

Labor

Maik Riedlsperger
3.FSC
2011

19.09.11	Webseitenplanung & Einführung
26.09.11	HTML & Screendesign
03.10.11	Screendesign & CSS
10.10.11	Grundlayout & Dropdown
21.11.11	JQuery
28.11.11	XAMPP
05.12.11	Websitebaker Template
12.12.11	Module
09.01.12	Homepage Übung
06.02.12	Datenbank auslesen und in Datenbank schreiben
20.02.12	Joomla
27.02.12	Log In System
05.03.12	Log In System
16.04.12	Jumploader
23.04.12	Tiny MCE
07.05.12	CMS
14.05.12	CMS

19.09.11

Webseitenplanung:

Bevor man mit dem erstellen einer Seite beginnt, sollte man zuerst einige Planungen vornehmen und sich einige Notizen zur zukünftigen Page machen. Um zu sehen und zu testen, wie die Seite aussehen soll, kann man sich mit einem Grafikprogramm ein sogenanntes Screendesign erstellen. Des weiteren sollte man folgendes beachten:

- Benutzerfreundlich / Barrierefrei / Suchen / Usability
- Design / Layout / Logo/Banner / Hintergrund / Screendesign
- Sprachen
- Inhalt / Text / Kontakt / Bilder / Video
- Sitemap / Startseite
- SEO → Suchmaschinenoptimierung
- Domain / www
- Webstatistiken

Vor dem Schreiben einer Seite, sollte man beispielsweise auf **nic.at** checken, ob die Gewünschte Domain noch vorhanden ist. Nic.at ist die *Österreichische Registrierungsbehörde für Domains* und hat ihren Sitz in der TU Wien. In Salzburg ist ein Büro vorhanden. Für eine .at Domain muss man *jährlich circa 20€* zahlen. Einen *Webserver* für die Daten der Seite (auch die .html / .php / ... Dokumente) gibt es schon ab *5€/Monat*.

Bei dem Design der Seite sollte man auf *CI / CD* achten. CI steht für Corporate Identity, CD für Corporate Design. Man sollte der Webseite *keine zu große Breite* geben, da man sonst auf Bildschirmen mit kleinerer Auflösung nach recht/links scrollen muss, was *ungewohnt und unpraktisch* ist. Vielleicht ändert sich dies in Tablet PC / iPad Zeiten ja noch.

Die meist verwendeten Auflösungen:

- | | |
|----------|-----------|
| – 22,5% | 1024x765 |
| – 15,6% | 1366x768 |
| – 14,55% | 1280x800 |
| – 7,69% | 1280x1024 |
| – 6,92% | 1440x900 |

Man kann sehen, dass die Breite einer Website *1024 Pixel nicht überschreiten* sollte. Geeignete Breiten sind *800 – 1000 Pixel*. Den Rest des Platzes kann man beispielsweise für Werbung verwenden, die dann eben auf kleineren Bildschirmen nicht gesehen werden.

26.09.11

HTML Tags:

<code><h1> </h1></code>	Überschrift
<code> </code>	Big/Dickere Schrift
<code><u> </u></code>	Unterstreichen
<code>
 </br></code>	Absatz
<code></code>	Bild einfügen: src = Pfad zum Bild; alt = Alternativtext
<code><table> </table></code>	Tabelle
<code><tr> </tr></code>	Bei Tabelle Zeilen einfügen
<code><td> </td></code>	Bei Tabelle Zellen einfügen
<code> </code>	Beginn einer Liste
<code> </code>	Listeneintrag

Jeder HTML Tag muss *geöffnet und geschlossen* werden! Bei `<b> </b>` ist beispielsweise `<b>` das Öffnen des Tags und `</b>` das Schließen.

Die Seite, die wir geschrieben haben:

Quellcode:

```
1 <b> Ich heie Maik Riedlsperger. </b> <br>
2 Ich besuche die Computerfachschule der <a href="http://www.htlsaalfelden.at" target="_blank">HTL Saalfelden</a>. <br> <br>
3
4
5  <br> <br>
6
7
8 <table border="1" width="200" height="150">
9
10 <tr>
11 <td align="center">x</td>
12 <td>&nbsp;</td>
13 <td>&nbsp;</td>
14 </tr>
15
16 <tr>
17 <td>&nbsp;</td>
18 <td align="center">x</td>
19 <td>&nbsp;</td>
20 </tr>
21
22 <tr>
23 <td>&nbsp;</td>
24 <td>&nbsp;</td>
25 <td align="center">x</td>
26 </tr>
27 </table>
28
29
30
31 <ul>
32 <li> Eintrag 1 </li>
33 <li> Eintrag 2 </li>
34 <li> Eintrag 3 </li>
35 <ul>
36 <li> Eintrag 3.1 </li>
37 <li> Eintrag 3.2 </li>
38 </ul> <br> <br>
39
```

Fertige Seite im Browser:

Ich heie Maik Riedlsperger.

Ich besuche die Computerfachschule der [HTL Saalfelden](http://www.htlsaalfelden.at).

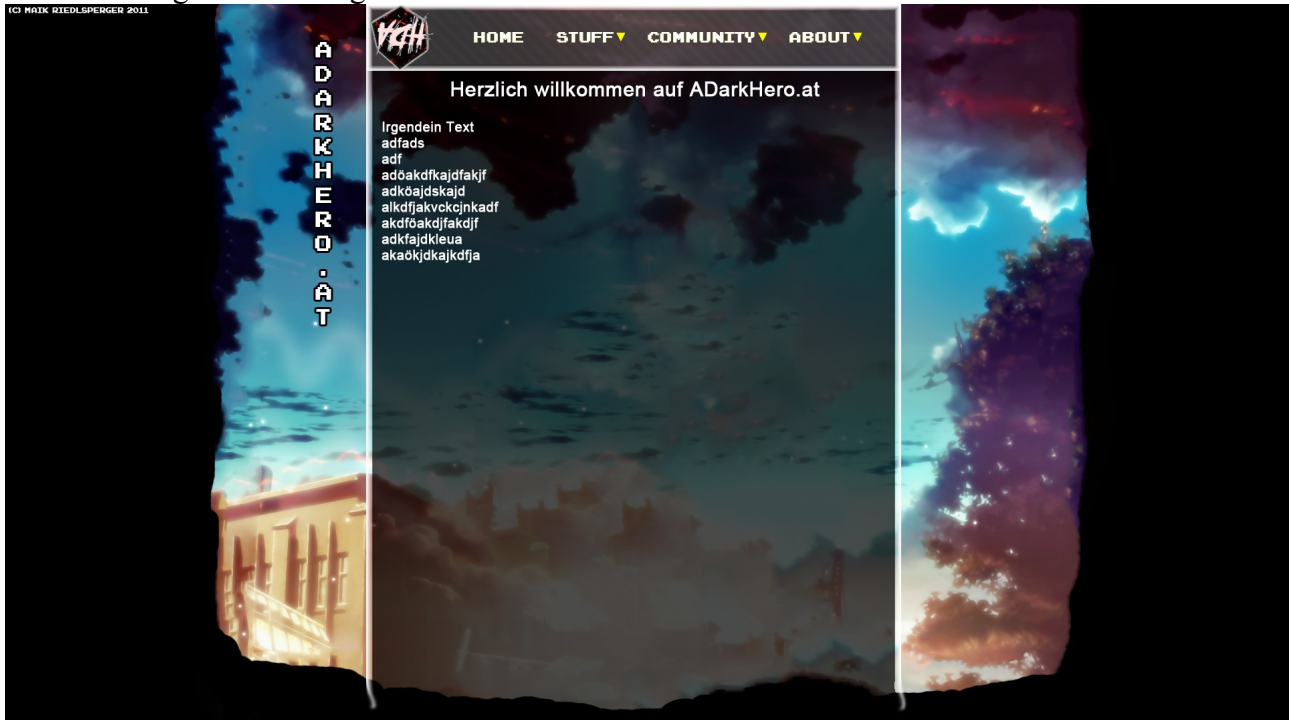


x		
	x	
		x

- Eintrag 1
- Eintrag 2
- Eintrag 3
 - Eintrag 3.1
 - Eintrag 3.2

Screendesign:

Hier das fertige Screendesign der Webseite:



Durch einen Klick auf *Home*, *Stuff*, *Community* oder *About* ändert sich die Farbe des jeweiligen Menüpunktes in Gelb.

Durch einen Klick auf *Stuff*, *Community* oder *About* öffnet sich des weiteren ein Dropdown Menü.

Links sieht man den Schriftzug *ADarkHero.at*, den Namen der Seite.

Ein *Logo* ist ebenfalls schon vorhanden und hat einen Platz neben den Links erhalten.

Für *größere Bildschirme* klingt der Hintergrund der Seite in einem *Schwarz* aus. Das Selbe ist der Fall, wenn längere Texte auf einer Seite sind oder die Seite durch beispielsweise Bilder höher wird. Auf *Full HD Monitoren* (1920 x 1080) ist ebenfalls ein *Copyright Schriftzug* zu erkennen.

Bei Stuff ist dies:

- Bilder
- Videos
- Texte
- Anderes

Bei Community:

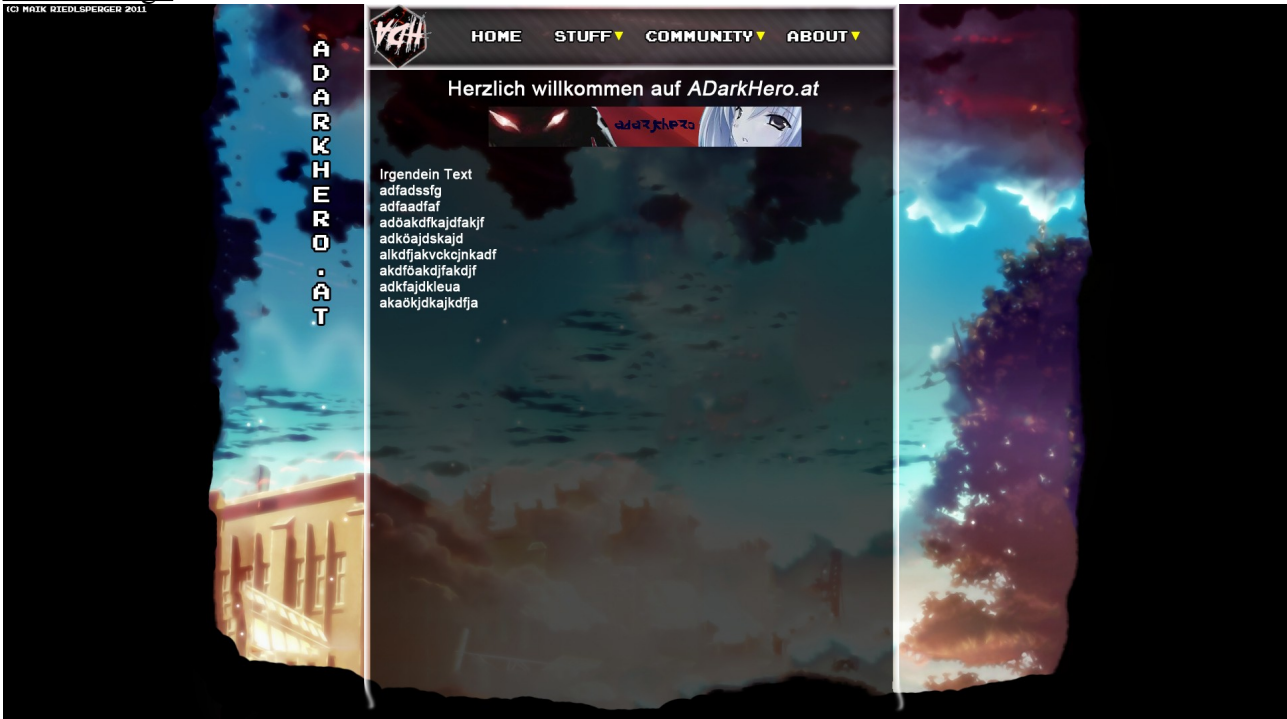
- Forum
- Chat
- Shoutbox
- Facebook
- Twitter

Bei About:

- Impressum
- About Us
- Partner
- Link Us
- F.A.Q.

03.10.11

Screendesign:



Hier das Screendesign am Ende des Tages. Es wurde ein Banner hinzugefügt, der sich bei Neuladen der Seite immer ändern soll. Ansonsten blieb der Großteil des Designs unverändert.

CSS:

Mit einem CSS Template testeten wir erste Befehle.

```
<div id="centering"> </div>
```

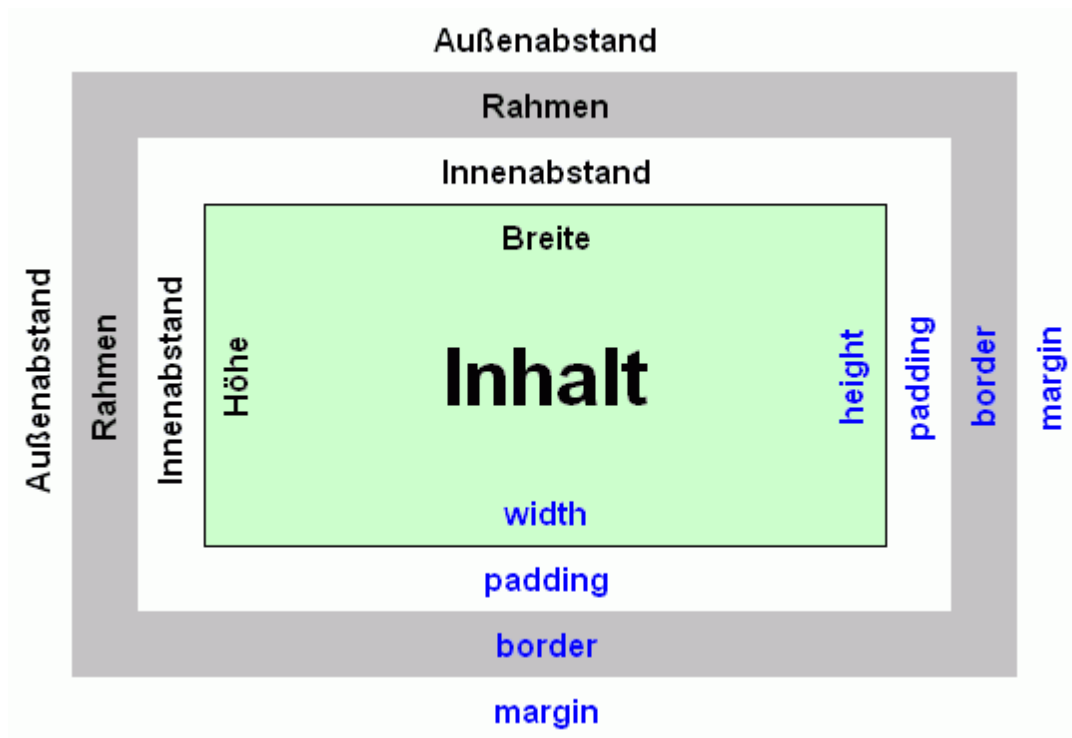
Dieser Befehl erstellt in der .html Datei eine Box.

```
#centering {  
  width:775px;  
  height: 950px;  
  margin:auto;  
  padding-left:25px;  
  padding-top:5px;  
}
```

Mit einer # beginnt man im .css Dokument einen Style. #centering bedeutet also, dass man diesen für die Box centering festlegt.

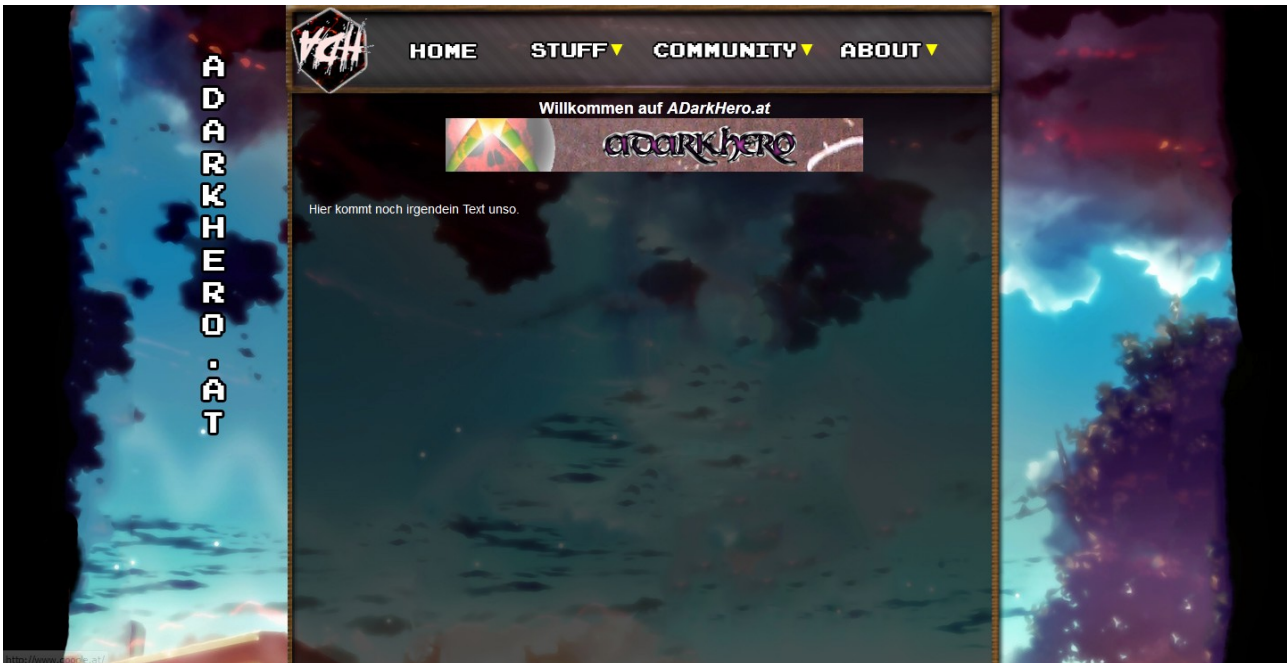
Width:	Gibt die Weite der Box an.
Height:	Gibt die Höhe der Box an.
Margin:	Gibt die Ausrichtung der Box an. Durch auto wird sie zentriert.
Padding:	Gibt den Abstand vom Text in der Box zum Boxrand an.
Border:	Gibt den Rand der Box an.

Verwendet man margin, padding oder border, muss die Gesamtgröße der Box (width/height) um den Wert von margin/padding/border verringert werden. Hat man also 25px padding auf der linken Seite und die Box soll 800px breit sein, muss man bei width 775px eingeben.



Diese Grafik zeigt die Änderung der Boxgröße durch width/padding/border/margin.

10.10.11



Hier ist die aktuelle Webseite. Es wurden bisher 4 Buttons zur Navigation eingefügt, sowie mit Dropdown Menüs versehen. Des weiteren wurde ein Logo eingefügt. Auf der Index Seite wird des weiteren ein Bild angezeigt, dass beim neuladen der Seite wechselt.

```
/* BOX DEFINITIONS */  
  
#centering {  
    width: 800px;  
    margin: auto;  
}  
  
#header {  
    width: 805px;  
    height: 99px;  
    margin-top: 1px;  
    margin-left: -5px;  
}  
  
#main {  
    width: 775px;  
    height: 495px;  
    margin: auto;  
    padding-left: 25px;  
    padding-top: 5px;  
}
```

Zum realisieren der Seite wurde 3 <div> Boxen verwendet. Die centering Box ist die komplette Seite. Die header Box ist für die Navigation, also die Menüpunkte/Buttons, sowie das Logo gedacht. In die Main Box kommt der Text der jeweiligen Seite.

```
<ul class="dropdown">
  <li>
    <ul class="sub_menu">
      <li><a href="#">Bilder</a></li>
      <li><a href="#">Videos</a></li>
      <li><a href="#">Videos</a></li>
      <li><a href="#">Anderes</a></li>
    </ul>
  </li>
</ul>
```

Mit Hilfe von JQuery lässt sich ein Dropdown Menü realisieren. Nach dem Einbinden der CSS Dateien in der HTML Seite, kann mit der Hilfe einer Liste ein Dropdown Menü erstellt werden.

21.11.11

Mit JQuery lassen sich Javascript Aufgaben, wie Bilderwechsel, Dropdowns oder Mouseovers realisieren.

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

<script type="text/javascript" src="../../template/js/jquery.js"></script>
<script type="text/javascript" src="../../template/js/jquery.easing.compatibility.js"></script>
<script type="text/javascript" src="../../template/js/jquery.cycle.all.js"></script>

<script type="text/javascript">
$(document).ready(function() {
    $('.slideshow').cycle({
        fx: 'fade' // choose your transition type, ex: fade, scrollUp, shuffle, etc...
    });
});
</script>
```

Um JQuery verwenden zu können, müssen die relevanten Dateien zuerst heruntergeladen werden und danach in die Webseite eingebunden werden.

Jquery Cycle (für Bilderwechsel) benötigt die Standard jquery.js und die jquery-cycle.js. Mit <script> wird der Befehl aufgerufen.

```
<div class="slideshow" style="text-align: center;">
    
    
</div>
```

Mit einem <div> Container wird der Bilderwechsel in die Seite eingebunden.

```
.slideshow { height: 232px; width: 232px; margin: auto }
.slideshow img { padding: 15px; }
```

In einer neuen .css Datei (z.B. cycle.css) wird dann der Style für die Slideshow eingefügt.

28.11.11

XAMPP ist ein lokaler Webserver für Windows.

Dieser wird benötigt um php Dateien lokal anzuzeigen.

<C:/xampp/htdocs> ist der Pfad zum lokalen Webserver. Kopiert man Seiten in diesen Ordner, kann man sie sich im Browser unter localhost oder 127.0.0.1 darauf zugreifen.

CMS steht für Content Management System. Dieses trennt die Struktur/Design vom eigentlichen Inhalt der Seite. So kann man das Design jederzeit ändern, ohne am eigentlichen Inhalt Änderungen vornehmen zu müssen.

Des weiteren können auch Leute, die mit HTML keine Erfahrung haben mit einem CMS neue Seiten erstellen, da sie einen WYSIWYG Editor zur Verfügung gestellt bekommen.

CMSs sind z.B. Joomla oder Wordpress.

Webseitebaker ist ein einfaches CMS, in dem man Seiten erstellen und verändern kann.

05.12.11

Um die Datenbank des Websitebakers auszulesen gibt es spezielle PHP Befehle.

```
<?php echo WEBSITE_HEADER; ?> → Kopfzeile  
<?php show_menu(1); ?> → Menü/Navigation  
<?php page_content(); ?> → Inhalt (der im CMS erstellt wird)
```

Vor Bilderpfaden oder CSS/JS Pfaden muss <?php echo TEMPLATE_DIR;?> eingefügt werden, wenn die jeweiligen Dateien im Template liegen.

Des weiteren wird eine so genannte info.php benötigt:

```
<?php  
$template_directory      = 'mytemplate';  
$template_name           = '3-spalten CSS';  
$template_version        = '2.6';  
$template_platform       = '2.6.x';  
$template_author         = 'Christian Sommer';  
$template_license         = 'GNU GPL';  
$template_description     = 'WB-Template auf Basis des Tutorials von Maxdesign.';  
  
// Definition der Menüelemente (wenn mehr als 1 Menü verwendet wird)  
$menu[1]                 = 'Hauptnavigation';  
$menu[2]                 = 'Unternavigation';  
  
// Definition der Inhaltsblöcke (wenn mehr als 1 Block verwendet wird)  
$block[1]                = 'Hauptinhalt';  
$block[2]                = 'News';  
?>
```

Danach kopiert man alle Dateien, die vom Template verwendet werden in ein .zip Archiv und installiert dies im Websitebaker.

12.12.11

Für CMSs, wie z.B. Websitebaker gibt es verschiedene Module zum Download. Mit diesen können verschiedene Arten von Seiten, wie in etwa eine Bildergalerie oder ein Terminkalender, eingefügt werden.

Auf <http://www.websitebaker-cms.de/download-module/module.php> können beispielsweise Module für den Websitebaker heruntergeladen werden.

Diese sind im .zip Format vorhanden.



Im Adminbereich der Seite können Module unter Erweiterungen → Module hinzugefügt oder entfernt werden.

Seite hinzufügen

Titel:		Administratoren:
Art:	Calendar	☒ Administratoren
Übergeordnete Datei:	Keine	
Sichtbarkeit:	Öffentlich	
Hinzufügen		Zurücksetzen

Will man nun eine neue Seite erstellen, hat man z.B. den Terminkalender zur Auswahl.

09.01.12



Dies ist das Ziel einer Homepageübung. Es sollen 6 Boxen erstellt werden, eine für Navigation, eine für ein Bild, eine mit Willkommenstext, eine mit Copyright, eine mit irgendwas und eine für Werbung.

```
<div id="centering">

  <div id="left">

    <br>

    <ul>
      <li><a href="index.html">Home</a>
    </ul>

    <ul>
      <li><a href="index.html">Test</a>
    </ul>

  </div>
```

Mit Hilfe des *centering* Containers wird die Homepage zentriert. Im *left* Container wird nun durch *ul* & *li* ein Menü aufgebaut. Mit *a href* wird ein Link eingefügt.

```
<div id="top">

  <center></center>

</div>
```

Mit *img* wird in dem *top* Container ein Bild eingefügt.

```
<div id="down">
  <br>
  <center>(c) Maik Riedlsperger 2012</center>
</div>
```

In der *down* Box wird *zentriert* das Copyright eingeblendet.

```
<div id="right">
  <br>
  <br>
  <center>
    Irgendwas
  </center>
</div>
```

In der *right* Box wird ein Probetext eingefügt. Dieser besteht aus 2 *Leerzeilen* und dem *zentrierten* Text *Irgendwas*.

```
<div id="werbung">
  <br>
  <br>
  <center>
    Werbung. :)
  </center>
</div>
```

Ganz rechts in der Werbungsbox. Hier gibt es einfach einen Text zum Simulieren einer Werbung.

```

#centering {
padding-top:5px;
width: 800px;
margin:auto;
}

#top
{
width: 400px;
height: 100px;
margin-left: 150px;
margin-top: -800px;
background-color: #FF0000;
}

#mid
{
width: 375px;
height: 650px;
margin-left: 150px;
background-color: #00FF00;
padding-left:25px;
}

#down
{
width: 400px;
height: 50px;
margin-left: 150px;
background-color: #0000FF;
color: #FFFFFF;
}

#left
{
width: 150px;
height: 800px;
background-color: #FFFF00;
}

#right
{
width: 125px;
height: 800px;
margin-left: 550px;
margin-top: -800px;
background-color: #00FFFF;
}

#werbung
{
width: 125px;
height: 800px;
margin-left: 675px;
margin-top: -800px;
background-color: #FF00FF;
}

```

Mit einer .css werden die Positionen der Boxen definiert.

06.02.12

 **Neue Datenbank anlegen** 

Zuerst wird eine neue Datenbank erstellt.

 **Neue Tabelle in Datenbank test erstellen**

Name: Anzahl der Spalten:

In dieser wird die Tabelle *schule* mit 4 Spalten erstellt.

Erzeuge Tabelle 

Tabellenname:

Spalte	Typ	Länge/Set1	Standard2
ID	INT		Kein(e)
Vorname	TEXT		Kein(e)
Nachname	TEXT		Kein(e)
Ort	TEXT		Kein(e)

Tabellen-Kommentar:

Tabellenformat:

Kollation:

Hier werden nun 4 Spalten erstellt, eine für die *ID* (eine *Integer Variable*), eine für *Vorname/Nachname/Ort*. Hier jedoch als *Text*.

ID	int(11)	1
Vorname	text	Maik
Nachname	text	Riedlsperger
Ort	text	Hinterglemm

OK

Hier werden nun nach der Reihe die Schüler der Klasse eingetragen.

```
<?
    $dbhost = "127.0.0.1";
    $dbuser = "root";
    $dbpass = "";
    $dbdata = "schule";
```

Dann erstellt man eine neue *PHP Datei*. In dieser definiert man 4 Variablen.

\$dbhost ist die IP Adresse der Datenbank (in diesem Fall *localhost* → *127.0.0.1*).

\$dbuser ist der User, mit dem man auf die Datenbank zugreift.

\$dbpass ist das Passwort, mit dem man zugreift (hier leer, da kein Passwort vergeben).

\$dbdata ist die Tabelle, die angezeigt werden soll.

```
@mysql_connect($dbhost, $dbuser, $dbpass);
// Auswahl der Datenbank
@$x=mysql_select_db($dbdata);
if (empty($x)) {
    echo "Fehler beim Verbinden mit dem Datenbankserver<br>";
    exit;
}
```

Dieser Code greift nun mit den angegebenen Variablen auf die Datenbank zu oder gibt eine Fehlermeldung aus.

20.02.12

Ähnlich wie Websitebaker ist auch Joomla ein CMS.

Man legt es ebenfalls auf seinen Webserver (oder wie in unserem Falle auf den lokalen xampp Webserver) und greift in unserem Fall *localhost/joomla* zu und installiert das System.

	css	20.02.2012 11:23	Dateiordner	
	images	20.02.2012 12:04	Dateiordner	
	favicon.ico	03.11.2011 20:45	IrfanView ICO File	10 KB
	index.php	20.02.2012 11:48	PHP-Datei	2 KB
	templateDetails.xml	20.02.2012 11:13	XML-Dokument	1 KB

Ähnlich wie in Websitebaker kann man auch hier ein eigenes Template erstellen.

```
<?php
// No direct access.
defined('_JEXEC') or die;

JHTML::_('behavior.framework', true);
$app = JFactory::getApplication();
?>

<?php echo '<!DOCTYPE html>'; ?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction; ?>" >
<head>
    <jdoc:include type="head" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/system/css/system.css" type="text/css" />
    <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo $this->template ?>/css/template.css" type="text/css" />
</head>
<body>
<div id="wrapper">
    <div id="head">
    </div><!--#head-->
    <div id="content">
        <div id="news"><jdoc:include type="modules" name="right" style="xhtml" /></div><!--#news-->
        <div id="component"><jdoc:include type="component" style="xhtml" /></div><!--#component-->
        <div id="menu"><jdoc:include type="modules" name="left" style="xhtml" /></div><!--#menu-->
    </div><!--#content-->
    <div id="footer"><jdoc:include type="modules" name="footer" style="xhtml" /></div><!--#footer-->
</div><!--#wrapper-->
</body>
</html>
```

Hier der Code der *index.php*.

```
<?xml version="1.0" encoding="utf-8"?>
<extension version="1.6" type="template">
  <name>mynewtemplate</name>
  <creationDate>2008-05-01</creationDate>
  <author>John Doe</author>
  <authorEmail>john@example.com</authorEmail>
  <authorUrl>http://www.example.com</authorUrl>
  <copyright>John Doe 2008</copyright>
  <license>GNU/GPL</license>
  <version>1.0.2</version>
  <description>My New Template</description>
  <files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <folder>images</folder>
    <folder>css</folder>
  </files>
  <positions>
    <position>breadcrumb</position>
    <position>left</position>
    <position>right</position>
    <position>top</position>
    <position>user1</position>
    <position>user2</position>
    <position>user3</position>
    <position>user4</position>
    <position>footer</position>
  </positions>
</extension>
```

Und hier der Code der *templatedetails.xml*.

Strict Standards: Declaration of JButtonPopup::fetchId() should be compatible with that of JButton::fetchId() in C:\xampp\htdocs\joomla\library



Erweiterungen: Installieren

[Installieren](#)[Aktualisieren](#)[Verwalten](#)[Überprüfen](#)[Warnungen](#)

FTP-Kontodaten

Zum Installieren und Deinstallieren von Erweiterungen, benötigt Joomla! FTP-Kontodaten. Bitte diese in die unteren Formularfelder eingeben.

Benutzername

Passwort

Paketdatei hochladen

Paketdatei

Aus Verzeichnis installieren

Pfad zum Paket

Von Webadresse installieren

URL zum Paket

Man erstellt eine *.zip* Datei und lädt diese danach hoch, um sie zu verwenden.
Dies funktioniert auch mit *Modulen* oder *anderen Erweiterungen*.

[Stile](#) | [Templates](#)

Filter: Suchen Zurücksetzen

Bereich auswählen

Template auswählen

	Stil	Bereich	Template	Standard	Zugewiesen	ID
☐	Atomic - Default	Site	atomic	☆		3
☐	Beez5 - Default-Fruit Shop	Site	beez5	☆		6
☐	Beez2 - Parkseiten	Site	beez_20	☆		114
☐	Beez2 - Default	Site	beez_20	☆		4
☐	Bluestork - Default	Administrator	bluestork	★		2
☐	Hathor - Default	Administrator	hathor	☆		5
☐	ADarkHero	Site	mynewtemplate	★		115

Anzeige # 20

In den *Templates* kann man nun das neu erstellte Template auswählen und nutzen.



So sieht schließlich das fertige Template aus.

27.02.12 / 05.03.12

Log in:

Dein Username:

Dein Passwort:

[Noch keinen Account? -> Registrieren!](#)

Das Ziel dieser Übung ist ein funktionierendes Log in- & Registrierungssystem.
Hierzu werden 5 einzelne Seiten, ein (lokaler) Webserver und eine Datenbank benötigt.

```
<?php
session_start();
?>

<h1>Log in:</h1>

<form action="login.php" method="post">
Dein Username:<br>
<input type="text" size="24" maxlength="50"
name="username"><br><br>

Dein Passwort:<br>
<input type="password" size="24" maxlength="50"
name="password"><br>

<input type="submit" value="Login">
</form>

<a href="register.html">Noch keinen Account? -> Registrieren!</a>
```

Wir beginnen mit der *index.php*. In dieser gibt man *Username & Passwort* ein, um sich einzuloggen. Hat man noch keinen Account, kann man sich über einen Link auf die *register.html* einen erstellen.

```

<?php
session_start();
?>

<?php
    $dbhost = "127.0.0.1";
    $dbuser = "root";
    $dbpass = "";
    $dbdata = "adarkhero";

    @mysql_connect($dbhost, $dbuser, $dbpass);

    @$x=mysql_select_db($dbdata);
    if (empty($x)) {
        echo "Fehler beim Verbinden mit dem Datenbankserver ";
        exit;
    }

$username = $_POST["username"];
$password = md5($_POST["password"]);

$abfrage = "SELECT username, password FROM login WHERE username LIKE '$username' LIMIT 1";
$ergebnis = mysql_query($abfrage);
$row = mysql_fetch_object($ergebnis);

if($row->password == $password)
{
    $_SESSION["username"] = $username;
    echo "Login erfolgreich! <br> <a href=\"geheim.php\">Weiter zur Hauptseite!</a>";
    exit();
}
else
{
    echo "Benutzername und/oder Passwort waren falsch. <a href=\"index.php\">Login</a>";
}
?>

```

In der *login.php* werden die Daten der *index.php* verarbeitet. Es wird überprüft, ob die eingegebenen Daten mit denen in der Datenbank übereinstimmen. Bei „ja“ erhält man einen Link auf die *geheim.php*, konnte man aufgrund von falschem Username oder Passwort nicht eingeloggt werden, erhält man eine Meldung diesbezüglich.

```

<?php
session_start();
?>

<form action="register.php" method="post">
Dein Username:<br>
<input type="text" size="24" maxlength="50"
name="username"><br>

Dein Passwort:<br>
<input type="password" size="24" maxlength="50"
name="password"><br>

Passwort wiederholen:<br>
<input type="password" size="24" maxlength="50"
name="password2"><br>

<input type="submit" value="Abschicken">
</form>

```

In der *register.html* erscheint ein Formular zur Erstellung eines Accounts. Hierzu werden *Username* und *Passwort* benötigt. Das Passwort muss zur Kontrolle doppelt eingegeben werden.

```

<?php
session_start();
?>

<?php
$dbhost = "127.0.0.1";
    $dbuser = "root";
    $dbpass = "";
    $dbdata = "adarkhero";

    @mysql_connect($dbhost, $dbuser, $dbpass);

    @$x=mysql_select_db($dbdata);
    if (empty($x)) {
        echo "Fehler beim Verbinden mit dem Datenbankserver ";
        exit;
    }

$username = $_POST["username"];
$password = $_POST["password"];
$password2 = $_POST["password2"];

if($password != $password2 OR $username == "" OR $password == "")
{
    echo "Eingabefehler. Bitte alle Felder korrekt ausfüllen. <a href=\"register.html\">Zurück</a>";
    exit;
}

$password = md5($password);

$result = mysql_query("SELECT ID FROM login WHERE username LIKE '$username'");
$menge = mysql_num_rows($result);

if($menge == 0)
{
    $eintrag = "INSERT INTO login (username, password) VALUES ('$username', '$password')";

    $eintragen = mysql_query($eintrag);

```

```

if($menge == 0)
{
    $eintrag = "INSERT INTO login (username, password) VALUES ('$username', '$password')";

    $eintragen = mysql_query($eintrag);

    if($eintragen == true)
    {
        echo "Benutzername <b>$username</b> wurde erstellt. <a href=\"index.php\">Login</a>";
    }
    else
    {
        echo "Fehler beim Speichern des Benutzernames. <a href=\"register.html\">Zurück</a>";
    }
}

else
{
    echo "Benutzername schon vorhanden. <a href=\"eintragen.html\">Zurück</a>";
}

?>

```

In der *register.php* wird die Eingabe von der *register.html* verarbeitet und in eine Datenbank geschrieben. Das Passwort wird mit einem *md5* Wert verschlüsselt.

```

<?php
session_start();

?>

<?php
if(!isset($_SESSION["username"]))
{
    echo "Bitte erst <a href=\"index.php\">einloggen</a>";
    exit;
}
else{
    echo "Willkommen auf dieser Seite! :)";
}

?>

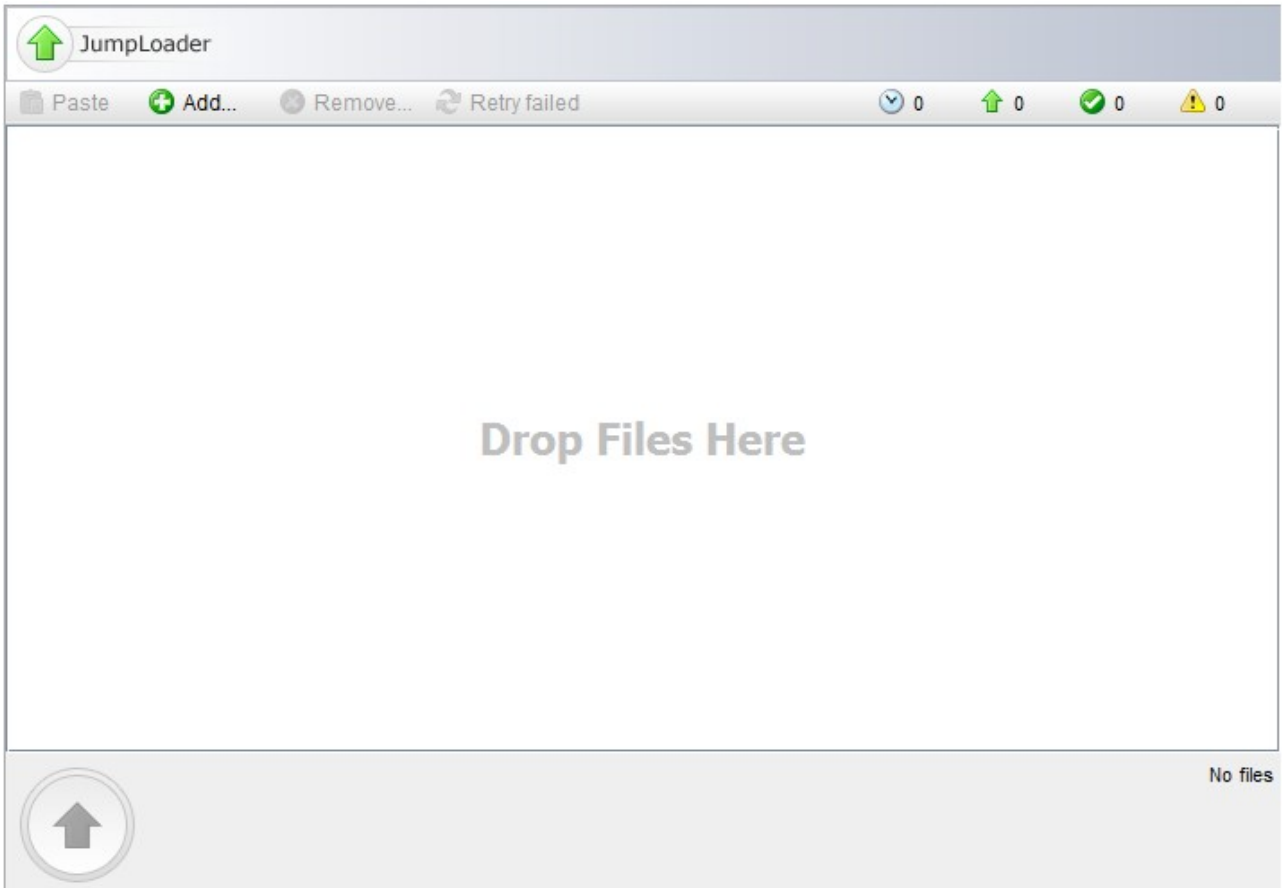
```

Ist man nun eingeloggt kann man auf die *geheim.php* zugreifen und der Text „Willkommen auf dieser Seite“ erscheint. Hat man sich zuvor nicht eingeloggt, erscheint ein Text mit der Aufforderung, dies zu tun.

session_start hat den Sinn, Variablen über längere Zeit zu speichern, damit man nicht beim Betreten einer anderen Seite gleich wieder ausgeloggt wird.

16.04.12

Jumploader ist eine Javadei, welche dem User erlaubt, Dateien ins Netz zu laden.



Mittels *Drag and Drop* oder dem *Add Button* lassen sich Dateien auswählen und mittels dem Upload Pfeil unten links hochladen.

```
<applet id="jumpLoaderApplet" name="jumpLoaderApplet"
  code="jmaster.jumploader.app.JumpLoaderApplet.class"
  archive="jumploader.jar"
  width="715"
  height="500"
  mayscript>
  <param name="uc_imageEditorEnabled" value="true"/>
  <param name="uc_uploadUrl" value="upload.php"/>
  <param name="uc_partitionLength" value="1000000"/>
</applet>
```

Mit diesem Code wird die *jumploader.jar* auf eine *HTML / PHP Webseite* eingebunden. In der markierten Zeile steht der Pfad zu einer weiteren PHP Datei, die dann die Befehle des Jumploaders ausführt.

```

<?php
//-----
//    partitioned upload file handler script
//-----

//
//    specify upload directory - storage
//    for reconstructed uploaded files
$upload_dir = $_SERVER[ 'DOCUMENT_ROOT' ] . "/schule/jumploader/uploaded/";

//
//    specify stage directory - temporary storage
//    for uploaded partitions
$stage_dir = $_SERVER[ 'DOCUMENT_ROOT' ] . "/schule/jumploader/uploaded/";

//
//    retrieve request parameters
$file_param_name = 'file';
$file_name = $_FILES[ $file_param_name ][ 'name' ];
$file_id = $_POST[ 'fileId' ];
$partition_index = $_POST[ 'partitionIndex' ];
$partition_count = $_POST[ 'partitionCount' ];
$file_length = $_POST[ 'fileLength' ];

//
//    the $client_id is an essential variable,
//    this is used to generate uploaded partitions file prefix,
//    because we can not rely on 'fileId' uniqueness in a
//    concurrent environment - 2 different clients (applets)
//    may submit duplicate fileId. thus, this is responsibility
//    of a server to distribute unique clientId values
//    (or other variable, for example this could be session id)
//    for instantiated applets.
$client_id = $_GET[ 'clientId' ];

```

In der 2. PHP Datei gibt man nun den Pfad (vom Server ausgehend) ein, in dem die hochgeladenen Dateien gespeichert werden sollen. Dieser Pfad wird in der Variable *\$upload_dir* bzw. *\$stage_dir* gespeichert.

23.04.12

Tiny MCE ist ein so genannter *What you see is what you get (WYSIWYG)* Editor. Dieser ist ein Editor, ähnlich Word, mit dem ein User Texte eingeben und formatieren kann, ohne sich mit HTML auskennen zu müssen.

```
<!-- /TinyMCE -->
</head>
<body role="application">
<form method="post" action="tiny.php">
  <div>
    <div>
      <h1>Titel der Webseite</h1>
      <input type="text" name="title"> <br>

      <h1>Inhalt</h1>
      <textarea id="tiny" name="tiny" rows="15" cols="80" style="width: 50%">
        Schreiben Sie hier den Text ihrer Website hinein!
      </textarea>
    </div>

    <br />
    <input type="submit" name="save" value="Submit" />
    <input type="reset" name="reset" value="Reset" />
  </div>
</form>

<script type="text/javascript">
if (document.location.protocol == 'file:') {
  alert("The examples might not work properly on the local file system due to securi
}
</script>
</body>
</html>
```

Zuerst benötigt man eine Datei, in welcher der Editor eingebunden wird. Damit dies funktioniert, benötigt man einige Zeilen an Javascript Code. Danach kann man den Editor mittels *textarea* einbinden. Mit *form* kann man nun den eingegebenen Text als HTML Code an eine 2. Datei übermitteln.

```
<?php
$tiny = $_POST["tiny"];
$title = $_POST["title"];

    $datafile = fopen("html/" . $title . ".php", "a+");
    fwrite($datafile, $tiny);

echo 'Success! :) <br> <a href="html/', $title, '.php">Zu Ihrer Website</a>'
?>
```

Mit der 2. Datei kann man nun die *Eingabe in eine Datei* schreiben. In diesem Fall konnte man auch noch den Titel der Datei, in die geschrieben werden soll eingeben.

07.05.12 / 14.05.12

Die aktuelle Aufgabe ist es, ein eigenes, kleines CMS zu schreiben.

Dies ist ein voll toller Header!

- [Index](#)
- [Stuff](#)
- [Impressum](#)
- [Admin](#)

Dies ist das Impressum.

Right

Dieser Footer wurde mit einem nicht patentierten System geändert.

Dies ist der Grundaufbau der Seite, dieser kann mittels einer *.css* Datei natürlich frei angepasst werden. Der Inhalt jeder *DIV Box* wird aus einer Datenbank ausgelesen.

Log in:

Dein Username:

Dein Passwort:

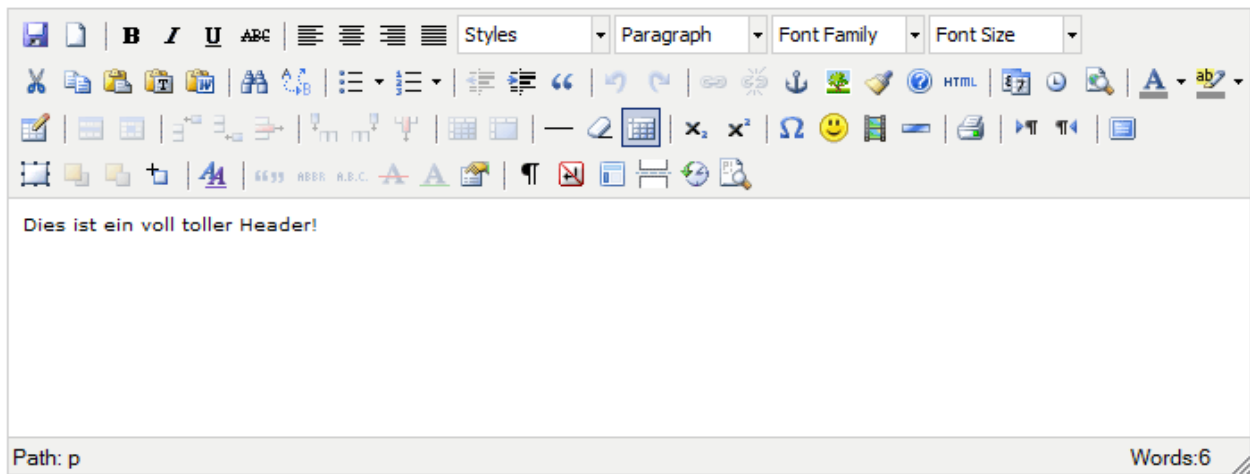
Des weiteren gibt es einen *Adminbereich*, zu dem nur der Administrator des CMS Zugriff hat. Dieser ist mit einem *Log in* geschützt.

Dateien hochladen

Unter *Seite ändern / löschen* kann man nun die Seite wählen, welche man bearbeiten oder löschen will. Will man eine Seite ändern, erhält man erhält man einen Editor, mit dem man den Content erneuern kann.

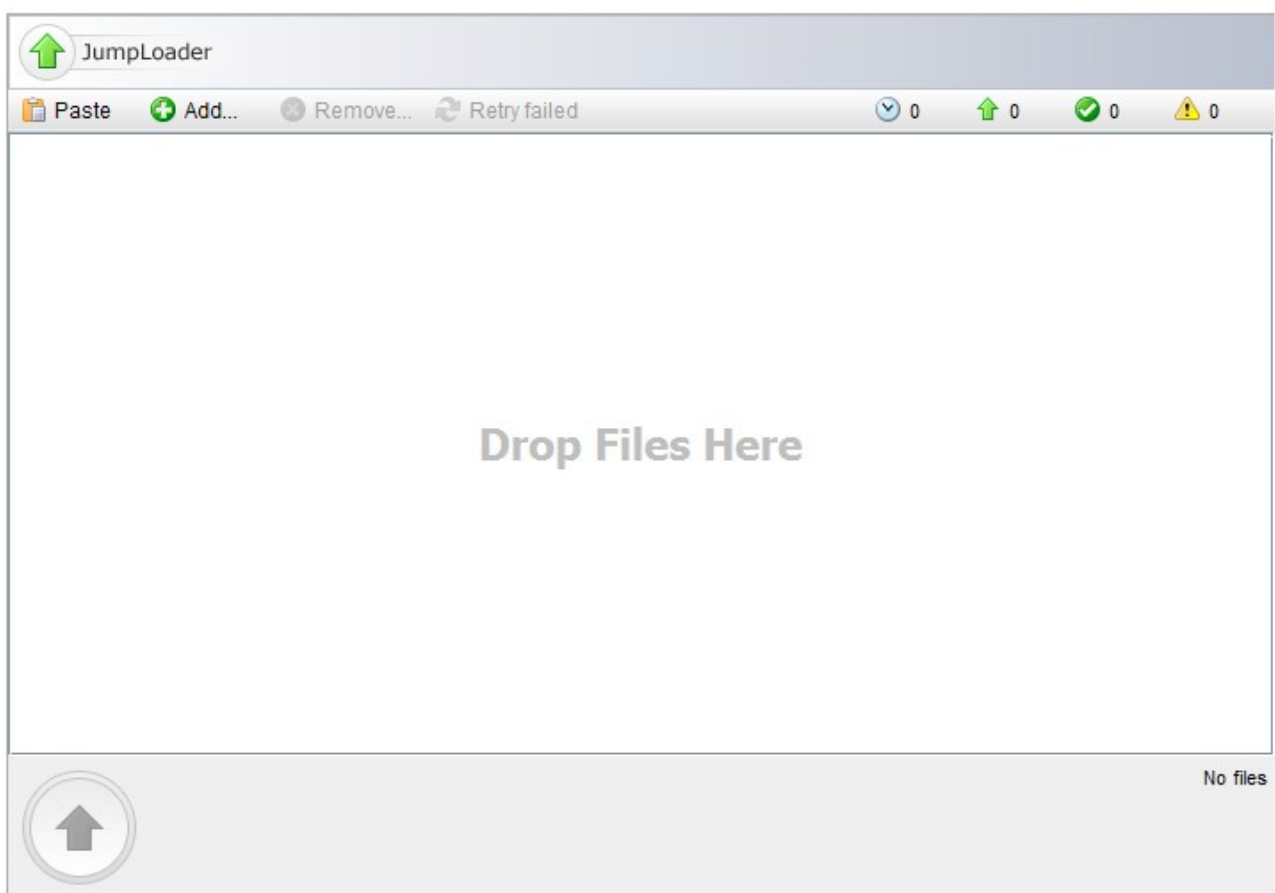
[Header](#) | [Footer](#) | [Links](#) | [Rechts](#) ||| [Zurück zum Adminbereich](#)

Header



The screenshot shows the Tiny MCE editor interface. At the top is a toolbar with various icons for text formatting (bold, italic, underline, ABC), alignment, lists, links, and other functions. Below the toolbar is a text area containing the text "Dies ist ein voll toller Header!". At the bottom of the editor, there is a status bar showing "Path: p" on the left and "Words:6" on the right.

Das selbe gilt für den *Header*, *Footer* etc. Mittels *Tiny MCE* kann man den Inhalt jederzeit ändern.



The screenshot shows the JumpLoader file upload interface. At the top, there is a header bar with the "JumpLoader" logo and name. Below this is a toolbar with buttons for "Paste", "Add...", "Remove...", and "Retry failed". To the right of these buttons are four status indicators: a clock icon with "0", an upload icon with "0", a checkmark icon with "0", and a warning icon with "0". The main area of the interface is a large white box with the text "Drop Files Here" in the center. At the bottom, there is a footer bar with a large circular button containing an upward arrow and the text "No files" on the right.

Mittels *Jumploader* lassen sich Dateien, wie z.B. Bilder ins CMS hochladen.