**IWM*Study* 2.0**

**Benutzerhandbuch**

Version 2.055 vom 02.03.2021, Autor: André Klemke

Inhalt

[1 Versuchsdurchführung 1](#_Toc65568633)

[2 Versuchsauswertung 2](#_Toc65568634)

[2.1 Lokal 2](#_Toc65568635)

[2.2 Server 2](#_Toc65568636)

[3 Versuchserstellung 3](#_Toc65568637)

[3.1 Grundsätzliche Organisation der Dateien 3](#_Toc65568638)

[3.2 Konfiguration 3](#_Toc65568639)

[3.2.1 Globale Konfiguration der Versuchsumgebung „globalconfig.js“ 3](#_Toc65568640)

[3.2.2 Konfiguration der Studie „studyconfig.js“ 3](#_Toc65568641)

[3.2.3 Konfiguration von Bedingungen 5](#_Toc65568642)

[3.3 Erstellung von Content 7](#_Toc65568643)

[3.3.1 Grundlagen 7](#_Toc65568644)

[3.3.2 Codebeispiele 7](#_Toc65568645)

[3.3.3 JavaScript API 11](#_Toc65568646)

[3.4 Erstellung von Layouts 11](#_Toc65568647)

# Versuchsdurchführung

Zur Versuchsdurchführung wird die Datei index.html im Hauptverzeichnis der Studie aufgerufen (Browserempfehlung Chrome) und die Option „Studie durchführen“ gewählt. Die Studie kann entweder auf einem Webserver bereitgestellt werden (z.B. https://versuche.iwm-tuebingen.de/demo01) oder sich lokal auf dem Versuchsrechner befinden. Lokal kann die index.html direkt aus dem Dateisystem aufgerufen werden, ein lokaler Server wird nicht benötigt. Je nach Konfiguration der Studie wird eine Seite mit verschiedenen Einstellungen angezeigt. Alternativ ist auch ein Aufruf ohne diese Seite mit URL-Parameterübergabe möglich. Neben dem Studiennamen (Pflichtfeld) wird eine Bedingung gewählt (keine Eingabe => Zufall). Optional können initiale Daten sowie eine vom normalen Ablauf abweichende Startseite eingegeben werden. Unter Logging-Modus wird entschieden, ob die Datenspeicherung lokal (indexdDB im Browser, Achtung niemals Browserdaten löschen) oder auf einem Server (z.B. Server-URL https://versuche.iwm-tuebingen.de) durchgeführt wird. „Studie durchführen“ wechselt in die Benutzereinstellungen.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Je nach Konfiguration wird ein automatischer Benutzer generiert oder ein Account kann angelegt werden. Felder sind dann Organisations-ID/Gruppe (optional), Benutzername (Pflicht), Passwort (optional, Standardwert experiment) und E-Mail (optional), der Einleitungstext kann angepasst werden. Entsprechend der Konfiguration ist auch das Einloggen mit einem bestehenden Account auf der rechten Seite möglich. Nach „anlegen“ oder „einloggen“ erscheint eine weitere Seite, über die die Studie in einem neuen Tab geöffnet werden kann.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

# Versuchsauswertung

## Lokal

Zur lokalen Versuchsauswertung wird die Datei index.html im Hauptverzeichnis der Studie aufgerufen und die Option „Studie verwalten (Gerät)“ gewählt. Unter Studie wird die Studien-ID erfasst und dann eines der verschiedenen Dateiformate generiert. Über „speichern“ kann die entsprechende csv-Datei abgelegt, über „anzeigen“ eine Vorschau durchgeführt werden.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

## Server

Zur serverbasierten Versuchsauswertung wird die Datei index.html im Hauptverzeichnis der Studie aufgerufen und die Option „Studie verwalten (Server)“ gewählt. Im folgenden Dialog sind die Server-URL und Studie einzutragen. Auch Benutzername und Passwort des Versuchsleiters (durch die Medienentwicklung vergeben) sind anzugeben. Nach dem Login werden, abhängig von den eigenen Rechten (masteradmin, studyadmin, groupadmin), verschiedene Registerkarten auswählbar. Über „Datenabruf“ können die Logdaten einer Studie identisch zur lokalen Auswertung abgerufen werden (masteradmin, studyadmin). „Probanden verwalten“ zeigt eine Übersicht der Probanden der Studie mit weiteren Details und der Möglichkeit das Passwort zurückzusetzen (masteradmin, studyadmin, groupadmin). „Benutzer verwalten“ ist nur für masteradmin verfügbar und ermöglicht das Anlegen und Ändern von Benutzern für das serverbasierte Studienmanagement.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

# Versuchserstellung

## Grundsätzliche Organisation der Dateien

Bei der Erstellung einer neuen Studie ist vorgesehen, nur Änderungen im Verzeichnis *studies* vorzunehmen. Die anderen Dateien und Verzeichnisse werden versuchsübergreifend genutzt und bei Bedarf zentral weiterentwickelt. Im Verzeichnis *studies* der Versuchsumgebung befindet sich neben einer grundsätzlichen Konfigurationsdatei „globalconfig.js“ für jedes Experiment ein Unterordner bezeichnet durch die ID der Studie. Dieses Verzeichnis enthält wiederum mehrere Unterverzeichnisse:

*config* Enthält für jede Bedingung zur Konfiguration eine js-Datei, benannt nach der ID der Bedingung. Zusätzlich befindet sich hier die Datei studyconfig.js, über die die Studie insgesamt konfiguriert wird. In diesem Verzeichnis werden auch die verschiedenen eigenen Layouts als css abgelegt.

*content* Hier wird der Content abgelegt. Es handelt sich hierbei z.B. um HTML-Dateien, Bilder oder Videos. Auch zum Content gehörendes CSS oder JavaScript wird hier abgelegt.

*draft* Hier liegt das Quellmaterial, das zur Erzeugung der Studie verwendet wurde. Dazu gehören z. B. Powerpoint- und Word-Dokumente, Excel-Ablauflisten oder Bild- und Videodateien

*studies*

|- *studienid*

|- *config*

|- *content*

|- *draft*

|- *globalconfig.js*

## Konfiguration

Eine Konfiguration verschiedener Eigenschaften wird für die Versuchsumgebung, die einzelnen Studien und die einzelnen Bedingungen durchgeführt.

### Globale Konfiguration der Versuchsumgebung „globalconfig.js“

Die Versuchsumgebung verfügt über eine zentrale Konfigurationsdatei globalconfig.js, die der Konfiguration der Seite „Einstellungen“ dient.

var IWM = IWM || {};

IWM.globalconfig =

{

"defaultstudyid": "demo01",

"defaultloggingmode": "local", // local, server

"defaultserverurl": "http://localhost",

"defaultconditionid": "detailed",

"setupviewvisible": true,

"appversion": 2.031,

"headerimg": "app/styles/header\_iwm01.jpg", // header\_iwm01.jpg, header\_lightgrey.jpg

"logoimg": "app/styles/logo\_iwm.png", // logo\_iwm.png, logo\_empty.png

"logotext": "Study"

}

### Konfiguration der Studie „studyconfig.js“

Jede Studie hat eine zentrale Konfigurationsdatei studyconfig.js. Sie dient z.B. dazu, die Willkommensseite zu konfigurieren und den Login-Modus (anonym, register) festzulegen sowie die verschiedenen Bedingungen zu registrieren. Eine beispielhafte studyconfig.js wird nachfolgend gezeigt. Es sollen ergänzend einige wichtige Details beschrieben werden.

Unter „version“ wird die aktuelle Versionsnummer des Materials im Format 0.00 angegeben und bei jeder Änderung erhöht. Dieser Wert wird später in den Logdaten mit ausgegeben.

Für jede Bedingung wird ein Block unter conditions angelegt, der die ID der Bedingung sowie den aktuellen Status und die benötigte Nutzerzahl festlegt. Soll zur Laufzeit der Studie eine Bedingung deaktiviert werden, kann „active“ dazu auf „false“ gesetzt werden. Wenn alle Bedingungen „desiredrunnumber“ erreicht haben, können dennoch weitere VPs teilnehmen und werden zufällig verteilt, eine gleichmäßige Vergabe wird angestrebt.

Über „welcomepage“ wird die Begrüßungs- und Registrierungsseite konfiguriert. Zurzeit werden zwei Login-Modi unterstützt. Der Modus „anonym“ zeigt dem Nutzer nur einen Begrüßungstext und es sind keine Eingaben erforderlich, da automatisch ein zufälliger Benutzername erzeugt wird. Der Modus „register“ dagegen verlangt eine Registrierung mit Benutzernamen, optional wird auch eine Gruppe, ein Passwort und die e-Mail Adresse erfasst. Sollte der Proband bereits Logindaten besitzen, kann er sich mit seinen bestehenden Daten einloggen. Die Begrüßungsseite unterstützt mehrere vom Nutzer wechselbare Sprachen (z. Zt. nicht verfügbar), eine Standardsprache ist definiert. Für jede gewünschte Sprache wird ein „welcometext“ mit der ID der Sprache angelegt. Darin werden die Texte für alle Text- und Steuerelemente definiert.

var IWM = IWM || {};

IWM.studyconfig =

{

"title": "demo01",

"version": 0.01,

"min\_appversion": 2.045,

"starttime": null,

"endtime": null,

"conditions":

[

{

"conditionid": "detailed",

"active": true,

"neededuser": 1,

"desiredrunnumber": 15

},

{

"conditionid": "overview",

"active": true,

"neededuser": 1,

"desiredrunnumber": 15

},

{

"conditionid": "examples",

"active": true,

"neededuser": 1,

"desiredrunnumber": 15

}

],

"welcomepage": {

"loginmode": "register", // register or anonym

"register\_groupvisible": true,

"std\_group": "IWM",

"register\_emailvisible": false,

"register\_passwordvisible": false,

"login\_visible": true,

"login\_preset": false,

"std\_language": "DE",

"welcometext":

{

"DE":

{

"registerheadline": "Herzlich willkommen",

"registerheadline\_prestart": "Herzlich willkommen",

"registerheadline\_postend": "Herzlich willkommen",

"registertext": "Danke für das Interesse an unserer Studie. Um ...",

"registertext\_prestart": "Danke für das Interesse an unserer Studie. Die Studie ...",

"registertext\_postend": "Danke für das Interesse an unserer Studie. Leider ...",

"registerformheadline": "Neuen Account anlegen:",

"registerformgrouplabel": "Organisations-ID:",

"registerformuserlabel": "Benutzername:",

"registerformpasswordlabel": "Passwort:",

"registerformpassword2label": "Passwort wiederholen:",

"registerformmaillabel": "E-Mail:",

"registerformbutton": "anlegen",

"loginformheadline": "Mit bestehendem Account einloggen:",

"loginformuserlabel": "Benutzername:",

"loginformpasswordlabel": "Passwort:",

"loginformbutton": "einloggen",

"anonymheadline": "Herzlich willkommen",

"anonymheadline\_prestart": "Herzlich willkommen",

"anonymheadline\_postend": "Herzlich willkommen",

"anonymtext": "Danke für das Interesse an unserer Studie. Um an ...",

"anonymtext\_prestart": "Danke für das Interesse an unserer Studie. Die Studie ...",

"anonymtext\_postend": "Danke für das Interesse an unserer Studie. Leider ...",

"anonymbutton": "weiter",

"studyrunningheadline": " ",

"studyrunningtext": "Die Studie läuft in einem neuen Tab/Fenster, bitte ...",

"opencontentwindowbuttontext": "Tab/Fenster öffnen"

}

}

}

}

### Konfiguration von Bedingungen

Die in der studyconfig.js registrierten Bedingungen werden jeweils über eine eigene Konfigurationsdatei im Verzeichnis *config* konfiguriert. Jede Bedingung definiert eine Reihe von Seiten („pages“) mit verschiedenen optionalen Eigenschaften. Die hier definierten Seiten setzen das Material aus dem Verzeichnis *content* sowie die Layouts aus dem Verzeichnis *config* ein. Material kann somit einfach in mehreren Bedingungen wiederverwendet werden. Neben der Definition der Seiten dient die Konfigurationsdatei z.B. auch zur Festlegung des Ablaufs (sowohl linear als auch vernetzt), zur zeitlichen und mengenmäßigen Begrenzung von Seitenaufrufen (auch in „pagegroups“) sowie zur Festlegung von alternativen Seitenabläufen. Über die Eigenschaft „progressmaxpagecount“ kann optional eine von der Anzahl der „pages“ abweichende Maximalzahl an Seiten für die Fortschrittsanzeige definiert werden. Dies wird notwendig, wenn ein Teil der Seiten alternative Ablaufpfade darstellt.

var IWM = IWM || {};

IWM.conditionconfig =

{

"conditionid": "example",

"title": "example",

"description": "Eine demo01-Bedingung.",

"language": "DE",

"progressmaxpagecount": null,

"pages":

[

{ "pageid": "start", "src": "start.html", "nextid": "example01", "first": true, "layout": "iwmgrey\_flex" },

{ "pageid": "example01", "src": "material\_fragen\_radio.html", "nextid": "ende", "layout": "iwmgrey\_flex" },

{ "pageid": "ende", "src": "ende.html", "nextallowed": false, "layout": "iwmgrey\_flex" }

],

"pagegroups": []

}

Jedes Element in „pages“ definiert eine Seite. Es besitzt mehrere, teils optionale, Attribute:

| **Attribut** | **opt.** | **def.** | **Beschreibung** |
| --- | --- | --- | --- |
| backallowed | x | true | false deaktiviert den zurück-Button. |
| backid | x | s. B. | Definiert die Seite, die beim Drücken des zurück-Buttons oder Ausführen der entsprechenden Funktion erscheinen soll. Wird keine backid angegeben, reagiert der zurück-Button wie eine history-Funktion. |
| backlabel | x | s. B. | Hier kann eine individuelle Beschriftung des „zurück“-Buttons angegeben werden. Default ist „zurück“. |
| controlsvisible | x | true | false entfernt weiter/zurück/Hinweisfelder |
| dataseparation | x | false | Bei weiteren Besuchen dieser Seite werden die keys der Logdaten modifiziert und um -X (X = visit) ergänzt. |
| firstpage | x | false | Die erste Seite der Studie muss durch firstpage: true markiert werden. |
| advpage | x | false | Mit advpage: true wird eine Seite markiert, bei deren Besuch die Wahrscheinlichkeit sehr hoch ist, dass die VP die Studie vollständig durchführen wird. |
| lastpage | x | false | Markiert eine oder mehrere mögliche letzte Seiten der Studie. |
| abortpage | x | false | Markiert die Abbruchseiten. |
| initdelay | x |  | Erst wenn die als initdelay angegebene Zeit (in ms) abgelaufen ist, wird die HTML-Seite durch die Versuchsumgebung initialisiert. Hierdurch erhält die HTML-Seite Zeit zur eigenen Initialisierung. |
| keyboardnextkeycode | x | s. B. | Definiert die Taste (Keycode), die „weiter“ auslöst. Default ist keine Tastatursteuerung. |
| layout |  |  | Hier muss der Name eines css-Layouts angegeben werden. Der Name des Layouts muss einer Datei im Verzeichnis config (ohne Dateiendung) entsprechen. |
| loggingmedia | x | true | video- und audio-Elemente werden bei Interaktion geloggt |
| maxtime | x |  | Gibt die maximale Verweildauer auf der Seite in Sekunden an. Nach Ablauf der Zeit wird automatisch die nächste Seite aufgerufen. Bei mehreren Besuchen auf einer Seite werden die Besuchszeiten kumuliert. |
| maxtimelabel | x | s. B. | Definiert den Text für den Maximalzeit-Zähler. Default „Zeitlimit: “. |
| maxvisits | x |  | Diese Seite darf nur so oft besucht werden, wie durch den Wert angegeben und wird danach ungültig. |
| mintime | x |  | Gibt die minimale Verweildauer auf der Seite in Sekunden an. Navigation und Weiter-Button sind bis zum Ende dieser Zeit gesperrt. |
| mintimelabel | x | s. B. | Definiert den Text für den Mindestzeit-Zähler. Default „Mindestzeit: “. |
| nextallowed | x | true | false deaktiviert den weiter-Button. |
| nextlabel | x | s. B. | Hier kann eine individuelle Beschriftung des „weiter“-Buttons angegeben werden. Default ist „weiter“. |
| nextid |  |  | Definiert die Seite, die beim Drücken des weiter-Buttons als nächstes angezeigt werden soll. Es kann hier eine oder mehrere pageids angegeben werden. Werden mehrere IDs durch „|“ getrennt angegeben, werden deren Beschränkungen in dieser Reihenfolge nacheinander auf Gültigkeit geprüft und die erste gültige Seite verwendet. Werden mehrere IDs durch „\*“ getrennt angegeben, wird aus den gültigen IDs zufällig eine ID ausgewählt. „\*“ und „|“ können zusammen verwendet werden, „\*“ bindet stärker. Die letzte Seite im Ablauf benötigt keine nextid, hier ist nextallowed: false zu setzen. |
| onlydata | x |  | Diese Seite ist nur gültig, wenn die Eingabewerte von vorherigen Seiten den hier definierten Werten entsprechen. Das Format ist benutzer.pageid.elementid==Wert. Als Benutzer kann eine beliebige Zahl oder der Ausdruck „self“ verwendet werden. Möglich ist auch das prüfen von mehreren Eingabewerten, hierzu wird wie beschrieben vorgegangen und mehrere Vergleiche durch „&&“ getrennt. |
| onlyuser | x |  | Noch nicht implementiert. Diese Seite ist nur für den entsprechenden Benutzer im Versuchslauf gültig. Mögliche Werte sind eine beliebige Zahl. Lautet der Wert z.B. 2, ist diese Seite nur für den zweiten Benutzer in jedem Versuchslauf gültig. |
| pageid |  |  | Jede Seite besitzt eine frei wählbare eindeutige ID. Kurze verständliche IDs ohne Umlaute sind sinnvoll. Die IDs treten in den Logfiles als Spaltenname auf. |
| pagegroup | x |  | Durch Angabe einer pagegroup kann die Seite einer Gruppe zugeordnet werden. Auf diese Weise können zeitliche und mengenmäßige Beschränkungen der Verweildauer innerhalb einer Gruppe von Seiten realisiert werden. |
| param | x |  | Ein beliebiger Wert, der über die Funktion getPageinfo() abgerufen werden kann. |
| progress | x |  | Eine Dezimalzahl, die den aktuellen Fortschritt der Studie angibt, z.B. 0.40. Ohne Angabe wird der Fortschritt anhand der insgesamt vorhandenen Seiten berechnet oder über das Attribut progressmaxpagecount der conditionconfig. |
| reloaddata | x | true | Bestehende Daten von vorherigen Besuchen dieser Seite werden geladen und Formularfelder ausgefüllt. |
| src |  |  | Jede Seite definiert eine HTML-Datei als Quelle, die bestimmte Elemente aufweist. HTML-Seiten können mit verschiedener pageid aber gleicher src mehrfach verwendet werden, z.B. als Pretest und Posttest. |
| syncid | x |  | Besteht eine syncid, wird die Seite automatisch verlassen sobald alle Nutzer im Versuchslauf eine Seite mit dieser syncid erreicht haben. |

## Erstellung von Content

### Grundlagen

Material wird im HTML-Format erstellt und weißt einige Pflichtelemente auf. Es wird im Ordner *content* abgelegt, ebenso zugehörige Bilder. Ein guter Ansatz für die Erstellung ist es, eine ähnliche bereits bestehende Datei zu kopieren und zu bearbeiten. Im Folgenden werden die Grundelemente gezeigt, die in jeder HTML-Datei vorkommen sollten. Sie dienen dazu, die Versuchsumgebung in die Seite zu integrieren und damit den Funktionsumfang der Versuchsumgebung (Logging, Ablaufsteuerung, Funktionsaufrufe etc.) verfügbar zu machen, sowie die für das konfigurierbare Layout vorgesehenen Elemente (iwmstudy\_masterbox, iwmstudy\_contentbox,..) bereitzustellen.

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>

<body>

<script src="../../../app/iwm\_studyaddon.js"></script>

<div id="iwmstudy\_masterbox"><div id="iwmstudy\_centerbox">

<!-- Überschrift -->

<div id="iwmstudy\_headlinebox">Überschrift</div>

<!-- Inhalt -->

<div id="iwmstudy\_contentbox">

Blindtext

</div>

</div></div>

</body>

Um Benutzereingaben zu speichern, können verschiedene Eingabeelemente genutzt werden. Die Versuchsumgebung analysiert alle Elemente einer Seite und speichert die Eingaben beim Seitenwechsel, wenn das Element zu den unterstützten Typen gehört und als „id“ einen Wert aufweist, der mit „store\_“ beginnt. Alle Elemente, die so ausgezeichnet sind, sind vom Probanden obligatorisch auszufüllen, damit zur nächsten Seite weitergeschaltet werden kann. Handelt es sich um ein optionales Eingabefeld, ist stattdessen der Wert „storeopt\_“ am Anfang der „id“ anzugeben. Eine Besonderheit stellen Radiobuttons dar. Hier wird „store\_“ bzw. „storeopt\_“ nicht als „id“ sondern als „name“ verwendet. Anonymes Logging ist mit „storeanonym\_“ und „storeoptanonym\_“ möglich. Gültige Eingabeelemente sind z.Zt.

<input type=“text“/>

<input type=“hidden“/>

<input type=“number“/>

<input type=“range“/>

<input type=“checkbox“/>

<input type=“radio“/>

<textarea></textarea>

<select></select>

### Codebeispiele

Die folgenden Codebeispiele können im HTML-Material direkt verwendet werden, um verschiedene übliche Aufgaben zu erfüllen.

#### Text

<div class="iwmstudy\_text">

Blindtext einspaltig

</div>

#### Liste (unsortiert)

<ul class="iwmstudy\_list">

<li>Element 1</li>

<li>Element 2</li>

<li>Element 3</li>

</ul>

#### Liste (sortiert)

<ol class="iwmstudy\_list">

<li>Element 1</li>

<li>Element 2</li>

<li>Element 3</li>

</ol>

#### Vorformatierter Text wie z.B. Code (mit optionaler Blockunterschrift)

<figure class="iwmstudy\_code">

<!-- vorformatierter Bereich -->

<pre>

&lt;span&gt;Text&lt;/span&gt;

</pre>

<!-- Ende vorformatierter Bereich -->

<figcaption>Blindtext für eine Blockunterschrift</figcaption>

</figure>

#### Tabelle (mit optionaler Tabellenunterschrift)

<figure class="iwmstudy\_table">

<table>

<tr>

<th>Spaltenüberschrift 1</th>

<th>Spaltenüberschrift 2</th>

<th>Spaltenüberschrift 3</th>

<th>Spaltenüberschrift 4</th>

</tr>

<tr>

<td>Wert 1</td>

<td>Wert 2</td>

<td>Wert 3</td>

<td>Wert 4</td>

</tr>

</table>

<figcaption>Blindtext für eine Tabellenunterschrift</figcaption>

</figure>

#### Bild (mit optionaler Bildunterschrift)

<figure class="iwmstudy\_image">

<img src="medien/bspbild.jpg" style="height:25%;"/>

<figcaption>Blindtext für eine Bildunterschrift</figcaption>

</figure>

#### Video (mit optionaler Bildunterschrift)

<figure class="iwmstudy\_video">

<video src="medien/bspvideo.mp4" autoplay="autoplay" controls="controls" style="height: 40%;"></video>

<figcaption>Blindtext für eine Videounterschrift</figcaption>

</figure>

#### Audio

<div class="iwmstudy\_audio">

<audio id="audio" preload="auto" src="medien/bspaudio.wav"></audio>

<img src="medien/speaker.png" style="width: 3%; cursor: hand;" onclick="audio.play();"/>

</div>

#### Frage – Text einzeilig

<div class="iwmstudy\_question\_singlelinetext">

<span>Fragetext</span><br/>

<input id="store\_varname" type="text" style="width:70%;" />

</div>

#### Frage – Zahl

<div class="iwmstudy\_question\_number">

<span>Fragetext</span><br/>

<input id="store\_varname" type="number" style="width:10%;" />

</div>

#### Frage – Text mehrzeilig

<div class="iwmstudy\_question\_multilinetext">

<span>Fragetext</span><br/>

<textarea id="store\_varname"></textarea>

</div>

#### Frage – Einfachauswahl (Radiobutton)

<div class="iwmstudy\_question\_singlechoice">

<span>Fragetext</span><br/>

<label><input type="radio" name="store\_varname" value="wert1"/>Antwort 1</label><br/>

<label><input type="radio" name="store\_varname " value="wert2"/>Antwort 2</label><br/>

<label><input type="radio" name="store\_varname " value="wert3"/>Antwort 3</label>

</div>

#### Frage – Skala

<div class="iwmstudy\_question\_scale">

<span>Fragetext</span>

<div>

<span>Stimmt überhaupt nicht</span>

<div><div>

<input type="radio" name="store\_varname" value="1" />

<input type="radio" name="store\_varname" value="2" />

<input type="radio" name="store\_varname" value="3" />

<input type="radio" name="store\_varname" value="4" />

<input type="radio" name="store\_varname" value="5" />

<input type="radio" name="store\_varname" value="6" />

<input type="radio" name="store\_varname" value="7" />

</div><div>

<span>1</span>

<span>2</span>

<span>3</span>

<span>4</span>

<span>5</span>

<span>6</span>

<span>7</span>

</div></div>

<span>Stimmt völlig</span>

</div>

</div>

#### Frage – Mehrfachskala

<div class="iwmstudy\_question\_multiscale">

<span>Wie sehr stimmen Sie folgenden Aussagen zu?</span><br/>

<span>(0 = keine Zustimmung, 5 = volle Zustimmung)</span>

<table>

<tr>

<th> </th>

<th>0</th>

<th>1</th>

<th>2</th>

<th>3</th>

<th>4</th>

<th>5</th>

</tr>

<tr>

<td>Aussagetext 1</td>

<td><input type="radio" name="store\_varname1" value="0" /></td>

<td><input type="radio" name="store\_varname1" value="1" /></td>

<td><input type="radio" name="store\_varname1" value="2" /></td>

<td><input type="radio" name="store\_varname1" value="3" /></td>

<td><input type="radio" name="store\_varname1" value="4" /></td>

<td><input type="radio" name="store\_varname1" value="5" /></td>

</tr>

<tr>

<td>Aussagetext 2</td>

<td><input type="radio" name="store\_varname2" value="0" /></td>

<td><input type="radio" name="store\_varname2" value="1" /></td>

<td><input type="radio" name="store\_varname2" value="2" /></td>

<td><input type="radio" name="store\_varname2" value="3" /></td>

<td><input type="radio" name="store\_varname2" value="4" /></td>

<td><input type="radio" name="store\_varname2" value="5" /></td>

</tr>

<tr>

<td>Aussagetext 3</td>

<td><input type="radio" name="store\_varname3" value="0" /></td>

<td><input type="radio" name="store\_varname3" value="1" /></td>

<td><input type="radio" name="store\_varname3" value="2" /></td>

<td><input type="radio" name="store\_varname3" value="3" /></td>

<td><input type="radio" name="store\_varname3" value="4" /></td>

<td><input type="radio" name="store\_varname3" value="5" /></td>

</tr>

</table>

</div>

#### Frage – Checkbox einzeln

<div class="iwmstudy\_question\_accept">

<label><input type="checkbox" id="store\_varname1" value="1"/>Zu bestätigende Aussage (Pflicht)</label>

</div>

#### Frage – Mehrfachauswahl (Checkbox)

<div class="iwmstudy\_question\_multiplechoice">

<span>Fragetext</span><br/>

<label><input type="checkbox" id="storeopt\_varname1" value="1"/>Antwort 1</label><br/>

<label><input type="checkbox" id="storeopt\_varname2" value="1"/>Antwort 2</label><br/>

<label><input type="checkbox" id="storeopt\_varname3" value="1"/>Antwort 3</label><br/>

<label><input type="checkbox" id="storeopt\_varname4" value="1"/>Antwort 4</label>

</div>

#### Frage – Dropdown

<div class="iwmstudy\_question\_dropdown">

<span>Fragetext</span><br/>

<select id="store\_varname1">

<option value="" disabled="disabled" selected="selected">bitte auswählen</option>

<option value="wert1">Antwort 1</option>

<option value="wert2">Antwort 2</option>

<option value="wert3">Antwort 3</option>

<option value="wert4">Antwort 4</option>

</select>

</div>

#### Frage – Slider

<div class="iwmstudy\_question\_slider">

<span>Fragetext</span><br/>

<div>

<datalist id="ticks">

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

<option>5</option>

<option>6</option>

<option>7</option>

<option>8</option>

<option>9</option>

</datalist>

<input id="store\_varname" type="range" min="1" max="9" value="5" step="1" list="ticks"/><br/>

<span>sehr positiv</span>

<span>neutral</span>

<span>sehr negativ</span>

</div>

</div>

#### Frage – individuell

<div class="iwmstudy\_question\_general">

<span>Fragetext</span><br/>

…beliebige Elemente…

</div>

### JavaScript API

Die Versuchsumgebung stellt außerdem in JavaScript eine Reihe von Funktionen und Variablen zur Verfügung, die auf der Seite verwendet werden können. Sie können über „window.iwmstudy\_access“ aufgerufen werden.

**Variablen**

additionallogdata = {} Zusätzliche Logdaten können in dieses Objekt als Schlüssel/Wert abgelegt werden.

nextbutton\_targetpage = null Die nächste Seite im Ablauf kann manuell geändert werden.

util Instanz enthält verschiedene nützliche Utility-Funktionen (z.B. für Zufallswerte)

**Funktionen**

linkto(target=null) Löst einen Seitenwechsel zur angegebenen pageid aus.

linkback() Löst einen Seitenwechsel zur vorherigen Seite bzw. in der Konfiguration über backid angebenen pageid aus.

getPageinfo() Liefert eine Vielzahl an Informationen über die aktuelle Seite.

getPagegroupinfo() Liefert eine Vielzahl an Informationen über die aktuelle Gruppe.

getStudyinfo() Liefert eine Vielzahl an Informationen über die aktuelle Studie.

getRuntimeData() Liefert ein Objekt, in dem Daten zur Laufzeit gespeichert werden können und zwischen einem Seitenwechsel erhalten bleiben. Daten stehen nach einem erneuten Login weiter zur Verfügung.

getExistingLogData() Gibt ein Objekt mit bei einem vorherigen Besuch auf dieser Seite gespeicherten Daten zurück (store\_, storeopt\_, additionallogdata).

logAction(action, data = {}) Loggt sofort eine action (String) mit einem Objekt, das beliebige Schlüssel/Werte enthält

logAnonymousData(type, data = {}) Loggt sofort Daten (Objekt mit beliebigen Schlüsseln/Werten) mit dem Keyword type. Die Daten lassen keinen Rückschluss auf die Versuchsperson oder den Versuchslauf zu.

storeFile(filename, blob) Speichert ein Blob unter dem filename in der Datenbank.

addInitFunction(funct) Ruft funct nach abgeschlossener Initialisierung der Versuchsumgebung auf der Seite auf. Über funct kann dann eine eigene weitere Initialisierung des Materials erfolgen.

addFinalizeFunction(funct, async=false) Definiert eine Funktion, die vor dem Verlassen der Seite ausgeführt wird.

removeFinalizeFunctionAsyncWait() Beendet das Warten auf eine asynchrone FinalizeFunction.

isPageReady() Dient der Abfrage, ob eine Seite vollständig initialisiert ist.

isExistingLogDataReady() Dient der Abfrage, ob bestehende Logdaten dieser Seite vollständig initialisiert wurden.

enableNextButton() aktiviert den Weiter-Button

disableNextButton() deaktiviert den Weiter-Button

doRPC(target, rpcname, rpcparams = []) Führt beim User target eine registrierte Funktion aus.

registerFunctionForIncomingRPC(name, f) Registriert eine Funktion für RPC.

sendChatMessage(text) Sendet eine Chatnachricht an alle User im Versuchslauf.

setChatReceiveFunction(funct) Legt eine Funktion zur Verarbeitung von eingehenden Chatnachrichten fest. Die Funktion muss als Parameter Sender und Text übernehmen können.

## Erstellung von Layouts

Layouts werden als css-Dateien im Ordner *config* abgelegt und über die jeweilige Bedingungs-Konfiguration verschiedenen Seiten zugewiesen. Jede Seite muss ein gültiges Layout zugewiesen bekommen. Das in der Studie demo01 enthaltene Layout iwmgrey\_flex deckt bereits viele Nutzungsszenarien ab und bildet eine gute Grundlage für individuelle Layoutanpassungen. Folgende Elemente sollten in einem Layout immer berücksichtigt werden, um die zentralen Elemente und die Ablaufsteuerung sinnvoll darzustellen.

#iwmstudy\_masterbox Container zur Ausrichtung

#iwmstudy\_centerbox Hauptcontainer mit Hintergrund, der jeglichen Inhalt enthält.

#iwmstudy\_headlinebox Container für Überschrift

#iwmstudy\_contentbox Container für Inhalt

#iwmstudy\_controlsbox Hintergrund für Steuerelemente der Ablaufsteuerung

#iwmstudy\_progressbox leerer Fortschrittsbalken

#iwmstudy\_progressbar Füllung des Fortschrittsbalkens

#iwmstudy\_mintimebox Anzeige der noch nötigen Mindestverweildauer

#iwmstudy\_maxtimebox Anzeige der noch verfügbaren Zeit

#iwmstudy\_backbutton Zurück-Button

#iwmstudy\_backbutton:disabled Zurück-Button (disabled)

#iwmstudy\_nextbutton Weiter-Button

#iwmstudy\_nextbutton:disabled Weiter-Button (disabled)

Zusätzlich gibt es eine Reihe von css-Klassen, die einzelne Elemente formatieren, wie z. B. div.iwmstudy\_text, div.iwmstudy\_question, img.iwmstudy\_image, video.iwmstudy\_video, div.iwmstudy\_caption, table.iwmstudy\_table, ul.iwmstudy\_list, ol.iwmstudy\_list, pre.iwmstudy\_code, div.iwmstudy\_slider, div.iwmstudy\_scale, table.iwmstudy\_multiscale.