

# Internetanwendungen für mobile Geräte – Prüfungsfragen

*Virtuelle Fachhochschule und Beuth Hochschule für Technik*

*Sommersemester 2018*

*Jörn Kreutel*

## Inhaltsverzeichnis

1	Strukturierung und Verknüpfung von Inhalten (HTM) . . . . .	2
2	Graphische Oberflächengestaltung (CSS) . . . . .	3
3	Interaktionssteuerung mit JavaScript (JSL) . . . . .	4
4	Client-Server Interaktion mit HTTP und XMLHttpRequest (JSR) . . . . .	6
5	CRUD Operationen via HTTP mit NodeJS und MongoDB (NJM) . . . . .	7
6	CRUD Datenzugriff mit Formularen (FRM) . . . . .	8
7	Fortgeschrittene Handhabung von Formularen (MFM) . . . . .	9
8	Verwendung von Multimedia (MME) . . . . .	10
9	Lokale Datenspeicherung (LDS) . . . . .	11
10	Offline Webapps (OFF) . . . . .	12

## 1 Strukturierung und Verknüpfung von Inhalten (HTM)

- Auszeichnungssprachen

1. Wozu dienen Auszeichnungssprachen im allgemeinen?
2. Welche wesentliche Unterscheidbarkeit muss bei Anwendung einer Auszeichnungssprache gewährleistet sein?
3. Wie kann die Unterscheidbarkeit bei maschinenlesbaren Auszeichnungssprachen gewährleistet werden?
4. Worin unterscheiden sich Auszeichnungssprachen außer in ihrer Syntax?
5. Welche Typen von Auszeichnungssprachen lassen sich im allgemeinen voneinander unterscheiden?

- XML

1. Inwiefern ist die XML eine Auszeichnungssprache?
2. Welche Struktur haben alle XML Dokumente gemeinsam?
3. Nennen Sie vier syntaktische Regeln der XML.
4. Erläutern Sie die Konzepte von *Wohlgeformtheit* und *Gültigkeit* eines Textdokuments im Sinne der XML.
5. Warum kann es wichtig sein, dass die Gültigkeit eines Markupsprachen-Dokuments verifiziert werden kann?

- HTML

1. Entspricht HTML den Anforderungen der XML?
2. Nennen Sie drei mögliche Abweichungen zwischen HTML und den Regeln der XML.
3. Weshalb hat sich XML konformes HTML nicht flächendeckend durchgesetzt?
4. Was ist ein Defizit von HTML, das auf dessen dokumentenbezogene Sichtweise zurückgeführt werden kann?
5. Was ist der Anwendungsbereich/ die Domäne von HTML?
6. Auf welcher Grundlage werden die Referenzen auf HTML Dokumente und andere Inhalte, die ein HTML Dokument enthalten kann, aufgelöst?

- HTML5

1. Inwiefern wird mit dem Begriff 'HTML5' eine Auszeichnungssprache bezeichnet und was bezeichnet der Begriff im weiteren Sinne?
2. Nennen Sie 5 Ausdrucksmittel, die mit der Bezeichnung 'HTML5' gemeint sein können.
3. Nennen Sie 4 kommerzielle Mehrwerte von HTML5.
4. Nennen Sie 5 Elemente, die HTML5 zur Strukturierung von Inhalten neu einführt.
5. Inwiefern dient HTML5 der Strukturierung von Multimedia-Inhalten?
6. Was sind technische Vorteile der semantischen Markup-Elemente von HTML5?
7. Was sind Vorteile des semantischen Markups aus Nutzersicht?

- Dokumentstruktur und Gestaltung

1. Wovon hängt die Entscheidung für eine konkrete Dokumentenstruktur ab?
2. Welcher Zusammenhang besteht zwischen Sinneinheiten/Bedeutungseinheiten und Gestaltungseinheiten?
3. Welche 4 Ausdrucksmittel stellt HTML für die Formulierung anwendungsspezifischer semantischen Markups zur Verfügung?

## 2 Graphische Oberflächengestaltung (CSS)

- Style
  1. Wie können ohne Verwendung von CSS Style-Properties auf einem HTML Element gesetzt werden?
  2. Nennen Sie 4 Gestaltungsmerkmale, hinsichtlich derer die Darstellung von HTML Elementen mittels CSS kontrolliert werden kann.
  3. Wofür werden Vendor-Prefixes in CSS verwendet? Nennen Sie ein Beispiel Prefix.
- Regeln
  1. Was ist die allgemeine Form einer Regel in einer Regelsprache und die spezifische Form einer CSS Regel?
  2. Welche Entscheidungsprobleme können bei Verwendung einer Regelsprache bezüglich der auszuführenden Aktion auftreten?
  3. Wie sollten Regeln idealerweise formuliert werden?
- CSS als Regelsprache
  1. Von welcher Art sind die Entscheidungen, die CSS als Regelsprache ermöglichen soll?
  2. Inwiefern ergänzen sich HTML und CSS?
  3. Welche beiden Ausdrucksmittel stellt CSS zur Formulierung der Vorbedingungen von Regeln zur Verfügung?
  4. Wie löst CSS das Problem, dass die Vorbedingung von mehr als einer CSS Regel zutreffen kann
- Selektoren
  1. Was tut ein Selektor?
  2. Wie können Elemente in HTML Dokumenten identifiziert werden?
  3. Über welche einfachen Selektoren verfügt CSS?
  4. Was wird durch ein Leerzeichen zwischen zwei CSS Selektoren zum Ausdruck gebracht?
  5. Welches Zeichen fungiert in CSS als Disjunktionsoperator ('oder')?
  6. Wie kann eine Negation in einem CSS Selektor ausgedrückt werden?
  7. Wofür werden Pseudoklassen in CSS verwendet?
  8. Wie wird die Spezifität eines Selektors ermittelt?
  9. Weshalb ist es nicht ausgeschlossen, dass einem Element inkonsistente Style-Properties zugewiesen werden?
  10. Kann ein Nutzer die durch einen Entwickler gesetzten Style-Properties 'überschreiben'?
- Media Queries
  1. Was ermöglichen Media Queries in Bezug auf die Anwendung von CSS Regeln?
  2. Welche Kategorien von Eigenschaften können in Media Queries überprüft werden?
  3. Wie verhalten sich Media Queries zur Spezifitätshierarchie für Selektoren und wie erfolgt ihre Auswertung?
- CSS Ausdrucksmittel
  1. Was wird als 'Vererbung' in CSS bezeichnet?
  2. Was wird durch die `display`-Property zum Ausdruck gebracht?
  3. Was können Sie tun, um die Darstellung eines Elements mittels CSS zu unterbinden?
  4. Welche Möglichkeiten haben Sie zur Angabe von Farbwerten in CSS?
  5. Nennen Sie von innen nach außen die 4 Boxen des CSS-Box-Modells.
  6. Werden Tochterelemente eines HTML Elements entsprechend der Reihenfolge ihres Auftretens in horizontaler oder vertikaler Anordnung dargestellt?

### 3 Interaktionssteuerung mit JavaScript (JSL)

- Ausdrucksmittel von JavaScript

1. Worin unterscheidet sich JavaScript als Sprache von HTML bzw. von CSS?
2. Was ist ECMAScript?
3. Nennen Sie fünf Statements, über die JavaScript verfügt.
4. Welche Operatoren verwendet JavaScript zur Überprüfung von Äquivalenz bzw. Identität zweier Werte?
5. Über welche Datentypen verfügt JavaScript?
6. Welche Funktion kann in JavaScript sowohl für das Entfernen, als auch für das Einfügen von Elementen in einen Array verwendet werden?
7. Notieren Sie zwei Formulierungen, mit denen Sie in JavaScript auf das Attribut `attr` eines Objekts `obj` zugreifen können.
8. Wie können Sie ein Attribut `attr` von einem Objekt `obj` entfernen?
9. Nennen Sie drei Aspekte, hinsichtlich derer JavaScript syntaktisch 'liberaler' ist als Java
10. Worin besteht ein dynamischer Aspekt bei der Handhabung von Datentypen in JavaScript?
11. Spielen in JavaScript Datentypen zur Laufzeit eine Rolle?
12. Müssen Sie sich in JavaScript darum sorgen, ob eine Variable einen nicht-`null` Wert hat?
13. Wie können Sie in einem `if`-Statement möglichst einfach überprüfen, ob eine `Object`-wertige Variable `myvar` einen Wert hat und weder `undefined`, noch `null` ist?
14. Für welche Werte primitiver Datentypen liefert die vorherige Überprüfung ein nicht korrektes Ergebnis, da sie ebenfalls weder `null`, noch `undefined` sind?
15. Wie lange sind die Werte globaler Variablen in JavaScript verfügbar?
16. Welche besondere Verwendungsmöglichkeit bietet Ihnen Java bezüglich Funktionen?
17. Wie können Sie eine Funktion, die Ihnen als Wert eines Arguments `func` übergeben wird, aufrufen?

- Interaktionssteuerung mit JavaScript

1. Was gehört zur *Interaktionssteuerung* einer graphischen Nutzerschnittstelle?
2. Auf welche Weise kann eine graphische Nutzerschnittstelle einem Nutzer Feedback bezüglich einer Nutzereingabe geben?
3. Was unterscheidet Toasts von Dialogen?
4. Welche Rolle übernehmen DOM Events und die DOM API in Bezug auf den Verarbeitungszyklus einer MVC Architektur?
5. Was ist das 'DOM Objekt'?
6. Wie können Sie in JavaScript auf das DOM Objekt zugreifen?
7. Nennen Sie 5 DOM Events
8. Wo können Sie Event Handler deklarieren, die das Auftreten von Ereignissen bezüglich der Elemente eines HTML Dokuments behandeln?
9. Welche besondere Möglichkeit bietet Ihnen die Zuweisung eines Event Handlers mittels der Methode `addEventListener()`?
10. Worin besteht die *Capture Phase* und worin die *Bubbling Phase* im Verarbeitungsmodell für DOM Events?
11. Was ist ein *event target*?
12. Wie können Sie verhindern, dass ein DOM Event in der Bubbling Phase nicht nur vom unmittelbar angezielten Event Target, sondern zusätzlich auch von dessen Vorgängerelementen verarbeitet wird?
13. Welche beiden Funktionen stellt Ihnen JavaScript für die Darstellung modaler Dialoge zur Verfügung?
14. Welche beiden Möglichkeiten bietet Ihnen die DOM API, um aus einem Dokument alle `<span>` Elemente auszulesen?
15. Worin unterscheiden sich die Methoden `querySelector()` und `querySelectorAll()`?

16. Wie können Sie in JavaScript ein neues `<li>`-Element zu einer existierenden nicht-leeren `<ul>`-Liste hinzufügen?
17. Warum ist der Zugriff auf `innerHTML` nicht geeignet, um ein neues `<li>`-Element zu einer existierenden nicht-leeren `<ul>`-Liste hinzuzufügen? Was können Sie stattdessen verwenden?
18. Wie können Sie dem `class` Attribut eines Elements `element` ohne Verwendung von String-Operationen einen neuen Wert hinzufügen?

- DOM, JavaScript und CSS

1. Müssen Sie nach Änderung des DOM Objekts zur Neudarstellung der dargestellten Ansicht CSS aufrufen?
2. Welche Pseudoklassen stellt CSS zur Reaktion auf Nutzerinteraktion zur Verfügung?
3. Was sind CSS Transitions?
4. Wie können Sie Ansichtselemente temporär aus einer dargestellten Ansicht 'entfernen'?
5. Wie kann in CSS ein Ansichtselement aus einer Darstellung entfernt werden?

## 4 Client-Server Interaktion mit HTTP und XMLHttpRequest (JSR)

- XMLHttpRequest, JSON, Callbacks

1. Wozu dient XMLHttpRequest in Web Applikationen?
2. Können HTTP Requests mit XMLHttpRequests auch synchron ausgeführt werden? Falls ja, auf welche Weise?
3. Bringen Sie die folgenden Methoden und -Attributzugriffe bezüglich eines XMLHttpRequest Objekts in eine sinnvolle Reihenfolge: `.status`, `.send()`, `.responseText`, `.setRequestHeader()`, `.open()`
4. Wie können Sie auf HTML Inhalte zugreifen, die Ihnen ein Server als Response in einem XMLHttpRequest mit dem Response-Header `Content-Type: text/html` übermittelt?
5. Was ist JSON?
6. Handelt es sich hierbei um JSON: `[{"name": "item 1", "id": "0"}]`? Begründen Sie Ihre Antwort.
7. Weshalb ist die Bezeichnung 'Callback-Funktion' sinnvoll?
8. Weshalb müssen Sie bei Aufruf asynchron ausgeführter Funktionen Callback-Funktionen anstelle von Rückgabewerten verwenden?
9. Formulieren Sie die Funktion `f1` im folgenden Beispiel um unter der Annahme, dass `f2` asynchron ausgeführt wird und ihr Ergebnis einer Callback-Funktion als Argument übergibt. Dieser Wert soll dem Nutzer/Caller von `f1` zur Verfügung gestellt werden:

```
function f1(a1) {  
    var v1 = f2(a1);  
    return v1;  
}
```
10. Erläutern Sie den Begriff 'AJAX'. Welche JavaScript APIs werden für die Umsetzung von 'AJAX' Funktionsmerkmalen verwendet?

## 5 CRUD Operationen via HTTP mit NodeJS und MongoDB (NJM)

- NodeJS

1. Was ist der Nachteil bei der Verwendung von Betriebssystem-Threads zur Bearbeitung der Anfragen an einen Server?
2. Was ist der Nachteil von synchron ausgeführten I/O Operationen, insbesondere falls mehrere Operationen nacheinander ausgeführt werden sollen?
3. Nennen Sie drei Typen von I/O Operationen, die in NodeJS asynchron ausgeführt werden können.
4. Welche Vorteile erwachsen aus der Handhabung von I/O Operationen in NodeJS?
5. Was ist die Aufgabe des 'Event Loop' in NodeJS?

- MongoDB

1. Nennen Sie drei Typen von Datenbanken, die als 'NoSQL' bezeichnet werden können.
2. Worin liegen die Vorteile relationaler Datenbanken gegenüber NoSQL Ansätzen?
3. Worin liegen die Vorteile von NoSQL Datenbanken gegenüber relationalen Datenbanken?
4. Was ist eine 'Collection' in MongoDB?
5. Für welchen Typ von Beziehung zwischen Instanzen komplexer Datentypen erleichtert eine Datenbank wie MongoDB die Repräsentation, verglichen mit relationalen Datenbanken? Welcher Typ von Beziehung erfordert ggf. höhere manuelle Aufwände bei der Implementierung?
6. Was für ein Programmierkonstrukt müssen Sie verwenden, um auf Ebene der JavaScript API für MongoDB auf das Ergebnis einer CRUD Operation zu reagieren?
7. Was bezeichnet der Begriff 'upsert' in MongoDB?

- CRUD Operationen

1. Was ist die Bedeutung der Bestandteile der Abkürzung CRUD?
2. Mittels welcher HTTP Methoden können CRUD Operationen identifiziert werden? Ordnen Sie die Methoden der jeweiligen Operation zu?
3. Für welche CRUD Operationen müssen Sie bei Identifikation durch HTTP Requests keinen Request Body verwenden?
4. Angenommen, `/elements/` identifiziert eine Menge von Objekten: wie können Sie durch eine URL ohne Verwendung von URL Query Parametern ein konkretes Element dieser Menge mit Identifikator ID identifizieren?
5. Beschreiben Sie die Schritte, die ausgeführt werden, wenn eine Webanwendung browser-seitig via HTTP auf eine server-seitige CRUD Operation zugreift.

## 6 CRUD Datenzugriff mit Formularen (FRM)

- **Formulare**

1. Welche beiden Typen von Anliegen können durch Formular zum Ausdruck gebracht werden
2. Was ist der grundlegende Unterschied zwischen Informations- im Ggs. zu Transaktionsanliegen?
3. In welchem Bezug stehen die beiden Anliegenstypen zu den vier CRUD Operationen?
4. Wann kann es erforderlich sein, die durch ein Formular ausgedrückten Sachverhalte zu modifizieren?
5. Was kennzeichnet Formulare als Bestandteile maschineller graphischer Nutzerschnittstellen?
6. Was sind 'Formulardaten'?
7. Was wird durch Formulardaten beschrieben und worin unterscheiden sich diesbezüglich die beiden Typen von Anliegen, die durch ein Formular ausgedrückt werden können?
8. Nennen Sie fünf Usability-Anforderungen bezüglich Formularen.
9. Nennen Sie fünf funktionale Anforderungen bzw. Gestaltungsanforderungen, die für die Bereitstellung nutzerfreundlich bedienbarer Formulare umgesetzt werden müssen.
10. Hinsichtlich welcher implementierungsbezogener Merkmale unterscheiden sich Ausdrucksmittel zur Implementierung von Formularen im Rahmen einer MVC Architektur?

- **HTML Formulare und Bedienelemente**

1. Für welche der allgemeinen implementierungsbezogenen Merkmale zur Behandlung von Formularen stellt HTML deklarative Ausdrucksmittel zur Verfügung?
2. Nennen Sie drei Typen von Eingabeelementen, die Sie in HTML für die Umsetzung einer Auswahl aus einer eingeschränkten Menge von Alternativen verwenden können.
3. Nennen Sie drei Möglichkeiten, um in einer mittels HTML realisierten Nutzeroberfläche ohne weitere Formatierungsinstruktionen Ihrerseits einen Button/Schaltfläche zu realisieren. Was ist der Unterschied zwischen `<input>` Elementen mit `type="submit"` vs. `type="button"` innerhalb eines Formulars?

- **Formularaktionen**

1. Nennen Sie drei Möglichkeiten, um in JavaScript auf ein Formular des dargestellten HTML Dokuments zuzugreifen – inklusive der Möglichkeit, die Sie im Rahmen der Behandlung z.B. eines `submit` Ereignisses bezüglich des Formulars haben.
2. Welche beiden Möglichkeiten haben Sie als Entwickler, um nach Betätigung eines `submit`-Elements in einem gültigen Formular die Ausführung der Formularaktion zu unterbinden?
3. Worin resultiert die Ausführung einer Formularaktion, falls als Wert von `action` *kein* `javascript:` angegeben wird?
4. Auf welche beiden Weisen können Sie Formularaktionen asynchron und ohne Neuladen des dargestellten Dokuments ausführen?
5. Welche HTTP Methoden können Sie in Formularen selbst verwenden und welche in JavaScript Code, der aus Formularen aufgerufen wird?
6. Worin unterscheiden sich `GET` und `POST` als Werte des `method` Attributs von Formularen im Hinblick auf die Übermittlung der Formulardaten via HTTP?

- **Data Binding**

1. Was wird im Rahmen einer MVC Architektur als 'Data Binding' bezeichnet?
2. Unterstützt HTML5 deklaratives Data Binding?
3. Nennen Sie zwei Typen *komplexer* GUI Elemente, für die Data Binding üblicherweise durchgeführt werden muss.
4. Können Formulare ineinander eingebettet werden? Was resultiert daraus bezüglich des Data Bindings?



## 7 Fortgeschrittene Handhabung von Formularen (MFM)

- Formularvalidierung

1. Welche drei Aspekte der Gültigkeit von Formularen können unterschieden werden?
2. Für welche Gültigkeitsaspekte stellt Ihnen HTML deklarative Ausdrucksmittel zur Verfügung?
3. Mittels welcher HTML Attribute können Sie die Bedingungen für die Vollständigkeit von Formularen deklarieren und welche Attribute verwenden Sie zur Deklaration der Wohlgeformtheitsbedingungen?
4. Wann benötigen Sie für `<select>` Elemente ein `required` Attribut?
5. Wann wird durch den Browser die Vollständigkeit von Formulardaten überprüft und wann die Wohlgeformtheit von Formularfeldwerten?
6. Welche Einschränkung weist das `disabled` Attribut hinsichtlich der browser-unterstützten Übermittlung von Formulardaten an eine `action`-URL auf?
7. Werden als `disabled` markierte Felder durch den Browser hinsichtlich Vollständigkeit und Wohlgeformtheit berücksichtigt?
8. Was sind 'versteckte' Eingabefelder und wofür sind sie sinnvoll?
9. Was wird durch das `invalid` Ereignis angezeigt und wie können Sie dieses Ereignis behandeln? Was passiert, wenn Sie keine Behandlung dafür implementieren?

- Anwendungsspezifische Validierung

1. Welche beiden Aspekte können hinsichtlich anwendungsspezifischer Validierung unterschieden werden?
2. Was wird durch das `ValidityState` Objekt repräsentiert und auf welcher Ebene eines Formulars ist dieses Objekt angesiedelt?
3. Was ermöglicht Ihnen die Methode `setCustomValidation()`?
4. Welche beiden Schritte sind erforderlich, um anwendungsspezifisch ein `invalid` Ereignis auszulösen?
5. Wie kann anwendungsspezifische Validierung vor Abschluss einer Formularbefüllung mittels `submit` initiiert werden?
6. Nennen Sie drei Eingabeereignisse, die bezüglich Formularelementen auftreten können.
7. Worin kann die Behandlung von Eingabeereignissen in Formularen bestehen?

- Formatierung von Formularen

1. Wie können Sie auf Ebene von CSS die Darstellung von Checkboxes und Radiobuttons in Formularen beeinflussen?
2. Wie können Texteingabefelder und Button-Darstellungen beeinflusst werden?
3. Nennen Sie drei CSS Pseudoklassen, die speziell für die Gestaltung von Formularen verwendet werden können.

- Multipart Formulare

1. Worin unterscheiden sich 'gewöhnliche' Formulardaten von Multipart-Formulardaten?
2. Können Multipart Formulardaten mit `GET` Requests übermittelt werden? Begründen Sie Ihre Antwort.
3. Woher weiß ein Server, welche Trennzeichenkette für die Segmentierung von Multipart Formulardaten verwendet wird?
4. Warum können in einem Multipart Formular Inhalte mehrerer Dateien mit unterschiedlichem Inhaltstyp übertragen werden?
5. Auf welche beiden Weisen können Multipart Requests initiiert werden?
6. Wie kann bei Übermittlung von Formulardaten an die `action` URL der vom Server übermittelte Response ohne Neuladen des dargestellten Dokuments verarbeitet werden?
7. Welche Vorteile bietet die Verwendung von `FormData` Objekten zur Übermittlung von Formulardaten?

## 8 Verwendung von Multimedia (MME)

- Hintergrund
  1. Inwiefern trifft die Charakterisierung von Bildern und Texten als 'statischen' Inhalten nur bedingt zu?
  2. Was ist mit dem Begriff des 'Prosumers' in Bezug auf die Nutzung des WWW gemeint?
  3. In welchem Zusammenhang stehen die beiden Begriffe 'Prosumer' und 'User Generated Content'?
  4. Was zeichnet native mobile Anwendungen hinsichtlich ihrer Eignung für Prosumer in der Regel gegenüber Webanwendungen aus?
- Wiedergabe von Audio und Video in Webanwendungen
  1. Welche drei 'architektonischen' Möglichkeiten bestehen derzeit für die Wiedergabe von Audio- und Videoinhalten aus dem WWW?
  2. Was ist ein Hindernis bezüglich der breiteren Nutzung der HTML5 Multimedia-Elemente?
  3. Welche Einschränkung weist Firefox bezüglich der Multimedia-Elemente auf und welche Mehraufwände entstehen dadurch ggf.?
  4. Wie können alternative Medienformate, die ggf. für verschiedene Browser erforderlich sind, bei der Verwendung der Multimedia-Elemente behandelt werden? Was ist damit nicht erforderlich?
  5. Was ist `HTMLMediaElement`?
  6. Was müssen Sie tun, um bei Verwendung der Multimedia-Elemente auf den Vorgang des Abspielens eines Videos reagieren zu können, z.B. um eine Fortschrittsanzeige zu aktualisieren?
  7. Was ist WebVTT?
- Aufnahme von Standbildern
  1. Welche Voraussetzungen müssen zur Aufnahme von Audio und Video mittels JavaScript gegeben sein?
  2. Welche Schritte sind erforderlich, um unter Verwendung von `video` und `canvas` ein Standbild aufzunehmen?
  3. Welche Rolle spielt das `<canvas>` Element bei der Erstellung eines Standbilds unter Verwendung der Kamera?
  4. Was sind Data URLs?
  5. Welches HTML Element stellt Ihnen seine Inhalte als Data URLs zur Verfügung?

## 9 Lokale Datenspeicherung (LDS)

- Lokale Datenspeicherung allgemein
  1. Nennen Sie 4 verschiedene Möglichkeiten zur client-seitigen Datenspeicherung, die von Webanwendungen genutzt werden können.
  2. Nennen Sie 3 Möglichkeiten zur Eindämmung des Sicherheitsrisikos beim Zugriff auf lokale Daten aus Webanwendungen?
  3. Was besagt die *Same Origin Policy* für Webanwendungen und wo kommt sie im Bereich der lokalen Datenspeicherung zum Einsatz?
- Caching
  1. Wozu dient Caching?
  2. Welche Ausdrucksmittel des HTTP Protokolls bzw. Bestandteile von Requests und Responses werden für Caching genutzt?
  3. Welche beiden Ansätze können in Bezug auf Caching voneinander unterschieden werden?
- Cookies
  1. Wozu dienen Cookies
  2. Welche Ausdrucksmittel des HTTP Protokolls werden für Cookies verwendet?
  3. Welche Einschränkungen bestehen bezüglich der Verwendung von Cookies in JavaScript?
- Web Storage
  1. Was ist ein Browsing Context?
  2. Was ist der Unterschied zwischen `sessionStorage` und `localStorage`?
  3. Was für eine Datenstruktur liegt der `Web Storage` API zugrunde?
  4. Was zeigt das `storage` Event an und wozu kann es verwendet werden?
  5. Welche Einschränkung bringt `Web Storage` bezüglich der Speicherung von Objekten mit sich und was ist dafür ggf. erforderlich?
- IndexedDB
  1. Was ist IndexedDB?
  2. Was sind Object Stores in IndexedDB?
  3. Was brauchen Sie, um das Ergebnis einer CRUD Operation in IndexedDB weiter verarbeiten zu können?
  4. Was ist der Vorteil der Verwendung von Transaktionen in IndexedDB?
  5. Wozu dient ein Cursor in IndexedDB und mit welchem Konstrukt aus Java ist seine Handhabung vergleichbar?

## 10 Offline Webapps (OFF)

- Offline Applikationen

1. Was legt die Cache Manifest Datei fest?
2. Was für ein Notationsformat (z.B. JSON, XML, andere Textformate) wird für die Cache Manifest Datei verwendet?
3. Was passiert, wenn ein Cache Manifest eine Ressource als offline Ressource referenziert und diese nicht geladen werden kann?
4. Was passiert, wenn im Onlinebetrieb auf eine Ressource zugegriffen wird, die gar nicht im Cache Manifest angegeben wird?
5. Wann greift der Browser auf die Cache Manifest Datei zu?
6. Wann werden Offline Ressourcen vom Browser ins Cache geladen? Bei der Erstellung oder Aktualisierung des Cache oder wenn der erste Zugriff auf die Ressource erfolgt?
7. Wann werden server-seitig aktualisierte Offline-Ressourcen vom Browser geladen und wann sind sie im Browser verfügbar?
8. Wo können im Cache Manifest Wildcards verwendet werden?
9. Wozu dient die `ApplicationCache` API?
10. Weshalb ist die Verwendung des Offline Cache nicht 'natürlicherweise' für das Caching dynamischer Inhalte, die durch die Nutzer einer Anwendung bereit gestellt werden, geeignet? Was für ein Workaround könnte verwendet werden, um es geeignet zu machen?

- Webapps

1. Ist der Begriff der 'Webapp' wohl definiert bzw. durch einen Standard abgedeckt?
2. Inwiefern werden mit Mozilla Open Web Apps (OWA) Funktionen und Handhabung nativer mobiler Anwendungen für Web Applikationen verfügbar gemacht? Nennen sie 4 Aspekte.
3. Sind Mozilla OWA grundsätzlich offline-fähige Anwendungen?
4. Wie kann eine OWA bei Verwendung eines gewöhnlichen Firefox Browsers installiert werden?
5. Was kann im Manifest einer OWA angegeben werden? Nennen Sie 4 Angaben.