```
/*
  Welcome,
  In the following we would like to see you through your first steps with the Q. script
  language.


  After you worked through this file with our Tutorial you will be able to...
  - add, edit, and delete our standard questions
  - conditionally fade up questions (filter)


  And by doing that you will get used a bit to the syntax of the language at the same time.


  So let's begin!


  This text is in a comment block. It has no influence on the computer operation and is only
  there for overview and contains for example comments and reminders.
*/


// comment block /* ... */
// Apart from that it makes comments line-by-line // ...



/*
  Before we start with the proper survey contents we import 2 extern files, which have
  settings for the basic image of the survey.
  They don't really matter for the rest of this Tutorial.
*/
// - import basic settings -
#include "formats.q"
// - use graphic buttons – COMMENT and TEST
#include "gbuttons.q"




// ------------------------------------------------------------
// ---------- Survey Contents ----------------------------
// ------------------------------------------------------------
/*
  Generally using upper and lower case at keywords doesn't matter.

  In theory questions in a script can be randomly aligned.
  The proper order in the online interview is defined at the end of the script in the
  so-called main-block.


  But sequential...



  Let's start with the first question or in fact with the welcome text without answer
  options. This can be realized with a TEXTQ.
  We call it 'questions1' (this is optional) and assign it with a text:
*/


textq question1;
text="Welcome to our first online interview!";


/*
  To make this question show up first on the screen, first we add the already mentioned
```

```
main-block to it. Please scroll down and look at the 'block main' ...

  Onwards with the first proper question. It should allow 1-out of-N answers.
  For that we need the type SINGLEQ. Let's call it 'question2'.
*/

singleq question2;
text="Did everything work okay so far?";
labels=
1 "yes"
2 "No"
;

/*
  In addition to the text there are now 'labels', which represent the answer options.
  Between = and ; any number of answers can be inserted.
  You start with any number and the answer text in quotation marks.
  Later in the data set the answers get coded according to those numbers.

  Next...
  Same game with an M-out of-N question. The MULTIQ.
*/

multiq question3;
text="Which films did you watch in 2010 at the cinema?";
title="several statements possible.";
labels=
1 "Avatar"
2 "Alice in Wonderland"
3 "Shrek 3D"
4 "The end is my beginning" random
5 "Frozen" random
6 "The Bounty Hunter" random
7 "Inception" random
99 "None of them" single
;

/*
  Beside the text there can be a title. This for example can be formatted differently and
  contain interviewer instructions.
  Apart from that the labels 4,5,6 and 7 are assigned with the keyword random.
  The positions of these 4 answers are evaluated at random from case to case.
  This works for the SingleQ and other types of questions as well.
  Only in the MultiQ the keyword 'single' is available. It annotates in this case answer 99
  -none of them- as being exclusive.

  Next...
  Now we would like to ask an open question. But only if 'question2' has been answered with
  -no-. For open questions there is the OPENQ.
*/

openq question4;
text="What didn't work that well so far?";
flt = (question2 eq 2);

/*
  The OpenQ is basically the same as the TextQ. There is only a box for text answers to be
```

```
  inserted.
  With the filter 'flt' a constraint is attached to this question. It only shows up if
  'question2' is 2. 2 is the label number for the answer -no- and 'eq' is available for
  'equals' (equality). Beside 'eq' stand the operators 'lt' (smaller), 'gt' (larger), 'not'
  (Negation), 'or' (or) and
  'and' (and).
  Filters like this can be attached to any type of question.

  Next...
  Now we would like to ask the interviewee to rate the film Avatar with a number between 1
  and 10.
  Obviously only if he watched it.
  Numeric answers are possible with the NUMQ.
*/


opennumformat area = 1 2 0 0 1 10 "wrong entry!";
numq question5;
text="And what did you think of Avatar?";
title="Please rate between 1(-very bad) and 10(-very good).";
labels=
1 "Avatar" format area
;
flt = (1 in question3);

/*
  To limit the permitted range of values an OPENNUMFORMAT is needed.
  It contains like the questions a name 'area' and is getting referenced like this again in
  the labels of the NumQ. The numbers in the OpenNumFormat here mean the following:
  1-2 integer places, 0-0 decimal places, 1-10 altogether allowed values.

  The filter here gets defined with 'in', because 'question3' is a MultiQ and might have
  several answers. The equality is not clear anymore. 'in' checks the existence of a certain
  answer.
*/


block main = ( question1 question2 question3 question4 question5 );
```