

LimeSurvey-Tutorial

Gimmicks
und andere Spielereien

September 2021 – Februar 2024

Inhaltsverzeichnis

Vorbemerkung.....	1
1. Informationen über den benutzten Browser.....	2
1.1. Anwendungsbeispiel.....	2
1.2. Implementierung.....	3
2. "Weiter"-Button erscheint erst nach einer gewissen Zeit.....	5
2.1. Anwendungsbeispiel.....	5
3. Graphiken aufgrund von Ergebnissen der gegebenen Antworten (z.B. Scores).....	7
3.1. Ergebnisse des einzelnen Teilnehmers.....	7
3.1.1. Beispiel mit der Graphik-Bibliothek „HighCharts“ (www.highcharts.com“).....	7
3.1.2. Beispiel mit Google Charts.....	10
3.1.3. Beispiel mit Chartjs.....	12
3.2. Ergebnisse der gesamten Umfrage.....	14
3.2.1. Anwendungsbeispiel 1.....	15
3.2.2. Anwendungsbeispiel 2.....	18
3.2.3. Implementierung.....	19
3.2.3.1. Plugin „statFunctions“, enthalten ab LimeSurvey-Version 5.x.....	19
3.2.3.2. Plugin „getStatInSurvey“.....	20
4. Lückentexte.....	22
4.1. Lückentext als Drop-Downs.....	22
4.2. Lückentext mit „kurzen Texten“.....	23
5. Eingabemasken.....	26
5.1. Mit der Bibliothek „Inputmask“ von Robin Herbots.....	26
5.1.1. Anwendungsbeispiel.....	26
5.1.2. Implementierung.....	26
5.2. Mit der Bibliothek „jquery mask“ von Igor Escobar.....	28
5.2.1. Anwendungsbeispiel.....	28
5.2.2. Implementierung.....	28
6. Größere Darstellung von Bildern.....	29
6.1. Anwendungsbeispiel.....	29
6.2. Implementierung.....	30
7. Lupenfunktion.....	31
8. Bewertung bzw. Auswahl mittels Pins in Bildern.....	33
8.1. Bewertung mittels Pins (Points of Interest) (positiv - negativ).....	33
8.2. Setzen einer Markierung in ein Bild.....	35
9. Cardsort.....	36
10. IAT (Implicit Association Test).....	37
10.1. Anwendungsbeispiel.....	37
10.2. Implementierung.....	38
11. So etwas wie Conjoint.....	39
11.1. Darstellung mittels (statischer) HTML-Tabellen im Fragetext.....	39
11.1.1. Layout-Möglichkeiten.....	39
11.1.2. Ideen zur Darstellung der Optionen.....	41

11.1.2.1. Fest verdrahtete Optionen.....	41
11.1.2.2. Zufällige Anzeige von n Kombinationen aus einem Pool von m.....	41
11.1.2.3. Vorher berechnete Kombinationen.....	42
11.1.3. Implementierung.....	43
11.2. Darstellung mittels Slider-Bibliotheken.....	45
11.2.1. Tiny Slider.....	45
11.2.1.1. Layout-Möglichkeiten.....	45
11.2.1.2. Implementierung.....	46
11.2.2. Slick-Bibliothek.....	48
11.2.2.1. Layout-Möglichkeiten.....	48
11.2.2.2. Implementierung.....	49
12. Erläuterungen mittels Tooltips, Popover, Akkordeon, Modal-Dialog.....	53
12.1. Collapse (oder Akkordeon).....	53
12.2. Modaler Dialog.....	54
12.3. Popover als Klick-Element in Matrix.....	55
12.4. Tooltips und Popover im Frage- und Antworttext.....	56
12.5. Modal-Dialoge, Alerts als Reaktion auf „Klick“	57
12.5.1. Modal-Dialoge als Reaktion auf Klick.....	57
12.5.2. Alerts als Reaktion auf Klick.....	59
13. Farben, Farben, Farben.....	61
13.1. Farbige Radio-Buttons und Checkboxes.....	61
13.2. Dual Matrix.....	63
13.2.1. Mit Trennern.....	63
13.2.2. Mit farbigen Bereichen.....	64
13.3. Matrix mit Färbung von Header und Zellen.....	65
13.3.1. Matrix mit Header und Farbgebung beim Überstreichen und bei Auswahl.....	65
13.3.2. Matrix mit farbigen Spalten.....	68
14. Mehrere Fragen nebeneinander.....	70
14.1. Anwendungsbeispiel.....	70
14.2. Implementierung.....	71
15. Chat-Funktion.....	72
16. Der „ajax-call“	74
16.1. Vorbemerkung.....	74
16.2. Beispiel (Anzeige eines selbst hochgeladenen Bildes).....	75
16.3. Implementierung.....	77
17. Anzeige einer Karte aufgrund der IP-Adresse.....	80
17.1. Anwendungsbeispiel.....	80
17.2. Implementierung.....	81
18. Anhang.....	84
18.1. Iss-Export einer Beispiel-Datei.....	84
18.1.1. zip-Datei der benötigten „highslide“-Dateien.....	84
18.2. Iss-Export einiger Beispiele zur Nutzung von „getStatInSurvey“	84
18.3. Iss-Export einiger Beispiele zur Nutzung von „statCount“ (nur ab Version 5.x).....	84

Vorbemerkung

Diese kleinen Tutorials sollen Möglichkeiten aufzeigen, durch Einsatz von javascript und css Effekte bei der Darstellung von Fragen zu zeigen, die den Teilnehmern die Beantwortung erleichtern.

Viele dieser Beispiele kann man auch auf „normalem“ Weg abfragen; doch ein bisschen „Gamification“ steigert sicher eher die Lust, den Fragebogen zu Ende zu bringen als 13 Itembatterien mit jeweils 15 Teilfragen.

In diesem Tutorial werde ich mir selbst etwas untreu.

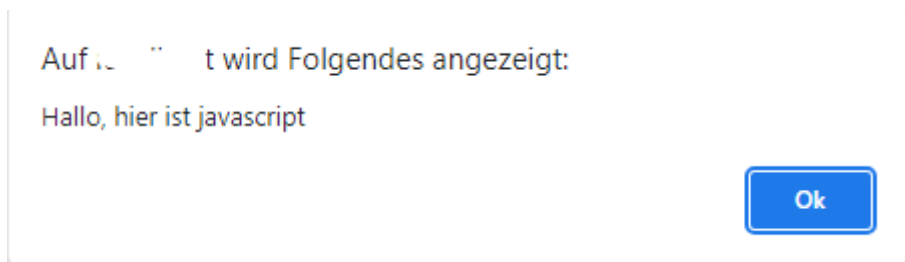
Hier wird es einige Beispiele geben, in denen Plugins bzw. Fragenvorlagen installiert werden.

Hier werden auch nicht mehr die gesamten javascript codes gezeigt, nur kleine, interessante, wichtige Details. Um die vollständigen scripte zu erhalten, lade man die im Anhang befindliche Datei herunter und importiere sie.

Ein einfacher Test, ob man javascript benutzen kann, ist dies in den Quellcode der ersten Frage einzufügen

```
<script type="text/javascript" charset="utf-8">
$(document).on('ready ajax:scriptcomplete',function(){
    alert("Hallo, hier ist javascript");
});
</script>
```

Wenn dann dieses Fenster aufploppt, ist alle in Ordnung; wenn nicht, ...



Achtung:

Einige Beispiele sind (noch) nicht LimeSurvey 6.x kompatibel; gerade Dinge, die bootstrap-Elemente benutzen.

LimeSurvey 6.x. beruht auf bootstrap 5, vorher war es bootstrap 3

Und dann noch dieses:

Solutions, code and workarounds presented in these text are given without any warranty, implied or otherwise.

Lösungen, Code und „Work-arounds“, die in diesem Text präsentiert werden, werden ohne jegliche stillschweigende oder sonstige Gewährleistung gegeben.

1. Informationen über den benutzten Browser

1.1. Anwendungsbeispiel

Oft ist es hilfreich zu wissen, mit welchem Browser der Teilnehmer die Umfrage beantwortet. Dies kann aus dem Wissen resultieren, dass bestimmte Browser keine Unterstützung liefern für ein Feature, welches man bei einer Frage eingebaut hat. Oder dass bestimmte Fragen schlecht dargestellt werden bei kleinen Devices (smartphones).

Dann kann man zwei Layouts einer Frage erstellen, und aufgrund der Information entsprechend die eine oder andere einblenden.

OS	Windows 7
Browser	Chrome 93 (93.0.4577.63)
Mobile	false
Bildschirmgröße	1920 x 1080
Viewport Breite	1920
Viewport Höhe	922
User Agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36

Hier sind einmal die wichtigsten Informationen in einer Frage vom Typ „mehrfache kurze Texte“ dargestellt.

1.2. Implementierung

Wie gesagt, in einer Frage vom Typ „mehrfache kurze Texte“ wird dieser javascript-Code in den Fragetext (Quellcode-Modus) eingefügt.

```
<script type="text/javascript" charset="utf-8">

  /**
   * JavaScript Client Detection
   * (C) viazenetti GmbH (Christian Ludwig)
   */

  (function (window) {
    /* Es werden die Daten ausgelesen, analysiert */
    /* -----*/
    /* und dann hier im Objekt „jscd“gespeichert

    window.jscd = {
      screen: screenSize,
      browser: browser,
      browserVersion: version,
      browserMajorVersion: majorVersion,
      mobile: mobile,
      os: os,
      osVersion: osVersion,
      cookies: cookieEnabled,
      flashVersion: flashVersion,
      vieww : vpw,
      viewh: vph
    };
  })(this));

  $(document).ready(function() {

    // Hide this question (remove double slash below to hide)
    //$('#question{QID}').hide();

    var thisQuestion = $('#question{QID}');

    // Fill the array
    $('input[type=text]:eq(0)', thisQuestion).val(jscd.os + ' ' + jscd.osVersion);
    $('input[type=text]:eq(1)', thisQuestion).val(jscd.browser + ' ' + jscd.browserMajorVersion +
      ' (' + jscd.browserVersion + ')');
    $('input[type=text]:eq(2)', thisQuestion).val( jscd.mobile );
    $('input[type=text]:eq(3)', thisQuestion).val( jscd.screen );
    $('input[type=text]:eq(4)', thisQuestion).val( jscd.vieww );
    $('input[type=text]:eq(5)', thisQuestion).val( jscd.viewh );
    $('input[type=text]:eq(6)', thisQuestion).val( navigator.userAgent );

  });
</script>
```

Wie man sieht, sammelt die erste Funktion alle Daten und speichert sie im Objekt „jscd“.

Am Ende werden die gewünschten Teile in den einzelnen Teilfragen gespeichert; mit „eq(x)“ werden die einzelnen Teilfragen ausgewählt, Start ist bei „0“.

Das wissend kann man also später einfach testen:

Arbeitet der Teilnehmer mit einem mobilen Gerät? `Q1_SQ003=="true"`

Ist der Bildschirm breiter als 800px? `Q1_SQ005>800`

Natürlich wird diese Frage in einer realen Umfrage nicht angezeigt; also werden in dieser Zeile

`//$('#question{QID}').hide();`

die Schrägstriche entfernt.

Diese Frage darf nicht alleine in einer Gruppe sein und die Umfrageeinstellung nicht „Frage für Frage“.

Ansonsten sieht der Teilnehmer einen leeren Bildschirm mit dem „Weiter“-Button.

2. "Weiter"-Button erscheint erst nach einer gewissen Zeit

2.1. Anwendungsbeispiel

Es gibt unzählige Beispiele, warum man dem Teilnehmer eine Seite eine Mindestzeit lang präsentieren möchte.

Er soll einen Text lesen, ein Bild ansehen, ohne sofort weiterzuklicken.

Dazu kann man den „Weiter“-Button für eine gewisse Zeit verstecken.

Der Standard-Code dafür ist:

```
<script type="text/javascript" charset="utf-8">
  $(document).ready(function(){

    // Verstecke den „Weiter“-Button
    $('#ls-button-submit').hide();

    setTimeout(function () {
      // Zeige den „Weiter“-Button
      $('#ls-button-submit').show()
    }, 10000); // Nach 10 Sekunden, Wert ist in Millisekunden
  });
</script>
```

Natürlich kann man auch andere Dinge tun:

- `$('#ls-button-previous').hide();` und `$('#ls-button-previous').show();`
versteckt und zeigt den „Zurück“-Button.
- `$('#ls-button-submit').prop('disabled',true);` und `$('#ls-button-submit').prop('disabled',false);`
versteckt nicht, sondern setzt den Button auf „disabled“ und wieder zurück.
- `$('#ls-button-submit').show().trigger('click');`
der „Weiter“-Button wird nicht nur wieder gezeigt, sondern sofort automatisch zur nächsten Frage gegangen (der „Klick“ wird simuliert). Dies ist besonders sinnvoll, wenn man einen Text, ein Bild NUR für eine bestimmte Zeit anzeigen will.

Normalerweise wird ein Video gestartet, der „Weiter“-Button versteckt, und erst nach Ende des Videos wieder gezeigt. So muss der Teilnehmer das Video zur Gänze anschauen (oder zwischendurch in die Küche gehen und sich einen Kaffee holen)

Da manche Browser Videos mit Ton nicht automatisch starten, ist im Beispiel der „Zurück“-Button versteckt; so dass man auf jeden Fall weiterkommt.

Hier ist dann der gesamte Code im Fragetext (Quellcode-Modus) mit auskommentierten (//) Alternativen.

```
<div class="embed-responsive embed-responsive-16by9">
<video autoplay="autoplay" id="myvideo"><source src="https://www.Mafosurvey.de/userfiles/happypeanuts.mp4"
type="video/mp4" /> browser unterstützt dieses Format nicht.</video>
</div>
<script type="text/javascript" charset="utf-8">
  $(document).ready(function() {

    // Hide the buttons
    $('#ls-button-submit').hide();
  //  $('#ls-button-previous').hide();

    var vid = document.getElementById("myvideo");
    vid.volume = 0.3;
    vid.onended = function() {
      // Show the "Previous" button
  //    $('#ls-button-previous').show();
      $('#ls-button-submit').show();
  //    $('#ls-button-submit').show().trigger('click');
    }
  });
</script>
```

Wie gefallen die Peanuts?



Bitte wählen Sie eine der folgenden Antworten:

sehr gut

gut

geht so

geht nicht so

schlecht

sehr schlecht

3. Graphiken aufgrund von Ergebnissen der gegebenen Antworten (z.B. Scores)

3.1. Ergebnisse des einzelnen Teilnehmers

Sicherlich ist es ganz anschaulich, wenn man einem Teilnehmer am Schluss seine Ergebnisse in einer Graphik präsentiert.

Zunächst erstellen wir ein paar Daten.

Matrix, um irgendwelche Werte für die Graphik zu erhalten						
	miserabel	sehr schlecht	schlecht	mittel	gut	sehr gut
Scorefrage 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Scorefrage 2	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scorefrage 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Scorefrage 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Scorefrage 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Scorefrage 6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Scorefrage 7	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scorefrage 8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scorefrage 9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scorefrage 10	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3.1.1. Beispiel mit der Graphik-Bibliothek „HighCharts“ (www.highcharts.com)



Zunächst ist in den Fragetext ein bootstrap-Container mit einer „row“ und einer „column“ eingefügt. In dieser befindet sich ein „Panel“, damit eine Überschrift über dem Gesamtbild möglich ist. Im „Panel Body“ sind nun wieder eine „row“ und zwei „columns“ für die zwei Graphiken „chart1“ und „chart2“

```
<div class="container" style="margin-top:50px;">
  <div class="row" style="margin-right:4%;margin-left:-1%;">
    <div class="col-sm-12">
      <div class="panel panel-default">
        <div class="panel-heading">Anzahl Antworten pro Option und Gesamtscore</div>
        <div class="panel-body">
          <div class="row">
            <div class="col-sm-6">
              <div id="chart1" style="float:left;width:100%;height:250px;"> </div>
            </div>
            <div class="col-sm-6">
              <div id="chart2" style="float:left;width:100%;height:250px;"> </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Um die Charts zu erzeugen, muss zunächst die Bibliothek geladen werden.

Diese holen wir direkt vom Hersteller aus dem CDN (Code Delivery Network)

```
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/highcharts-more.js"></script>
```

Dann folgt der Code für die eigentlichen Charts.

```
<script type="text/javascript" charset="utf-8">

// Create the chart
Highcharts.chart('chart1', {
  chart: {
    type: 'column',
    backgroundColor: 'transparent'
  },
  series: [{
    name: 'Total',
    colorByPoint: true,
    data: [{countif("1",that.Q3a.NAOK)},{countif("2",that.Q3a.NAOK)},{countif("3",that.Q3a.NAOK)},
    {countif("4",that.Q3a.NAOK)},{countif("5",that.Q3a.NAOK)},{countif("6",that.Q3a.NAOK)}]
  }]
});
```

Man sieht, dass hier die Funktion „countif“ benutzt wurde, um die Anzahl der Nennungen der Frage Q3a pro Code (1-6) zu erhalten.

Den Code der gesamten Charts ersehe man in der Beispiel-Datei.

Desweiteren sei auf die Dokumentation und die Beispiele hingewiesen.

<https://www.highcharts.com/docs/index>

<https://www.highcharts.com/demo>

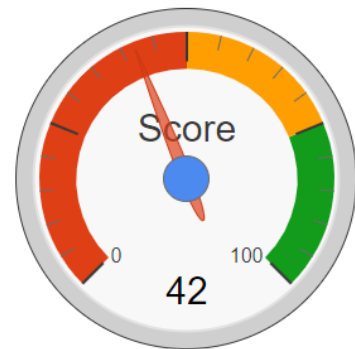
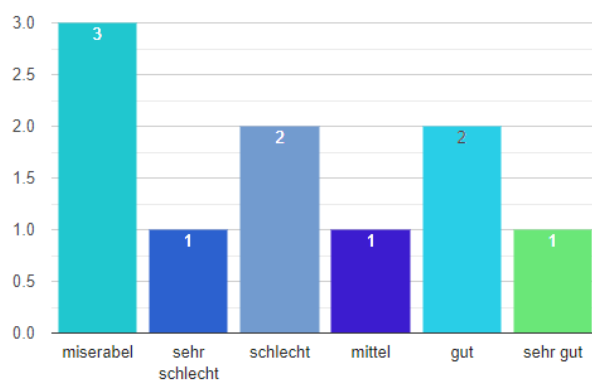
Diese Bibliothek ist nur für „Non-Profit“, persönliche und Schul-Zwecke kostenlos. Man muss aber dennoch eine Lizenz erwerben.

Are you using Highcharts for a non-profit organisation, personal website or school project?

[Get yourself a non-commercial license](#), governed by the Creative Commons (CC) Attribution-Noncommercial license, grab the code and start using Highcharts free of charge today.

3.1.2. Beispiel mit Google Charts

Anzahl Antworten pro Option und Gesamtscore (mithilfe von Google Chart)



Vom Aufbau her identisch; es werden wieder die bootstrap-Container erzeugt.

Das script zur Anzeige ist nun dies:

Zunächst wird die API geladen

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
```

```
<script type="text/javascript">

google.load('visualization', '1', { 'packages': ['corechart'] });
google.load('visualization', '1', { 'packages': ['gauge'] });

google.setOnLoadCallback(drawColumnChart);
google.charts.setOnLoadCallback(drawGaugeChart);

function drawColumnChart() {
  var data = google.visualization.arrayToDataTable([
    ["Element", "Anzahl", { role: "style" } ],
    ["miserabel", {countif("1",that.Q3a.NAOK)},'#1CC4CC'],
    ["sehr schlecht", {countif("2",that.Q3a.NAOK)},'#275BCC'],
    ["schlecht", {countif("3",that.Q3a.NAOK)}, '#6C96CC'],
    ["mittel", {countif("4",that.Q3a.NAOK)}, '#3618CC'],
    ["gut", {countif("5",that.Q3a.NAOK)}, '#24CBE5'],
    ["sehr gut", {countif("6",that.Q3a.NAOK)}, '#64E572'],
  ]);
}
```

```

var view = new google.visualization.DataView(data);
view.setColumns([0, 1,
                  { calc: "stringify",
                    sourceColumn: 1,
                    type: "string",
                    role: "annotation" },
                  2]);

var options = {
    legend: { position: 'none'},
    bar: { groupWidth: '85%' },
};

var chart = new google.visualization.ColumnChart(document.getElementById("chart1"));
chart.draw(view, options);
}

function drawGaugeChart() {
    var data = google.visualization.arrayToDataTable([
        ['Label', 'Value'],
        ['Score', {eqSum}],
    ]);

    var options = {
        width: 400, height: 300,
        redFrom: 0, redTo: 50,
        yellowFrom: 50, yellowTo: 75,
        greenFrom: 75, greenTo: 100,
        minorTicks: 5
    };

    var chart = new google.visualization.Gauge(document.getElementById('chart2'));
    chart.draw(data, options);
}
</script>

```

Desweiteren sei auf die Dokumentation hingewiesen; dort gibt es viele lauffähige Beispiele.

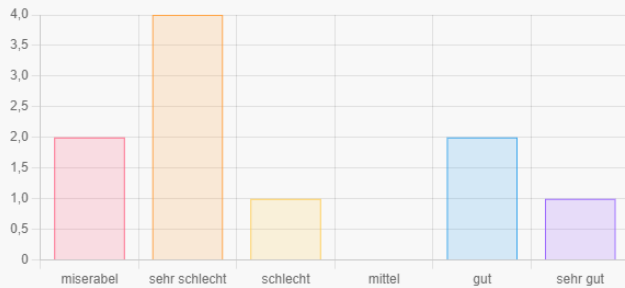
[Google Charts Dokumentation](#)

3.1.3. Beispiel mit Chartjs

Graphiken aufgrund von Ergebnissen der gegebenen Antworten (z.B. Scores)

Benutzte Bibliothek: Chartjs (diese enthält keine "Gauge"-Element)

Anzahl Antworten pro Option und Gesamtscore



Auch hier wird wieder der bootstrap-Container erzeugt.

Dann wird die Bibliothek geladen

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Und hier das script zur Erzeugung der Graphik-Bibliothek

```
<script>
const ctx = document.getElementById('chart1');

new Chart(ctx, {
  type: 'bar',
  data: {
    labels: ['miserabel', 'sehr schlecht', 'schlecht', 'mittel', 'gut', 'sehr gut'],
    datasets: [{
      //label: '# of Votes',
      data: [{countif("1",that.Q3a.NAOK)}, {countif("2",that.Q3a.NAOK)}, {countif("3",that.Q3a.NAOK)},
{countif("4",that.Q3a.NAOK)}, {countif("5",that.Q3a.NAOK)}, {countif("6",that.Q3a.NAOK)}],

      backgroundColor: [
        'rgba(255, 99, 132, 0.2)',
        'rgba(255, 159, 64, 0.2)',
        'rgba(255, 205, 86, 0.2)',
        'rgba(75, 192, 192, 0.2)',
        'rgba(54, 162, 235, 0.2)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(201, 203, 207, 0.2)'
      ]
    }]
  }
});
```

```
],  
  borderColor: [  
    'rgb(255, 99, 132)',  
    'rgb(255, 159, 64)',  
    'rgb(255, 205, 86)',  
    'rgb(75, 192, 192)',  
    'rgb(54, 162, 235)',  
    'rgb(153, 102, 255)',  
    'rgb(201, 203, 207)'  
  ],  
  borderWidth: 1  
}]  
},  
options: {  
  scales: {  
    y: {  
      beginAtZero: true  
    }  
  }  
}  
});  
</script>
```

Desweiteren sei auf die Dokumentation hingewiesen.

[Chartjs Dokumentation](#)

3.2. Ergebnisse der gesamten Umfrage

LimeSurvey verfügt bis zur Version 5.x. über keine Funktionen zum Zugriff auf andere Daten als die des aktuellen Teilnehmers.

Allerdings gibt es das Plugin „getStatInSurvey“ von Denis Chenu. Dazu muss der Teilnehmer also die Rechte haben, Plugins zu installieren.

<https://gitlab.com/SondagesPro/ExportAndStats/getStatInSurvey>

Hiermit kann man drei Werte einer Frage oder Teilfrage bestimmen:

- Anzahl
- Prozentwert
- Mittelwert

Seit Version 5.x ist eine solche Funktion eingebaut.

Hier gibt es die beiden Funktionen:

- `statCount(qCode, submitted = true)`
- `statCountIf(qCode, Antwort, submitted = true)`

Der Parameter „submitted“ ist mit „true“ vorbesetzt, muss in diesem Falle nicht mitangegeben werden. Nur wenn man nicht nur vollständige, sondern auch unvollständige Befragungen miteinbeziehen will, muss der Parameter „false“ hinzugefügt werden.

Da die Beispiele etwas umfangreicher sind, sind dafür separate lss-Dateien im Anhang.

3.2.1. Anwendungsbeispiel 1

Ein typisches Beispiel wäre, einem Teilnehmer die Verteilung und den Mittelwert einer (Skalen)-Frage aller bisherigen Teilnehmer anzuzeigen.

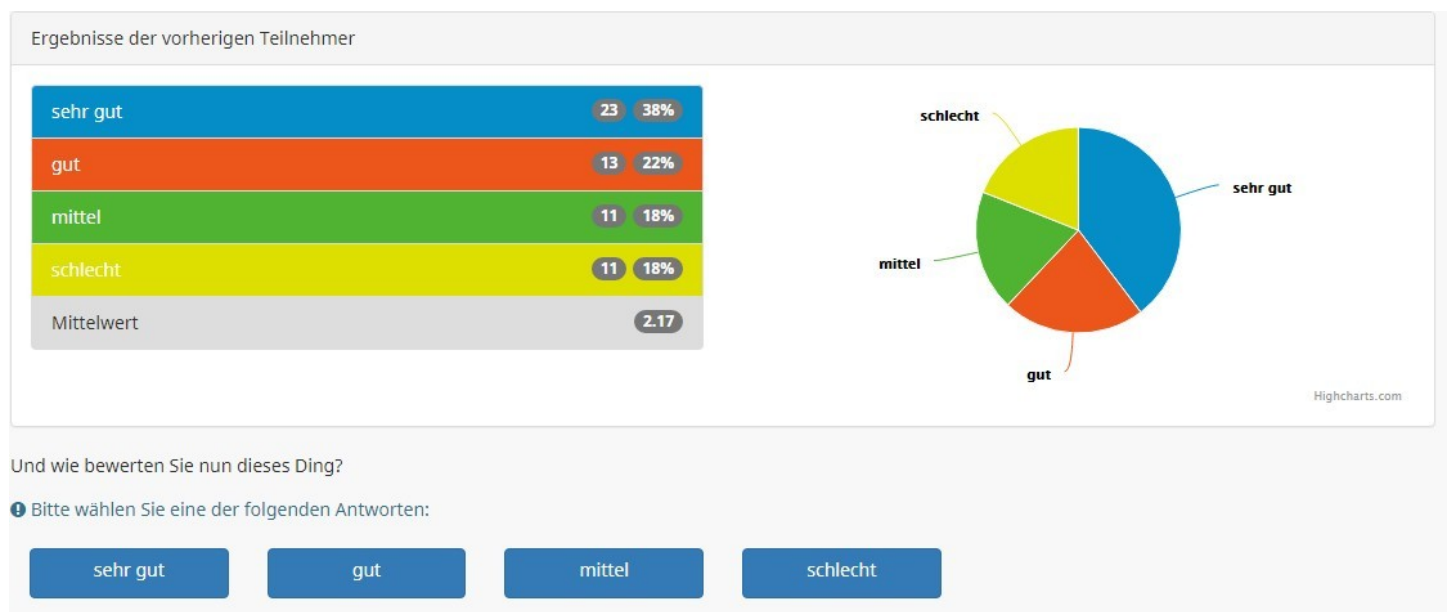
Nehmen wir an, die Frage Q1 hätte fünf Antwortoptionen

- 1 = sehr gut
- 2 = gut
- 3 = mittel
- 4 = schlecht
- 5 = sehr schlecht

Mit dem Plugin „getStatInSurvey“ würde dann die Verteilung so angezeigt

1: {'[Q1.nb.1]'}
2: {'[Q1.nb.2]'}
4: {'[Q1.nb.3]'}
4: {'[Q1.nb.4]'}
5: {'[Q1.nb.5]'}
und der Mittelwert : {'[Q1.moy2]'} (auf zwei Nachkommastellen gerundet)

Kann man dann wieder gut als einfache Tabelle oder auch als Graphik anzeigen (Bitte, keine Kommentare zu den Farben)



Implementiert wurde das Ganze als bootstrap-panel mit zwei gleichbreiten Spalten („col-sm-6“)

```
<div class="row">
  <div class="col-sm-12">
    <div class="panel panel-default">
      <div class="panel-heading">Ergebnisse der vorherigen Teilnehmer</div>
      <div class="panel-body">
        <div class="row">
          <div class="col-sm-6">
            <ul class="list-group">
              <li class="list-group-item" style="background-color:#058DC7;color:white"><span class="badge">{'
[Q1.pc.1]'}%</span><span class="badge">{' [Q1.nb.1]'}</span> sehr gut</li>
              <li class="list-group-item" style="background-color:#ED561B;color:white"><span class="badge">{'
[Q1.pc.2]'}%</span><span class="badge">{' [Q1.nb.2]'}</span> gut</li>
              <li class="list-group-item" style="background-color:#50B432;color:white"><span class="badge">{'
[Q1.pc.3]'}%</span><span class="badge">{' [Q1.nb.3]'}</span> mittel</li>
              <li class="list-group-item" style="background-color:#DDDF00;color:white"><span class="badge">{'
[Q1.pc.4]'}%</span><span class="badge">{' [Q1.nb.4]'}</span> schlecht</li>
              <li class="list-group-item" style="background-color:#DDDDDD;color:#333333"><span
class="badge">{' [Q1.moy2]'}</span> Mittelwert</li>
            </ul>
          </div>
          <div class="col-sm-6">
            ... Hier analog weiter, allerdings mit der Graphik. Wiederum eine HighChart-Graphik, wie schon
            einmal beschrieben.
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Mit dem ab Version 5.x. vorhandenen Plugin „statFunctions“:

1: {statCountIf(Q1.sgqa,'1')}

2: {statCountIf(Q1.sgqa,'2')}

3: {statCountIf(Q1.sgqa,'3')}

4: {statCountIf(Q1.sgqa,'4')}

5: {statCountIf(Q1.sgqa,'5')}

und der Mittelwert; da keine direkte Funktion hierfür vorhanden ist, wird er „mit der Hand“ gerechnet.

```
{sum(  
  statCountIf(Q1.sgqa,'1') *1,  
  statCountIf(Q1.sgqa,'2') *2,  
  statCountIf(Q1.sgqa,'3') *3,  
  statCountIf(Q1.sgqa,'4') *4,  
  statCountIf(Q1.sgqa,'5') *5)  
/statCount(Q1.sgqa)}
```

3.2.2. Anwendungsbeispiel 2

Ein weiteres beliebtes Beispiel ist die Vergabe von Plätzen bei Veranstaltungen. Wenn es im Seminarraum eben nur 10 Stühle gibt, oder die Exkursion nur eine Gruppengröße von 5 Teilnehmern hat.

Nehmen wir folgendes Startscenario:

An welchen Veranstaltungen wollen Sie teilnehmen?

- ☐ Ameisendressur (noch 10 Plätze frei)
- ☐ Biberdamm-Konstruktion (noch 5 Plätze frei)
- ☐ Chinchilla-Haarpflege (noch 10 Plätze frei)
- ☐ Dachsbau-Statik (noch 4 Plätze frei)

Nach der Auswahl dieser drei Veranstaltungen werden dem nächsten Teilnehmer nur noch die verbleibenden Plätze angezeigt

An welchen Veranstaltungen wollen Sie teilnehmen?

- ☒ Ameisendressur (noch 10 Plätze frei)
- ☒ Biberdamm-Konstruktion (noch 5 Plätze frei)
- ☐ Chinchilla-Haarpflege (noch 10 Plätze frei)
- ☒ Dachsbau-Statik (noch 4 Plätze frei)

An welchen Veranstaltungen wollen Sie teilnehmen?

- ☐ Ameisendressur (noch 9 Plätze frei)
- ☐ Biberdamm-Konstruktion (noch 4 Plätze frei)
- ☐ Chinchilla-Haarpflege (noch 10 Plätze frei)
- ☐ Dachsbau-Statik (noch 3 Plätze frei)

Und nach dem Klick auf den letzten verbleibenden Platz wird diese Veranstaltung nicht mehr angezeigt.

An welchen Veranstaltungen wollen Sie teilnehmen?

- ☐ Ameisendressur (noch 8 Plätze frei)
- ☐ Biberdamm-Konstruktion (noch 3 Plätze frei)
- ☒ Chinchilla-Haarpflege (noch 7 Plätze frei)
- ☒ Dachsbau-Statik (noch 1 Plätze frei)

An welchen Veranstaltungen wollen Sie teilnehmen?

- ☐ Ameisendressur (noch 8 Plätze frei)
- ☐ Biberdamm-Konstruktion (noch 3 Plätze frei)
- ☐ Chinchilla-Haarpflege (noch 6 Plätze frei)

Man kann diese nicht mehr wählbaren Veranstaltungen auch weiterhin anzeigen, jedoch ausgegraut. Dies hängt von der Einstellung in „Matrix Filter Typ“ ab. Dann sollte der Text besser etwas geändert werden.

An welchen Veranstaltungen wollen Sie teilnehmen?


- ☐ Ameisendressur (noch 8 Plätze frei)
- ☐ Biberdamm-Konstruktion (noch 3 Plätze frei)
- ☐ Chinchilla-Haarpflege (noch 6 Plätze frei)
- ☐ Dachsbau-Statik (noch 0 Plätze frei)

3.2.3. Implementierung

3.2.3.1. Plugin „statFunctions“, enthalten ab LimeSurvey-Version 5.x

Es wird die eingebaute Funktion „statCountIf“ benutzt.

Damit sehen die Teilfragen folgendermaßen aus:

Code	Subquestion 
SQ001	Ameisendressur (noch {10 - statCountIf(Q6_SQ001.sgqa,'Y')} Plätze frei)
SQ002	Biberdamm-Konstruktion (noch {5 - statCountIf(Q6_SQ002.sgqa,'Y')} Plätze frei)
SQ003	Chinchilla-Haarpflege (noch {10 - statCountIf(Q6_SQ003.sgqa,'Y')} Plätze frei)
SQ004	Dachsbau-Statik (noch {4 - statCountIf(Q6_SQ004.sgqa,'Y')} Plätze frei)

Die Gleichung `{10 - statCountIf(Q6_SQ001.sgqa,'Y')}` macht jetzt nichts anderes, als die Anzahl der bereits erfolgten Nennungen für die Teilfrage „SQ001“, die als „Y“ gespeichert sind, zu berechnen (Hinweis: In Mehrfachnennungen werden ausgewählte Items mit „Y“ gespeichert, nicht ausgewählte mit „“) und dann von 10 (der Gesamtzahl vorhandener Plätze) abzuziehen; damit haben wir die verbleibenden Plätze.

Die Relevanzgleichungen (um „volle“ Veranstaltungen auszublenzen) sehen dann entsprechend aus:

Relevance equation

`statCountIf(Q6_SQ001.sgqa,'Y') < 10`

`statCountIf(Q6_SQ002.sgqa,'Y') < 5`

`statCountIf(Q6_SQ003.sgqa,'Y') < 10`

`statCountIf(Q6_SQ004.sgqa,'Y') < 4`

Heißt also, die Teilfrage wird angezeigt, solange die Anzahl der getätigten Nennungen kleiner ist als die Gesamtzahl vorhandener Plätze.

3.2.3.2. Plugin „getStatInSurvey“

Dieses Plugin kann keine Mehrfachnennungen zählen, wie wir es oben taten.

Daher müsste man, um mit Mehrfachnennungen wie oben zu arbeiten, im Anschluss an die Frage die einzelnen Teilfragen mittels Gleichungen in „1/0“ Antworten mehrerer „kurze Text“-Fragen umwandeln. Das geht, soll aber hier nicht gezeigt werden; dafür ist eine spezielle Beispiel-Datei im Anhang.

Nehmen wir also eine Einfachnennung. Um hier ebenfalls „volle“ Veranstaltungen auszublenden, muss man zu dem üblichen Trick greifen, eine Mehrfachfrage entsprechend zu besetzen und diese dann als Matrix-Filter für die Einfachnennung zu benutzen. (Wie – hoffentlich – bekannt, oder schmerzlich erfahren, besitzen Einfachnennungen keine Relevanz auf die Antwortoptionen.

Wir starten also wie oben:

*An welcher Veranstaltung möchten Sie teilnehmen?

- ☐ Ameisendressur (Noch 10 Plätze frei)
- ☐ Biberdamm-Konstruktion (Noch 5 Plätze frei)
- ☐ Chichilla-Haarpflege (Noch 10 Plätze frei)
- ☐ Dachsbau-Statik (Noch 4 Plätze frei)

Die erste Auswahl ergibt dann dieses

*An welcher Veranstaltung möchten Sie teilnehmen?

- ☐ Ameisendressur (Noch 10 Plätze frei)
- ☒ Biberdamm-Konstruktion (Noch 5 Plätze frei)
- ☐ Chichilla-Haarpflege (Noch 10 Plätze frei)
- ☐ Dachsbau-Statik (Noch 4 Plätze frei)

*An welcher Veranstaltung möchten Sie teilnehmen?

- ☐ Ameisendressur (Noch 10 Plätze frei)
- ☒ Biberdamm-Konstruktion (Noch 4 Plätze frei)
- ☐ Chichilla-Haarpflege (Noch 10 Plätze frei)
- ☐ Dachsbau-Statik (Noch 4 Plätze frei)

Und dann etwas später

*An welcher Veranstaltung möchten Sie teilnehmen?

- ☐ Ameisendressur (Noch 9 Plätze frei)
- ☒ Biberdamm-Konstruktion (Noch 1 Plätze frei)
- ☐ Chichilla-Haarpflege (Noch 7 Plätze frei)
- ☐ Dachsbau-Statik (Noch 2 Plätze frei)

*An welcher Veranstaltung möchten Sie teilnehmen?

- ☐ Ameisendressur (Noch 9 Plätze frei)
- ☒ Chichilla-Haarpflege (Noch 7 Plätze frei)
- ☐ Dachsbau-Statik (Noch 2 Plätze frei)

(Ich weiß, dass das Vieh „Chinchilla“ heißt)

Hier ist die Prozedur etwas mühsamer.

Da ich Schwierigkeiten habe den Wert des Plugins direkt in die Gleichung zum Setzen der Mehrfachnennung zu integrieren, schiebe ich eine Frage vom Typ „mehrfache kurze Texte“ mit diesem javascript-Code ein.

```
<script>
$(document).on('ready ajax:scriptcomplete',function(){
    // Identify this question
    var thisQuestion = $('#question{QID}');
    // Fill the array
    $('input[type=text]:eq(0)', thisQuestion).val( {'[Q1.nb.1]'} );
    $('input[type=text]:eq(1)', thisQuestion).val( {'[Q1.nb.2]'} );
    $('input[type=text]:eq(2)', thisQuestion).val( {'[Q1.nb.3]'} );
    $('input[type=text]:eq(3)', thisQuestion).val( {'[Q1.nb.4]'} );
    $('#question{QID}').hide();
    $('#ls-button-submit').trigger('click');
});
</script>
```

Die letzte Zeile simuliert den Klick auf den „Weiter“-Button. Dies ist nicht nötig, sogar falsch, falls man diese Frage irgendwo in einer Gruppe am Anfang der Umfrage integriert.

Dann sind also die Anzahlen der einzelnen Codes der Frage „Q1“ in den Teilfragen dieser Frage. Hier sei sie „QV“ genannt.

Nun wird eine versteckte Mehrfachnennungsfrage erzeugt (HM für „Hidden Mehrfach“) mit so vielen Teilfragen, wie später in der Einfachnennung sind und auch mit denselben Codes.

Jetzt kommt eine Gleichung ins Spiel die die Teilfragen von HM setzt, falls noch Plätze frei sind.

```
{HM_1=if(QV_SQ001 lt 10,"Y","")}
{HM_2=if(QV_SQ002 lt 5,"Y","")}
{HM_3=if(QV_SQ003 lt 10,"Y","")}
{HM_4=if(QV_SQ004 lt 4,"Y","")}
```

Erste Zeile: Wenn die Anzahl in der ersten Teilfrage von QV kleiner als 10 ist, wird HM_1 auf „Y“, also ausgewählt gesetzt, sonst bleibt es leer. (lt = less than, Synonym für „<“)

Und nun wird diese Mehrfachnennungsfrage „HM“ als Matrix-Filter für unsere eigentliche Frage „Q1“ gesetzt.

Eine weitere Möglichkeit ist, die Antwortoption einer vollen Veranstaltung nur zu kennzeichnen, und Teilnehmer, die diese anklicken, auf eine Warteliste zuleiten

4. Lückentexte

Lückentexte sind ein nettes Gimmick für irgendwelche Tests. Daher hier als Beispiel eine kleine Spanisch-Übung.

4.1. Lückentext als Drop-Downs

Complete las frases usando los siguientes pronombres indeterminados:
algo, nada, alguien, nadie, alguna, algunas, ninguna.

Voy al mercado para comprar cosas para la cena.

Conoces canción de Shakira?

Tengo hambre. Todavía no he desayunado .

He oído un ruido raro, pero no he visto a .

Han dicho importante en las noticias?

Ha llamado para mí?

Antes tenía muchas faldas, pero ahora ya no tengo .

Complete las frases usando los siguientes pronombres indeterminados:
algo, nada, alguien, nadie, alguna, algunas, ninguna.

Voy al mercado para comprar cosas para la cena.

Conoces canción de Shakira?

Tengo hambre. Todavía no he desayunado .

He oído un ruido raro, pero no he visto a .

Han dicho importante en las noticias?

Ha llamado para mí?

Antes tenía muchas faldas, pero ahora ya no tengo .

Sie haben 7 richtige Antworten.

Wie man sieht, kann man hier gut ein Ergebnis berechnen, da die Antworten ja definiert sind. Anders ist es bei der nächsten Möglichkeit, dem Lückentext als offene Nennungen.

4.2. Lückentext mit „kurzen Texten“

Complete las frases usando los siguientes pronombres indeterminados:
algo, nada, alguien, nadie, alguna, algunas, ninguna.

Voy al mercado para comprar cosas para la cena.

Conoces canción de Shakira?

Tengo hambre. Todavía no he desayunado .

He oído un ruido raro, pero no he visto a .

Han dicho importante en las noticias?

Ha llamado para mí?

Antes tenía muchas faldas, pero ahora ya no tengo .

Die Implementation ist für beide Arten nahezu gleich. Der Fragetext enthält alle notwendigen „Lücken“ als `...`-Elemente.

Typ des Fragetextes ist also entweder „Klappbox“ oder „Kurzer Text“ und danach folgen noch Fragen desselben Typs für die weiteren „Lücken“.

Complete las frases usando los siguientes pronombres indeterminados:

<i>algo, nada, alguien, nadie, alguna, algunas, ninguna.</i>

Voy al mercado para comprar cosas para la cena.

Conoces canción de Shakira?

Tengo hambre. Todavía no he desayunado .

He oído un ruido raro, pero no he visto a .

Han dicho importante en las noticias?

Ha llamado para mí?

Antes tenía muchas faldas, pero ahora ya no tengo .

```

<script type="text/javascript" charset="utf-8">
    $(document).on('ready',function(){
        // Für Dropdowns

        var thisSelect = $('#question{QID} select[id^="answer"]:eq(0)');
        var secondSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(0)').find('select[id^="answer"]:eq(0)');
        var thirdSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(1)').find('select[id^="answer"]:eq(0)');
        var fourthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(2)').find('select[id^="answer"]:eq(0)');
        var fifthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(3)').find('select[id^="answer"]:eq(0)');
        var sixthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(4)').find('select[id^="answer"]:eq(0)');
        var seventhSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(5)').find('select[id^="answer"]:eq(0)');

        // Für „kurze Texte“

        var thisSelect = $('#question{QID} input[id^="answer"]:eq(0)');
        var secondSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(0)').find('input[id^="answer"]:eq(0)');
        var thirdSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(1)').find('input[id^="answer"]:eq(0)');
        var fourthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(2)').find('input[id^="answer"]:eq(0)');
        var fifthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(3)').find('input[id^="answer"]:eq(0)');
        var sixthSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(4)').find('input[id^="answer"]:eq(0)');
        var seventhSelect = $
        ($('#question{QID}').nextAll('[id^="question"]:eq(5)').find('input[id^="answer"]:eq(0)');

        // Move the elements

        $('#question{QID} .inserted-element:eq(0)').append(thisSelect);
        $('#question{QID} .inserted-element:eq(1)').append(secondSelect );
        $('#question{QID} .inserted-element:eq(2)').append(thirdSelect );
        $('#question{QID} .inserted-element:eq(3)').append(fourthSelect );
        $('#question{QID} .inserted-element:eq(4)').append(fifthSelect );
        $('#question{QID} .inserted-element:eq(5)').append(sixthSelect );
        $('#question{QID} .inserted-element:eq(6)').append(seventhSelect );

        // Hide the next questions
        $('#question{QID}').nextAll('[id^="question"]:eq(0)').hide();
        $('#question{QID}').nextAll('[id^="question"]:eq(1)').hide();
        $('#question{QID}').nextAll('[id^="question"]:eq(2)').hide();
        $('#question{QID}').nextAll('[id^="question"]:eq(3)').hide();
        $('#question{QID}').nextAll('[id^="question"]:eq(4)').hide();
        $('#question{QID}').nextAll('[id^="question"]:eq(5)').hide();
        // $('#question{QID}').nextAll('[id^="question"]:eq(6)').hide();
    });
</script>

```

Hier das css. In beiden muss der Wert „height“ je nach Theme angepasst werden, um ein einheitliches Aussehen zu erreichen. Desweiteren sollte in der „kurze Text“-Lösung „width“ angepasst werden an die erwartete Länge der Einträge.

```
<style type="text/css">
  #question{QID} input {
    display: inline-block;
    margin: 0 0 0 -0.25em;
    width: 8em;
    height: 2em;
  }
</style>
```

```
<style type="text/css">
  #question{QID} select {
    display: inline-block;
    margin: 0 0 0 -0.25em;
    width: auto;
    height: 2em;
    padding: 0;
  }
</style>
```

5. Eingabemasken

Eingabemasken können helfen, zumindest die korrekte Struktur einer Eingabe von vornherein sicherzustellen. Zumindest dies kann erzwungen werden. An den entsprechenden Stellen ist nur ein Zeichen des richtigen Typs möglich.

Hat man allerdings weitere Informationen, sollte dann auch auf richtige Logik geprüft werden, wie „eine niederländische Postleitzahl beginnt nicht mit „0“, was auch immer

Selbstverständlich ist dies ebenso mittels „Regulärer Ausdrücke“ zu bewerkstelligen.

Der Vorteil ist die visuelle Hilfe durch die Unterstriche bzw. Platzhalter sowie die Unmöglichkeit „typfremde“ Zeichen einzugeben; der „reguläre Ausdruck“ lässt dies zu, meckert dann aber einen Fehler an.

5.1. Mit der Bibliothek „Inputmask“ von Robin Herbots

<https://github.com/RobinHerbots/Inputmask>

5.1.1. Anwendungsbeispiel

Postcode Niederlande Format: 1234 AB	<input type="text"/>
Zipcode Kanada Format: A1B 2C3	<input type="text"/>
Telefonnummer USA Format: (123) 456-7890	<input type="text"/>

5.1.2. Implementierung

Zunächst wird die Bibliothek direkt vom CDN geladen mit

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/inputmask/4.0.9/jquery.inputmask.bundle.min.js"></script>
```

Danach werden nur noch die Masken definiert.

Entweder in Fragen vom Typ „mehrfache kurze Texte“

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready ajax:scriptcomplete',function(){
    $('#question{QID} .question-item:eq(0) input[type="text"]').inputmask({
      'mask': '9999 AA',
      'placeholder': '_',
      'removeMaskOnSubmit': false,
      'rightAlign': false,
    });
    $('#question{QID} .question-item:eq(1) input[type="text"]').inputmask({
      'mask': 'A9A 9A9',
      'placeholder': '_',
      'removeMaskOnSubmit': false,
      'rightAlign': false,
    });
    $('#question{QID} .question-item:eq(2) input[type="text"]').inputmask({
      'mask': 'AAAAA99A99A999A',
      'placeholder': '_',
      'removeMaskOnSubmit': false,
      'rightAlign': false,
    });
    $('#question{QID} .question-item:eq(3) input[type="text"]').inputmask({
      'mask': '(999) 999-9999',
      'placeholder': '_',
      'removeMaskOnSubmit': false,
      'rightAlign': false,
    });
  });
}</script>
```

Wie schon früher erwähnt, wird die Zeile (Teilfrage) mittels **:eq(x)** bestimmt, wobei die erste Teilfrage den Index 0 hat.

Oder auch in einer Matrix(Texte)

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready ajax:scriptcomplete',function(){
    $('#question{QID} .answer_cell_X002 input[type="text"]').inputmask({
      'mask': '9999 AA',
      'placeholder': '_',
      'removeMaskOnSubmit': false,
      'rightAlign': false,
    });
  });
}</script>
```

Hier greifen wir direkt auf die Spalte zu mittels „answer_cell_X002“.

Hier hat also die zweite Spalte der Matrix den Code „X002“.

5.2. Mit der Bibliothek „jquery mask“ von Igor Escobar

<https://igorescobar.github.io/jQuery-Mask-Plugin/>

Auf dieser Bibliothek beruht auch das Plugin „jquerySimpleMask“ von DenisChenu, welches eine etwas „abgespeckte“ Version für „mehrfache kurze Texte“ bietet.

<https://gitlab.com/SondagesPro/QuestionSettingsType/jquerySimpleMask>

5.2.1. Anwendungsbeispiel

Dies ist nahezu analog.

Postcode Niederlande	<input type="text" value="1234 AB"/>
Zipcode Kanada	<input type="text" value="A1B 2C3"/>
Telefonnummer USA	<input type="text" value="(123) 456-7890"/>
Zahleingabe, zwei Kommastellen	<input type="text"/>

5.2.2. Implementierung

Zunächst binden wir die Bibliothek vom CDN ein

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.16/jquery.mask.js"></script>
```

Danach werden die Masken definiert.

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready ajax:scriptcomplete',function(){
    $('#question{QID} .question-item:eq(0) input[type="text"]').mask('0000 ZZ', {
      translation: {
        'Z': { pattern: /[A-Z]/ } // Z wird als Großbuchstabe definiert
      },
      placeholder: "1234 AB"
    });
    $('#question{QID} .question-item:eq(1) input[type="text"]').mask('Z0Z 0Z0', {
      translation: {
        'Z': { pattern: /[A-Z]/ } // Z wird als Großbuchstabe definiert
      },
      placeholder: "A1B 2C3",
    });
    $('#question{QID} .question-item:eq(2) input[type="text"]').mask('(000) 000-0000', { placeholder:
"(123) 456-7890" });
    $('#question{QID} .question-item:eq(3) input[type="text"]').mask('#.##0,00', { reverse: true });
  });
</script>
```

Hier zusätzlich gezeigt, wie ein neuer Typ definiert wird, mittels „translation“.

6. Größere Darstellung von Bildern

6.1. Anwendungsbeispiel

Ein Anwendungsfall wäre, dass man ein Bild bereits einmal bewertet hat, wobei es groß dargestellt wurde. Dies kann zum Beispiel in dem später beschriebenen „Points of Interest“ geschehen sein, oder sonstwie.

Später im Fragebogen werden dann weitere Fragen zum Bild gestellt. Hier muss es nicht noch einmal in voller Größe dargestellt werden, den Teilnehmern aber doch für den Fall der Fälle als Vollbild zur Verfügung stehen.

Inwieweit können Sie folgenden Aussagen zustimmen?

Bitte wählen Sie pro Zeile eine Antwort.

Klick auf das Bild vergrößert es.



	stimme voll und ganz zu				stimme überhaupt nicht zu	weiß nicht/ keine Angabe
	5	4	3	2	1	
Die Seite ist sehr übersichtlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Alle notwendigen Informationen sind sofort zugänglich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das Arbeiten macht Spaß	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mir gefällt die dezente Farbgebung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Powered by: Highslide JS

Home / Chats list

Inwieweit können Sie folgenden Aussagen zustimmen?

Bitte wählen Sie pro Zeile eine Antwort.

Klick auf das Bild vergrößert es.

weiß nicht/ keine Angabe

Die Seite ist sehr übersichtlich

Alle notwendigen Informationen sind sofort zugänglich

Das Arbeiten macht Spaß

Mir gefällt die dezente Farbgebung

Close selected Delete selected

Search Print Export (4 chats)

Information

Operator	Department	Status
Visitor, 2021-08-29 21:33:09, Support	Support	Closed chat
Visitor, 2021-03-11 16:28:19, Support	Support	Closed chat
Visitor, 2021-03-11 13:41:51, Support	Support	Closed chat
Visitor, 2021-03-11 15:11:23, Support	Support	Closed chat

6.2. Implementierung

Hierzu wird die Bibliothek „Highslide“ benutzt. <http://highslide.com/>

Bisher haben wir solche Bibliotheken direkt von einem CDN (Content Delivery Network) geladen. Dies ist in diesem Fall nicht möglich (gibt es nicht).

Daher muss man die Dateien von der Homepage des Herstellers herunterladen und irgendwo auf dem eigenen Server ablegen, z.B. in das „files“-Verzeichnis der Umfrage.

Bei mir sehen die ersten Zeilen des Fragetextes so aus:

```
<script type="text/javascript" src="https://www.mafoserver.de/userfiles/highslide/highslide-full.js"></script>
<link href="https://www.mafoserver.de/userfiles/highslide/highslide.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">
  // override Highslide settings here
  // instead of editing the highslide.js file
  hs.graphicsDir = 'https://www.mafoserver.de/userfiles/highslide/graphics/';
  hs.wrapperClassName = 'wide-border';
  hs.align = 'center';
</script>
```

D.h. ich habe die beiden Dateien „highslide-full.js“ und „highslide.css“ in einen Ordner auf meinen Server gelegt und auch das gesamte Verzeichnis „graphics“, welches noch ein Unterverzeichnis enthält.

Damit ist die Bibliothek eingebunden, es geht weiter

```
<a class="highslide pull-right" href="https://www.mafoserver.de/userfiles/chat_mid.jpg" onclick="return
hs.expand(this)" rel="highslide">
  
</a>
```

Hier ist zweimal dasselbe Bild „chat_mid.jpg“ eingefügt; die Größe des zweiten, kleinen ist durch den Parameter „width=‘180px‘“ bestimmt.


Man kann auch zwei verschiedene Bilder benutzen, die vorher auf eine angemessene Größe gebracht wurden.

Achtung bei kleinen Devices (smartphones): Der Effekt ist hier logischerweise nicht besonders ausgeprägt. Breiter als der Bildschirm wird auch das große Bild nicht.

7. Lupenfunktion

Zunächst präsentiert sich die Frage recht normal.

Wie gefällt das Bild denn so?





Bitte wählen Sie eine der folgenden Antworten:

☐ sehr gut ☐ gut ☐ mittel ☐ schlecht ☐ sehr schlecht

Wenn man aber mit der Maus über das Bild fährt, öffnet sich eine zweite vergrößerte Ansicht dieses Bereiches.

Wie gefällt das Bild denn so?



Bitte wählen Sie eine der folgenden Antworten:

☐ sehr gut ☐ gut ☐ mittel ☐ schlecht ☐ sehr schlecht

Für dieses Feature benutzen wir die Bibliothek „drift“.

Diese wird wieder vom CDN geladen.

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/drift-zoom/1.4.0/drift-basic.css" media="screen, projection" rel="stylesheet" />
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/drift-zoom/1.4.0/Drift.js"></script>
```

Es wird zweimal dasselbe Bild eingebunden mit verschiedenen Größen (w=1200 und w=200)

```
<div class="wrapper">
```

Für Weiteres lese man auf der Github-Seite von „drift“ .

<https://github.com/imgix/drift>

Mir erschließt sich auch – noch – nicht alles.

8. Bewertung bzw. Auswahl mittels Pins in Bildern



8.1. Bewertung mittels Pins (Points of Interest) (positiv - negativ)

In diesem Kapitel wird eine „Fragenvorlage“ benutzt.

Diese hier ist in den neueren Version des 5.x. Zweiges im Core enthalten. Sonst muss sie installiert werden.

Points of interest

Home / Chats list

Nick: E-mail: Department: Department group: User: User group:

















Date range from: Hour and minute from [23:03:55]: Date range to: Hour and minute to [23:03:55]: Wait time:

Vote status: Chat status: Product: Chat ID: Chat duration:

Subject: Proactive invitation: ☐ Has unread messages ☐ Unanswered chat ☐ Anonymised

Select bot: ☐ Chats without an operator ☐ Chats with an operator ☐ Chats which had a bot ☐ Chats which did not had a bot



Search Print Export (4 chats)

Information	Operator	Department	Status
<input type="checkbox"/>    Visitor, 2021-08-29 21:33:09, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 16:23:19, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 15:41:51, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 15:17:29, Support	Jo...	Support	 Closed chat

Close selected Delete selected

Points of interest

Home / Chats list

Nick: E-mail: Department: Department group: User: User group:

















Date range from: Hour and minute from [23:03:55]: Date range to: Hour and minute to [23:03:55]: Wait time:

Vote status: Chat status: Product: Chat ID: Chat duration:

Subject: Proactive invitation: ☐ Has unread messages ☐ Unanswered chat ☐ Anonymised

Select bot: ☐ Chats without an operator ☐ Chats with an operator ☐ Chats which had a bot ☐ Chats which did not had a bot

Search Print Export (4 chats)

Information	Operator	Department	Status
<input type="checkbox"/>    Visitor, 2021-08-29 21:33:09, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 16:23:19, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 15:41:51, Support	Jo...	Support	 Closed chat
<input type="checkbox"/>    Visitor, 2021-03-11 15:17:29, Support	Jo...	Support	 Closed chat

Close selected Delete selected

Wie man sieht, gibt es positive und negative Marker, die man auf entsprechende Stellen des Bildes ziehen und auch noch mit einem Kommentar versehen kann.

Dies alles beruht auf einer Frage vom Typ „Mehrfache kurze Texte“.

Pro Marker werden drei Teilfragen benötigt für

- das Vorzeichen (+ für den grünen, - für den roten Marker)
- die Koordinaten (links oben ist (0,0), rechts unten (100,100))
- den Kommentar

Zum Beispiel habe ich die Codes einmal so vergeben:

Code	Teilfrage
SQ011	1-sign
SQ012	1-coords
SQ013	1-comment
SQ021	2-sign
SQ022	2-coords
SQ023	2-comment

Der Code drückt schon aus, zu welchem Marker er gehört und welche Daten dort gespeichert werden.

Dies erleichtert die Auswertung später ungemein.

Man kann zum Beispiel in einem Scatter-Diagramm die Punkte wieder auf das Bild legen und sieht dann Häufungen an bestimmten Stellen.

Die (Anzahl der Teilfragen / 3) bestimmt dann die zur Verfügung stehende Anzahl Marker.

Nachdem die Fragenvorlage installiert und ausgewählt wurde, genügt es, den Pfad zum Bild anzugeben.

Custom options

Image path ⓘ

https://www.mafoserver.de/userfiles/chat_full.jpg

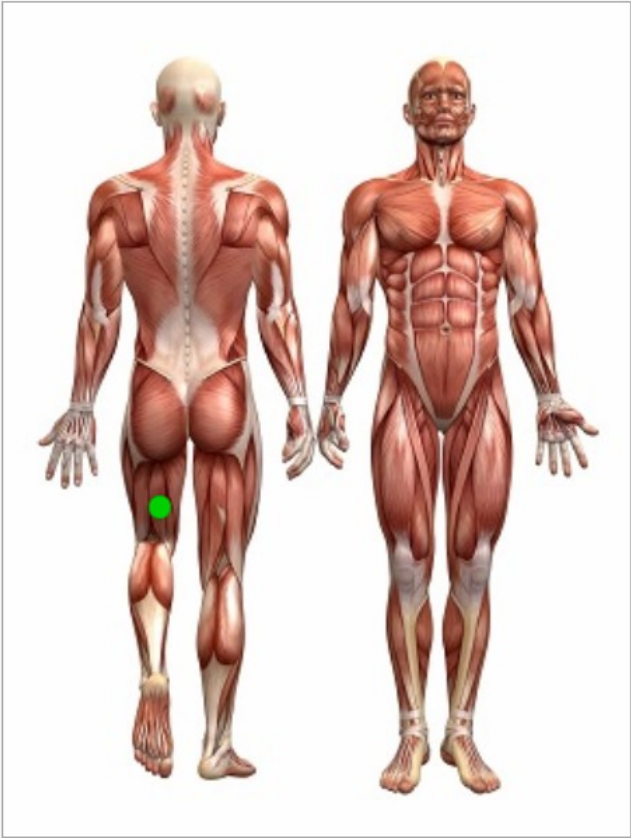
8.2. Setzen einer Markierung in ein Bild

Im Gegensatz zum vorherigen Beispiel wird hier nur ein einziger Pin gesetzt, und es ist auch keine Bemerkung möglich.

Daher kann man dies gut benutzen um auf einem Bild etwas suchen oder bestimmen zu lassen.

Beispiel:

Bitte, klicken Sie auf das Bild, wo Sie den **Musculus levator scapulae** vermuten!



25.20,60.66

Da ist er zwar nicht, aber auf jeden Fall werden die Koordinaten des Klicks gespeichert.

In der „scharfen“ Studie wird man dieses Eingabefeld natürlich ausblenden.

Dies wird im Code, den man in der Lss anschauen möge, mittels dieser Zeilen gesteuert.

```
// Hide the text input - un-comment below
// $('input.text', thisQuestion).hide();
// $('.ls-answers', thisQuestion).hide();
```

Bei der Auswertung könnte man zum Beispiel wieder dieses Bild als Hintergrund nehmen und mittels eines Scatter-Plots die von den Teilnehmern behaupteten Punkte anzeigen, oder man legt einen gewissen Koordinatenbereich fest, in welchem die Antwort als richtig gewertet wird.

Dann könnte man sogar eine direkte Rückmeldung geben.

9. Cardsort

Auch hier dreht sich alles um „Gamification“. Es sieht eben etwas „knackiger“ aus, als die Logos einfach in einer Matrix zu bewerten.

Diese Fragenvorlagen (für 3.x. und 5.x.) stammen von Tony Partner (tpartner). Ebenfalls in den neueren Versionen des 5.x. Zweiges enthalten.

<https://github.com/tpartner/LimeSurvey-Card-Sorting-3.x>

<https://github.com/tpartner/LimeSurvey-Card-Sorting-5.x>

Cardsort

sehr gut

gut

mittelmäßig

schlecht

Hier wurde in der „cardsort.css“ ein bisschen herumgespielt.

Das Original sieht so aus.

Family

Your Country

Religion

Happiness

Career

Very Important

Education

Fairly Important

Wealth

Not very important

Health

Wieder wird eine Frage vom Typ „Mehrfache kurze Texte“ benutzt. Die Logos bzw. Adjektive sind die Teilfragen. Die Beschriftung der „Kästen“ wird hier vorgenommen:

Das ist dann schon alles. Gespeichert wird in den Teilfragen der Text des Kästchens.

Custom options

Sorting area names ⓘ

sehr gut, gut, mittelmäßig, schlecht

10. IAT (Implicit Association Test)

10.1. Anwendungsbeispiel

Dazu ein bisschen aus Wikipedia über diesen Test.

https://de.wikipedia.org/wiki/Impliziter_Assoziationstest



Damit haben wir die Anwendung bereits abgehakt.

10.2. Implementierung

Diese Frage beruht auf einer Frage vom Typ Matrix(Texte).

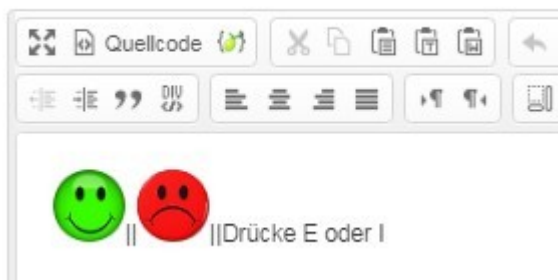
Die Teilfragen der y-Achse werden von den zu bewertenden Objekten gebildet.

In der x-Achse gibt es zwei Felder: die gedrückte Taste und die benötigte Zeit.

Code	Teilfrage
key	Key Pressed
elapsed	Elapsed Time (ms)

Die Betextung (hier also die beiden Smileys sowie „Drücke E oder I“) wird im Hilfetext der Frage platziert.

Hilfetext:



Die drei Elemente werden durch doppelte | getrennt.

Und dann wird wie üblich der javascript- und css- Code in den Fragetext (im Quellcode-Modus) eingefügt.

Dieser ist sehr lang.

Bei Interesse sehe man es in der beigefügten Beispielstudie an.

Diese Fragevorlage gibt es für die Version 3.x. von Tony Partner.

<https://github.com/tpartner/LimeSurvey-Implicit-Association-Test-3x>

11. So etwas wie Conjoint

Wichtig: Die Fragen der Gruppen G11a, G11b, G11c sind abhängig von einer Vorbesetzung, welche in Gruppe G10 stattfindet. Daher können diese drei Gruppen nicht sinnvoll in der „Fragengruppenvorschau“ angezeigt werden.

11.1. Darstellung mittels (statischer) HTML-Tabellen im Fragetext

11.1.1. Layout-Möglichkeiten

Zunächst ein paar Bildchen dazu:

Der Urvater – das Apple-Banana-Beispiel

	BANANAS	GREEN APPLES
Type of Vita-mins	XX	XXXX
Hidratos	SUGAR Bla blabla bla bla bla bla bla bla bla bla bla blabla bla bla bla bla bla bla bla bla bla blabla bla bla bla bla bla bla bla bla bla blabla bla bla bla bla bla bla bla bla bla blabla .	ACID JUICE Bla blabla bla bla bla bla bla bla bla bla bla blabla bla bla bla bla bla bla bla bla bla blabla bla bla bla bla bla bla bla bla bla bla bla bla .
Price	1 Kg - 1€	1Kg - 0.95 €
	<input type="radio"/>	<input type="radio"/>

Oder so:

*		
Which car do you prefer?	Car 1	Car 2
Type	Station wagon	Sedan
kW	90	140
Max.Speed [km/h]	180	195
Price	21000	19000
Ihre Wahl	<input type="radio"/>	<input type="radio"/>

Hier ist ein Beispiel, welches aus 100 verschiedenen Routen eine beliebige Anzahl zufälliger Paare anzeigt.

*Welche der beiden Routen würden Sie bevorzugen?	Route 1	Route 2
Gesamtstrecke	37,5 km (31min)	21,7 km (23min)
davon auf:		
Fahrbahn	5%	22%
Radfahrstreifen	15%	7%
Radweg	7%	51%
Gemeinsamer Geh- und Radweg	70%	11%
Gehweg	3%	9%
Verkehrsaufkommen:		
keine Autos	25%	20%
geringes Verkehrsaufkommen	25%	30%
mittleres Verkehrsaufkommen	25%	15%
starkes Verkehrsaufkommen	25%	35%
Oberfläche:		
Schotter	15%	4%
Kopfsteinpflaster	22%	43%
Beton	11%	22%
Asphalt	52%	31%
Grünflächenanteil	hoch	mittel
Kreuzungen	58	70
Ihre Wahl:	<input type="radio"/>	<input type="radio"/>

11.1.2. Ideen zur Darstellung der Optionen

Natürlich ist dies keine richtige Conjoint-Analyse. Aber es gibt doch verschiedenen Möglichkeiten, eine Analyse-light nachzubilden.

11.1.2.1. Fest verdrahtete Optionen

Hier kann man so viele Fragen erstellen, wie man Kombinationen darstellen will (möglicherweise als orthogonales Modell berechnet).

Dann werden die Texte fest eingefügt.

Zusätzlich gibt es natürlich die Möglichkeit, mittels einer Zufallszahl nur eine Teilgruppe jeweils anzuzeigen, diese auch in unterschiedlicher Reihenfolge.

Das sind aber Basics, die im vierten Tutorial noch einmal ausführlich gezeigt werden.

11.1.2.2. Zufällige Anzeige von n Kombinationen aus einem Pool von m

Hier ist die Idee, dass man zunächst alle gewünschten Kombinationen wohl definiert als Vorgabe-Antwort in eine Frage vom Typ „langer Text“ (H1) einfügt.

Das sieht dann so aus:

Sedan	120	195	18000	SUV	90	180	21000	Coupe	90	190	...
-------	-----	-----	-------	-----	----	-----	-------	-------	----	-----	-----

Wohl definiert heißt, dass alle Texte eine ganz bestimmte Länge haben, hier 10 Zeichen.

Wenn man dann in H0 einen String von $2 \cdot n$ Zufallszahlen, jede drei Zeichen lang, bestimmt wie

```
009012020004015013006014008017019003005016002010001007011018
```

kann man mit einfachen Funktionen definiert auf jeden einzelnen Wert zugreifen.

Auf den 3. Wert des Fahrzeuges, welches an 5. Stelle des Textes steht, also „015“

```
{trim(substr(substr(H1, (substr(H0, 12, 3) - 1) * 40, 40), 20, 10))}
```

Das ist alles sehr schön, aber wohl unmöglich auszuwerten.

11.1.2.3. Vorher berechnete Kombinationen

Man kann zum Beispiel in SPSS ein Modell rechnen lassen, welches als Ergebnis eine Anzahl „Karten“ ergibt.

	Typ	PS	Preis	Speed
1	Sedan	75	25000	180
2	Coupe	120	15000	210
3	Hatch	75	15000	140
4	SUV	150	15000	140
5	Hatch	120	15000	180
6	SUV	120	25000	140
7	Sedan	150	15000	140
8	Hatch	90	35000	140
9	Coupe	75	15000	140
10	Sedan	120	35000	140
11	SUV	90	15000	180
12	Coupe	90	25000	140
13	Coupe	150	35000	180
14	Sedan	90	15000	210
15	Hatch	150	25000	210
16	SUV	75	35000	210

Diese kann man nun per Zufallszahl auf die Teilnehmer verteilen, oder zufällig als zusätzliches Attribut jedem Teilnehmer zuordnen.

Dies ist natürlich recht ähnlich zu 11.2.1.

11.1.3. Implementierung

In den Fragetext wird eine HTML-Tabelle „gebastelt“.

Dort können dann auch Platzhalter eingefügt werden, um auf Texte von anderen Quellen zuzugreifen.

```
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse:collapse;" width="100%">
  <tbody>
    <tr>
      <td style="background-color:#efefef;text-align: left;vertical-align:middle;width:50%;font-size:18pt;">Which car do you prefer?</td>
      <td style="background-color:#308737;text-align: center;vertical-align:middle;width:25%;font-size:18pt;font-family:Calibri,sans-serif;color:#ffffff;"><strong>Car 1</strong></td>
      <td style="background-color:#308737;text-align: center;width:0%;border-right:0;border-left:0;"> </td>
      <td bgcolor="#308737" style="text-align: center;vertical-align:middle;width:25%;font-size:18pt;font-family:Calibri,sans-serif;color:#ffffff;"><strong>Car 2</strong></td>
    </tr>
    <tr>
      <td style="background-color:#c1c1c1;text-align: left;vertical-align:middle;font-size:14pt;">Type</td>
      <td style="background-color:#f2f2f2;text-align: center;vertical-align:middle;font-size:14pt;"><strong>{trim(substr(substr(H1, (substr(H0, 12, 3) - 1) * 80, 80),0,20))}</strong></td>
      <td style="background-color:#f2f2f2;text-align: center;" width="0%"> </td>
      <td style="background-color:#f2f2f2;text-align: center;vertical-align:middle;font-size:14pt;"><strong>{trim(substr(substr(H1, (substr(H0, 15, 3) - 1) * 80, 80),0,20))}</strong></td>
    </tr>
  </tbody>
</table>
```

Hier ist die in jedem „<td>“ enthaltene style-Orgie nur der Deutlichkeit halber. Hier ist es besser css-Klassen zu verwenden.

Dieses kleine javascript zieht nun die Radio-Buttons aus dem Antwort-Teil ans Ende dieser Tabelle.

```
<script type="text/javascript" charset="utf-8">

$(document).on('ready ajax:scriptcomplete',function(){

  // Identify this question
  var thisQuestion = $('#question{QID}');

  // Move the radios
  $('.question-text table:eq(0) tr:last td:eq(1)', thisQuestion).append($('.subquestion-list .answers-list:eq(0) .answer-item:eq(0) *', thisQuestion));
  $('.question-text table:eq(0) tr:last td:eq(3)', thisQuestion).append($('.subquestion-list .answers-list:eq(0) .answer-item:eq(1) *', thisQuestion));

  // Some classes for presentation
  $('.question-text table:eq(0) input:radio', thisQuestion).closest('td').addClass('answer-item radio-item text-center radio');
  $('.question-text table:eq(0) .radio-item label', thisQuestion).show();

  // Click event on the table cells
```

```

$('.question-text table:eq(0) .radio-item', thisQuestion).on('click', function(e) {
    $('input:radio', this).trigger('click');
});
$('.question-text table:eq(0) input:radio', thisQuestion).on('click', function(e) {
    e.stopPropagation();
});

// Clean-up styles
$('.answer-container', thisQuestion).hide();
$('.question-text table:eq(0) .label-text', thisQuestion).remove();
$('.question-text table:eq(0) .radio-text', thisQuestion).css({
    'cursor': 'pointer'
});
});
});
</script>

```

Um dies auch für kleinere Bildschirme verfügbar zu machen, gibt es die Möglichkeit, den gesamten HTML-Teil, also `<table>... </table>` mit

`<div class="table-responsive">`

...

`</div>`

zu umschließen.

Dann wird die Tabelle horizontal gescrollt.

11.2. Darstellung mittels Slider-Bibliotheken

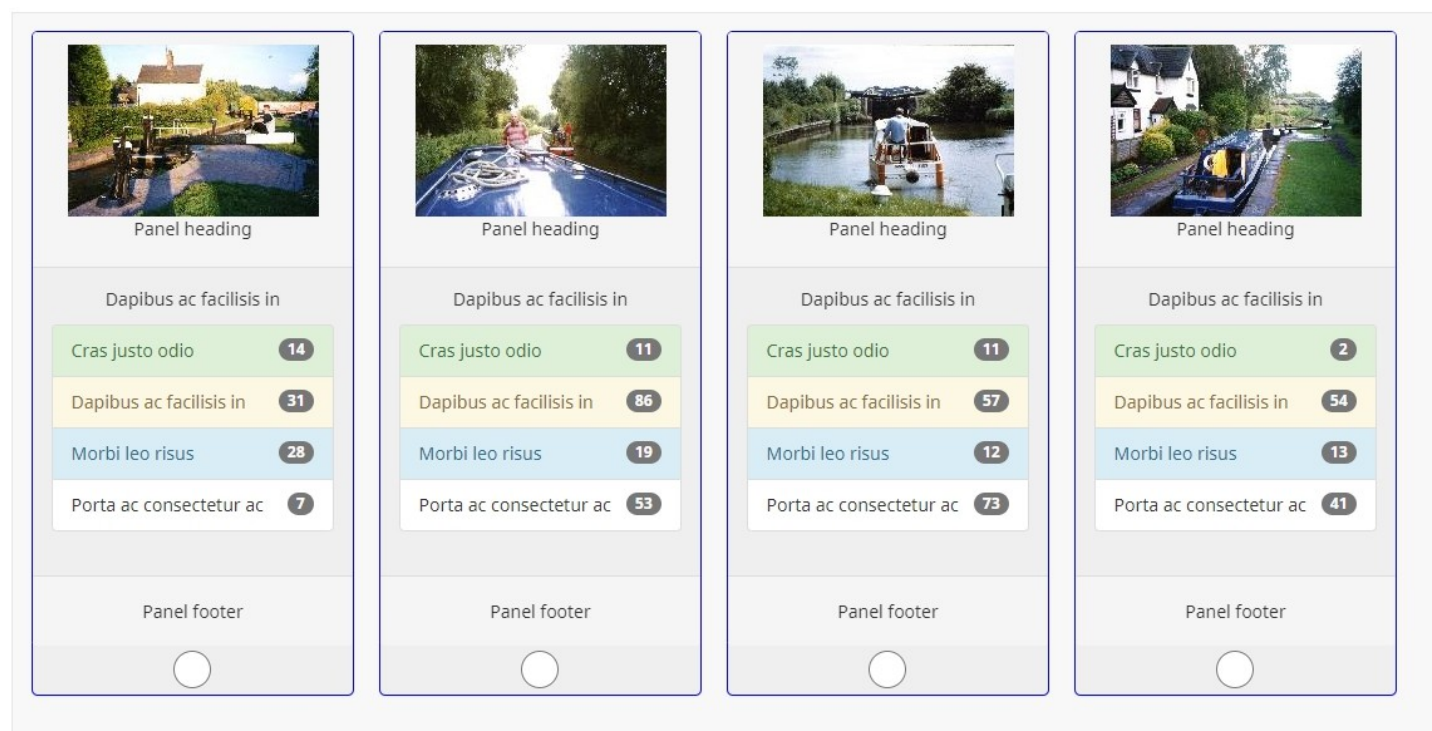
Ich möchte hier zwei Bibliotheken zeigen, mit deren Hilfe man sehr schön Karussells erstellen kann. Hier dient dies dazu, ebenfalls auf kleinen Bildschirmen eine ansprechende Möglichkeit der Beantwortung zu schaffen.

11.2.1. Tiny Slider

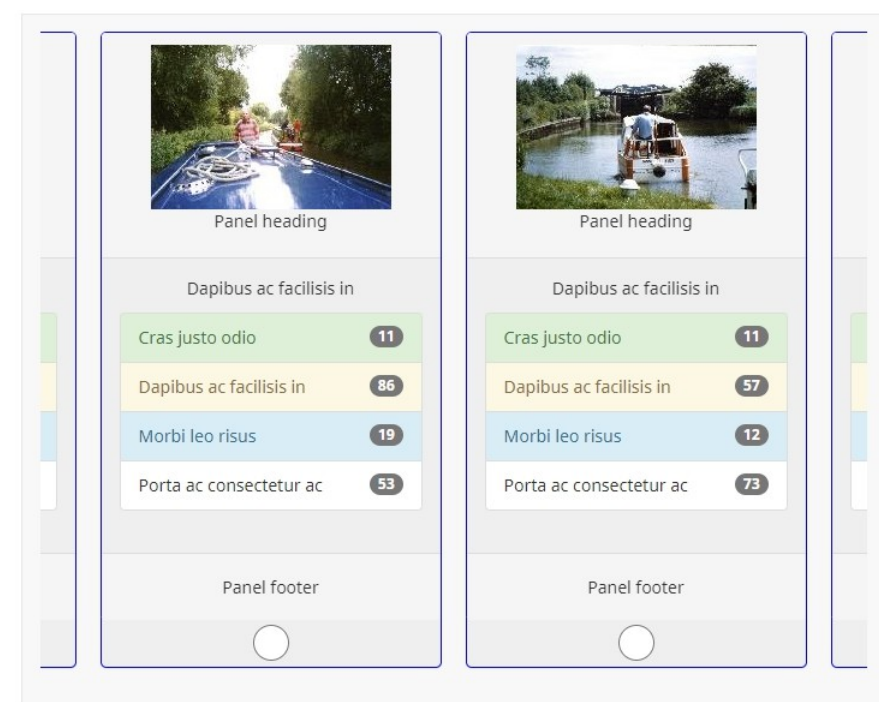
<https://github.com/ganlanyuan/tiny-slider>

11.2.1.1. Layout-Möglichkeiten

Nehmen wir einmal ein Beispiel, bei dem 4 Optionen gleichzeitig gezeigt werden sollen. Auf einen breiten Bildschirm wird ganz normal angezeigt.



Sobald er aber schmaler wird, verringert sich die angezeigte Anzahl



Um dem Teilnehmer anzuzeigen, dass es rechts und/oder links noch weitere Optionen gibt, werden diese im Anschnitt angezeigt.

Dies ist eine Einstellungssache, man muss sie also nicht zeigen.

Mit der Maus oder dem Finger bewegt man nun die verfügbaren Antwortoptionen hin und her.

[Klick für Beispiel-Video](#)

11.2.1.2. Implementierung

Der grundsätzliche Aufbau ist dieser:

Auch hier wird die Struktur in den Fragetext eingebaut.

Schematisch:

```
<div class="slider" id="slider">
  <div class="item">
    <div class="img-1">
      Hier wird mit Inhalt gefüllt
    </div>
  </div>

  <div class="item">
    <div class="img-2">
      Hier wird mit Inhalt gefüllt
    </div>
  </div>

  <div class="item">
    <div class="img-3">
      Hier wird mit Inhalt gefüllt
    </div>
  </div>
</div>
```

Das Einfügen der Radio-Buttons geschieht wie im vorigen Beispiel:

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready ajax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');

    // Move the radios
    $('.question-text div.item .img-1', thisQuestion).append($('.subquestion-list .answers-list:eq(0) .answer-item:eq(0) *', thisQuestion));
    $('.question-text div.item .img-2', thisQuestion).append($('.subquestion-list .answers-list:eq(0) .answer-item:eq(1) *', thisQuestion));
    $('.question-text div.item .img-3', thisQuestion).append($('.subquestion-list .answers-list:eq(0) .answer-item:eq(2) *', thisQuestion));
  });
</script>
```

Wie man sieht, werden hier genau die Klassen „img-1“, „img-2“, ... angesprochen, die oben in der Struktur angelegt wurden.

Da es sich bei der Frage um eine Matrix-Frage mit einer einzigen Teilfrage und drei Antwortoptionen handelt, ist die „answer-list:eq(0)“ – also die erste (und hier einzige) – und die „answer-item:eq(x)“ werden – ebenfalls bei 0 beginnend – eingetragen.

Nun fehlt noch das Wichtigste, nämlich die Karussell-Funktion.

Dazu muss zunächst die Bibliothek eingebunden werden.

[Klick für Beispiel-Video](#)

Hier vom CDN (Content Delivery Network) heruntergeladen

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/tiny-slider/2.9.4/tiny-slider.css" rel="stylesheet" />
<script src="https://cdnjs.cloudflare.com/ajax/libs/tiny-slider/2.9.2/min/tiny-slider.js"></script>
```

Es ist allerdings besser, diese Dateien vom oben angegebenen Link herunterzuladen, auf dem eigenen Server zu speichern und dann von dort einzubinden.

Nun folgt noch der Aufruf mit einigen Optionen

```
<script type="text/javascript" charset="utf-8">
  var slider = tns({
    "container": "#slider",
    "responsive": {
      "500": {
        "items": 1,
      },
      "1000": {
        "items": 2
      },
      "1680": {
        "items": 3
      }
    },
    "edgePadding": 15,
    "loop" : false,
    "mouseDrag": true,
    "swipeAngle": false,
    "controls" : false,
    "nav" : false,
    "arrowKeys" : true,
    "speed": 500
  });
</script>
```

Für die Bedeutung dieser Optionen lese man ebenfalls die o.a. Dokumentation und schaue die Demo an.

11.2.2. Slick-Bibliothek


<http://kenwheeler.github.io/slick/>

Die Features dieser Bibliothek sind ähnlich wie oben; es gibt kleine Unterschiede, und jeder möge für sich entscheiden, welche besser geeignet ist.

11.2.2.1. Layout-Möglichkeiten

Hier möchte ich einmal eine Matrix-Frage darstellen, welche nach jedem Klick weiterscrollt zum nächsten Item. So können diese langweiligen Matrizen vielleicht ein bisschen aufgelockert werden.

Zunächst ein Bildchen



Boot 4

Manchmal ist es auch feucht

sehr gut	gut	eher schlecht als recht	schlecht
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Statt der üblichen Boot-Bilder können hier natürlich auch Statements gezeigt werden, typographisch aufgehübscht oder wie auch immer.

11.2.2.2. Implementierung

Der grundsätzliche Aufbau ist dieser:

Auch hier wird die Struktur in den Fragetext eingebaut.

Die einzelnen Seiten werden als bootstrap panel dargestellt. Man lese dazu hier:

<https://getbootstrap.com/docs/3.4/components/#panels>

Schematisch (hier etwas ausführlicher, da für die vier Radio-Buttons pro Seite zusätzlich eine Tabelle eingefügt ist).

```
<div class="slider">
  <div class="item">
    <div class="panel panel-default img-1">
      <div class="panel-heading">
        
        <p>Boot 1</p>
      </div>

      <div class="panel-body">
        <p>Ein Text</p>
      </div>

      <div class="panel-footer">
        <table class="table" style="margin-bottom:0;border-top: 0;">
          <tbody>
            <tr>
              <td style="width:10%"> </td>
              <td style="width:20%;text-align:center">sehr gut</td>
              <td style="width:20%;text-align:center">gut</td>
              <td style="width:20%;text-align:center">eher schlecht als recht</td>
              <td style="width:20%;text-align:center">schlecht</td>
              <td style="width:10%"> </td>
            </tr>
            <tr>
              <td style="min-width:10%"> </td>
              <td style="min-width:20%;text-align:center"> </td>
              <td style="min-width:20%;text-align:center"> </td>
              <td style="min-width:20%;text-align:center"> </td>
              <td style="min-width:20%;text-align:center"> </td>
              <td style="min-width:10%"> </td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>

  <div class="item">
    <div class="panel panel-default img-2">
```

```

    <div class="panel-heading">
        ...
    </div>

    <div class="panel-body">
        <p>Ein Text</p>
    </div>

    <div class="panel-footer">
        <table>
            ...
        </table>
    </div>
</div>
</div>

<div class="item">
    <div class="panel panel-default img-3">
        <div class="panel-heading">
            ...
        </div>

        <div class="panel-body">
            <p>Ein Text</p>
        </div>

        <div class="panel-footer">
            <table>
                ...
            </table>
        </div>
    </div>
</div>
</div>

```

Auch hier kommen jetzt die Radio-Buttons hinzu.

Es handelt sich wiederum um eine Matrix, diesmal aber mit mehreren Teilfragen.

```
<script type="text/javascript" charset="utf-8">
$(document).on('ready pjax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');

    $('.question-text div.item .img-1 table:eq(0) tr:eq(1) td:eq(1)', thisQuestion).append($
('subquestion-list .answers-list:eq(0) .answer-item:eq(0) *', thisQuestion));

    $('.question-text div.item .img-1 table:eq(0) tr:eq(1) td:eq(2)', thisQuestion).append($
('subquestion-list .answers-list:eq(0) .answer-item:eq(1) *', thisQuestion));

    $('.question-text div.item .img-1 table:eq(0) tr:eq(1) td:eq(3)', thisQuestion).append($
('subquestion-list .answers-list:eq(0) .answer-item:eq(2) *', thisQuestion));

    $('.question-text div.item .img-1 table:eq(0) tr:eq(1) td:eq(4)', thisQuestion).append($
('subquestion-list .answers-list:eq(0) .answer-item:eq(3) *', thisQuestion));

    $('.question-text div.item .img-2 table:eq(0) tr:eq(1) td:eq(1)', thisQuestion).append($
('subquestion-list .answers-list:eq(1) .answer-item:eq(0) *', thisQuestion));

    $('.question-text div.item .img-2 table:eq(0) tr:eq(1) td:eq(2)', thisQuestion).append($
('subquestion-list .answers-list:eq(1) .answer-item:eq(1) *', thisQuestion));

    $('.question-text div.item .img-2 table:eq(0) tr:eq(1) td:eq(3)', thisQuestion).append($
('subquestion-list .answers-list:eq(1) .answer-item:eq(2) *', thisQuestion));

    $('.question-text div.item .img-2 table:eq(0) tr:eq(1) td:eq(4)', thisQuestion).append($
('subquestion-list .answers-list:eq(1) .answer-item:eq(3) *', thisQuestion));

    $('.question-text div.item .img-3 table:eq(0) tr:eq(1) td:eq(1)', thisQuestion).append($
('subquestion-list .answers-list:eq(2) .answer-item:eq(0) *', thisQuestion));

    $('.question-text div.item .img-3 table:eq(0) tr:eq(1) td:eq(2)', thisQuestion).append($
('subquestion-list .answers-list:eq(2) .answer-item:eq(1) *', thisQuestion));

    $('.question-text div.item .img-3 table:eq(0) tr:eq(1) td:eq(3)', thisQuestion).append($
('subquestion-list .answers-list:eq(2) .answer-item:eq(2) *', thisQuestion));

    $('.question-text div.item .img-3 table:eq(0) tr:eq(1) td:eq(4)', thisQuestion).append($
('subquestion-list .answers-list:eq(2) .answer-item:eq(3) *', thisQuestion));

});
</script>
```

Wie oben benutzen wir wieder die Klassen „img-x“ um die Karussellseite zu bestimmen.

„table:eq(0)“ findet nun die erste Tabelle auf dieser Seite (es gibt ja auch nur eine)

„tr:eq(1)“ findet die zweite Zeile (in der ersten Zeile stehen ja die Antworttexte)

„td:eq(x)“ findet die entsprechende Spalte (hier beginnt es bei der zweiten Spalte; die erste Spalte ist ja eine Leerspalte ohne Text)

Dahinein wird jetzt eingesetzt

„.answers-list:eq(x)“ Dies ist die erste (0), zweite (1) oder dritte (2) Teilfrage.

„.answer-item:eq(x)“ Dies ist die erste (0), zweite (1), dritte (2) oder vierte (3) Antwortoption.

Bleibt noch die Funktionsfähigkeit des Karussells und das automatische Weiterscrollen.

Für das Karussell benötigen wir wieder die entsprechenden Bibliotheken

```
<link href="//cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.css" rel="stylesheet" type="text/css" />
<link href="//cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick-theme.css" rel="stylesheet"
type="text/css" />
<script type="text/javascript" src="//cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.min.js">
</script>
```

Auch hier der Hinweis: Besser diese Dateien von der o.a. Seite herunterladen und auf dem eigenen Server speichern.

Das eigentliche Script ist wieder sehr ähnlich zu oben:

```
<script type="text/javascript">
$( '.slider' ).slick({
  dots: false,
  infinite: false,
  slidesToShow: 1,
  slidesToScroll: 1,
  fade: false,
  speed: 1000,
  swipe: false,
  centerMode: true,
  centerPadding: '30px',
  arrows: true,
});
</script>
```

Für die einzelnen Optionen schaue man in die o.a. Dokumentation.

Und das Scrollen wird hiermit erledigt

```
<script type="text/javascript">
  $(document).on('ready ajax:scriptcomplete',function(){

    // Hide the submit button
    $('#ls-button-submit').hide();
    // Sollte auch ein „Zurück“-Button vorhanden sein, wird dieser ebenfalls versteckt
    $('#ls-button-previous').hide();

    // Listener on the (hidden) radio inputs
    $('#question{QID} :radio').on('click', function(e) {
      var pos=$('#.slider').slick('slickCurrentSlide');
      // Wenn das letzte Item erreicht ist, gehe zur nächsten Frage
      if(pos==3) {
        $('#ls-button-submit').trigger('click');
      } else {
        $('#.slider').slick('slickNext');
      }
    });
  });
</script>
```

12. Erläuterungen mittels Tooltips, Popover, Akkordeon, Modal-Dialog

Hier sollen einige Möglichkeiten gezeigt werden, die es ermöglichen, zusätzliche Informationen bei Bedarf anzuzeigen.

Da LimeSurvey auf der bootstrap-Bibliothek beruht, sind alle diese Beispiele nahezu ohne zusätzliches javascript zu realisieren.

Nur bei „Tooltips“ und Popover“ benötigt man eine Zeile, je nachdem, was benutzt wird. Auf der entsprechenden bootstrap-Seite sind eine Menge zusätzlicher Optionen beschrieben.

```
<script>
$('[data-toggle="popover"]').popover();
$('[data-toggle="tooltip"]').tooltip();
</script>
```

Hier werden sie nur gezeigt, sowie der Link zum entsprechenden Beispiel auf der „bootstrap“-Seite. Dort entnehme man Näheres sowie im Code der einzelnen Fragen in der Beispiel-Datei.


12.1. Collapse (oder Akkordeon)


<https://getbootstrap.com/docs/3.4/javascript/#collapse>


Adalbert Ameise möchte sich auf die Züchtung der "Braunen Rossameise" spezialisieren.

Wie stehen Sie als Mitglied des Ameisenzüchtervereins zu dieser Idee?

Klick für weitere Informationen

 Positiv

 Neutral

 Negativ

Und der Klick bringt dann sie Erläuterung auf den Bildschirm

Adalbert Ameise möchte sich auf die Züchtung der "Braunen Rossameise" spezialisieren.


Wie stehen Sie als Mitglied des Ameisenzüchtervereins zu dieser Idee?


Klick für weitere Informationen


Rossameisen (Camponotus) oder Holzameisen sind eine Gattung der Ameisen (Formicidae) aus der Unterfamilie der Schuppenameisen (Formicinae). Ihr gehören weltweit über 1000 Arten an, in der Paläarktis kommen davon mindestens 100 sehr große Arten vor.

Die Antennen entspringen deutlich hinter dem Clypeushinterrand. Die Öffnung der Metapleuraldrüse fehlt. Im Gegensatz zu den Waldameisen ist die Oberseite des Thorax gleichförmig gebogen, ohne tiefere Einkerbungen.

Die Schwarze Rossameise (Camponotus herculeanus) und die Braunschwarze Rossameise (Camponotus ligniperda) sind die größten mitteleuropäischen Ameisen. Sie werden bis zu 18 Millimeter lang. Die Stöpselkopfameise (Camponotus truncatus) weist als einzige in Deutschland heimische Art einen Kastendimorphismus auf.

 Positiv

 Neutral

 Negativ

Wie später noch gezeigt wird, muss es kein Button sein, auf den geklickt wird. Kann man auch als Link gestalten.

12.2. Modaler Dialog

<https://getbootstrap.com/docs/3.4/javascript/#modals>

Adalbert Ameise möchte sich auf die Züchtung der "Braunen Rossameise" spezialisieren.

Wie stehen Sie als Mitglied des Ameisenzüchtervereins zu dieser Idee?

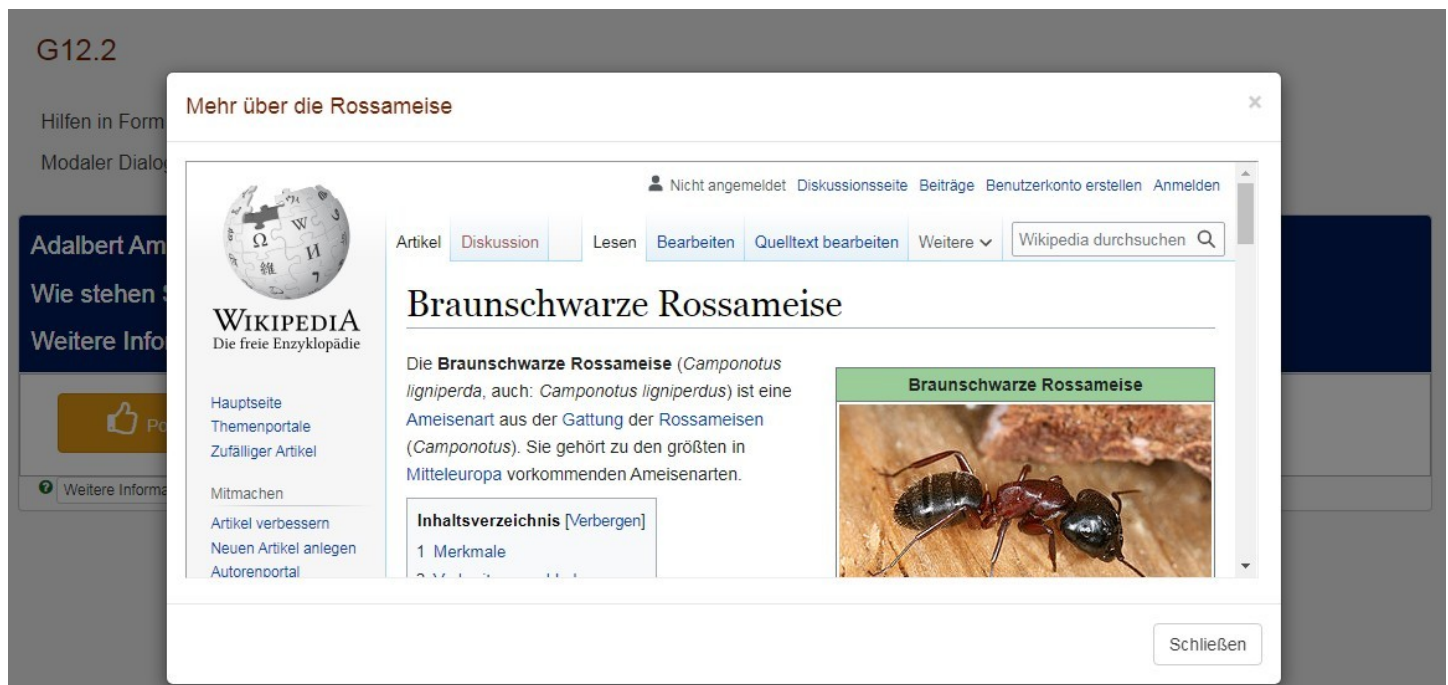
Weitere Informationen zur "Braunen Rossameise" finden sie [hier](#)

☐ Positiv ☐ Neutral ☐ Negativ

[Weitere Informationen](#)

Hier ist das „Klick“-Element einmal oben im Fragetext als Link, einmal im Hilfetext als Button eingebaut. Der Klick öffnet nun den Modal-Dialog, welcher hier die Wikipedia-Seite lädt und anzeigt.

Das kann natürlich auch alles andere sein, insbesondere einfacher Text.



12.3. Popover als Klick-Element in Matrix

<https://getbootstrap.com/docs/3.4/javascript/#popovers>

Hier soll gezeigt werden, dass das Layout dieser sogenannten „Popover“ mit den Standard-HTML zu beeinflussen ist.

Welche der folgenden Ameisen-Arten sehen Sie in der Favoritenrolle in den einzelnen Disziplinen?			
	Blutrote Raubameise	Pharaoameise	Schwarze Rossameise
	<button>Erläuterung</button>	<button>Erläuterung</button>	<button>Erläuterung</button>
Speerwerfen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gewichtheben	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kanuslalom	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dressurreiten	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

🔗 Für eine Erläuterung der Art können Sie jeweils den Button 'Erläuterung' anklicken.

Erläuterung unterhalb des Buttons (mit Aufzählung)

Welche der folgenden Ameisen-Arten sehen Sie in der Favoritenrolle in den einzelnen Disziplinen?			
	Blutrote Raubameise	Pharaoameise	Schwarze Rossameise
	<button>Erläuterung</button>	<button>Erläuterung</button>	<button>Erläuterung</button>
Speerwerfen	<input type="radio"/>	<div>Erläuterung<ul style="list-style-type: none">• Am gelben Körper mit dunklem Hinterleib zu bestimmen• Erreichen geringe Länge von bis zu 4,5 mm• Suchen verstärkt menschliche Nahrungsmittel• Können hoch infektiöse Krankheiten übertragen• Befall sollte deshalb zeitnah bekämpft werden</div>	<input type="radio"/>
Gewichtheben	<input type="radio"/>		<input type="radio"/>
Kanuslalom	<input type="radio"/>		<input type="radio"/>
Dressurreiten	<input type="radio"/>		<input type="radio"/>

🔗 Für eine Erläuterung der Art können Sie jeweils den Button 'Erläuterung' anklicken.

Erläuterung links (mit Bild)

Welche der folgenden Ameisen-Arten sehen Sie in der Favoritenrolle in den einzelnen Disziplinen?			
	Blutrote Raubameise	Pharaoameise	Schwarze Rossameise
	<button>Erläuterung</button>	<button>Erläuterung</button>	<button>Erläuterung</button>
Speerwerfen	<input type="radio"/>	<div>Erläuterung<p>Die Schwarze Rossameise heißt in der Botanik <i>Camponotus herculeanus</i> und lebt überwiegend in bergigen Regionen.</p><p>Sie zählt zu den größten Ameisenarten, welche in Deutschland heimisch sind.</p></div>	<input type="radio"/>
Gewichtheben	<input type="radio"/>		<input type="radio"/>
Kanuslalom	<input type="radio"/>		<input type="radio"/>
Dressurreiten	<input type="radio"/>		<input type="radio"/>

🔗 Für eine Erläuterung der Art können Sie jeweils den Button 'Erläuterung' anklicken.

12.4. Tooltips und Popover im Frage- und Antworttext

<https://getbootstrap.com/docs/3.4/javascript/#tooltips>

Größte in Deutschland heimische Art

Rossameise oder doch eine andere Art?

Bitte wählen Sie eine der folgenden Antworten:

☐ Rossameise

☐ Rotgelbe Knotenameise

☐ Sonstige

Tippen in Form von Tooltips, Popovers, Modals und Dialogen

Tooltips und Popover

Diese Arten sind alle kleiner als die Rossameise

Rossameise oder doch eine andere Art?

Bitte wählen Sie eine der folgenden Antworten:

☐ Rossameise

☐ Rotgelbe Knotenameise

☐ Sonstige

Rossameise oder doch eine andere Art?

Bitte wählen Sie eine der folgenden Antworten:

☐ Rossameise **Ernährt sich vorwiegend von Rosskastanien**

☐ Rotgelbe Knotenameise

☐ Sonstige

In diesem benutzten Theme sind die Farben nicht vernünftig angepasst.

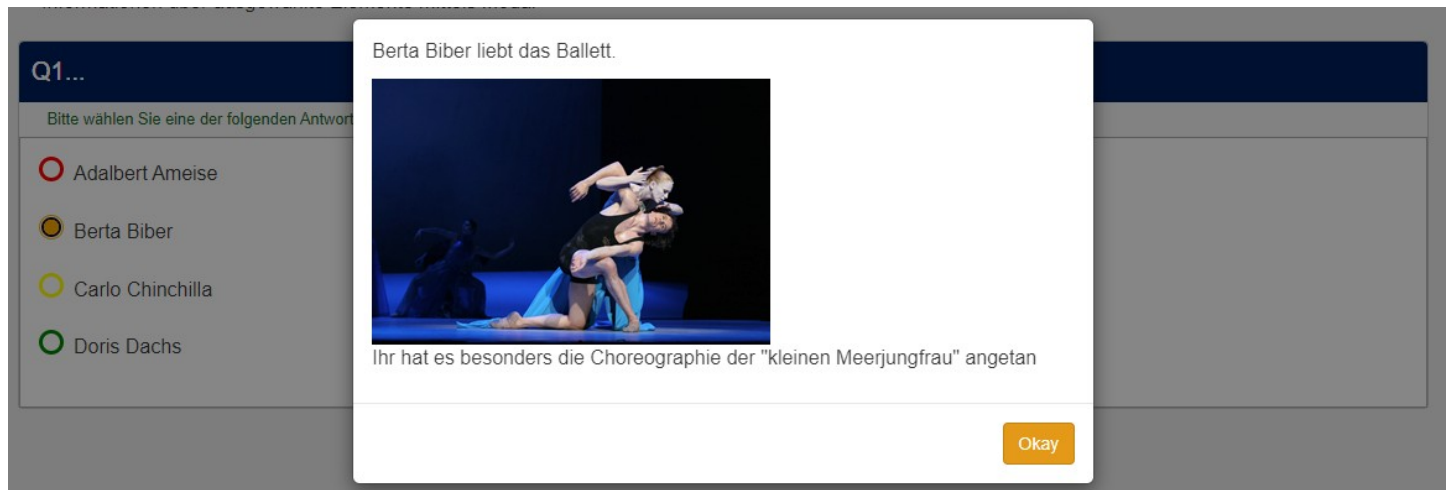
In der Beispiel-Datei (mit Theme „fruity“) sieht dies anders aus.

12.5. Modal-Dialoge, Alerts als Reaktion auf „Klick“

Hier zeige ich einmal, was es so gibt, um auf einen Klick in einer Frage vom Typ „List(Radio)“ zu reagieren.

12.5.1. Modal-Dialoge als Reaktion auf Klick

Eine größere Menge an Informationen kann man in einem Dialog anzeigen.



Für dieses Konstrukt werden nach der eigentlichen Frage noch ebenso viele Fragen vom Typ „Textanzeige“ angelegt, in deren Text der Inhalt des Dialogs eingefügt wird.

Zugegeben, ich weiß auch nicht, wozu man dies benutzen sollte; man kann denselben Aufruf des Modals auch – vielleicht sogar besser – mithilfe des Links (siehe 12.2.) erreichen.

Der Link würde dann in der Antwortoption eingefügt.

Sei's drum.

Und in die Frage wird dieses javascript (im Quellcode-Modus) eingefügt

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready pjax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');

    // Insert the modals
    $('input:radio', thisQuestion).each(function(i) {
      $('body').append('<div class="modal inserted-modal fade" id="inserted-modal-'+i+1+'" aria-
hidden="true" data-backdrop="static" data-keyboard="false">\
        <div class="modal-dialog" role="document">\
          <div class="modal-content">\
            <div class="modal-body"></div>\
            <div class="modal-footer">\
              <button type="button" class="btn btn-primary"
data-dismiss="modal">Okay</button>\
            </div>\
          </div>\
        </div>\
      </div>');
      $('#inserted-modal-'+i+1+' .modal-body').append($
(thisQuestion).nextAll('.boilerplate:eq('+i+')').find('.question-text'));
    });

    // Remove the text-display questions
    $(thisQuestion).nextAll('.boilerplate:lt('+$('input:radio', thisQuestion).length+')').remove();

    // Listeners on the radios
    $('input:radio', thisQuestion).on('click', function(e) {
      var thisIndex = $(this).closest('.answer-item').index();
      $('#inserted-modal-'+thisIndex+1+'').modal('show');
    });
  });
</script>
```

Wichtig: Die folgenden Textanzeige-Fragen müssen den Namen „boilerplate1“, „boilerplate2“, usw. haben.

12.5.2. Alerts als Reaktion auf Klick

Ähnlich ist dies: Nach Klick auf einen Radio-Button erscheint ein sogenannter Alert, der nach einer bestimmten Zeit wieder verschwindet.

Q3...

Gratulation! Durch die Wahl von Adalbert Ameise bekommen Sie dreimal pro Woche eine kross gegrillte Ameisenhaxe frei Haus geliefert

Bitte wählen Sie eine der folgenden Antworten:

☒ Adalbert Ameise

☐ Berta Biber

☐ Carlo Chinchilla

☐ Doris Dachs

Q3...

Information! Das selbstgehäkelte Nachtjäckchen aus Chinchillawolle kostet zur Zeit nur €239,99

Bitte wählen Sie eine der folgenden Antworten:

☐ Adalbert Ameise

☐ Berta Biber

☒ Carlo Chinchilla

☐ Doris Dachs

Man kann natürlich den Alert auch so einstellen, dass er durch einen Klick (auf das kleine Kreuz rechts) selbst geschlossen werden muss.

Q3...

Gefahr! Diesen blöden Alert muss man ja noch mit der Hand schließen ×

Bitte wählen Sie eine der folgenden Antworten:

☐ Adalbert Ameise

☐ Berta Biber

☐ Carlo Chinchilla

☒ Doris Dachs

Die Alerts werden einfach als DIVs im Quellcode der Frage angelegt.

```
<div class="conditional-text alert alert-success" role="alert"><strong>Gratulation!</strong> Durch die Wahl von Adalbert Ameise bekommen Sie dreimal pro Woche eine kross gegrillte Ameisenhaxe frei Haus geliefert</div>

<div class="conditional-text alert alert-warning" role="alert"><strong>Achtung!</strong> Die immer in den höchsten Tönen gelobte Biberschwanzsuppe ist nicht zu empfehlen</div>

<div class="conditional-text alert alert-info" role="alert"><strong>Information!</strong> Das selbstgehäkelte Nachtjäckchen aus Chinchillawolle kostet zur Zeit nur €239,99</div>
```

Beim vierten sieht man den zusätzlichen Teil, der das Kreuz zum Schließen erzeugt.

```
<div class="conditional-text alert alert-danger alert-dismissible" role="alert"><button aria-label="Close" class="close" data-dismiss="alert" type="button"><span aria-hidden="true">x</span></button><strong>Gefahr!</strong> Diesen blöden Alert muss man ja noch mit der Hand schließen</div>
```

Nun gibt es noch ein bisschen javascript, welches die Alerts einblendet – und auch wieder ausblendet.

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready pjax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');
    $('.question-text .conditional-text', thisQuestion).hide();

    // Listeners on the radios
    $('input:radio', thisQuestion).on('click', function(e) {
      $('.question-text .conditional-text', thisQuestion).hide();
      var thisIndex = $(this).closest('.answer-item').index();
      $('.question-text .conditional-text:eq('+thisIndex+')', thisQuestion).fadeIn(300);
    });
  });
</script>
```

Mit diesem kleinen IF schließen wir nur die ersten drei Alerts (die Zählung beginnt wie üblich bei „0“).

Und es sind 5000 Millisekunden bis zur Schließung eingestellt.

```
    if(thisIndex<3) {
      setTimeout(function () {
        $('.question-text .conditional-text:eq('+thisIndex+')', thisQuestion).fadeOut(500);
      }, 5000);
    }
  });
</script>
```

Zur Anlage der Alerts lese man auch

<https://getbootstrap.com/docs/3.4/components/#alerts>

13. Farben, Farben, Farben

13.1. Farbige Radio-Buttons und Checkboxes

Die einzelnen Radio-Buttons und Checkboxes können in unterschiedlichen Farben dargestellt werden:

Q1...
Bitte wählen Sie eine der folgenden Antworten:

☐ Adalbert Ameise
☐ Berta Biber
☐ Carlo Chinchilla
☐ Doris Dachs

Q2...
Bitte wählen Sie die zutreffenden Antworten aus:

☐ Adalbert Ameise
☐ Berta Biber
☐ Carlo Chinchilla
☐ Doris Dachs

Natürlich können die Antworttexte ebenfalls farbig sein; doch dazu sind ja nur die Antwortoptionen mit einem

`...`

zu umschließen.

Diese Farbgebung erreicht man einfach durch ein bisschen css (im Quellcode der Frage).

Hier für „list(radio)“

```
<style type="text/css">#question{QID} .radio-item label {  
    padding-left: 8px;  
}  
#question{QID} .radio-item label::before {  
    width: 20px;  
    height: 20px;  
    left: 0;  
    top: 0;  
    border: 3px solid #ccc;  
}  
#question{QID} .radio-item label::after {  
    width: 12px;  
    height: 12px;  
    left: 4px;
```



```

    top: 4px;
  }

  #question{QID} .radio-item:nth-child(1) label::before {
    border-color: red;
  }
  #question{QID} .radio-item:nth-child(2) label::before {
    border-color: orange;
  }
  #question{QID} .radio-item:nth-child(3) label::before {
    border-color: yellow;
  }
  #question{QID} .radio-item:nth-child(4) label::before {
    border-color: green;
  }

  #question{QID} .radio-item:nth-child(1) input[type="radio"]:checked + label::after {
    background: red;
  }
  #question{QID} .radio-item:nth-child(2) input[type="radio"]:checked + label::after {
    background: orange;
  }
  #question{QID} .radio-item:nth-child(3) input[type="radio"]:checked + label::after {
    background: yellow;
  }
  #question{QID} .radio-item:nth-child(4) input[type="radio"]:checked + label::after {
    background: green;
  }
}
</style>

```

Man sieht, dass die einzelnen Buttons mit „nth-child(x)“ angesprochen werden.

Zählung beginnt hier bei „1“.

Falls nur das Innere des Buttons farbig (und in einer einzigen Farbe) dargestellt werden soll, genügt dies

```

<style type="text/css">
  .radio-item label::before {
    border-color: black;
  }
  .radio-item input[type="radio"]:checked + label::before {
    border-color: red;
  }
  .radio-item input[type="radio"]:checked + label::after {
    background: red;
  }
</style>

```

Um Checkboxes farbig zu gestalten, wird im css_Text nur jedes „radio“ durch „checkbox“ ersetzt.

13.2. Dual Matrix

Um die beiden Skalen einer Dual Matrix optisch besser zu trennen, hier zwei Beispiele

13.2.1. Mit Trennern

Dual Matrix mit farbigen Trennern							
	Größe				Gewicht		
	unter 1.60m	1.60m-1.80m	über 1.80m		unter 50kg	50kg - 80kg	über 80kg
Adalbert Ameise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Berta Biber	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Carlo Chinchilla	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Doris Dachs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Die einzufügenden Scripte (javascript und css) sehe man sich in der Beispiel-Umfrage an.

Wichtig: In früheren Versionen von 3.x. gab es einen Schreibfehler in der Programmierung.

Statt mit

```
.array-flexible-dual-scale
```

wurde die Klasse mit

```
.array-flexible-duel-scale
```

angesprochen.

Außerdem sieht man hier, dass eine css-Klasse „.bordered-array“ eingeführt wird.

Diese muss in den Settings der Frage unter „Anzeige“ eingefügt werden.

CSS-Klasse(n) ⓘ

bordered-array

13.2.2. Mit farbigen Bereichen

Dual Matrix mit farbigen Bereichen							
	Größe				Gewicht		
	unter 1.60m	1.60m-1.80m	über 1.80m		unter 50kg	50kg - 80kg	über 80kg
Adalbert Ameise							
Berta Biber							
Carlo Chinchilla							
Doris Dachs							

Dies ist vom script her deutlich kürzer, nämlich

```
<script type="text/javascript" charset="utf-8">
  $(document).on('ready ajax:scriptcomplete',function(){
    $('.coloured-array.array-flexible-dual-scale').each(function(i) {
      // Assign column-specific classes
      $('table.subquestion-list tr.ls-heading:not(.groups)', this).each(function(i) {
        $('.answer-text', this).each(function(i){
          $(this).addClass('column-'+(i+1));
        });
      });
    });
  });
</script>
```

Und das css

```
<style type="text/css">
.coloured-array.array-flexible-dual-scale .column-1 { background-color: #FF9933;}
.coloured-array.array-flexible-dual-scale .column-2 { background-color: #FF6600;}
.coloured-array.array-flexible-dual-scale .column-3 { background-color: #FF0000;}
.coloured-array.array-flexible-dual-scale .column-4 { background-color: #0066ff;}
.coloured-array.array-flexible-dual-scale .column-5 { background-color: #4c99ff;}
.coloured-array.array-flexible-dual-scale .column-6 { background-color: #99ccff;}
td.answer_cell_1_1 { background-color: #FF9933;}
td.answer_cell_1_2 { background-color: #FF6600;}
td.answer_cell_1_3 { background-color: #FF0000;}
td.answer_cell_2_1 { background-color: #0066ff;}
td.answer_cell_2_2 { background-color: #4c99ff;}
td.answer_cell_2_3 { background-color: #99ccff;}
</style>
```

Analog zum vorigen Beispiel wird hier die css-Klasse „coloured-array“ in „Anzeige / css-Klassen“ eingefügt.

13.3. Matrix mit Färbung von Header und Zellen

Etwas ähnliches ist natürlich auch mit normalen Matrizen möglich.

13.3.1. Matrix mit Header und Farbgebung beim Überstreichen und bei Auswahl

Matrix mit farbigem Header und Farbgebung beim Herüberstreichen und nach Auswahl											
	10	9	8	7	6	5	4	3	2	1	0
Adalbert Ameise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Berta Biber	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Carlo Chinchilla	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Doris Dachs	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bei diesem Theme (abgeleitet von „TFR-retro“ von „tools for research“) funktioniert die Farbgebung beim Hinüberstreichen nicht, in „vanilla“ dagegen schon.

Matrix mit farbigem Header und Farbgebung beim Herüberstreichen und nach Auswahl											
	10	9	8	7	6	5	4	3	2	1	0
Adalbert Ameise	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Berta Biber	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Carlo Chinchilla	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Doris Dachs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bei „Adalbert Ameise“ ist die „9“ ausgewählt, der Cursor befindet sich aber zur Zeit über der „2“.

Das javascript ist recht kurz – es macht eigentlich nichts anderes als die Spalten durchnummerieren und beim Klick die Klasse zu ändern.

```
<script type="text/javascript" charset="utf-8">

$(document).on('ready pjax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');

    // Add a question class
    thisQuestion.addClass('custom-array');
```

```

// Column-specific classes
$('table.subquestion-list tr', thisQuestion).each(function(i) {
    $('th, td', this).each(function(i) {
        $(this).addClass('column-' + i);
    });
});

// Listener on the radios
$('input:radio', thisQuestion).on('click', function(i) {
    $(this).closest('tr').find('.active-item').removeClass('active-item');
    $(this).closest('td').addClass('active-item');
});
});
</script>

```

Dagegen ist das css etwas länger.

Im unteren Bereich sieht man, dass die Farben für die aktive Zelle und für die Zelle, über die der Cursor gerade geht, immer gleich sind.

Dies könnte man natürlich auch ändern, so dass evtl. die Farbe beim Darüberstreichen etwas blasser / transparenter ist.

```

<style type="text/css">.custom-array table.subquestion-list thead td,
.custom-array table.subquestion-list thead th {
    border-bottom-width: 8px;
    border-bottom-style: solid;
}

.custom-array table.subquestion-list thead .column-1 { border-bottom-color: #00A800; }
.custom-array table.subquestion-list thead .column-2 { border-bottom-color: #5AC100; }
.custom-array table.subquestion-list thead .column-3 { border-bottom-color: #9DD600; }
.custom-array table.subquestion-list thead .column-4 { border-bottom-color: #9CE400; }
.custom-array table.subquestion-list thead .column-5 { border-bottom-color: #E2EC00; }
.custom-array table.subquestion-list thead .column-6 { border-bottom-color: #ECEC00; }
.custom-array table.subquestion-list thead .column-7 { border-bottom-color: #ECE200; }
.custom-array table.subquestion-list thead .column-8 { border-bottom-color: #E4C900; }
.custom-array table.subquestion-list thead .column-9 { border-bottom-color: #D69D00; }
.custom-array table.subquestion-list thead .column-10 { border-bottom-color: #C15A00; }
.custom-array table.subquestion-list thead .column-11 { border-bottom-color: #A80200; }

.custom-array td.column-1:hover,
.custom-array td.active-item.column-1 { background-color: #00A800; }

.custom-array td.column-2:hover,
.custom-array td.active-item.column-2 { background-color: #5AC100; }

.custom-array td.column-3:hover,
.custom-array td.active-item.column-3 { background-color: #9DD600; }

.custom-array td.column-4:hover,
.custom-array td.active-item.column-4 { background-color: #9CE400; }

```

```
.custom-array td.column-5:hover,  
.custom-array td.active-item.column-5 { background-color: #E2EC00; }  
  
.custom-array td.column-6:hover,  
.custom-array td.active-item.column-6 { background-color: #ECEC00; }  
  
.custom-array td.column-7:hover,  
.custom-array td.active-item.column-7 { background-color: #ECE200; }  
  
.custom-array td.column-8:hover,  
.custom-array td.active-item.column-8 { background-color: #E4C900; }  
  
.custom-array td.column-9:hover,  
.custom-array td.active-item.column-9 { background-color: #D69D00; }  
  
.custom-array td.column-10:hover,  
.custom-array td.active-item.column-10 { background-color: #C15A00; }  
  
.custom-array td.column-11:hover,  
.custom-array td.active-item.column-11 { background-color: #A80200; }  
</style>
```

13.3.2. Matrix mit farbigen Spalten

Analog zur Dual Matrix auch hier die Möglichkeit, die gesamte Spalte durch Farben hervorzuheben.

Matrix mit farbigen Spalten					
	sehr schlecht	schlecht	neutral	gut	sehr gut
Adalbert Ameise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Berta Biber	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Carlo Chinchilla	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Doris Dachs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Man sieht, der javascript-Teil ist gleich wie im vorigen Beispiel

```
<script type="text/javascript" charset="utf-8">

$(document).on('ready ajax:scriptcomplete',function(){

    // Identify this question
    var thisQuestion = $('#question{QID}');

    // Add a question class
    thisQuestion.addClass('custom-array');

    // Column-specific classes
    $('table.subquestion-list tr', thisQuestion).each(function(i) {
        $('th, td', this).each(function(i) {
            $(this).addClass('column-'+i);
        });
    });

    // Listener on the radios
    $('input:radio', thisQuestion).on('click', function(i) {
        $(this).closest('tr').find('.active-item').removeClass('active-item');
        $(this).closest('td').addClass('active-item');
    });
});
</script>
```

Und das css ist auch schon mehr oder weniger bekannt.

Hier könnte man nun – statt nur den unteren Rand der Titelzeile zu färben, die gesamte Zelle färben, also auch „background-color“ benutzen.

```
<style type="text/css">.custom-array table.subquestion-list thead td,
    .custom-array table.subquestion-list thead th {
        border-bottom-width: 8px;
        border-bottom-style: solid;
    }

    .custom-array table.subquestion-list thead .column-5 { border-bottom-color: #00A800; }
    .custom-array table.subquestion-list thead .column-4 { border-bottom-color: #9CE400; }
    .custom-array table.subquestion-list thead .column-3 { border-bottom-color: #ECE200; }
    .custom-array table.subquestion-list thead .column-2 { border-bottom-color: #D69D00; }
    .custom-array table.subquestion-list thead .column-1 { border-bottom-color: #A80200; }
```

Und hier die einzelnen Zellen; wichtig ist: „1“, „2“, ... ist der Code der Antwortoption.

```
td.answer_cell_1 {background-color: #A80200;}
td.answer_cell_2 {background-color: #D69D00;}
td.answer_cell_3 {background-color: #ECE200;}
td.answer_cell_4 {background-color: #9CE400;}
td.answer_cell_5 {background-color: #00A800;}

</style>
```

Wichtig: Bei den einzelnen Zellen ist: „1“, „2“, ... der Code der Antwortoption.

14. Mehrere Fragen nebeneinander

14.1. Anwendungsbeispiel

Manchmal ist es wünschenswert mehrere inhaltlich zusammenhängende Fragen, die in der Horizontalen wenig Platz beanspruchen, nebeneinander darzustellen.

<p>Q1</p> <p>Bitte wählen Sie eine der folgenden Antworten:</p> <p><input type="radio"/> Option 1</p> <p><input type="radio"/> Option 2</p> <p><input type="radio"/> Option 3</p> <p><input type="radio"/> Option 4</p> <p><input type="radio"/> Option 5</p>	<p>Q2</p> <p>Bitte wählen Sie eine der folgenden Antworten:</p> <p><input type="radio"/> Option 1</p> <p><input type="radio"/> Option 2</p> <p><input type="radio"/> Option 3</p> <p><input type="radio"/> Option 4</p> <p><input type="radio"/> Option 5</p> <p><input type="radio"/> Option 6</p> <p><input type="radio"/> Option 7</p>	<p>Q3</p> <p>Bitte wählen Sie eine der folgenden Antworten:</p> <p><input type="radio"/> Option 1</p> <p><input type="radio"/> Option 2</p> <p><input type="radio"/> Option 3</p>
---	---	---

Das Besondere ist hier, dass die Höhe der drei Fragen gleich ist, obwohl eine unterschiedliche Anzahl von Antwortoptionen vorhanden ist.

Natürlich können die Fragen auch unterschiedliche Typen haben.

Oder man blendet die weiteren in Abhängigkeit von den vorherigen Fragen ein.

14.2. Implementierung

Alle Fragen sind in einer Gruppe und bekommen die css-Klasse „flex-layout“.

In den Fragetext der ersten Frage wird eingefügt (im Quellcode-Modus)

1. Dieser javascript-Teil

```
<script type="text/javascript" data-author="Tony Partner">
  $(document).ready(function() {

    // The number of questions to wrap in the flex container
    var numberQuestions = 3;

    // Identify the questions
    var thisQuestion = $('#question{QID}');
    var rowQuestions = thisQuestion.add(thisQuestion.nextAll('.question-container:lt('+(numberQuestions-1)+'')'));

    // Wrap the questions in a <div> element
    rowQuestions.wrapAll('<div class="question-flex-container" />');
  });
</script>
```

2. Dieser css-Teil

```
<style type="text/css">
@media only screen and (min-width: 576px) {
  .question-flex-container {
    display: flex;
    justify-content: space-evenly;
  }
  .flex-layout.question-container {
    width:100%;
  }
}
</style>
```

Wichtig ist, dass die Zeile „**var numberQuestions = 3**“ an die tatsächliche Anzahl angepasst wird.

15. Chat-Funktion

Dies ist nun wirklich etwas ganz Besonderes.

In einer Umfrage kann man eine Chat-Funktion einbauen, dass ein Teilnehmer bei Unklarheiten bei irgendwelchen Fragen mit einem Agenten einen Chat führen kann.

G8
Bewertung mittels Pins (Points of Interest)

Points of interest

-
+

Home / Chats list

Nick

E-mail

Department
Choose department ▾

Department group
Choose department group ▾

User
Select user ▾

User group
Select group ▾

Date range from
Eg 2021-09-08

Hour and minute from [23:03:55]
Select hour ▾ Select minute ▾

Date range to
Eg 2021-09-15

Hour and minute to [23:03:55]
Select hour ▾ Select minute ▾

Wait time
More than ▾ Less than ▾

Vote status
Any ▾

Chat status
Choose status ▾

Product
Select product ▾

Chat ID
<id> [<id>]

Chat duration
From ▾ Till ▾

Subject
Subject ▾

Proactive invitation
Choose proactive invitation ▾

IP

☐ Has unread messages

☐ Unanswered chat

☐ Anonymised

Select bot ▾

☐ Chats without an operator

☐ Chats with an operator

☐ Chats which had a bot

☐ Chats which did not had a bot

Search
Print
Export (4 chats)

<input type="checkbox"/>	Information	Operator	Department	Status
<input type="checkbox"/>	Visitor, 2021-08-29 21:33:09, Support	J...	Support	Closed chat
<input type="checkbox"/>	Visitor, 2021-03-11 16:23:19, Support	J...	Support	Closed chat
<input type="checkbox"/>	Visitor, 2021-03-11 15:41:51, Support	J...	Support	Closed chat
<input type="checkbox"/>	Visitor, 2021-03-11 15:11:23, Support	J...	Support	Closed chat

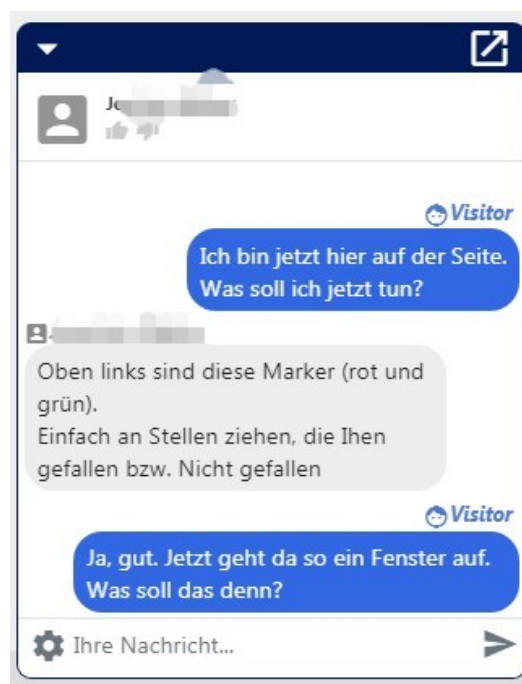
Close selected
Delete selected

Ziehen Sie die Symbole auf gefallene bzw. nicht gefallene Stellen des Bildes!

Zusätzlich ist ein Kommentar möglich.

Bei Fragen oder Schwierigkeiten können Sie sich gerne per E-Mail an uns wenden

Zunächst ist nur dieses Symbol unten rechts. Nach Klick öffnet sich die Chat-Box.



Als Chat-Programm wird hier „livehelperchat“ benutzt.

<https://livehelperchat.com/>

Das Einbinden in LimeSurvey gestaltet sich überraschend einfach:

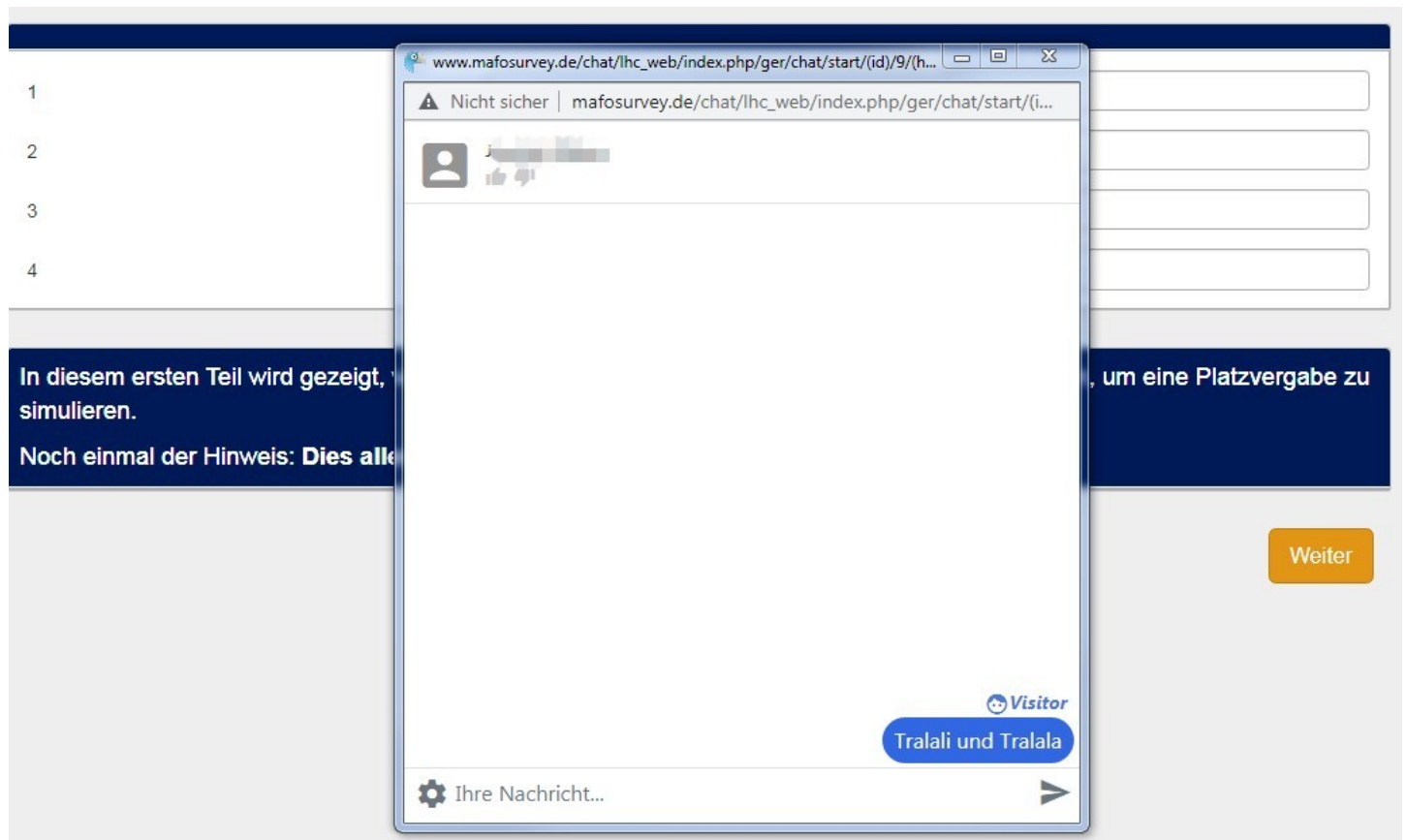
In der Chat-Installation wird der notwendige Code nach den Vorgaben generiert.

Dieser wird dann in die „footer.twig“ am Ende eingefügt.

```
<script>var LHC_API = LHC_API||{};
LHC_API.args =
{mode:'widget',lhc_base_url:'//www.mafoserver.de/chat/lhc_web/index.php/',wheight:450,wwidth:350,pheight:520,pwidth:500,leaveamessage:true,theme:2,check_messages:false,lang:'ger/'};

(function() {
var po = document.createElement('script'); po.type = 'text/javascript';
po.setAttribute('crossorigin','anonymous'); po.async = true;
var date = new Date();po.src = '//www.mafoserver.de/chat/lhc_web/design/defaulttheme/js/widgetv2/index.js?' +
(''+date.getFullYear() + date.getMonth() + date.getDate());
var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(po, s);
})();
</script>
```

Man kann noch weitere scripte einfügen, z.B. dass der Teilnehmer ein Pop-Up-Fenster öffnen kann, usw.



Vorteil: Man kann es selbst auf seinem eigenen Server installieren. Bei vielen anderen bleibt das Hosting des Paketes beim Hersteller.

Nachteil: Ich halte die Oberfläche für den Agenten für „etwas unübersichtlich“.

16. Der „ajax-call“

16.1. Vorbemerkung

Unter einem „ajax-call“ versteht man den Aufruf eines externen scriptes aus einem Programm heraus. „Ajax“ bedeutet: „Asynchronous JavaScript and XML“. Es werden also Daten zu einem Server geschickt bzw. von dort geholt oder auch eine Seite im Browser geändert, ohne dass die Seite neu geladen werden muss (asynchron).

Dies soll auch nicht weiter erläutert werden. Lieber will ich Anwendungsmöglichkeiten in LimeSurvey zeigen.

Man kann also z.B.

- eine Artikelnummer eingeben und in einer externen Datenbank den zugehörigen Artikel und die noch verfügbare Anzahl nachschlagen
- eine Zahl auf Gültigkeit prüfen (wenn dies ein sehr aufwendiges Verfahren ist)
- aus einem Pool von Objekten eins aussuchen, welches nur ein einziges Mal ausgewählt werden kann

Es gibt nun verschiedenen Methoden, wobei das generische \$.ajax sowie \$.get und \$.post wohl die meistbenutzten sind.

Genauer les man hier:

https://www.w3schools.com/jquery/jquery_ref_ajax.asp

Da bei einer Nutzung in LimeSurvey ja meist Daten gesendet werden, benutzen wir „\$.post“.

Der Standardaufruf sieht so aus:

Wir nehmen eine Frage vom Typ „kurzer Text“ und im Fragetext (Quellcode-Modus) steht dann dies:

```
<script type="text/javascript" charset="utf-8">
$(document).on('ready ajax:scriptcomplete',function(){
$.post('https://www.myServer/myScript.php' , { para1: "{para1}", para2: "{para2}" },function(data) {
    $('#question{QID} input[type="text"]').val(data)
});
});
</script>
```

Hier wird ein php-script aufgerufen; diesem werden zwei Werte übergeben (para1 und para2) und das Ergebnis (data) wird im Textfeld der aktuellen Frage gespeichert.

Das php script muss nun in der Lage sein, diese beiden Parameter zu verwerten und ein Ergebnis zurückgeben.

```
<?php
$para1 = $_POST["para1"];
$para2 = $_POST["para2"];

Irgendetwas wird gemacht und $result berechnet
und hier zurückgegeben

echo $result;
?>
```


16.2. Beispiel (Anzeige eines selbst hochgeladenen Bildes)

Da die Frage gerade im Forum auftauchte (und auch früher schon aufgeworfen wurde), hier als Beispiel die Anzeige eines Bildes, welches der Teilnehmer irgendwann früher in der Umfrage hochgeladen hat.

Man könnte sich vorstellen, dass jemand ein Bild hochlädt und es dann später in einer „Points of interest“-Frage (siehe Kapitel 8.) bearbeitet.

Dazu muss man zunächst wissen, wie das Hochladen und Abspeichern eines Bildes vor sich geht.


Nach der Auswahl des Bildes gibt es noch die Möglichkeit, einen Titel und einen Kommentar einzugeben.



Boot004_gross.jpg

Title

Comment


 Delete



Save changes

In der Übersicht wird noch einmal alles gezeigt.

Upload

Please upload at most one file

 Upload files

	Title	Comment	File name	
	Das ist der Titel	Und auch einen Kommentar gebe ich ein	Boot004_gross.jpg	 Edit

Next

Was steht nun in der Antworttabelle von LimeSurvey?

Dort sieht man dies, was zunächst ganz gut aussieht, denn der Name des Bildes ist da.

UL
Upload

Boot004_gross.jpg	0.08 Mb	Das ist der Titel	Und auch einen Kommentar gebe ich ein
-------------------	---------	-------------------	---------------------------------------

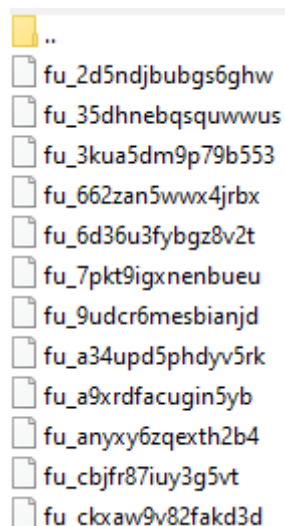
Wenn man nun aber versucht auf diese Variable zuzugreifen mit ExpressionScript/Manager, also ganz normal mit {UL}, sieht man – dummerweise – dies:

```
{ "title": "Das ist der Titel", "comment": "Und auch einen Kommentar gebe ich ein", "size": "82.2431640625", "name": "Boot004_gross.jpg", "filename": "fu_zhczmmv2brc2qfr", "ext": ".jpg" }
```

Gut, alle Informationen sind da, nur kein Bild.

Das Bild wird nämlich nicht in der Datenbank gespeichert, sondern im Verzeichnis „files“ der Umfrage. Und dazu auch noch unter dem oben angegebenen kryptischen file-name; gut, „fu“ steht für file-upload.

Hier sieht man dieses Verzeichnis mit einigen Dateien.



Gewohnterweise lädt man ein Bild mittels `` oder eben aus dem files-Verzeichnis mit ``

Das würde alles kein Problem sein. Das Problem ist: Die Datei hat keine Endung, aus welcher der Typ ersichtlich ist. Oben sieht man, dass die Endung separat gespeichert ist („ext“: „jpg“).

Und jetzt endlich kommt der „ajax-call“ ins Spiel.

Wir rufen ein Programm auf, dem wir die StudienID, den kryptischen file-name und die Endung übergeben.

Dieses nennt dann die Datei so um, dass die Endung angehängt wird.

Nun kann man sie wie gewohnt aufrufen.

16.3. Implementierung

Was benötigen wir?

1. die StudienID – diese haben wir, nämlich {SID}
2. die Endung – diese holen wir uns aus dem, was in {UL} steht, mittels einer simplen Gleichung (eqExt), nämlich
`{substr(UL,strpos(UL,'ext')+6,3)}`
Wir gucken also nach, wo der Text „ext“ beginnt, zählen sechs Zeichen hinzu, nämlich die drei von „ext“ und die drei von „:“. Dort beginnt also die Endung, welche drei Zeichen lang ist.
3. Den file-name – ebenfalls mit einer Gleichung (eqFname) ähnlich der vorigen
`{substr(UL,strpos(UL,'filename')+11,strpos(UL,'ext') - strpos(UL,'filename') -14)}`
Diese etwas umständliche Vorgehensweise ist der Tatsache geschuldet, dass ich nicht sicher bin, dass dieser kryptische file-name immer dieselbe Länge hat.
Falls ja, verkürzt sich die Gleichung zu
`{substr(UL,strpos(UL,'filename')+11,18)}`

Beispielhaft angezeigt:

fu_uujctwi746dgw7q

jpg

Previous

Next

Jetzt ist schon klar, wie der „ajax-call“ aussieht. Das script soll heißen „renameFile.php“

```
<script type="text/javascript" charset="utf-8">
$(document).on('ready ajax:scriptcomplete',function(){
$.post('https://www.myServer/renameFile.php',{sid: "{SID}", fname: "{eqFname}", ext: "{eqExt}" },function(data) {
    $('#question{QID} input[type="text"]').val(data)
});
});
</script>
```


Das script haben wir auch oben schon gesehen:

```
<?php
//Die übergebenen Variablen werden in lokale Variable abgespeichert
$ssid = $_POST["sid"];
$sfname = $_POST["fname"];
$sExt = $_POST["ext"];

// Die Datei wird lokalisiert.
// In diesem Beispiel liegt die Datei „renameFile.php“ in einem Verzeichnis,
// welches auf derselben Ebene, also parallel zur LimeSurvey-Installation liegt
$sfullname = '../limesurvey/upload/surveys/'.$ssid.'/files/'.$sfname;
$sNewName=$sfullname.'.'.$sExt;

// Es wird umbenannt
$x=rename($sfullname,$sNewName);
// Der file-name mit angehängter Endung wird zurückgegeben.
echo $sfname.'.'.$sExt;
?>
```

Die Frage vom Typ „kurzer Text“ soll einmal „Q2“ heißen.
Sie zeigt dann folgendes an:

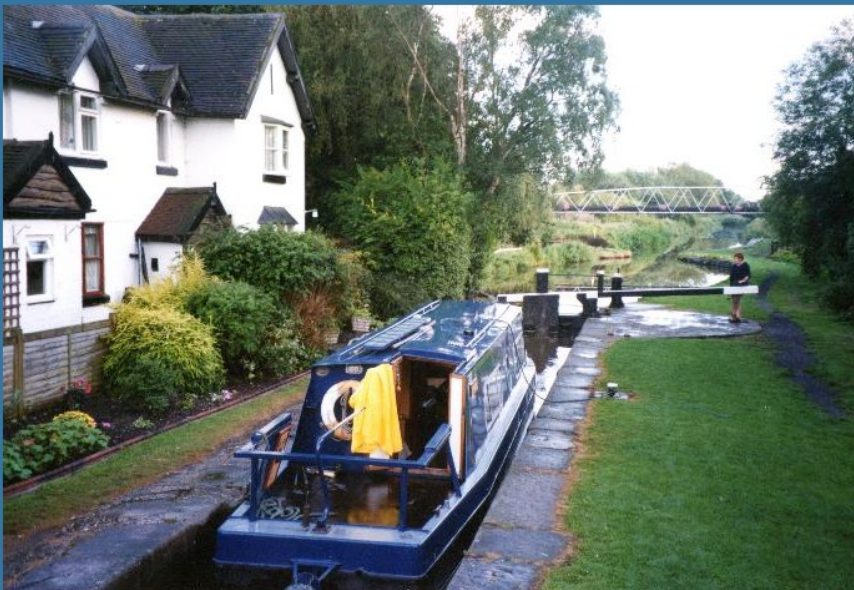
fu_uujctwi746dgw7q.jpg

[Previous](#)

[Next](#)

In einer richtigen Umfrage wird sie – wie auch die anderen Gleichungsfragen – versteckt. Mit der css-Klasse „hidden“.

Der Aufruf des Bildes wäre nun ``



[Previous](#)

[Next](#)

Man kann sogar auch den Titel des Bildes oder auch den Kommentar anzeigen.

Analog zur Extraktion des file-name.

Hier muss man natürlich wirklich die Länge des Titels bzw. die Länge des Kommentars errechnen – indem man eben guckt wo der nächste Teil beginnt, z.B. mit strpos(UL,“:‘comment“)

Dieses alles ist nicht in der lss-Datei enthalten.

Dies wäre unsinnig, da ein php script auf dem eigenen Server installiert werden muss, in welchem auch die relative Lage zur LimeSurvey-Installation beachtet werden muss.

17. Anzeige einer Karte aufgrund der IP-Adresse

17.1. Anwendungsbeispiel

Hier kann man sich verschiedene Anwendungen vorstellen.

Eine erste wäre, den Teilnehmer zu bitten, seinen Wohnort auf einer Karte zu markieren.

Das ist zunächst ja kein Problem.

Man nimmt eine Frage vom Typ „kurzer Text“ und wählt als Kartendienst „OpenStreetMap“ oder „Google Maps“ (sofern man einen Schlüssel bei Google beantragt hat) unter dem Menue „Lokation“ aus.

Dann erhält man als Mittelpunkt der Karte „Hamburg“.

Das ist nun leider nicht so optimal., wenn der Teilnehmer in Miesbach wohnt.

Also wählt man als Mittelpunkt irgendwelche Koordinaten der Mitte Deutschlands oder macht die „Karten-Zoom-Stufe“ so klein, dass ganz Deutschland zu sehen ist.

Besser wäre es sicherlich, dem Teilnehmer schon anfangs einen Kartenausschnitt zu präsentieren, der ungefähr seiner Region entspricht.

Eine zweite Anwendung wäre, eine Angabe des Teilnehmers zu überprüfen. Zum Beispiel bei einer Frage nach dem Land, in welcher sich der Teilnehmer gerade befindet.

So könnte man testen, ob die Angabe „Uruguay“ oder „West Samoa“ wirklich zutreffen kann. Dann würde man keine Karte anzeigen, sondern nur im Hintergrund die Werte speichern und später entsprechend auswerten.

17.2. Implementierung

Hier muss man zunächst einen Dienst aufrufen, der diese Arbeit übernimmt.

Es gibt mehrere, ich benutze hier <https://www.ipapi.co>

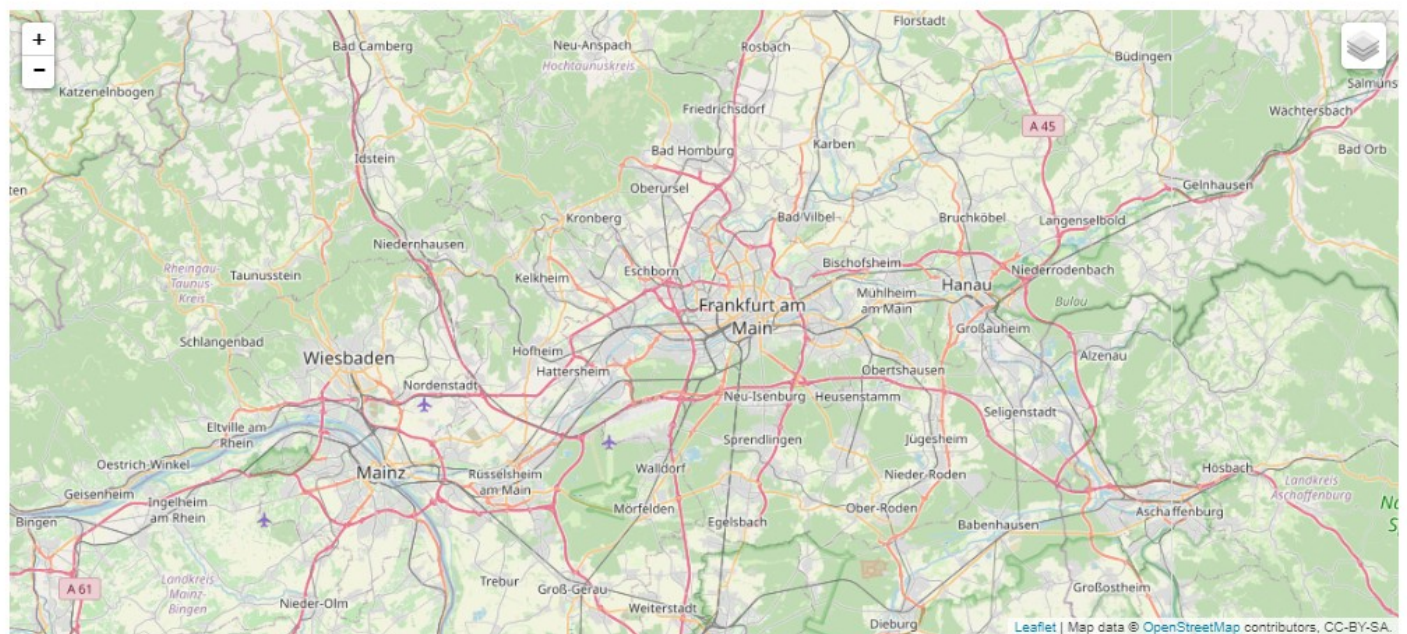
Da von diesem Dienst sehr viele Werte zurückgeliefert werden, benutzt man am besten eine Frage vom Typ „mehrfache kurze Text“

Hier einmal, was dieser Dienst bei dieser IP-Adresse anzeigt

IP Address	IP Lookup
IP Address	2001:9e8:28e3:aa00:b925:26d0:15e4:3cc8
City	Frankfurt am Main <small>Update</small>
Region	Hesse
Country	 DE Germany
Postal Code	60326
European Union	true
Latitude / Longitude	50.1049 , 8.6295
Time Zone	Europe/Berlin (+0200)
Calling Code	+49
Currency	EUR
Languages	de

Und daher zeigt die Karte dies. Und das ist nicht schlecht, nur 30km daneben.

Bitte, markieren Sie Ihren Wohnort!



Klicken Sie, um die Position einzustellen oder ziehen Sie den Stift. Sie können auch Koordinaten eingeben

Zum script selbst.

Zunächst ist klar, dass das script nicht auf derselben Seite wie die Karte sein darf.

Also am besten in eine vorher abgefragte Fragengruppe.

Dieser css-Teil versteckt die Frage.

```
<style type="text/css"> /* Hides the question from view */
#question{QID} {
    display:none;
}
</style>
```

Und das javascript ruft den Dienst auf und schreibt das Ergebnis in die einzelnen Teilfragen.
Natürlich kann man sich auf „Latitude“ und „Longitude“ beschränken.

```
<script>
$(document).ready(function() {
    $.get("https://ipapi.co/json/", function(data){
        $('#question{QID} input[type="text"]:eq(0)').val(data.city);
        $('#question{QID} input[type="text"]:eq(1)').val(data.region);
        $('#question{QID} input[type="text"]:eq(2)').val(data.postal);
        $('#question{QID} input[type="text"]:eq(3)').val(data.country_calling_code);
        $('#question{QID} input[type="text"]:eq(4)').val(data.country_code);
        $('#question{QID} input[type="text"]:eq(5)').val(data.country_name);
        $('#question{QID} input[type="text"]:eq(6)').val(data.latitude);
        $('#question{QID} input[type="text"]:eq(7)').val(data.longitude);
        $('#question{QID} input[type="text"]:eq(8)').val(data.currency);
        $('#question{QID} input[type="text"]:eq(9)').val(data.currency_name);
    });
});
</script>
```

Dann genügt es, in der Frage, die die Karte selbst anzeigt, in „Lokation“ dies als „Startposition“ eingetragen wird.

Zum Beispiel:

{Q17a_SQ007} {Q17a_SQ008} (mit Leerzeichen)

In LimeSurvey gibt es diese Einstellungsmöglichkeit

IP als Startpunkt ⓘ

Ja

Tja, es wäre schön, **wenn es funktionieren würde**. Dann hätte man sich dieses Kapitel schenken können.

In der Beispiel-Studie ist noch ein etwas anderes Herangehen gezeigt.

Hier ist die Frage zur Bestimmung der Koordinaten in einer separaten Gruppe, aber als einzige Frage.

Daher werden hier zunächst die Frage und auch die „Weiter“- und „Zurück“-Buttons versteckt.

Und da es einen kleinen Moment dauert, bis der Dienst aufgerufen ist und die Ergebnisse geliefert sind, ist am Ende mithilfe der „setTimeout“-Funktion eine halbe Sekunde Pause eingeschoben (500 Millisekunden), bis dann der Druck auf „Weiter“ simuliert wird.

```
<style type="text/css"> /* Hides the question and buttons from view */
#question{QID}, #ls-button-submit, #ls-button-previous {
    display:none;
}
</style>
```

```
<script>
$(document).ready(function() {
    $.get("https://ipapi.co/json/", function(data){
        $('#question{QID} input[type="text"]:eq(0)').val(data.city);
        $('#question{QID} input[type="text"]:eq(1)').val(data.region);
        $('#question{QID} input[type="text"]:eq(2)').val(data.postal);
        $('#question{QID} input[type="text"]:eq(3)').val(data.country_calling_code);
        $('#question{QID} input[type="text"]:eq(4)').val(data.country_code);
        $('#question{QID} input[type="text"]:eq(5)').val(data.country_name);
        $('#question{QID} input[type="text"]:eq(6)').val(data.latitude);
        $('#question{QID} input[type="text"]:eq(7)').val(data.longitude);
        $('#question{QID} input[type="text"]:eq(8)').val(data.currency);
        $('#question{QID} input[type="text"]:eq(9)').val(data.currency_name);
    });
    setTimeout(function () {
        $('#ls-button-submit').trigger('click');
    }, 500);
});
</script>
```

18. Anhang

18.1. Iss-Export einer Beispiel-Datei

Hier ist der Link zu einer gezippten Iss-Datei, die die hier beschriebenen Beispiele enthält.

https://www.mafosurvey.de/lime/Tutorial_Gimmicks.zip

18.1.1. zip-Datei der benötigten „highslide“-Dateien

https://www.mafosurvey.de/lime/highslide_Bibliotheken.zip

18.2. Iss-Export einiger Beispiele zur Nutzung von „getStatInSurvey“

https://www.mafosurvey.de/lime/Gimmicks_getStatInSurvey.zip

18.3. Iss-Export einiger Beispiele zur Nutzung von „statCount“ (nur ab Version 5.x)

https://www.mafosurvey.de/lime/Gimmicks_statCount.zip