

# bgsound Objekt des Internet Explorer

**bgsound Objekt des Internet Explorer .....1**  
**Erzeugung ..... 1**  
**Zugriff ..... 2**  
**Alternative zu BGSOUND..... 6**  
**Eigenschaften..... 7**  
**Methoden ..... 9**

Hintergrundmusik (wave, midi, mp3 etc. ) oder Videoclip-Sound  
per HTML-Element ab IE 3.x  
per Script ohne ID-Verwendung ab IE 4.x  
mit ID-Verwendung ab IE 5.x

Sound ist solange aktiv laut Anzahl der Wiederholungen der Sounderzeugung,  
bis zum Entladen des des Dokumentes  
bis zur Stummschaltung (siehe unten)

Alternativen zum bgsound Objekt sind: .style.time2 Behavior  
Windows Media Player 7.1

Hinweis: Wenn ein Fenster mit dem Unterstrichsymbol rechts oben in der Fensterecke minimiert wurde, dann ist es durch Windows in die Hintergrund-Prioritätenfolge eingeordnet worden und arbeitet im Hintergrund bzw. garnicht. Mit anschließendem Maximieren (Symbol in der Fensterleiste rechts oben) wird das Fenster wieder angezeigt und das geladene Dokument **erneut** geöffnet. Konsequenz daraus ist, dass dieses Maximieren einem Neustart des Dokumentes entspricht, auch wenn der Programmierer diesen Zustand nicht erwünscht. Sound, der z.B. mit Start des Dokumentes erzeugt wird, erklingt erneut mit Maximierung nach einer Fensterminimierung. Analogon ist die Fenstereinengung im Browser durch z.B. Ein- und Ausblenden der Favoritenleiste. Dieses Verhalten des Fensters mit seinem Dokument ist aber **nicht identisch** mit dem Verhalten nach einer Fenstergrößenänderung z.B. durch Rahmenverschiebung (resize). Die Aktionen des Fensters und seines Dokumentes bezüglich Resize müssen programmiert werden, sind also nicht standardmäßig vorhanden - im Gegensatz zum Verhalten mit/nach Fensterminimierung/-maximierung per Symbole in der rechten oberen Ecke der Fensterleiste. Sollte ein Frameset minimiert werden, dann wird das für den gesamten Frameset getan (Frameset hat 1 Fenster für alle Frames gesamt), allerdings mit Maximierung wird auch das Frameset-Dokument neu geladen. Wichtig dabei sind für Zeiger-Bezüge der Frames auf den Frameset per parent-Zeiger, dass die Daten im Frameset-Dokument mit Maximierung initialisiert werden und somit ebenfalls Daten der Frames, die im Frameset-Dokument abgelegt wurden, also z.B. Daten, die Verhaltensweisen der Frames vor einer Änderung der Frames-Inhalte gespeichert haben.

## Erzeugung

```
.....
<BGSOUND
    ID="freies_id"
    SRC="url_oder_dateiname"
    LOOP="infinite" oder Anzahl der Wiederholungen ab 1
    BALANCE =Wert
    VOLUME=Wert
>
.....
```

Hinweise: Es sind **mehrere** BGSOUND-Objekte (mit verschiedenen ID's) in einem Dokument instanzierbar, die auch **parallel** arbeiten. Man beachte dabei unbedingt die **intensive** Timer-Ressourcennutzung der BGSOUND-Objekte. Es ist dringend anzuraten, das Objekt, dessen Eigenschaft .loop auf unendlich belegt wurde, per Urlzuweisung an die Eigenschaft .src stumm zuschalten, sobald der Sound nicht mehr benötigt wird: Es wird eine leere Url (Leerkette) zugewiesen. Eine Anwendung der Stummschaltung ist auch in der Realisierung eine Playliste aus Sounds anhand des BGSOUND-Objektes möglich (Folge von Urlzuweisungen, deren zeitlicher Abstand die jeweilige bekannte Sounddauer ist (per Anweisung window.setTimeout()) lässt sich die Wiedergabezeit der Sounds verwalten)). Die Nutzung von Playlisten-Alternativen zum BGSOUND-Objekt ist nur dann nötig, wenn Steuerungselemente zur Media-Datei oder zur Lautstärkeinstellung, die über den Rahmen der windoweigenen gehen, genutzt werden sollen.

Jede Urlzuweisung an die Eigenschaft .src bewirkt den automatischen Neustart der Medium-Wiedergabe per BGSOUND-Objekt mit den aktuellen Einstellungen der Objekt-Eigenschaften sowie den Einstellungen innerhalb der Media-Datei. Wurde .src mit einer Leerkette belegt, so wird kein Sound wiedergegeben (Stummschaltung ohne Veränderung der windoweigenen Lautstärkeregelung).

Für **jeden** Medientyp ist ein **eigenständiges** BGSOUND-Objekt zu kodieren. Werden verschiedene Medientypen wie z.B. Midi und mp3 per gemeinsamen BGSOUND-Objekt verwaltet, so **kann** es passieren, dass die aktuelle Reglereinstellung zum Medientyp (z.B. mp3/wave) der windoweigenen Lautstärkeinstellung für die Soundwiedergabe nicht korrekt so belassen wird, wie sie der User dort permanent eingestellt hat, sondern für die Wiedergabe des Medientyps unerwünscht verändert wird, egal ob die Eigenschaft .volume des BGSOUND-Objektes durch den Programmierer verändert wurde oder nicht.



Das Vorladen eines Sounds per Script geht nur per BGSOUND selbst, da es keine Funktion analog zu new Image() gibt. BGSOUND muss mit VOLUME auf -10 erzeugt werden (also stumm), wobei die Zuweisung zu .src das Laden in den Browser-Cache bewirkt,

## Zugriff

z.T. erst ab IE 5.x

```
<BGSOUND ID="freies_id" .....>
```

in Script dann **freies\_id** verwenden per **freies\_id.eigenschaft** etc.

Beispiele für LOOP-Varianten:

```
<BGSOUND SRC="file:///c:/test/wav/test.wav">          genau 1 mal
<BGSOUND SRC="file:///c:/test/wav/test.wav" LOOP>   genau 1 mal
<BGSOUND SCR="file:///c:/test/wav/test.wav" LOOP=>  genau 1 mal
<BGSOUND SCR="file:///c:/test/wav/test.wav" LOOP=0> genau 1 mal
<BGSOUND SCR="file:///c:/test/wav/test.wav" LOOP=1> genau 1 mal
<BGSOUND SCR="file:///c:/test/wav/test.wav" LOOP=10> genau 10 mal
<BGSOUND SCR="file:///c:/test/wav/test.wav" LOOP=-1> endlos
```

```
<HEAD>
<SCRIPT>
    function KanalVerteilung(Wert)
    { ID_bgsound.balance = Wert;}

    function GenauEinmal()
    {
        ID_bgsound.loop = 1;
        ID_bgsound.src = ID_bgsound.src; // restart
    }

    function Endlos()
    {
        ID_bgsound.loop = -1;
        ID_bgsound.src = ID_bgsound.src; // restart
    }
</SCRIPT>
<BGSOUND ID="ID_bgsound" SRC="sound.wav">
</HEAD>
<BODY>
    <BUTTON onclick="GenauEinmal()"></BUTTON>
    <BUTTON onclick="Endlos()"></BUTTON>
    <BUTTON onclick="KanalVerteilung(-10)"></BUTTON>
    <BUTTON onclick="KanalVerteilung(10)"></BUTTON>
    <BUTTON onclick="KanalVerteilung (0)"></BUTTON>
    <B>Volume control:</B>&nbsp;&nbsp;&nbsp;
    <INPUT TYPE ="radio" NAME="ID_RadioButtonGruppe"
        onpropertychange="ID_bgsound.volume=-10;"
    >Mute

    < INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
        onpropertychange="ID_bgsound.volume=-7;"
    >25% Volume

    < INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe" CHECKED
        onpropertychange="ID_bgsound.volume=-5;"
    >50% Volume

    < INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
        onpropertychange="ID_bgsound.volume=-2;"
    >75% Volume

    < INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
        onpropertychange="ID_bgsound.volume=0;"
    >100% Volume
</BODY>
```

Beispiel für Sound mit Sekundenanzeige:

```
<HTML>
<HEAD>
<SCRIPT>
```



```

// ++++++ globale Variablen, die verändert werden können
var SoundUrl = "56sec.mid";
var SoundDauerInSekunden = 56; // Dauer muss exakt stimmen !

var PixelBreiteProBalkenErweiterung = 10;

// ++++++ Browser-Typ ermitteln
// Dieser Quellcode muss VOR allen anderen Routinen codiert sein, damit zuerst abgearbeitet
var ns = document.layers ? true : false;
var ie = document.all ? true : false;

// ++++++ Routinen der Sekundenzählung
var SekundenZahler = 0;
var SekundenZahlerTimeoutID = null;

function SekundenZaehlen() // wird durch Rekursion alle Sekunde neu gestartet
{
    // Zähler erhöhen
    SekundenZahler++;

    // visuelle Anzeige im Dokument auffrischen, also alle Audrucke der Style-Werte
    // neu berechnen und damit alle DIV's neu visualisieren
    document.recalc();
}

function SekundenZaehlen_Start()
{
    // prüfen ob Sekundenzählen nicht bereits läuft
    if (SekundenZahlerTimeoutID == null)
    {
        // nicht aktiv

        // Rekursion starten: SekundenZaehlen() wird permanent alle Sekunde aktiviert
        SekundenZahlerTimeoutID = setInterval("SekundenZaehlen()", 1000);
    }
}

function SekundenZaehlen_Stop()
{
    // prüfen ob Sekundenzählen aktiv ist
    if (SekundenZahlerTimeoutID != null)
    {
        // aktiv, also stoppen
        clearInterval(SekundenZahlerTimeoutID);
        SekundenZahlerTimeoutID = null;
    }
}

// ++++++ Routine zur Erzeugung Sound-Objekt
function SoundObjektErzeugen(SoundFileUrl, SoundFileDauerInSekunden)
{
    this.SoundFileUrl = SoundFileUrl;
    this.SoundFileDauerInMillisekundeSekunden = SoundFileDauerInSekunden * 1000;
    // Timerzeit für Rekursion
    this.SoundFileBeendet = true; // kein Sound aktiv
}

// ++++++ Routinen zur Wiedergabe Sound-Objekt
var SoundTimeoutID=0;

function SoundAbspielen()
{
    // prüfen ob Sound nicht bereits aktiv ist
    if (SoundObjekt.SoundFileBeendet)
    {
        // Anzeige intialisieren
        SekundenAnzeigeInit();

        // Sekundenzähler starten, wobei das Zählen eigenständig und parallel erfolgt
        SekundenZaehlen_Start();

        // Sound erzeugen und sofort starten durch Url-Zuweisung
        ID_BGSound.src=SoundObjekt.SoundFileUrl ;
    }
}

```



```

        SoundObjekt.SoundFileBeendet=false;

        // Millisekunden warten und danach die Funktion SoundAbspielen() neu aufrufen
        SoundTimeoutID = setTimeout(
            "SoundAbspielen()",
            SoundObjekt.SoundFileDauerInMillisekundeSekunden
        );
    }
    else
    {
        // Dieser Zweig wird erst mit dem 2. Aufruf der Funktion abgearbeitet

        // Sound zu Ende
        SoundObjekt.SoundFileBeendet=true;

        // Sekundenzähler stoppen
        SekundenZaehlen_Stop();

        // und Meldung
        var TimerUngenauigkeit1 = SoundDauerInSekunden - SekundenZahler;
        var TimerUngenauigkeit2 = TimerUngenauigkeit1 / SoundDauerInSekunden;
        alert(
            "Wiedergabe beendet\nUngenauigkeit des Timers = "
            + TimerUngenauigkeit1.toString()+" Sekunden\n"
            + "also " + TimerUngenauigkeit2.toString() + " Ticks pro Sekunde"
        );
    }
}

// ++++++ Sekunden-Anzeige initialisieren
function SekundenAnzeigeInit()
{
    // ---- Variablen init
    SekundenZahler = 0;
    SekundenZahlerTimeoutID = null;

    // ---- visuelle Anzeige erzeugen
    // - - Sekundenbalken und Sekundenzähler dynamisch visualisieren
    //     Es wird jedem DIV als Style-Wert ein Ausdruck hinterlegt, also kein Wert.
    //     Der Ausdruck liefert den Wert , welcher sofort das Layout der
    //     DIV's beeinflusst.
    //     Jeder Ausdruck besitzt den SekundenZahler als Komponente.
    //     Damit ändert sich der Wert des Ausdrucks.
    //     Für die Neuberechnung des Ausdrucks ist der Aufruf von
    //     document.recal()
    //     nötig.
    //     Dieser Aufruf erfolgt in SekundenZaehlen(), also permanent pro Sekunde.
    //     Damit wird der Style-Wert permanent neu berechnet.
    //     Damit visualisieren sich die DIV's permanent neu.

    // Sekundenbalken in der Style-Eigenschaft width (Breite) mit Ausdruck belegen,
    //     also dynamisch anzeigen
    ID_DIV_Balken.style.setExpression( "width",
        "SekundenZahler * PixelBreiteProBalkenErweiterung"
    );

    // Sekundenzähler in der Eigenschaft .innerText mit Ausdruck belegen,
    //     also dynamisch anzeigen
    ID_DIV_SekundenZahler.setExpression("innerText","SekundenZahler.toString()");

    // - - Messlatte statisch anzeigen
    ID_DIV_MessLatte.style.width = SoundDauerInSekunden * PixelBreiteProBalkenErweiterung;
    ID_DIV_MessLatte.innerText = "Der Sound dauert "
        + SoundDauerInSekunden.toString()
        + " Sekunden";
}

// ##### Dieser Teil wird mit dem Laden des Dokumentes abgearbeitet #####
if (ie)
{
    document.write('<BODY></BODY>');

    document.write(
        '<BGSOUND ID= "ID_BGSound" LOOP="0">'
    );
}

```



```

document.write(    '<DIV ID="ID_DIV_Balken"'
                  +    'STYLE="background-color:lightblue"'
                  + '>'
                  + '</DIV>'
                  + '<BR>'
                );

document.write(    '<DIV ID="ID_DIV_SekundenZahler"'
                  +    'STYLE="color:hotpink;font-weight:bold"'
                  + '>'
                  + '</DIV>'
                  + '<BR>'
                );

document.write(    '<DIV ID="ID_DIV_MessLatte"'
                  +    'STYLE="color:white;background-color:gray"'
                  + '>'
                  + '</DIV>'
                );

// ++++++ Sound initialisieren und starten mit Laden des Dokumentes

// ---- Sound-Objekt erzeugen anhand globaler Variablen
SoundObjekt = new SoundObjektErzeugen(SoundUrl, SoundDauerInSekunden);

// ---- Sound-Objekt wiedergeben
SoundAbspielen(); // meldet wenn Wiedergabe beendet ist
}
</SCRIPT>
</HEAD>
<BODY>
<!-- BODY-Teil muss leer bleiben -->
</BODY>
</HTML>

```

Beispiel für Start und Stop des Sounds:

Das Objekt bgsound besitzt **keine** Methoden und auch nicht für Start und Stop des Sounds. Folgendes Beispiel schafft aber Abhilfe:

```

function SoundObjektErzeugen(Kette,Wert)
{
    this.SoundFileUrl      = Kette;
    this.SoundFileDauerInSekunden = Wert;
}

function SoundStummInstanzieren()
{
    document.write(    '<BGSOUND ID="ID_BGSound"'
                    +    ' SRC="' + StummSoundObjekt.SoundFileUrl + "'
                    +    ' LOOP="1"'
                    + '>'
                ); // alternativ auch SRC-Attribut weglassen
}

var SoundEndlosAbspielenAktiv=false;

function SoundEndlosAbspielen()
{
    ID_BGSound.src= SoundObjekt.SoundFileUrl;
    ID_BGSound.loop='infinite';
}

function SoundStoppen()
{
    ID_BGSound.src= StummSoundObjekt.SoundFileUrl; // alternativ Leerkette zuweisen
    ID_BGSound.loop='1';
}

SoundDauerInSekunden = 88;
SoundUrl              = "sound/test88.mid";
StummSoundUrl        = "sound/stumm.mid"; // leere MIDI-Datei ohne Daten, also ohne Tonwiedergabe

```







```

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=-7;"
>25% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe" CHECKED
onpropertychange="ID_bgsound.volume=-5;"
>50% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=-2;"
>75% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=0;"
>100% Volume

```

</BODY>

.disabled

Interaktionsfähigkeit  
nur wenn sichtbar so User-Interaktion möglich

.id

Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID)

.isDisabled

Interaktionsfähigkeit

nur wenn sichtbar so User-Interaktion möglich

.loop

Anzahl der Wiederholungen

nur Objekt BGSOUND bzw. IMG mit Sound bzw. INPUT-Element mit Sound

-1 endlos

0 genau 1 mal

> 0 Anzahl der Wiederholungen

Standard ist 1

1 entspricht 0

Beispiel:

```

<HEAD>
<SCRIPT>
function KanalVerteilung(Wert)
{ ID_bgsound.balance = Wert;}

function GenauEinmal()
{
    ID_bgsound.loop = 1;
    ID_bgsound.src = ID_bgsound.src; // restart
}

function Endlos()
{
    ID_bgsound.loop = -1;
    ID_bgsound.src = ID_bgsound.src; // restart
}
</SCRIPT>
<BGSOUND ID="ID_bgsound" SRC="sound.wav">
</HEAD>
<BODY>
<BUTTON onclick="GenauEinmal()"></BUTTON>
<BUTTON onclick="Endlos()"></BUTTON>
<BUTTON onclick="KanalVerteilung(-10)"></BUTTON>
<BUTTON onclick="KanalVerteilung(10)"></BUTTON>
<BUTTON onclick="KanalVerteilung (0)"></BUTTON>
<B>Volume control:</B>&nbsp;
<INPUT TYPE ="radio" NAME="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=-10;"
>Mute

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=-7;"
>25% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe" CHECKED
onpropertychange="ID_bgsound.volume=-5;"
>50% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
onpropertychange="ID_bgsound.volume=-2;"
>75% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"

```







Behavior .style.time2..... 1  
Bezeichner des Objektes..... 8  
bgsound Objekt..... 1  
bgsound Objekt mehrere im Dokument ..... 1  
Dokument mehrere bgsound Objekte..... 1  
HIDDEN ..... 7  
Hintergrundmusik ..... 1  
Hintergrundmusik Wiederholungen ..... 8  
ID ..... 8  
IE und Sound ..... 1  
Interaktionsfähigkeit Objekt..... 8  
Lautstärke Wiedergabe..... 9  
Lautstärkeeinstellung von Windows ..... 1  
LOOP ..... 1, 7  
Objekt Bezeichner ..... 8  
Objekt bgsound..... 1  
Objekt bgsound mehrere im Dokument ..... 1

Objekt Interaktionsfähigkeit ..... 8  
Objekt Referenzierung ..... 8  
Objekt Status ..... 9  
Referenzierung Objekt ..... 8  
Sound im IE..... 1  
Sound Wiederholungen..... 8  
Sounderzeugung ..... 1  
Soundplayer..... 7  
Status des Objektes ..... 9  
Videoclip-Sound ..... 1  
VOLUME..... 1  
Wiedergabe-Lautstärke ..... 9  
Wiederholungen Hintergrundmusik..... 8  
Wiederholungen Sound..... 8  
Windows Media Player 7.1 ..... 1  
windowseigene Lautstärkeeinstellung ..... 1  
Zeiger..... 8

