

Webtechnologien 1

Aufgabenblatt 3: JavaScript & jQuery

Das Ziel dieses Übungskomplexes ist das Erlernen der Verwendung von JavaScript mit Hilfe des JS-Frameworks jQuery und dessen Einsatz für die Gestaltung des Blogs.

Falls Ihre Lösung zu vorherigen Übungen unvollständig ist, können Sie die entsprechende Musterlösung verwenden, welche im Moodle als Material zum Download bereitsteht.

Aufgabe 1

Entwerfen Sie eine geeignete Datenstruktur (Objekt) zur Speicherung der Blogartikel. Wählen Sie für das Datum vorerst den Datentyp Zeichenkette. Legen Sie eine neue JavaScript-Datei `data.js` an. Vereinbaren Sie in dieser Datei eine Array-Variable `articles`:

```
var articles = [];
```

Überlegen Sie sich einige Blogartikel (oder nehmen Sie die aus Ihrer bestehenden Lösung) und legen Sie diese in der zuvor entworfenen Datenstruktur in dem Array ab. Binden Sie die Datei `data.js` in Ihre `artikel.html` ein.

Laden Sie die `artikel.html` im Browser und lassen Sie sich auf der Konsole (Teil der Entwicklertools des Browsers) den Inhalt der Variablen `articles` ausgeben:

```
> articles;
```

Aufgabe 2

Machen Sie sich mit dem JavaScript-Framework `jQuery` vertraut, laden Sie es herunter und binden Sie es in Ihre `artikel.html`-Datei ein. Sorgen Sie dafür, dass mit `alert` ein Meldungsfenster geöffnet wird, sobald die Seite vollständig geladen ist.

Hängen Sie in das `<header>`-Element der `artikel.html` ein beliebiges weiteres Element mit jQuery ein.

Beispiel (Das Markup Ihrer eigenen Lösung kann sich auch deutlich von diesem unterscheiden):

vorher

```
<header>  
  <h1>Das ist mein Blog</h1>  
</header>
```

nachher

```
<header>  
  <h1>Das ist mein Blog</h1>  
  <div>und das ist ein dynamisch eingehängtes Element</div>  
</header>
```

Aufgabe 3

Der bislang in der Datei `artikel.html` enthaltene Dummy-Artikel soll dynamisch aus dem Array `articles` geladen werden.

Lesen Sie aus dem Array mit den Blogartikeln (`articles`) den ersten Artikel und hängen Sie ihn mit jQuery in das DOM der Datei `articles.html` ein. Verwenden Sie dazu die HTML-Struktur eines Artikels, die Sie sich in der Übung zum Komplex HTML überlegt haben. Die Social-Media-Links können vorerst Dummies bleiben.

Beispiel (Das Markup Ihrer eigenen Lösung kann sich auch deutlich von diesem unterscheiden):

Vorher:

```
<main class="col-12 col-md-9">
</main>
```

Nachher:

```
<main class="col-12 col-md-9">
  <article>
    <h2>Überschrift meines ersten Artikels</h2>
    <p>14. März 2018 Uhr von Susi Sommer</p>
    <p><b>Das ist der Anrisstext zu meinem ersten Artikel</b></p>
    <p>Das ist der Artikeltext zu meinem ersten Blogartikel, der aus dem
      Array kommt. Könnte noch viel länger sein....</p>
    <div>
      <a class="badge badge-pill badge-primary"
        href="tagliste.html">Tag1</a>
      <span class="badge badge-pill badge-primary">Tag2</span>
    </div>
    <br>
    <div>
      <a title="Facebook" href="https://facebook.com/share.php?url=..."
        target="_blank">Teilen auf Facebook</a> |
      <a title="Twitter" href="https://twitter.com/intent/tweet?url=..."
        target="_blank">Twittern</a> |
      <a href="teilenEmail.html">Teilen via E-Mail</a>
    </div>
  </article>
</main>
```

Erstellen Sie eine JavaScript-Datei `blog.js` und binden Sie sie in die `artikel.html` ein. Diese Datei wird die JavaScript-Funktionen zur Steuerung des Blogs enthalten.

Implementieren Sie in der `blog.js` eine Funktion

```
var createArticle = function(article) {
  ...
};
```

die ein Artikel-Objekt entgegennimmt und ein jQuery-Objekt zurückliefert. Das zurückgelieferte jQuery-Objekt enthält den DOM-Knoten mit dem Blogartikel (das `<article>`-Element und dessen gesamten Inhalt). Verwenden Sie weitestgehend den Code aus der vorherigen Aufgabe.

Ersetzen Sie den in der vorherigen Aufgabe erstellten Code in der `artikel.html` durch einen Aufruf der Funktion `createArticle` und das anschließende Einhängen des Rückgabewerts in das DOM.

Aufgabe 4

Binden Sie jQuery, `data.js` und `blog.js` in die `index.html` ein. Nutzen Sie die Funktion `createArticle`, um in einer Schleife alle Artikel des Arrays `articles` zu erzeugen und in das DOM der Seite einzuhängen.

Aufgabe 5

Schreiben Sie in der `blog.js` eine Funktion `createSidebar`, die ein jQuery-Objekt zurückliefert, welches das Markup der Seitenleiste enthält (in der Musterlösung das `<aside>`-Element und dessen gesamten Inhalt).

Binden Sie jQuery, `data.js` und `blog.js` in alle HTML-Dateien ein und verwenden Sie die Funktion `createSidebar`, um die Seitenleiste zu erzeugen.

Aufgabe 6

Schreiben Sie in der `blog.js` eine Funktion `createTagCloud(articles)`, welche ein Array mit Blogartikeln entgegennimmt und ein jQuery-Objekt zurückliefert, das die Tag-Cloud enthält.

Verwenden Sie die Funktion `createTagCloud` in der Funktion `createSidebar` zur Anzeige der Tag-Cloud.

Hinweis: Als Zwischenschritt empfiehlt es sich, die Tag-Cloud nicht mit verschiedenen Schriftgrößen zu realisieren, sondern einfach nur alle vorhandenen Tags aus allen Artikeln zu sammeln.

Aufgabe 7

Erweitern Sie die Datei `artikelNeu.html` so, dass beim Klick auf Absenden mittels jQuery die Formularinhalte ausgelesen werden. Erzeugen Sie aus den Formulardaten ein Artikel-Objekt und geben Sie das Objekt auf der Konsole aus. Später wird an dieser Stelle das neue Objekt an den Server geschickt.

Sorgen Sie dafür, dass nach der Ausgabe auf der Konsole automatisch die Seite `index.html` angezeigt wird.

Aufgabe 8

Erweitern Sie die Datei `artikelNeu.html` so, dass beim Laden der erste Artikel aus dem Array `articles` in das Formular geladen wird. Dies könnte später dazu genutzt werden, einen bereits bestehenden Artikel zu bearbeiten. Das neue Objekt würde dann an dieser Stelle an den Server geschickt.

Aufgabe 9

Diese Aufgabe steht nicht in Zusammenhang mit dem Blogprojekt.

Der Foto-Dienst flickr bietet eine öffentliche API zur Abfrage von Inhalten:
https://www.flickr.com/services/feeds/docs/photos_public/

Nutzen Sie jQuery-Funktionen, um zehn Bilder mit dem Tag **Dortmund** per AJAX abzurufen und diese in einer HTML-Datei anzuzeigen.