HTML Einführungskurs

Autor	Nachricht	
oxpus	Betreff des Beitrags:	Di 08.Nov, 2005 13:03

Den hatten wir doch schon ☹

Richtig, aber es gibt ja noch etwas mehr, nämlich Frames, Scripte, Styles, Metas, etc.

Hier sollen nun die Grundlagen für das Gerüst der Seite erklärt werden. Schliesslich will man nicht alles in schwarzem Text mit Schriftart "Times Roman" und Standardgrösse formatiert auf einer weissen leeren Seite weitergeben und auch weniger oft soll eine Seite nur aus einem Frame (jede Seite ist ein Frame, wenn man es genau nimmt) bestehen.

Also ran an den Speck.

Die Tags HTML, HEAD und BODY habt Ihr ja nun kennen gelernt, aber wie machen das andere, im Browser-Titel einen Namen anzugeben? Wie funktioniert das mit einer automatischen Weiterleitung und wodurch wird z.B. einer Suchmaschine mitgeteilt, was in einer Seite vorhanden ist?

Wie, das ist alles mit HMTL machbar? Ja, seht selber:

> Zitat:
>
> <title>OXPUS :: HTML-Kurs</title>

Wer bis hierher aufgepasst habt, weiss, was diese Zeile bewirkt:

Richtig! In der Titelleiste des Browsers erscheint "OXPUS :: HTML-Kurs".

Und auch genau nur dafür ist der Tag <title> da. Dabei sollte man beachten, daß hier keine Formatierungen, Sonderzeichen, Grafiken oder ähnliches verwendet werden dürfen. Lediglich Text wird hier fehlerfrei dargestellt.

Diese Zeile und auch die folgenden werden alle im Kopf der HTML-Datei, also zwischen <head> und </head> untergebracht.

Doch wie werden weitere Inhalte deklariert, bzw. anderen Programmen zur Verfügung gestellt?

Es gibt ja schliesslich in Word-Dokumenten auch Autoren, wann erstellt und zuletzt geändert und weiter Angaben?

Dafür ist der Tag <meta> zuständig. Hiermit lassen sich viele Angaben zu der Seite machen. Sei es wie im Beispiel weiter oben der Zeichensatz, der Autor, Bemerkungen und auch automatische Weiterleitungen.

der Tag besteht aus 2 grundsätzlichen Varianten:

> Zitat:
>
> <meta name=name content=inhalt>
>
> <meta http-equiv=bezeichner content=inhalt>

Mit <meta> wird diese Zeile eingeleitet, "name", bzw. "http-equiv" stellen den Bezeichner dar und "content" ist der Inhalt der Bezeichner.

Das Prinzip funktioniert ähnlich einer Variablenzuweisung in z.B. Basic, bedeutet aber etwas ganz anderes:

Während "name" lediglich allgemeine Definitionen enthält, stellt "http-equiv" die zugewiesenen Informationen anderen Programmen (z.B. Suchmaschinen) zur Verfügung.

Da man mit Meta-Tags eine Menge definieren kann, seien hier nur die wichtigsten als Beispiele genannt. Dazu immer in Blau eine kurze Erklärung dazu:

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

[color=blue]Weist dem Browser an, den entsprechenden Zeichensatz "iso-8859-1 zu verwenden

<meta name="author" content="oxpus">

[color=blue]Definiert den Autoren dieser Seite

<meta http-equiv="refresh" content="1; URL=http://www.oxpus.de/phpbb/portal.php">

[color=blue]Erzeugt eine automatische Weiterleitung nach X Sekunden zur angegebenen "URL="

<meta name="description" content="Dieser Beschreibungstext soll einem Anwender im Suchdienst bei Auffinden dieser Datei erscheinen.">

[color=blue]Beschreibung einer Seite für Suchdienste

<meta name="date" content="2001-12-15T08:49:37+00:00">

[color=blue]Datum der Datei. Format muss nach den internationalen Regeln erstellt werden

<meta name="robots" content="noindex">

[color=blue]Weist Suchrobotern an, diese Seite nicht zu indizieren

<meta name="keywords" lang="de" content="Ferien, Griechenland, Sonnenschein">

[color=blue]Legt deutsche Schlüsselwörter für Suchroboter fest

<meta name="keywords" lang="en-us" content="vacation, Greece, sunshine">

[color=blue]Legt englische Schlüsselwörter für Suchroboter fest

<meta http-equiv="Content-Script-Type" content="text/javascript">

[color=blue]Definiert die enthaltenden Script-Abschnitte als Javascript

<meta http-equiv="Content-Style-Type" content="text/css">

[color=blue]Definiert die enthaltenen Style-Abschnitte im CSS-Format

<meta http-equiv="expires" content="0">

[color=blue]Weist einen Proxy an, diese Seite immer erneut zu laden und nicht als dem Cache zu verwenden

<meta name="generator" content="TextPad">

[color=blue]Stellt die Information zur Verfügung, mit welchem Programm die Datei erstellt wurde

<meta http-equiv="cache-control" content="no-cache">

[color=blue]Weist einen Server an, diese Seite nicht zwischenzuspeichern

<meta name="revisit-after" content="20 days">

[color=blue]Weist einen Suchroboter an, in 20 Tagen nochmal vorbei zu schauen

Es kann natürlich wild kombiniert und auch der Inhalt unter "content" frei verwendet werden. Es ist aber zu bedenken, daß nicht jeder Browser alle Metas verstehen und auslesen kann!

So, daß war es zunächst vom Aufbau einer HTML-Seite. Im nächsten Beitrag widmen wir uns der Einteilung der Seite in Frames, um unterschiedliche Inhalte darstellen zu können.