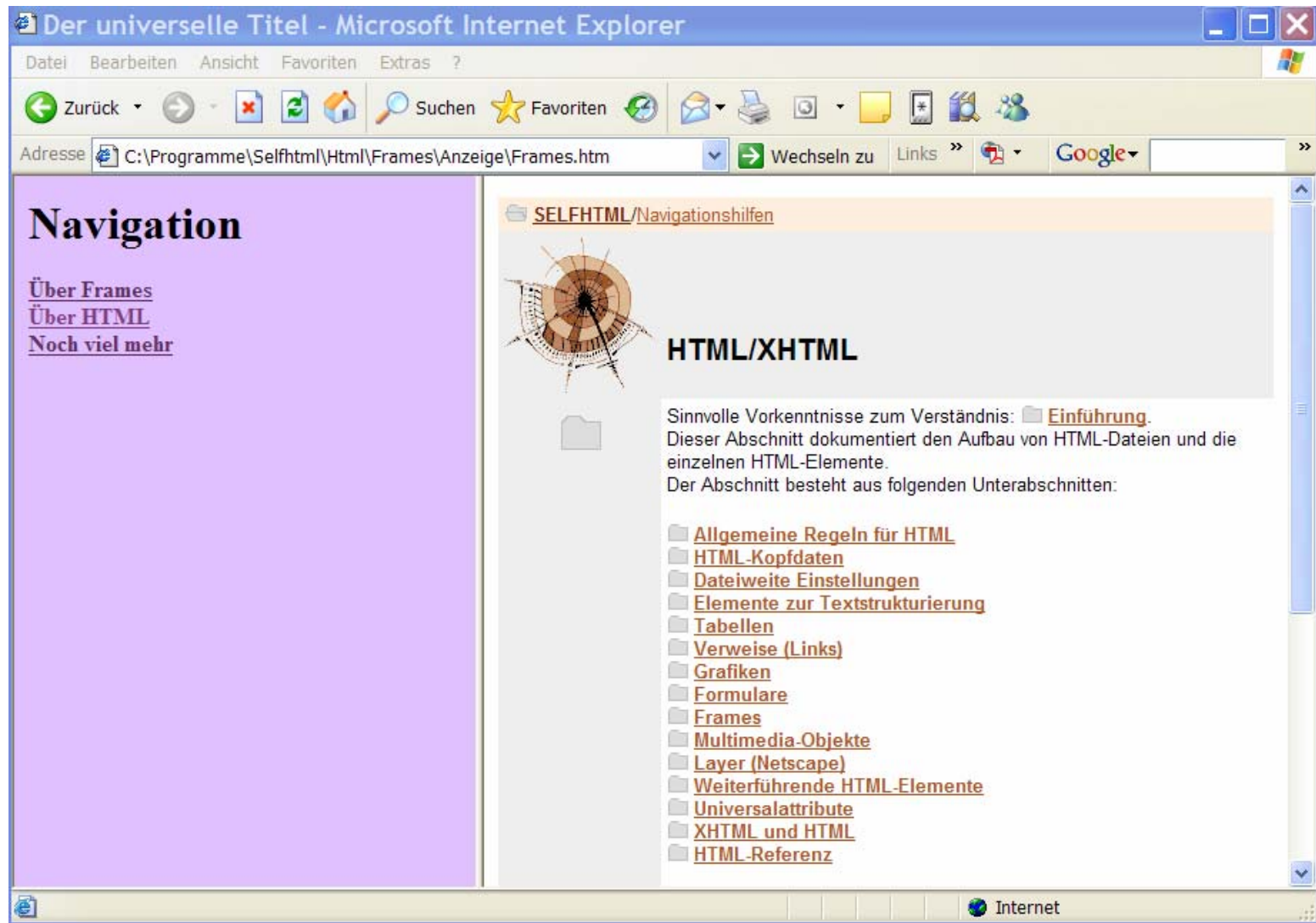


Frameset als Layoutmanager – Beispiel



bei Puristen verpönt;
nicht in der Variante "strict" enthalten,
sondern eigene Variante "frameset"

- Aufteilung des Browserfensters in mehrere Abschnitte
 - ⇒ horizontal oder vertikal
 - ⇒ fest oder verschiebbar
 - ⇒ jeder Abschnitt mit eigenem Scrollbalken bei Bedarf
- komplexere Aufteilung durch Schachtelung von Framesets
- in jedem Abschnitt ("Frame") wird eine andere HTML-Datei dargestellt

Pro

- ⇒ stabile Navigationsleiste
- ⇒ stabile Kopf- oder Fußzeile
- ⇒ Vergleich von Infos

Contra

- ⇒ nicht jeder Browser kann's
- ⇒ Problem bei geringer
Bildschirmauflösung
- ⇒ Ladezeiten für mehrere Dateien

Struktur eines Frameset

```
<!DOCTYPE HTML PUBLIC  
    "-//W3C//DTD HTML 4.01 Frameset//EN"  
    "http://www.w3.org/TR/html4/frameset.dtd">
```

eigene HTML-Variante,
nicht die "reine Lehre"

```
<html >
```

```
<head>
```

```
    <title>Text des Titels</title>
```

```
</head>
```

```
<frameset ... >                <!-- Aufteilung des Framesets -->  
    <frame ... >                <!-- Definition Framefenster 1 -->  
    <frame ... >                <!-- Definition Framefenster 2 -->  
    <noframes>  
        Ihr Browser kann keine Frames anzeigen.  
    </noframes>  
</frameset>
```

```
</html >
```

Aufteilung eines Frameset

■ zeilenweise aufteilen

```
<frameset rows="100, *, 60" >
```

```
<frameset rows="1*, 5*, 3*" >
```

■ spaltenweise aufteilen

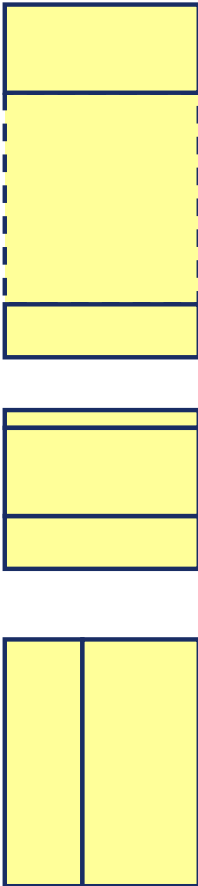
```
<frameset cols="40%, 60%" >
```

■ Maßangaben

- ohne absolut in Pixel
- * restlicher Platz
- Zahl * anteilig
- % relativ in Prozent

Rest

Verhältnis



Frame definieren und füllen

- einzelnes Frame und seine Attribute definieren

<code><frame src="startseite.htm"</code>	Anfangsseite
<code>name="rechts"</code>	Name für Hyperlinks
<code>frameborder="1"</code>	mit / ohne Rand
<code>marginwidth="5" marginheight="7"</code>	Innenabstand in Pixel
<code>noresize</code>	Größe nicht änderbar
<code>scrolling="yes"></code>	mit / ohne Scrollbalken

- Hyperlinks bekommen zusätzlich das Zielfenster

`News`

transitional

- Zielfensterbasis ggfs. im HTML-Header definieren

`<base target="rechts">`

- Frameset verlassen

`Raus`

Frameset vs. Tabelle

Frameset → mehrere HTML-Seiten

- Frames unabhängig voneinander scrollbar
- Navigationsmenü wird nur einmal geladen
- Direktes Ansteuern von Frames von außen kaum möglich
- schnell verwirrend

Tabelle → eine HTML-Seite

- Navigationsmenü scrollt weg
- übersichtlichere Zustandsverwaltung
- schwierig für Screenreader (Barrierefreiheit!)
 - ⇒ Screenreader liest Tabellen als Tabellen, nicht als Layout

mit CSS geht
es auch...

Barrierefreies Layout ohne Tabellen

Ziel: Webseite soll für Sehbehinderte von einem Screenreader vorgelesen werden können

- Seite wird als inhaltlich logische Sequenz von Blöcken aufgebaut
 - ⇒ diese Sequenz definiert die Vorlesereihenfolge des Screenreaders
- einzelne Blöcke werden mit CSS auf die Seite geschoben
 - ⇒ Attribute `float`, `margin`, `clear`
- www.fbi.h-da.de ist so aufgebaut

Zusammenfassung

- Problematik des Layouts (WYSINWYG)
- Tabelle als Layoutmanager
 - ⇒ Struktur einer Tabelle mit Zeilen und Spalten
 - ⇒ Bemaßung von Rahmen, Abständen etc.
- Framesets als Layoutmanager
 - ⇒ Struktur eines Framesets (Aufteilung einer Seite in unabhängige Bereiche)
 - ⇒ Bemaßung von Rahmen, Abständen etc.
- Barrierefreies Layout ohne Tabellen (⇒ CSS)

Jetzt können Sie die Elemente auf einer einfachen HTML-Seite anordnen!

Steuerelemente für Formulare

Einzeiliges Textfeld	<input type="text" value="normalerweise für Eingabe"/>
Textfeld mit Scrollbalken	<input type="text" value="Beliebiger mehrzeiliger Text"/>
Checkboxen	<input checked="" type="checkbox"/> <input type="checkbox"/>
Radiobuttons	<input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>
Combo-Box	<input type="text" value="1. Möglichkeit"/>
Schaltflächen	<input type="button" value="Abschicken"/> <input type="button" value="Zurücksetzen"/>

Platzierung mittels blinder Tabelle

Funktion von Formularen

■ Formulare dienen der Eingabe von Daten

⇒ eingegebene Daten werden an Server übermittelt und dort ausgewertet

⇒ es gibt 2 Möglichkeiten der Datenübertragung

– **get** übermittelt Parameter für Abfrage (z.B. Suchmaschine)

– **post** übermittelt Daten zwecks Speicherung (z.B. Bestellung)

vgl. Reload
im Browser

■ beliebigen Bereich im HTML-Body markieren

```
<form action="/cgi-bin/Echo.pl" method="get">
```

Steuerelemente (Eingabefelder, Auswahllisten,

Buttons...) und sonstige HTML-Tags und CSS-Formatierung

```
</form>
```

hier: Übergabe der
Daten an Perl-Skript

■ Alternative Aktion: Formulardaten per eMail verschicken

```
action="mailto:Meier@xyz.de"
```

⇒ unsicher, weil von der Installation beim Surfer abhängig

Eingabefelder

Einzeiliges Textfeld	<input type="text" value="normalerweise für Eingabe"/>
Textfeld mit Scrollbalken	<input type="text" value="Beliebiger mehrzeiliger Text"/>

■ einzeilige Textbox

```
<input type="text" name="zuname" size="30" maxlength="40"
value="Mustermann" readonly>
```

standalone tag

⇒ `name` und `value` wird an Server übermittelt

⇒ `value` kann vorbelegt sein

⇒ mit `readonly` reine Anzeige

⇒ `size` und `maxlength` für Anzeigelänge und Speichergröße

■ Variante: Passwortfeld mit *-Anzeige (aber ohne verschlüsselte Übertragung!)

⇒ wie oben, jedoch `type="password"`

■ mehrzeiliges Textfeld (bei Bedarf mit Scrollbalken)

```
<textarea name="feedback" cols="50" rows="10">
  editierbarer Text
</textarea>
```

3.4 HTML: Formulare

Auswahllisten



■ Listbox

```
<select name="top4" size="3">
  <option>1. Möglichkeit</option>
  <option>2. Möglichkeit</option>
  <option value="3">3. Möglichkeit</option>
  <option>4. Möglichkeit</option>
</select>
```

- ⇒ `size` bestimmt die Höhe in Zeilen
- ⇒ Vorauswahl ggfs. mit `<option selected>`
- ⇒ angezeigter Text wird als ausgewählter Wert übertragen, sofern kein `<option value="xyz">` definiert ist

■ Mehrfachauswahl mit zusätzlichem Attribut `multiple`

■ Combobox wie Listbox, jedoch `size="1"`

Radiobuttons und Checkboxes

Checkboxes Radiobuttons	<input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>
----------------------------	--



- Radiobuttons als Gruppe von Knöpfen, die sich gegenseitig auslösen

(Auswahl 1 aus n)

- ⇒ Gruppierung erfolgt durch identischen `name`

- ⇒ der `value` wird als Wert der Gruppe übertragen

```
<input type="radio" name="credit" value="MC">
```

```
Mastercard<br>
```

```
<input type="radio" name="credit" value="Visa">
```

```
Visa<br>
```

```
<input type="radio" name="credit" value="Amex">
```

```
American Express
```

- Checkboxes für Boole'sche Eingabe

"on" wenn value fehlt

```
<input type="checkbox" name="zutat" value="salami"> Salami <br>
```

- ⇒ übermittelt wird der `value` nur für angekreuzte Checkboxes

- ⇒ Vorauswahl durch Attribut `checked`

Schaltflächen und verborgene Felder

Schaltflächen

Abschicken

Zurücksetzen

■ allgemeine Schaltflächen für JavaScript-Ereignisse

```
<input type="button" name="Start" value="Startseite"
  onClick="self.location.href='http://www.xyz.de/' ">
<button type="button" name="Start" value="Startseite"
  onClick="self.location.href='http://www.xyz.de/' "> Text
und/oder Bild
</button>
```

■ Schaltfläche zum Absenden der Formulardaten

⇒ wie oben, jedoch `type="submit"`

■ Schaltfläche zum Löschen der Formulardaten

⇒ wie oben, jedoch `type="reset"`

■ verborgenes Datenfeld (z.B. für Sessionverwaltung)

```
<input type="hidden" name="sessionId" value="4711">
```

Daten werden nur
übertragen, wenn die
Felder ein name-Attribut
haben!

Gestaltung und Tastatursteuerung

- optische Gestaltung des Formulars mit den üblichen Mitteln von HTML
 - ⇒ exakte Ausrichtung durch blinde Tabelle oder CSS
 - ⇒ Formatierung mittels CSS
- Tabulatorreihenfolge entsprechend der Reihenfolge in der HTML-Datei
 - ⇒ oder explizit mit Attribut `tabindex="1"` usw.
- Tastaturkürzel definieren mit `accesskey="x"`
 - ⇒ verlangt vom Benutzer alt+x
(nicht erkennbar, muss beschriftet werden !)

Zusammenfassung

- Grundidee Formulare (Übertragung von Daten an den Web-Server)
- Aufbau von Formularen
 - ⇒ 1- und mehrzeiliges Textfeld (`<input type="text"... >` bzw. `<textarea>`)
 - ⇒ Listbox und Combobox (`<select...><option>...`)
 - ⇒ Radiobuttons und Checkboxes (`<input type="radio" name="x"... >` bzw. `<input type="checkbox"... >`)
 - ⇒ Schaltflächen und verborgene Felder `<input type="button" ... onclick... >` bzw. `<input type="hidden"... >`
 - ⇒ Tastatursteuerung (`tabindex="1"` bzw. `accesskey="x"`)

Jetzt wissen Sie alles um eine (statische) HTML-Seite zu entwickeln!

■ vergleichbar mit komfortabler Programmierumgebung

- ⇒ Hilfe beim Einfügen von HTML Code
 - übersichtliche, menügeführte Auswahl der HTML-Tags
 - Dialoge und Wizards für komplexere Auswahlen
- ⇒ Auffinden durch syntaxabhängige Einfärbung unterstützt
- ⇒ oft mit Preview des Ergebnisses

■ aber: man editiert und sieht letztlich den HTML-Code

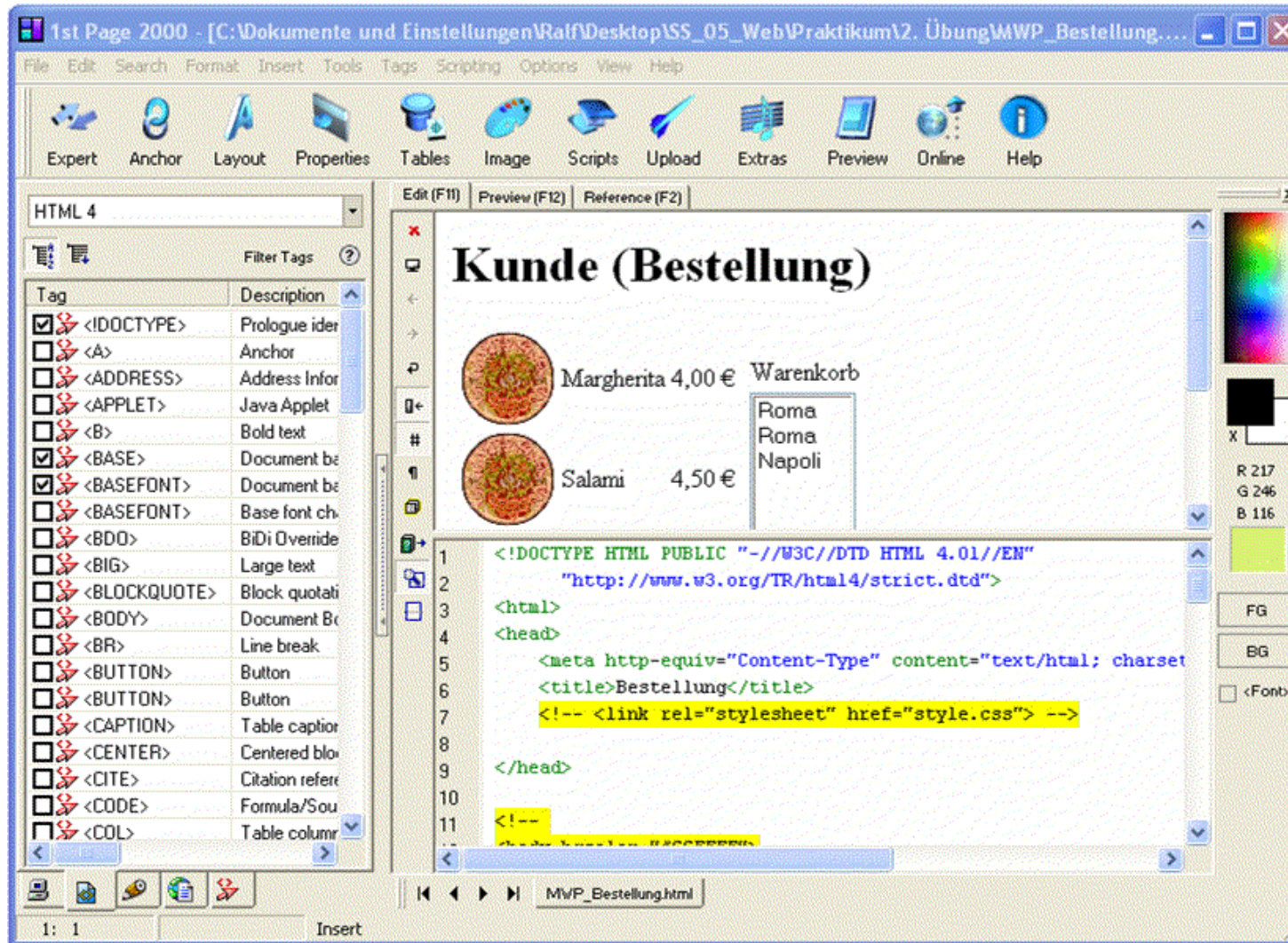
- ⇒ der Autor denkt und kontrolliert visuell
- ⇒ er muss Änderungswünsche in HTML "übersetzen" und die richtige Stelle im Code finden - wie ein Programmierer

■ Vorteil: hand-optimierter HTML-Code, neueste Features nutzbar

■ Nachteil: Programmierer muss die Schnittmenge der Browser finden

3.5 HTML: Werkzeuge

HTML Editor - Beispiel



CSE HTML Validator, <http://www.w3.org>,
<http://validator.de.selfhtml.org>, ...

- Browser ignorieren normalerweise unbekannte oder falsche Tags und Attribute
 - ⇒ es gibt keine Fehlermeldung, allenfalls Fehlverhalten
 - ⇒ Tippfehler werden nicht gemeldet
- Browser sind unterschiedlich tolerant gegenüber Fehlern in HTML
 - ⇒ was der eine ignoriert, bewirkt beim anderen u.U. sehr fehlerhafte Darstellung
- deshalb: HTML Code vor der Veröffentlichung validieren
 - ⇒ d.h. Prüfung anhand der Spezifikation: Syntax, Tag- und Attributnamen, Schachtelungsregeln, ...
 - ⇒ Prüftiefe und -güte verschiedener HTML-Validatoren variieren
- auch generiertes HTML (z.B. aus PHP) validieren !



The W3C Markup Validation Service is a free service that checks Web documents in formats like HTML and XHTML for conformance to W3C Recommendations and other standards.

VALIDATE YOUR MARKUP

VALIDATE BY URL

Address:

Enter the URL of the page you want to check. Advanced options are available from the [Extended Interface](#).

VALIDATE BY FILE UPLOAD

Local File:

Select the file you want to upload and check. Advanced options are available from the [Extended File Upload Interface](#).

Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#) on the W3C QA Website.

WYSIWYG Tools

GoLive, Dreamweaver, FrontPage, Publisher

■ Hoher Komfort (ähnlich Word)

- ⇒ HTML wird nicht mehr "programmiert"; Anweisungen und Attribute werden problemorientiert über Dialoge definiert
- ⇒ HTML ist nur "internes Datenformat"
- ⇒ HTML kann vom Autor betrachtet werden; muss aber nicht

■ nur die Formatiermöglichkeiten von HTML sind erlaubt

- ⇒ Tabellen und Grafikeinbindung gemäß HTML

■ Darstellung etwa so wie im Browser

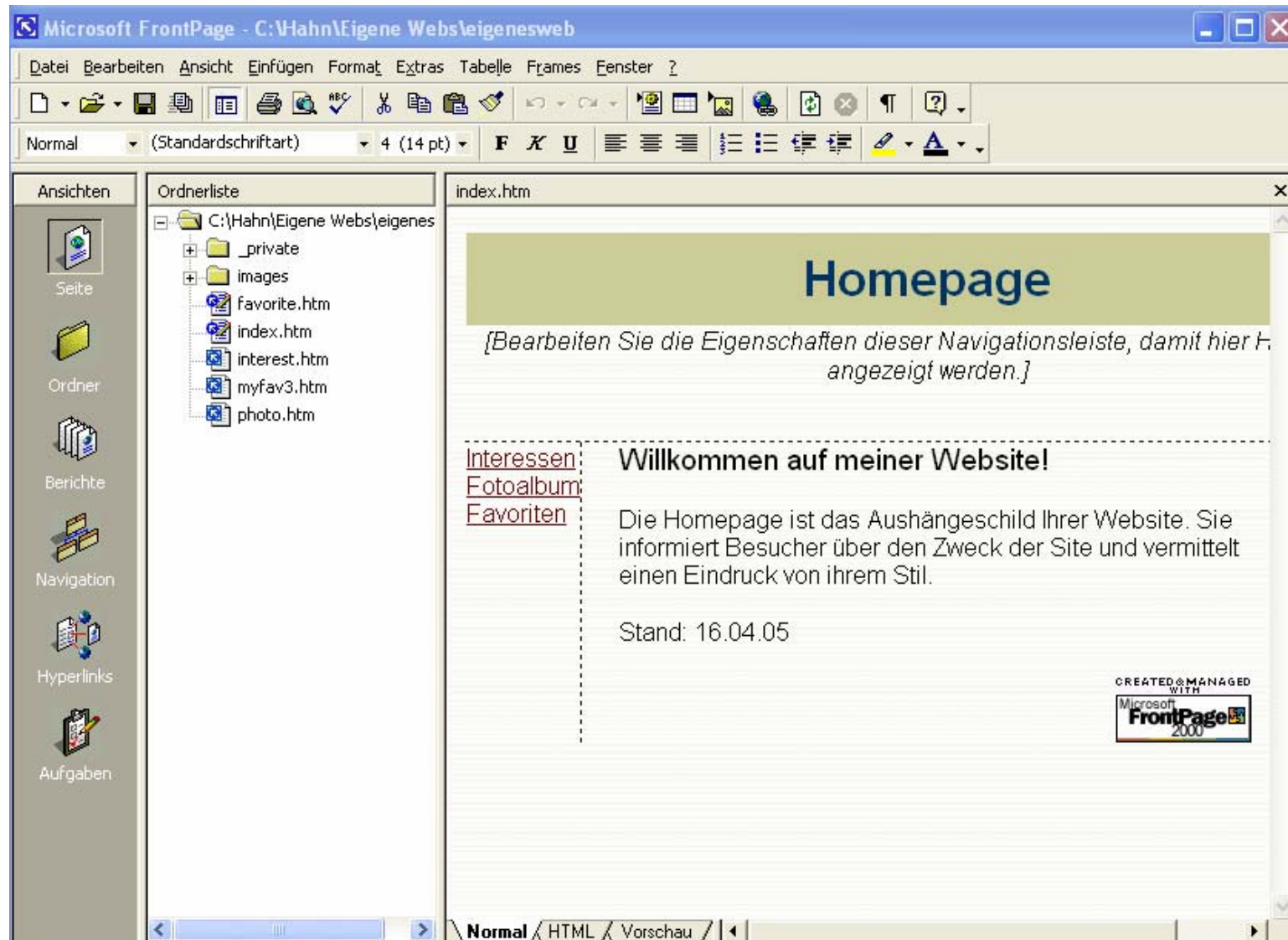
- ⇒ eine mögliche WYSIWYG-Variante unter vielen
- ⇒ zusätzlich Vorschau mit verschiedenen Browsern

■ generiertes HTML ist meist "multi-browser-tauglich"

man muss die
Prinzipien
verstehen

da haben die Entwickler des Tools bereits getüftelt

WYSIWYG Tool - Beispiel

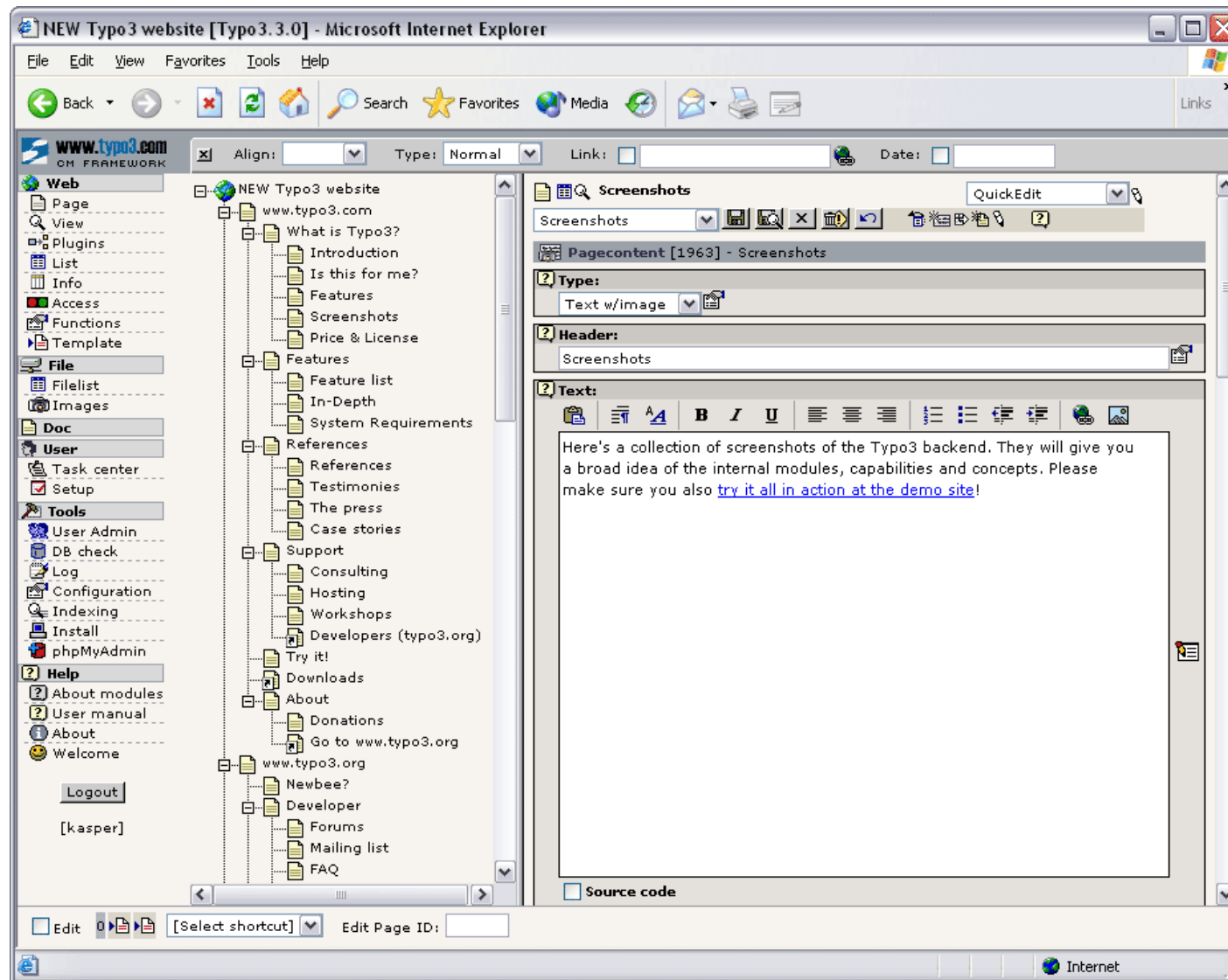


Website Verwaltung

- verwaltet Website mit allen zugehörigen Dateien
 - ⇒ Websites haben meist sehr viele kleine Dateien;
darunter leidet leicht die Übersicht
 - ⇒ eigener problemangepasster "Explorer"
- Übersichtsdarstellungen
 - ⇒ einlaufende und abgehende Hyperlinks jeder Seite
 - ⇒ alle zugehörigen Bilder einer Seite
 - ⇒ alle Seiten, die ein Bild verwenden
- Konsistenzprüfung und Korrektur von Hyperlinks
 - ⇒ korrigiert Website-interne Hyperlinks bei Datei-Umbenennung
- Suchen und Ersetzen über gesamte Website

spezielle
Problematik
des Web

Website Verwaltung - Beispiel



Vorlagen und Assistenten

Ziel:
Web Publishing
für jedermann

- fertige Vorlagen für typische Websites
 - ⇒ in großer Zahl verfügbar
 - ⇒ entsprechen oft nicht den eigenen Vorstellungen
 - ⇒ sind aber ein guter Startpunkt für weitere Abwandlungen
 - ⇒ helfen gegen das Vergessen wichtiger Elemente
- Vorlagen sind über Dialogfolge (Assistenten) konfigurierbar
- Beispiele für Sites und Seiten
 - ⇒ Diskussionsforum, Firmenpräsenz, Persönliche Homepage, Kundenunterstützung
 - ⇒ Benutzerregistrierung, Feedback-Formular, Gästebuch, Stellenangebot, Produktbeschreibung

Export aus anderen Tools

- früher unbrauchbar, heute erstaunlich gut
 - ⇒ z.B. PowerPoint für MSIE mit XML und Style Sheets
 - ⇒ sogar in der Größe skalierbar
- generierter HTML-Code sehr komplex, Vielzahl von Dateien
 - ⇒ praktisch schon fast wieder proprietäres Dateiformat
- nicht unterstützt: HTML-Rohform als Zwischenstufe bei Konvertierung existierender Dokumente
 - ⇒ Export für Nachbearbeitung in HTML Tool
- (spezialisierte) Tools konkurrieren bezüglich Benutzungsoberfläche
 - ⇒ keine Herstellerbindung mehr via Dateiformate

Zusammenfassung

■ HTML-Grundlagen

- ⇒ Grundgerüst: DOCTYPE, <html>, <head>, <body>, <title>, charset...
- ⇒ Schreibregeln und Syntax
- ⇒ Tags und Attribute
- ⇒ Hyperlinks

■ Layout

- ⇒ Tabellen (nicht nur für Layout!), Framesets

■ Formulare

- ⇒ Buttons, Listen, Datenübermittlung

■ Werkzeuge

Jetzt wissen Sie alles um eine komplette und strukturierte HTML-Seite zu entwickeln!