1 (0280 words)

Login

[Impressum](#) , [Datenschutz](#) , [Privacy](#)

p001_wortfee_login



p001_wortfee_intro

Logout Guest List

Question: beach

[Next Question](#)

Leiter



Taschenrechner

p001_wortfee_learn_1

laufen

Time left: 14:25

Progress: 0 %

Start Time: 2019-12-30 18:03:33

Nickname: László

Languages: English - German

Level: 1 (0280 words)

Questions: 1

OK Answers: 1

Success-0: 0279

Success-1: 0001

Success-2: 0000

Success-3: 0000

Success-4: 0000

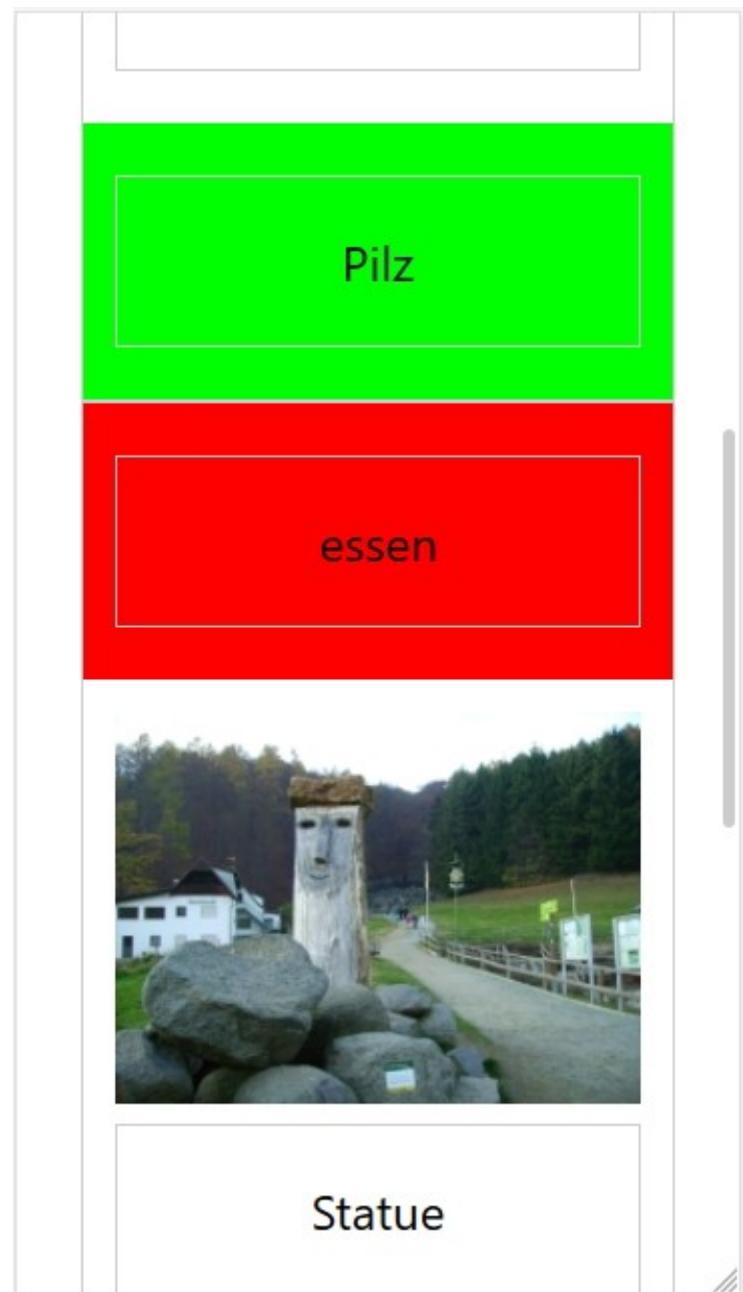
Success-5: 0000

Users online: 1

[Impressum](#) , [Datenschutz](#) , [Privacy](#)

p001_wortfee_learn_2

130



p001_wortfee_learn_3

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Wortfee.net Learning page https://wortfee.net/cgi-bin/p001_wortfee_learn

Logout Guest List

Question: door

Next Question

Feuerwehrwagen

Tür

Mutter

Buch

Kleid

Glas

The screenshot shows a web-based learning interface. At the top, there's a menu bar with German words like 'Datei', 'Bearbeiten', 'Ansicht', etc. Below it is a browser-like header with the URL 'https://wortfee.net/cgi-bin/p001_wortfee_learn'. A black navigation bar at the very top has 'Logout' and 'Guest List' buttons. The main content area has a title 'Question: door' and a 'Next Question' button. There are four options displayed in boxes: 'Feuerwehrwagen' (fire truck) with an image of a red fire truck; 'Tür' (door) with an image of a wooden door; 'Mutter' (mother) with an image of an open book; and 'Glas' (glass) with an image of a glass. To the right of the door image is a box containing the word 'Kleid' (dress). The word 'Tür' is highlighted with a green rectangular overlay.

p001_wortfee_learn_4

Go Back

Wortfee.net

This is a free tool to build your foreign language vocabulary.

On this page you can see a list of users who have successfully completed a level. You can be the next one!

Nickname: László

Languages: English - Hungarian

Level: 1 (280 words)

Questions: 1616

OK Answers: 1600

Start Time: 2019-12-28 09:02:34

End Time: 2019-12-28 11:54:18

[Impressum](#) , [Datenschutz](#) , [Privacy](#)

p001_wortfee_glist

4. Bibliography and references

1. Undine Schrader: Webprogrammierung mit Perl und CGI, m. CD-ROM
ISBN: 978-3-772-36880-6, Franzis (2004)
2. Thomas Theis: Einstieg in PHP 5.4 und MySQL 5.5
ISBN 978-3-836-21877-1, Galileo Press, Bonn 2012
3. Lynn Beighley: jQuery für Dummies
ISBN: 978-3-527-70863-5, WILEY-VCH Verlag GmbH & Co. KgaA,
Weinheim 2012
4. Kyle Honeycutt: Building Bitcoin Websites
ISBN: 978-1-534-94544-9, 2016
5. Mario Heiderich, Christian Matthies, fukami, Johannes Dahse:
Sichere Webanwendungen
ISBN: 978-3-836-21194-9, Galileo Press, Bonn 2009
6. Michael Cross:
Developer's Guide to Web Application Security
ISBN: 978-1-597-49061-0, Syngress Publishing, Inc.
7. Cristian Darie, Emilian Balanescu, Mihai Bucica:
Beginning PHP and PostgreSQL E-Commerce: From Novice to
Professional
ISBN: 978-1590596487, Apress