

Tutorial

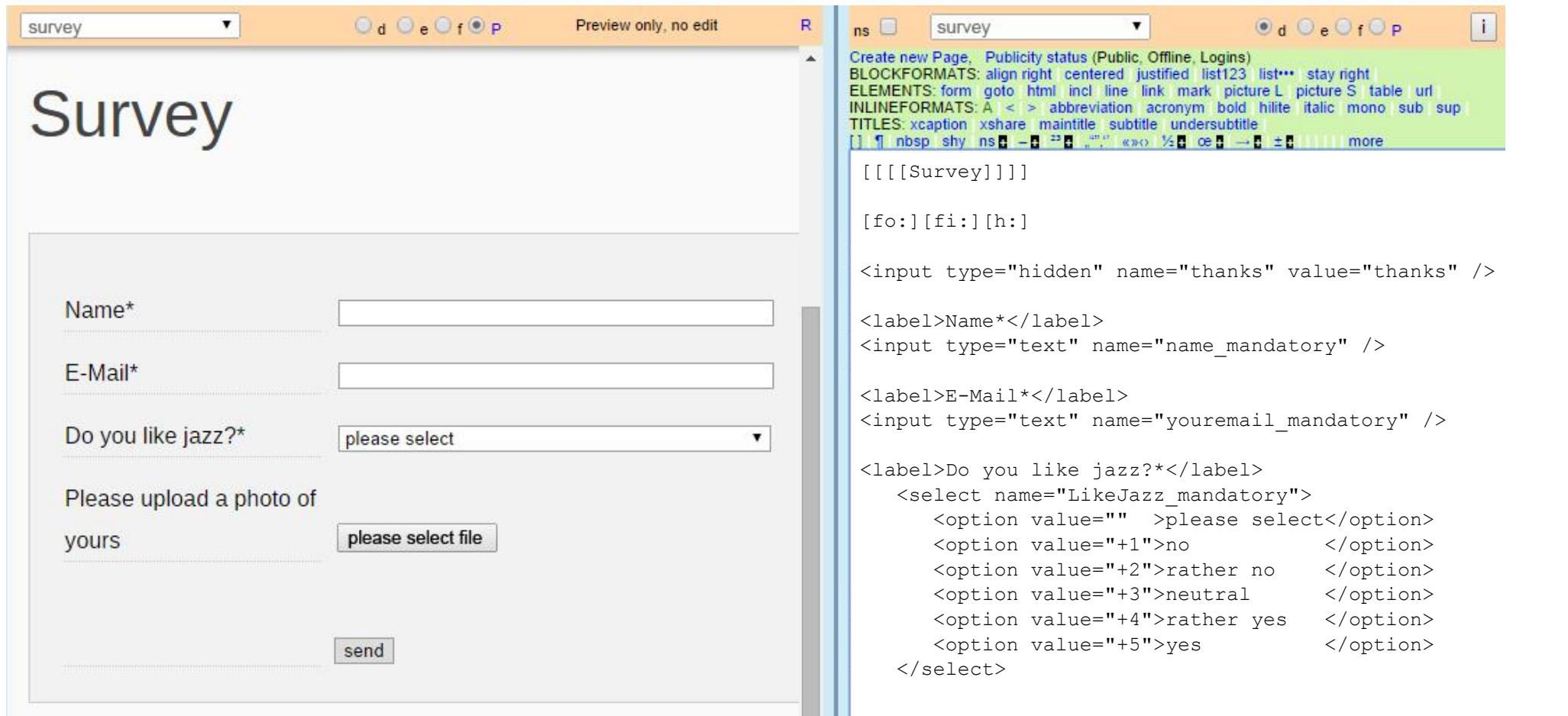
Step by step from the most easy query for rendering form data to a very nice output-table

Overview

1. Create an HTML form, e. g. a survey
2. Let your friends fill their data into the HTML form. Of course you can import data from excel
3. See how the data is stored in the u5CMS's database and presented as table and chart
4. And now, finally: Render the received data on a page of your u5CMS-driven website!

1. Create an HTML form, e. g. a survey

Go to the u5CMS-backend¹ and create a new page² containing an HTML form³ with this code:



The image shows two side-by-side screenshots from the u5CMS-backend. The left screenshot displays a preview of a survey form titled "Survey". The form includes fields for "Name*", "E-Mail*", and "Do you like jazz?*" (a dropdown menu with "please select" selected). Below these is a section for uploading a photo, labeled "Please upload a photo of yours", with a "please select file" button. A "send" button is at the bottom. The right screenshot shows the HTML code for the form, with a green header bar containing menu options like "Create new Page", "Publicity status", "BLOCKFORMATS", "ELEMENTS", "INLINEFORMATS", and "TITLES". The code includes a hidden input for "thanks", labels for the form fields, a text input for "name_mandatory", a dropdown for "LikeJazz_mandatory" with options "no", "rather no", "neutral", "rather yes", and "yes", and a "send" submit button.

1: Open the u5CMS-backend: See <http://yuba.ch/u5cms/manual> page 4

2: Create a new page: See <http://yuba.ch/u5cms/manual> page 32

3: Create HTML forms: See <http://yuba.ch/u5cms/manual> page 35

The look of the "select file"-button can differ.

u5CMS tutorial about <http://yuba.ch/u5cms/renderformdata>

```
<label>Please upload a photo of yours</label>
```

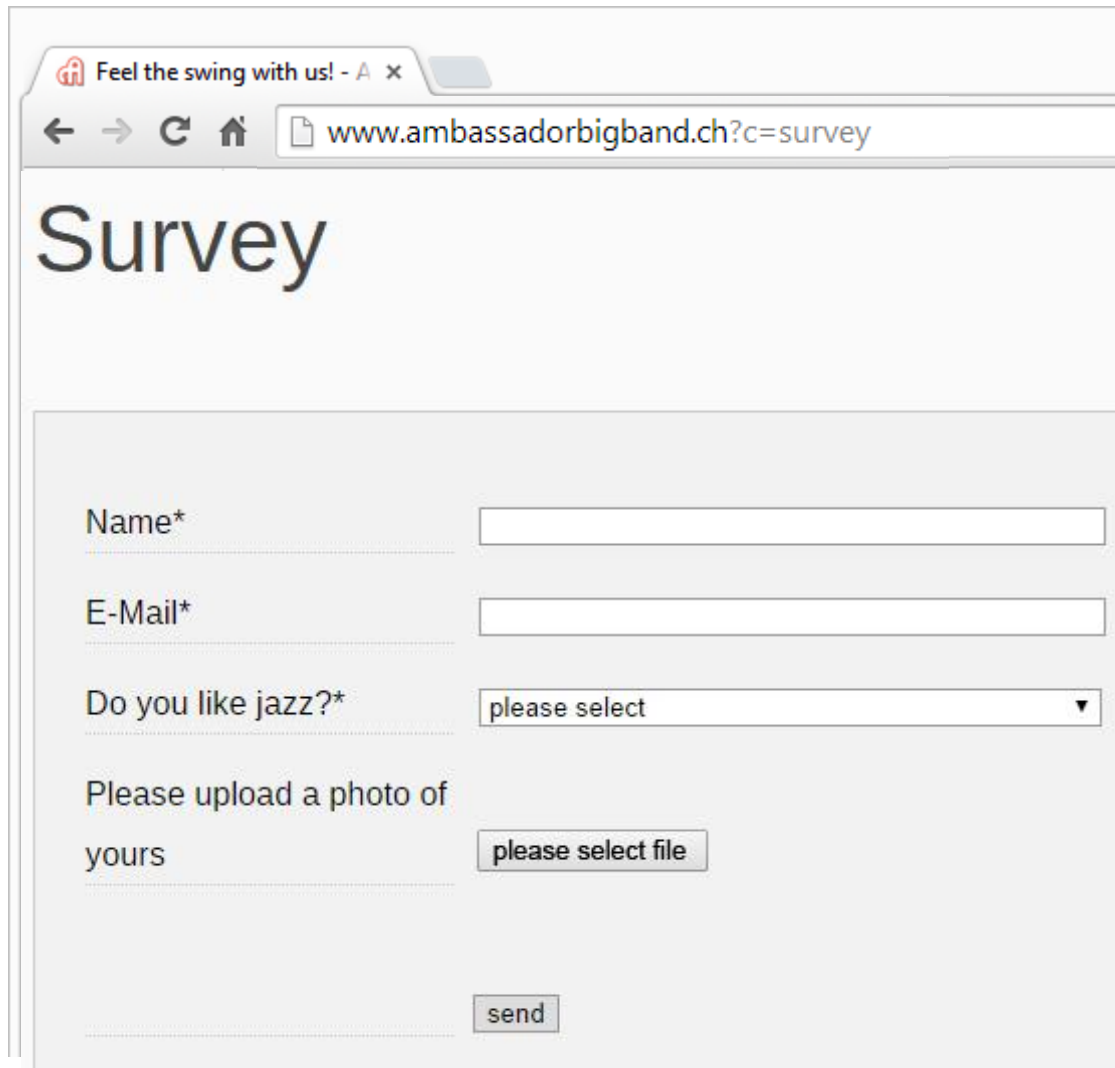
```
<script src="upload"></script>
```

```
<label>&nbsp;</label><input type="submit" value="send" />
```

```
[:h] [:fi] [:fo]
```

2. Let your friends fill their data into the HTML form

If you named the page “survey”, the URL is <http://www.yoursite.ext?c=survey> or <http://www.yoursite.ext/anydirectory?c=survey>



The image shows a web browser window with the following elements:

- Browser tab: "Feel the swing with us! - A x"
- Address bar: "www.ambassadorbigband.ch?c=survey"
- Page title: "Survey"
- Form fields:
 - Name* (text input)
 - E-Mail* (text input)
 - Do you like jazz?* (dropdown menu with "please select" and a downward arrow)
 - Please upload a photo of yours (file selection button labeled "please select file")
- Submit button: "send"

The look of the “select file”-button can differ.

3. See how the data is stored in the u5CMS's database and presented as table and chart

You'll find the received form data in the u5CMS-backend (there you also find the import-function to import data from excel)

The screenshot illustrates the u5CMS backend interface for handling form data. It is divided into several key sections:

- Code Editor:** Shows the HTML code for a survey form with fields for Name, E-Mail, and a question "Do you like jazz?".
- Special functions:** A menu with options like "form data" (to view data as table and charts) and "backup content".
- Form Data:** A dropdown menu with options "html", "chart", and "xls".
- Chart:** A bar chart for the question "Do you like jazz?*" showing the distribution of responses: "rather no" (1, 25%), "neutral" (1, 25%), and "yes" (2, 50%).
- Data Table:** A table listing individual form submissions with columns for ID, Status, Notes, name, youremail, question, userupload1, Sent, and IP.

ID	Status	Notes	name*	youremail*	question*	userupload1	Sent	IP
31	1) new		Satya	satya@microsoft.com	-3	.jpg	2014.10.11 16:40:02	84.227.238.170
30	1) new		Tim	tim@apple.com	-5	.jpg	2014.10.11 16:39:13	44.255.137.88
29	1) new		Steve	steve@apple.com	-5	.jpg	2014.10.11 16:38:23	112.127.254.17
28	1) new		Bill	bill@microsoft.com	-2	.jpg	2014.10.11 16:37:41	89.101.258.109

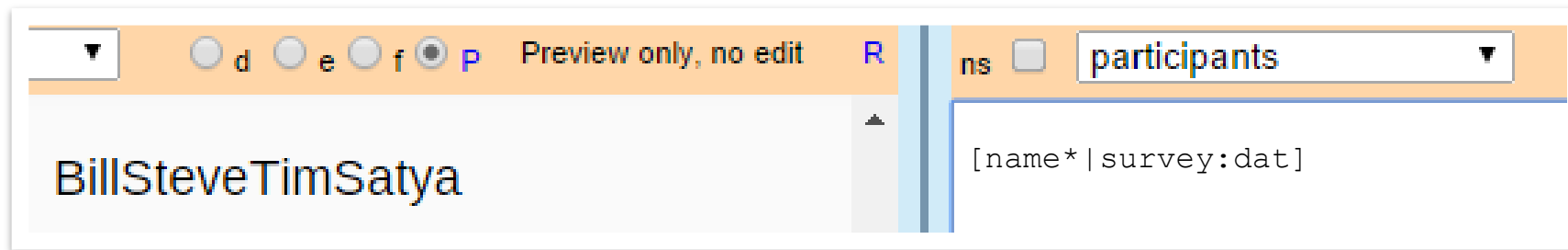
4. Render the received data on a page of your u5CMS-driven website!

Now, in the u5CMS-backend, create a new empty page named “participants”.

The command to access received form data is `[fieldname|formname:dat]` (see <http://yuba.ch/u5cms/renderformdata>)

In our example the formname is **survey** and we asked for the **name**. You remember the HTML-form? There, the field for the name was coded as `<input type="text" name="name_mandatory" />`

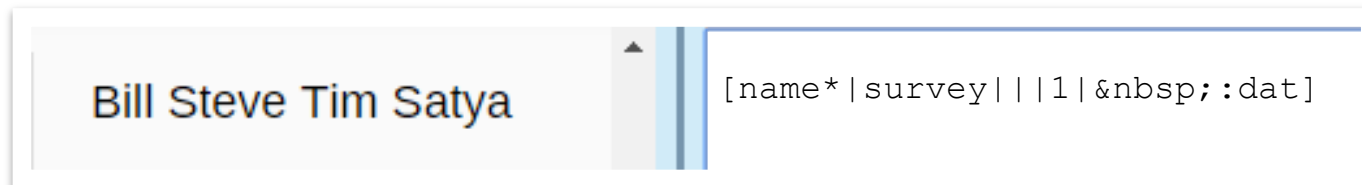
Therefore, we have to write `[name_mandatory|survey:dat]` or shorter `[name*|survey:dat]` then all the names entered in the HTML form are shown:



Be aware that this shows *all* records, even deleted ones. We introduce now a parameter which says “only records with status 1” (=new ones) or “only records with status 4” (=done, this status you have to give manually, so you could do a review before a record is rendered (published) on your site). It is the 5th parameter in the command: `[name*|survey||1:dat]` says “only records with status 1” or `[name*|survey||4:dat]` says “only records with status 4”.

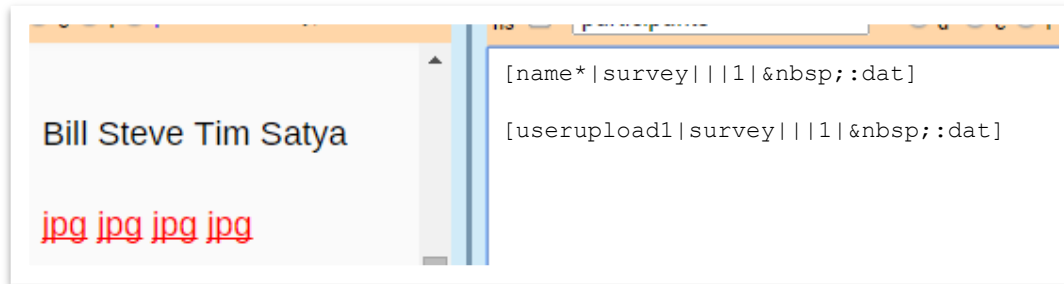


As 6th parameter you can enter a piece of HTML, which is outputted *before* every record, e. g. a no-break space ` `;

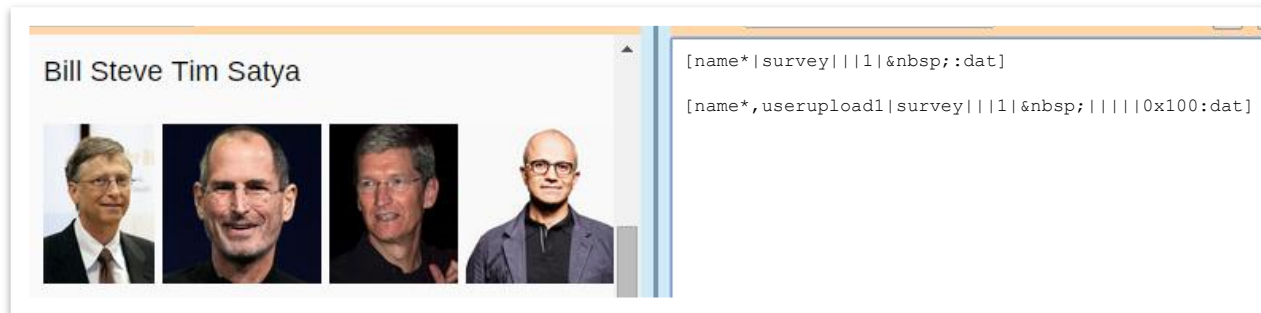


formname is the name of the form-carrying page. E.g. survey is the name of the page on which the form is, i.e. the value of the GET parameter c of its URL.

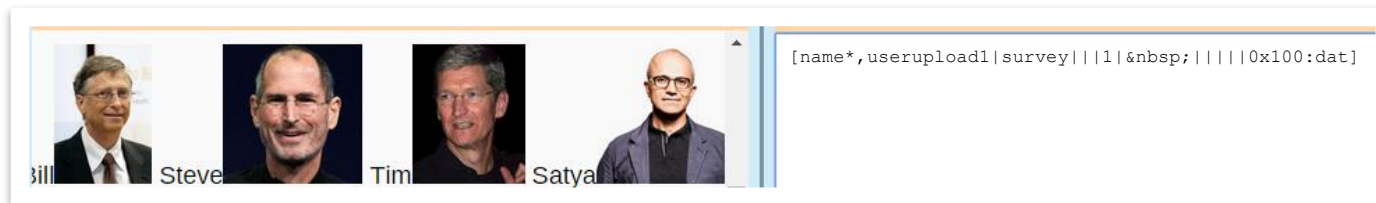
As next step we also output the uploaded images. You have to know the name of the field! You remember the HTML-form, the upload was just this piece of HTML: `<script src="upload"></script>` and we do not see a name here. So we have to know that the fieldname of the first upload-script in a form is **userupload1** and if it was `<script src="upload_mandatory"></script>` then it was **userupload1_mandatory** or short **userupload1***



If you do not want to render links to the images but the images themselves, add the 11th parameter telling the system how big the images shall appear, let's say any (=0) width but exactly 100 pixels high:



Now see what happens if we combine these two lines of code into one line of code. In other words, the first parameter of the command can be a comma-separated list of fieldnames:



Let's add the other field names in the comma-separated list of the first parameter:

The image shows a side-by-side comparison of a web application's data rendering and its underlying code. On the left, a preview window titled 'participants' shows a list of four participants, each with a profile picture and an email address:

- Billbill@microsoft.com+2 (with a picture of Bill Gates)
- Stevesteve@apple.com+5 (with a picture of Steve Jobs)
- Timtim@apple.com+5 (with a picture of Tim Cook)
- Satyasatya@microsoft.com+3 (with a picture of Satya Nadella)

On the right, a code editor window titled 'participants' shows the corresponding code for the first parameter of a data list:

```
[name*, youremail*, LikeJazz*, userupload1|survey|||1|&nbsp;| |||0x100:dat]
```

Now, let's transform this into an HTML-table.



We have to start with a piece of HTML, namely the opening table-tag `<table>` and because it is HTML and not u5CMS-syntax, it is between `[h:]` and `[:h]`. And at the end we set the closing table-tag `</table>`.

Further, we replace the content of the 6th parameter (until now this was a no-break space ` `;) with the HTML-tags `<tr><td>` what means "new table-row, new table-cell" (you remember, the 6th parameter says what to output BEFORE each record).

Then we introduce the 7th parameter which defines the HTML to be outputted AFTER each record: `</td></tr>` says "end of table-cell, end of table-row".

And, the 8th parameter says what to output BETWEEN the fields (name*, youremail*, ...) WITHIN a record, here `</td><td>` meaning "end of table-cell, start a new table-cell" (all these parameters are described on <http://yuba.ch/u5cms/renderformdata>).





The screenshot shows two side-by-side windows from the u5CMS interface. The left window, titled 'participants', is in 'Preview only, no edit' mode and displays a table with three rows. The first row contains 'Bill', 'bill@microsoft.com', '+2', and a photo of Bill Gates. The second row contains 'Steve', 'steve@apple.com', '+5', and a photo of Steve Jobs. The third row is partially visible with a photo of another man. The right window shows the raw HTML code for the table, enclosed in `[h:]<table>[:h]` tags. The code is: `[name*,youremail*,LikeJazz*,userupload1|survey|||1|<tr><td>|</td></tr>|</td><td>|||0x100:dat]`.

Bill	bill@microsoft.com	+2	
Steve	steve@apple.com	+5	
			

```
[h:]<table>[:h]
[name*,youremail*,LikeJazz*,userupload1|survey|||1|<tr><td>
|</td></tr>|</td><td>|||0x100:dat]
[h:]</table>[:h]
```


As last step we make the ultimate beautifying. Let's change the values +1, +2, +3, +4 and +5 to words and give them a background color. We do this with JavaScript. As you see in the example, it is important to name the table of interest with a class, therefore we change `[h:]<table>[:h]` to `[h:]<table class="JazzTable">[:h]`

The screenshot shows a web application interface with a table of participants and a code editor. The table has columns for name, email, a score, and a photo. The code editor shows the HTML structure and JavaScript used to style the table based on the scores.

Bill	bill@microsoft.com	+2	
Steve	steve@apple.com	+5	
Tim	tim@apple.com	+5	
Satya	satya@microsoft.com	+3	

```

[h:]<table class="JazzTable">[:h]
[h:]<th>Name</th><th>E-Mail</th><th>Answer</th><th>Photo</th>[:h]
[name*,youremail*,LikeJazz*,userupload|survey|||1|<tr><td>|</td></tr>|</td><td>|
||0x100:dat]
[h:]</table>[:h]

[h:]
<script>
tds=document.getElementsByClassName('JazzTable')[0].getElementsByTagName('td');
for(i=0;i<tds.length;i++) {

if(tds[i].innerHTML=='+1')tds[i].style.background='lightred';
if(tds[i].innerHTML=='+2')tds[i].style.background='orange';
if(tds[i].innerHTML=='+3')tds[i].style.background='lightblue';
if(tds[i].innerHTML=='+4')tds[i].style.background='lightyellow';
if(tds[i].innerHTML=='+5')tds[i].style.background='limegreen';





tds[i].innerHTML=tds[i].innerHTML.replace("+1","no");
tds[i].innerHTML=tds[i].innerHTML.replace("+2","rather no");
tds[i].innerHTML=tds[i].innerHTML.replace("+3","neutral");
tds[i].innerHTML=tds[i].innerHTML.replace("+4","rather yes");
tds[i].innerHTML=tds[i].innerHTML.replace("+5","yes");

}
</script>
[:h]

```

Addendum: with this line we added table headings

You cannot see the effects in the preview, but on the live website:

Bill	bill@microsoft.com	rather no	
Steve	steve@apple.com	yes	
Tim	tim@apple.com	yes	
Satya	satya@microsoft.com	neutral	

Very important: When pasting the above code, make sure really no physical linebreak is added between `</td><td>` and `||0x100 !!!`