HTML Multimedia

Stand: 04.07.2022

Multimedia mit HTML
Objekt zum umgebenden Text

Da das `object`-Element ein **Inline-Element** ist, können Objekte mitten in einem Text platziert werden. Wenn das Objekt höher ist als die Zeilenhöhe, dann muss der Text der gleichen Zeile vertikal zum Objekt ausgerichtet
werden. Wenn Sie nichts anderes angeben, wird der Text untenbündig zum Objekt ausgerichtet. Sie können jedoch mit Hilfe eines Attributs selbst bestimmen, wie der Text zum Objekt ausgerichtet werden soll. Dieses Attribut ist allerdings als
deprecated gekennzeichnet und soll künftig aus dem HTML-Standard entfallen. Die gleiche Wirkung lässt sich nämlich auch mit Stylesheets erzielen. Die Angaben sind jedoch nur von Bedeutung, wenn die Anzeige mit Hilfe eines Plug-Ins erfolgt
und direkt im Anzeigefenster des WWW-Browsers möglich ist.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>HIER SPIELT <object data="haste.mid" type="audio/x-midi" width="200"
hight="100" align="middle">
Ihr Browser kann das Objekt leider nicht anzeigen!
</object>DIE MUSIK</h1>
</body>
</html>
```

Info Mit dem Attribut `align=` können Sie umgebenden Text vertikal zum Objekt ausrichten (align = Ausrichtung).

- Mit `align="top"` wird der Text obenbündig zum eingebundenen Objekt ausgerichtet (top = oben).
- Mit `align="middle"` wird der Text mittig zum eingebundenen Objekt ausgerichtet (middle = mittig).
- Mit `align="bottom"` wird der Text untenbündig zum eingebundenen Objekt ausgerichtet (bottom = unten).

Um Objekte und Texte sauber zueinander auszurichten, können Sie auch mit rahmenlosen („blinden“) Tabellen arbeiten.

- Mit `align="texttop"` richten Sie den Text obenbündig zum Objekt aus, und zwar an der Oberkante des kleinsten Textes (texttop = oben am Text).
- Mit `align="absmiddle"` richten Sie den Text absolut mittig zum Objekt aus, auch bei unterschiedlichen Textgrößen (absmiddle = absolute middle = in jedem Fall mittig).
- Mit `align="absbottom"` richten Sie den Text untenbündig zum Objekt aus, und zwar an der Unterkante des kleinsten Textes (absbottom = absolute bottom = in jedem Fall untenbündig).
- Mit `align="baseline"` richten Sie den Text genauso aus wie mit der Standardangabe `align="bottom"`. Benutzen Sie deshalb besser die Standardangabe.

Text um ein Objekt fließen lassen

Objekte können Sie links oder rechts ausrichten. Der umgebende Text fließt dabei um das Objekt. Mit zwei weiteren Attributen können Sie Abstand zum umfließenden Text erzeugen, damit der Text nicht direkt am Objekt „klebt“. Den automatischen Textumfluss
können Sie auch abbrechen und die Textfortsetzung unterhalb des Objekts erzwingen. Alle dazu notwendigen Attribute sind allerdings als deprecated gekennzeichnet und sollen künftig aus dem HTML-Standard entfallen. Die gleiche Wirkung lässt sich
nämlich auch mit **Stylesheets** erzielen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body bgcolor="#333333" text="#FFFFFF">
<h1><object classid="CLSID:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=4,0,0,0"
width="200" height="200" align="left" hspace="10" vspace="7">
<param name="movie" value="neucler.swf">
</object>Animation!</h1>
<p>Kann schön sein, wenn sie nicht zu aufdringlich ist.
Dies ist ein Flash-Filmchen von gerade mal einem Kilobyte Größe.
<br>
clear="all"></p>
<p>Der Text geht unterhalb weiter.</p>
</body>
</html>
```

Info Mit den Angaben `align="left"` bzw. `align="right"` veranlassen sie den Browser, das objekt linksbündig bzw. rechtsbündig auszurichten und den darauffolgenden Fließtext rechts bzw. links neben dem Objekt anzuzeigen
und um das Objekt herumfließen zu lassen (align = Ausrichtung, left = links, right = rechts). Der umgebende Text kann verschiedene Elemente enthalten, zum Beispiel auch überschriften oder Listen. Um

Abstand zwischen Objekt und umfließendem
Text zu erzeugen, stehen die Attribute `hspace=` und `vspace=` zur Verfügung.

- Mit `hspace=` [Pixel] bestimmen Sie den Abstand zwischen Objekt und anderen Elementen links bzw. rechts davon (hspace = horizontal space = horizontaler Abstand). Mit `vspace=` [Pixel] bestimmen Sie den Abstand
  zwischen Objekt und anderen Elementen darüber bzw. darunter (vspace = vertical space = vertikaler Abstand).
- Das Attribut `hspace=` betrifft immer den linken und den rechten Randabstand vom Objekt, `vspace=` immer den oberen und den unteren Randabstand. Wenn Sie also beispielsweise bei einem links ausgerichteten Objekt `hspace=` definieren, wird nicht nur rechts vom Objekt zum nebenstehenden Text ein Abstand erzeugt, sondern auch links zum Rand des Anzeigefensters hin. Um dies zu verhindern, können Sie exakter arbeiten, indem Sie Stylesheets verwenden.
- Sie können beide Attribute notieren oder auch nur eines davon.

Um den automatischen Textumfluss zu unterbrechen und zu erzeugen, dass alles was folgt unterhalb des Objekts angezeigt wird, können Sie einen Zeilenumbruch mit dem Attribut `clear=` notieren.

- Mit `<br clear="all">` erreichen Sie, dass der Textfluss ab der nächsten Zeile in jedem Fall unterhalb des Objekts fortgesetzt wird.
- Mit `<br clear="left">` erreichen Sie, dass der Textfluss ab der nächsten Zeile in jedem Fall eines linksbündig ausgerichteten Objekts fortgesetzt wird.
- Mit `<br clear="right">` erreichen Sie, dass der Textfluss ab der nächsten Zeile unterhalb eines rechtsbündig ausgerichteten Objekts fortgesetzt wird.

Flash-Einbindung

Keine andere Technologie hat das Web in den letzten Jahren optisch nachhaltiger geprägt als der Siegeszug von **Flash**. Egal ob für Online-Spiele, animierte Illustrationen, schicke Site-Navigation oder Startseiten-Trailer eingesetzt:
Flash hat in all diesen und anderen Bereichen konkurrierende Techniken wie Java-Applets oder auch DHTML verdrängt. Grund genug, die standardkonforme Einbettung eines Flash-Movies in HTML zu behandeln.

Flash-Dateien werden in HTML über das `object`-Element eingebunden.

```
<object type="application/x-shockwave-flash" data="/flash/intro.swf"
style="width:740px; height:380px">
<param name="movie" value="/flash/intro.swf">
<param name="quality" value="high">
Sie benötigen einen Flash-Player als Browser-Plugin.
Ein solches Plugin bekommen Sie auf
http://www.macromedia.com/software/flashplayer/
</object>
```

Info Möglicherweise sind Ihnen bereits wesentlich komplexere Quelltextkonstrukte zum Einbetten von Flash begegnet. Diese greifen beispielsweise zusätzlich auf das `embed`-Element zurück, ein nicht zum HTML-Standard gehörendes
Element, das speziell Netscape 4.x bedienen soll. Das ganze HTML-Dokument wird dadurch syntaktisch `invalid`. Oder Sie haben im einleitenden `<object>`-Tag Attribute wie `classid=` und `codebase=` mit zum Teil merkwürdigen Wertzuweisungen gesehen.

Das oben vorgestellte, standardkonforme Quelltextkonstrukt

ist jedoch in heutiger Zeit völlig ausreichend und funktioniert mit allen verbreiteten Browsern. Im einleitenden `<object>`-Tag sind dabei zwei Attribute wichtig: `type=` muss notiert werden, um den Browser über eine
Mime-Type-Angabe darüber zu informieren, welche Art von Daten das Objekt enthält. Der Mime-Type für Flash lautet `application/x-shockwave-flash`. Das Attribut `data=` ist ebenfalls wichtig und muss notiert werden, obwohl
die Dateiquelle des Flash-Movies in einem `<param>`-Tag innerhalb des `object`-Elements nochmals referenziert wird. Anzugeben ist ein URI, also eine vollständige Internetadresse, oder eine lokale Referenz auf die *swf*-Datei
oder die Datenquelle, welche den *swf*-Code sendet. Ähnlich wie beim `img`-Element sind beim `object`-Element die **HTML-Attribute** `width=` und `height=` erlaubt, um dem Browser
die Pixelausmaße des einzubettenden Objekts vorab mitzuteilen. Diese können aber wie im Beispielcode gezeigt ebenso in CSS-Form angegeben werden. Als Elementinhalt von `<object>` ... `</object>` können unter anderem
alleinstehende `<param>`-Tags notiert werden (bei XHTML beachten: `<param .../>`). Diese haben die obligatorischen Attribute `name=` und `value=`. Sie definieren Parameter, die
an das aufgerufene Objekt übergeben werden. Bei Flash-Movies sollten auf jeden Fall die beiden im Beispiel gezeigten Parameter (`name="movie"` und `name="quality"`) notiert werden. Beim `movie`-Parameter sollte
als Wert nochmals der URI des Flash-Movies angegeben werden. Beim `quality`-Parameter empfiehlt sich die Standard-Wertzuweisung `high`. Das `object`-Element kann als Elementinhalt außerdem Text und andere Block-
oder Inline-Elemente enthalten. Diese werden angezeigt, falls das Objekt nicht angezeigt werden kann. Im obigen Beispiel haben wir etwas Text notiert, der einen Anwender, bei dem das Flash-Movie nicht angezeigt werden kann, über die Möglichkeit
aufklärt, wie er ein Plug-In erwerben kann.

Ohne Flash-Plug-In können Flash-Movies nicht im Browser angezeigt werden.

Die meisten Browser verfügen heute jedoch in der Standardinstallation bereits über ein entsprechendes Plug-In. Zum Erstellen von Flash-Movies benötigen Sie eine Flash-Autorensoftware. Das Produkt, welches Flash-Erfinder Macromedia selbst anbietet, ist

bestens geeignet, aber auch nicht unbedingt preiswert. Es gibt jedoch preiswerte oder kostenlose Drittanbieter-Alternativen.

Flash ohne HTML

Da Flash vektororientiert ist und sich damit problemlos skalieren lässt, kann es beispielsweise für Produktpräsentationen, Spiele und Ähnliches auch sinnvoll sein, das Flash-Movie selbst anstelle eines HTML-Dokuments mit eingebettetem Flash-Movie aufzurufen.
In einer vorgeschalteten Seite verlinken Sie dazu einfach die *swf*-Datei anstelle eines entsprechenden HTML-Dokuments, beispielsweise so:

```
<a href="produkt.swf" type="application/x-shockwave-
flash">Produktpräsentation</a>
(Flash erforderlich!)
```

Info Rückverweise zur übrigen Site sind innerhalb von Flash möglich.

Java-Applets einbinden

Java-Applets sind mittlerweile seltener im Web als vor dem Aufkommen von Flash. Dafür werden **Java-Applets** häufiger für seriösere und anspruchsvollere Zwecke genutzt, wie etwa Online-Banking, Online-Brokering und Online-Learning. HTML
stellte zunächst ein separates `applet`-Element zum Einbinden von Java-Applets zur Verfügung. Heute gehört dieses Element noch zur Transitional-Variante von HTML 4.01. In der Strict-Variante ist es nicht mehr vorhanden und auch nicht
mehr nötig. Das universeller konzipierte `object`-Element bietet stattdessen alles, was zum Einbinden von Java-Applets erforderlich ist. Da auch alle modernen Browser mit dieser Methode zurechtkommen, darf das `applet`-Element
mittlerweile mit gutem Gewissen der Vergessenheit anheim fallen.

```
<object classid="java:Beispiel.class" codetype="application/java-vm"
style="width:600px; height:400px">
Java erforderlich! Die Java-Runtime-Umgebung können Sie von
http://java.sun.com/ downloaden.
</object>
```

Das `object`-Element kennen Sie bereits vom Abschnitt über die Einbindung von Flash. Im Fall von Java-Applets müssen im einleitenden `<object>`-Tag jedoch andere Attribute notiert werden. Erforderlich sind die Attribute
`classid=` und `codetype=`. Beim `classid`-Attribut weisen Sie als Wert die Zeichenfolge `java:`, gefolgt vom Namen der `class`-Datei des Java-Applets zu. Dabei darf nur der reine Dateiname angegeben werden, keine Internetadresse oder Adressierung eines anderen Verzeichnisses. Sollte sich das **Applet**

nicht im gleichen Verzeichnis befinden wie das HTML-Dokument oder auf einem ganz anderen Hostrechner liegen, benötigen sie zusätzlich
das Attribut `codebase=`. Diesem Attribut weisen Sie als Wert den URI des Verzeichnisses zu, in dem die `class`-Datei liegt, also beispielsweise `codebase="../classes/"`. Beim `codetype`-Attribut geben Sie den Mime-Type für den Objektcode des Java-Applets an. Dieser Mime-Type lautet standardmäßig `application/java-vm`. Parameter, die dem Java-Applet beim Start übergeben werden sollen, können über `<param>`-Tags notiert werden.
Einem Applet, das z.B. einen Wert zu einem Parameter namens `init_value` erwartet, kann etwa Folgendes mitgeteilt werden:

```
<object...>
<param name="init_value" value="3">
</object>
```

Multimedia einbinden

Unter **Multimedia** ist hier zu verstehen: Datendateien, die in Fremdformaten vorliegen und Ausgaben erzeugen, die über Text und www-gerechte Grafik hinausreichen. Also zum Beispiel Sound, Video, Animation, Tabellen, Präsentations,- Vektor-
und CAD-Grafik, 3D-Grafik, VRML-Dateien usw.

Ein WWW-Browser kann solche Dateien anzeigen, wenn der Anwender ein entsprechendes Plugin installiert hat. Wenn das Plugin installiert ist, kann der WWW-Browser die Datei so in seinem Anzeigefenster präsentieren, wie sie von dem Ursprungsprogramm erstellt
wurde. Bei Abspielvorgängen, etwa von Videos oder Sound, wird ein entsprechender Player angezeigt – je nachdem, wie das Plugin beschaffen ist. Wenn dem Browser eine Verknüpfung zwischen der Dateiendung und einem Fremdprogramm bekannt ist, kann er das
Fremdprogramm mit der betreffenden Datei starten.

```
<embed src="tabelle.xls">
<embed src="../video.avi">
<embed src="multimedia/virtuelleWelt.wrl">
<embed src="http://www.netzwelt.com/yrag.mid">
<embed src="file://localhost/c:/windows/progman.hlp">
```

Mit `<embed src=>` definieren Sie Plugin-Referenz (embed = einbetten, src = source = Quelle). Hinter dem Istgleichzeichen notieren Sie die gewünschte Datei. Die Angabe muß in Anführungszeichen stehen. Sie können auch Dateien
in anderen Verzeichnissen und weltweit auf anderen Internet-Servern einbinden. Dabei gelten die gleichen Regeln wie beim Einbinden von Grafiken.

> Ob und wie die Datei letztendlich beim Anwender „angezeigt“ wird,

hängt von vielen Faktoren ab, zum Beispiel von der Art der Datei, von den Hardware-Voraussetzungen des Anwenders, von den installierten Plugins und von vorhandenen, dem Browser bekannten Fremdprogrammen. Binden Sie Multimedia-Dateien daher niemals kommentarlos
in Ihre HTML-Dateien ein, sondern weisen Sie den Anwender darauf hin, um welche Art von Daten es sich handelt, und unter welchen Voraussetzungen eine korrekte Anzeige möglich ist. Viele **Multimedia-Dateien**, vor allem Videos und 3D-Grafiken,
sind vom Umfang her ungeeignet für die heutige üblichen Übertragungsraten im WWW. Wenn Sie größere Dateien einbetten, weisen Sie den Anwender im umgebenden Text auf die Größe der Datei hin. Bei Dateitypen, zu denen der Browser ein Plugin kennt, fragt
er den Anwender, falls dieser das Plugin nicht hat, ob er das Plugin haben möchte, und ruft auf Wunsch die Übersichtsseiten auf dem WWW-Server auf, von wo aus der Anwender sich das Plugin downloaden kann.

> Wenn der WWW-Browser die Datei nicht verarbeiten kann,

sollte er eine Fehlermeldung ausgeben und anstelle der Datei ein Symbol für nicht darstellbare Daten anzeigen. Wenn WWW-Browser auf einen Dateityp stößt, den er nicht unmittelbar verarbeiten kann, zu dem er aber eine Verknüpfung mit einem Fremdprogramm
kennt, wird im Anzeigefenster an der Stelle, wo die Datei angezeigt werden soll, zunächst ein Icon des Fremdprogramms dargestellt, das beim Doppelklick das Programm mit der eingebundenen Datei startet. Sie können auch Verweise auf beliebige Dateien setzen.
Der Unterschied ist, daß die Datei bei einem Verweis erst nach dem Anklicken des Verweises separat aufgerufen wird, während sie beim Einbinden der Multimedia-Datei so gut wie möglich direkt in die Bildschirmanzeige einer HTML-Datei integriert wird.

Eingebettete Objekte aller Art

In den Abschnitten über Flash, Java und SVG haben wir gesehen, dass die Einbindung der Ressourcen stets über das universelle `object`-Element in **HTML** erfolgt.

Dieses Element ist so universell, dass sich damit jede beliebige Ressource einbinden lässt. So ist es auch schon absehbar, dass das `img`-Element in neueren XHTML-Versionen verschwindet. Grafiken werden dann ebenfalls über das `object`-Element
referenziert. Das sieht beispielsweise so aus:

```
<object type="image/png" data="bild.png" style="width:354px; height:314px">
</object>
```

Auch eine nackte Textdatei lässt sich über dieses Element einbinden.

```
<object type="text/plain" data="e-mail-thomas-003.txt" style="width:80%;
```

```
height:300px; overflow:scroll">
</object>
```

Info Wichtig sind im Zweifelsfall stets die Attribute `data=` zum Referenzieren der Quelle und `type=` zur Angabe des Mime-Type. Mit dem `style`-Attribut lässt sich der Anzeigebereich anpassen. Im Beispiel der referenzierten
Textdatei haben wir eine Anzeigegröße vorgegeben und mit `overflow:scroll` sichergestellt, dass der Text, sofern er mehr Platz einnimmt, vom Anwender gescrollt werden kann. Typische, über das `object`-Element eingebundene Objekte
sind auch Sounds und Video. In der Regel ist dazu jedoch eine Angabe zum `classid`-Attribut erforderlich, die ein bestimmtes Plug-In oder Abspielprogramm im Browser erzwingt, sowie die Angabe von `<param>`–Tags für Initialwerte.
Für Standard-Software gibt es `classid`-Codes, die dem `classid`-Attribut zugewiesen werden können. Das folgende Beispiel bindet den Windows Media Player ein und übergibt ihm eine MP3-Datei zum Abspielen:

```
<object id="MediaPlayer" classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6"
type="application/x-oleobject"
style="width:300px; height:200px">
<param name="src" value="beispiel.mp3">
<param name="autoplay" value="false">
<param name="controller" value="true">
</object>
```

Info Das Beispiel funktioniert in dieser Form jedoch nur im MS Internet Explorer, da die Einbettung über ein ActiveX-Control realisiert wird, eine bei anderen Browsern verpönte, weil für Sicherheitslücken anfällige Technologie. Zahlreiche andere
Formate haben ihre eigene **Syntax** zum Einbetten. Dem Wildwuchs ist hier leider noch Tür und Tor geöffnet, da vernünftige Standards fehlen. Anbieter spezieller Techniken wie RealMedia (Audio- und Video-Streaming) und andere informieren
auf ihren Seiten darüber, wie Sie solche Formate in HTML einbetten. In einigen Fällen sind zusätzlich Einstellungen oder Erweiterungen beim Webserver erforderlich.

SVG-Vektorgrafiken

Leider unterstützen selbst nach modernsten Kriterien entworfene Browser wie Firefox bis heute von Haus aus ausschließlich Pixelgrafikformate. Dabei ist **Vektorgrafik** für Anwendungsfälle wie Illustrationen, technische Zeichnungen, Logos,
kunstvolle Schriftzüge usw. eindeutig besser geeignet als Pixelgrafik. In Verbindung mit Scriptsprachen ist auch die Darstellung von Animation und bewegten Abläufen möglich.

SVG (Scalable Vektor Graphics) ist ein in den letzten Jahren eigens für den Einsatz im Web entwickeltes Vektorgrafikformat. Die vektorgrafischen Informationen werden in XML-Form gespeichert. SVG ist also ein

XML-Derivat. Es wird ebenso wie HTML und CSS
vom W3-Konsortium als offen dokumentierter Standard spezifiziert und weiterentwickelt. Führende Vektorgrafikprogramme wie CorelDRAW oder Adobe Illustrator sind längst in der Lage, Daten als SVG-Grafik zu exportieren. Auch wenn SVG-Unterstützung der Browser
noch lausig ist, so ist doch zu erwarten, dass sich die Situation noch verbessern wird, was sicherlich zu einer weiteren Verbreitung von SVG beitragen wird. In **HTML** können Sie eine SVG-Datei wie auch Flash und Java über das `object`-Element
einbinden.

```
<object type="image/svg+xml" data="beispiel.svg" style="width:350px;
height:176px">
SVG-Viewer erforderlich! Auf
http://www.w3.org/Graphics/SVG/SVG-Implementations
erhalten Sie Informationen über SVG-Implementierungen.
</object>
```

Die SVG-Quelle wird im `data`-Attribut als URI referenziert. Das `type`-Attribut sollte unbedingt notiert werden und den Mime-Type `image/svg+xml` als Wert zugewiesen bekommen.
Info Wenn Sie XHTML statt HTML verwenden, sind Sie bei SVG übrigens nicht auf die Referenzierung angewiesen. In diesem Fall können Sie über die Modularisierungsphilosophie von XML den SVG-Code direkt in den Quelltext integrieren. Dazu wird das
einleitende `<html>`-Tag wie folgt erweitert:

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:svg="http://www.w3.org/2000/svg">
```

Es wird also zusätzlich zum Namensraum von XHTML derjenige von SVG deklariert. Der XML-Code des SVG-Dokuments kann nun an der gewünschten Stelle im Dokument wie im folgenden Minibeispiel integriert werden:

```
<svg:svg height="100px" width="100px">
<svg:rect x="0" y="0" width="100" height="100" fill="red" />
</svg:svg>
```

Bei jedem SVG-Tag muss dazu das Präfix `svg:` notiert werden, da HTML als Default-Namensraum definiert wurde. In der Browser-Praxis ist dies jedoch noch weitgehend Zukunftsmusik, da für eine korrekte Interpretation der XML-Parser des Browsers
benötigt wird.