5. Website Building

Adrian Adiaconitei

LINK Academy

Obiective

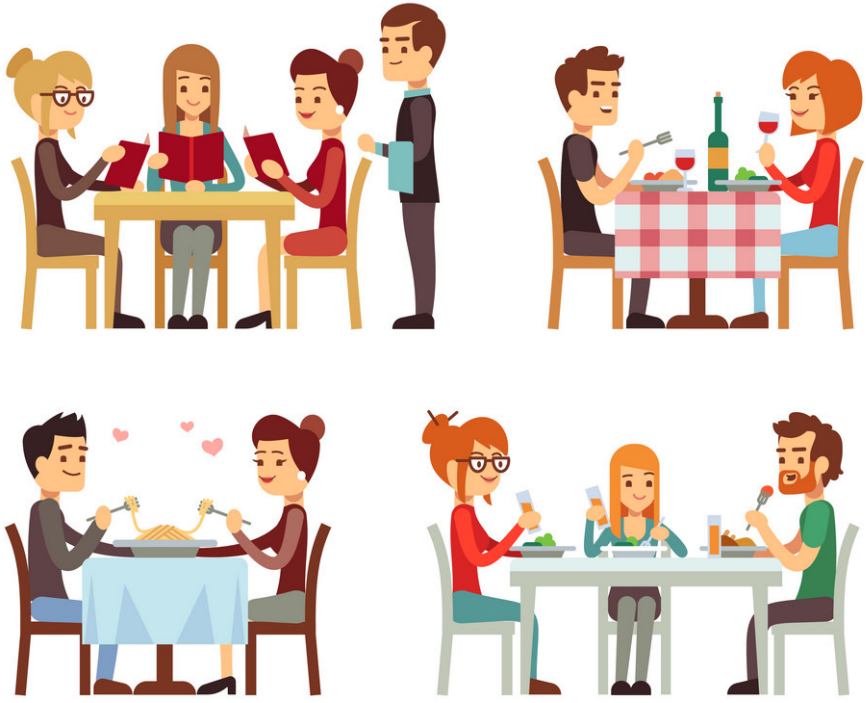
- ✓ Javascript/jQuery AJAX
  - ✓ Callback
  - ✓ PROMISE
  - ✓ ASYNC/AWAIT
- ✓ SVG

LINKAcademy

SINCRON

Ap1.html

ASINCRON



Ap2.html

Callback

- ✓ Funcția de tip callback este o funcție apelată în interiorul altei funcții(care și-a terminat executia) pentru a executa o altă acțiune

```
const button = document.getElementById('button');
function callback(){
        console.log("Hello world");
}

button.addEventListener('click',callback);
```

ap3.html

Callback

Funcția de tip callback $.ajax()

- ✓ Success
- ✓ Error
- ✓ Complete

Ap4.html

Promises jQuery >1.8

- ✓ done
- ✓ fail
- ✓ Always

Ap5.html

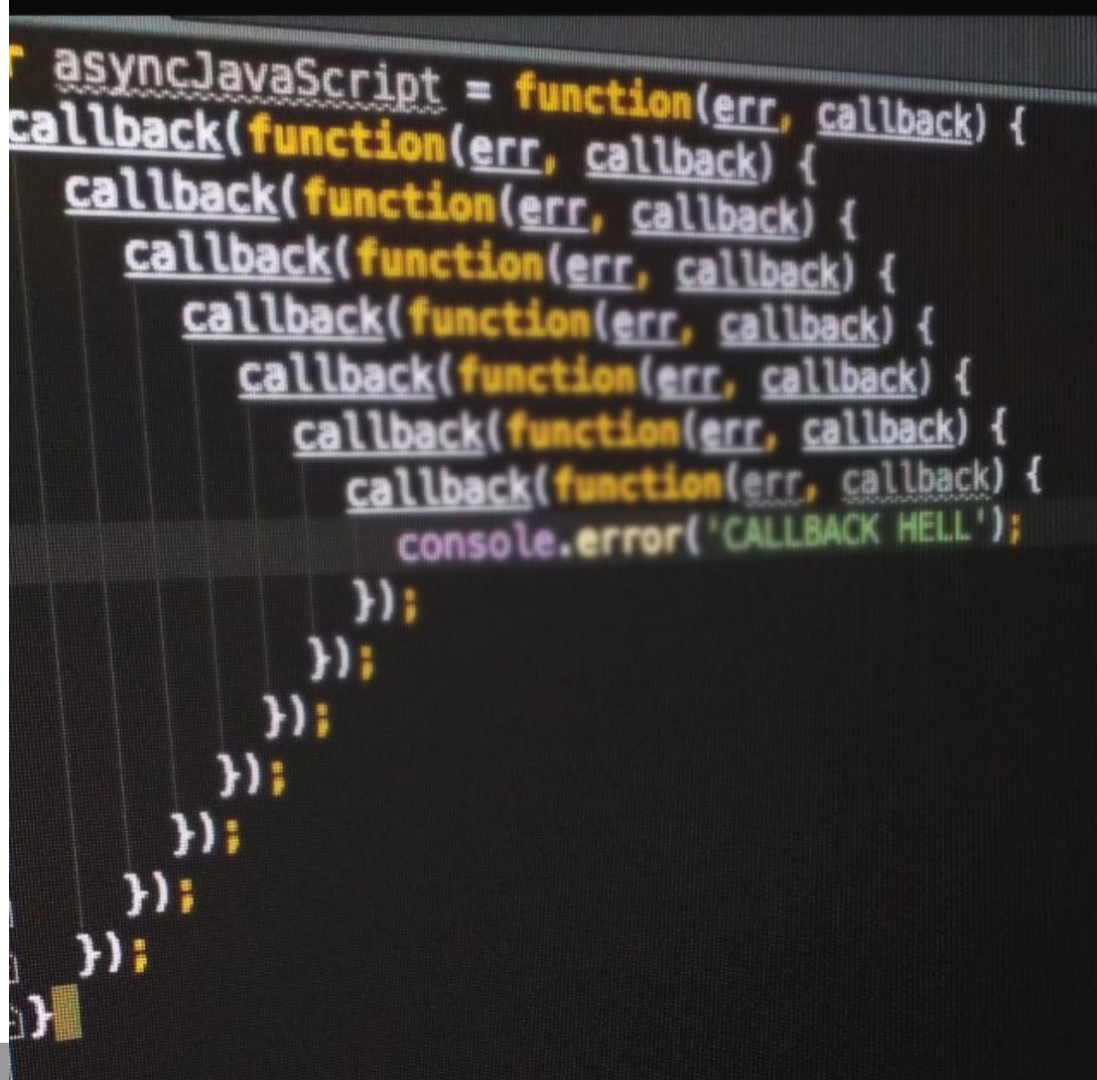
- ✓ https://codesport.io/coding/jquery-ajax-success-and-done/

LINKAcademy

Callback hell

- ✓ Apelam sincron mai multe funcții de tip callback, ceea ce va duce la un cod greu de administrat

- ✓ ap6.html



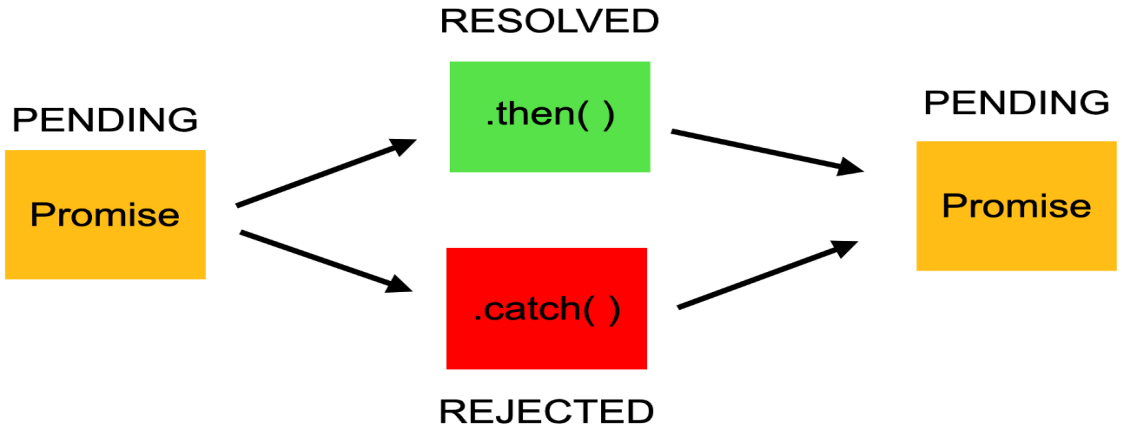
Promises

O promisiune: un obiect JavaScript care leagă un cod care poate dura ceva timp și un cod care trebuie să aștepte rezultatul ECMAScript 2015(ES6)

- **pending**: stare inițială, în asteptare, nici îndeplinită, nici respinsă
- **fulfille**: operațiunea a fost finalizată cu success
- **rejected**: operațiunea a eșuat

https://www.w3schools.com/js/js_promise.asp

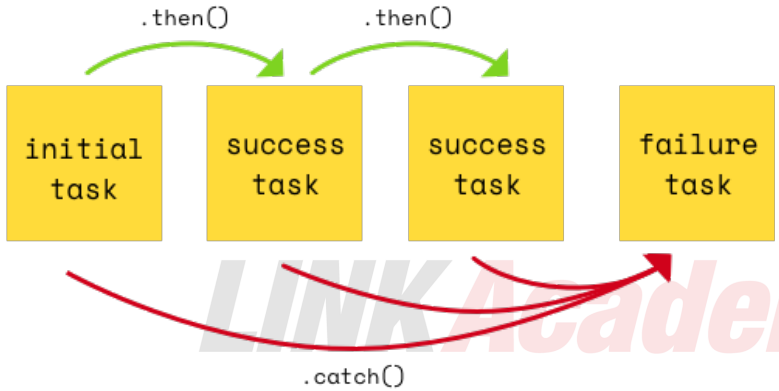
Ap7.html



Promises

Dacă este nevoie de a executa două sau mai multe operații asincrone, astfel încât fiecare operație ulterioară să înceapă atunci când operația anterioară reușește, de asemenea, operația ulterioară ia rezultatul operației anterioare. Acest lucru se realizează prin crearea **Promise Chaining** .

Ap7_1.html Ap7_2.html
Ap8.html



Async/await

```
async function f() {
return 1;
}
f().then(alert);
```

```
async function f() {
Return Promise.resolve(1);
}
 f().then(alert);
```

Async/await

Async/await – este **Syntactic sugar** pentru **PROMISES**

Syntactic sugar

- ✓ funcționalitate proiectată pentru rescrierea altei funcționalități făcând codul mai ușor de înțeles
- ✓ NU ne permite să facem ceva ce nu puteam face și înainte folosind alte metode(PROMISES)

Async/await

- ✓ Există două moduri principale de a gestiona codul asincron: then/catch (ES6 - 2015) și **async/await** (ES8-2017)
- ✓ Marele avantaj al async / await este că face ca codul asincron să pară sincron. (fă cererea, așteaptă să se termine și apoi dă-mi rezultatul)
- ✓ **Async** face ca o funcție să returneze o Promisiune
- ✓ **Await** face ca o funcție să aștepte pentru o Promisiune
- ✓ **Await** poate fi utilizat numai în cadrul unei funcții **async**. <ES2022
- ✓ **Await** oprește execuția unei funcții asincrone până când Promisiunea este rezolvată. Ap9.html. Ap9_1.html

Async/await

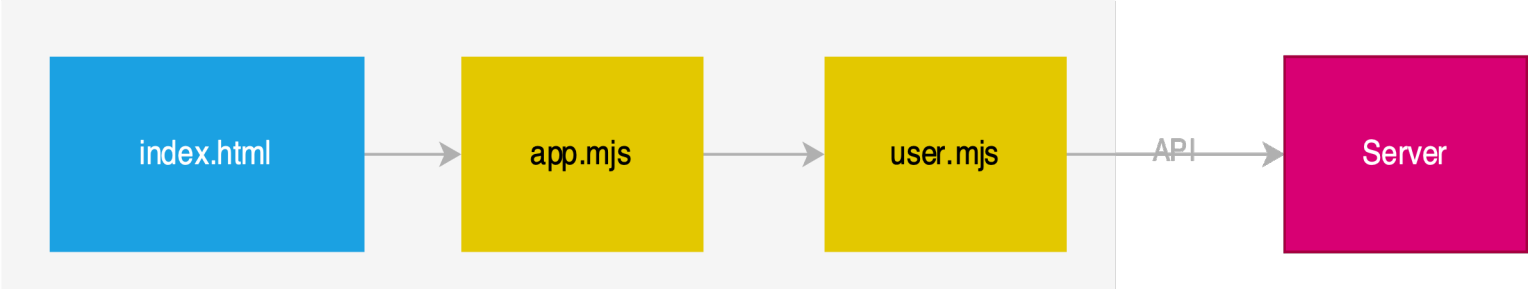
Ap10.html : Provocare Rescriem aplicatia ap2.html folosind async/await

Ap11.html : Provocare Rescriem aplicatia ap7_1.html folosind async/await

Async/await

Aplicație: Să afișam informațiile primate din acest API folosind async/await. (top-level await : ES2022):

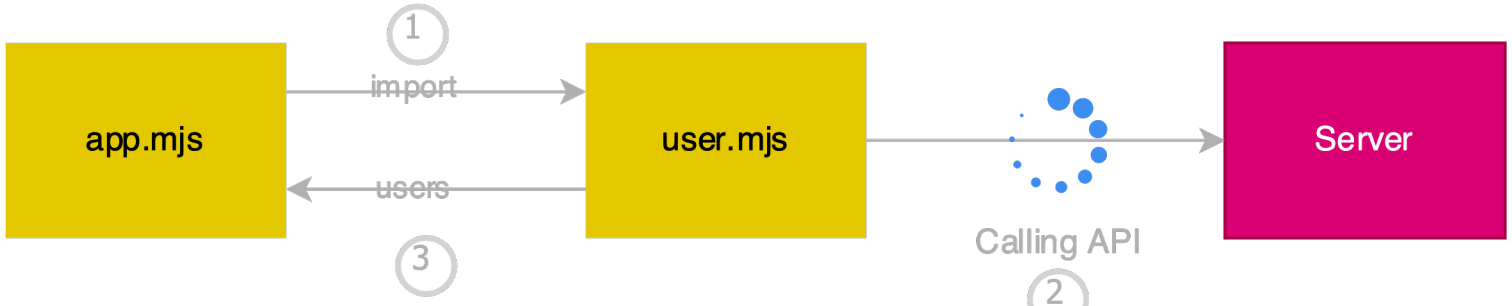
https://jsonplaceholder.typicode.com/users



Async/await

Aplicație: Să afișam informațiile primate din acest API folosind async/await. (top-level await : ES2022):

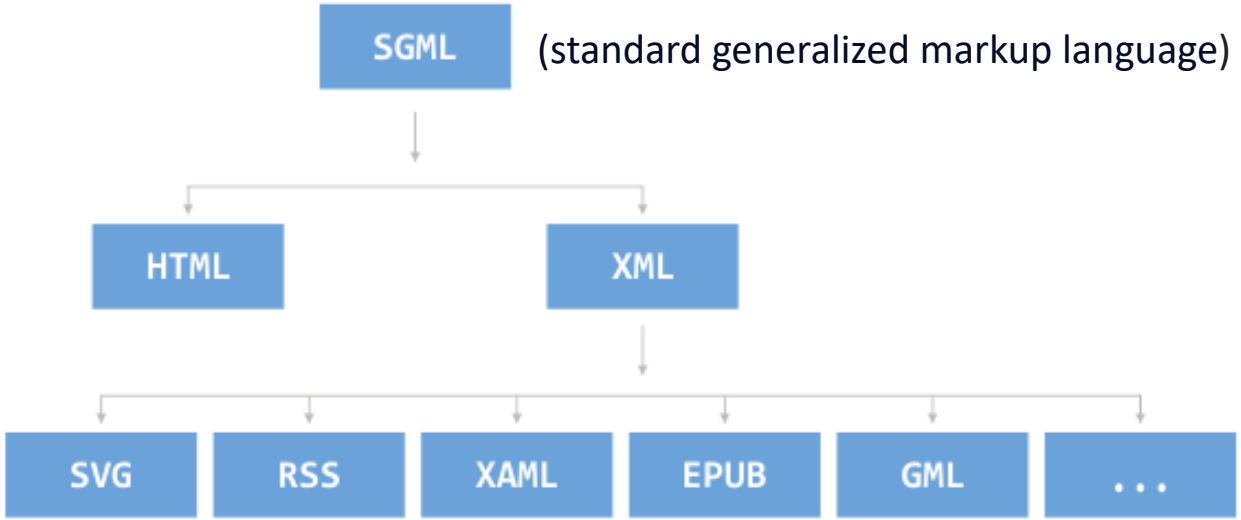
https://jsonplaceholder.typicode.com/users



SVG- Scalable Vector Graphics

SVG este prescurtarea de la *Scalable Vector Graphics*, care este, în esenţă, un limbaj descriptiv bazat pe XML folosit pentru a reprezenta grafica vectorială.

```
<rect x="10" y="10" width="200" height="100" fill="#4D4D4D"
stroke="#4F95FF" stroke-width="3" />
```



SVG- Scalable Vector Graphics

Avantaje

- ✓ dimensiune mică a fişierului, important pe web;
- ✓ capacitatea de a defini orice dimensiune;
- ✓ absenţa pixelizării, adică pierderea calităţii la mărirea graficii.

Cum folosim SVG:

- ✓ <img src="img/logo.svg" alt="shop logo">
- ✓ <object type="image/svg+xml" data="img/logo.svg">
  <img src="img/logo.png" alt="logo"></object>
- ✓ <svg xmlns="http://www.w3.org/2000/svg" width="560" height="200" version="1.1" viewBox="0 0 560 200">
  <rect id="my-rect" x="15" y="15" width="300" height="100" /></svg>

SVG- Scalable Vector Graphics

Hărți SVG:
Ap12.html
Ap13.html

Fonts

- ✓ https://fontawesome.com/search?m=free
- ✓ https://fonts.google.com/icons
- ✓ https://developers.google.com/fonts/docs/material_icons
- ✓ https://icons.getbootstrap.com/
- ✓ https://lineicons.com/icons/?type=free
- ✓ https://www.flaticon.com/animated-icons
- ✓ https://fontello.com/
- ✓ https://material.io/resources/icons/

LINKAcademy

Joc1: ***Numar puzzle*** *Ordonați numerele*

Mutari: 0

6	7	2
4	8	5
3	1	

Reset

Provocare afișați numărul de mutări

Resurse

https://api.jquery.com/jquery.ajax/

https://baconipsum.com/json-api/

https://www.w3schools.com/js/js_callback.asp

https://www.w3schools.com/js/js_async.asp

https://stackoverflow.com/questions/23714383/what-are-all-the-possible-values-for-http-content-type-header

https://www.iana.org/assignments/media-types/media-types.xhtml

Evenimente

https://events.geekle.us/

https://www.meetup.com/find/?source=EVENTS&keywords=javascript

https://www.devafterwork.ro/events

https://www.eventbrite.com/d/online/javascript/?page=1