

Document Object Model

Prof. Dr.-Ing. Thomas Barth
Anwendungsentwicklung und IT-Systemarchitekturen
Studiengang Wirtschaftsinformatik
RFH Köln
barth@rfh-koeln.de

Statische vs. dynamische Webseiten mit HTML

- ✓ Generelle Fragestellung:
Welche Möglichkeiten bestehen, die Benutzbarkeit einer Webseite zu erhöhen?
- ✓ Klassischer Ablauf mit (statischen) HTML-Seiten:
 - HTML-Dokument komplett laden und lesen ... Interaktion möglich durch ...
 - ... das Ausfüllen von Formularen
 - ... und das Senden der ausgefüllten Formulare per Knopfdruck
 - ... per Link Seite wechseln
 - dieses verlinkte HTML-Dokument komplett laden, lesen, interagieren usw.
- ✓ Zur Interaktion steht auf Client-(Browser-)Seite JavaScript zur Verfügung
- ✓ Zur Verwaltung der Daten, die dargestellt werden können, steht auf dem Server z.B. PHP (Programmiersprache) und eine Datenbank zur Verfügung
- ✓ „Dynamik“ auf HTML-Seite (also NICHT von vorneherein feststehender, statisch in HTML formulierter Inhalt) wird dadurch erreicht, dass ...
 - der Inhalt aus der Datenbank ausgelesen und
 - in eine HTML-Seite integriert wird
 - die dann zum Client übertragen wird

Statische vs. dynamische Webseiten mit HTML

- ✓ Interaktion (in einer „klassischen“ Desktop-Anwendung) verläuft meist anders:
 - Daten auswählen
 - Funktionalität auf Daten ausführen (z.B. suchen, erzeugen, ändern, einfügen, löschen)
 - Daten „verarbeiten“
 - ✓ Sichten auf die Daten und deren Verarbeitung erfolgt meist in einer geschlossenen Anwendung, in unterschiedlichen Fenstern
 - ✓ Reaktion der Anwendung auf Aktion des Nutzers erfolgt meist **hinreichend schnell** (aber: Eindruck subjektiv!)
 - ✓ Der „Rahmen“ der Anwendung bleibt über die Nutzung meist relativ konstant
 - ✓ Der Inhalt von Dialogen, Fenster usw. ändert sich gemäß der Interaktionen
 - ✓ Unterschied zur Interaktion mit HTML-basierten Webseiten??!
 - Neuladen einer HTML-Seite erfordert „viel“ Zeit (... meist mehr Zeit als die Änderung der GUI einer „normalen“, nicht-WWW-basierten Anwendung)
- bessere Benutzbarkeit könnte durch Beschleunigung der Aktualisierung von Bildschirmhalten erreicht werden

Statische vs. dynamische Webseiten mit HTML

- ✓ Verbesserungsmöglichkeit für die Benutzung wäre eine schnellere Reaktion auf die Aktionen des Nutzers OHNE Nachladen einer kompletten HTML-Seite
- ✓ „**Dynamic HTML**“ (**DHTML**) beschreibt eine Menge von Technologien, um die Interaktion zu beschleunigen und Web-Anwendungen benutzbarer zu machen
- ✓ DHTML beschreibt damit KEINE Erweiterung von HTML um weitere Tags o.ä.
- ✓ „Dynamic“ bezeichnet die Möglichkeit, das aktuell **im Browser geladene** und dargestellte **HTML-Dokument** zu **ändern**, **OHNE** es **neuladen** zu müssen
- ✓ Die notwendigen „Werkzeuge“ dafür sind
 - Eine **Repräsentation eines HTML-Dokuments**, die für „alle“ Browser standardisiert ist
 - Eine Möglichkeit, diese **Repräsentation zur Laufzeit zu ändern** und darstellen zu können
- ✓ Diese Werkzeuge sind konkret
 - Das **Document Object Model (DOM)** als standardisierte Repräsentation
 - **JavaScript** als Programmiersprache zur Änderung des DOM

Document Object Model

- ✓ HTML-Dokumente (XML-Dokumente ebenfalls) sind strukturell **Bäume**
 - Browser wandeln geladene HTML/XML-Dokumente intern in Baum-Strukturen um (durch sog. **parser**)
- ✓ **Dynamische Änderung von HTML-Seiten** bedeutet also die **Änderung von Daten innerhalb eines Baums durch JavaScript**
- ✓ Beispiel:

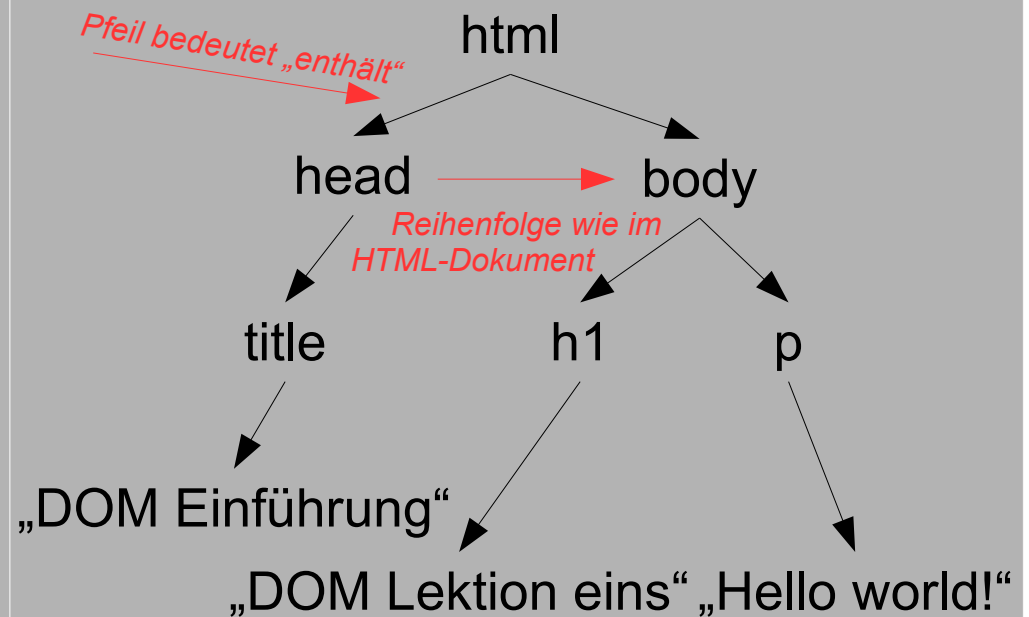
HTML-Beschreibung

```
<html>
  <head>
    <title>DOM Einführung</title>
  </head>
  <body>
    <h1>DOM Lektion eins</h1>
    <p>Hello world!</p>
  </body>
</html>
```

„tag ‚html‘ enthält die tags ‚head‘ und ‚body““

vs.

DOM im Web-Browser



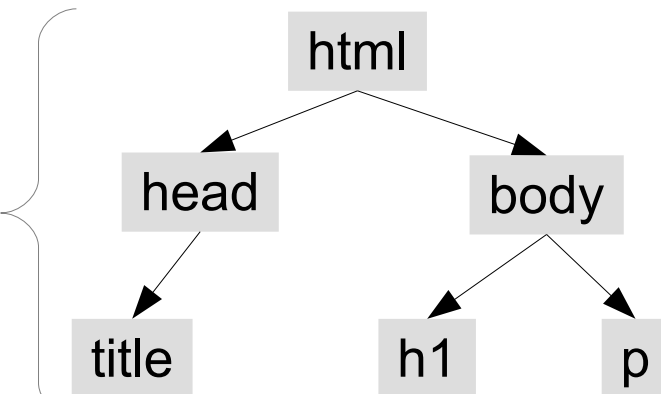
Exkurs: Bäume

- ✓ **Bäume** sind in der **Graphentheorie** als Paar von zwei Mengen $G=(V, E)$ definiert und werden in vielen Anwendungsbereichen (der Informatik) verwendet, z.B. sind Dateisysteme ebenfalls baumartig strukturiert
- ✓ Bäume sind spezielle Graphen, die durch eine Menge von **Knoten** („**Vertices**“) und **Kanten** („**Edges**“) als Verbindungen zwischen zwei Knoten beschrieben werden

- ✓ **Beispiel:**

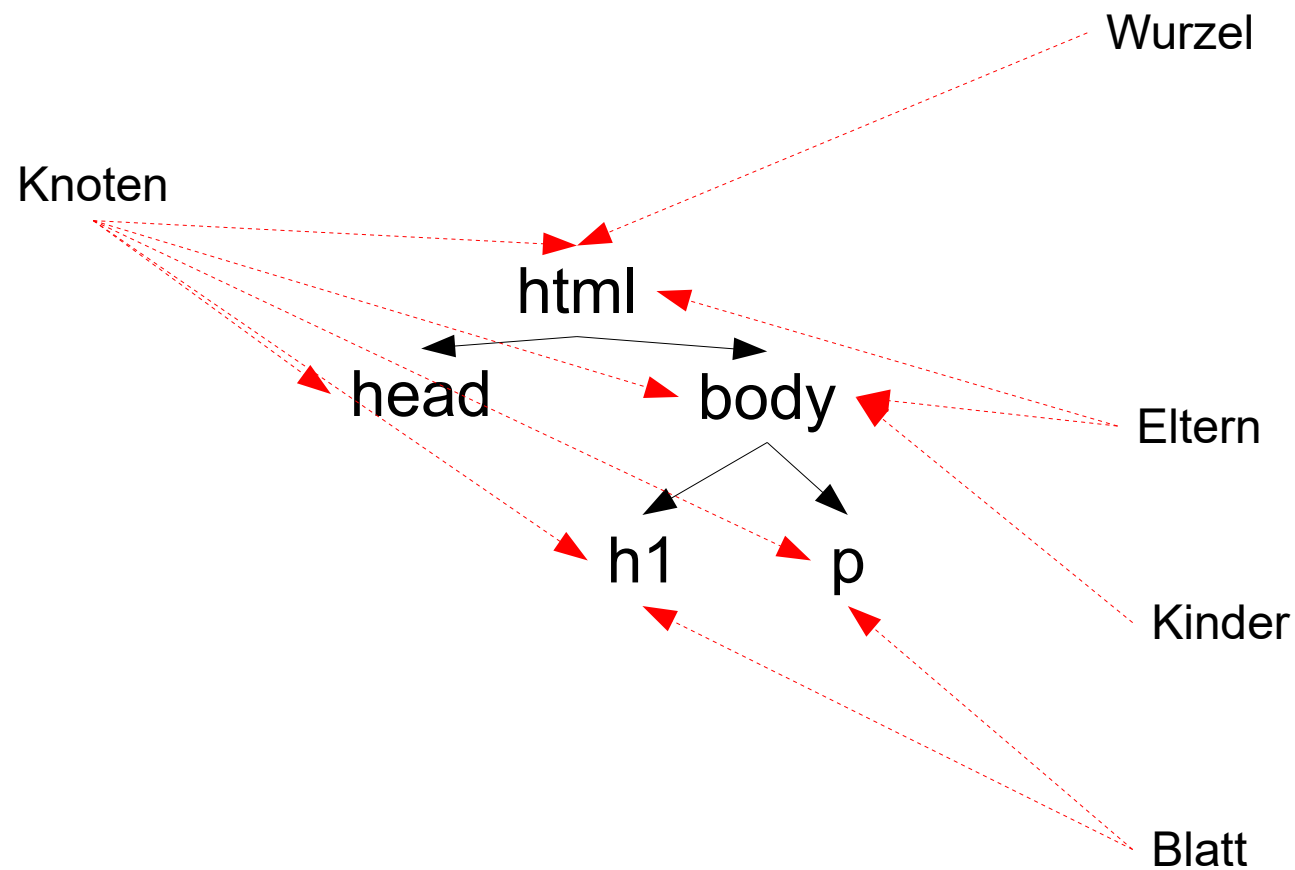
$G=(\{\text{html, head, body}\},$
 $\{(\text{html,head}),(\text{head,title}),(\text{html,body}), (\text{body,h1}), (\text{body,p})\})$

Beschreibung



- ✓ Bäume zeichnen sich durch folgende Eigenschaften aus:

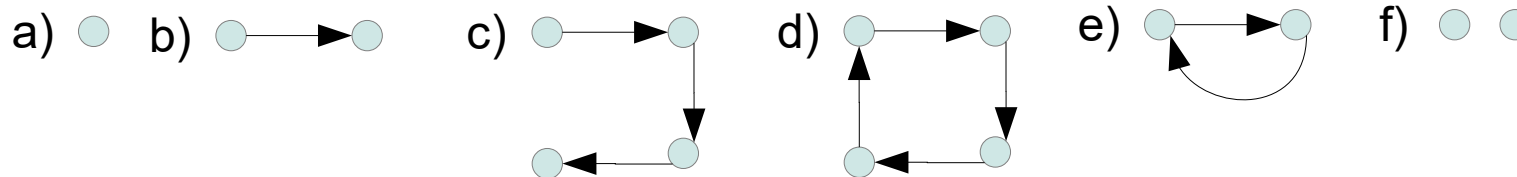
- **Genau** ein Knoten, die **Wurzel** („**root**“), hat keine Vorgänger
- Die Nachfolger eines Knotens heißen **Kinder/Kindknoten** („**child**“, „**children**“)
- Der Vorgänger eines Knotens heißt **Elter/Elternknoten** („**parent**“)
- Knoten ohne Nachfolger heißen **Blätter** („**leafs**“)
- Knoten mit gleichem Vorgänger sind **Geschwister** („**siblings**“)
- Alle Knoten, die nicht Wurzel sind, haben genau einen Vorgänger
- Alle Knoten, die keine Blätter sind, haben beliebig viele Nachfolger



✓ **Q1:**

Beschreiben Sie, welche Eigenschaften einen Baum (Graph!) auszeichnen.

Sind die folgenden Graphen Bäume? Begründen Sie jeweils die Antwort.



✓ **Q2:**

Stellen Sie ein beliebiges HTML-Dokument als Baum dar und identifizieren Sie Wurzel, innere Knoten und Blätter als Beispiele für Eltern-, Kind- und Geschwisterknoten.

✓ **Q3:**

Wenn Sie einzelne Knoten in diesem Baum (aus Q2) gezielt auswählen müssten, welche Möglichkeiten können Sie sich dazu vorstellen?