```
/*
  Welcome,
  In the following based on the introduction you will get to know further basics and
  functions in Gess Q.

  After you worked through this script you will be able to:
  - use text replacement
  - work with the back-button
  - use HTML interfaces (for example for images, text formatting)
  - modify the presentation of SingleQ and MultiQ
  - create clear structures with blocks
  - filter answer options individually or with "restrict"
*/

/*
  Like in the introduction we are importing some basic settings again.
*/
// - import basic settings -
#include "formats.q"
// - use graphic buttons -
#include "gbuttons.q"





// --------------------------------------------------------------
// ---------- Survey Contents ------------------------------
// --------------------------------------------------------------
/*
  So let’s begin...

  ... with a SingleQ, which contains a short welcome-text and 2 labels to choose from.
  Depending on which one gets chosen the interview is supposed to head in different
  directions:
*/

singleq question1;
text="{headline}Welcome to our continuative demo-survey!{end_headline}";
title = "Please choose.";
labels=
1 "branch 1 – filter particular answers"
2 "branch 2 - the <i>restrict</i> command"
;

HTML{ backButton = yes; };

/*
  We put down 'question1' on the first place of the main-block (see below).

  3 new functions appear:
  - curly brackets {} in text areas
  - HTML notation in text areas
  - a back-button in the survey

  Curly brackets in Q. stand for text replacement. Any variable can be put in curly
  brackets. The variables 'headline' and 'end_headline' need some text formatting in HTML.
  They are defined in the file formats.q imported at the start.
  The same is getting done with the commands <i> and </i>. They are start- and end-notations
  in HTML for italic type (i for italic).
  It works the same with <u></u> (underline) and <b></b> (bold).

  This way fonts formatting can be modified as desired.
  But for the time being enough about HTML.

  The command backButton = yes inserts a back-button into all following questions up to the
  end of the interview or until backbutton = no switches the button off.
  This can be helpful for example for testing the filter and the interview process as well.
  At this point we are not going to go more into detail with the necessary framing with
  HTML{  };



  Depending on the answer given in 'question1' there must be now 2 different branches in the
  process.

  First of all only BRANCH 1.
  After a short information a MultiQ in terms of a 2x2 matrix is supposed to follow, which
  in the answer labels doesn't contain texts but images of actors.
*/

textq z1_q1;
text="You are in branch 1 (filter particular answers).";


HTML{ maxLabelsPerCol = 2; };
multiq z1_q2;
text="Which actors did you see lately on Free-TV?";
title="several statements possible";
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'>"
;

/*
  The answers of the SingleQ and MultiQ are in standard listed underneath each other.
  The command 'maxLabelsPerCol' states how many answers maximum can stand underneath each
  other. If the number is exceeded, a new column is inserted.

  Now in the answer labels per HTML command <img ...> (image) a graphic each gets inserted.
  The attribute src='...' names the target file in the media directory.

  Out of the chosen actors the one who has been seen the most is supposed to be chosen in
  the following SingleQ.
*/

singleq z1_q3;
text="And which actor have you lately seen the most on Free-TV?";
labels=
1 "<img src='./media/qintro2/clooney.jpg'>"  flt(1 in z1_q2)
2 "<img src='./media/qintro2/cage.jpg'>"     flt(2 in z1_q2)
3 "<img src='./media/qintro2/crowe.jpg'>"    flt(3 in z1_q2)
```

```
4 "<img src='./media/qintro2/travolta.jpg'>" flt(4 in z1_q2)
;
flt = (num(z1_q2) ge 2);

/*
  Even particular answer labels can be provided with filters by putting them down behind the
  text. Here each actor is getting provided with a filter on its corresponding label from
  z1_q2.

  Now in z1_q3 only those labels are shown, which have been chosen in z1_q2
  before. (In branch 2 comes an abbreviated from)

  Apart from that question z1_q3 only shows up if at least 2 actors are available.
  The command num(...) counts the amount of the answers given.
*/



block branch1 = ( z1_q1 z1_q2 z1_q3 );
flt = (question1 eq 1);

/*
  The questions z1_q1, z1_q2 and z1_q3 are getting combined in the block 'branch1'.
  The complete block can now (exactly like particular questions) be filtered.

  In this case the block only shows up if branch 1 has been chosen in ‚question'.


  Now to BRANCH 2.
  In this branch the same is supposed to be achieved with (nicer) graphic buttons and the
  'restrict'-command instead of particular filters.

  The actors themselves are supposed to act as buttons:
*/

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

/*
  For that marked and unmarked buttons have to be defined.
  With a property-mechanism (which is not meant to be explained in detail at this point) the
  properties (settings) 'button_checked'
  and 'button_unchecked' are added and provided with the 4 images.

  That was the most problematic. The branch is completed with the questions
  z2_q1, z2_q2 und z2_q3:
*/

textq z2_q1;
text="You are in branch 2 (filter with 'restrict').";

multiq z2_q2;
text="Which actor have you lately seen the most on Free-TV?";
title="several statements possible.";
labels=
1 ""
```

```
2 ""
3 ""
4 ""
;

singleq z2_q3;
text="And which actor have you lately seen the most on Free-TV?";
labels=
1 ""
2 ""
3 ""
4 ""
;
flt = (num(z2_q2) ge 2);
restrict = z2_q2;

/*
  Because significant buttons are getting used label texts are not necessary now.

  The filters on particular labels (like in g1_q3) can now be shortly written with the
  command 'restrict'. There is automatically a filter set on each label. The label numbers
  of course have to match.

  Afterwards the second branch is combined and filtered in a block.
  Blocks can be nested and integrated into the main-block.
*/


block branch2 = ( z2_q1 z2_q2 z2_q3 );
flt = (question1 eq 2);


block branches = ( branch1 branch2 );

block main = ( question1 branches );
```