Interaktive Medien

Human-Computer Interaction
Fachbereich Informatik
Universität Hamburg

Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Prof. Dr. Frank Steinicke
Human-Computer Interaction, Universität Hamburg

Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Trennung von Inhalt und Darstellung

Inhalt ↔ Darstellung

- Unterscheidung von **logischen** (z.B.) und **physischen** (z.B. <b>) Auszeichnungen
- Auszeichnung erfolgt immer auf **zwei Ebenen**:
  1. Ebene definiert und kennzeichnet Textbereiche
  2. Ebene definiert Formatierung

Inhalt ↔ Darstellung

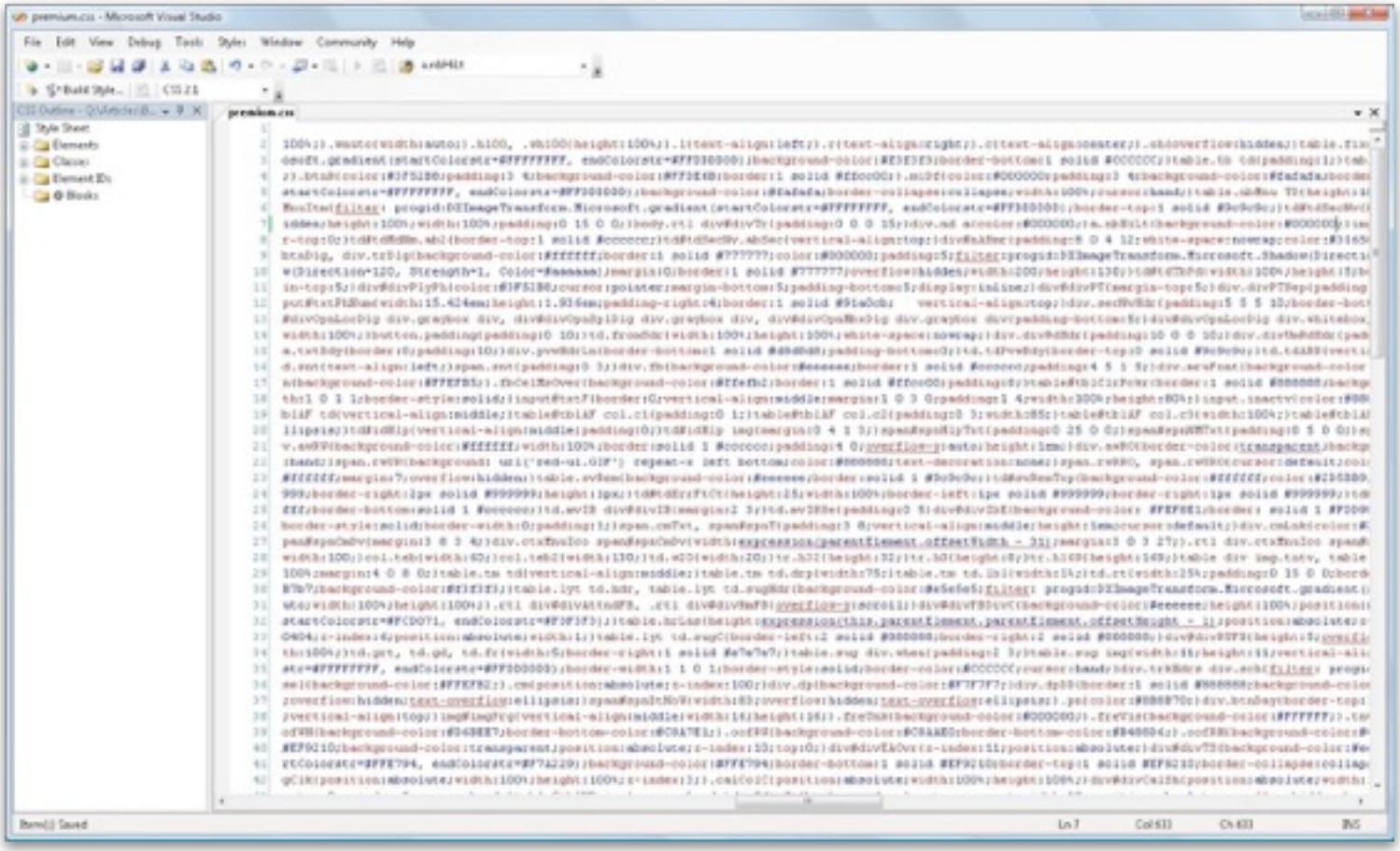
- Browser haben während Anfangszeit von HTML die Formatierung übernommen

  → zu wenig Kontrolle

- **Aber**: Formatierungsangaben nehmen zu und werden komplexer

Inhalt ⟷ Darstellung

Beispiel



Inhalt ↔ Darstellung

➡ **Trennung** von Inhalt und Darstellung

- **Vorteil**: Formatierung eines Stils kann zu späterem Zeitpunkt einfach geändert werden, ohne dass Dokument selbst verändert werden muss
- Änderungen wirken sich auf **alle Dokument-teile**, denen dieser Stil zugewiesen wurde, aus (z.B. Links, Überschriften, Tabellen, etc.)

Stylesheet-Sprachen

- **Stylesheet-Sprachen** sind Beschreibungs-sprachen und mit Formatvorlage zu vergleichen
  - Stilvorlagen (engl. *stylesheet*)
- **Stylesheet-Sprachen** ermöglichen Definition von Stilen für HTML
  1. ***Cascading Style Sheets (CSS)***
  2. *Extensible Stylesheet Language (XSL)*

Cascading Style Sheets

- **Cascading Style Sheets (CSS)** ist deklarative Sprache für Stilvorlagen von strukturierten Dokumenten
- CSS wird vor allem zusammen mit HTML und XML (z.B. bei SVG) eingesetzt

Cascading Style Sheets

- Syntaktisch sehr einfache Sprache, die den **Eigenschaften** von **Blöcken** bestimmte **Werte** zuweist
- Beispiele von Eigenschaften der Schriftformatierung:


```
font, font-family, font-style,
font-variant, letter-spacing,
color, ...
```

Cascading Style Sheets

Beispiel

```
h1 {
    font-family: courier, courier-new, serif;
    font-size: 20pt;
    color: blue;
    border-bottom: 2px solid blue;
}
p {
    font-family: arial, verdana, sans-serif;
    font-size: 12pt;
    color: #6B6BD7;
}
.red_txt {
    color: red;
}
```

CSS-Stile
Einbindung

- Stilangaben können auf unterschiedliche Art und Weise **eingebunden** werden:
  - **Inline** Style
  - **Interner Style Sheet**
  - **Externer Style Sheet**

CSS-Stile

Inline Einbindung

- Über `style`-Attribut eines Elements
- Beispiel:

```
<p style="font-weight: bold;
font-size: 200%">
    Beispieltext
</p>
```

Teilbereiche

- Bereich, der **nicht exakt** von einem HTML-Element umschlossen ist, kann mit `span`- oder `div`-Element umschlossen werden
- Beispiel:

```
<div style="color:#0000FF">
  <h3>Blaue Überschrift.</h3>
  <p>Blauer Text.</p>
</div>
```

CSS-Stile

Interner Style Sheet

- Intention der Style Sheets ist einheitliche Formatierung eines oder mehrerer HTML-Dokumente (z.B. wegen *corporate design*)
- Definition CSS-Stile für alle gleich-artigen Elemente im `head` des HTML-Dokuments

CSS-Stile

Interner Style Sheet

```
<style>
    p {
        font-family: Verdana;
        font-size: 16pt;
    }
    h1 {
        font-family: Verdana;
        color: green;
    }
</style>
```

(legt `style` für Blöcke `p` und `h1` fest)

Cascading Style Sheets

Beispiel: CSS

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page</title>
        <style type="text/css">
            h1 {
                font-family: courier, courier-new, serif;
                font-size: 20pt;
                color: blue;
                border-bottom: 2px solid blue;
            }
            p {
                font-family: arial, verdana, sans-serif;
                font-size: 12pt;
                color: #6B6BD7;
            }
            .red_txt {
                color: red;
            }
        </style>
    </head>
    <body>
        <h1>Arduino SD Card Page with CSS</h1>
        <p>Welcome to the Arduino web page with CSS styling.</p>
        <p class="red_txt">This text is red.</p>
        <p>This paragraph has one word that uses <span class="red_txt">red</span> text.</p>
    </body>
</html>
```

CSS-Stile

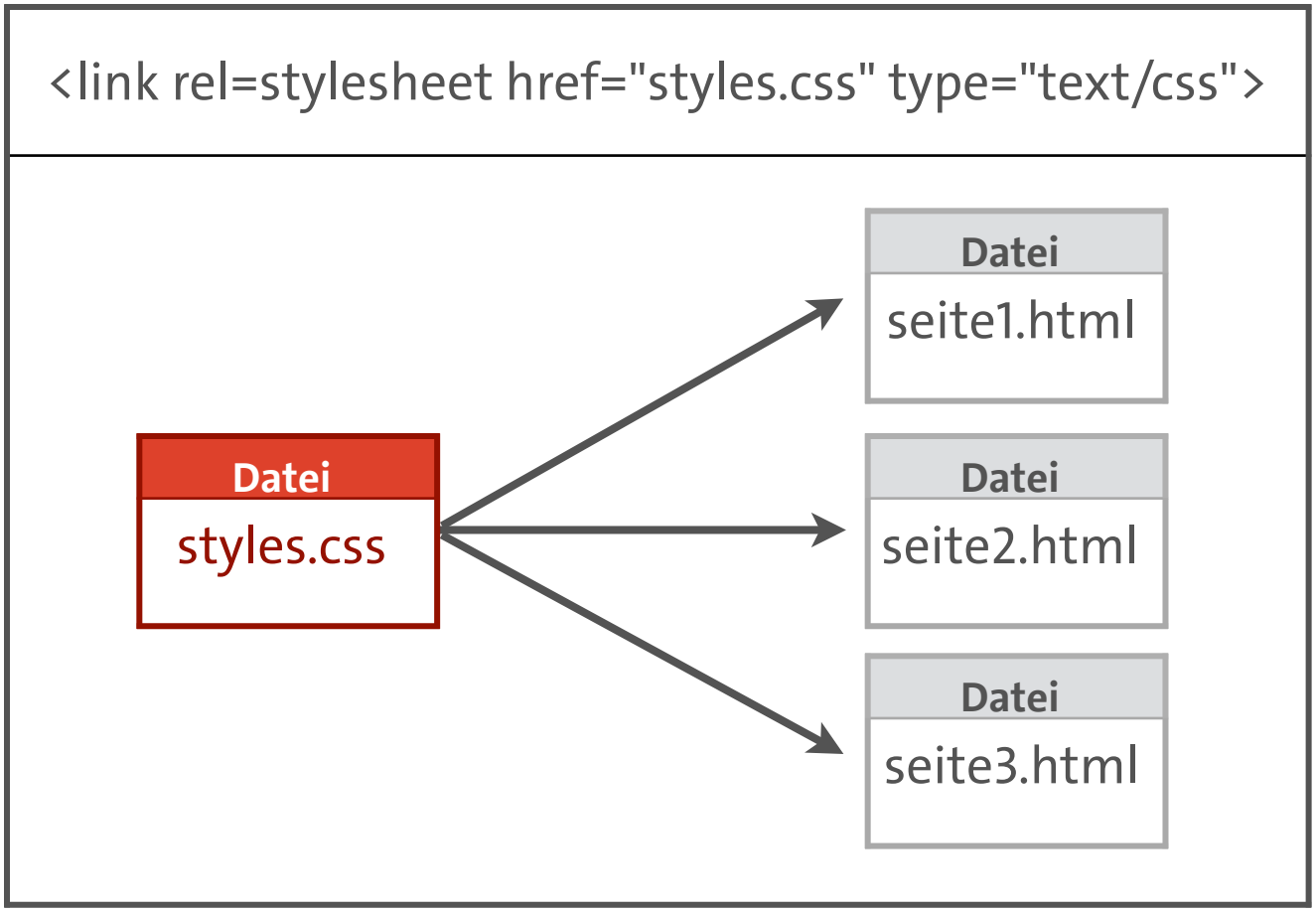
Externer Style Sheet

- Häufigste Einbindungsform:
  CSS-Datei als **externe Datei**, die in `head` des HTML-Dokuments eingebunden wird

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

CSS-Stile

Externer Style Sheet



CSS Selektoren

- definieren auf welche Elemente einer HTML-Seite ein bestimmter Stil angewandt werden soll

```
Selektor {
    Eigenschaft1: Wert1;
    Eigenschaft2: Wert2;
    …
}
```

CSS Selektoren

Beispiel: Formatierung aller Fremdworte in einem Text

---

1. Definition der **Klasse** „`fremdwort`“ als CSS-Stil:

```
.fremdwort {
    color: blue;
}
```

CSS Selektoren

2. Anwendung der **Klasse** in HTML:

```
<h1 class="fremdwort">
    Fremdwort
</h1>
<p>
    Ein <span class="fremdwort">
    Fremdwort </span>.
</p>
```

CSS Selektoren

Arten

- Typselektor: `element`
- ID-Selektor: `#id`
- Nachfahrenselektor: `a b`
- Klassenselektor: `.class`
- Universalselektor: `*`
- …

CSS Selektoren

Beispiele

Selektor	CSS	HTML
Typ	`p {…}`	`<p>…</p>`
ID	`#special {…}`	`<p id="special">…</p>`
Nachfahre	`table p {…}`	`<table><p>…</p></table>`
Klasse	`.bold {…}`	`<p class="bold">…</p>`
Universal	`* {…}`	`jedes Element`

CSS Eigenschaften

- Nach Selektion erfolgt Zuweisung eines Stils über **Eigenschaft-Wert-Paare**, z.B.

```
font-size: 12;
font-style: italic;
```

- Eigenschaft-Wert-Paare können mit Semikolon getrennt **verkettet** werden, z.B.

```
font-style: italic; font-size: large;...
```

CSS Eigenschaften

- Eigenschaft kann auch **mehrere priorisierte** Werte besitzen, z.B.

  ```
  font-family: ‘Times New Roman‘, ‘Times‘, serif;
  ```

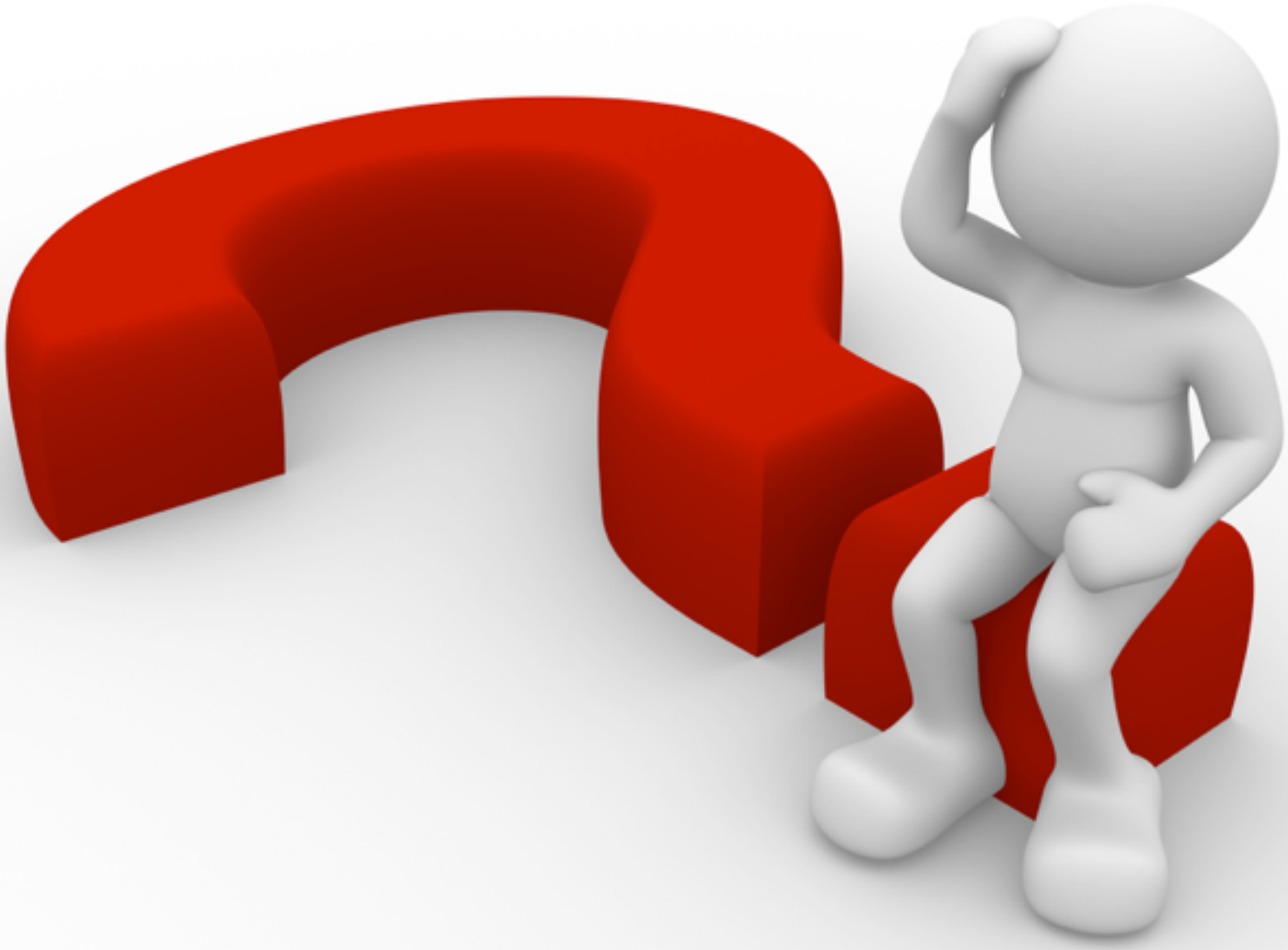
- Eigenschaften sind nach **absteigender Priorität** geordnet, falls Wert nicht verfügbar sein sollte, z.B.:

  Zielsystem kennt Times New Roman nicht, dann wird Times gewählt usw.

CSS Eigenschaften

Anwendungsbereiche

- Textformatierung
- Layout
- Abstände
- …



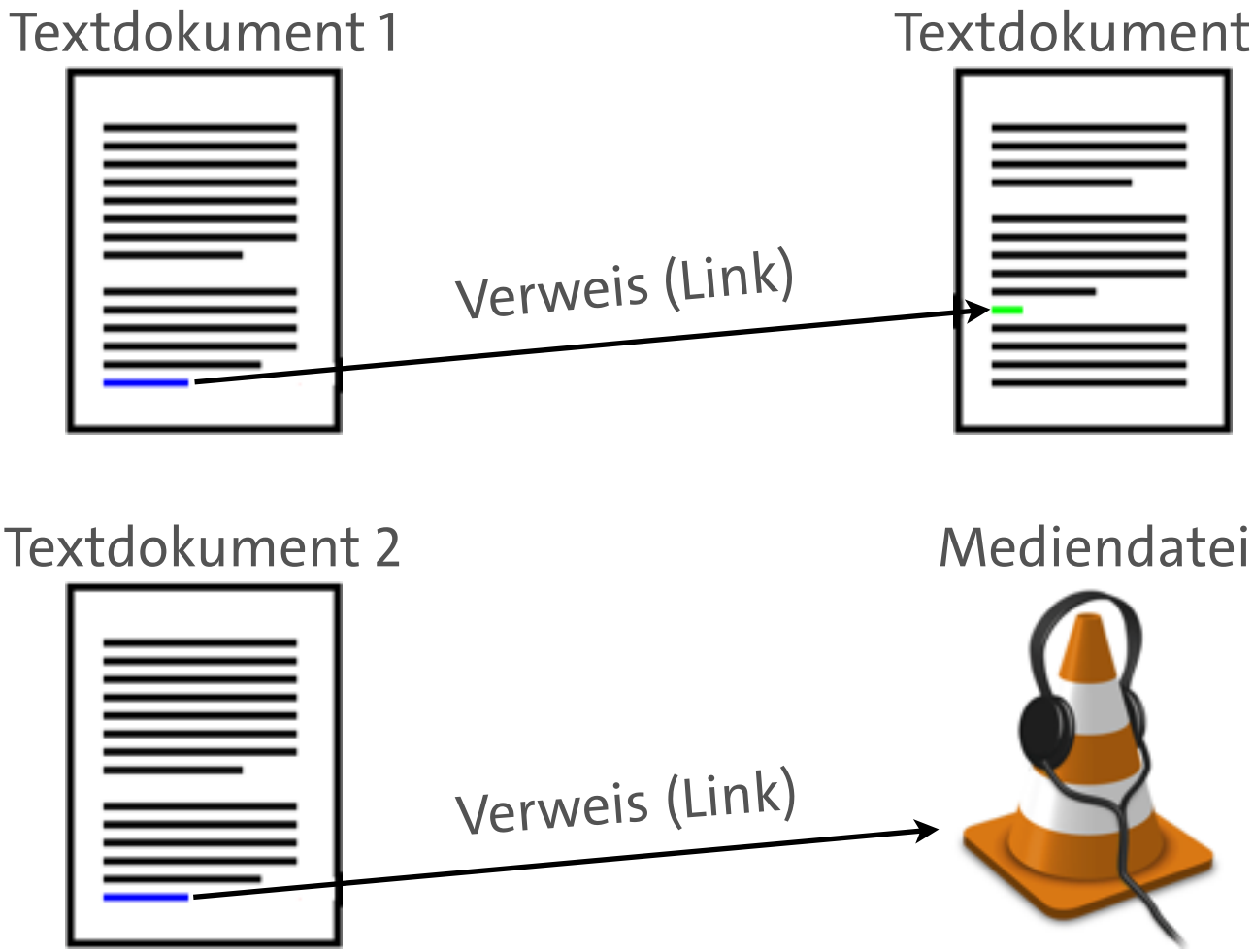
Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Online-Multimedia

HTML

Beispiel: Verweise



HTML
Medieneinbettung

- Multimedia-Dokumente benötigen mehr als nur Text-Elemente, z.B.
  - Bilder
  - Audio
  - Video
  - Präsentationen
  - …

HTML

Medieneinbettung

- Problematisch in früheren Versionen, da dies ursprünglich in HTML nicht vorgesehen war
- In HTML5 durch neue Elemente wesentlich vereinfacht

HTML 5
Bilder

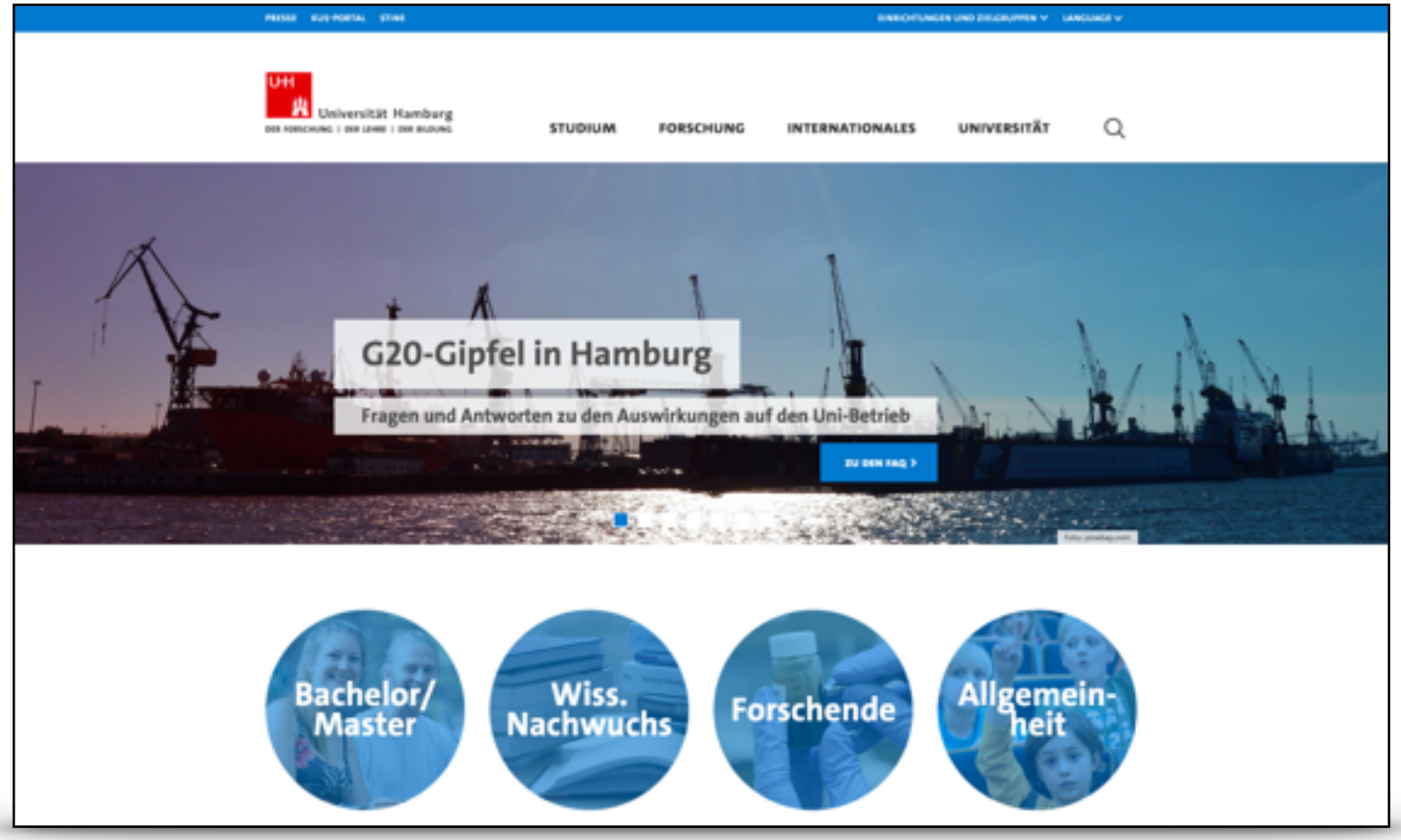
- Einfache Einbindung:

```
<img src="Eiffelturm.jpg"
width="350" alt="Eiffelturm">
```

- `width` und `height` zum Skalieren in Pixeln oder Prozent
- `alt` gibt textuelle Alternativbeschreibung an, falls Bild nicht angezeigt wird

HTML 5

Beispiel: Bilder



HTML 5
Audio

- Eingeführt in HTML5 (vorher Plug-In nötig):

```
<audio controls>
    <source src="horse.wav"
    type="audio/wav">
    <source src="horse.mp3"
    type="audio/mpeg">
Your browser does not support the
audio element.
</audio>
```

HTML 5
Audio

- `controls` fügt Steuerungselemente wie Play, Pause oder einen Lautstärkeregler hinzu
- Bei mehreren `source`-Elementen wählt der Browser das erste erkannte Format
- Text zwischen `<audio>`-Tags wird angezeigt, wenn Browser `audio`-Element nicht unterstützt

HTML 5
Audio

https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_audio_all

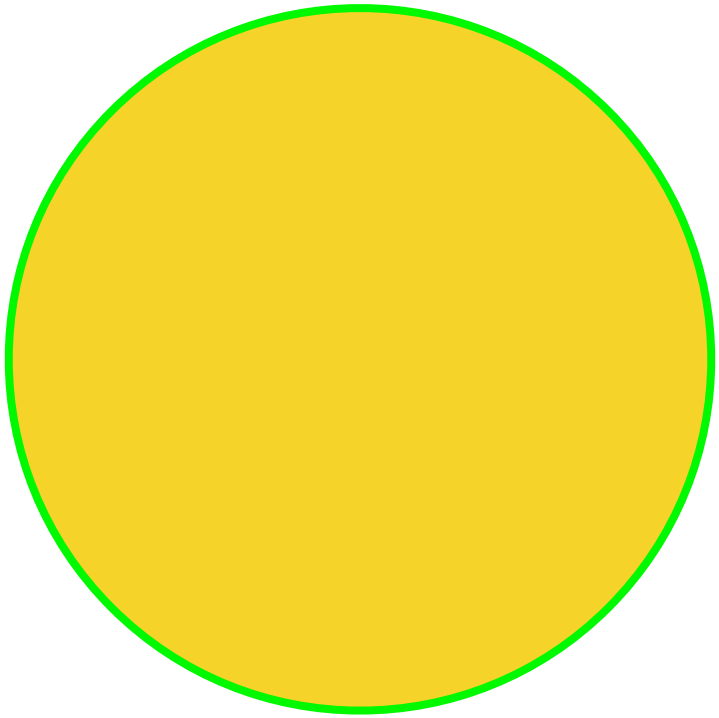
HTML 5
SVG

- Einbindung von Vektorgrafiken im *Scalable Vector Graphics* (SVG) Format über <svg>-Tag
- Beispiel:

```
<svg width=„100" height="100">
 <circle cx="50" cy="50" r="40"
 stroke="green" stroke-width="4"
 fill="yellow" />
</svg>
```

HTML 5

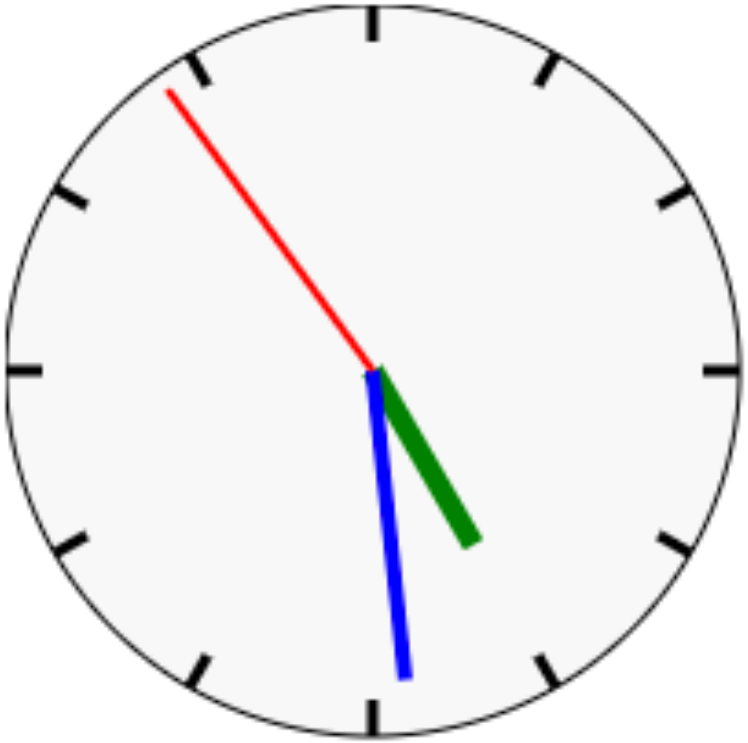
Beispiel: SVG



https://www.w3schools.com/graphics/svg_inhtml.asp

HTML 5

Beispiel: SVG



https://html5demos.com/svg-clock/

HTML 5
Video

- Eingeführt in HTML5 (vorher Plug-In nötig):

```
<video width="320" height="240"
controls>
        <source src="movie.mp4"
        type=„video/mp4">
        <source src="movie.ogg"
        type="video/ogg">
video element is not supported.
</video>
```

HTML 5
Video

- Angabe von `width` und `height` empfohlen, um Browser Größe des Videos mitzuteilen
- `controls` fügt wie bei Audio Steuerungselemente hinzu
- Bei mehreren `source`-Elementen wählt der Browser das erste erkannte Format
- Text zwischen `<video>`-Tags wird angezeigt, wenn Browser `video`-Element nicht unterstützt

HTML 5

Beispiel: Video

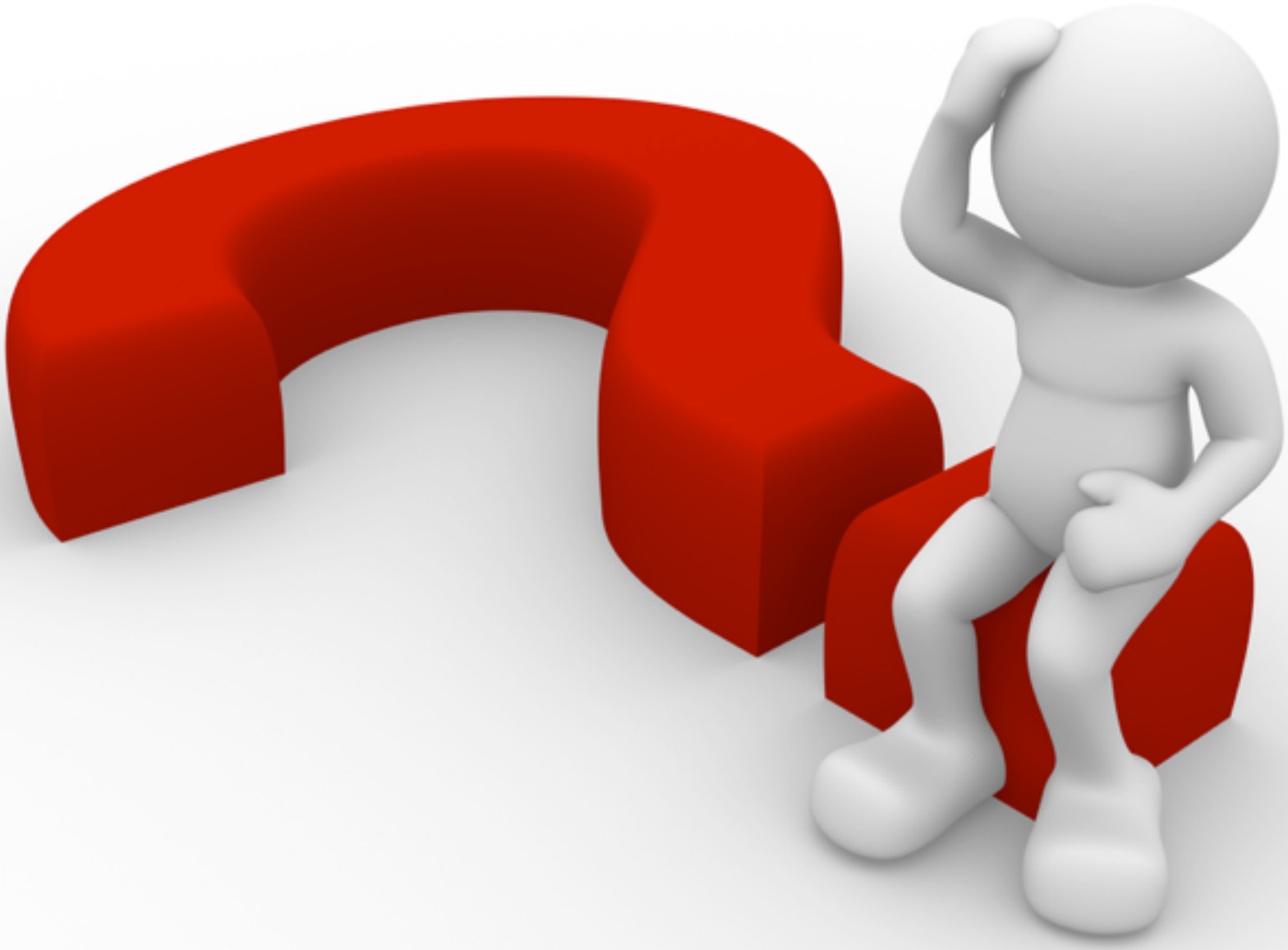
https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_video

HTML 5
Sonstige Medien

- Einbindung anderer Multimedia-Typen, z.B. Animationen, Präsentationen, ...:
  - Durch im Browser integriertes **Plug-In**
  - Über allgemeines **<object>**-Element:

```
<object width="400" height="400"
data=„animation.swf“></object>

<object width="400" height="400"
data="fol.pdf#page=2"></object>
```



Interaktive Medien

Kapitel Mediensysteme & World Wide Web

Dynamik im WWW

Dynamische Webseiten

- **Dynamische Webseiten** werden vor jeder Anzeige *neu* berechnet
- **Statische Webseiten** besitzen festgelegte Inhalte
  - **Dynamische Inhalte** können nur durch interaktive Multimedia-Dokumente umgesetzt werden, z.B. Flash-Dokumente ...

Dynamische Webseiten

- Auswertung zur **dynamischen Anzeige** findet i.d.R. anhand von **Skriptsprachen** statt
- Skript ist Bestandteil der vom Browser aufgerufenen Datei
- Beispiele:
  - Inhalt eines Online-Warenkorbs
  - Verwendung aktueller Uhrzeit
  - …

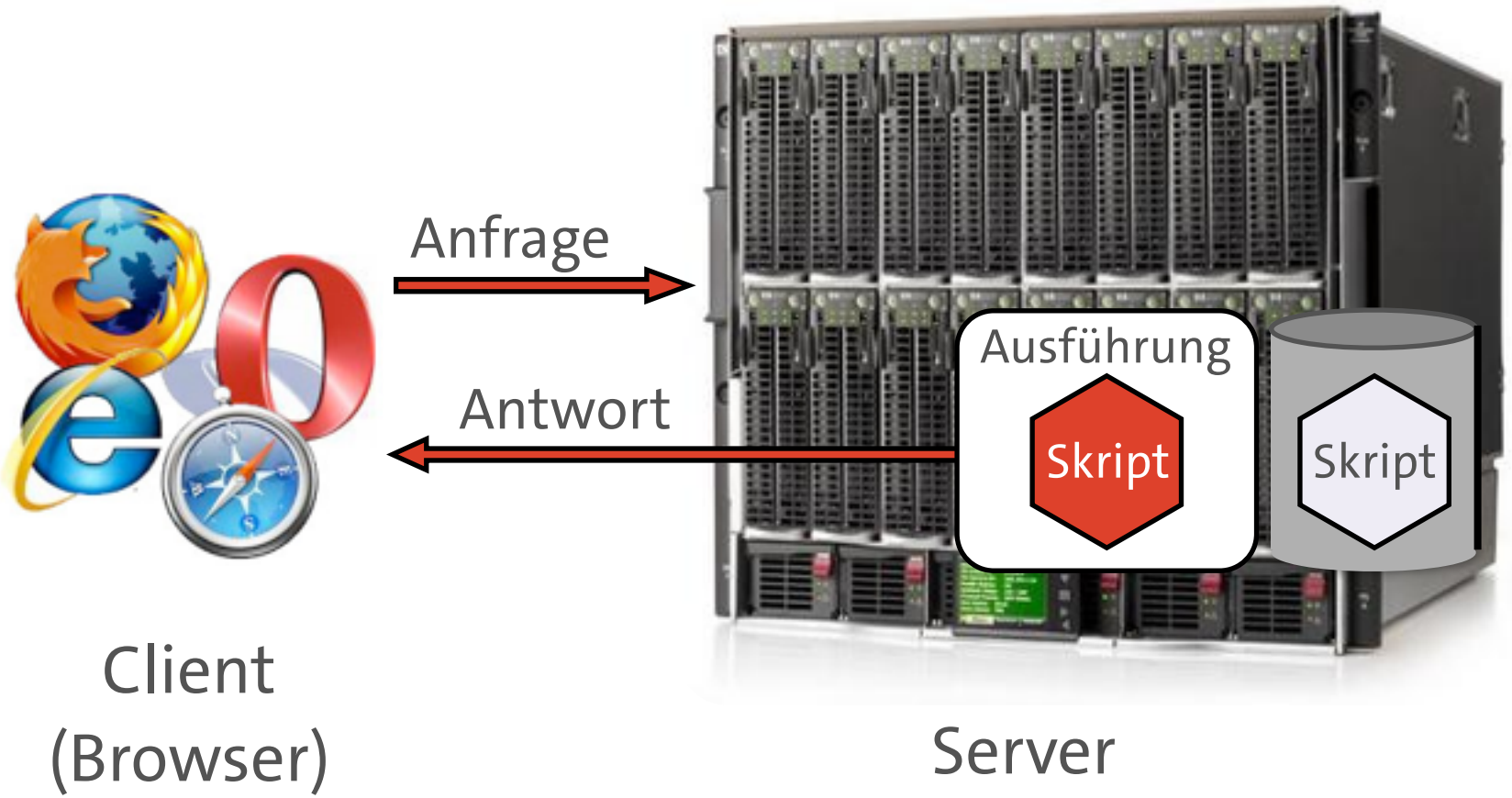
Dynamische Webseiten

Ansätze

1. **Serverseitige Dynamik**: Berechnung findet auf **Server** (**Rechner mit anzuzeigenden Informationen**) statt
2. **Clientseitige Dynamik**: Berechnung findet auf **Rechner des Webnutzers** statt

Serverseitige Dynamik

Ablauf



Serverseitige Dynamik

Ablauf

1. Zugriff auf angeforderte Datei
2. Standardkonformes HTML-Dokument wird direkt beim Server erzeugt
3. HTML-Dokument wird an Client übertragen
4. Anzeige im Browser basierend auf dem erhaltenen HTML-Dokument

Diskussion

Was sind die Vor- und Nachteile von serverseitiger Dynamik?

Serverseitige Dynamik

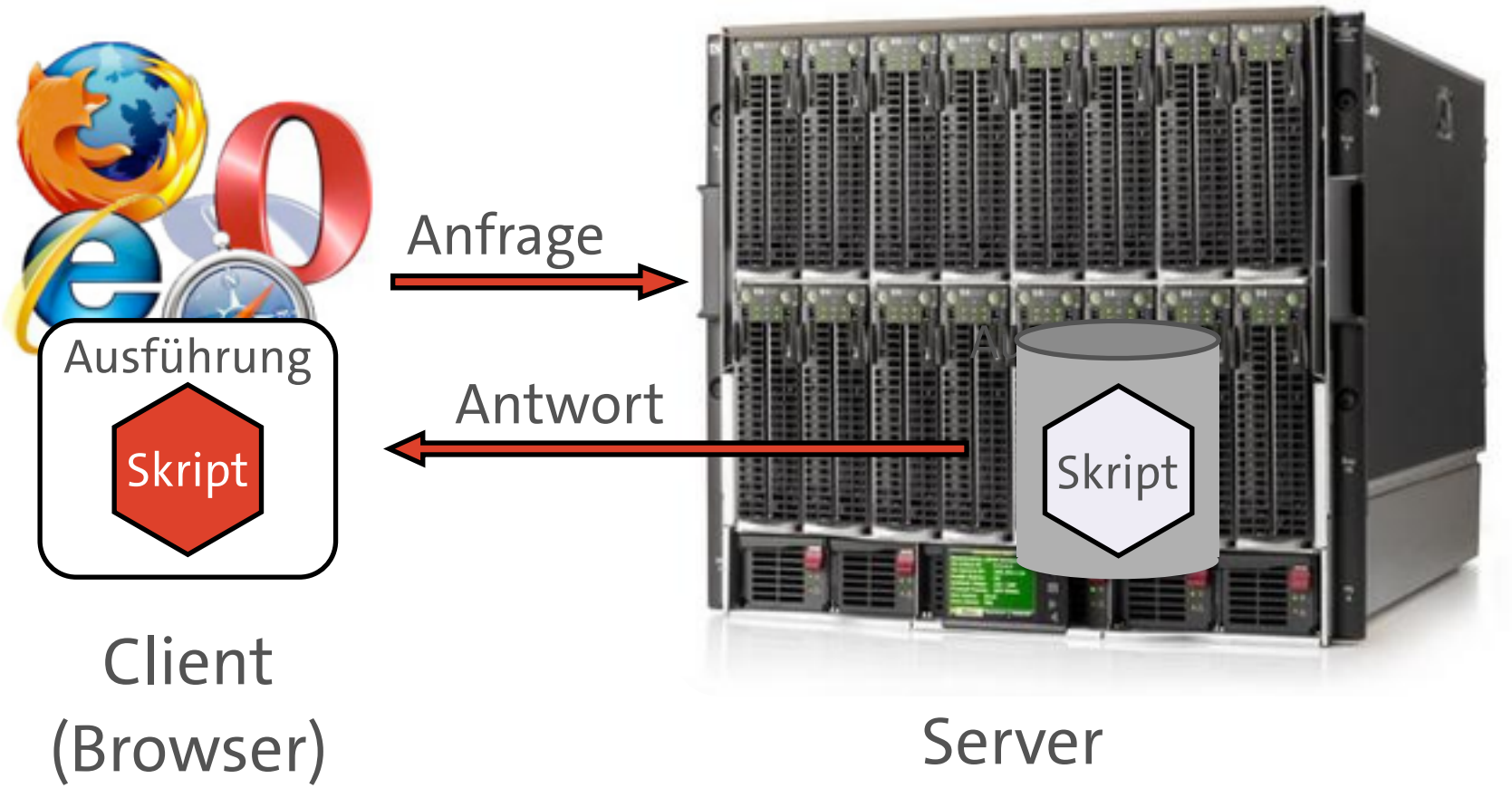
- Vorteil:
  - Konfiguration des clientseitigen Browsers spielt keine Rolle
  - Abstimmung zwischen Serversoftware und Dateien einfacher
  - einzige Möglichkeit für Zugriff auf Datenbestände (z.B. Datenbanken) auf Server
- Nachteil:
  - dynamische Vorgänge hängen von Internetverbindung ab

Serverseitige Dynamik
Sprachen

- Beispiele für typische Sprachen serverseitiger Ausführung:
  - **PHP**
  - ASP
  - ASP.Net
  - Java Server Pages (JSP)
  - …

Clientseitige Dynamik

Ablauf



Clientseitige Dynamik

Ablauf

1. Datei mit Skriptinformationen liegt auf Server
2. Server leitet Datei (mit Skript) an Client
3. Skript wird beim Client ausgeführt
4. Berechnung startet
5. Anzeige wird je nach Berechnung modifiziert

Diskussion

Was sind die Vor- und Nachteile von clientseitiger Dynamik?

Clientseitige Dynamik

- Vorteil:
  - Ausführung des Skripts benötigt keinen Internetzugriff mehr
  - ➡ sehr schnelle Reaktionszeit
- Nachteil:
  - verwendeter Browser (beim Nutzer) muss Skriptsprache unterstützen und zulassen

Clientseitige Dynamik
Sprachen

- Beispiele für typische Sprachen clientseitiger Ausführung:
  - **JavaScript**
  - ECMAScript
  - VBScript

JavaScript

- **JavaScript (JS)** ist Skriptsprache, die hauptsächlich für **dynamisches HTML** in Web-Browsern eingesetzt wird
- Ursprung 1995 bei Firma Netscape
- Inzwischen internationaler Standard für JavaScript: **ISO-16262** bzw. **ECMAScript**

JavaScript

Typische Anwendungsgebiete

- Dynamische Manipulation von Webseiten
- Plausibilitätsprüfung (Datenvalidierung) von Formulareingaben noch vor der Übertragung zum Server
- Sofortiges Vorschlagen von Suchbegriffen
- Banner oder Laufschriften
- …

JavaScript
Einbindung in HTML

- Einbindung über `script`-Element
- Direkte Anbindung an HTML-Elemente als Ereignisbehandlung (***Event-Handler***) möglich

  → mehr dazu in Übungen
- Einbindung externer JavaScript-Datei

JavaScript

Einbindung: Beispiel

```
<html>
   <head>
      <title>Test</title>
      <script>
         alert("Hallo Welt!");
      </script>
   </head>
   <body>
   </body>
</html>
```