

Grundlagen Interaktiver Systeme

MIB und OMB

Entwicklung Interaktiver Anwendungen

MKB

GIS / EIA

HTML Bilder und Verweise

Prof. Dr. Norbert Schnell

mit Materialien von

Prof. Dr. Gabriel Rausch

HTML Bilder und Bildformate

Neben Texten kann ein HTML-Dokument auch *Grafiken* enthalten.

Dabei wird im HTML-Dokument ein Verweis auf die Grafik angegeben:

```

```

Die meisten Browser können die folgenden Formate darstellen:

- **GIF** (*Graphics Interchange Format*) – CompuServe, 1987
- **JPEG** – Joint Photographic Experts Group, 1992
- **PNG** (*Portable Network Graphics*) – W3C working group, 1996
- **WebP** – Google, 2010 (still experimental in Safari)
- **SVG** (*Scalable Vector Graphics*) –

HTML Bilder und Bildformate

| | Komprimierung verlustfrei | Farben | Transparenz | Animation |
|-------|--------------------------------------|----------------------|--------------------|------------------------|
| GIF | ja | 256 (8 bit) | ja - 1 bit | ja |
| JPG | nein | 16,7 Mio (24 bit) | nein | nein |
| PNG8 | ja | 256 (8 bit) | ja - 1 bit | nein |
| PNG24 | ja | 16,7 Mio (24 bit) | ja - 8 bit | apng (Sonderformat) |
| SVG | wird berechnet | 16,7 Mio (24 bit) | ja | ja |
| WebP | beides | 16,7 Mio (24 bit) | ja | nein |

Bildformat **GIF**

Geeignet für **kurze Animationssequenzen**

- + erlaubt Animationen
- nur 256 Farben
- feste Farbpalette kann die Darstellung negativ beeinflussen



<https://medium.com/beginners-guide-to-mobile-web-development/web-image-formats-googles-webp-17e2fe5fc53e>

Bildformat JPEG

Geeignet für **Fotos**

- + hohe Kompression / kleine Dateien
- Kompression erzeugt u.U. sichtbare Bild-Artefakte

Quality 100 | Jpeg compression



Quality 50 | Jpeg compression



Quality 10 | Jpeg compression



Quality 100 | Jpeg compression



Quality 50 | Jpeg compression



Quality 10 | Jpeg compression



Bildformat **PNG**

Geeignet für **komplexe Bildmaterialien, Logos, Ikonen**

- + verlustfrei
- + Unterstützt Transparenz
- große Dateien



Bildformat **SVG**

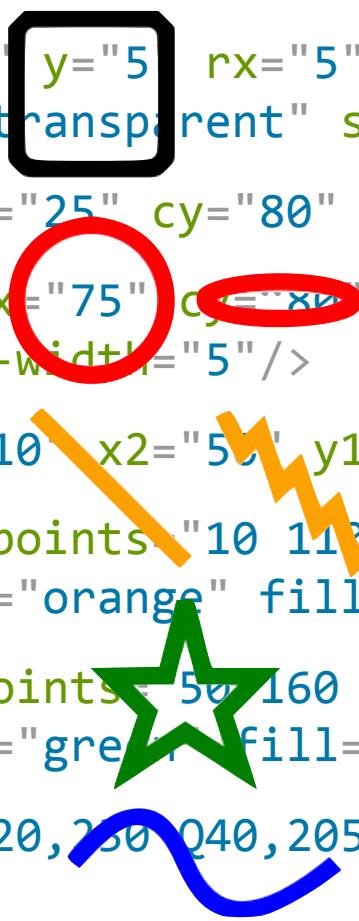
Scalable Vector Graphics (SVG) ist eine XML-basierte Auszeichnungssprache zur Beschreibung zweidimensionaler Vektorgrafiken.

SVG wurde geschaffen, um mit anderen Standards des W3C zusammenzuarbeiten, wie z.B. CSS, DOM und SMIL (*Synchronized Multimedia Integration Language*).

SVG Beispiel

```
<?xml version="1.0" standalone="no"?>

<svg width="200" height="250" version="1.1" xmlns="http://www.w3.org/2000/svg">
  <rect x="5" y="5" rx="5" ry="5" width="40" height="40" stroke="black"
        fill="transparent" stroke-width="5"/>
  <circle cx="25" cy="80" r="20" stroke="red" fill="transparent" stroke-width="5"/>
  <ellipse cx="75" cy="80" rx="20" ry="5" stroke="red" fill="transparent"
            stroke-width="5"/>
  <line x1="10" x2="50" y1="110" y2="150" stroke="orange" stroke-width="5"/>
  <polyline points="10 110 15 120 20 115 25 130 30 125 35 140 40 135 45 150 50 145"
             stroke="orange" fill="transparent" stroke-width="5"/>
  <polygon points="50 160 55 180 70 180 60 190 65 205 50 195 35 205 40 190 30 180 45 180"
             stroke="green" fill="transparent" stroke-width="5"/>
  <path d="M20,230 Q40,205 50,230 T90,230" fill="none" stroke="blue" stroke-width="5"/>
</svg>
```



Grafikmodi **immediate** und **retained**

Immediate mode und *retained mode* bezeichnen zwei verschiedenen Arbeitsweisen von Grafikumgebungen.

Im **retained mode** ist die Grafik durch ein Modell beschrieben, aus dem sie bei Veränderungen (re-)generiert wird und das auch Funktionalitäten wie *Double-Buffering* und *Occlusion-Culling* (Ausschluss verdeckter Bereiche) unterstützen kann. Beispiele für diesen Ansatz sind **HTML/CSS** Grafiken und **SVG**.

Im **immediate mode** gibt es außer den gezeichneten Pixeln keine gespeicherten Daten, die das dargestellte beschreiben. Zum Beispiel funktioniert der **2D-Context** des HTML Grafik-Canvas nach diesem Ansatz.

HTML Verweise

Verweise (engl.: *links*) sind ein entscheidender Bestandteil eines jeden Hypertext-Dokuments. Sie werden mit dem Element `<a>` (ursprünglich *Anker*) angegeben, z.B.:

```
< a href="uebung-1.html">Übung 1</a>  
< a href="uebung-1"></a>
```

Jeder Verweis auf ein anderes Dokument oder Objekt
(z.B. ein Bild in einem ``-Element) kann *intern* oder *extern* sein.

Interne Verweise

Interne Verweise werden mit einem *Pfad* angegeben, wie z.B.:

/images/portrait.jpeg ... absoluter Pfad

../images/portrait.jpeg ... relativer Pfad

portrait.jpeg ... Dateiname (Spezialfall eines relativen Pfads)

Je nach Struktur der verlinkten Dokumente, kann eine geschickte Verwendung von *relativen* und *absoluten* Pfaden es ermöglichen die Verzeichnisstruktur zu ändern, ohne die internen Verweise anpassen zu müssen.

Interne Verweise mit absolutem Pfad

Ein *absoluter* Pfad wird von dem Wurzelverzeichnis des HTTP-Servers definiert.

Beispiel: Ein *absoluter* Pfad in **index.html** (siehe rechts):

```
  
<a href="/html/info.html">Mehr Information</a>
```

Mit einem absolutem Pfad können die Dokumente und Objekte wie **portrait.jpeg** und **info.html** von jedem Dokument aus gleich verlinkt werden.

```
images/  
    portrait.jpeg  
    logo.svg  
    landscape.jpeg  
html/  
    index.html  
    info.html
```

Interne Verweise mit relativem Pfad

Ein *relativer* Pfad eines internen Verweises wird von dem Verzeichnis definiert, in dem die Datei liegt, in dem der Pfad verwendet wird.

Beispiel: Ein relativer Pfad in **index.html** (siehe rechts):

```
  
<a href="info.html">Mehr Information</a>
```

Die beiden Punkte im relativen Link verweisen auf das nächst höherliegende Verzeichnis, in dem sich das Verzeichnis *images* befindet, wo die Bilddatei **portrait.jpeg** zu finden ist.

Da das Dokument **info.html** im gleichen Verzeichnis liegt wie **index.html**, beschränkt sich der Pfad auf den Dateinamen.

```
images/  
    portrait.jpeg  
    logo.svg  
    landscape.jpeg  
html/  
    index.html  
    info.html
```

Interne Verweise innerhalb einer Seite

Interne oder externe Verweise können auch auf Elemente innerhalb einer Seite verweisen:

<https://de.wikipedia.org/wiki/Kuckuck#Fortpflanzung>

Ursprünglich wurden dir Fragmente einer Webpage durch Anker-Elemente markiert:

< a id="Fortpflanzung" >

Fragmente verweisen auf jegliche Elemente innerhalb einer Seite, die eine Eigenschaft **id** oder **name** mit dem Namen des Fragments als Wert haben:

< h2 id="Fortpflanzung" > Fortpflanzung </h2 >
< img name="Fortpflanzung" src="cuckoos-mating.jpeg" />

Externe Verweise

Externe Verweise werden mit einer *URL* angegeben, wie z.B:

<https://en.wikipedia.org/wiki/Cuckoo>

Uniform Ressource Locator (URL)

Eine **URL** identifiziert und lokalisiert eine Ressource, die zu verwendende *Zugriffsmethode* (z. B. das verwendete Netzwerkprotokoll wie *HTTP* oder *FTP*) und den *Ort* der Ressource in Computernetzwerken.

Im allgemeinen Sprachgebrauch auch als *Internetadresse* oder *Webadresse* bezeichnet, wobei dann URLs von Webseiten gemeint sind.

Sie besteht grundsätzlich aus einer, durch Doppelpunkt getrennten, *Schema-Bezeichnung* und einem *Schema-spezifischen* Teil:

<schema>:<schema-specific-part>

z.B. <http://www.hs-furtwangen.de>

URL Schemata

Die Schema-Bezeichnung legt fest, mit welcher technischen Methode die Ressource angesprochen werden soll.

http ... *hypertext transfer protocol*

https ... *HTTP secure*

mailto ... email address

file ... *file of local files ystem*

ftp ... *file transfer protocol*

geo ... geographic coordinates

news ... *news group oder news article*

sip ... *session initiation protocol* (e.g. for voice over IP)

ssh ... *secure shell*

ws/wss ... *web socket connection*

HTTP URL

Beim HTTP-Protokoll gilt für den Schema-spezifischen Teil:

`http://norbert.schnell:pwd123@hs-furwangen.de:33000/~sor/path/data?key=value#fragid`



The URL is annotated with colored brackets below it: 'scheme' (green) covers 'http://'; 'user information' (green) covers 'norbert.schnell:pwd123@'; 'host' (red) covers 'hs-furwangen.de'; 'port' (purple) covers ':33000'; 'path' (orange) covers '~sor/path/data'; 'query' (yellow) covers '?key=value'; and 'fragment' (pink) covers '#fragid'.

host, Server Adresse `https://youtu.be/xSkEvO0ExZo?t=7m19s`

port, erlaubt die Ansteuerung eines TCP/IP-Ports

path, beschreibt eine bestimmte Ressource auf dem Server

query, damit können zusätzliche Informationen zur Anfrage übertragen werden

fragment, referenziert einen Teil der Ressource

HTTP URL Beispiele

Liste der Fakultäten an der HFU

https://www.hs-furtwangen.de/studierende/fakultaeten ← path

Absatz eines Wikipedia Artikels

https://de.wikipedia.org/wiki/Kuckuck#Fortpfanzung ← fragment

YouTube Video-Link mit Startzeit

https://youtu.be/xSkEv00ExZo?t=7m12s ← query

Video auf ARTE

https://api.arte.tv/player/v1/fr/054-19-A/autostart=1&lang=fr ← ???

Serverseitige Verarbeitung von URLs

Insbesondere bei dynamischen Websites hängt die Reaktion auf alles was in einer URL auf die *Hostadresse* (und eventuell einem *Port*) folgt von dem jeweiligen Webserver ab.

Bei statischen Websites verweist der Pfad direkt auf ein auf dem Server bereitstehendes Dokument.

Die statische Verarbeitung von Pfaden, die nicht direkt auf eine Datei verweisen, sondern auf ein Verzeichnis, hängt von der Konfiguration des Servers ab. Die meisten statischen Server sind so konfiguriert, dass Pfade, die auf ein Verzeichnis verweisen durch *index.html* oder *index.php* vervollständigt.

So verweisen z.B. die folgenden URLs auf die gleichen Dokumente:

<https://webuser.hs-furtwangen.de/~sor/>

<https://webuser.hs-furtwangen.de/~sor/index.html>

Im Bündel

- Web Bildformate
- *immediate* und *retained* Grafikmodi
- *interne* und *externe* Verweise
- URL Syntax und *Schemata*

