

# Web Engineering

## Charakter encoding

ASCII (American Standard for Information

Interchange)

- 7 bit Code = 128 Characters (No special characters)
- ISO 8859-1 = 256 characters
  - Windows - 1252 based on ISO 8859-1
  - Unicode more than 100.000 characters
    - first 128 ch. ~~ISO 8859-1~~ ASCII
    - first 256 ch. ISO 8859-1
  - important UTF-8 & UTF-16
- Web Server defines Encoding
- if defined in HTML & in XML ; XML has priority
- Server can force encoding (can be changed by admin or on server-side scripts)
- ASCII don't know german umlauts

## Tabelle

```
<table border="1">  
<tr>  
<td> First</td>  
<td> Second</td>  
</tr>  
<tr>  
<td> Third</td>  
<td> Fourth</td>  
</tr>  
</table>
```

First	Second
Third	Fourth

HTML-Elemente Webseiten

Übersicht über alle

**<table>**

Tabellentag

**<tr>**

Table Row

**<th>**

Table Head

**<td>**

Table Data

**<caption>** Tabellenüberschrift

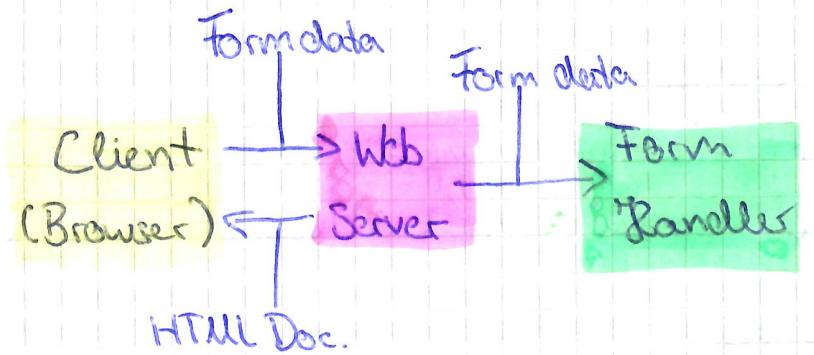
**<th colspan="2">** zwei Zellen horizontal verbinden

# Tabellen

<table>	Tabelle öffnen
<th>	Tabellenfeld (Kopf)
<td>	Tabellenfeld (Körper)
<tr>	Tabellenzeile
<thead>	Tabellenkopf (nicht unbedingt nötig)
<tbody>	Tabellenkörper (nicht unbedingt nötig)
<td colspan="2">	Tabellenfeld (2 verbundene Zellen)
<caption>	Tabellenüberschrift
<td rowspan="2">	Tabellenfeld (2 Zellen vertikal verbunden)
<table border="1">	Rand um Tabelle (1 Pixel dlich)
<table width="50%">	Tabellengröße 50% des Screens
<table width="350px">	Tabellengröße 350 px

Information Name  
③ Chaitin-Graeber  
[REDACTED]  
[REDACTED]

# Forms



Transfer:

1. **HTTP GET** Arguments transferred in URL
2. **HTTP POST** transparently transferred in HTTP Request

The Form Tag:

```
<form id="myform" action="form_handler_URL.php"  
method="get || post">  
<!-- form elements -->  
</form>
```

Text Input Boxes:

```
<input type="text" name="MyText" id="idMyText"  
value="initial value" size="size-of-field"  
maxLength="max_characters_allowed"/>
```

Example: Name:

```
<p>Name: <input type="text" name="StudentID" id="ID-  
studentID" value="" size="10" maxLength="10"/></p>
```

Password Input Boxes:

```
<input type="password" ... /> Siehe Text Input Boxes
```

# Forms

## Radio Buttons:

```
<input type="radio" name="control_group_name"  
id="ID" checked="checked" value="value_if_selected"/>
```

## Checkboxes:

```
<input type="checkbox" name="field_name" ... />
```

## Select Radio Buttons

## Listboxes :

```
<select name="name" id="ID" size="number_of_items-  
visible" multiple="multiple">  
<optgroup label="group_labelname">  
<option> Option 1</option>  
</optgroup>  
</select>
```

## Large Text Areas:

- contains up to 1024 characters

```
<textarea name="..." cols="number_of_cols"  
rows="number_of_rows"> default_text </textarea>
```

## Hidden Fields:

- invisible & used to store data (across pages)

```
<input type="hidden" name="field_name" value="value" />
```

Field 1
---------

page1.html

Field 1
Field 2

page2.html

Field 1
Field 2
Field 3

page3.html

- invisible
- visible

# Forms

## Buttons:

```
<input type="button" name="button-name" value="button-text" />
```

## Images:

- used as graphical buttons with JavaScript event handles

```
<input type="image" name="name" src="url-to-img"/>
```

## File Fields:

- attach a file to form
- needs multipart encoding and POST transfer

```
<form action="handler" method="post"
```

```
enctype="form/multipart">
```

```
<input type="file" name="field-name" size="displayed-size" />
```

```
</form>
```

## Submit & Reset Button:

```
<input type="submit" name="name" id="ID" value="button-text" />
```

```
<input type="reset" ... />
```

## Field Set

- Grouping of form elements

```
<fieldset>
```

```
<[REDACTED] legend> <label>[REDACTED]</label>
```

```
<input ... />
```

```
<input ... />
```

```
</fieldset>
```



# Forms

## Form Field Accessibility:

- `TabIndex` specifies the order in which fields are focused after using the Tab Key
- `accesskey` provides a shortcut via using `<ALT>+<accesskey>`  
`<input ... tabIndex="1" accesskey="F"/>`  
`<input ... tabIndex="2" accesskey="E"/>`

## Preventing Change of Data:

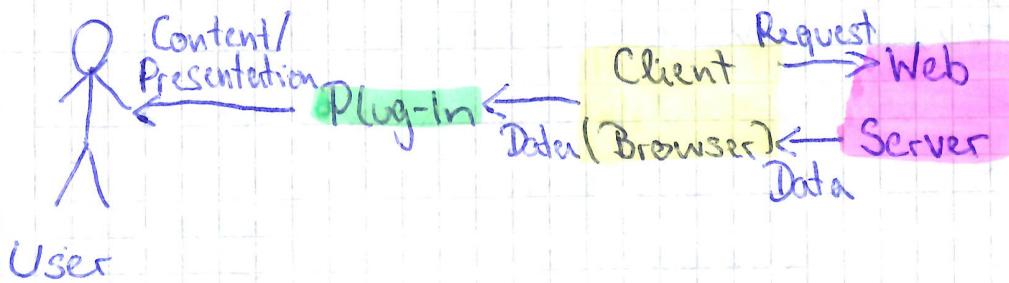
### 1. Method: Read only

```
<input type="text" ... readonly="readonly"/>
```

### 2. Method: Disabling a Field

```
<input type="text" ... disabled="disabled"/>
```

# Plug-Ins



- Allow the Browser to show non HTML contents (MIDI, Java Applet, Flash)
- represented with `<object>` tag. (former `<embed>` o.)
- `<applet>`

`<embed>` (veraltet) :

```
<embed src="c:\temp\test.midi">  
</embed>
```

↳ embed with Filpath

```
<embed src="http://www.youtube.com/v/1JwgP44Ap3E">  
</embed>
```

↳ embed with URL

Problem: if needed plug-in isn't installed, you won't get informed that there's content which can't be display (some browsers have a solution for that problem)

The `<object>` tag:

```
<object classid="..." id="ID-Value" codebase=  
"base_for_object_code_URL" codetype="MIME-type">  
<param name="parameters_name" value="parameters_value"/>  
...  
</object>
```

# Plug-ins

Problem:

- some newer browsers don't know <embed>
- some older browsers don't know <object>

Solution: combine both!

Don't know  
that tag  
(will be ignored)

<object ....>  
<param ...></param>  
<embed ...></embed>  
</object>

don't know that tag  
(will be ignored)

Legende

- Old Browser
- New Browser

no Parameter (will be ignored)

# Cascading Style Sheets

Describes the presentation of a Markup Language

so können z.B. Klassen definiert werden, die einer bestimmten Formatierung entsprechen

Durch Klassen einfacher auf viele Stellen anzuwenden

Beispiel (CSS im HTML Head)

<head>

```
<style type="text/css">  
p.heading { font-weight: bold; text-decoration:  
underline; }  
</style>
```

</head>

```
<p class="heading">Heading 1</p>
```

Beispiel ohne CSS

```
<p><b><u> Heading 1 </u></b></p>
```

CSS Style Definition:

```
<style type="MIME-type" media="destination-media">  
</style>
```

MIME-type:

is usually "text/css"

Weitere sind z.B.

all - all media type devices

aural - speech & sound synthesizers

print - printers

tv - television-type devices

# Cascading Style Sheets

Where to put Style Definitions:

Method 1: directly in the head of the html doc.

Method 2: link a separate file in the head. <link> tag

<link> tag:

<head>

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

</head>

It is also possible to link two style sheets to one html doc.