

```

/*
Herzlich willkommen,
im Folgenden werden Sie, aufbauend auf der Einführungsintro,
weitere Grundlagen und Funktionalitäten in Gess Q. kennen lernen.

Nachdem Sie dieses Skript durchgearbeitet haben können Sie:
- Textersatz verwenden
- mit dem Zurückbutton arbeiten
- HTML Schnittstellen benutzen (z.B. für Bilder, Textformatierung)
- die Darstellung von SingleQ und MultiQ modifizieren
- übersichtlich mit Blöcken strukturieren
- Antwortmöglichkeiten einzeln oder mit "restrict" filtern
*/

/*
Wie schon in der Einführungsintro importieren wir erneut einige
Grundeinstellungen.
*/
// - Grundeinstellungen importieren -
#include "formats.q"
// - Grafische Buttons verwenden -
#include "gbuttons.q"

// -----
// ----- Inhalte der Umfrage -----
// -----
/*
Also fangen wir an...

... mit einer SingleQ, die einen kurzen Begrüßungstext enthält
und 2 Labels zur Auswahl anbietet. Je nachdem, welches
ausgewählt wird, soll das Interview anders ablaufen:
*/

singleq fragel;
text="@insert(headline)Herzlich Willkommen zur weiterführenden Demoumfrage!@insert(end_headline)";
title = "Bitte wählen Sie aus.";
labels=
1 "Zweig 1 - einzelne Antworten filtern"
2 "Zweig 2 - die <i>restrict</i> Anweisung"
;

HTML{ backButton = yes; };

/*
Wir schreiben 'fragel' an die erste Stelle in den main-block (siehe unten).

Es tauchen 3 neue Funktionalitäten auf:
- der Befehl @insert() in Textbereichen
- HTML Notation in Textbereichen
- ein Zurückbutton in der Umfrage

@insert steht in Q. für Textersatz. In den Klammern kann eine beliebige Variable
eingesetzt werden. Hinter den Variablen 'headline' und 'end_headline' versteckt
sich etwas Textformatierung in HTML.
Sie sind in der anfänglich importierten Datei formats.q definiert.

Selbiges im kleinen Stil machen die Anweisungen <i> und </i>. Es handelt sich
hierbei um Anfang- und Endnotation in HTML für kursive Schrift (i für italic).
Gleiches geht mit <u></u> (underline-unterstrichen) und <b></b> (bold-fettgedruckt).

Auf diese Art und Weise können Schriftformatierungen beliebig modifiziert werden.
Aber erstmal genug mit HTML.

Die Anweisung backButton = yes fügt in allen folgenden Fragen einen Zurückknopf
ein. Bis zum Interviewende oder bis backButton = no den Knopf wieder ausstellt.
Dies kann z.B. auch zum Testen der Filter und des Interviewablaufs hilfreich sein.
Auf die notwendige Umrahmung mit HTML{ }; soll an dieser Stelle noch nicht weiter
eingegangen werden.

```

Abhängig von der gegebenen Antwort in 'frage1' soll es nun 2 unterschiedliche Zweige im Ablauf geben.

Erstmal nur ZWEIG 1.

Nach einem kurzen Infotext soll eine MultiQ in Form einer 2x2 Matrix folgen, die in den Antwortlabels keine Texte, sondern Bilder von Schauspielern enthält.

\*/

**textq** z1\_f1;

**text**="Sie befinden sich in Zweig 1 (einzelne Antworten filtern).";

**HTML**{ **maxLabelsPerCol** = 2; };

**multiq** z1\_f2;

**text**="Welche Schauspieler haben Sie zuletzt im Free-TV gesehen?";

**title**="Mehrfachnennung möglich.";

**labels**=

1 "<img src='./media/qintro2/clooney.jpg'>"

2 "<img src='./media/qintro2/cage.jpg'>"

3 "<img src='./media/qintro2/crowe.jpg'>"

4 "<img src='./media/qintro2/travolta.jpg'>"

;

/\*

Die Antworten der SingleQ und MultiQ werden standardmäßig untereinander gelistet.

Die Anweisung 'maxLabelsPerCol' gibt an, wie viele Antworten sich maximal untereinander befinden dürfen. Wird die angegebene Zahl überschritten, wird eine neue Spalte eingefügt.

In den Antwortlabels wird nun per HTML Befehl <img ...> (image-Bild) jeweils eine Grafik eingefügt. Das Attribut src='...' gibt die Zielfile im Medienverzeichnis an.

Von den ausgewählten Schauspielern soll nun in einer folgenden SingleQ derjenige gewählt werden, der am häufigsten gesehen worden ist.

\*/

**singleq** z1\_f3;

**text**="Und welchen Schauspieler haben Sie zuletzt am häufigsten im Free-TV gesehen?";

**labels**=

1 "<img src='./media/qintro2/clooney.jpg'>" **flt**(1 in z1\_f2)

2 "<img src='./media/qintro2/cage.jpg'>" **flt**(2 in z1\_f2)

3 "<img src='./media/qintro2/crowe.jpg'>" **flt**(3 in z1\_f2)

4 "<img src='./media/qintro2/travolta.jpg'>" **flt**(4 in z1\_f2)

;

**flt** = (num(z1\_f2) ge 2);

/\*

Auch einzelne Antwortlabels können mit Filtern versehen werden, indem man sie einfach hinter den Textbereich schreibt. Hier wird jeder Schauspieler mit einem Filter auf sein korrespondierendes Label aus z1\_f2 versehen.

Nun werden in z1\_f3 nur die Labels eingeblendet, die zuvor in z1\_f2 gewählt wurden. (In Zweig 2 folgt eine Kurzschreibweise)

Zusätzlich wird Frage z1\_f3 überhaupt nur dann angezeigt, wenn wenigstens 2 Schauspieler zur Auswahl stehen.

Der Befehl num(...) zählt die Anzahl der gegebenen Antworten.

\*/

**block** zweig1 = ( z1\_f1 z1\_f2 z1\_f3 );

**flt** = (frage1 eq 1);

/\*

Die Fragen z1\_f1, z1\_f2 und z1\_f3 werden in block 'zweig1' zusammengefasst. Der komplette Block kann nun (genau wie einzelne Fragen) gefiltert werden.

In diesem Fall wird der Block nur dann angezeigt, wenn Zweig 1 in 'frage1' gewählt wurde.

Nun weiter mit ZWEIG 2.

In diesem Zweig soll dasselbe mit (schöneren) grafischen Buttons und der 'restrict'-Anweisung statt einzelner Filter erreicht werden.

Die Schauspieler selbst sollen als Buttons fungieren:

\*/

```

HTML{
  addProperty( "button_checked" "xclooney.jpg xcage.jpg xcrowe.jpg xtravolta.jpg" );
  addProperty( "button_unchecked" "clooney.jpg cage.jpg crowe.jpg travolta.jpg" );
};

/*
  Hierzu müssen markierte und nicht-markierte Buttons definiert werden.
  Über einen Property-Mechanismus (der an dieser Stelle nicht weiter erläutert
  werden soll) werden hierzu die Properties (Einstellungen) 'button_checked'
  und 'button_unchecked' zugefügt und mit den 4 Bildern versehen.

  Das war schon das Schwierigste. Der Zweig wird mit den Fragen
  z2_f1, z2_f2 und z2_f3 vervollständigt:
*/

textq z2_f1;
text="Sie befinden sich in Zweig 2 (filtern mittels 'restrict').";

multiq z2_f2;
text="Welche Schauspieler haben Sie zuletzt im Free-TV gesehen?";
title="Mehrfachnennung möglich.";
labels=
1 ""
2 ""
3 ""
4 ""
;

singleq z2_f3;
text="Und welchen Schauspieler haben Sie zuletzt am häufigsten im Free-TV gesehen?";
labels=
1 ""
2 ""
3 ""
4 ""
;
flt = (num(z2_f2) ge 2);
restrict = z2_f2;

/*
  Da aussagekräftige Buttons verwendet werden, sind Labeltexte nun überflüssig.

  Die Filter auf den einzelnen Labels (wie in f1_f3) können kurz mit der Anweisung
  'restrict' geschrieben werden. Es wird automatisch ein Filter auf jedes Label
  gesetzt. Die Labelnummern müssen hierzu natürlich übereinstimmen.

  Abschließend wird auch der zweite Zweig in einem Block zusammengefasst und
  gefiltert.
  Blöcke können beliebig geschachtelt und in den main-Block integriert werden.
*/

block zweig2 = ( z2_f1 z2_f2 z2_f3 );
flt = (fragel eq 2);

block zweige = ( zweig1 zweig2 );

block main = ( fragel zweige );

```