**Udemy - The Complete JavaScript Course 2023 From Zero to Expert! – Udemy**

Browser commands:

alert("Hello World!") – V-a afisa o alerta in browser cu mesajul Hello World!

let js = 'amazing'

if (js === 'amazing') alert ('JavaScript is FUN!')

40+8+23-10 //Math

!!!HTML.CSS.JS HTML este responsabil cu continutul paginii(text, imagini, butoane si tot continutul pe care il vedem in pagina). CSS este responsabil cu prezentarea continutului(stilizarea si asezarea elementelor in pagina). JS este adevaratul limbaj de programare al internetului, lasand dezvoltatorii sa creeze efecte dinamice si interactive pe orice pagina; poate de altfel sa manipuleze continutul sau CSS-ul, incarca date din serverele la distanta, construieste intreaga aplicatie in browser.

Analogia substantivelor(HTML), adjectivelor(CSS) si a verbelor(JS) pentru a face separarea rolurilor mai usor de inteles.

Ex.: HTML(NOUNS) Elementul P este un paragraph <p></p>

CSS(ADJECTIVES) p{color:red} inseamna “textul paragrafului este rosu”. Aici rosu este adjectival descries de substantive.

JS(VERB) p.hide(); Inseamna “ascunde paragraful”

!!!JavaScript este high-level, object-oriented, multi-paradigm programing language. Explicand aceastea, programing language este un instrument care ne lasa sa scriem cod si v-a intrui calculatorul sa faca ceva. Highe-level se refera la faptul ca nu trebuie sa ne facem grihi despre complexitatea lucrurilor cum ar fi managerierea memoriei calculatorului cat timp ruleaza programul. Object-oriented, limbajul este mai ales bazat pe obiect, pentru stocarea celor mai multe date. Multi-paradigm inseamna ca este un limbaj flexibil si versatil, ce poate folosi tot felul destiluri de programe diferite, imperative si declarative. Cand lucram in JS spunem ca construim aplicatii web in front-end. Pentru a lucre in JS in avara browserului se poate folosi node.js Versiunile de JS supranumite si ES care vine de la ECMAScript: ES5🡪ES6/ES2015🡪ES7/ES2016🡪ES8/ES2017🡪ES9/ES2018🡪ES10/ES2019🡪ES11/ES2020🡪…pana in present, in acelasi ritm. ES5 a fost numit ECMAScript iar restul au fost numite modern JavaScript

!!!// Variabilele sunt containere pentru stocarea valorilor. Dupa crearea variabilelor le putem da valori, asta numindu-se initializare; ex.: let name; name”James”; Semnul = este folosit pentru a atribui o valoare variabilei. Variabilelel isi pot schimba valoare, de aceea sunt numite variabile; ex.: let age = 17; // after 1 year age=18; console.log(age);

In veriunile JavaScript mai vechi, cuvantul VAR a fost folosit in locul cuvantului LET, pentru a declara variabile. LET este calea moderna pentru a declara variabilesi are un numar de avantaje fata de VAR, cum ar fi, VAR ne lasa sa declaram variabile cu acelasi nume, in timp ce LET rezula eroare, si face codul mai sigur.. Este recomandat sa se foloseasca LET in schimbul lui VAR, cand declaram variabile.

Constantele. Sunt cazuri cand trebuie sa spunem programului ca variabilele nu-si pot schimba valorile pe tot parcursul programului. Constantele sunt similar variabilelor si sunt declarate folosind cuvantul CONST. Constantele trebuie sa aibe o valoare cand sunt declarate si ele nu-si vor schimba valoarea.

Reguli in numirea variabilelor: 1)numele variabilelor trebuie sa inceapa cu o litera, un underscore \_ , sau un semn al dolarului $. 2)Numele variabilelor nu pot continue spatiu. 3)Numele variabilelor sunt cazuri sensitive, cee ace inseamna ca, spre exemplu variabilele Name si name sunt diferite.

**9.Conectarea unui fișier JavaScript**

Inline

JavaScript inline poate fi realizat prin utilizarea etichetei Script în interiorul corpului HTML și, în loc să specificăm sursa (src=”…”) a fișierului JavaScript în eticheta Script, trebuie să scriem tot codul JavaScript în interiorul etichetei Script. .

Sintaxă:

<script>

// Cod JavaScript

</script>

External

Atributul src specifică adresa URL a unui fișier script extern.

Dacă doriți să rulați același JavaScript pe mai multe pagini dintr-un site web, ar trebui să creați un fișier JavaScript extern, în loc să scrieți același script din nou și din nou. Salvați fișierul script cu o extensie .js, apoi faceți referire la el utilizând atributul src din eticheta <script>.

Sintaxa:

<script src="*URL*">

**10.Valori si Variabile**

Valorile sunt bucati de date. Ex.: console.log(“Jonas”); console.log(23); In acest caz Jonas este valoarea. La fel si 23(la numere nu este cazul sa le punem intre ghilimele). In linia de cod: console.log(40 + 8 + 23 – 10);, 40, 8, 23 si 10 sunt valori, iar + si – sunt operatori matematici.

In linia let firstName = “Jonas” , declaram o valoare unei variabile, valoare Jonas este atribuita variabilei firstName.Pe inteles, avem o cutie numita firstName si in cutie vom pune o valoarea lui Jonas, si acum daca vrem sa folosim aceasta valoare tot c eave de facut este practice sa folosim aceasta eticheta sau cu alte cuvinte numele variabilei(firstName). De cate ori JS v-a vedea variabila firstName v-a inlocui valoarea pe care am atribuito variabilei. Fara variabile ar trebui sa schimbam manual peste tot valoarea.

Conventii si reguli pentru numirea variabilelor:

-Modul in care se numesc variabilele se numeste camelCase notation, care inseamna ca de fiecare data cand avem cuvinte multiple in numele variabilei, vom scrie primul cuvant cu litera mica si apoi toate celelalte cuvinte cu litera mare. Se poate scrie si folosindu-se \_ intre cuvinte dar nu este recomandat.

-Exista nume de variabile ilegale, cum ar fi inceperea cu o cifra sau un symbol(aceasta nu trebuie sa existe nici in interiorul cuvintelor), sau cuvantului new(cuvantul new fiind rezervat in JS), cuvantul function(el pentru a putea fi scris trebuie sa I se alature \_ sau $: \_function sau $function), nu trebuie sa inceapa cu o litera mare, ea fiind numita de catre Consola browserului Uncaught SyntaxError; Invalid or unexpected token. Chiar dava variabila name nu intoarce o eroare nu este recomandat s-o numim name.

-Trebuie sa fim siguri ca numele variabilei este descriptive. Ex.: let myFirstJob = 'Programmer';

let myCurrentJob = 'Teacher'; Este un mod in care variabila este descriptiva VS let job1 = 'programmer';        let job2 = 'teacher'; //Un mod nu tocmai recomandat sa scriem - Este un mod in care variabila nu este foarte descriptiva