Kursplan

Målet med kursen är att den studerande ska genom teori och praktiska övningar, utveckla sina kunskaper och färdigheter i att använda programmeringsspråket JavaScript; hur språket skrivs, hur det exekverar samt dess olika möjligheter och begränsningar. Kursen introducerar de olika datatyperna i JavaScript och ger en inblick i hur språket fungerar och kan användas i en front-end miljö. De studerande får öva att hantera händelser, göra AJAX-anrop, manipulera DOM och dynamiskt förändra webbsidor.

Kursplan

- Kursen omfattar följande moment
    - Fördjupning i JavaScript
    - Versionshantering i Git
    - Felsökning av JavaScript
    - Event-hantering med JavaScript
    - Manipulera DOM med JavaScript

Kursplan

- Efter genomförd kurs ska den studerande ha kunskaper i/om:
    - versionshantering, Git
    - AJAX-anrop
    - manipulering av DOM
    - dynamisk förändring av CSS
    - mer avancerade JS-koncept som *this* och *prototype*

Kursplan

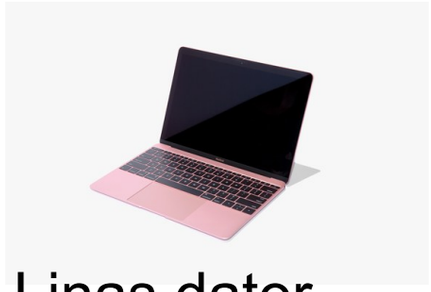
- Efter genomförd kurs ska den studerande ha färdigheter i att:
  - manipulera DOMen med hjälp av JavaScript
  - göra asynkrona AJAX-anrop för att hämta samt skicka data mellan frontend och backend
  - kunna hämta, ändra och skapa olika HTML-element med JavaScript
  - versionshantering med Git

Kursplan

- Efter genomförd kurs ska den studerande ha kompetens för att:
  - arbeta med versionshantering
  - implementera en problemställning i JavaScript
  - självständigt använda JavaScript för att skapa dynamiska sidor
  - utveckla interaktiva webbsidor med JavaScript

Client - Server

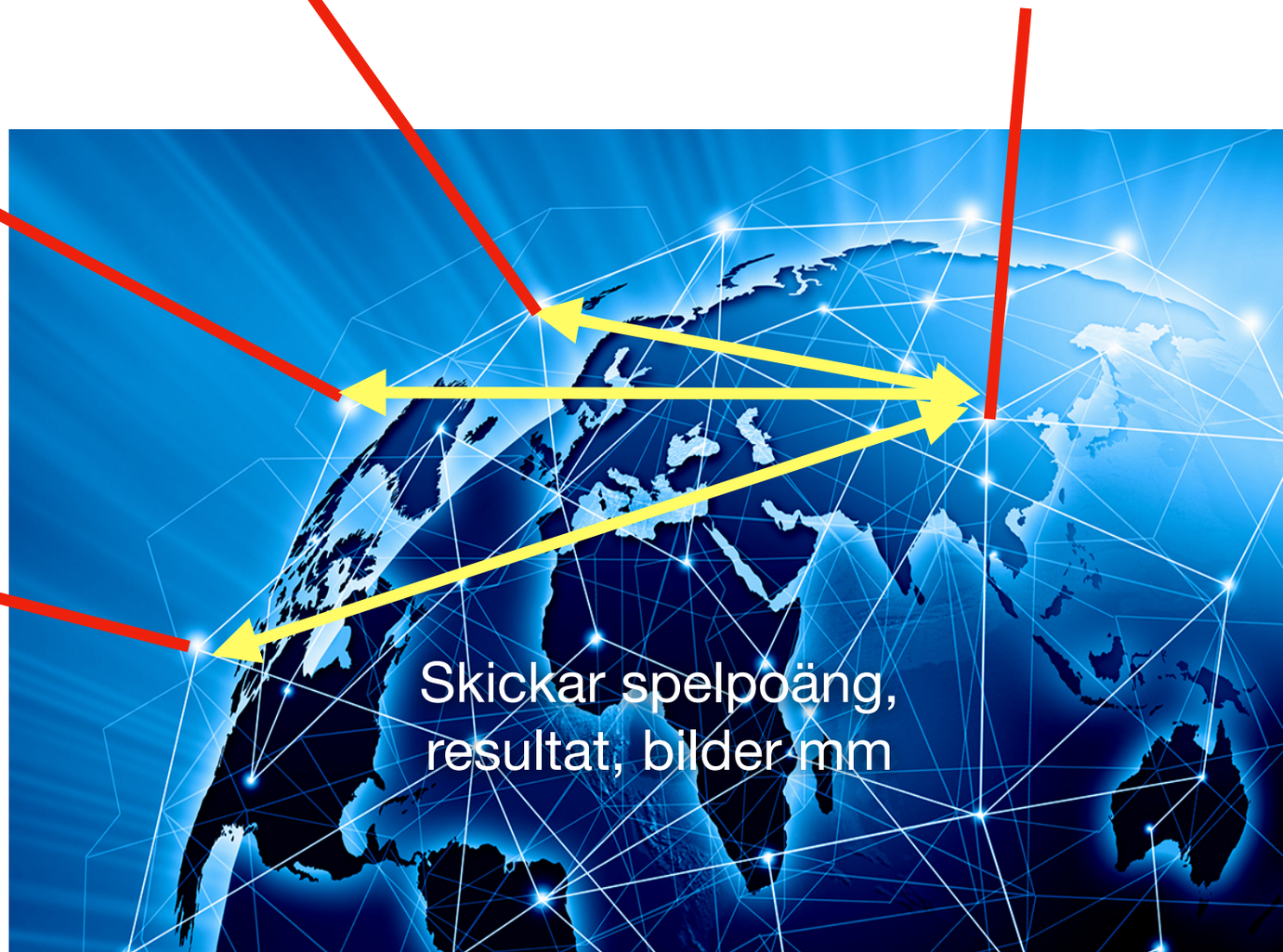
www.yatzy.se
Server



Linas dator
Spelare 3
Klient

Jessicas dator
Spelare 2
Klient

Mickes dator
Spelare 1
Klient



Client - Server

- Vi har inte lärt oss någon backendprogrammering.

- Kan vi spara data på klienten?

- Ja, men den är bara giltig för vår klient, vår yatzyspelare. Ingen annan spelare kan se data vi sparar lokalt.

Spara data i klienten

- Cookies
- Session storage / Local storage (Web storage)
- Cache API
- IndexedDB
- FileSystem API
- https://www.html5rocks.com/en/tutorials/offline/storage/

Cookies

- Skapa

```
document.cookie = "username=John Doe;
expires=Thu, 18 Dec 2013 12:00:00 UTC";
```

- Läsa

```
let x = document.cookie;
```

- Lämpar sig bara i undantagsfall för att spara små mängder data.

Web storage

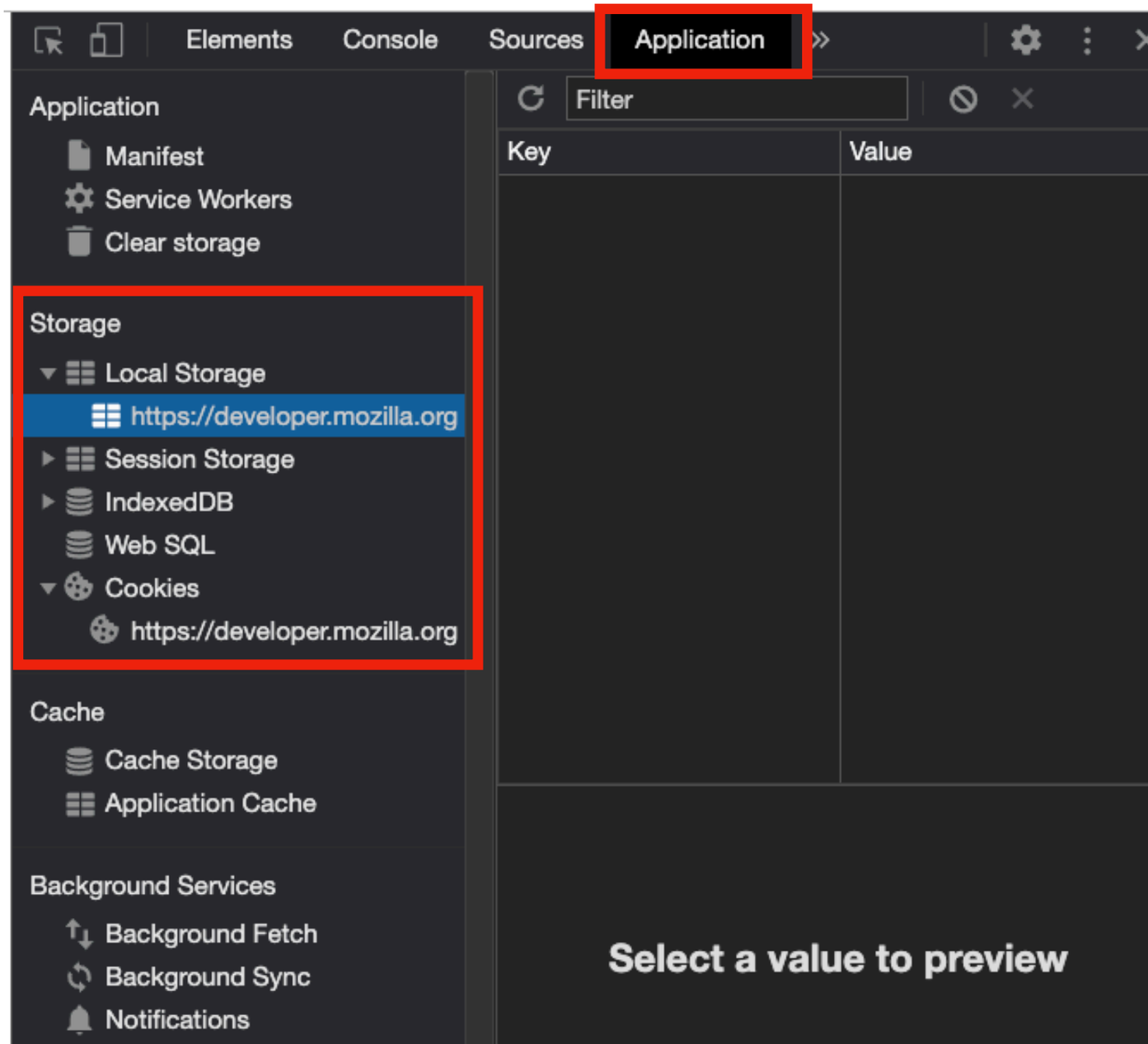
- *sessionStorage* maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores)
    - Stores data only for a session, meaning that the data is stored until the browser (or tab) is closed.
    - Data is never transferred to the server.
    - Storage limit is larger than a cookie (at most 5MB).
- *localStorage* does the same thing, but persists even when the browser is closed and reopened.
    - Stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser cache / Locally Stored Data.
    - Storage limit is the maximum amongst the three.

Local storage

- `localStorage.setItem('myCat', 'Tom');`
- `let cat = localStorage.getItem('myCat');`
- `localStorage.removeItem('myCat');`
- `localStorage.clear();`
- https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API

localStorage

- Vi kan inspektera med webbutvecklarverktyg.



Cache API

- The Cache API was created to enable service workers to cache network requests so that they can provide fast responses, regardless of network speed or availablity. However, the API can also be used as a general storage mechanism.

- https://web.dev/cache-api-quick-guide/

IndexedDB

- IndexedDB is a low-level API for client-side storage of significant amounts of structured data, including files/ blobs. This API uses indexes to enable high-performance searches of this data.

- IndexedDB API is powerful, but may seem too complicated for simple cases.

- https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API

FileSystem API

- The File System API and FileWriter API provide methods for reading and writing files to a sandboxed file system. While it is asynchronous, it is not recommended because it is only available in Chromium-based browsers.

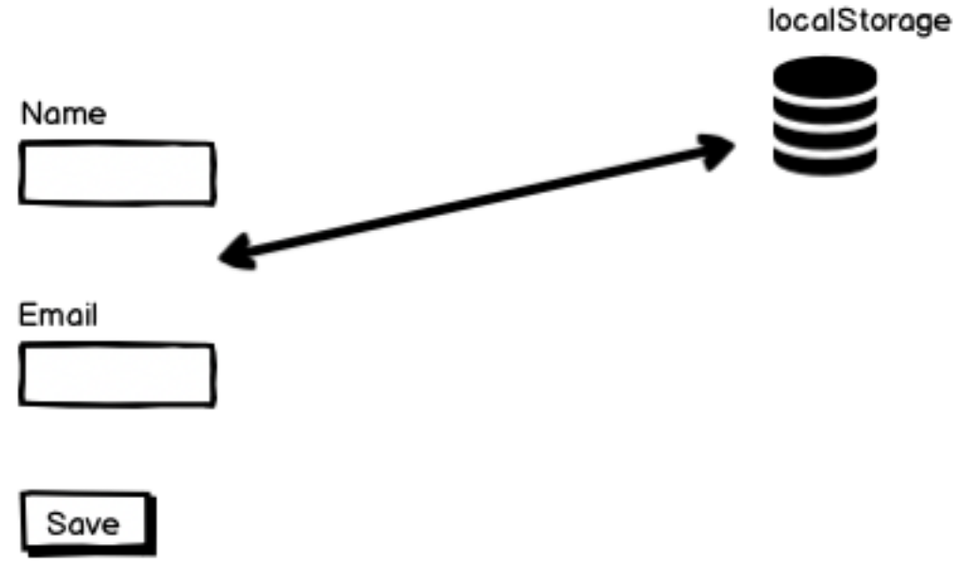
- The File System Access API was designed to make it easy for users to read and edit files on their local file system. The user must grant permission before a page can read or write to any local file, and permissions are not persisted across sessions.

localStorage

- *Uppgift:* Gör ett formulär med namn och epost.

- Spara uppgifterna när användaren klickar på en knapp och se att uppgifterna lagras i localStorage.

- Kontrollera om localStorage har något innehåll när ni laddar in sidan och sätt i så fall värdena i formuläret till det sparade datat.



Destructuring assignment

- The destructuring assignment syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.

```
let a, b;

[a, b] = [10, 20];

console.log(a); // expected output: 10
```

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

- https://hacks.mozilla.org/2015/05/es6-in-depth-destructuring/

Export / import

The export statement is used when creating JavaScript modules to export live bindings to functions, objects, or primitive values from the module so they can be used by other programs with the import statement.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/export

Export / import

- Funkar lite olika i Node / browser.
- You need to include this script in your HTML with a <script> element of type="module", so that it gets recognized as a module and dealt with appropriately.

this

A function's this keyword behaves a little differently in JavaScript compared to other languages. It also has some differences between strict mode and non-strict mode.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/this

prototype

- JS är egentligen inte ett klassiskt objektorienterat språk utan prototyp-baserat.

- Varje gång ett objekt skapas lägg en prototyp till som en egenskap i det objektet.

- Förr var prototyper det enda sättet man kunde lägga till och förändra egenskaper och metoder för en viss sorts funktioner.

- Nu har vi klasser, som inte helt gör samma sak, men i de flesta fall.

- https://www.geeksforgeeks.org/prototype-in-javascript/

- Lokal lagring
- Destructuring assignment
- Export/import
- this
- prototype