

Online surveys



[main website](http://www.psychtoolkit.org) (<http://www.psychtoolkit.org>) | [documentation 2.0.9](http://www.psychtoolkit.org/doc2.0.9) (<http://www.psychtoolkit.org/doc2.0.9>)

Structure of questionnaire

- General structure of a question

- Comments for programmers

Formatting text

Question types

- Radio: select one out of many

- Check: select one or more out of many

- Non-listed option for radio and check items

- info: Show basic information, message, etc

- Play a sound

- Youtube: show a video

- Experiment: Embed a reaction time experiment

- Jump to a question (conditionally or unconditionally)

- Textline: Ask for a line of text

- Textbox: Ask for one or more lines of text

- Range: Enter a numerical value

- Scale: Likert style scales

- Multiradio: Have multiple one-choice items in one questions

- Images

- Change the text of the button

- Randomize question order

- Scoring when running questionnaire using "set"

- Running different sections of questionnaire for different participants

For an introduction, see the [introduction to online surveys](#).



The survey scripting language is **completely different** from the experiment scripting language. If you want to embed an experiment in an online survey, you need to follow these instructions.

Structure of questionnaire

Questionnaires are simple text files. Each line has a meaning. The computer reads the questionnaire line by line. The first character of each line typically has a special meaning.

- Each questionnaire consists of multiple questions, coded as a simple text (which you can edit in your browser)
- The questionnaire needs to be manually entered, line for line (examples below)
- Each question consists of a couple of lines
- Questions are separated from one another by one (or more) empty line
- Each question, irrespective of type, has a simple basic structure (read more about the Question Structure)
- Each questionnaire can have sections that randomize the order (read more about randomized order)
- Each questionnaire can have comments on lines *starting* with the hashmark sign (#)

General structure of a question

- Each question consists of a number of text lines.
- Questions are separated from one another with an empty line
- Each line (typically) starts with a letter followed by a colon. The letter describes an important features, such as the name of the question, or the type of question.
- Each question has one or more items, each one starting with a minus sign

The first line of a question starts with **l:**. The letter **l** stands for "label". Every question has a label, or a "name". This is for two reasons: 1) The computer needs a way of identifying a question, and it uses the label. 2)



The user will need a way of identifying answers to questions, and the label is the easiest way to do this.



A label can only have letters (upper or lower case) and numbers, and the underscore character.



Each question needs to start with **l:**, but you do not necessarily need to give a label. In that case, the software will call the question "auto_" followed by a number. This is handy if you are not willing to label your questions. You can still identify them by the whole question. And for "jumps", the label typically is not necessary for users at all.

Example of structure of question. The arrow ← is for explanation only

```
l: MyQuestion1          <-- this is the label
t: radio                <-- this is the type of question
q: What is your favourite movie? <-- this is text of the question the participant
sees
- Toy Story             <-- answer 1
- Spirited Away         <-- answer 2
- Babe                  <-- answer 3
- Ratatouille           <-- answer 4
                        <-- the empty line separating this question from
the next
```



Each question can show an image at the top. You need to upload the image file (png, bmp, gif, jpg, etc). See images.



You can use the option line **o: random** to randomize the order of the items.

Comments for programmers

When you program your survey, you might sometimes want to make a comment that is just for you, and that the computer should ignore. You can use the hashmark for that.



Unlike PsyToolkit experiment scripts, the comment symbol "#" must be at the beginning of a line. In other words, you cannot have comments starting later in a line.

Formatting text

You can format text using standard HTML tags. The most common are those for bold, italic, and text color. The general screen color cannot be changed.

- Everything between `` and `` will be presented in bold font.
- Everything between `<i>` and `</i>` will be presented in italic font.
- Everything between `` and `` will be presented in red, etc.

Example of formatting text

```
l: MyQuestion
t: radio
q: What is your <i>favourite</i> movie?
- <b>Toy Story</b>
- <font color='blue'>Spirited</font> Away
- Babe
- Ratatouille
```

Question types

There are the following types of questions:

- radio One choice out of many options
- check Multiple choices out of many options (check boxes)
- experiment Embed an experiment
- scales Likert type scales
- range Let user enter a number range between two values
- text line Participant can enter exactly one line of text
- text box Participant can enter multiple lines of text

- multiradio Multiple one-choice items in one question

There are the following types that just present info and a continue button

- info Just show some information, message, etc
- youtube Show a YouTube video

There are various control items:

- jump Jump to a question label (conditionally or unconditionally)
- set Set a question label (for scoring and random numbers)
- random Randomize questions

Radio: select one out of many

Use this when you want that only one possible answer is selected. The name "radio" comes from classic radios, which had a set of buttons of which maximally one could be pushed down.

Example

```
l: MyQuestion1
t: radio
q: What is your favourite movie?
- Toy Story
- Spirited Away
- Babe
- Ratatouille
```

This is item 1 out of 1 items.

What is your favourite movie?

☐ Toy Story

☐ Spirited Away

☐ Babe

☐ Ratatouille

Click this button to continue

The "q:" statement is the only statement that can run over multiple lines. This can sometimes be handy to have longer questions, like the one in the



example below, which uses HTML formatting.

Example with question over multiple lines

```
l: MyQuestion1
t: radio
q: In the following question, we would like to which movie you like
  best. Make sure you understand the following:
<ul>
<li>This is only about movies
<li>You are allowed to say that you are not sure.
</ul>
- Toy Story
- Spirited Away
- Babe
- Ratatouille
- I am not sure
```



Instead of text, you can show an image. You need to upload the images (as for topimage) and then use the {image=filename} option. In view files, you can see which filenames you already have. Don't use files with spaces. You must enter the whole filename including extension, such as jpg, png, bmp, or gif. This allows you to even upload little animations using animated gifs.

Example with uploaded radio type images

```
l: MyQuestion1
t: radio
q: In the following question, we would like to which movie you like
  best. Make sure you understand the following:
<ul>
<li>This is only about movies
<li>You are allowed to say that you are not sure.
</ul>
- {image=toystory.jpg}
- {image=spirited.jpg}
- {image=babe.png}
- {image=ratatouille.bmp}
- I am not sure
```



The default scores of a radio question with three answer options would be 1, 2, and 3. You can change these to other scores using the method below using the {score=?} option, where ? needs to be a number.

Example of changing scores

```
l: weather
t: radio
q: What is the weather like?
- {score=3} Very bad
- {score=2} It could be better
- {score=1} Very Sunny
```



Frequently asked question. How to add an option "other". Check here.

Check: select one or more out of many

Example

```
l: MyQuestion2
t: check
q: Which of the following did you eat today?
- Rice
- Meat
- Tofu
- Fish
- Beans
- Soup
```



You can randomize the order of the items in a radio question by setting the **o: random** options (see second example).



You can require that participants select at least one, or at least two, or at least *n* of the options, see example below. Only when people select at least two items, the continue button will become clickable.

Example

```
l: MyQuestion2
t: check
o: require 2
q: Which of the following food do you like? Select at least 2 items.
- Rice
- Meat
- Tofu
- Fish
- Beans
- Soup
```

You can even set the maximum of options a participant can select with adding a second number to the **o: require** statement. The second number specifies the maximum number of items that can be selected. For example, if you want that people select at least 2 but no more than 3 items, you can do that as follows:

Example

```
l: MyQuestion2
t: check
o: require 2 3
q: Which of the following food do you like? Select at least 2 items, but no more
than 3 items.
- Rice
- Meat
- Tofu
- Fish
- Beans
- Soup
```

This is item 2 out of 2 items.

Which of the following did you eat today?

☒ Rice

☐ Meat

☒ Tofu

☐ Fish

☐ Beans

☒ Soup

Click this button to continue



Frequently asked question. How to add an option "other". Check here.

Non-listed option for radio and check items

A common question is if it is possible to have an "other" option and a line where people can give an alternative. This is not possible on one screen, but you can easily set it up as follows:

Example

```
l: MyQuestion
t: radio
q: What is your favourite movie?
- Toy Story
- Spirited Away
- Babe
- Ratatouille
- Another movie not listed here

l:
t: jump
- if $MyQuestion != 5 then goto MyNextQuestion

l: OtherFavouriteMovie
t: textline
q: Type the name of your favourite movie
- My most favourite movie is

l: MyNextQuestion
t: radio
q: Did you do sports yesterday?
- Yes
- No
```

In the above example, the line "if \$MyQuestion != 5 then goto MyNextQuestion" can be explained as follows:

- **\$MyQuestion** (the \$ indicates that it is a variable) refers to the line number of the option selected. Thus, if someone selects "Toy Story", they get value 1, and Another movie not listed here gets value 5 (the fifth line).
- The **!=** means in computer language *is not equal*. Thus if the selected line *is not equal to* value 5, then people do not need to enter anything in text. They are ready to go on. But if they selected the fifth line, they will go on to the next question where they can enter a value.

info: Show basic information, message, etc

Sometimes you just want people to read something or give feedback without and question. For that you can use the type **info**. Like other types, you can present a top image above it. It has (currently) no lines starting with "-". You can use HTML to format the text and you can insert variables like in other questions types.

Example 1

```
l: some_message
t: info
q: Hello<br>
Here is some information without questions to answer.
```

Example 2

```
l: some_message
i: some_image.png
t: info
q: Thank you. Here is some feedback.<br>
Your score was: {$score}
```

Play a sound

For each question type, you can add one (or more) sound file(s). Below are two examples:

Example 1

```
l: q1
a: horse.mp3
t: info
q: Play the sound. Which animal do you hear?
- Horse
- Pig
- Cow
```



MP3 files are much smaller than WAV files, so you better use MP3, although you can use any sound file type.



Sound files take up much "bandwidth". If possible, put sound files on an external website and refer to it using http: prefix, like in the example

below.

Example 2

```
l: q1
a: http://www.my_webpage.com/sounds/horse.wav
t: info
q: Play the sound. Which animal do you hear?
- Horse
- Pig
- Cow
```

Youtube: show a video

You can embed a YouTube video. On the "-" option line, you need to enter the video's code, which you can get from the URL. It is the bit that follows the word watch. Thus in the following line **<https://www.youtube.com/watch?v=m9WxvT-82Xg>**, it is the **m9WxvT-82Xg**

Example

```
l: MyQuestion3
t: youtube
q: Please watch the following video
- m9WxvT-82Xg
```

Experiment: Embed a reaction time experiment

You can use the name of one of your experiments. Make sure it works before you embed it. Note that you can leave the "q:" to save some space above the canvas area where the experiment is being shown.

Example

```
l: MyQuestion4
t: experiment
q: Please do the following experiment
- myexperiment
```

Example without a "q:", which saves some screen space

```
l: MyQuestion4  
t: experiment  
- myexperiment
```

Note: You can add the optional {fullscreen} to request the experiment to run in fullscreen. Of course, it is up to the end user to do this, and it might not run in all browsers (but it does in Firefox and Chrome).

Example of experiment in fullscreen mode

```
l: MyQuestion4  
t: experiment  
- {fullscreen} myexperiment
```

Jump to a question (conditionally or unconditionally)

With **jump** items, you can jump through the questionnaire. It is typically only necessary if you want to have some questions conditionally. For example, if you have a question for women only, there is practical to first ask the participant about their gender, and then depending on the answer jump to questions.



Jump items do not necessarily need to have a label (although they should start with *l:), because the jumps themselves are not interesting for your analysis. That said, if you want to jump to another jump, you still will need a label.

You will need to have at least one line starting with a "-". In the "-" line, you can have an "if" built up in the following ways. You can have an if (conditional) or just a goto (unconditional jump):

- if <value> <comparator> <value> then goto <label>
- goto <label>

In the above, the value can be a \$ character with a label or a number (see examples). The comparator can be one of the following:

Comparator	Meaning

==	is equal? (just one "=" works too)
!=	is not equal?
<	is smaller than?
>	is larger than?
<=	is smaller than or equal to?
>=	is greater than or equal to?



In a jump question, you can have multiple if..then lines, and they are carried out one after another until the if condition applies.



It is necessary that you have spaces between items of the comparator part (thus write "if \$q1 == 1", and not "if \$q==1"). If you do not use spaces, it will not work.



The dollar sign in front of the label is important, it means that you are looking for the value of the answer to that question. After the goto statement, however, there is no dollarsign.



You can also compare the answers of two questions. This **works only for radio questions**.

Example of jump

```
l: sex
t: radio
q: What is your biological sex
- Male
- Female

l: jump1
t: jump
- if $sex == 1 then goto car

l: birthcontrol
t: radio
q: Do you take birth control pills?
- Yes
- No
```

```
l: car
t: radio
q: Do you have a car?
- Yes
- No
```

```
l: birthcontrol
t: radio
q: What type of birthcontrol do you use?
- pill
- coil
- condoms
- do not wish to answer this question
```



Sometimes you just want to jump from one location in the questionnaire to another. In the following example, the "blabla" question will never be carried out:

Example of unconditional jump

```
l: somejump
t: jump
- goto myotherquestion

l: blabla
t: radio
q: How are you?
- Just normal
- Fine

l: myotherquestion
t: radio
q: What is the weather like?
- Rain
- Dry and overcast
- Dry and sunny
- Something different.
```

If all you want to do is an unconditional jump like in the example above, you can use the **j:** command with just the label you want to jump to:

Example of unconditional jump with short j: statement

```
j: myotherquestion

l: blabla
t: radio
q: How are you?
- Just normal
- Fine

l: myotherquestion
t: radio
q: What is the weather like?
- Rain
- Dry and overcast
- Dry and sunny
- Something different.
```

Finally, in "check" questions, you can test whether a specific item of your list has been checked. Here is a common example:

Example of jump on check-question item

```
l: food
t: check
q: What food did you just eat?
- Bread
- Rice
- Noodles
- Something else

l:
t: jump
- if $food.4 == 0 then goto drinkquestion

l: otherfood
t: textline
q: Please enter what food you ate
- The name of the food

l: drinkquestion
t: radio
q: What did you just drink?
- Water
- Fruit juice
- Fizzy drink
- Tea
- Coffee
- Alcoholic beverage
```



The notation with the dot gets the value of a sub-item. The \$food.4 can be 0 (not checked) or 1 (checked). Thus the checked sub-items always are 0 or 1.



This does, of course, not work for "radio" questions, in which can you should just check the value of \$food (whether it is 4 or not).

Textline: Ask for a line of text

Sometimes, you want that a participant enters some text, such as their participant number or their name. You can just use for a line of text (as you would use to enter your name) or for a whole text box (see below). The can also ask specifically that the entered value must be a number.

Example


```
l: MyQuestion6
t: textline
q: Please enter your first name and surname
- First name
- Surname
```



You can **require** people to fill in a field, but this uses a HTML5 feature which is only present in modern browsers (for example, Firefox, Chrome, and Internet Explorer version 10 and higher). You need to check if your browser has it, and if you really need it, you need to specify browser requirements. There is an option for this on the edit survey page.

Example

```
l: MyQuestion6
t: textline
q: Please enter your first name and surname
- First name
- {require} Surname
```

You can force the shown line length to have certain number of characters with the `{length=X}` option, whereby X is a whole number:

Example

```
l: MyQuestion6
t: textline
q: Please enter your first name and surname
- First name
- {length=10} Surname
```

Finally, you can force people to enter a number between two values. To do so, you must specify min and max, and then the survey will force user to enter a number.

Example

```

l: MyQuestion6
t: textline
q: Please enter your first name and surname
- First name
- {length=10} Surname
- {min=10,max=100} Type your age (must be numerical)

```



Sometimes, you just want to have the main question and no text for the line starting with "-". This can be done, see example below.

Example with empty element line

```

l: MyQuestion6
t: textline
q: Please enter your name
-

```

Textbox: Ask for one or more lines of text

Sometimes, you want that a participant enters some more text, such as feedback on the survey.

Example

```

l: MyQuestion6
t: textline
q: Please enter your feedback and (optionally) your name and address
- Feedback
- Name and address

```

While textline has the option "length", the textbox has width and height, which can be set in different ways; see example:

Example

```

l: MyQuestion6
t: textbox
q: Please enter your feedback
- Your first comment (note: no width or height provided)
- {w=100,h=5} Your second comment (here we have 100 characters width and 5 lines high
- {width=100,height=5} Options spelled out in full works too

```

As for textline, the "require" option can be used.

Range: Enter a numerical value

Sometimes you want to enter a numerical value, such as age or the value on a Likert scale. With "range" you will get a slider, and you see the number in a yellow-highlighted box on the right of the slider. Participants can use mouse or arrow keys to change the number.

Demo survey

How old are you?

Select your range with the slider

25

Click this button to continue



You can specify the minimum and maximum value with **min** and **max** as shown below. If not, the scale runs from 0 to 100.



Some researchers do not like sliders and want that people can just enter a number with the keyboard. That can be done: You can use the textline instead and request a number. See text line.

Example

```
l: age
t: range
q: What is your age?
- {min=18,max=100}

l: agefamily
t: range
q: How old are you parents if alive
- {min=18,max=120} Father
- {min=18,max=120} Mother

l: odd
t: range
q: Which number is your favorite?
- {min=1,max=100,by=2} Favorit odd number
- {min=2,max=100,by=2} Favorit even number

l: smallnumber
t: range
q: Choose a number between 0 and 1
- {min=0,max=1,by=0.1} Number between zero and one
```



You can use the question option "require" (see example below). Some participants have complaint that they did not get past this, and this needs some more piloting. Test it yourself and see if it works for you.

Example which forces people to click/touch the slider at least once

```
l: age
t: range
o: require
q: What is your age?
- {min=18,max=100}
```

There are a number of other options you can specify in the "elements" of the range question:

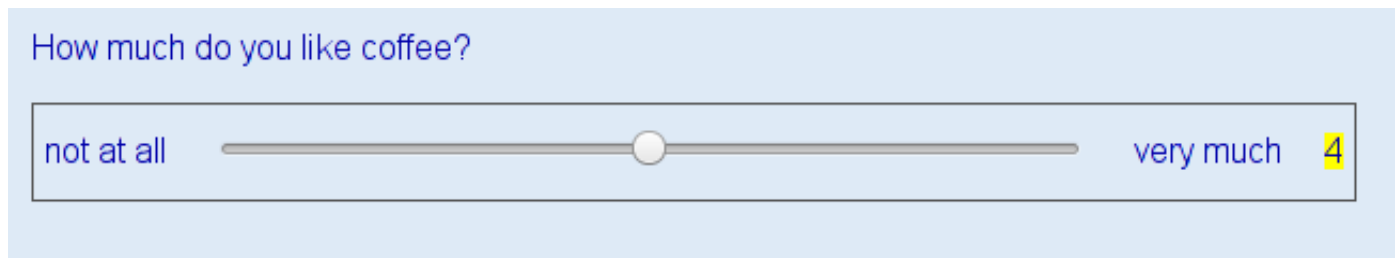
- *left* and *right* specify text left and right of the slider
- *start* specifies a start value
- *reverse* takes the reverse score

Example of *reverse*: If you have a scale from 1 to 7, and if the *reverse* option for an item is set, then if a participant selects 6, the score will be 2. This can be handy for reverse score items, which is fairly common in psychological questionnaires to prevent a response bias (e.g., a participant might tend to always score the maximum value).

Example with left, right, and start value

```
l: coffee
t: range
q: How much do you like coffee?
- {min=1,max=7,left=not at all,right=very much,start=4}
```

This will look like this (using Firefox):



Scale: Likert style scales

A common type is the Likert type of scale. For this, you need to make sure you first define a scale, which you can then use in one or more questions.

Just have a look at an example, it explains the basics best:

Example

```
scale: agree
- Disagree
- Neither agree nor disagree
- Agree

l: my_question
q: How much do you agree with each of the following
t: scale agree
- I see myself as Extraverted, enthusiastic
- I see myself as Critical, quarrelsome
- I see myself as Dependable, self-disciplined
- I see myself as Anxious, easily upset
```

Which will look like this for the participants:

How much do you agree with each of the following?

Item	Disagree	Neither agree nor disagree	Agree
I see myself as Extraverted, enthusiastic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I see myself as Critical, quarrelsome	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I see myself as Dependable, self-disciplined	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I see myself as Anxious, easily upset	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Click this button to continue



It is important to note that in order to use Likert scales, you need two different elements, namely the **scale** and the **scale question** type. The **scale** needs to be described only once and can, in principle, be used in many different questions.

The **scale** description gives a name to the scale and then you can list each of the scale-points, such as "Disagree" to "Agree" in the above example. Each of those scale-points starts with the "-" character (followed by a space). Once defined, you can use this scale in as many questions as you want (see examples below).

Now that you know the basics, let's go into some of the details:

1. You need to describe each scale you use **only once**. The idea is that if you want, you can reuse a scale over and over in as many questions as you want. A common mistake is that users think that the scale itself need to be described multiple times.
2. You can use many different scales and use them in different questions. An example is below (multiple scales).
3. If you wish, You can randomize the order of the items in a scale question by setting the **o: random** options (see example below).



In the example below, you see that you only need to define a scale once, and that you can use that scale in different questions. Further, it is shown that you can define more than one scale. I find it often useful to have the scale descriptions at the top of my survey script, which I have here.

Example with multiple scales and questions

scale: agree

- Disagree strongly
- Disagree moderately
- Disagree a little
- Neither agree nor disagree
- Agree a little
- Agree moderately
- Agree strongly

scale: like

- Do not like
- Like a bit
- Like very much

l: tipi

q: I see myself as ...

t: scale agree

- I see myself as Extraverted, enthusiastic
- I see myself as Critical, quarrelsome
- I see myself as Dependable, self-disciplined
- I see myself as Anxious, easily upset
- I see myself as Open to new experiences, complex
- I see myself as Reserved, quiet
- I see myself as Sympathetic, warm
- I see myself as Disorganized, careless
- I see myself as Calm, emotionally stable
- I see myself as Conventional, uncreative

l: socmedia

q: How much do you like the following social media?

t: scale like

- Facetube
- Tweeter
- Tinderbook
- Flopflix

l: people

t: scale agree

q: How much do you agree the following people?

- Gwendolyn
- Peter
- Greg
- Lisa

In the example below, we set the score values. Although in the example, the score values are the default ones, you could set them to different values. Further, you can even overrule the scoring of individual scale



question elements (see example). In that case, you must have the different scores separated by commas, and there must be exactly as many as in the scale (in this example 7).

Example

```
scale: agree
- {score=1} Disagree strongly
- {score=2} Disagree moderately
- {score=3} Disagree a little
- {score=4} Neither agree nor disagree
- {score=5} Agree a little
- {score=6} Agree moderately
- {score=7} Agree strongly

l: tipi
q: I see myself as ...
o: random
t: scale agree
- I see myself as Extraverted, enthusiastic
- I see myself as Critical, quarrelsome
- I see myself as Dependable, self-disciplined
- I see myself as Anxious, easily upset
- I see myself as Open to new experiences, complex
- I see myself as Reserved, quiet
- I see myself as Sympathetic, warm
- I see myself as Disorganized, careless
- I see myself as Calm, emotionally stable
- {reverse} I see myself as Conventional, uncreative
```

The scale question is shown as a "table" on the screen (i.e., there are rows and there are columns). There are different ways to control the way the table is being formatted:

- Use linebreaks in the scale items (using the html `
`).
- Use the option width. The parameter is the percentage of the scale table used for the item and the rest of the scale options will use the same width for each scale answer.

The example below has the line **o: width 50%**, which means that 50% is for the items text and the remaining for the 7 options. Each option will get the same $50/7=7.1\%$ of the table.

Example of scale width options

```

scale: agree
- {score=1} Disagree strongly
- {score=2} Disagree moderately
- {score=3} Disagree a little
- {score=4} Neither agree nor disagree
- {score=5} Agree a little
- {score=6} Agree moderately
- {score=7} Agree strongly

l: tipi
q: I see myself as ...
o: width 50%
t: scale agree
- I see myself as Extraverted, enthusiastic
- I see myself as Critical, quarrelsome
- I see myself as Dependable, self-disciplined
- I see myself as Anxious, easily upset
- I see myself as Open to new experiences, complex
- I see myself as Reserved, quiet
- I see myself as Sympathetic, warm
- I see myself as Disorganized, careless
- I see myself as Calm, emotionally stable
- {reverse} I see myself as Conventional, uncreative

```



In specifying the 50%, you the % sign is optional, just for readability. You can just put **o: width 50**, though.

Example of custom scoring for certain items

```

scale: agree
- {score=1} Disagree
- {score=2} Neutral
- {score=3} Agree

l: extroversion
q: I see myself as ...
t: scale agree
- I see myself as talkative
- I see myself as outgoing
- {reverse} I see myself as shy and quiet

```

The example below does **exactly** the same, but writes out the opposite scores.

Example of custom scoring for certain items

```
scale: agree
- {score=1} Disagree
- {score=2} Neutral
- {score=3} Agree

l: extroversion
q: I see myself as ...
t: scale agree
- I see myself as talkative
- I see myself as outgoing
- {score=3/2/1} I see myself as shy and quiet
```

Custom scores

For each scale question, you can use custom scores that deviate from the standard scoring. This is often done in questionnaires that require reverse scoring for some questions to prevent answering bias.

In PsyToolkit, there are two ways to do that.

1. You can just use the **reverse** option (or the shorter **rev**).
2. You can specify the scores. Below are two examples of this that have exactly the same outcome. The reverse option can be used in **scale** and in **range** questions.

Multiradio: Have multiple one-choice items in one questions

The **multiradio** item is not so common in questionnaires, but you see it sometimes. Using this type of questionnaire item, you can ask participants to choose between multiple pairs (or triplets, etc). The example explains what it does:

Example

```
l: forcedchoices
t: multiradio 2
q: For each pair of situations, pick the one you like best.
- Going to the movies
- Going to a concert
- Eating icecream
- Eating sorbet
- Listening to Bach
- Listening to Bieber
- Drinking wine
- Drinking whiskey
```

A full example can be viewed in the [Narcissistic Personality Inventory demo](http://www.psychtoolkit.org/survey-library/narcism-npi16.html) (<http://www.psychtoolkit.org/survey-library/narcism-npi16.html>).



The number in the multiradio command shows how many lines are taken together. In this case, we have two lines taken together.



You can use the option random

By default, the scores for each element in a pair starts with 1. You can set the scores using the option "scores".

Example with random order of items and a specific scoring for each element in a pair

```
l: forcedchoices
t: multiradio 2
o: scores 0 1
o: random
q: For each pair of activities, pick the one you like best.
- Going to the movies
- Going to a concert
- Eating icecream
- Eating hotdog
- Listening to Bach
- Listening to Bieber
- Drinking water
- Drinking wine
```

Images

Each question can show one or more images above the question. You can upload the images via the "files browse" button, followed by "save" under the questionnaire code. Make sure these images are not too big for a website. A good size is 400 by 300 pixels (note that standard images from cameras are way too big—you would need software to make them smaller).

Alternative to uploading, you can use the weblink of an image.

Example with an uploaded file named "inkblot.png"

```
l: rorschach
i: inkblot.png
q: What do you see?
t: textbox
- Describe what do you see in the drawing above
```

The participant will see the following:



What do you see?

Describe what do you see in the drawing above

[Click this button to continue](#)

You can also show multiple questions next to one another.

Example with an uploaded file named "inkblot.png"

```
l: preference
i: food1.jpg food2.jpg
q: Which picture do you like better?
t: radio
- The food in the left picture
- The food in the right picture
```

Example with an external image

```
l: preference
i: http://www.psychtoolkit.org/surveys.png
q: What does this image represent?
t: radio
- A survey
- A type of cheese
- None of the above
```

And finally, there you can add the options 'center'. This will center the image.

Example of a centered picture above the question

```
l: preference
i: {center} food1.jpg
q: Do you like this?
t: radio
- Yes
- No
```

Change the text of the button

You can not only change the language on the buttons, you can also change the text of the continue button for every question, if you wish. You do this with the b: operator, see the example below.

Example of a custom button text

```
l: food
t: radio
b: Can you please press the button to go to the next question?
q: What food did you just eat?
- Bread
- Rice
- Potatoes
```

Randomize question order

Sometimes, people want to randomize the questions of a part of the survey. This can be done easily in PsyToolkit. You need to put the items in your survey that you want in a random order each time the survey is being run between **random: begin** and **random: end**.



You can have as many sections as you want between **random: begin** and **random: end**.



Within a randomized-order section, you can enforce certain questions to always follow other questions (read here).



Make sure you have a space following the word **random:**, otherwise it will not work. Also, before and after the **random:** statement must be empty lines.



In a section of your survey that is being randomized, you typically should not have **jump** items. This because **jump** items assume a specific order of questions.

In the example survey below, the first and last question always are at the beginning and the end. In contrast, the questions food, drink and movie will come in a random order for each participant.

Example with three questions in random order

```
l: age
t: textline
q: How old are you?
- {min=18,max=100} Enter your age:
```

```
random: begin
```

```
l: food
t: radio
q: What food did you just eat?
- Bread
- Rice
- Potatoes
- Something else
```

```
l: drink
t: radio
q: What drink did you just have
- Tea
- Coffee
- Something else
```

```
l: movie
t: radio
q: What movie did you just watch
- Spiderman
- Citizen Kane
- The Orphan
- Something else
```

```
random: end
```

```
l: finalquestion
t: radio
q: Did you enjoy the questionniare
- yes
- no
```

You can use random: begin and random:end as often you like.

Example with two sections in random order

```
l: age
t: textline
q: How old are you?
- {min=18,max=100} Enter your age:
```

```
random: begin
```



```
l: food
t: radio
q: What food did you just eat?
- Bread
- Rice
- Potatoes
- Something else
```

```
l: drink
t: radio
q: What drink did you just have
- Tea
- Coffee
- Something else
```

```
random: end
```

```
l: internetquestion
t: radio
q: Are you ready to go on with some questions about the internet?
- Yes
- No
```

```
random: begin
```

```
l: visits
t: check
q: Which of the following websites have you recently visited?
- Google
- YouTube
- Twitter
- BBC News
```

```
l: computertype
t: check
q: Which of the following devices do you use to browse?
- Mobile phone
- Tablet
- Laptop
- Desktop
```

```
random: end
```

```
l: finalquestion
t: radio
q: Did you enjoy the questionnaire
- yes
- no
```

You can enforce certain questions to follow up on preceding questions. This makes sometimes sense, for example when three questions follow up on one another logically. If you want this, add the **o: link** option to that question, and that way it will always follow the question just before. This way you can link as many questions as you want. In the example below, there is a series of questions that will be randomized for each participant, but still, the question labeled "visits" will **always** follow the question "internetquestion". They are "linked".

Example with two sections in random order

```
l: age
t: textline
q: How old are you?
- {min=18,max=100} Enter your age:
```

```
random: begin
```

```
l: food
t: radio
q: What food did you just eat?
- Bread
- Rice
- Potatoes
- Something else
```

```
l: drink
t: radio
q: What drink did you just have
- Tea
- Coffee
- Something else
```

```
l: internetquestion
t: radio
q: Are you ready to go on with some questions about the internet?
- Yes
- No
```

```
l: visits
t: check
o: link
q: Which of the following websites have you recently visited?
- Google
- YouTube
- Twitter
- BBC News
```

```
l: computertype
t: check
q: Which of the following devices do you use to browse?
- Mobile phone
- Tablet
- Laptop
- Desktop
```

```
random: end
```

```
l: finalquestion
t: radio
q: Did you enjoy the questionnaire
- yes
- no
```

Scoring when running questionnaire using "set"

Sometimes, you might want to calculate a score. For example, if you are running a depression inventory, you might want to calculate a score and save that to the datafile, or based on that score ask further questions. That is all possible. For this, there is a special questionnaire item named "set".

In summary:

- **set** is a questionnaire item and must have a label
- **set** can calculate take the answers and scores of other questions and add these, average these, etc.
- **set** will set the label to the value and this can be used in conditional jumps, etc.

Example of set

```
l: food
t: radio
q: How much did you like the FOOD you had before?
- disliked it very much
- disliked it
- neither liked nor disliked
- liked it
- liked it very much

l: drink
t: radio
q: How much did you like the DRINK you had before?
- disliked it very much
- disliked it
- neither liked nor disliked
- liked it
- liked it very much

l: satisfactionscore
t: set
- sum $food $drink

l:
t: jump
- if $satisfactionscore > 2 then goto endquestion

l: why_dislike
t: textbox
q: We are sorry you did not like the food. If you wish, please leave a comment
here
- Please leave a comment about why you were not so happy with the dinner

l: endquestion
t: textline
q: If you wish, please leave your contact number
- Contact phone number
```

Sometimes, you want to sum all the questions starting with MyItems. In the following, there are three questions in MyScale and there is a simple way of referring to them using a placeholder character. In the example below, the set adds the answers of all the questions starting with MyItems. There are some extensive examples of this available in the survey library.



You can only use the placeholder as the last character, like in the examples below.

Example of set with placeholder

```
l: MyItems1
t: radio
q: How do you feel today?
- good
- neutral
- not so good

l: MyItems2
t: radio
q: How did you feel yesterday?
- good
- neutral
- not so good

l: MyItems3
t: radio
q: How do you feel last week?
- good
- neutral
- not so good

l: myscore
t: set
- sum $MyItems*

l: score1and2
t: set
- sum $MyItems1 $MyItems2
```

You can also use a different type of function than "sum", namely "mean":

Example of mean

```
l: myscore
t: set
- mean $MyItems*
```

You can also use "calc" to use more complex calculations:

Example of calc

```
l: myscore
t: set
- calc ( $MyItems1 + $MyItems2 ) / 2.0
```



If you have questions with scale items, you can get the score of a dot notation. Look for an advanced example of this in [this survey in the library](http://www.psytoolkit.org/survey-library/political-conservatism.html) (<http://www.psytoolkit.org/survey-library/political-conservatism.html>).

There are a couple of rules that you need to stick to when using "calc":

- All operators need to use spaces between them (like in the example above)
- If you want to make sure that things are not rounded too much in a division like the one above, you need to make sure that you use numbers decimal point (2.0 instead of just 2).

Running different sections of questionnaire for different participants

Some participants want to make sure that different people who click the same questionnaire/survey link actually do different questionnaires. This is sort of a random assignment of participants to a condition. This can be done. These are the steps to take:

1. Decide how many different versions you want? Two, or maybe even more?
2. Choose a random number
3. Jump to a section of the questionnaire depending on that random number.

Below are two examples, one for just a survey, and second a way to use this for counterbalancing groups of participants doing an experiment.

In the following example, the questionnaire sets a random item (chooserandom) to 1 or 2, and based on that, the questionnaire jumps to the appropriate part of the questionnaire.

Example of random assignment of questionnaires

```
l: choosrandom
t: set
- random 1 2

l:
t: jump
- if $choosrandom == 1 then goto drink_set1
- if $choosrandom == 2 then goto food_set2

## survey 1 about drinks #####

l: drink_set
t: radio
q: Do you like cola?
- Yes
- No

l: drink2
t: radio
q: Do you like fanta?
- Yes
- No

l:
t: jump
- goto endquestion

## survey 2 about food #####

l: food_set
t: radio
q: Do you like bread?
- Yes
- No

l: drink2
t: radio
q: Do you like pancakes?
- Yes
- No

l: endquestion
q: Did you like this survey?
- Yes
- No
```

Sometimes, you have two different versions of an experiment, say with different block order, and you want to automatically give this to different people.

For this example, we will have two different experiments, **experiment_order1** and **experiment_order2**. We will randomly assign participants to one of these conditions.

Example of counterbalancing experiments

```
l: choose_a_number
t: set
- random 1 2

l:
t: jump
- if $choose_a_number == 1 then goto Group1
- if $choose_a_number == 2 then goto Group2

l: Group1
t: experiment
- experiment_order1

l: alwaysjump
t: jump
- goto GoOn

l: Group2
t: experiment
- experiment_order2

l: GoOn
t: radio
q: How did you like the experiment
- I liked it
- I did not like it
```

You can also use random differently, choosing numbers from a set of numbers:

Example of using "random from"

```
l: choose_a_number
t: set
- random from 1 2 4 5
```

Last updated 2016-08-16 10:43:05 CEST