

EINFÜHRUNG IN DAS ERSTELLEN VON SCRIPTEN FÜR
OUTLOOK-EXPRESS STATIONERIES

Visual Basic Script **Einführung zum Erstellen** **von Stationeries**





Inhaltsverzeichnis

Einleitung	3
Lektion 1: Hintergrund Scrolling.....	4
Lektion 1a: EML-Datei erstellen.....	9
Lektion 2: Hintergrund-Scrolling eines Div-TAGS	11
Lektion 3: Ebenen verwenden, gleichzeitiges Scrollen mehrerer Div-Elemente	15
Lektion 4 Filter: Teil 1: Shadow- und Alphafilter.....	19
Lektion 5: Bilder überblenden mit dem Alphafilter.....	24
Lektion 6: der revealTrans - Filter	30
Lektion 7: der Chroma-Filter	39
Lektion 8: Meta Creations Filter.....	43
Lektion 9 - Bewegte Grafiken Einführung	48
Lektion 10: ein etwas komplexeres Animationsprogramm für Objekte	51



Einleitung

Briefpapier für Outlook Express, das ist das Thema dieses Tutorials.

Welche Kenntnisse sollten vorhanden sein?

Nun, Sie sollten Grundkenntnis in Html und der Erstellung von Internetseiten haben und ebenfalls sollten Sie schon mal mit Outlook-Express gearbeitet haben.

Der Teil 1 des Tutorials beschäftigt sich mit Hintergrundbewegung, diversen Filtern und bewegten Objekten.

Es wird ebenfalls auf die Umsetzung und Anpassung von HTML auf Outlook-Express eingegangen.

Sie erfahren, wie Sie mit Hilfe von Cascade Style Sheet Anpassungen, wie Positionierung, Größe, Breite und Farbanpassung von Objekten durchführen können.

Als Scriptsprache verwende ich im Tutorial Visual Basic Script, wer Kenntnisse in Javascript besitzt, kann die Beispiele leicht in diese Sprache umsetzen, da sie sich doch sehr ähnlich sind.

Das Tutorial soll keine Einführung in HTML darstellen, obwohl manche TAGS, das sind die Anweisungen in HTML, besprochen werden.

Nun wünsche ich Ihnen viel Spaß mit dem Tutorial, es würde mich sehr freuen, wenn ich das eine oder andere Ergebnis bewundern dürfte.

Mit freundlichen Grüßen

Werner Jakob



Lektion 1: Hintergrund Scrolling

Entpacken Sie die Zip-Datei Lektionen 1-10.zip.
Wählen Sie nun den Ordner Lektion 1

In der Lektion 1 erfahren Sie, wie ein bewegter Hintergrund erstellt wird und in verschiedene Richtungen bewegt werden kann.
Zusätzlich werden die benutzten CSS-Anweisungen erklärt.

1. Schritt:

Öffnen sie mit einem Texteditor oder Phase 5 oder einen von Ihnen bevorzugten HTML-Editor ein das Dokument Lektion1.htm und geben Sie folgenden Code ein:

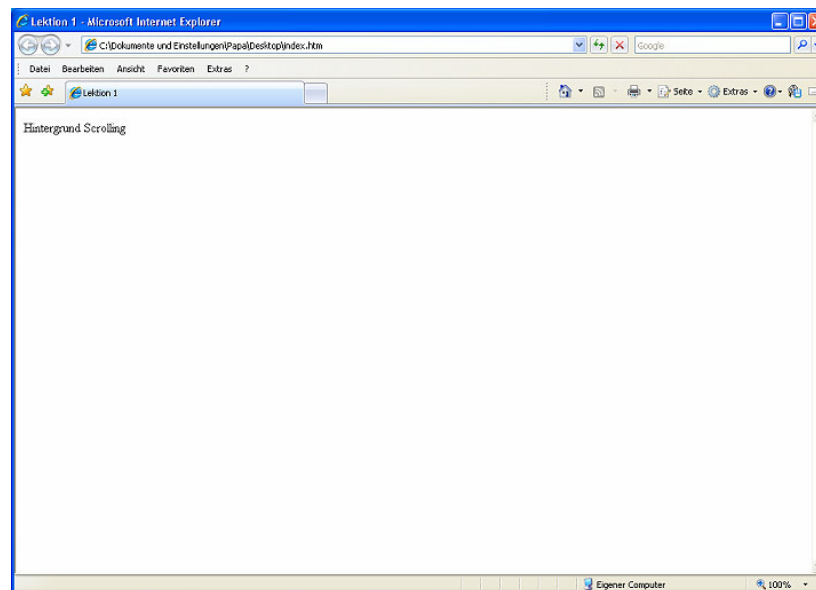
```
<html>
<head>
<title>Lektion 1</title>
</head>
<body>
<p>Hintergrund Scrolling</p>
</body>
</html>
```

Kurze Erklärung:

```
<p>Hintergrund Scrolling</p>
```

Ein Paragraph (Absatz) wurde erstellt, der auf der HTML-Seite den Text ‚Hintergrund Scrolling‘ darstellt.

Wenn Sie mit einem normalen Texteditor arbeiten, speichern Sie nun die Datei mit dem Namen Lektion1.htm in Ihrem erstellten Ordner ab. Anschließend doppelklicken Sie auf die erstellte Datei, sie wird nun mir dem Internetexplorer geöffnet und sie sehen das Resultat.





2. Schritt:

In nächsten Schritt werden wir den Text im Paragraph formatieren. Dazu erstellen wir eine CSS-Anweisungsfolge.
Fügen Sie folgenden Code zu dem bisher erstellten hinzu. Der neue Code ist rot dargestellt.

```
<html>
<head>
<style type="text/css">
#lek1 {      color:brown;
             font-size:44;
             text-align:center;
             font-style:italic }
</style>
<title>Lektion 1</title>
</head>
<body>
<p id=lek1 >Hintergrund Scrolling</p>
</body>
</html>
```

Mit `<style type="text/css">` leiten wir die Stylesheet Anweisung ein, mit `</style>` wir sie wieder beendet.

`#lek1` bedeutet, wir definieren eine so genannte Individualklasse, die später durch eine ID einem TAG zugeordnet werden kann.

In unserem Fall wird die Individualklasse `#lek1` als `id="lek1"` dem Paragraph zugeordnet. Es folgen die Anweisungen.

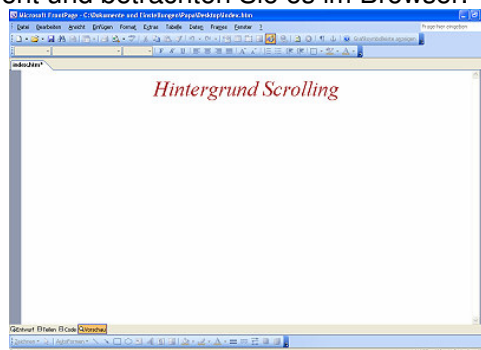
Die Anweisungen müssen dabei in eine geschweifte Klammer eingeschlossen werden, jede Anweisung muss mit einem Semikolon abgeschlossen werden, außer der letzten Anweisung vor der schließenden Klammer.

```
{ Anweisung 1;Anweisung 2;Anweisung 3;Anweisung 4 }
```

Der Marker soll folgende Einstellungen erhalten:

<code>color:brown</code>	Farbe braun
<code>font-size:44</code>	Schriftgröße 44
<code>text-align:center</code>	Der Absatz soll horizontal zentriert im Dokument dargestellt werden.
<code>font-style:italic</code>	Der Text im Absatz soll kursiv erscheinen.

Speichern Sie das Dokument und betrachten Sie es im Browser.





Schritt 3:

Wir fügen nun eine Hintergrundgrafik hinzu und blenden die Bildlaufleiste links aus. In Ihrem Ordner befindet sich eine Datei mit dem Namen h1.jpg.

Diese Bilddatei möchte ich nun als Hintergrund haben und zugleich die Bildlaufleiste ausblenden. Fügen Sie folgenden, rot dargestellten Code neu hinzu.

```
<html>
<head>
<title>Lektion 1</title>
<style type="text/css">
#lek1      { color:brown;font-size:44;text-align:center;font-
style:italic }
</style>
</head>
<body scroll=no background="h1.jpg">
<p id=lek1>Hintergrund Scrolling</p>
</body>
</html>
```

Schritt 4:

Nun wollen wir den Hintergrund bewegen.

Fügen Sie roter Farbe dargestellten Code hinzu ,den ich dann anschließend erklären werde.

```
<html>
<head>
<title>Lektion 1</title>
<style type="text/css">
#lek1      { color:brown;font-size:44;text-align:center;font-
style:italic }
</style>
</head>
<body scroll=no background="h1.jpg">
<p id=lek1>Hintergrund Scrolling</p>
</body>
<script language=vbscript>
x=0
Bewegen()
sub bewegen()
x=x+1
document.body.style.backgroundColor=x setTimeout "bewegen()",32
end sub
</script>
</html>
```

Wir verwenden hier Visual Basic Script als Programmiersprache. Ebenso könnte JavaScript oder JSScript verwendet werden, die Sprachen sind ähnlich.

Visual Basic Script und JSScript können nur vom Internetexplorer, Outlook-Express oder Outlook interpretiert werden.

Der Scriptteil wird eingeleitet mit `<script language=vbscript>` und endet mit `</script>`.

Es gibt in VBS(Visual Basic Script) drei Befehle, mit denen man den Hintergrund eines Dokuments, einer Tabelle oder eines DIV-Objekts bewegen kann.

Die Befehle sind:

```
document.body backgroundColor
document.body.backgroundColor und
document.body.backgroundColor
```



VB- SCRIPTING FÜR STATIONERIES

Mit `document.body.backgroundPositionX` kann ich den Hintergrund horizontal nach links oder rechts bewegen.

Mit `document.body.backgroundPositionY` kann ich den Hintergrund vertikal nach oben oder unten bewegen.

Mit `document.body.backgroundPosition` kann ich beide Richtungen gleichzeitig ansprechen.

Nun zu unserem Beispiel von Seite 5:

`x=0` Hier definiere ich eine Variable, die den Wert 0 bekommt.

`bewegen()` Das ist ein Sprungbefehl. Er verzweigt zu einer Subroutine, die den Namen haben muss, den ich hier verwende. In unserem Beispiel eben `bewegen`.

Eine Subroutine beginnt immer mit: `Sub name` und endet mit `end sub`.

Eine Subroutine wird erst ausgeführt, wenn auf sie durch einen Sprungbefehl verzweigt wird. Es gibt ein paar Ausnahmen, die ich später im Kurs noch ansprechen werde.

```
sub bewegen()
```

```
x=x+1
```

```
document.body.style.backgroundPositionX=x setTimeout "bewegen()",32
```

```
end sub
```

Nachdem die Subroutine durch den Befehl `bewegen` angesprungen wird, werden die Befehle in der Subroutine ausgeführt.

```
x=x+1
```

X hatte beim ersten Mal den Wert 0. Dieser Wert wird nun um 1 erhöht, so dass x den neuen Wert 1 besitzt.

```
document.body.style.backgroundPositionX=x
```

Der Hintergrund des Dokuments wird nun an die Position verschoben, die durch den Wert von X bestimmt ist. Der äußerste, linke Rand eines Dokuments hat die Position 0.

Das Pixel, das durch das Verschieben des Hintergrundbildes rechts aus dem Dokument ‚fällt‘, wird links wieder eingefügt.

```
setTimeout "bewegen()",32
```

`setTimeout` ist ein Zeitsteuerbefehl.

Nach der Anweisung `setTimeout` muss der Name einer Subroutine stehen die, nach der Zeit, die in tausendstel Sekunden eingetragen wird und durch ein Komma getrennt sein muss, angesprungen werden soll.

In unserem Fall bedeutet das:

Die Subroutine `bewegen()` soll nach 32'tausenstel Sekunden erneut aufgerufen werden.

Dadurch geschieht folgendes.

x wird wieder um eins erhöht, hat also beim zweiten Durchlauf den Wert 2. Der Hintergrund wird dadurch an die Position 2 geschoben.

Es wird wieder 32'tausenstel Sekunden gewartet, dann wird die Subroutine wieder durchlaufen.

Das geschieht nun so lange, bis das Dokument geschlossen, oder die Schleife programmtechnisch verlassen wird.

Weitere Möglichkeit.

geben sie statt `x=x+1` folgendes ein: `x=x-1`

Betrachten Sie das Ergebnis wieder in der Vorschau oder im Internetexplorer.

Spielen Sie nun etwas mit der Zeit, z.B. `setTimeout „bewegen“,1000`, so dass die Bewegung nur jede Sekunde ausgeführt wird.

Erhöhen Sie auch mal den Wert der Addition, z.B. `x=x+10` usw.

Wenn Sie verstanden haben, was da abläuft, dann machen Sie mit der Dokumentation auf der nächsten Seite weiter.



VB- SCRIPTING FÜR STATIONERIES

Wir wollen nun den Hintergrund nach unten, bzw. nach oben bewegen. Dazu tauschen wir einfach den Befehl:

```
document.body.style.backgroundPositionX=x um in  
document.body.style.backgroundPositionY=x
```

Wenn wir nun $x=x+1$ eingeben, wird der Hintergrund nach unten bewegt, bei $x=x-1$ wird er nach oben bewegt.

Im nächsten Beispiel wollen wir den Hintergrund in zwei Richtungen bewegen. dazu erweitern wir das Script um drei Befehle.

```
<script language=vbscript>  
x=0  
y=0  
bewegen()  
sub bewegen()  
                                x=x+1  
y=y+1  
document.body.style.backgroundPositionX=x  
document.body.style.backgroundPositionY=y  
setTimeout "bewegen()", 32  
end sub  
</script>
```

Nun wird der Hintergrund nach rechts unten bewegt.

Versuchen Sie nun selbst, den Hintergrund in folgende Richtungen zu bewegen.

Nach rechts,
rechts unten,
unten,
links unten,
links,
links oben,
oben,
rechts oben

Man kann die zwei Bewegungsbefehle auch zu einem zusammenfassen.

Beispiel:

```
<script language=vbscript>  
x=0  
y=0  
bewegen()  
sub bewegen()  
                                x=x+1  
y=y+1  
document.body.style.backgroundPosition=x&" "&y  
setTimeout "bewegen()", 32  
end sub  
</script>
```

Damit haben Sie alle Möglichkeiten kennen gelernt, um einen Hintergrund zu bewegen. In der nächsten Lektion werden wir das Dokument an Outlook-Express übertragen und eine EML-Datei erstellen, also eine Datei, die wir dann immer als Briefpapier zur Verfügung haben.



Lektion 1a: EML-Datei erstellen

Wir wollen nun aus unserer HTML-Datei ein Mail erstellen, das wir in Zukunft nur noch anklicken müssen, um es sofort parat zu haben.

Dabei gibt es eigentlich nur eine Sache zu beachten, alle externen Daten in unserem Dokument müssen die volle Pfadangabe haben.

Z.b.: aus `<body background="h1.jpg"` muss werden -

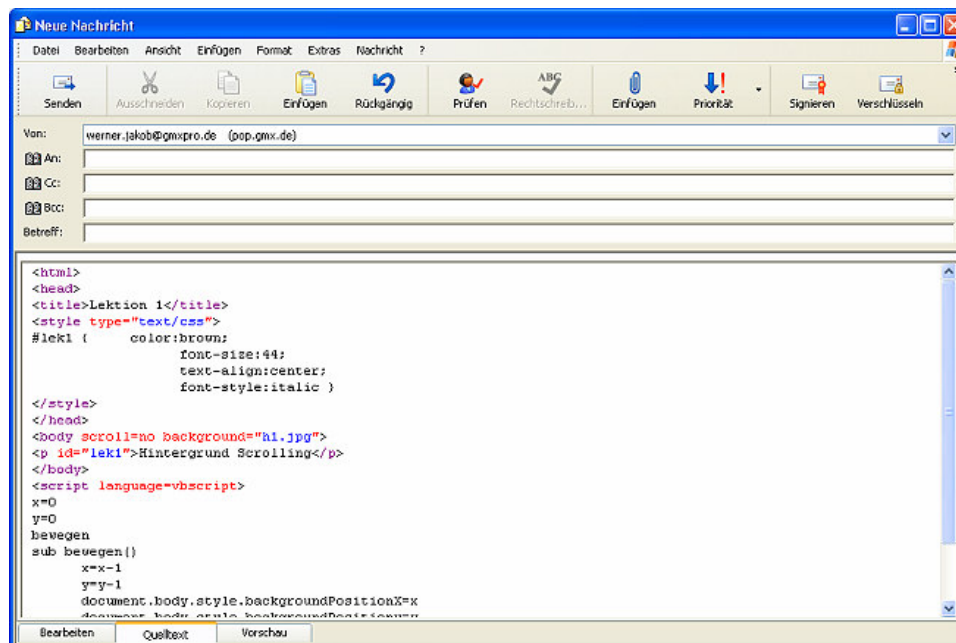
`<body background=file:///C:\Dokumente und Einstellungen\tut_stationery\Lektion1\h1.jpg">`

Wobei hier nach `file:///` natürlich ihr Pfad zu dem Bild eingetragen werden muss.

Diese volle Pfadangabe müssen Sie für alle Bilder, Musik oder sonstige Dateien angeben, die in OE-Mail dargestellt werden sollen.

Vorgehensweise:

1. Öffnen Sie ihr Outlook-Express und erstellen Sie eine neue Mail, ebenfalls das von und erstellte HTML-Dokument in Ihrem Editor öffnen.
2. Gehen Sie im E-Mailformular auf Menü- Format und stellen Sie sicher das bei Rich-Text(HTML) ein schwarzer Punkt ist, d.h. dass ihr E-Mail HTML bearbeiten kann.
3. Wählen Sie nun im neuen E-Mail Formular links unten: Quelltext.
4. Sie sehen nun den bereits vorgegebenen Quelltext, den sie jetzt markieren und löschen können.
5. Wechseln Sie nun zum Quelltext ihres HTML-Dokuments, markieren alles. Kopieren und anschließend den gesamten HTML-Code in Ihr Emailformular einfügen.





VB- SCRIPTING FÜR STATIONERIES

6. Sie können nun in der Vorschau ihr Werk betrachten und es sollte genau so funktionieren wie vorher als HTML-Datei.
7. Läuft alles nach Wunsch, können Sie jetzt das E-Mail speichern, um es an sofort immer zur Verfügung zu haben.
8. Dazu gehen Sie auf Datei speichern unter und vergeben einen gewünschten Namen. Die Dateiendung sollte dabei .eml sein. Es werden dabei alle Bilder und eingebundenen Dateien mit gespeichert und der Verweis auf die eingebundenen Dateien gelegt. Sollten Sie die Meldung erhalten: Mindestens ein Bild in der Datei wurde nicht gefunden, dann haben Sie eine Pfadangabe nicht richtig angegeben und sie sollten Ihren Code noch mal dahingehend überprüfen.

Wenn Sie Ihre eml-Datei mal mit einem Texteditor öffnen, werden Sie feststellen, das alle Pfade Ihre Bilder und Dateien, die Sie im Code eingegeben hatten, ersetzt wurde durch Eine Zeile die in etwa so aussieht [cid:000e01c745e9\\$a03394e0\\$0100007f@Werner](mailto:cid:000e01c745e9$a03394e0$0100007f@Werner)
Das ist nun die Adresse ihres Hintergrundbildes, das nun direkt im E-Mail gespeichert ist.

Wenn Sie im Text weiter nach unten gehen, finden Sie einen weiteren Eintrag:

```
-----=_NextPart_000_0013_01C745F2.053A56A0
Content-Type: application/octet-stream;
      name="h1.jpg"
Content-Transfer-Encoding: base64
Content-ID: <000e01c745e9$a03394e0$0100007f@Werner>

/9j/4AAQSkZJRgABAQAAQABAAD/2wBDAAYEBQYFBAYGBQYHBwYIChAKCgkJChQODww
QFxQYGBcU
FhYaHSUfGhSjHBYWICwglyYnKSopGR8tMC0oMCUoKSj/2wBDAQcHBwoIChMKChMoGhYaKC.
..... usw.
```

Genau an dieser Stelle beginnt der Code des Bildes.

In der nächsten Lektion möchte ich Ihnen zeigen, wie Sie einen DIV erstellen und dessen Hintergrund ändern.



Lektion 2: Hintergrund-Scrolling eines Div-TAGS

In Lektion 1 hatten wir den Hintergrund eines Dokuments bewegt.
In dieser Lektion werden wir ein <DIV> erzeugen, das ebenso, wie ein Dokument einen eigenen Hintergrund haben kann. Diesen Hintergrund werden wir wieder in Bewegung setzen. Wählen Sie den Ordner Lektion2 und öffnen Sie anschließend mit einem Editor die Datei Lektion2.htm.

Folgender Code wird angezeigt:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
</head>
<body >

</body>
</html>
```

Als erstes füge ich im <body> die Hintergrundgrafik hinzu.
Über CSS werde ich nun den body-TAG formatieren.
Dazu ändere ich den Code. Der rote Teil kommt neu hinzu.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body { border:4 darkgreen groove;
margin-left:220 }
</style>
</head>
<body scroll=no background="h1.jpg">

</body>
</html>
```

Erklärung Cascade Style Sheet Anweisungen:

body	Anders wie in Lektion 1 habe ich hier kein #-zeichen vor body geschrieben. Das bedeutet, der TAG, hier body, wird ohne dass ich eine id vergeben muss, direkt mit den angegeben Formatierungen beeinflusst.
border:4 darkgreen groove	Das Dokument soll einem Rahmen um alle vier Seiten bekommen. Die Dicke des Rahmen ist 4 Pixel, Farbe des Rahmens ist braun, der Style ist groove. (siehe dazu meine Seiten über CSS)
margin-left:220	Wenn später in die E-Mail Text eingeben wird, ist der Einzug schon 220 Pixel vom linken Rand entfernt.

Im <body>-TAG habe ich noch die Hintergrundgrafik eingebunden,
background="h1.jpg"
und die Scrollleiste wurde ausgeblendet, scroll=no



VB- SCRIPTING FÜR STATIONERIES

Als nächstes füge ich einen DIV-TAG hinzu und auch gleich die Grafik für den Hintergrund des Div's, beide formatiere ich über CSS.

Ein Div-Tag können Sie sich als Fenster im Dokument vorstellen, das ebenso wie das Dokument selbst einen eigenen Hintergrund, Texte, Tabellen, Grafiken usw. beinhalten kann.

Nur kann ich leider in Outlook-Express das Bild nicht sofort als Hintergrund für den <DIV> festlegen, wir machen das mit einem kleinen Trick.

Fügen Sie nun zuerst die roten Codezeilen ein.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body      {      border:4 darkgreen groove;
              margin-left:220  }
#lek2     {      position:absolute;
              top=0;
              left=0;
              width:210 }
#bld2     {      position:absolute;
              left:-3000 }
</style>
</head>
<body scroll=no background="h1.jpg">

<div id=lek2></div>
</body>
</html>
```

Der <DIV> bekommt die ID=lek2.

Die Position des <DIV> ist absolute, das heißt ich kann ihn an einer beliebigen stelle positionieren. In unserem Fall ganz links oben: top=0, left=0.

Der <DIV> soll eine Breite von 210 Pixel haben. width=210 .

Das Bild stellen wir ganz normal mit ein, allerdings auch über CSS formatiert.

position:absolute, d.h., frei positionierbar, und nun kommt unser Trick,

left=-3000!!!

Damit schieben wir das Bild soweit über den linken Fensterrand, dass es nicht mehr zu sehen ist. Außerdem erhält es eine ID, hier die ID bld2.



VB- SCRIPTING FÜR STATIONERIES

Nun werden wir mit Hilfe von VBS ein Script erzeugen, mit dem wir Höhe des DIV-TAG berechnen werden. Er soll die Höhe des Fensters besitzen, wenn das Fenster verkleinert oder vergrößert wird, soll sich die Höhe automatisch anpassen.

Des weiteren werden wir den Hintergrund des DIV's mit unserem Bild, bld2, füllen. Das soll geschehen, noch bevor unsere Seite, oder später das E-Mail, geladen ist.

Fügen sie den rot gekennzeichneten Code hinzu, anschließend folgt die Erklärung.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body      {      border:4 darkgreen groove;
              margin-left:220   }
#lek2     {      position:absolute;
              top=0;
              left=0;
              width:210   }
#bld2     {      position:absolute;
              left:-3000  }
</style>
</head>
<body scroll=no background="h1.jpg">

<div id=lek2></div>
</body>
<script language=vbscript>
sub window_onResize()
    lek2.style.height=document.body.clientHeight
end sub
sub window_onLoad()
    lek2.style.backgroundImage="url(" & bld2.src & ")"
    lek2.style.height=document.body.clientHeight
end sub
</script>
</html>
```

Erklärung:

sub windowOnload() Eine Subroutine in VBS, die immer automatisch beim Laden einer Html-Seite, eines E-Mails in Html-Formel oder einer News im HTML-Format im Internetexplorer oder in Outlook-Express aufgerufen wird.

lek2.style.backgroundImage="url(" & bld2.src & ")"

Hier bestimmen wir die Hintergrundgrafik für unser DIV.

Unsere Grafik hat die ID bld2. Sie wurde ja im Dokument erstellt, aber um 3000 Pixel nach links verschoben, das sie nicht sichtbar ist. Allerdings kann ich sie jetzt als Hintergrundkachelung verwenden.

lek2.style.height=document.body.clientHeight

Mit diesem Befehl gleiche ich die Höhe meines DIV's der Höhe des Dokumentes an.

sub window_onResize Eine Subroutine, die immer dann automatisch ausgeführt wird, wenn sich die Größe des Dokuments am Bildschirm ändert. In diesem Fall muss die Größe des DIV's neu berechnet werden.



VB- SCRIPTING FÜR STATIONERIES

So, nun bleibt nur noch den Hintergrund meines div-Elements in Bewegung setzen, es gelten dabei die selben Regeln, wie in Lektion 1 schon gezeigt.

Hier der komplette Quelltext, die neu hinzugekommen Teile sind wieder rot.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body      {      border:4 darkgreen groove;
              margin-left:220  }
#lek2     {      position:absolute;
              top=0;
              left=0;
              width:210  }
#bld2     {      position:absolute;
              left:-3000  }
</style>
</head>
<body scroll=no background="h1.jpg">

<div id=lek2></div>
</body>
<script language=vbscript>
x=0
bewegen1
sub bewegen1 ()
    x=x+1
    lek2.style.backgroundColor=x
    setTimeout "bewegen1",32
end sub
sub window_onResize()
    lek2.style.height=document.body.clientHeight
end sub
sub window_onLoad()
    lek2.style.backgroundColor="url(" & bld2.src & ")"
    lek2.style.height=document.body.clientHeight
end sub
</script>
</html>
```

Damit haben wir die Grundlagen geschaffen, um einen Hintergrund in einem Div-Element zu bewegen.

Das gilt übrigens genauso für die Hintergründe, die ich einer Tabelle oder Tabellenfeldern zuordnen kann.

In der nächsten Lektion erfahren Sie, wie man mehrere Div_Elemente übereinander legen und in verschiedenen Ebenen darstellen kann.



Lektion 3: Ebenen verwenden, gleichzeitiges Scrollen mehrerer Div-Elemente

In dieser Lektion möchte ich Ihnen zeigen, wie sie mehrere div-Elemente übereinander legen können, in verschiedenen Ebenen aufgeteilt.

Öffnen Sie den Ordner Lektion3 und öffnen Sie anschließend die Datei Lektion3.htm mit einem Editor oder Html-Editor. Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 3</title>
<style type="text/css">
</style>
</head>
<body scroll=no>
<p id=tx>mehrfaches Scrollen mit div-TAGS</p>
<div id=lek3_1></div>
<div id=lek3_2></div>
</body>
</html>
```

Sie sehen im body-Tag das Attribute scroll=no, um die Scrollleiste auszublenden. Außerdem einen Paragraph mit der id=tx, einen div-TAG mit der id=lek3_1 und Einen div-TAG mit der id=lek3_2,

Für diese drei TAGS werde ich nun über CSS die Formatierungen bestimmen.

```
<style type="text/css">
body      {      border:4 brown groove   }
#lek3_1   {      position:absolute;
                border:4 brown groove;
                height:440;
                width:440;
                z-index:1 }
#lek3_2   {      position:absolute;
                border:4 brown groove;
                height:400;
                width:400;
                z-index:2 }
#tx       {      text-align:center;
                color:white;
                font-size:30;
                font-style:italic }
</style>
```



Erklärung der neuen CSS-Anweisungen

z-index

Das ist der Befehl zum Festlegen der Ebene eines Objekts.

Es können 99 Ebenen definiert werden.

Wird für ein Objekt keine Ebene definiert, liegt es automatisch in der Ebene 0.

Definiere ich ein Objekt in einer höheren Ebene, dann wird dieses Objekt angezeigt und Teile, die tiefer definiert sind, werden durch das höher liegende Objekt verdeckt.

Beispiel:

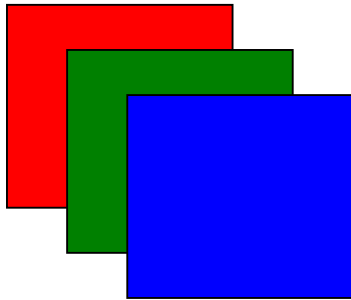
Ich habe hier 3 Objekte, ein rotes, ein grünes und ein blaues.

Das rote liegt in der Ebene 1, z-index=1,

das grüne liegt in der Ebene 2, z-index=2,

das rote liegt in der Ebene 3, z-index=3.

Dadurch ergibt sich folgendes Bild:



Im obigen Beispiel habe ich 2 DIV's definiert, lek3_1 und lek3_2.

Der DIV lek3_1 hat als Formatierung: z-index=1, liegt also in der Ebene 1.

Der DIV lek3_2 hat als Formatierung: z-index=2, liegt also in der Ebene 2 und überlagert somit den DIV in der Ebene 1.

Beide DIV's sollen noch einen Hintergrund bekommen, den wir aber wegen Outlook-Express erst in unserem VBScript Teil zuordnen können.

Doch die Bilder, die den Hintergrund bilden sollen, brauchen wir schon im HTML-Teil.

Also laden wir wieder die Bilder und setzen sie 3000Pixel nach links, damit sie im Dokument nicht sichtbar werden.

Fügen Sie folgende Zeilen hinzu:

Nach dem <body> Tag

```
  

```

Und als Formatierung im CSS Teil:

```
#bld1      {      position:absolute;  
                left:-3000 }  
#bld2      {      position:absolute;  
                left:-3000 }
```



Der gesamte Quellcode bis hierher:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body      {      border:4 brown groove  }
#lek3_1   {      position:absolute;
                border:4 brown groove;
                height:440;
                width:440;
                z-index:1 }
#lek3_2   {      position:absolute;
                border:4 brown groove;
                height:400;
                width:400;
                z-index:2 }
#bld1     {      position:absolute;
                left:-3000 }
#bld2     {      position:absolute;
                left:-3000 }
#tx       {      text-align:center;
                color:white;
                font-size:30;
                font-style:italic }
</style>
</head>
<body scroll=no background="h1.gif">


<p id=tx>mehrfaches Scrollen mit div-TAGS</p>
<div id=lek3_1></div>
<div id=lek3_2></div>
</body>
</html>
```

Als nächstes widmen wir uns dem Scriptteil.

Im Prinzip gibt es da nicht viel Neues.

1. Beim Laden sollen die Hintergründe für die DIV's festgelegt werden,
2. Die Größe und Breite der DIV's wird bestimmt
3. Die Hintergründe der Seite und der beiden DIV's sollen in verschiedene Richtungen bewegt werden.

Den gesamten Code sehen Sie auf der nächsten Seite.

Probieren Sie wieder selbst verschiedene Bewegungen der Hintergründe aus.



Gesamter Code Lektion 3:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 2</title>
<style type="text/css">
body      {      border:4 brown groove   }
#lek3_1   {      position:absolute;border:4 brown groove;
                height:440; width:440;z-index:1 }
#lek3_2   {      position:absolute;border:4 brown groove;
                height:400; width:400;z-index:2 }
#bld1     {      position:absolute;left:-3000 }
#bld2     {      position:absolute;left:-3000 }
#tx       {      text-align:center;color:white;
                font-size:30;font-style:italic }
</style>
</head>
<body scroll=no background="h1.gif">


<p id=tx>mehrfaches Scrollen mit div-TAGS</p>
<div id=lek3_1></div>
<div id=lek3_2></div>
</body>
<script language=vbscript>
x=0
y=0
bewegen1
sub bewegen1 ()
    x=x+1
    y=y+1
    document.body.style.backgroundPosition=-x&" "&y
    lek3_1.style.backgroundPosition=-x&" "&-y
    lek3_2.style.backgroundPositionX=x
    setTimeout "bewegen1",64
end sub
sub window_onResize()
    lek3_1.style.top=(document.body.clientHeight-lek3_1.clientHeight)/2
    lek3_1.style.left=(document.body.clientWidth-lek3_1.clientWidth)/2
    lek3_2.style.top=(document.body.clientHeight-lek3_2.clientHeight)/2
    lek3_2.style.left=(document.body.clientWidth-lek3_2.clientWidth)/2
end sub
sub window_onLoad()
    lek3_1.style.background="url(" & bld1.src & ")"
    lek3_2.style.background="url(" & bld2.src & ")"
    lek3_1.style.top=(document.body.clientHeight-lek3_1.clientHeight)/2
    lek3_1.style.left=(document.body.clientWidth-lek3_1.clientWidth)/2
    lek3_2.style.top=(document.body.clientHeight-lek3_2.clientHeight)/2
    lek3_2.style.left=(document.body.clientWidth-lek3_2.clientWidth)/2
end sub
</script>
</html>
```



Lektion 4 Filter: Teil 1: Shadow- und Alphafilter

Mit Filtern kann man die schöne Effekte bei der Darstellungen von Bildern und Grafiken Grafiken erreichen. Das geht von einfachen aus- und einblenden bis zum automatischen Fotoalbum.

Es gibt statische und dynamische Filter. Zuerst möchte ich Ihnen die wichtigsten statischen Filter etwas näher bringen.

Die am meisten benutzten sind dabei der Alpha-Filter, um Bilder transparent zu machen und der Shadow-Filter um Schatteneffekte, hauptsächlich von Texten zu erzeugen.

Öffnen Sie den Ordner Lektion 4 und laden sie das Dokument Lektion4.htm in einen Texteditor oder ihren HTML-Editor.

Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 4</title>
<style type=text/css>
</style>
</head>
<body scroll=no background="h1.gif">
<p >Alpha Filter und Shadow-Filter</p>
</body>
</html>
```

Den ersten Filter, den ich Ihnen etwas näher bringen möchte, ist der SHADOW-Filter.

Er ist gedacht, Schatteneffekte zu erzeugen.

Wir wollen einen ganz normalen Text in einem HTML-Dokument erstellen und diesen Text anschließend mit einem Filter hinterlegen.

Fügen sie den rot markierten Code wieder hinzu.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 4</title>
<style type=text/css>
#tx { color:white;
font-size:40;
position:absolute;
filter:progid:DXImageTransform.microsoft.shadow(color=black,direction=135) }
</style>
</head>
<body background="h1.gif" scroll=no>
<p id=tx >Alpha Filter und Shadow-Filter</p>
</body>
</html>
```

Wir haben hier einen Paragraph, mit der ID tx.

In CSS definieren wir die Formate. Die Schrift soll weiß sein, color=white.

Die Schriftgröße ist 40.

Die Position ist absolut, das heißt wir werden den Paragraph selbst positionieren.

Als nächstes kommt der Filter für den Paragraph.

Jeder Filter sollte beginnen mit:

Filter:progid:DXImageTransform.microsoft.art des Filters, in unseren Beispiel **shadow**.



VB- SCRIPTING FÜR STATIONERIES

Der Filter selbst benötigt zwei Angaben, einmal die Schattenfarbe, als zweites die Richtung des Schattens.

Die Schattenfarbe können Sie über die normalen HTML-Farben definieren.

Parameter für den Schattenfilter:

Schattenfarbe festlegen:

color=gewünschte Farbe

Damit legen Sie die Schattenfarbe fest. Folgende Schreibweisen sind zulässig:

color=#FFFFFF Farbe in Hexadezimal, (00-ff rotanteil,00-ff grünanteil,00-ff blauanteil)

color=black Farbe mit Namen HTML-Konform

Schattenrichtung festlegen:

direction=0 gerade nach hinten

direction=45 nach rechts oben

direction=90 nach rechts

direction=135 nach rechts unten

direction=180 nach unten

direction=225 nach links unten

direction=270 nach links

direction=315 nach links oben

Testen Sie jetzt einfach die verschiedenen Varianten durch und betrachten Sie das Ergebnis.

Als nächstes möchte ich den Paragraph zentriert in meinem Dokument darstellen.

Dazu schreibe ich eine Subroutine in VBS, ist Ihnen schon bekannt, es ist die Routine sub window Onload(), damit der Text sofort beim Laden das Dokuments zentriert wird.

Fügen Sie also folgenden, rot markierten Code hinzu:

```
html>
<head>
<title>Lektion 4</title>
<style type=text/css>
#tx { color:white;
      font-size:40;
      position:absolute;
filter:progid:DXImageTransform.microsoft.shadow(color=black,direction=135) }
</style>
</head>
<body scroll=no background="h1.gif">
<p id=tx>Alpha Filter und Shadow-Filter</p>
</body>
<script language=vbscript>
sub window_onLoad()
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
end sub
</script>
</html>
```

Den Paragraph zentriere ich folgendermaßen:

(Breite des dargestellten Dokuments – Breite des Paragraphes) / 2 ist der linke Punkt an dem mein Text beginnen soll.

Beispiel: Das Dokument ist momentan 800 Pixel breit, der Paragraph, also der Text ist 400 Pixel breit, dann wäre der linke Anfangspunkt: (800-400) / 2 ist 200 Pixel.



VB- SCRIPTING FÜR STATIONERIES

Testen Sie selbst unterschiedliche Texte, sie werden sehen, der Text erscheint immer zentriert.

Als nächstes möchte ich den Alphafilter ansprechen.
Mit dem Alphafilter kann ich bestimmte Teile eines Bildes transparent machen.

Um das zu demonstrieren, müssen wir zuerst ein Bild hinzufügen, das Bild über den Filter teilweise transparent machen, über VBS möchte ich das Bild noch zentrieren.

Fügen Sie also wieder den rot hinterlegten Code hinzu.

```
html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 4</title>
<style type=text/css>
#tx { color:white;
font-size:40;
position:absolute;
filter:progid:DXImageTransform.microsoft.shadow(color=black,direction=135) }
#bild1 { position:absolute;
filter:progid:DXImageTransform.Microsoft.Alpha(opacity=100,style=2) }

</style>
</head>
<body scroll=no background="h1.gif">
<p id=tx>Alpha Filter und Shadow-Filter</p>
<img id=bild1 src=bl.jpg></img>
</body>
<script language=vbscript>
sub window_onResize()
tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
bild1.style.left=(document.body.clientWidth-bild1.clientWidth)/2
bild1.style.top=(document.body.clientHeight-bild1.clientHeight)/2
end sub
sub window_onLoad()
tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
bild1.style.left=(document.body.clientWidth-bild1.clientWidth)/2
bild1.style.top=(document.body.clientHeight-bild1.clientHeight)/2
end sub
</script>
</html>
```

Sie sehen, das unser Bild ‚eingebettet‘ dargestellt wird, da die Ränder abasteigend ausgeblendet wurden.

Parameter des Alphafilters:

opacity= Stärke der Transparenzopacity = 0 - 100 im 1-er Schritt
0= voll transparent, unsichtbar
100= voll sichtbar, abhängig vom style

Style= 0 - 3 (Transparenzübergang festlegen)

0= das Bild wird als Ganzes transparent, opacity=100, das Bild ist voll sichtbar,
opacity=0, das Bild ist vollkommen transparent.

1= das Bild wird von rechts nach links ausgeblendet
2= das Bild wird von Innen nach außen, ovalförmig ausgeblendet
3= das Bild wird von Innen nach außen, rechteckig ausgeblendet.



VB- SCRIPTING FÜR STATIONERIES

Es gibt noch weitere Parameter, die den Anfangs- oder Endwert der Transparenz festlegen lassen. das sind:

startX=Startkoordinat der Grafik

startY=StartKoordinate der Grafik

finishX=Endkoordinat der Grafik

finishY=Endkoordinat der Grafik

Die Syntax dazu wäre:

Filter:prog... (opacity=xx,style=x,startx=x,starty=xx,finishX=xx,finishY=xx)

Aber diese Parameter werden nur selten verwendet.

Als nächstes möchte ich das unser Bild beim Anzeigen des Dokuments langsam eingeblendet wid.

Dazu ist eine kleine Änderung des Filters notwendig.

In unserem CSS Teil für das Bild stelle ich den Filter auf:

`filter:progid:DXImageTransform.Microsoft.Alpha(opacity=0,style=2)`

Dadurch ist das Bild unsichtbar.

Ich möchte es nun in einer Schleife programm-technisch langsam erscheinen lassen.

Dazu muss ich den Wert von opacity in bestimmten Abständen erhöhen, bis 100 erreicht ist.

```
x=0
setTimeout "bildblende()", 3000
sub bildblende()
    if x>100 then exit
    subbild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" & x
    & ", style=2)"
    setTimeout "bildblende()", 32
end sub
```

Erklärung:

X=0

setTimeout „bildblende“,3000

Sub bildblende

If x>100 then exit

es wird eine Variable definiert mit dem Anfangswert 0

es wird 3 Sekunden gewartet, dann wird die Subroutine bildblende ausgeführt.

Hier beginnt die Subroutine bildblende

Falls die Variable x größer als 100 ist, soll die Subroutine bildblende verlassen werden.

```
subbild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" & x & ",style=2)"
```

Hier findet die langsame Einblendung statt.

Machen wir ein paar Schritte:

Beim ersten Mal ist: **x=1**

Der Filter bekommt folgende String zugeordnet.

```
"(opacity=" & x & ",style=2)"
```

Durch das &-Zeichen wird dem Programm gesagt, füge zu dem Text den Wert von x also 1 hinzu. Der dertige String wäre also: **"(opacity=1,style=2)"**

Wäre der Wert von x 21 dann würde das Programm verstehen: **"(opacity=21,style=2)"**

Auf der nächsten Seite ist der gesamte Programmcode.



VB- SCRIPTING FÜR STATIONERIES

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 4</title>
<style type=text/css>
#tx {      color:white;
          font-size:40;
          position:absolute;
          filter:progid:DXImageTransform.microsoft.shadow(color=black,direction=135) }
#bild1 {  position:absolute;
          filter:progid:DXImageTransform.Microsoft.Alpha(opacity=0,style=2) }

</style>
</head>
<body scroll=no background="h1.gif">
<p id=tx>Alpha Filter und Shadow-Filter</p>
<img id=bild1 src=bl.jpg></img>
</body>
<script language=vbscript>
x=0
settimeout "bildblende",3000
sub bildblende()
  x=x+1
  if x>100 then exit sub
  bild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" & x &
  ",style=2) "
  settimeout "bildblende()",32
end sub
sub window_onResize()
  tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
  bild1.style.left=(document.body.clientWidth-bild1.clientWidth)/2
  bild1.style.top=(document.body.clientHeight-bild1.clientHeight)/2
end sub
sub window_onLoad()
  tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
  bild1.style.left=(document.body.clientWidth-bild1.clientWidth)/2
  bild1.style.top=(document.body.clientHeight-bild1.clientHeight)/2
end sub
</script>
</html>
```

Versuchen Sie selbst, den Programmcode so zu erweitern, dass das Bild, nachdem es ganz sichtbar ist, automatisch wieder ausgeblendet wird. Nachdem es wieder unsichtbar ist, soll es wieder langsam sichtbar werden, usw.

In der nächsten Lektion werden sie erfahren, wie Sie mit Hilfe des Alphafilters mehrere Bilder nacheinander ein- und ausblenden können.



Lektion 5: Bilder überblenden mit dem Alphafilter

In dieser Lektion wollen wir das Arbeiten mit dem Alphafilter noch etwas vertiefen. Wir haben drei Bilder, die wir nacheinander sichtbar bzw. unsichtbar machen wollen, also langsam überblenden. Öffnen Sie den Ordner Lektion5 und laden Sie die Datei Lektion5.htm in den Text- oder HTML-Editor. Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 5</title>
<style type=text/css>

</style>
</head>
<body scroll=no background="h1.gif">
<div id=tx>Müßte nicht jedes Paradies<br>ein Stück Wüste<br>haben<br>eine
Insel<br>der
Leere<br>wo<br>nur die Stille<br>laut ist<br>die ganze Fülle<br>im
Nichts<br>liegt</div>
</body>
</html>
```

Wir haben ein DIV-Objekt in den ein Text mit mehreren Zeilen steht, `
` ist ein HTML-TAG und bedeutet neue Zeile im Text.

Zuerst definiere ich die Formate für den DIV-Tag, der meinen Text enthält. Wenn Sie den Code ansehen, dann enthält der DIV-tag schon die Zuordnung: **id=tx**. Fügen Sie im Style-Bereich folgendes hinzu:

```
#tx { color:white;
font-size:30;
font-style:italic;
text-align:center;
position:absolute;
filter:progid:DXImageTransform.Microsoft.Shadow(color=black,direction=135) }
```

Kommt Ihnen das bekannt vor?

Ja, es wieder der Shadowfilter, den wir benutzen um die Schrift hervorzuheben.

Ich füge jetzt noch 3 Bilder in das Dokument und über CSS werden sie formatiert.

Auf der nächsten Seite der Code bisher, der rote Teil muss wieder hinzugefügt werden.



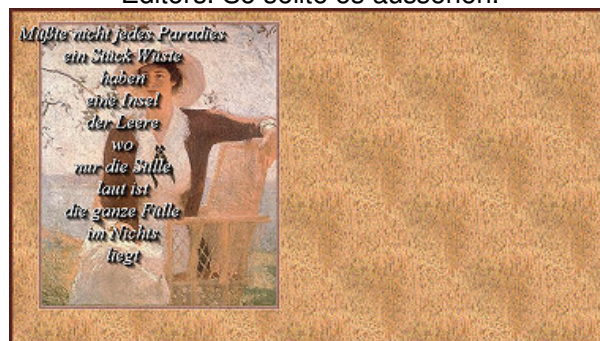
VB- SCRIPTING FÜR STATIONERIES

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Iktion 5</title>
<style type=text/css>
body {      border:8 #652D22 outset }
#bild1 {    position:absolute;
            filter:progid:DXImageTransform.Microsoft.Alpha(opacity=100,style=0) }
#bild2 {    position:absolute;
            filter:progid:DXImageTransform.Microsoft.Alpha(opacity=0,style=0) }
#bild3 {    position:absolute;
            filter:progid:DXImageTransform.Microsoft.Alpha(opacity=0,style=0) }
#tx {      color:white;
            font-size:30;
            font-style:italic;
            text-align:center;
            position:absolute;
            filter:progid:DXImageTransform.Microsoft.Shadow(color=black,direction=135) }
#rahmen1   {      position:absolute;left:40 }
</style>
</head>
<body scroll=no background="h1.gif">
<div id=rahmen1>
    </img>
    </img>
    </img>
</div>
<div id=tx>
    Müßte nicht jedes Paradies<br>ein Stück Wüste<br>haben<br>
    eine Insel<br>der Leere<br>wo<br>nur die Stille<br>laut ist<br>die ganze
    Fülle<br>im Nichts<br>liegt
</div>
</body>
</html>
```

Wenn Sie den Code betrachten, werden sie feststellen, das wir jetzt zwei DIV-Objekte haben. Einmal den DIV mit der ID=rahmen1 und dann den DIV mit der ID=tx. Beide Objekte sind absolut positioniert. Der DIV mit der ID=rahmen1 hat als Inhalt drei Bilder und ist 40 Pixel vom linken Dokumentrand nach rechts verschoben. Die obere Position ist noch nicht festgelegt.

Schauen wir uns nun noch die Formatierung der drei Bilder an. Alle 3 Bilder sind mit dem Alphafilter hinterlegt. Aber! Das erste Bild unterscheidet sich durch den Wert von opacity, also der Transparenz. Die Bilder b2 und b3 haben opacity=0, sind also unsichtbar. Bild b1 hat opacity=100, ist also sichtbar.

Betrachten Sie das bisherige Ergebnis im Internetexplorer oder in der Vorschau Ihres HTML-Editors. So sollte es aussehen.





VB- SCRIPTING FÜR STATIONERIES

Als nächstes werden wir die Schrift rechts anordnen und das Bild richtig positionieren. Das ganze soll geschehen wenn unser Dokument geladen wird.

Fügen Sie also nach `</body>` folgendes ein:

Wenn sie jeden Schritt in seiner Auswirkung betrachten wollen, dann fügen sie innerhalb von `sub window:onLoad()` einfach immer nur eine Zeile des Codes ein und betrachten dann das Ergebnis im Browser oder der Vorschau.

```
<script language=vbscript>
sub window_onLoad()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
</script>
```

Fangen wir mit dem Text an.

Der obere linke Punkt des Textes ist: **(Höhe des Dokuments-Höhe des Textes)/2**

```
tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
```

Das heißt, der Text wird horizontal zentriert.

Der linke Punkt des Textes wird bestimmt durch:

(Breite des Dokuments+Breite des Bildes1-Breite des Textes)/2

```
tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
```

Der Text wird also in der Mitte zwischen dem Bild und unserem Dokument dargestellt.

Der Rahmen, der um das Bild gezeigt wird, soll so breit und hoch sein wie das erste Bild.

```
rahmen1.style.width=bild1.clientWidth
rahmen1.style.height=bild1.clientHeight
```

Der Rahmen selbst soll vertikal im Dokument zentriert sein.

```
rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
```

Die nicht sichtbaren Bilder sollen im Rahmen sollen später ebenfalls im Rahmen horizontal zentriert werden.

```
bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
```

Damit das diese Positionierungen auch stattfinden wenn unser Dokument vergrößert bzw. verkleinert wird, füge ich diese Anweisungen auch noch in die Subroutine:

`sub windows_onResize()` ein.

```
sub window_onResize()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
```



VB- SCRIPTING FÜR STATIONERIES

Bis jetzt ist alles noch ziemlich statisch.

Deshalb erstellen wir jetzt die Routinen zum Wechseln der Bilder.

Das ist auf den ersten Blick gar nicht so einfach, aber wenn das Prinzip mal verstanden ist, dann geht es eigentlich recht gut.

Betrachten wir erst mal den theoretischen Ablauf.

Am Anfang ist Bild 1 zu sehen, Bild 2 und Bild 3 sind unsichtbar.

Bild1 soll ausgeblendet werden und Bild2 soll gleichzeitig eingeblendet werden.

Das realisieren wir über den Alphafilter, wie in der vorigen Lektion gezeigt.

Ich benötige dazu zwei Variable.

1. Variable zum Ausblende=x
2. Variable zum Einblenden =y

Danach das Gleiche noch mal, Bild2 wird ausgeblendet und gleichzeitig Bild3 eingeblendet. Ist das geschehen, dann soll das Ganze wieder von Vorne mit Bild1 ein und Bild3 aus, beginnen.

Doch beginnen wir langsam.

Nun soll langsam Bild 1 ausgeblendet und Bild 2 eingeblendet werden.

Der Code dazu sieht so aus:

```
X=140
y=-40
sub bild1blend()
  x=x-1
  y=y+1
  if x=-40 then
    x=140
    y=-40
    exit sub
  end if
  bild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=&x&, style=0) "
  bild2.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=&y&, style=0) "
  setTimeout "bild1blend()", 62
end sub
```

Erklärung:

X ist zu Beginn der Schleife 140.

Erinnern wir uns, opacity=100 heißt sichtbar. Also erst wenn x den Wert <100 hat, wird unser Bild langsam unsichtbar. Dadurch das ich x am Anfang mit 140 festlege, dauert es also eine Weile bis das Bild unsichtbar wird.

Das Gleiche gilt für Bild2, der Startwert ist -40, es dauert also eine Weile bis es langsam sichtbar wird.

Wenn x den Wert -40 erreicht hat, dann soll x wieder auf 140 gesetzt werden, y wieder auf -40 und die Schleife soll verlassen werden.

Auf der nächsten Seite noch mal der gesamte VisualBasic-Code, den Sie jetzt bitte hinzufügen.



VB- SCRIPTING FÜR STATIONERIES

```
<script language=vbscript>
x=140
y=-40
bild1blend()
sub window_onLoad()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
sub window_onResize()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
sub bild1blend()
    x=x-1
    y=y+1
    if x=-40 then
        x=140
        y=-40
        exit sub
    end if
    bild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=&x&, style=0)"
    bild2.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=&y&, style=0)"
    setTimeout "bild1blend()", 62
end sub
</script>
```

Nun wollen wir aber das nach dem Anzeigen von Bild 2, Bild2 wieder ausgeblendet und Bild 3 eingeblendet wird. Dazu bauen wir jetzt einfach eine 2. Schleife, die das durchführt. Zugleich müssen wir aber in der ersten Schleife dafür sorgen, dass die zweite Schleife gestartet wird. Dazu setzen wir in der ersten Schleife einfach vor dem exit sub den Aufruf von Schleife 2 ein.

Auf der nächsten Seite der Code für beide Schleifen.



VB- SCRIPTING FÜR STATIONERIES

```
<script language=vbscript>
x=140
y=-40
bild1blend()
sub window_onLoad()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
sub window_onResize()
    tx.style.top=(document.body.clientHeight-tx.clientHeight)/2
    tx.style.left=(document.body.clientWidth+bild1.clientWidth-tx.clientWidth)/2
    rahmen1.style.height=bild1.clientHeight
    rahmen1.style.width=bild1.clientWidth
    rahmen1.style.top=(document.body.clientHeight-rahmen1.clientHeight)/2
    bild2.style.left=(rahmen1.clientWidth-bild2.clientWidth)/2
    bild3.style.left=(rahmen1.clientWidth-bild3.clientWidth)/2
end sub
sub bild1blend()
x=x-1
y=y+1
if x=-40 then
    x=101
    y=-1
    bild2blend
    exit sub
end if
bild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &x& ", style=0) "
bild2.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &y& ", style=0) "
setTimeout "bild1blend()", 62
end sub
sub bild2blend()
x=x-1
y=y+1
if x=-40 then
    x=101
    y=-1
    bild3blend
    exit sub
end if
bild2.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &x& ", style=0) "
bild3.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &y& ", style=0) "
setTimeout "bild2blend()", 62
end sub
sub bild3blend()
x=x-1
y=y+1
if x=-40 then
    x=101
    y=-1
    bild1blend
    exit sub
end if
bild3.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &x& ", style=0) "
bild1.style.filter="progid:DXImageTransform.Microsoft.Alpha(opacity=" &y& ", style=0) "
setTimeout "bild3blend()", 62
end sub
</script>
```

In der nächsten Lektion bespreche ich den revealTrans-Filter.



Lektion 6: der revealTrans - Filter

Der revealTrans-Filter ist ein dynamischer Filter, der verwendet wird um die Übergänge von sichtbaren zu nicht sichtbaren Elementen zu steuern. In der Regel werden das Bilder sein, die wie in einer Slideshow ein- und ausgeblendet werden. Dieser Filter ermöglicht dabei 23 verschiedene Übergänge.

Syntax des Filters:

progid:DXImageTransform.Microsoft.revealTrans(transition=x,duration=y)

Statt **y** setzen sie eine Zeit in Sekunden ein, die angibt wie lange die Ein- bzw. Ausblendung dauern soll.

Statt **x** setzen sie einen der folgenden Werte ein.

0	=	Einschließen
1	=	Ausschließen
2	=	Einkreisen
3	=	Auskreisen
4	=	Von unten sichtbar werden
5	=	Von oben sichtbar werden
6	=	Von links sichtbar werden
7	=	Von rechts sichtbar werden
8	=	Vertikales Einblenden
9	=	Horizontales Einblenden
10	=	Karomuster
11	=	Schachbrettmuster
12	=	Zufallsauflösung
13	=	Senkrechte Aufteilung nach innen
14	=	Senkrechte Aufteilung nach außen
15	=	Waagrechte Aufteilung nach innen
16	=	Waagrechte Aufteilung nach außen
17	=	Streifen nach links unten
18	=	Streifen nach links oben
19	=	Streifen nach rechts unten
20	=	Streifen nach rechts oben
21	=	Waagrechte Zufallsstreifen
22	=	Senkrechte Zufallsstreifen
23	=	Zufallseffekt aus den Vorangegangenen



VB- SCRIPTING FÜR STATIONERIES

In dieser Lektion wollen wir diesen Filter zum Einsatz bringen um 3 Bilder nacheinander anzeigen zu lassen.

Öffnen Sie den Ordner Lektion6 und laden Sie die Datei lektion6.htm in Ihren Editor.

Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 6</title>
<style type=text/css>
</style>
</head>
<body scroll=no>
<p id=tx>Der revealTrans-Filter</p>
</body>
</html>
```

Ein Paragraph ist im Dokument mit der **ID id=tx**

Betrachten Sie das Dokument kurz in der Vorschau. Das sollten Sie sowieso nach jedem Teilschritt machen.

Wie bei allen bisherigen Beispielen werde ich zuerst ein die Grundformatierung über CSS einfügen. Fügen Sie wie immer den roten Code hinzu.

```
<style type=text/css>
body { border:5 brown groove }
#tx { color:darkgreen;
font-size:36;
position:absolute;
filter:progid:DXImageTransform.Microsoft.shadow(color=black,direction=135) }
</style>
```

Der Text bekommt die Farbe dunkelgrün, Größe ist 36 Pixel, Position ist absolute und der Text wird mit einem Schattenfilter dargestellt.



VB- SCRIPTING FÜR STATIONERIES

Ich erzeuge nun einen DIV-Tag, der später als Container für meine Bilder dienen soll.
Die Bilder stelle ich gleich rein.

```
<html>
<head>
<title>Lektion 6</title>
<style type=text/css>
body { border:5 brown groove }
#tx { color:darkgreen;
font-size:36;
position:absolute;
filter:progid:DXImageTransform.Microsoft.shadow(color=black,direction=135) }
#box1 { position:absolute;
width:448;
height:283;
border:1 brown groove;
z-index:2 }
</style>
</head>
<body scroll=no background="h1.jpg">

<p id=tx>Der revealTrans-Filter</p>
<div id=box1>



</div><body scroll=no>
<p id=tx>Der revealTrans-Filter</p>
</body>
</html>
```

Folgende Formatierungen habe ich vorgenommen:

Position ist absolute, die **Breite ist 448 Pixel**, die **Höhe=283 Pixel**, der DIV-Tag bekommt einen **Rahmen** und Wichtig!!! Ich setze ihn in die zweite Ebene **z-index:2**

Der DIV-TAG bekommt die ID **id=box1**



Nun folgt ein kleiner Trick.

Ich habe eine kleine, goldfarbene, animierte Gif-Datei, erstellt mit PI und möchte nun dem DIV box1 einen Glitzerrand geben. Dazu erzeuge ich einen weiteren DIV-Tag, der etwas größer ist als der erste und lege diesen eine Ebene nach hinten. Zugleich ordne ich dem neuen DIV-Tag als Hintergrund diese Gif-Animation zu.

Die Gif-Datei stelle ich über CSS wieder 3000px nach links und in VB werde ich den Hintergrund später zuordnen.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252"> <title>Lektion 6</title>
<style type="text/css">
body {      border:5 brown groove }
#tx  {      color:darkgreen;
            font-size:36;
            position:absolute;
            filter:progid:DXImageTransform.Microsoft.shadow(color=black,direction=135) }

#box1 {     position:absolute;
            width:448;
            height:283;
            border:1 brown groove;
            z-index:2 }

#box2 {     position:absolute;
            width:458;
            height:293;
            border:1 darkgreen double;
            z-index:1 }

#bild1 {    position:absolute;
            left:-3000 }

</style>
</head>
<body scroll=no background="h1.jpg">

<p id=tx>Der revealTrans-Filter</p>
<div id=box1>
    
    
    
</div>
<div id=box2></div>
</body>
</html>
```

Was jetzt noch an Formatierung bleibt sind die drei Bilder in DIV ID=box1. Das erste Bild soll angezeigt werden, die Bilder 2 und 3 sollen erst mal unsichtbar sein.

Fügen Sie im Stylesheet folgende Anweisung hinzu:

```
#bild1      {      position:absolute;
                visibility:visible }
#bild2      {      position:absolute;
                visibility:hidden }
#bild3      {      position:absolute;
                visibility:hidden }
```

visibility:visible bedeutet: Das Objekt ist sichtbar.
Visibility:hidden bedeutet: Das Objekt ist unsichtbar.



Wie immer zuerst die Positionen beim Start und bei Änderung der Größe des Dokuments festlegen.

Der Text soll horizontal zentriert werden.

Die beiden DIV-TAGS, box1 und box2, sollen in der Mitte des Dokuments positioniert werden. Den folgenden Code wieder nach `</body>` einfügen.

```
<script language=vbscript>  
sub window_onResize()  
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2  
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2  
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2  
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2  
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2  
end sub  
sub window_onLoad()  
    box2.style.background="url(" & bld1.src & ")"  
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2  
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2  
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2  
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2  
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2  
end sub  
</script>
```

Wenn Sie das Dokument nun im Browser betrachten, werden Sie den kleinen Glitzerrand sehen können.

Nun folgt der die Programmierung des Filters:
Dazu wieder ein paar Grundlagen:

Folgender Ablauf ist vorgesehen:

Die Bilder, bild1 bis bild3 sollen abwechselnd dargestellt werden. Dabei soll der Wechsel der Bilder über den `revealTrans`-Filter mit seinen 23 Möglichkeiten durchgeführt werden.

Beim Starten des Dokuments ist bild1 schon angezeigt, die Bildfolge soll sein:

Bild2 einblenden und bild1 ausblenden, bild3 einblenden und bild2 ausblenden, bild1 einblenden bild3 ausblenden usw.

Das soll solange gehen, bis alle 23 Filtermöglichkeiten dargestellt wurden.

Anschließend soll der Filter wieder auf 0 gesetzt werden, dass alles wieder von vorne beginnt.

Um ein Bild mit dem auszublenden, muss das Bild natürlich zuerst sichtbar sein, was wir ja mit `visibility=visible` in der CSS Anweisung gemacht haben.

Der Code zum Ausblenden sieht so aus:

```
Bild1.style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=0)"  
Bild1.filters(0).apply()  
Bild1.style.visibility="hidden"  
Bild1.filters(0).play(2)
```

Die erste Zeile ist die Zuordnung des Filters zu Bild1

In der zweiten Zeile wird festgelegt, dass die Ausführung des nächsten Befehls solange verzögert wird, bis ein `play()` für den Filter erfolgt.



VB- SCRIPTING FÜR STATIONERIES

Ergänzen Sie nun Ihr VBScript mit dem roten Code und testen Sie anschließend das Dokument. Es sollte das Bild1 verschwinden, bis nur noch der Hintergrund von `DIV id=box2` zu sehen ist. Ein ziemliches Geflimmer ;-)

Hier habe ich eine weitere Möglichkeit verwendet um den Filter zu starten.

Bild1.filters(0).play(y).

Y gibt die Zeit in Sekunden an.

```
<script language=vbscript>
ausblenden1()
sub window_onResize()
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2
end sub
sub window_onLoad()
    box2.style.background="url(" & bld1.src & ")"
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2
end sub
sub ausblenden1()
    Bild1.style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=0)"
    Bild1.filters(0).apply()
    Bild1.style.visibility="hidden"
    Bild1.filters(0).play(2)
end sub
</script>
```

Um das Bild 2 einzublenden brauchen wir nichts weiter zu tun, als unsere Routine zu erweitern mit `visibility="visible"` statt wie im ersten Teil mit `visibility="hidden"`. Natürlich auf das zweite Bild angewendet.

Wir fügen also direkt in der Routine `ausblenden1` folgendes hinzu:

```
sub ausblenden1()
    Bild1.style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=0)"
    Bild1.filters(0).apply()
    Bild1.style.visibility="hidden"
    Bild1.filters(0).play(2)
    Bild2.style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=0)"
    Bild2.filters(0).apply()
    Bild2.style.visibility="visible"
    Bild2.filters(0).play(2)
end sub
```

Testen Sie das Ganze wieder und spielen Sie auch mal mit der Zeit.

Wechseln sie auch die `transition`, um zu sehen, wie sich die Transitionen auswirken.



VB- SCRIPTING FÜR STATIONERIES

Jetzt könnten wir natürlich mehrere Routinen hintereinander machen, um alle Filterübergänge für die 3 Bilder zu machen, also insgesamt 24 Routinen, aber es geht auch anders.

Wir definieren einfach Variable die wir in einer Schleife abfragen.
Sehen wir uns mal an, was wir alles haben.

1. Bilder, Anzahl der Bilder ist drei, einmal ausblenden und einmal ausblenden.
2. 24 Transitionen

Aber wie kann ich die Bilder über eine Variable ansprechen?
Dazu gibt es die Array-Funktion!

Ich kann die ID eines Objekts in einer Array-Variablen abspeichern und anschließend mit einer Zahl auf diese Variable zugreifen.

Beispiele für Arrays:

Im ersten Array speichere ich Texte, die anschließend abgefragt werden können.
be1=array("Hans","Peter","Ulrich")

Auslesen kann ich sie über einen Index, wenn nicht anders festgelegt beginnt der Index immer mit 0.

Im Beispiel oben:
msgbox(be1(0)) würde ausgeben: Hans
msgbox(be1(1)) würde ausgeben: Peter

Aber in einem Array können sie auch Zahlen oder Objekte speichern.
z.B. **be2=array(2,5,9,12)**
msgbox(be2(2)) würde die Zahl 9 ausgeben

Ach Objekte können in einem Array gespeichert werden. Dazu können Sie die ID des Objekts verwenden. In unserem Dokument also z.B.
be3=array(bild1,bild2,bild3)

Der Befehl **be3(1).style.visibility="visible"** würde in dem Beispiel das Bild2 sichtbar machen.

Wir benötigen für unsrer Aufgabe also erst mal vier Dinge.
3 Variable, eine Array für die Bilder.

Definieren wir einfach:

bi=array(bild1,bild2,bild3) → Unsere Bilder
x=0 → Bildzähler fürs Einblenden
y=-1 → Bildzähler fürs Ausblenden.
z=0 → Der Zähler für die Transitionen



Jetzt geht's los mit der Schleife.

Es beginnt damit das Bild1 ausgeblendet werden soll und Bild 2 eingeblendet werden soll. Bild1 ist mit der Variablen bi(0) ansprechbar, Bild2 mit der Variablen bi(1). Die erste Transition ist mit 0 ansprechbar.

Ist das dritte Bild angezeigt, soll es wieder mit Bild1 beginnen. Es sollen alle 23 Transitionen gezeigt werden, wenn alle gezeigt sind, soll es wieder mit Transition 0 beginnen.

```
sub anz1()  
    x=x+1  
    y=y+1  
    z=z+1  
    if z=24 then z=0  
    if x=3 then x=0  
    if y=3 then y=0  
    bi(x).style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=" &z & ")"  
    bi(x).filters(0).apply()  
    bi(x).style.visibility="visible"  
    bi(x).filters(0).play(2)  
    bi(y).style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=" &z & ")"  
    bi(y).filters(0).apply()  
    bi(y).style.visibility="hidden"  
    bi(y).filters(0).play(2)  
    setTimeout "anz1()",4000  
end sub
```

Sehen wir uns den Ablauf an:

Bevor die Routine aufgerufen wird sind:

x=0, y=-1, z=-1

x=x+1	→	x=1
y=y+1	→	y=0
z=z+1	→	z=0
if z=24 then z=0	→	Wenn z 24 erreicht, dann wäre es eins zuviel, die Transitionen gehen von 0 – 23, also beginne ich wieder bei 0.
if x=3 then x=0	→	Wenn der Bildzähler 3 erreicht ist es eins zuviel, die Bilder im Array nur von bi(0= bis bi(2), also beginne ich wieder bei 0.
If y=3 then y=0	→	Wenn der Bildzähler 3 erreicht ist es eins zuviel, die Bilder im Array nur von bi(0= bis bi(2), also beginne ich wieder bei 0.

Im ersten Durchgang gilt:

```
bi(x).style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition=" &z & ")"  
bi(x) = bi(1) = bild2  
bi(y) = bi(0) = bild1  
transition=" & z & )" = transition=0
```

Somit haben wir nur eine Routine, die durch das setTimeout() alle 4 Sekunden wiederholt wird.



VB- SCRIPTING FÜR STATIONERIES

Hier nun der gesamte VBScript Code.

```
<script language=vbscript>
bi=array(bild1,bild2,bild3)
x=0
y=-1
z=-1
window_onResize
setTimeout "anz1()",3000
sub window_onResize()
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2
end sub
sub window_onLoad()
    box2.style.background="url(" & bld1.src & ")"
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    box1.style.left=(document.body.clientWidth-box1.clientWidth)/2
    box1.style.top=(document.body.clientHeight-box1.clientHeight)/2
    box2.style.left=(document.body.clientWidth-box2.clientWidth)/2
    box2.style.top=(document.body.clientHeight-box2.clientHeight)/2
end sub
sub anz1()
    x=x+1
    y=y+1
    z=z+1
    if z=24 then z=0
    if x=3 then x=0
    if y=3 then y=0
    tx.firstChild.nodeValue="Filter Index: "&z
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    bi(x).style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition="&z&")"
    bi(x).filters(0).apply()
    bi(x).style.visibility="visible"
    bi(x).filters(0).play(2)
    bi(y).style.filter="progid:DXImageTransform.Microsoft.revealTrans(transition="&z&")"
    bi(y).filters(0).apply()
    bi(y).style.visibility="hidden"
    bi(y).filters(0).play(2)
    setTimeout "anz1()",4000
end sub
</script>
```

Versuchen Sie wieder eigene Einstellungen und eigene Bilder.



Lektion 7: der Chroma-Filter

Mit dem Chroma-Filter kann man beliebige Farben transparent machen. Dazu bieten sich Bilder an, bei denen ein Hintergrund sichtbar werden soll. Außerdem sind sie damit in der Lage, auch andere Bildformate transparent zu machen. In dieser Lektion wollen wir ein Dokument mit einem bewegten Hintergrund erstellen. Über den Hintergrund legen wir einen DIV-Tag mit der selben Größe wie das Dokument. Den DIV-Tag ordnen wir einen Chroma-Filter zu, der Grafiken oder Schriften mit einer bestimmten Farbe durchscheinen lässt. Gehen Sie zum Ordner Lektion7 und öffnen Sie die Datei Lektion7.htm mit Ihrem Editor. Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 7</title>
<style type=text/css>
</style>
</head>
<body scroll=no background="sterne.jpg">
<div id=vorne>
<p id=tx>Chroma-Filter</p>
</div>
</body>
</html>
```

Die wichtigsten Merkmale:

<body scroll=no> Keine Scrollleisten im Dokument.
Das Dokument bekommt als Hintergrundbild das Bild:sterne.jpg
Ein DIV-Objekt mit id=vorne
Im DIV-Objekt ein Absatz mit id=tx und dem Text Chroma-Filter.

Zuerst wie gewohnt das Dokument über CSS formatieren. Fügen Sie den roten teil wieder in Ihr Dokument ein:

```
<style type=text/css>
#tx { color:black;
font-size:80;
text-align:center }
</style>
```

Der Text wird schwarz, in der Schriftgröße 80pt und zentriert zum Dokument dargestellt.



Als nächstes wird der DIV-Tag formatiert:

```
<style type=text/css>
#tx {      color:black;
          font-size:80;
          text-align:center }
#vorne {  position:absolute;
          top:0;
          left:0;
          z-index:1;
          filter:progid:DXImageTransform.Microsoft.Chroma( color=#000000) }
</style>
```

Die Formatierungen im Einzelnen:

position:absolute der DIV-Tag wird absolute positioniert, d.h., die Maße müssen von Hand bestimmt werden.

top:0; left:0 der DIV-Tag wird im Dokument am oben links dargestellt.

z-index:1 der DIV-Tag wird eine Ebene über das Dokument gestellt

filter:progid:DXImageTransform.Microsoft.Chroma(color=#000000) }

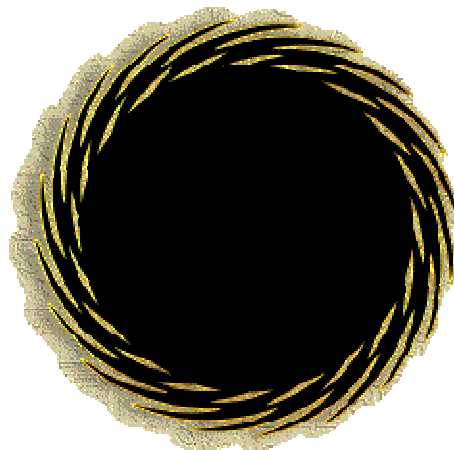
Es wird ein Chroma-Filter dem DIV-Tag zugeordnet. Die Farbe, die transparent dargestellt werden soll, ist schwarz. **color=#000000** entspricht schwarz .
Betrachten Sie das Resultat in der Vorschau und sie sehen im DIV-Tag den Text transparent dargestellt.

Nun füge ich, ebenfalls im DIV-Tag drei Bilder hinzu. Die Bilder bekommen eine ID.

```
<body scroll=no background="sterne.jpg">
<div id=vorne>
<p id=tx>Chroma-Filter</p>
<img id=s1 src=stern1.gif></img>
<img id=s2 src=stern1.gif></img>
<img id=s3 src=stern1.gif></img>
</div>
</body>
```

Die Bilder: sind als Gif-Grafiken gespeichert, der Bildinhalt ist hauptsächlich schwarz, da wir schwarz als transparente Farbe im Filter definiert haben, wird der schwarze Teil in unseren DIV-Tag transparent und der Hintergrund des Dokuments wird durch die Grafik zu sehen sein.

Wieder in der Vorschau betrachten.





VB- SCRIPTING FÜR STATIONERIES

Diese drei Bilder werde ich nun wieder in CSS formatieren.
Kurze Anmerkung warum drei Bilder und nicht Kopien, da alle drei Bilder gleich sind.
In Outlook-Express ist eine Routine, die verhindert das in einer Mail ein Bild mit dem selben namen öfters vorkommen kann. Mir ist noch nicht klar, ob das ein BUG oder beabsichtigt ist.
Deshalb habe ich den drei Bildern drei unterschiedliche Namen gegeben.

Die Formatierung:

```
<style type=text/css>
#tx { color:black;
      font-size:80;
      text-align:center }
#vorne { position:absolute;
          left:0;
          z-index:1;
          filter:progid:DXImageTransform.Microsoft.Chroma( color=#000000) }
#s1 { position:absolute;
       left:0;
       top:0 }
#s2 { position:absolute;
       right:0;
       bottom:0 }
#s3 { position:absolute }
</style>
```

Das erste Bild setze ich in die linke, obere Ecke.

Das zweite Bild setze ich in die rechte, untere Ecke.

Das dritte Bild bekommt wird noch nicht richtig positioniert, das geschieht später im Script.
Damit sind wir auch schon mit dem Dokument soweit fertig.

Es folgt das Script.

Als erstes werde ich das dritte Bild in die Mitte des Dokuments setzen.
Dann werde ich den Hintergrund des Dokuments, nicht des DIV-Tags animieren, wie in Lektion1 gezeigt.

Der gesamte Quellcode ist auf der nächsten Seite.
Versuchen Sie selbst, das Script zu lesen und zu verstehen, es ist nichts Neues dabei



VB- SCRIPTING FÜR STATIONERIES

```
<html>
<head>
<title>Lektion 7</title>
<style type=text/css>
#tx          {      color:black;
                  font-size:80;
                  text-align:center }
#vorne       {      position:absolute;
                  top:0;
                  left:0;
                  z-index:1;
                  filter:progid:DXImageTransform.Microsoft.Chroma(color=#000000) }
#s1          {      position:absolute; left:0; top:0 }
#s2          {      position:absolute; right:0; bottom:0 }
#s3          {      position:absolute }
#bld1        {      position:absolute; left:-3000 } <!--Hintergrund DIV ->

</style>
</head>
<body scroll=no background="sterne.jpg">

<div id=vorne>
<p id=tx> Chroma-Filter</p>
</img>
</img>
</img>
</div>
</body>
<script language=vbscript>
x=0
lauf
sub window_onLoad()
vorne.style.background="url(" & bld1.src & ")"
vorne.style.width=document.body.clientWidth
vorne.style.height=document.body.clientHeight
s3.style.left=(document.body.clientWidth-s3.clientWidth)/2
s3.style.top=(document.body.clientHeight-s3.clientHeight)/2
end sub
Sub window_onResize()
vorne.style.background="url(" & bld1.src & ")"
vorne.style.width=document.body.clientWidth
vorne.style.height=document.body.clientHeight
s3.style.left=(document.body.clientWidth-s3.clientWidth)/2
s3.style.top=(document.body.clientHeight-s3.clientHeight)/2
end sub
sub lauf()
x=x+1
document.body.style.backgroundColor=x
setTimeout "lauf()",32
end sub
</script>
</html>
```



Lektion 8: Meta Creations Filter

Diese 3D-Filter der Fa. Meta Creations funktionieren ähnlich wie der revealTrans-Filter, den wir schon besprochen haben.

Es gibt momentan 21 Filterarten, die mit play() gestartet werden können.

Es gibt allerdings einen Unterschied: wenn Sie mit Meta Creation Filtern arbeiten, müssen Sie ein Copyright mit angeben. Die Filter wurden von Microsoft für DirectX erworben und können auf Webseiten oder Stationeries eingesetzt werden.

In der fertigen Version der Lektion8 habe ich Ihnen ein vollständiges Script eingefügt, das alle Filter anzeigt. Hier in der Lektionsaufgabe werde ich nur einen einbinden, da sonst zu Quellcode überlang werden müsste.

Wir werden zuerst die Lektion durcharbeiten, anschließend werde ich Ihnen die Filter mit dem Copyright vorstellen.

Öffnen Sie die Datei Lektion8,htm in Ihrem Editor.

Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html charset=windows-1252">
<title>Lektion 8</title>
<style type=text/css>
</style>
</head>
<body scroll=no background="h1.jpg">
<p id=tx>Mein Photoalbum</p>
<div id=container>
<img id=bild1 src=b1.jpg></img>
<img id=bild2 src=b2.jpg></img>
<img id=bild3 src=b3.jpg></img>
<img id=bild4 src=b4.jpg></img>
<img id=bild5 src=b5.jpg></img>
</div>
</body>
</html>
```

Kurze Erläuterung.

Ein Absatz mit id=tx.

Ein DIV-Tag mit der id=container.

Im DIV-Tag 5 Bilder mit dem id's bild1 bis bild5.



Wie immer zuerst die Formatierungen:

```
<style type=text/css>
#tx { color:white;
      font-size:40;
      position:absolute;
      top:0;
      filter:progid:DXImageTransform.Microsoft.shadow(color=black,direction=135) }

#bild1 { position:absolute;
          visibility:visiblle }
#bild2 { position:absolute;
          visibility:hidden }
#bild3 { position:absolute;
          visibility:hidden }
#bild4 { position:absolute;
          visibility:hidden }
#bild5 { position:absolute;
          visibility:hidden }

#container { position:absolute;
              height:492;
              width:652;
              bottom:20;
              border:6 darkgreen groove }
</style>
```

Der Text in `p=tx` wird wieder mit dem Shadowfilter hinterlegt.

Das erste Bild ist sichtbar, die 4 anderen sind unsichtbar.

Der Container hat genaue Maße, die selben Maße wie die Bilder, alle Bilder sind gleich groß.

Es folgt nun der Scriptteil für einen Wechsel.

In der Variablen `blende1` steht die Definition des Filters, hier ein Ripple-Effekt.

```
<script language=vbscript>
blende1="progid:DXImageTransform.MetaCreations.Ripple"
blende2="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D03-06A3-11D2-8F98-00C04FB92EB7}"
x=-1
y=0
bilder=array(bild1,bild2,bild3,bild4,bild5)
window_onResize
setTimeout "anzeigen()",3000
sub window_onLoad()
tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
container.style.left=(document.body.clientWidth-container.clientWidth)/2
end sub
sub window_onResize()
tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
container.style.left=(document.body.clientWidth-container.clientWidth)/2
end sub
sub anzeigen()
    x=x+1
    y=y+1
    if x=5 then x=0
    if y=5 then y=0
    tx.firstChild.nodeValue=blende1
    tx.style.left=(document.body.clientWidth-tx.clientWidth)/2
    bilder(x).style.filter=blende1
    bilder(x).filters(0).Copyright =blende2
    bilder(x).filters(0).Apply()
    bilder(x).style.visibility="hidden"
    bilder(x).filters(0).Play(6.000)
    bilder(y).style.filter=blende1
    bilder(y).filters(0).Copyright =blende2
    bilder(y).filters(0).Apply()
    bilder(y).style.visibility="visible"
    bilder(y).filters(0).Play(6.000)
    setTimeout "anzeigen()",8000
end sub
</script>
```



Erklärung des Codes:

blende1="progid:DXImageTransform.MetaCreations.Ripple"

Etwas anders als bei den bisherigen beispielen, speichere ich diesmal den Filtertext zuerst in eine Variable, die ich blende1 nenne. Als Filter nehme ich den Filter Ripple .

Zu jedem Filter gehört ein einziger Copyrighthinweis, der als kompletter Text angegeben werden muss. Ich stelle zum Schluss der Lektion eine komplette Übersicht ins Dokument.

blende2="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D03-06A3-11D2-8F98-00C04FB92EB7}"

Das ist der dazugehörige Teil für den Ripple Filter.

Alle Leerzeilen und Abstände müssen dabei stimmen.

Auch hier habe ich den Filter-Copyright Text einer Variablen zugeordnet.

Variablenname: **blende2**

tx.firstChild.nodeValue=blende1

Der Text oben wird ersetzt durch den Inhalt der Variablen **blende1**

Der weitere Programmablauf entspricht genau dem Ablauf, wie ich ihn beim revealTrans-Filter gezeigt habe.

Eine Ausnahme:

Das Copyright muss ebenfalls angegeben werden.

bilder(x).filters(0).Copyright =blende2

Mit dieser Lektion verlassen wir die Filter vorläufig.

In den Lektionen 11 - 20 werden Sie noch weitere Filter kennenlernen.

In den nächsten beiden Lektionen werden wir die Grundlagen zur Animation erarbeiten.

Auf der nächsten Seite finden Sie eine komplette Übersicht über die Metacreations Filter.



Filter1:= "progid:DXImageTransform.MetaCreations.GlassBlock"
Filter2:= "progid:DXImageTransform.MetaCreations.Liquid"
Filter3:= "progid:DXImageTransform.MetaCreations.Twister"
Filter4:= "progid:DXImageTransform.MetaCreations.CenterCurls"
Filter5:= "progid:DXImageTransform.MetaCreations.PageCurl"
Filter6:= "progid:DXImageTransform.MetaCreations.Water"
Filter7:= "progid:DXImageTransform.MetaCreations.LightWipe"
Filter8:= "progid:DXImageTransform.MetaCreations.RollDown"
Filter9:= "progid:DXImageTransform.MetaCreations.Wormhole"
Filter10:= "progid:DXImageTransform.MetaCreations.Lens"
Filter11:= "progid:DXImageTransform.MetaCreations.FadeWhite"
Filter12:= "progid:DXImageTransform.MetaCreations.Jaws"
Filter13:= "progid:DXImageTransform.MetaCreations.FlowMotion"
Filter14:= "progid:DXImageTransform.MetaCreations.Vacuum"
Filter15:= "progid:DXImageTransform.MetaCreations.Grid"
Filter16:= "progid:DXImageTransform.MetaCreations.Threshold"
Filter17:= "progid:DXImageTransform.MetaCreations.Ripple"
Filter18:= "progid:DXImageTransform.MetaCreations.Curls"
Filter19:= "progid:DXImageTransform.MetaCreations.PeelABCD"
Filter20:= "progid:DXImageTransform.MetaCreations.Curtains"
Filter21:= "progid:DXImageTransform.MetaCreations.BurnFilm"
Filter22:= "progid:DXImageTransform.MetaCreations.ColorFade"

FilterCopyright1:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C913-07AA-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright2:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D0A-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright3:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045CF-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright4:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D0C-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright5:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D08-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright6:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045C5-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright7:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045C8-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright8:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {9C61F46E-0530-11D2-8F98-00C04FB92EB7}"

FilterCopyright9:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {0E6AE022-0C83-11D2-8CD4-00104BC75D9A}"

FilterCopyright10:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045CA-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright11:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045CC-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright12:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C904-07AA-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright13:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C90B-07AA-11D2-8D6D-00C04F8EF8E0}"



VB- SCRIPTING FÜR STATIONERIES

FilterCopyright14:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C90D-07AA-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright15:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C911-07AA-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright16:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C915-07AA-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright17:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D03-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright18:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D0E-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright19:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D10-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright20:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {AA0D4D12-06A3-11D2-8F98-00C04FB92EB7}"

FilterCopyright21:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {107045D0-06E0-11D2-8D6D-00C04F8EF8E0}"

FilterCopyright22:="Copyright MetaCreations Corp. 1998. Unauthorized duplication of this string is illegal. {2A54C908-07AA-11D2-8D6D-00C04F8EF8E0}"



Lektion 9 - Bewegte Grafiken Einführung

Mit dieser Lektion werde ich Ihnen die Grundlagen der Animation von Objekten näher bringen. Zuerst mal etwas einfaches, bevor wir in Lektion 10 etwas tiefer einsteigen. Öffnen Sie die Datei Lektion9.htm. Folgender Code ist zu sehen:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252" ><title>Lektion
9</title>
<style type=text/css>

</style>
</head>
<body scroll=no background="h1.jpg">
<div id=tx>
    Wenn unser Herz<br>aufbricht<br>wenn das Eis<br>
    die Klammer löst<br>werden wir wieder sehen<br>hören, zuhören, fühlen<br><br>
    dann<br>werden wir nicht mehr<br>allein sein
</div>
</body>
</html>
```

Es ist ein DIV-Tag definiert, mit `id=tx`.
Im Div-Tag ist ein Text dargestellt.
Nun zuerst wieder die Grundformatierungen über CSS.

```
<style type=text/css>
Body {      border:5 darkgreen groove }
#tx  {      position:absolute;
            right:100;
            top:100;
            width:400;
            height:400;
            color:darkgreen;
            font-size:30;
            text-align:center;
            filter:progid:DXImageTransform.Microsoft.Shadow(c olor=black,direction=135);
            scrollbar:auto }

</style>
```

Zum `<body>` ist nicht viel zu sagen. Er bekommt einen Rahmen.
er DIV-Tag mit dem Text wird wie folgt formatiert:
Die Position ist absolut.
Er ist 100 Pixel vom rechten Rand und 100 Pixel von oben entfernt. Die Breite und Höhe beträgt 400 Pixel.
Die Schriftfarbe des Textes im DIV ist dunkelgrün, die Schriftgröße 30pt, der Text wird im DIV zentriert ausgerichtet.
Der Text bekommt einen Schattenfilter.
Sollte der Text die Maße des DIV überschreiten, dann werden Scrollbars automatisch am DIV eingeblendet.

scrollbar:auto



VB- SCRIPTING FÜR STATIONERIES

Nun füge ich ein Bild hinzu, eine animierte GIF-Datei, die ich mit PhotoImpact erstellt habe. Dieser Grafik ordne ich einen Alpha-Filter hinzu, so dass die Grafik mit dem Hintergrund leicht verschmilzt.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252"> <title>Lektion 9</title>
<style type=text/css>
body {      border:5 darkgreen groove }
#tx  {      position:absolute;
            right:100;
            top:100;
            width:400;
            height:400;
            color:darkgreen;
            font-size:30;
            text-align:center;
            filter:progid:DXImageTransform.Microsoft.Shadow(c olor=black,direction=135);
            scrollbar:auto }
#bild1 {    position:absolute;
            left:-100;
            bottom:0;
            filter:progid:DXImageTransform.Microsoft.Alpha(op acity=100,style=2) }
</style>
</head>
<body scroll=no background="h1.jpg">
<img id=bild1 src=water.gif></img>
<div id=tx>
    Wenn unser Herz<br>aufbricht<br>wenn das Eis<br>
    die Klammer löst<br>werden wir wieder sehen<br>hören, zuhören, fühlen<br><br>
    dann<br>werden wir nicht mehr<br>allein sein
</div>
</body>
</html>
```

Wie Sie sehen, habe ich den Position links auf -100 Pixel gesetzt. das bedeutet, dass das Bild 100 Pixel über den linken Rand verschoben wird, also nicht mehr ganz sichtbar ist.

Ich füge nun ein weiteres GIF-Bild ein, einen animierten Vogel.

```
#bird {      position:absolute;
            left:0;
            filter:progid:DXImageTransform.Microsoft.Alpha(op acity=100,style=2) }
</style>
<body scroll=no>
<img id=bird src=bird.gif></img>
<img id=bild1 src=water.gif></img>
```

Dem Vogel habe ich ebenfalls einen Alpha-Filter zugeordnet, da er so besser mit dem Hintergrund harmonisiert.

Jetzt gilt es als nächstes, den Vogel in Bewegung zu setzten.



VB- SCRIPTING FÜR STATIONERIES

Das Script wie immer nach `</body>` einfügen.

```
<script language=vbscript>  
x=0  
flieg()  
sub flieg()  
    x=x+2  
    if x>document.body.clientWidth then x=0  
    bird.style.left=x  
    setTimeout "flieg()",16  
end sub  
</script>
```

Mit x bestimme ich die position meines Vogels.

Ich definiere eine Variable x mit dem Wert 0.

x=0

Anschließend springe ich zur Subroutine flieg()

flieg()

In der Subroutine addiere ich 2 zu x.

Da x vorher 0 war, ist x jetzt im ersten Durchgang 2.

x=x+2

Jetzt frage ich ab, ob x größer ist als die Breite meines Dokuments, wenn ja, soll x wieder zu 0 werden.

Setze ich x nicht wieder zurück, dann würde der Vogel logischerweise nach rechts aus dem Dokument fliegen und nie wieder auftauchen ;-)

if x>document.body.clientWidth then x=0

Nun setze ich die linke Position des Vogels neu, 2 Pixel weiter vom linken Rand entfernt als die letzte Position war, der Vogel bewegt sich also nach rechts.

bird.style.left=x

Nun noch eine Zeitschleife, die alle 16'tausenstel Sekunde diese Subroutine wiederholt.

Settimeout "flieg()",16

Ja, das war's schon!

Durch die Positionen eines Objekts kann ich das Objekt in alle Richtungen bewegen.

Die wichtigen Punkte dabei sind:

- Left** - Eine negativer Wert bewegt nach links, ein positiver Wert nach rechts
- Right** - Ein negativer Wert bewegt nach rechts, ein positiver Wert bewegt nach links
- Top** - Ein negativer Wert bewegt nach oben, ein positiver Wert bewegt nach unten
- Bottom** - Ein negativer Wert bewegt nach unten, ein positiver Wert bewegt nach oben



Lektion 10: ein etwas komplexeres Animationsprogramm für Objekte

In dieser Lektion möchte ich mehrere grafische Objekte über zufällig erzeugte Werte in Bewegung setzen. Öffnen Sie die Datei Lektion10.htm in Ihrem Editor.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Lektion 10</title>
<style type=text/css>

</style>
</head>
<body scroll=no background="h1.jpg">
</body>
</html>
```

Da in diesem Dokument sehr viel Stylesheet-Anweisungen gebraucht werden und dabei etwas in die Trickkiste gegriffen wird, gehe ich langsam, Schritt für Schritt vor.

Ich möchte über CSS und DIV-Tags Linien zeichnen, die goldig glitzern sollen. ;-)
Vier Linien oben links, nach unten versetzt, vier Linien unten rechts, nach oben versetzt.



Auf der nächsten Seite zeige ich Ihnen den gesamten HTML-Teil und die dazugehörigen Formatierungen.



```
<style type=text/css>
div.linie1      {      position:absolute;
                    font-size:5;
                    left:10;
                    top:20;
                    width:500;
                    border-top:1 black double;
                    border-bottom: 1 black double }

div.linie2      {      position:absolute;
                    font-size:5;
                    bottom:20;
                    width:500;
                    right:10;
                    border-top:1 black double;
                    border-bottom: 1 black double }

#bild1          {      position:absolute;left:-30 }
#tx1            {      position:absolute;top:20;right:40 }
#note1          {      position:absolute;left:150;visibility:hidden }
#note2          {      position:absolute;left:150;visibility:hidden }
#note3          {      position:absolute;left:150;visibility:hidden }
#note4          {      position:absolute;left:150;visibility:hidden }
#note5          {      position:absolute;left:150;visibility:hidden }
#bld1           {      position:absolute;left:-3000 }
</style>
</head>
<body scroll=no background="h1.jpg">

<div class=linie1 id=l0></div>
<div class=linie1 id=l1 style="top:30;width:400"></div>
<div class=linie1 id=l2 style="top:40;width:300"></div>
<div class=linie1 id=l3 style="top:50;width:200"></div>
<div class=linie2 id=l4 ></div>
<div class=linie2 id=l5 style="bottom:30;width:400"></div>
<div class=linie2 id=l6 style="bottom:40;width:300"></div>
<div class=linie2 id=l7 style="bottom:50;width:200"></div>
</img>
</img>





</body>
```

Fangen wir mit dem HTML-Teil an:

Das Hintergrundscrolling ist blockiert, als Hintergrundgrafik ist [h1.jpg](#) bestimmt.

Als nächstes folgt die Hintergrundgrafik für die 8 Balken, [gold.gif](#).

Nun werden die 8 Balken als `<div>`'s definiert. Dabei ist zu beachten, ich spreche sie als Klasse an `class=linie1` bzw. `linie2`, und sie bekommen zusätzlich eine ID.

Warum? Weil ich die Formatierungen für jeweils 4 `<div>`'s zusammenfasse, aber doch jeden DIV über eine eigene ID ansprechen muss.

Es folgt das Bild der Blume, [b1.jpg](#).

Es folgt das Bild mit dem Text: Happy Birthday, [text.gif](#).

Anschließend fünf animierte Bilder mit Musiknoten, [note1.gif](#) – [note5.gif](#)



Sehen wir uns nun die Formatierungen an.

```
div.linie1 { position:absolute;
             font-size:5;
             left:10;
             top:20;
             width:500;
             border-top:1 black double;
             border-bottom: 1 black double }
div.linie2 { position:absolute;
             font-size:5;
             bottom:20;
             width:500;
             right:10;
             border-top:1 black double;
             border-bottom: 1 black double }
```

Die beiden div's, **Linie1** und **Linie2** gebe ich diesmal als **class** an, nicht als **id**.

Also: statt **#linie1** sage ich: **div.linie1**.

Die Position ist absolut, wie gewohnt

Jetzt kommt der erste Trick:

normalerweise gebe ich die Größe und Breite eines DIV-Tags über die Style-Befehle **width** und **height** an. Nun ist es leider so, das ein DIV in der Höhe nur so klein gemacht werden kann, wie die Schriftgröße in diesem DIV ist. Wird keine Schriftgröße angegeben, dann ist es die Einstellung der Standardschriftart. Damit ich also die Höhe noch kleiner bekomme, setze ich hier die Schriftgröße auf 5pt.

font-size:5;

Die nächsten Maße wie gewohnt:

10 Punkte vom linken Rand, 20 Pixel von oben, Breite ist 500 Pixel.

Für die Linie2 gilt das Selbe, nur das hier nicht von oben sondern von unten und von rechts gemessen wird.

Jetzt möchte ich in meinem Dokument aber jeweils 4 Linien, die erstens versetzt sind und zweitens kürzer werden.

Deshalb formatiere ich jeweils 3 div's meiner Linien im HTML-Teil.

```
<div class=linie1 id=10></div>
<div class=linie1 id=11 style="top:30;width:400"></div>
<div class=linie1 id=12 style="top:40;width:300"></div>
<div class=linie1 id=13 style="top:50;width:200"></div>
<div class=linie2 id=14 ></div>
<div class=linie2 id=15 style="bottom:30;width:400"></div>
<div class=linie2 id=16 style="bottom:40;width:300"></div>
<div class=linie2 id=17 style="bottom:50;width:200"></div>
```

Im Eröffnungstag eines HTML-Objekts kann ich ebenfalls Stylesheet-Anweisungen verwenden. Das sollte zwar nicht oft gemacht werden, das es unübersichtlich ist, aber es ist jederzeit erlaubt.

Da ein Dokument beim Laden in den Browser sequentiell von oben nach unten abgearbeitet wird, werden Die Styleanweisungen von oben überschrieben und meine neuen Anweisungen in TAG gültig.



VB- SCRIPTING FÜR STATIONERIES

```
#bild1      {      position:absolute;left:-30 }
#tx1       {      position:absolute;top:20;right:40 }
#note1     {      position:absolute;left:150;visibility:hidden }
#note2     {      position:absolute;left:150;visibility:hidden }
#note3     {      position:absolute;left:150;visibility:hidden }
#note4     {      position:absolute;left:150;visibility:hidden }
#note5     {      position:absolute;left:150;visibility:hidden }
#bild1     {      position:absolute;left:-3000 }
```

Bild1 ist absolute positioniert, 30Pixel nach links über den Bildschirmrand verschoben.
Das Bild mit dem Text ist ebenfalls absolut positioniert, 20 Pixel von oben und 40 Pixel vom rechten Rand entfernt.

Die 5 Bilder mit den animierten Noten sind erst mal unsichtbar, 150 Pixel vom linken Rand entfernt.

Das Bild mit dem goldenen Glitzer, das als Hintergrund für die Linien dienen soll, ist erst mal ganz weit nach links verschoben, damit es nicht sichtbar ist.

Bevor wir uns nun dem Script zuwenden, erst ein paar Grundlagen. Zuerst das Erzeugen von Zufallszahlen.

Eine Zufallszahl wird erzeugt über den Befehl **rnd**.

Allerdings wird beim Start des Programmes immer die selbe Zufallsszahlen-Reihenfolge erzeugt. Um das zu ändern, muss am Anfang des Programms noch der Befehl **randomize** eingefügt werden.

Möchte ich jetzt eine Zufallszahl zwischen 1 und 49 erzeugen, Lotto lässt grüßen ;-)) Dann würde die Formel so aussehen: **zahl=int((rnd*49)+1)**

Erklärung:

zahl ist eine frei definierbare variable, könnte auch x,y,z oder Hans oder Eimer sein, wie Sie wollen.

zahl=int() Das int steht für Integer und bedeutet, es soll als Ergebnis eine Zahl ohne Kommastellen stehen. Die rnd-Funktion gibt einen Wert zurück, der kleiner als 1, aber größer als oder gleich Null ist.

Nehmen wir an, die Zufallszahl durch rnd wäre 0,7453645372888746352. Dann ist der Wert **rnd*49 = 36,5228623271548571248**

Das Ganze +1 = **37,5228623271548571248**

Davon nur die Ganzzahl durch int = **37**

Wir hätten also die Zufallszahl 37.



Als nächstes ein klein wenig über Schleifen.

Wenn wir eine bestimmte Folge von Befehlen öfters nacheinander ausführen möchten, dann können dazu Schleifen verwendet werden. Eine Schleife ist die FOR-Schleife.
Beispiel:

```
x=0
For i=1 to 10
    x=x+1
next
msgbox "x hat nach Durchlaufen der Schleife den Wert: " &x
```

for i=1 to 10 bedeutet:

Wiederhole Anweisungen die zwischen for und next stehen solange, bis i den Wert 10 erreicht hat. Bei jeder Wiederholung wird i automatisch um 1 erhöht.

Die Anweisung x=x+1 wird also 10mal ausgeführt, anschließend wird die Schleife verlassen und der Befehl nach der next-Anweisung ausgeführt.

msgbox bedeutet, erzeuge eine kleine Meldungsbox und gebe in dieser Box etwas aus, im Beispiel oben den Wert von x.

Geben Sie dieses Beispiel mal ein und betrachten Sie das Ergebnis in der Vorschau:

```
<script language=vbscript>
x=0
For i=1 to 10
    x=x+1
next
msgbox "x hat nach Durchlaufen der Schleife den Wert: " &x
</script>
```

Ein weiteres Beispiel diesmal zum Erzeugen von einer Zufallszahl: Ersetzen Sie den Code von vorher mit folgendem:

```
<script language=vbscript>
ausgabe="Die Reihenfolge ist: "
for i=1 to 6
zahl=int((rnd*49)+1)
ausgabe=ausgabe & zahl & ","
next
msgbox ausgabe
</script>
```

Es werden 6 Zufallszahlen zwischen 1 und 49 ausgegeben.



VB- SCRIPTING FÜR STATIONERIES

```
<script language=vbscript>
randomize
dim bitop,y
set bo=document.body
bi=array(note1,note2,note3,note4,note5)
bileft=array(150,150,150,150,150)
lin=array(10,11,12,13,14,15,16,17)
window_onResize
lauf()
sub window_onResize()
    bild1.style.top=(bo.clientHeight-bild1.clientHeight)/2
    y=(bo.clientHeight-note1.clientHeight)/2
    bitop=array(y,y,y,y,y)
    for i=0 to 4
        bi(i).style.top=y
        bi(i).style.visibility="visible"
    next
end sub
sub window_onLoad()
    for i=0 to 7
        lin(i).style.background="url(" & bld1.src & ")"
    next
    bild1.style.top=(bo.clientHeight-bild1.clientHeight)/2
    y=(bo.clientHeight-note1.clientHeight)/2
    bitop=array(y,y,y,y,y)
    for i=0 to 4
        bi(i).style.top=y
        bi(i).style.visibility="visible"
    next
end sub
sub window_onResize()
    for i=0 to 7
        lin(i).style.background="url(" & bld1.src & ")"
    next
    bild1.style.top=(bo.clientHeight-bild1.clientHeight)/2
    y=(bo.clientHeight-note1.clientHeight)/2
    bitop=array(y,y,y,y,y)
    for i=0 to 4
        bi(i).style.top=y
        bi(i).style.visibility="visible"
    next
end sub
sub lauf
    bileft(0)=bileft(0)+int(rnd*2)
    bitop(0)=bitop(0)-int(rnd*3)
    bileft(1)=bileft(1)+int(rnd*4)
    bitop(1)=bitop(1)+int(rnd*5)
    bileft(2)=bileft(2)+int(rnd*6)
    bitop(2)=bitop(2)-int(rnd*4)
    bileft(3)=bileft(3)+int(rnd*5)
    bitop(3)=bitop(3)-int(rnd*2)
    bileft(4)=bileft(4)+int(rnd*5)
    bitop(4)=bitop(4)+int(rnd*1)
    for i=0 to 4
        if (bileft(i)>bo.clientWidth) or (bitop(i)>bo.clientHeight) or (bitop(i)<0) then
            bitop(i)=y
            bileft(i)=150
        end if
        bi(i).style.left=bileft(i)
        bi(i).style.top=bitop(i)
    next
    settimeout "lauf()",64
end sub
</script>
```



Erklärung des Codes:

randomize Es wird eine zufällige Reihenfolge der Zufallszahlen erzeugt.

dim bitop,y Zwei Variable, bitop und y, werden definiert, allerdings noch ohne Werte

set bo=document.body Eine variable mit dem Namen bo wird erzeugt, die als Inhalt das Objekt document.body beinhaltet, also den gesamten Teil des Dokumentes zwischen `<body>` und `</body>`

bi=array(note1,note2,note3,note4,note5)

bileft=array(150,150,150,150,150)

lin=array(l0,l1,l2,l3,l4,l5,l6,l7)

Ich erzeuge zwei Arrays. Im ersten Array speichere ich die 5 Bilder der Noten ab, im zweiten Array die 5 Startpositionen von links der einzelnen Noten.

Das 3. Array, lin, spricht die div's mit den Linien an.

Als nächstes folgt die Subroutine zur Position und Größenberechnung beim Start und bei Änderung der Fenstergröße:

```
sub window_onLoad()  
  for i=0 to 7  
    lin(i).style.background="url(" & bld1.src & ")"  
  next  
  bild1.style.top=(bo.clientHeight-bild1.clientHeight)/2  
  y=(bo.clientHeight-note1.clientHeight)/2  
  bitop=array(y,y,y,y,y)  
  for i=0 to 4  
    bi(i).style.top=y  
    bi(i).style.visibility="visible"  
  next  
end sub
```

lin(i).style.background="url(" & bld1.src & ")"

Die 8 Linien bekommen den goldenen Hintergrund.

y=(bo.clientHeight-note1.clientHeight)/2

in der Variablen y speichere ich die vertikale Position der Noten, die beim Start vertikal zentriert dargestellt werden sollen.

bitop=array(y,y,y,y,y)

Diese Positionen speichere ich nun in einem Array.

```
for i=0 to 4  
  bi(i).style.top=y  
  bi(i).style.visibility="visible"  
next
```

Über eine Schleife, die 5 mal durchlaufen wird, 0,1,2,3,4 ...positioniere ich nun die 5 Noten und mache sie sichtbar.



Nun kommt die Animation:

```
sub lauf
byleft(0)=byleft(0)+int(rnd*2)
bitop(0)=bitop(0)-int(rnd*3)
byleft(1)=byleft(1)+int(rnd*4)
bitop(1)=bitop(1)+int(rnd*5)
byleft(2)=byleft(2)+int(rnd*6)
bitop(2)=bitop(2)-int(rnd*4)
byleft(3)=byleft(3)+int(rnd*5)
bitop(3)=bitop(3)-int(rnd*2)
byleft(4)=byleft(4)+int(rnd*5)
bitop(4)=bitop(4)+int(rnd*1)
for i=0 to 4
if (byleft(i)>bo.clientwidth) or (bitop(i)>bo.cli entheight) or (bitop(i)<0) then
    bitop(i)=y
byleft(i)=150
end if
bi(i).style.left=byleft(i)
bi(i).style.top=bitop(i)
next
settimeout "lauf()",64
end sub
```

Für jede der 5 Noten berechne ich nun über Zufallszahlen die neue Position:

```
byleft(0)=byleft(0)+int(rnd*2)
bitop(0)=bitop(0)-int(rnd*3)
```

die neue linke Position der Note1 ist: die alte Position+der Zufallszahl zwischen 0,1 oder 2
die neue obere Position der Note1 ist: die alte Position+der Zufallszahl zwischen 0,1,2 oder 3
Das führe ich nun für jede einzelne Note aus.

```
for i=0 to 4
if (byleft(i)>bo.clientwidth) or (bitop(i)>bo.clie ntheight) or (bitop(i)<0) then
    bitop(i)=y
byleft(i)=150
end if
bi(i).style.left=byleft(i)
bi(i).style.top=bitop(i)
next
```

In dieser Schleife speichere ich die neuen Positionen ab und berechne, ob eine Note das Fenster verlassen hat.

```
if (byleft(i)>bo.clientwidth) or (bitop(i)>bo.clie ntheight) or (bitop(i)<0) then
```

Falls der linke Abstand der Note größer als die fensterbreite ist, oder der Abstand von oben größer als die Fenstergröße ist, oder der Abstand von oben kleiner als Null ist, dann würde die Note verschwinden, deshalb:

```
bitop(i)=y    Die neue Position oben ist wieder y, als die vertikale Mitte
byleft(i)=150 Die neue linke Position ist wieder 150, also die Startposition der Note.
```



bi(i).style.left=bileft(i) Die neue linke Position wird gespeichert bi(i).style.top=bitop(i)
Die neue obere Position wird gespeichert.

setTimeout "lauf()",64

Die gesamte Routine wird nach 64'tausenstel sekunden neu aufgerufen.

So, das war es.

Im fertigen Beispiel habe ich noch eine Textbox integriert, in der später der Anwender seinen E-Mailtext eingeben kann.

Eine kleine Vorschau auf die Lektionen 11 - 20.

Dort werden neue Filter besprochen,

wir werden uns mit HTML+TIME beschäftigen, eine andere Methode um Animationen zu erzeugen und wir werden etwas VLM benutzen, 3-D Animationen direkt über DirectX.

Mich würde es freuen wenn ich Rückmeldungen bekomme, die mir zeigen ob etwas geändert werden soll, ob es verständlich war, ob Sie mit den Unterlagen und den Übungen klar kommen.

Selbstverständlich freue ich mich auch über Lob ! ;-))