

Fetch API Övningar

Beskrivningar av olika övningar för att träna på Fetch API och AJAX



Instruktioner - Viktigt!

Mappstruktur och Filorganisation:

För varje övning du löser, skapa en ny mapp med följande struktur:

Exempel på mappstruktur:

```
fetch-exercises/  
├─ 1-exercise/  
│   ├─ index.html  
│   └─ script.js  
├─ 2-exercise/  
│   ├─ index.html  
│   └─ script.js  
├─ 3-exercise/  
│   ├─ index.html  
│   ├─ script.js  
│   └─ img/  
│       └─ loading.gif  
└─ ...och så vidare
```

Steg för varje övning:

1. Skapa en ny mapp med namnet "X-exercise" (där X är övningsnumret)
2. Skapa en index.html fil i mappen
3. Skapa en script.js fil i mappen
4. Länka din JavaScript-fil i HTML-filen med: `<script src="script.js"></script>`
5. Läs övningsbeskrivningen noggrant
6. Implementera lösningen i dina filer
7. Testa att allt fungerar korrekt innan du går vidare till nästa övning

Tips:

- Håll din kod organiserad och kommenterad

- Använd tydliga variabelnamn
- Testa varje övning i webbläsaren innan du fortsätter
- Du kan använda `console.log()` för att debugga

Övning 1: Fetch Text Data

Skapa en knapp "Fetch data".

När man klickar på knappen,

så skall texten från https://codexplained.se/api/lorem_text.php hämtas och visas på sidan.

Hämta texten med hjälp av AJAX, se kodgenomgången

Övning 2: Roll Dices

Skapa ett textfält och en knapp "Roll dices".

I textfältet skall man kunna ange ett nummer,

som är antal tärningar(slumpad siffra mellan 1-6), som skall hämtas från https://codexplained.se/api/dice_json_array.php?numberOfDice= värdet från textfältet

Undersök vad som visas i webbläsaren, med följande URLer:

https://codexplained.se/api/dice_json_array.php?numberOfDice=1

https://codexplained.se/api/dice_json_array.php?numberOfDice=4

Datan skall i sin tur visas i en lista, där varje tärning placeras i en listItem ``

Övning 3: Roll Dices with Loading Indicator

Kopiera in din lösning från övning 2

Följande skall justeras

Anropet sker nu istället till följande URL:

https://codexplained.se/api/dice_json_array_slow.php?numberOfDice=

Det gör samma saker, men anropet är betydligt långsammare via den nya URL:en

Medan anropet görs, skall en `img/loading.gif` bild visas, för att visa att anropet håller på att laddas/utföras.

Så fort datan har hämtats, ta bort gif-bilden och visa resultatet i en ny rad i tabellen.

Övning 4: Fetch Lorem Words

Skapa ett textfält och en knapp 'Fetch data'.

I textfältet skall man kunna ange ett nummer, som är antal ord som man vill hämta från

https://codexplained.se/api/lorem_json_array.php?numberOfWords=
värdet från textfältet

Undersök vad som visas i webbläsaren, med följande URLer:

https://codexplained.se/api/lorem_json_array.php?numberOfWords=3

https://codexplained.se/api/lorem_json_array.php?numberOfWords=10

Datan skall läggas in i en tabell <table></table>, med 2 kolumner:

- Kolumn 1 skall ha rubriken 'Number of words', och innehålla nummret som angavs
- Kolumn 2 skall ha rubriken 'Result', och innehålla datan. Datan är en array med ord, där varje ord skall visas i en egen listItem

Varje anrop skall hämta och placera datan i en ny rad, i tabellen

Skall ungefär se ut på följande sätt:

----- -----
Number of words Result
----- -----
2 .wer
.sfd
----- -----
3 .wer
.sfd
.ert
----- -----

Övning 5: Table with Multiple Buttons

Skapa en tabell med 5 kolumner och 2 rader

Första raden i tabellen skall innehålla en knapp per kolumn, dvs totalt fem knappar på först raden.

- Första knappen skall heta "1", och hämta endast ett ord från [https://codexplained.se/api/lorem_json_array.php?numberOfWords=](https://codexplained.se/api/lorem_json_array.php?numberOfWords=1)värdet som anges
- Andra knappen skall heta "2", och hämta två ord från samma URL
- Tredje knappen skall heta "3", och hämta tre ord från samma URL
- Gör samma sak för knapp 4 och 5

Undersök vad som visas i webbläsaren, med följande URLer:

https://codexplained.se/api/lorem_json_array.php?numberOfWords=3

https://codexplained.se/api/lorem_json_array.php?numberOfWords=10

Varje knapp hämtar datan och placerar datan under respektive knapp, i andra raden.

Datan är en array med ord, dessa ord skall visas i en lista där varje ord är en listitem

Skall ungefär se ut på följande sätt, efter att varje knapp gjort ett anrop

1	2	3	4	5
.asd	.asd	.qwe	.qwe	.wer
	.weq	.ewr	.gfd	.sfd
		.ewr	.gfd	.cvx
			.gfd	.dff
				.bvc

Övning 6: Previous/Next Navigation with Animals

Skapa 2 knappar, "Previous" och "Next"

Dessa knappar anropar samt skickar in ett värde samtidigt, som endast får vara en siffra mellan 0-4

Exempel:

- Om nuvarande värde är 2, då anropar "Next"-knappen med värde 3
- Om nuvarande värde är 1, då anropar "Previous"-knappen med värde 0
- Om nuvarande värde är 4 och "Next"-knappen trycks, här får ni välja mellan att inte göra något, eller räkna om från 0.

Knapparna anropar och hämtar data från

https://codexplained.se/api/animals_json.php?animal= Värdet som anges. Se exemplen nedan

Undersök vad som visas i webbläsaren, med följande URLer:

https://codexplained.se/api/animals_json.php?animal=4

https://codexplained.se/api/animals_json.php?animal=0

Datan är en sträng som är separerad med tecknet *

Hämta datan och placera in de olika delarna i:

<h1>

<i>

<p>

Exempel på hur datan delas upp och läggs in i HTML-elementen:

<h1>Slear</h1>

<i>A cross between a bear and a sloth</i>

<p>Lorem ipsum</p>

Övning 7: Simple Navigation Menu (About & Author)

Skapa en meny med 3 länkar: 'Blog posts', 'Author', 'About'

Beroende på vilken länk man trycker, skall sidans innehåll

uppdateras via API anrop till:

https://codexplained.se/api/simple_json.php

Notera att länkarna skall EJ ladda om sidan, utan se till att

anropen endast uppdaterar en HTML element (ex div#content), där

sidans innehåll placeras

Klickar man på 'About'-länken

- Visa då endast en rubrik "About" tillsammans med texten som finns i 'about'-parametern

Klickar man på 'Author'-länken

- Visa då endast en rubrik "Author" tillsammans med texten som finns i 'author'-parametern

Övning 8: Navigation Menu with Blog Posts

Kopiera in koden från föregående uppgift "7-exercise"

Skapa en 3:e länk: 'Blog posts'

Beroende på vilken länk man trycker, skall sidans innehåll uppdateras via API anrop till:

https://codexplained.se/api/simple_json.php

Notera att länkarna skall EJ ladda om sidan, utan se till att anropen endast uppdaterar en HTML element (ex div#content), där sidans innehåll placeras

Klickar man på 'Blog posts'-länken

- Sidan skall innehålla en lista av alla blogg-inlägg, som finns i 'blog_posts'-parametern.
- Ordet lista i detta fall syftar EJ på att man använder , utan se till att inläggen visas under varann likt en lista
- Varje inlägg skall visa:
 - 'title'- parametern som en h2-rubrik
 - 'date'- parametern under rubriken i kursiv textstil
 - 'text'- parametern under datum, i en egen <p>-element

Övning 9: Blog Posts with Tags

Kopiera in koden från föregående uppgift "8-exercise"

Bygg vidare på inläggslistan

- Se till att varje inlägg visar tillhörande taggar, under inläggs-texten. Nedan illustrerar hur ett inlägg bör se ut.

Today is awesome!

2015-11-18

Bacon ipsum dolor amet pastrami tenderloin pork chop jerky corned beef ground round bresaola flank.

Salami porchetta pancetta tenderloin, flank shoulder turducken pig jerky filet mignon.

Flank sausage leberkas corned beef venison hamburger.

Turkey pork loin short ribs meatball shankle sausage leberkas, bresaola pancetta pork jerky.

Beef ribs flank corned beef pastrami.

Tags: Bacon ipsum, tenderloin, pork, jerky, pancetta, pastrami

Övning 10: Collapsible Blog Posts

Kopiera in koden från föregående uppgift "9-exercise"

Bygg vidare på inläggslistan

- Se till att varje inlägg endast visar sin rubrik, resten av inlägget skall vara dolt
- Rubriken skall vara en klickbar länk (som EJ laddar om sidan)
- När man trycker på en rubrik, då skall inläggets innehåll visas under rubriken. Trycker man igen, då döljs innehållet återigen.
- Extra (jQuery): Försök lägga till en animation som visar/döljer inläggsinnehållet på ett snyggt sätt. Ex slide up/slide down