

**Ziel des Tutorials:**

Am Ende dieses Tutorials solltet Ihr in der Lage sein, ein einfaches Frameset zu gestalten, und dieses über eine erstellte Swish-Navigation zu steuern.

**Anwendung:**

Einige Web-Designer nutzen diese Form der Architektur einer Webpräsenz. Die Vorteile liegen unter anderem darin, dass die Mischung zwischen php-, HTML-, und Flashtechnologie sehr variabel nutzbar ist. So können im Hauptfenster, das über die Navigation gesteuert wird, beliebige Inhalte aufgerufen werden, ganz gleich, ob es HTML-Dateien, dynamische php-Dateien (z.B. in Kombination mit MySQL-Datenbanken), Bilder, Filme, und so weiter und so weiter. Die Möglichkeiten sind extrem vielfältig.

Hallo und herzlich willkommen zum

**Tutorial „Steuerung von Frameset mittels Swish-Navigation.“**

Am Ende dieses Tutorials solltet Ihr in der Lage sein, ein einfaches Frameset zu gestalten, und dieses über eine erstellte Swish-Navigation zu steuern.

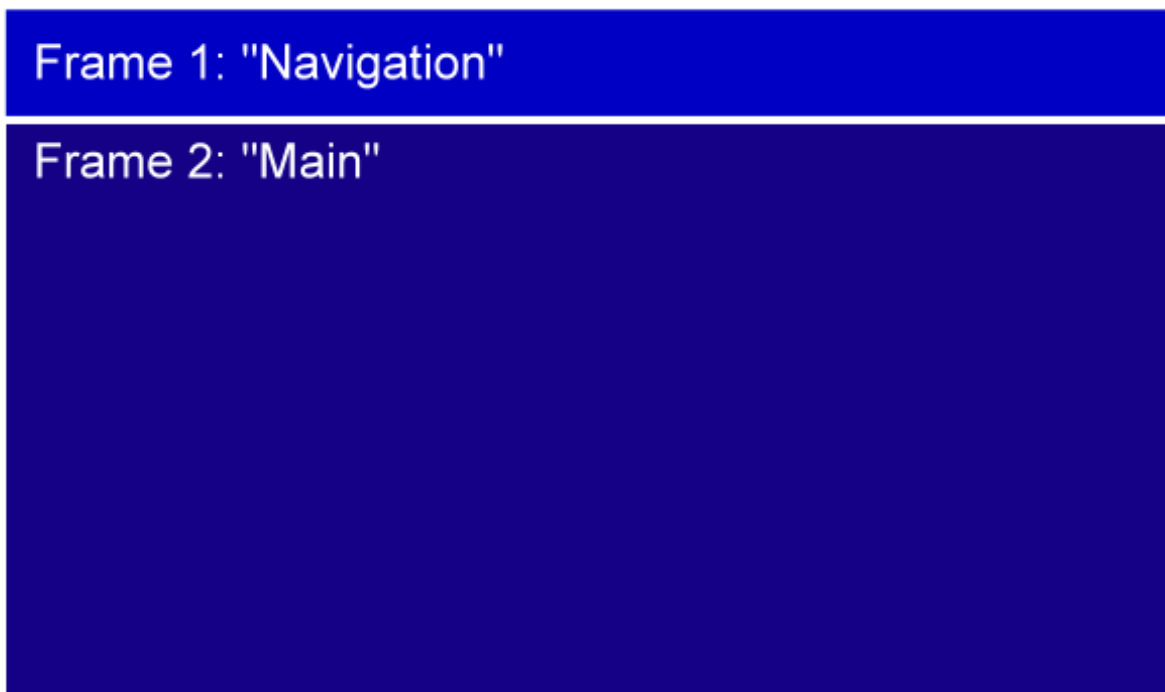
Zunächst einmal ein kurzer Überblick, was wir im Folgenden machen:

- 1) Wir erstellen ein Frameset in HTML, benennen die Frames und speichern es ab.
- 2) Wir erstellen eine kleine Navigation in Swish, die in der Lage ist, den Hauptframe zu steuern

**ERSTELLEN EINES FRAMESETS**

Zunächst einmal muss natürlich erklärt sein, was ein Frameset überhaupt ist. Ein Frame-Set unterteilt ein Fenster im Webbrowser in verschiedene Bereiche, die alle einen eigenen Namen und einen eigenen Inhalt haben können.

**In unserem Beispiel teilen wir den Bildschirm wie folgt:**



Wir bauen also 2 Frames, einen mit dem Namen „Navigation“, der später die Swish-Navigation als Inhalt erhält, und einen zweiten namens „Main“, der den wechselnden Hauptinhalt anzeigen wird, gesteuert über die Navigation.

Dazu brauchen wir einen HTML-Editor, da die zu erzeugende Datei eine HTML-Datei sein wird. Wer noch keinen hat, kann sich, z.B. über [www.filepilot.de](http://www.filepilot.de) mit dem Suchbegriff „1stpage“ den FirstPage-Editor herunterladen.

Kommen wir also zum HTML-Code:

Neben dem üblichen Kopf <head></head> und deren Inhalte wie Titel und Metatags, dessen Erklärung den Rahmen heute sprengen würde, gibt es auch den Hauptteil einer HTML, den sogenannten body, auf den wir uns jetzt konzentrieren. Dieser beinhaltet nämlich den für unser Vorhaben so wichtigen Code. Schreiben wir also drauf los:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>MUSTER zu Tutorial Steuerung von Frameset mittels Swish-Navigation von
sdhp.de</title>
</head>

<body>
<frameset rows="100,*">
<iframe src="Navigation.html" height = "100" width = "800" frameborder = "0"
framespacing="0" marginwidth="0" marginheight="0" align = "center"
name="Navigation">
<iframe src="START.html" height = "570" width = "800" frameborder = "0"
framespacing="0" marginwidth="0" marginheight="0" align = "center" name="Main">
</frameset>
<noframes>Ihr Browser kann nicht mit Frames arbeiten. Bitte nutzen Sie einen
Browser, der Frames zulässt, z.B. Internet-Explorer neue Version</noframes>
</body>
</html>
```

### Was bedeuten diese Zeilen?

Mit <frameset> eröffnen wir ein sogenanntes Frameset, wie oben erläutert. Über das Attribut rows teilen wir dem Browser mit, dass er dem ersten zu erzeugenden Frame die Höhe 100 geben soll, während dem zweiten Frame über das \*-Symbol eine variable Höhe zugeteilt wird.

Jetzt muss der Browser erfahren, was er in das obere Frame packen soll. Das erklären wir ihm mit <iframe src="Navigation.html"> Jetzt weiss er, dass dieser Frame die Navigation beinhaltet, die wir später noch unter Navigation.html speichern werden. Die anderen Attribute steuern die Höhe, die Breite, ob er einen Rahmen haben soll oder nicht, und weitere Formatierungseinstellungen.

Wer hierzu mehr erfahren möchte, kann sich detailliert informieren unter <http://de.selfhtml.org/html/frames/eingebettete.htm> - definieren

Besonders wichtig jedoch ist noch der tag name="Navigation". Dies ist elementar! Denn diesen Namen behält der Frame, und dies ist der Name, über den der Inhalt später gesteuert wird.

Das gleiche geschieht mit dem Hauptfenster, dem wir sinnvollerweise den Inhalt „START.html“ zuweisen, die wir nachher gegebenenfalls noch gestalten können. Hier besonders wichtig ebenfalls die Tatsache, dass wir diesem Frame den Namen „Main“ verpassen, denn den werden wir später noch häufiger brauchen, dem werden wir noch wieder begegnen.

Jetzt haben wir alles beieinander, und teilen dem Browser das mit, in dem wir den frameset-tag wieder schliessen mittels </frameset>.

Einige „vorsintflutliche“ Browsertypen unterstützen Frames nicht. Das ist heutzutage jedoch nur noch selten der Fall. Trotzdem wollen wir diesen Betrachern wenigstens mitteilen, dass sich dieses so verhält. Diesen Teil packen wir innerhalb die tags <noframes></noframes>. Was dazwischen ist, wird ausschliesslich den Besuchern angezeigt, deren Browser Frames nicht anzeigen können.

Damit beschliessen wir den Body, und die erste HTML-Datei, die das Frameset definiert.

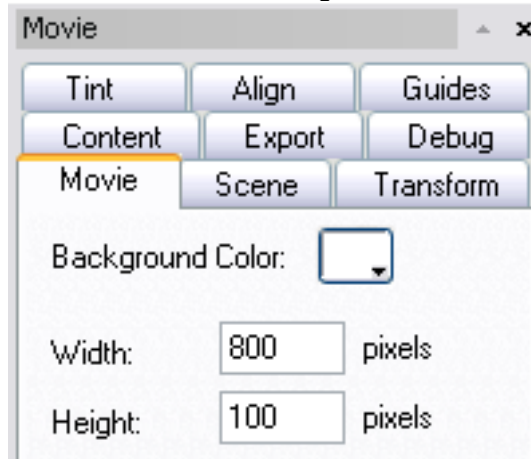
Diese speichern wir nun ab. Damit diese gleich zu Beginn aufgerufen wird, speichern wir diese gleich unter dem Namen „index.html“.

Anmerkung: Bitte die erste Zeile innerhalb des obigen Fensters die unter anderem das Wort „Transitional“ beinhaltet mitkopieren bzw. benutzen, sonst kann es Probleme geben.

Der erste Schritt ist getan, wenden wir und jetzt Schritt 2 zu.

## WIR BAUEN UNS IN SWISH EINE KLEINE NAVIGATION ZUSAMMEN

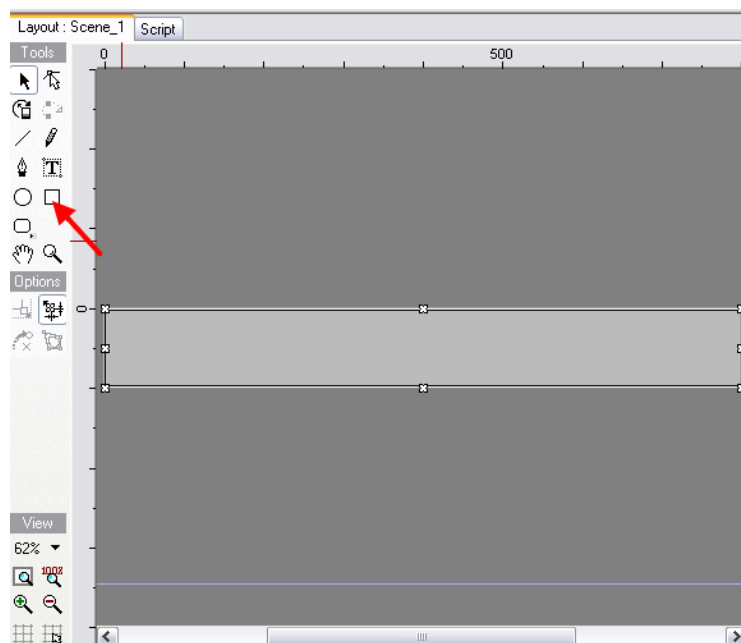
Gleich nach dem Start von SwishMax sollten wir nicht vergessen, dem Film die richtige Größe zu verpassen. Das sind (wir erinnern uns an die Framevorgaben weiter oben) 800 x 100.



Jetzt können wir die Navigation ganz nach unseren Wünschen und Vorstellungen gestalten, Effekte hinzufügen, so viele wir wollen, und genau das ist das Tolle an dieser Geschichte, da eine schöne Navigation sehr sehr entscheidend am Erfolg oder Misserfolg einer Seite beteiligt ist.

In unserem Musterbeispiel geht es vor allem um die Erklärung der Funktion an sich. Deshalb werden wir das ganz nüchtern halten, und uns auf das Wesentliche beschränken.

Mit einem Shape-Objekt Rechteck fangen wir an. Das ist der Untergrund für unsere Navigation. Hier ist schon die erste Gestaltungsmöglichkeit, denn das kann natürlich auch ein bewegtes Sprite, oder eine schöne Hintergrundgrafik, ein Foto, ein Verlauf oder sonst was sein. Der Kreativität sollten hier keine Grenzen gesetzt sein.

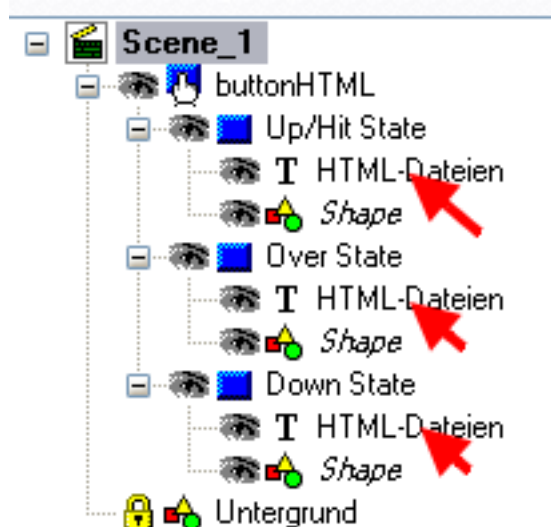


Da dieser Untergrund häufig versehentlich angeklickt wird und verschoben wird, sollte man diesen mittels Klick auf dem Auge links neben dem Objektname mit einem Schloss versehen. Damit spart man sich eine Menge Nerven...



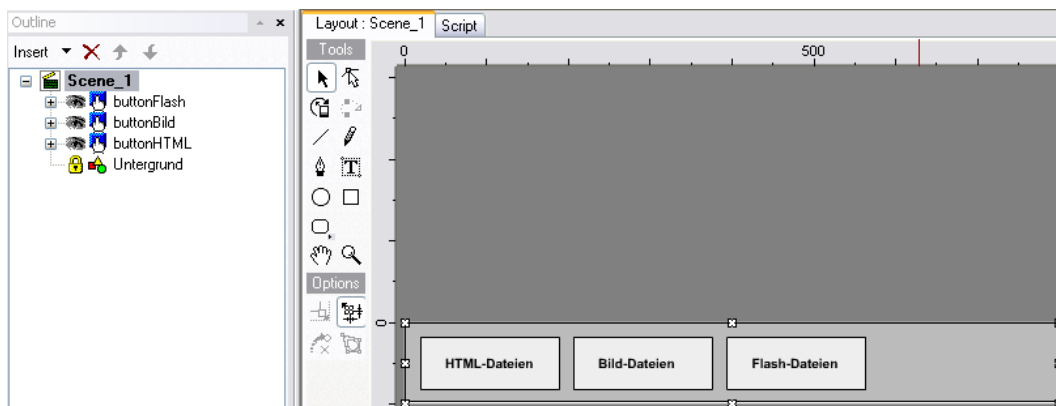
So, jetzt brauchen wir Buttons.

Eine einfache und effektive Weise, einen Button herzustellen ist es, ein shape-Objekt (z.B. ein Rechteck zu erstellen) und einen Text darauf zu legen. Diese beiden werden dann zusammen markiert, und über die Funktion (rechte Maustaste, Group as Button (zu Knopf gruppieren)) von Swish zu einem Knopf zusammen gebaut. Damit der Betrachter den Knopf ohne Probleme als solchen identifiziert, kann man mit Farben in den verschiedenen Zustand Maus über, Maus über gedrückt usw. arbeiten. Einfach rechts die Kästchen separate over und down-state anklicken, und dann links die einzelnen Texte anklicken, und denen unterschiedliche Farben geben:



So. Da wir als Programmierer ja bekannterweise faul sind, und wir wollen, das alle Knöpfe gleich aussehen, nehmen wir den ersten Button, und kopieren ihn in seiner ganzen Pracht gleich 2 Mal. Dazu klicken wir den Button an, und kopieren mittels rechter Maustaste copy object und paste in place(!). Dann klicken wir die beiden neuen Buttons an, und verschieben diese mittels Pfeiltasten an die Position, an der wir sie haben möchten. Dann müssen wir bei den Kopien noch die Titel und den Text innerhalb der Buttons ändern, das wars.

**Jetzt sieht das ganze so aus:**



JETZT geht's ans Eingemachte, denn jetzt kommt der wesentliche Kernteil des ganzen Tutorials!

Denn jetzt teilen wir der Navigations-Swish-Datei mit, dass sie auf Knopfdruck dem erstellten Frame mit dem Namen MAIN (Ihr erinnert Euch dunkel) einen neuen Inhalt zuweisen soll.

### Und genau da wollten wir hin! Also los!

Wir wählen also den Button HTML-Dateien an. Dieser soll auf Knopfdruck im Hauptfenster eine HTML-Datei anzeigen. Damit er das auch tut, wechseln wir über den Layer Script in den Script-Editor. Es empfiehlt sich, den GuidedModus eingeschaltet zu haben.

#### Jetzt wählen wir zunächst aus:

Events/Button/on release

Dann GetURL und HIER wirds richtig spannend:

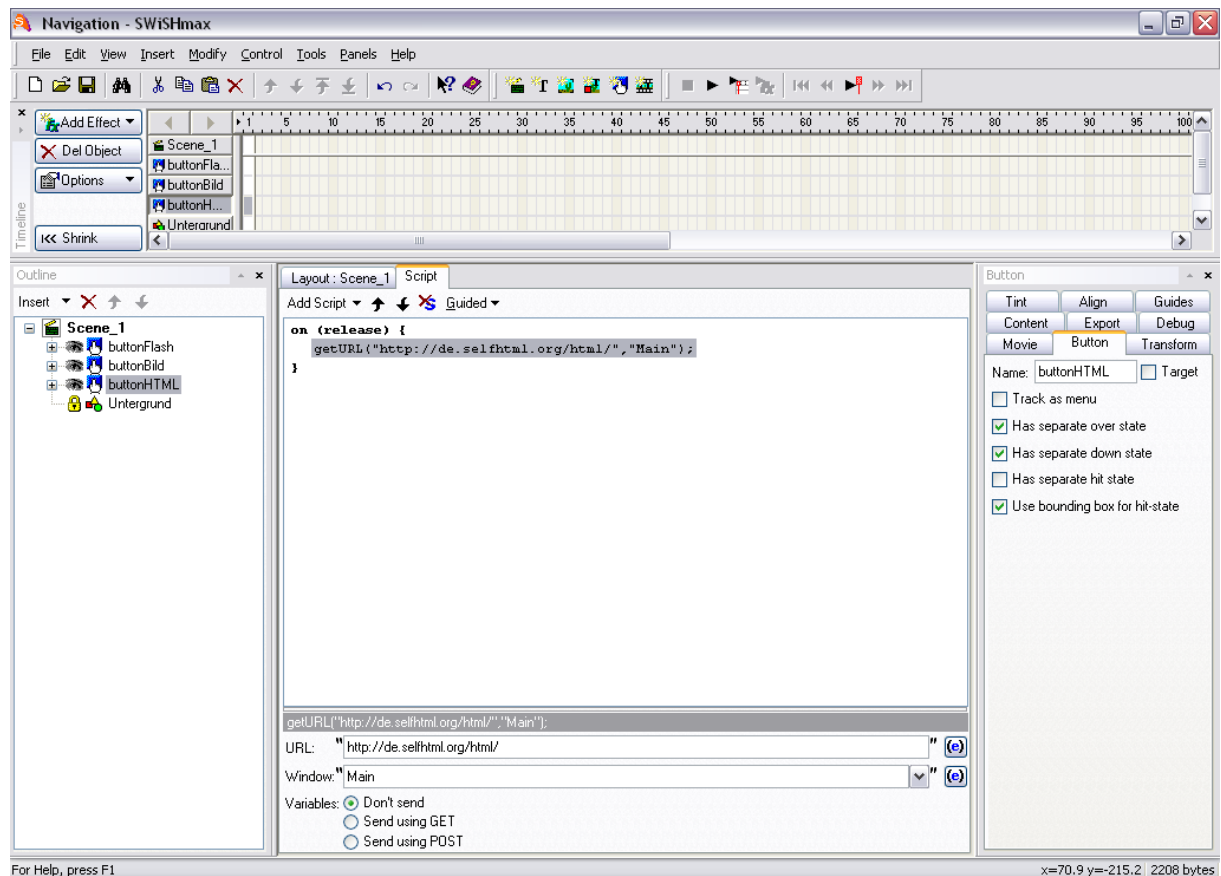
Denn hier geben wir eine beliebige HTML-Datei als Muster ein, z.B. <http://de.selfhtml.org/html/>

#### Jetzt kommt der alles entscheidende und extrem wichtige Teil, das absolute Kernstück des Tutorials:

Im Feld Window(Fenster) geben wir nun den Namen des Hauptframes aus der zuvor erstellten Index.html, also „Main“ ein, in dem wir die HTML-Datei anzeigen lassen wollen.

HIER IST ES EXTREM WICHTIG, DASS DIE GENAUE SCHREIBWEISE EINGEHALTEN WIRD! Gross- und Kleinschreibung beachten!!! Sonst macht der Browser mangels bekannten Namens einfach ein neues Fenster auf, anstatt den Inhalt im Frame anzuzeigen!

Wenn alles richtig gemacht wurde, sieht in Eurem Scriptfenster das Ergebnis jetzt so aus:



Das wiederholen wir jetzt noch innerhalb der anderen Buttons, ich habe hierzu im Download-Paket eine Mustergif beigefügt, die mit dem Namen GIFMuster.gif bezeichnet ist.

Diese lege ich hinter den Button Bild-Dateien. Da die Datei im gleichen Ordner auf dem Webspaces liegt, muss kein besonderer Verweis innerhalb der URL geschehen, es reicht die einfache Eingabe des Dateinamens. Natürlich kann man auch auf einen anderen Ordner verweisen, ich habe diesen Teil deshalb so gestaltet, dass man sieht, dass auch die Möglichkeit eines direkten Verweises innerhalb des gleichen Ordners problemlos möglich ist.

Hinter Flash-Dateien legen ich die URL eines alten Tutorials, beachtenswerterweise lege ich den Link nicht auf die html-Datei desselben, sondern direkt auf die SWF-Datei. Natürlich wird diese dann auf die gesamte Größe des Frames angepasst. Dies sollte man bei der Auswahl der Filmgröße beachten, wenn man einen aufzurufenden Alternativ-Film baut. Denn dann kann man von vorneherein gleich die richtige Größe, nämlich auf die des Frames festlegen.

So, jetzt speichern wir diese Swishdatei noch unter Navigation ab, und exportieren sie auch gleich, per Export SWF+HTML.

Jetzt haben wir alle Dateien zusammen.

Die START-Datei habe ich dem Download-Paket ebenfalls beigefügt wie alle Dateien. Diese kann natürlich beliebig verändert oder mit anderen Inhalten, z.B. einer anderen Swish-Datei gefüllt werden.

So, nun sind wir am Ende des Tutorials angelangt, jetzt können wir das gesamte Ergebnis in einen Testordner auf den Webspaces legen, und uns am Ergebnis freuen.

Diese Art und Weise der Architektur einer Seite ist eine interessante Alternative zu loadMovie, und bietet unzählige Möglichkeiten der Gestaltung.

Gut möglich, dass der eine oder andere sie für die Zukunft für sich entdeckt.

Fragen, Wünsche und Anregungen wie immer per PN an mich, oder gern auch innerhalb des Tutorial-Threads gestellt.

Ich wünsche viel Erfolg, und hoffe, dass alle Erklärungen so eingängig waren, dass sie vom Einsteiger bis zum Fortgeschrittenen problemlos genutzt werden können.



Eurer Nico  
sdhp.de