Projektarbeit Multimedia-Technologien

Erstellung eines E-Learnings zum Online-Tool WireWax

5.11.2015 bis 21.01.2016

Carolin Schneider (Matrikelnummer: 40127)

Inhaltsverzeichnis

Ziel des Projekts

Das Ziel dieser Projektarbeit war es zum einen ein Thema aus dem Multimedia-Bereich auf die Verwendung in der Technischen Dokumentation bzw. als E-Learning zu analysieren und gesammelte Thesen mit Beispielen zu belegen. Zum anderen soll das Projekt zu einer Sammlung zur Bewertung neuster technologischer Themen im Umfeld der Technischen Dokumentation beitragen.

Ziel dieses speziellen Projektes war es ein E-Learning für das Online-Tool von WireWax mit HTML. Nutzer, die das Tool noch nie verwendet haben, sollen nach diesem E-Learning in der Lage sein, dass Tool ohne größere Probleme zu verwenden.

Damit liegt der Fokus dieses Projektes eindeutig auf der Umsetzung eines E-Learnings und weniger auf den theoretischen Hintergründen und der Sammlung von Thesen.

Der zeitliche Umfang dieses Projektes beträgt 60 Arbeitsstunden.

Projektverlauf

Zu Beginn des Projektes wurde ein detaillierter Zeitplan ausgearbeitet (Zeitplan.xlsx). Dieser soll alle Tätigkeiten abbilden, die bis zum Ende des Projektes erledigt werden müssen. Den Anfang bildeten dabei die zu erstellenden Dokumente Projektplan, Abschlussbericht, Zeiterfassung und das eigentliche E-Learning.

Folgende Arbeitsschritte haben sich herauskristallisiert:

- Projektplanung
  - Zeitplan erstellen
  - Vorlage Zeiterfassung erstellen
- Recherche: Welche Tätigkeiten gibt es in WireWax?
  - Was kann man mit WireWax tun?
  - Was soll ins E-Learning übernommen werden?
  - Was führt zu weit und wird deshalb im E-Learning nicht behandelt?
- Drehbuch ausarbeiten
  - Wie viele Kapitel werden benötigt?
  - Wie viele Seiten werden pro Kapitel benötigt?
  - Welche Bilder und Texte werden auf der Seite verwendet?
  - Welche Bereiche werden wie markiert und mit welchen Zusatzinformationen versehen?
- Layout für E-Learning entwerfen
  - Wie soll das E-Learning aussehen: Seitenaufbau, Farbgebung, Links, Animationen, …
- Layout in CSS umsetzen
- Screenshots erstellen
- Texte für E-Learning schreiben
  - Formulierungsvorgaben
  - Texte für Beschreibungen und Zusatzinformationen formulieren
- Website programmieren
  - Grundgerüst für Website mit HTML und JavaScript erstellen (inkl. jQuery-Verwendung
- Inhalte aus erstellten Texten und Screenshots in Grundgerüst einpflegen
- Dokumentation fertigstellen
- Abschlussvortrag vorbereiten

Das ursprüngliche Ziel des Projektes war es, ein E-Learning für WireWax mit WireWax zu erstellen, also als interaktives Video. Daher wurde nach der zeitlichen Planung des Projektes mit dem Einlernen in Adobe Captivate begonnen. Mit diesem Tool sollten die Videos erstellt werden, um dann mit WireWax bestimmte, für den Lerner relevante, Bereiche mit Interaktivität und Zusatzinformationen zu versehen.

Schon während dieser Einlernphase hat sich herausgestellt, dass ein Videotutorial nicht das geeignete Medium für das WireWax-Tutorial ist. Dieses würde hauptsächlich aus Standbildern bestehen, die sowohl schlecht abzufilmen als auch für den Nutzer wenig motivierend sind.

In Absprache mit Herrn Schober wurde dann das Projektziel dahingehend geändert, dass es nun kein interaktives Video-Tutorial mehr erstellt werden soll, sondern ein E-Learning mit HTML und JavaScript für WireWax erstellt wird.

Anschließend wurde gemäß dem Projektplan recherchiert, welche Tätigkeiten man in WireWax überhaupt ausführen kann und welche davon für ein E-Learning für Anfänger geeignet sind und welche zu anspruchsvoll sind bzw. zu wenig Nutzen für den Lerner bieten. Aus dieser Recherche haben sich die drei Bereiche Grundlagen, Editor, Metrics und Einstellungen ergeben, die die drei Hauptkapitel des E-Learnings bilden. Abhängig von der Zugehörigkeit zu diesen Kapiteln wurden dann die eigentlichen Tätigkeiten bzw. Informationen zu WireWax abgebildet. Durch diese Zuordnung sind die Kapitel auch unterschiedlich lang, das Grundlagenkapitel umfasst beispielsweise nur fünf Seiten mit Lerninhalt, das Metrics-Kapitel hat 15 Seiten, da hier sehr viele Messwerte erklärt werden müssen.

Die Unterseiten wurden anschließend im Drehbuch abgebildet und den entsprechenden Screenshots zugeordnet. Die zur Erklärung erforderlichen Texte sowie die fehlenden Screenshots wurden ebenfalls erstellt und im Drehbuch erfasst.

Parallel dazu wurde das Layout des E-Learnings entworfen (vgl.: Kapitel Design) und in CSS umgesetzt.

Im Anschluss daran wurde die Website in HTML erstellt, dabei wurde zuerst ein Grundgerüst der Seite entworfen, in das später alle Inhalte eingepflegt werden konnten.

Damit war das E-Learning erstellt und es konnte die Abschlussdokumentation, die bisher nur aus Notizen bestand, ausformuliert und fertiggestellt werden. Außerdem wurde festgehalten, was wie im Abschlussvortrag präsentiert werden soll.

Aufbau E-Learning

Im E-Learning zu WireWax soll der Nutzer Schritt für Schritt an die Nutzung des Online-Tools auf Wirewax.com zur Erstellung interaktiver Videos herangeführt werden. Das E-Learning ist in vier Hauptkapitel gegliedert.

Startseite

Auf der Startseite wird der Nutzer begrüßt und mit den grundlegenden Informationen über WireWax versorgt sowie dem Ziel des E-Learnings konfrontiert. Er erhält die URL zur WireWax-Website und erfährt, wo er sich den Browser Google Chrome herunterladen kann, da der Editor nur damit funktioniert. Genauere Informationen finden sich in der Datei Drehbuch.xlsx.

Kapitel 1 – Grundlagen

Im Grundlagenkapitel lernt der Nutzer die einfachsten Grundlagen im Umgang mit WireWax. Dazu gehören das Einloggen und die Orientierung auf der persönlichen Startseite. Er erfährt außerdem, wie er das E-Learning am besten verwendet und wie er darin navigiert.

Kapitel 2 – Editor

Im zweiten Kapitel lernt der Nutzer, wie er mit dem Tag-Editor von WireWax umgeht. Er lernt Videos hochzuladen, in den Bearbeitungsmodus zu wechseln, Tags zu erstellen und zu bearbeiten.

Kapitel 3 – Metrics

Im dritten Kapitel lernt der Nutzer, was die einzelnen Angaben im Metrics-Bereich bedeuten.

Kapitel 4 – Einstellungen

Im vierten Kapitel lernt der Nutzer den Einstellungen-Bereich seines Nutzerkontos kennen.

Design

Das Layout des E-Learning soll ansprechend, motivierend und modern auf den Nutzer wirken. Außerdem soll der Lerner grundsätzliche Elemente von der WireWax-Homepage wiedererkennen. Daher werden ähnliche Farben und Schriften verwendet.

Seitenaufbau

Für das E-Learning werden ein Header, eine Navigation und ein Content-Bereich benötigt.

Im Header stehen grundsätzliche Informationen wie die Kapitelüberschrift oder das Willkommen auf der Startseite. Außerdem gibt es hier einen Link zurück zur Startseite. In der Navigation stehen Links zu den verschiedenen Kapiteln des E-Learnings. Im Content-Bereich stehen alle Texte und Bilder, die der Nutzer für das E-Learning benötigt.

Das gesamte E-Learning ist auf eine Bildschirmauflösung von 1280x720 Pixel optimiert, was zu folgender Größenverteilung führt:

Hierzu wurde zuerst die optische Gestaltung der Website in Illustrator ausgearbeitet. Die Website wird auf eine Bildschirmauflösung von 1366 x 768 Pixel optimiert, da diese laut entwickler.de seit mehreren Jahren die am häufigsten genutzte Display-Auflösung ist. Grundsätzlich soll sich das E-Learning optisch an die Website WireWax.com anlehnen, d.h. es werden die gleichen Farben und Schriften verwendet, sofern diese frei verfügbar und kein Firmeneigentum sind.

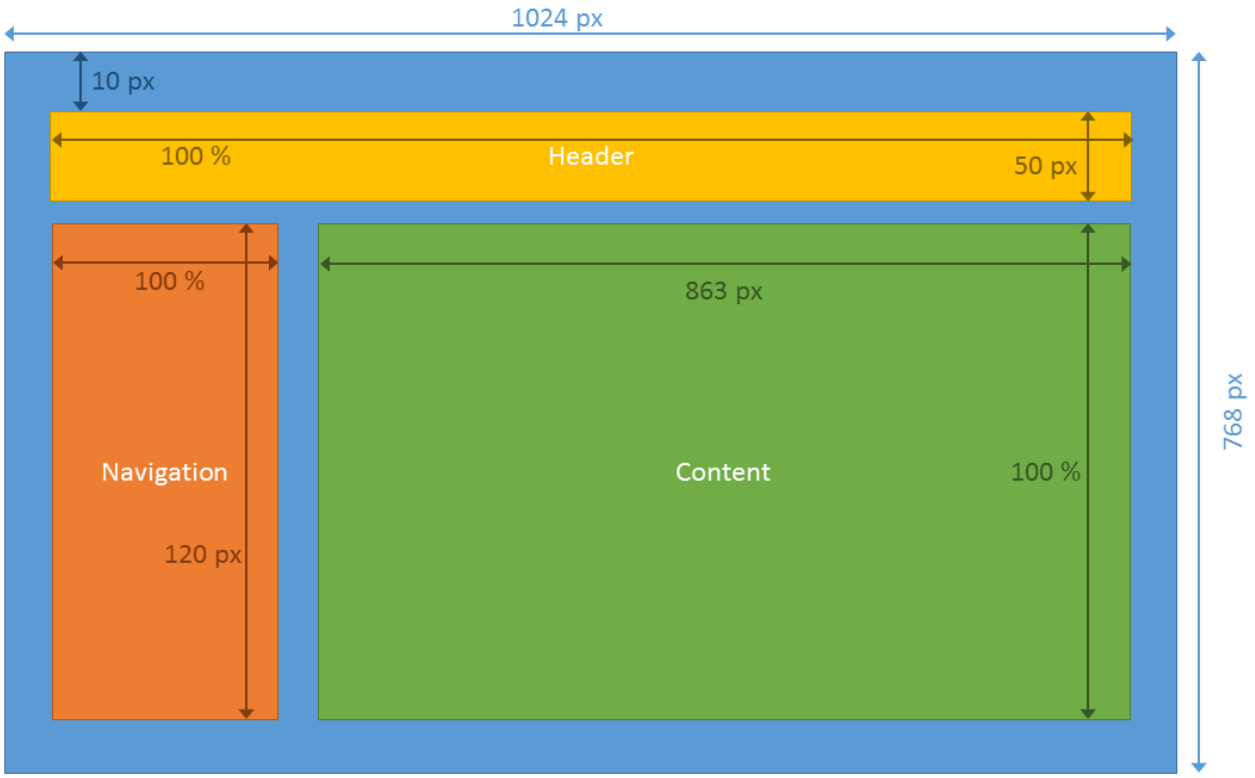

Abbildung 1: Seitenaufbau und Größen

Markierungen und Auszeichnungen

Auch hier soll der Nutzer die Elemente von der WireWax-Homepage wiedererkennen. Deshalb wird das gleiche Aussehen für Links und ähnliche Farben für Auszeichnungen im Text verwendet.

Element	Aussehen/Verhalten
Fließtext	• Font: Segoe UI (Serifenlose, klare Schrift) • Auf der WireWax-Homepage wird Helvetica verwendet, diese Schrift kostet aber leider Geld und wurde deshalb nicht verwendet. • Schriftfarbe: weiß
Überschriften	• Font-weight: normal
Links	• Schriftfarbe: #8BBCA7 • Nicht unterstrichen • Beim Überfahren ändert sich die Farbe nach weiß
Betonung im Text (für Schlüsselwörter)	• font-weight:normal • Schriftfarbe: #428BCA
Hintergrund Seite	• rgba(56,56,56,0.4)
Hintergrund Body	• #292C2E
Informationen (mouseover für Zusatzinformationen)	• Rahmen: gestrichelt, weiß, 1px breit • Cursor ändert sich beim Überfahren mit der Maus auf Hilfesymbol
Box für Informationen	• Hintergrundfarbe: #333 • Schriftgröße: 10pt • Zuerst unsichtbar, erscheint erst, wenn mit der Maus die Information überfahren wird
Aufforderung: Klicken	• Rahmen: durchgezogen, aqua, 2px breit • Cursor ändert sich beim Überfahren mit der Maus auf Klicksymbol • blinkend

Verwendete Farben

Alle im E-Learning verwendeten Farben (außer weiß) sind hier noch einmal mit einem Farbbeispiel aufgeführt.

Abbildung 2: Farbbeispiele

Formulierungen

Grundsätzlich ist der Anspruch, dass alle Anweisungen, Fließtexte usw. gleich formuliert sind. Der Leser wird im E-Learning geduzt, nicht gesiezt und erhält direkte Handlungsanforderungen im Imperativ.

Textart	**Formulierung**
Bedingte Anweisung	Wenn du A möchtest, dann klicke hier.
Informationen	Kurz und prägnant
Links zur Seitennavigation	Besteht nur aus einem Wort
Zusatzinformationen	Als bedingte Anweisung formuliert oder Einleitung mit „hier“ wenn etwas näher erläutert wird.

Entwurf der Seiten

Nachdem der grundsätzliche Aufbau und die zu verwendenden Farben geklärt waren, wurde mit Adobe Illustrator ein Entwurf der beiden Seitenvarianten Startseite und Kapitel erstellt. Diese Entwürfe sollen der leichteren Umsetzung des Designs in CSS dienen und die schnelle Kontrolle ermöglichen, ob das Aussehen im Browser dem gewünschten Aussehen entspricht.

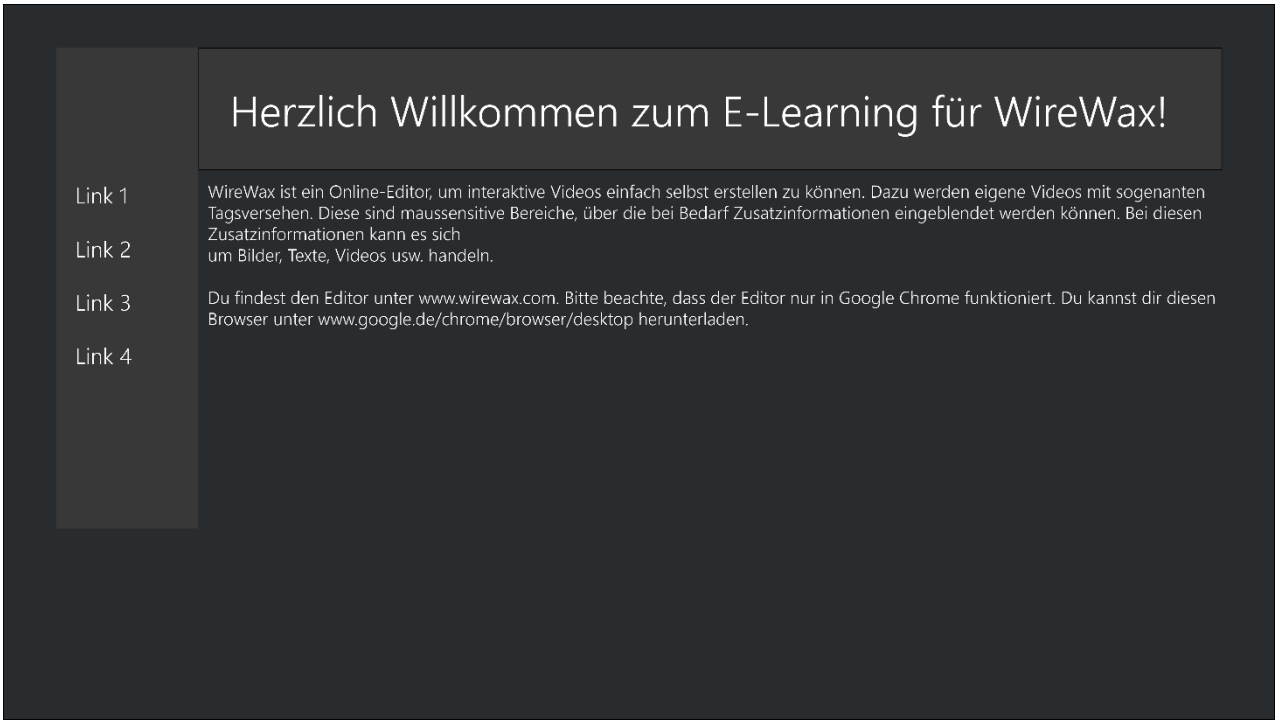

Abbildung 3: Entwurf Startseite

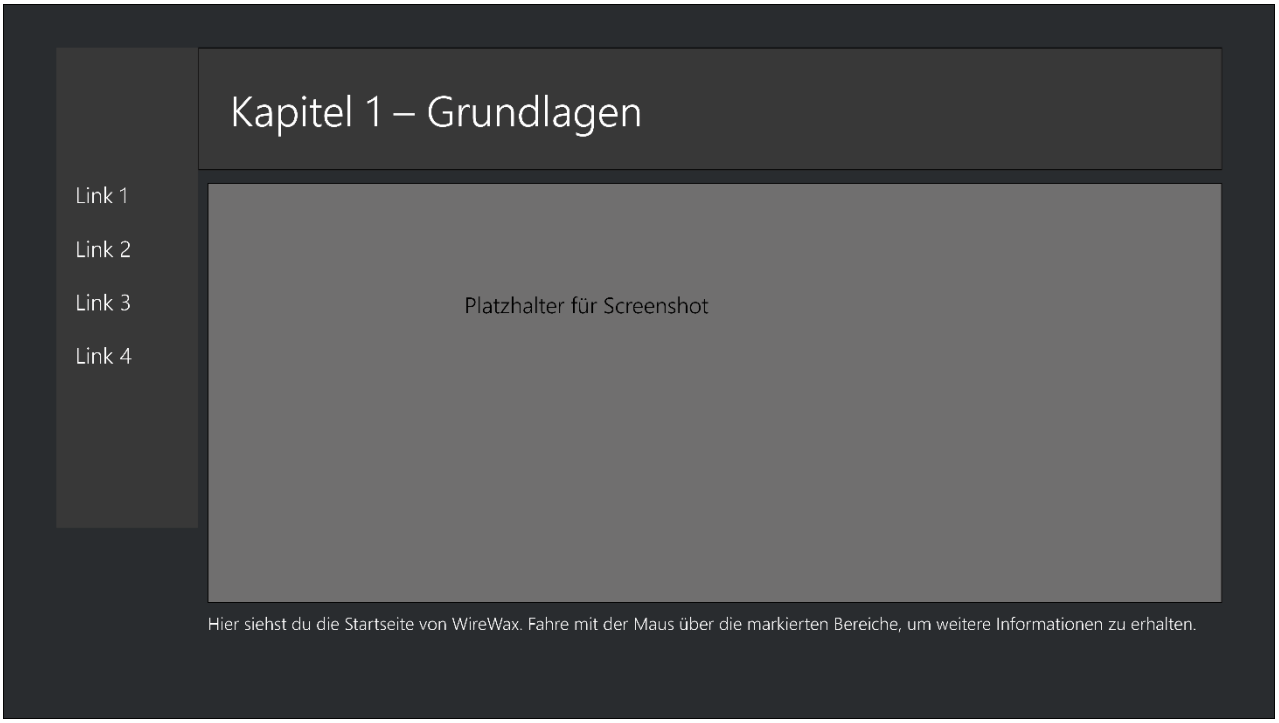

Abbildung 4: Entwurf Kapitel

Umsetzung in CSS

Die Umsetzung in CSS war durch den einfachen Seitenaufbau gut umzusetzen. Trotzdem gab es ein paar Kleinigkeiten zu beachten.

body{
font-family: Segoe UI;
background-color: #292C2E;
color: white;
}
	Für den Body-Bereich werden die Schriftart und die Schriftfarbe festgelegt, da diese dann automatisch für alle enthaltenen Elemente zählt und nicht einzeln festgelegt werden muss.

#wrapper{
width: 1024px;
margin: auto;
height: 768px;
}	

#navigation{
width: 120px;
height: 100%;
float: left;
min-height: 400px;
background-color: rgba(56,56,56,0.4) ;
border-right: 1px solid #363638;
padding: 10px;
}
	Für den Navigationsbereich werden die Breite und die Höhe angegeben. Die Höhe beträgt 100 %, damit sie immer die komplette Bildschirmhöhe ausfüllt, unabhängig von tatsächlichen Inhalt des Containers.
	Mit float: left wird dafür gesorgt, dass der Div-Container auf der linken Seite angeordnet ist.

#header{
height: 50px;
font-size: 24pt;
text-align: center;
background-color: rgba(56,56,56,0.4);
margin-top: 20px;
padding-top: 10px;
}	

#home{
position: relative;
top: -30px;
right: -450px;
font-size: 10pt;
}
	Der home-Button ist in einem eigenen Div-Container enthalten, damit man ihn besser auf der Seite anordnen kann. Das geschieht über die Abstandsangaben top und right und position: relativ. Position: relative wurde gewählt, damit sich der div-Container auf der Seite nicht verschiebt, wenn das Browserfenster in seiner Größe verändert wird.

#content{ width: 863px; height: 100%;	Der Div-Container, der den eigentlichen Content enthält, hat float:right, damit er sich immer auf der rechten Seite platziert. So können die

float: right;
min-height: 400px;
background-color: rgba(56,56,56,0.4);
padding: 10px;
}
	Navigationsleiste und der Content-Bereich nebeneinander, statt untereinander stehen.

h1{
font-weight: normal;
}
	Bei der Überschrift erster Ebene wird die Standardeinstellung von font-weight: bold mit font-weight: normal überschrieben.

a{
color: #8BBCA7;
text-decoration: none;
}
	Die Links werden anders gefärbt und die Unterstreichung wird weggenommen.

a:hover{
color: white;
}
	Beim Überfahren der Links mit der Maus werden sie weiß, anstatt eine Unterstreichung zu bekommen.

.marker{
border: dashed white 1px;
position: relative;
z-index: 2;
cursor: help;
}
	Mit marker wurde eine Klasse an alle Markierungen für alle Div-Container über relevanten Bereichen vergeben.
	Diese haben einen gestrichelten, weißen Rahmen, eine relative Positionierung und liegen über der Seitenoberfläche. Beim Überfahren mit der Maus ändert sich der Mauszeiger nach  .
	Die Größe und tatsächliche Position müssen noch im HTML-Code für jeden Div-Container einzeln eingegeben werden (style="…"), da alle Container unterschiedlich groß sind.

.info{
background-color: #333;
width: 150px;
font-size: 10pt;
position: relative;
z-index: 4;
display: none;
padding: 2px;
}
	Info ist ebenfalls eine neue Klasse. Sie ist für die Div-Container gedacht, die die Zusatzinformationen enthalten und ein- und ausgeblendet werden.
	Die Breite ist für alle festgelegt, die Höhe passt sich dem Inhalt an. Die Positionierung ist ebenfalls relativ und sie sind standardmäßig ausgeblendet.
	Im HTML-Code muss nur noch die tatsächliche Position mit top und left angegeben werden.

.aktion{ border: solid aqua 2px; z-index: 5;	Mit der Klasse aktion werden alle Div-Container ausgezeichnet, die den Nutzer auf eine von ihm erwartete Aktion im WireWax-Tool aufmerksam

`cursor: pointer;`
`position: relative;`
`}`
	machen sollen. Diese werden mit jQuery/JavaScript noch zum Blinken gebracht.
	Bei diesen Div-Containern ändert sich der Mauszeiger beim Überfahren mit der Maus nach 

`img{`
`width: 840px;`
`}`
	Bilder haben eine Standardbreite von 840px.

`b{`
`font-weight:normal;`
`color: #428BCA;`
`}`
	Die Auszeichnung b wird von font-weight: bold auf font-weight: normal und die Schriftfarbe mittelblau geändert.

Animationen

Für die Animationen im E-Learning wurde jQuery benutzt. So konnte viel Quellcode eingespart werden.

Im Prinzip gibt es in diesem E-Learning nur zwei verschiedene Animationsarten: Ein- und Ausblenden der Zusatzinformationen und blinkende Rahmen für die Aufforderung zu klicken. Diese Animationen wurden folgendermaßen umgesetzt.

Ein- und Ausblenden

Der Rahmen um das betreffende Symbol ist ein Div-Container. Die Zusatzinformationen befinden sich ebenfalls in einem Div-Container. Beide haben jeweils eine ID, deren Aufbau immer gleich ist: Die Rahmen werden durchnummeriert, z. B. 1, 2 usw. während die Zusatzinformationen die dazugehörige Nummer und den Zusatz _Info haben, also z. B. 1_Info, 2_Info usw.

Diese werden dann ein- und ausgeblendet:

`<script type="text/javascript">`	
`$(document).ready(function() {`	Wenn das Dokument geladen ist, dann führe folgende Funktionen aus.

`$("#1").mouseover( function() {`
`$("#1_Info").show();`
`});`
	Wenn der Div-Container mit der ID 1 mit der Maus überfahren wird, dann führe für den Div-Container mit der ID 1_Info die Funktion show() aus (Ändert die CSS-Eigenschaft display von none auf block).

`$("#1").mouseout( function() {`
`$("#1_Info").hide();`
`});`
	Wenn der Mauszeiger den Div-Container mit der ID 1 verlässt, dann führe für den Div-Container mit der ID 1_Info die Funktion hide() aus (Ändert die CSS-Eigenschaft display von block auf none).

`});`	
`</script>`	

Blinken

Die blinkenden Rahmen sind ebenfalls Div-Container, die über eine jQuery-Funktion angesteuert werden. Da diese aber immer wieder ausgeführt werden soll, solange die Seite angezeigt wird, muss eine sinnvolle Endlosschleife programmiert werden.

`<script type="text/javascript">`	
`$(document).ready(step());`	Wenn das Dokument fertig geladen ist, dann führe die Funktion step() aus.

`function step(zahl){`
`blink(zahl);`
`zahl++;`
`setTimeout("step('"+zahl+"')",1000);`
`}`
	Führe die Funktion blink() zahl-mal aus, zähle die Variable zahl um 1 hoch und mach Pause.

`function blink() {`
`$(".aktion").fadeOut(800, function(){});`
`$(".aktion").fadeIn(800, function(){});`
`};`	

`</script>`	

Bei allen anderen Varianten, wie einem rekursiven Aufruf (Funktion ruft sich selbst immer wieder auf) oder einer while-Schleife mit der Bedingung true, treten massive Probleme, wie ein immer voller werdender Speicher oder eine viel zu schnelle Ausführung des Blinkens auf.

Quellen

https://entwickler.de/online/webmagazin/die-top-10-monitorauflosungen-2014-317.html

[09.12.2015; 14:30]

http://wiki.selfhtml.org/

[letzter Zugriff am 14.01.2015; 15:30]

http://www.wirewax.com/home/create-own

[letzter Zugriff am 14.01.2015; 15:30]

http://unicode-table.com/de/

[letzter Zugriff am 14.01.2015; 15:30]

http://api.jquery.com/

[letzter Zugriff am 14.01.2015; 15:30]

http://www.mediaevent.de/tutorial/farbcodes.html

[letzter Zugriff am 14.01.2015; 15:30]