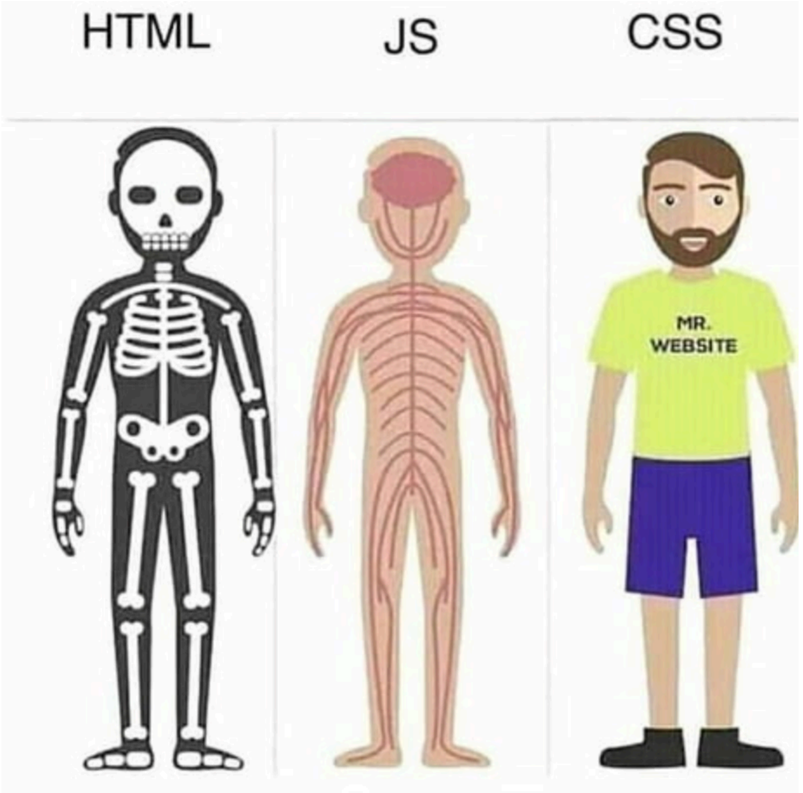
Web Development with Python Lesson 23

>>>

Ing. Patrik Dendis

STEP IT ACADEMY BRATISLAVA


HTML
JS
CSS
MR.
WEBSITE

OBSAH PREZENTÁCIE

- Javascript
- Základy JS
- Syntax
- DOM
- Event handling

OPAKOVANIE

- Popíšte HTML, CSS, JS
- Načo slúžia tabulky?
- Aké sú základné tagy pri tabuľkách a na čo slúžia?
- Čo je Jinja2?
- Ako sa prepojí Flask a Frontend?
- Ako vložíte premennú z Flasku do HTML?
- Čo je dedičnosť pri templatoch?

JavaScript

JavaScript je skriptovací jazyk, ktorý beží na strane klienta (v prehliadači). Používa sa na manipuláciu s HTML a CSS, čo umožňuje dynamické aktualizácie obsahu stránky bez potreby obnoviť stránku.

Bol pôvodne navrhnutý pre manipuláciu s HTML dokumentmi v prehliadači, avšak jeho použitie sa rozšírilo aj na server-side (napríklad pomocou Node.js).

POUŽITIE JS

- **Interaktivita:** JavaScript umožňuje vytvárať interaktívne webové stránky, napríklad reagujúce na akcie užívateľa (kliknutia, pohyby myši).
- **Dynamický obsah:** Umožňuje meniť obsah webovej stránky bez jej znovunačítania (napr. dynamické aktualizácie údajov).
- **Validácia formulárov:** Pomáha v validácii údajov zadaných používateľom ešte pred ich odoslaním na server.
- **Animácie:** JavaScript sa používa na vytváranie animácií a grafických efektov.

SYNTAX JS

- **Deklarácia:** Na deklaráciu premenných sa používajú kľúčové slová *var*, *let* a *const*
  - *var*: Má funkčný alebo globálny rozsah a môže byť znovu deklarovaná.
  - *let*: Má blokový rozsah a nie je možné ju znovu deklarovať v rovnakom rozsahu.
  - *const*: Deklaruje konštanty, ktoré nemôžu byť premenené po ich priradení.

```
1    var name = "John";
2    let age = 30;
3    const PI = 3.14;
```

SYNTAX JS

- **Datové typy:**
  - **Primitívne typy**: *String, Number, Boolean, Null, Undefined, Symbol*.
  - **Referenčné typy**: *Objects (vrátane polí a funkcií).*

```
1  let text = "Hello";
2  let number = 123;
3  let isTrue = true;
4  let person = {firstName: "John", lastName: "Doe"};
5  let numbers = [1, 2, 3];
6  let func = function() { return "Hello World"; };
```

SYNTAX JS

- **Operátory:** Rovnako ako v Pythone

```
1    let a = 5;
2    let b = 10;
3    let result = a + b; // 15
4    let isEqual = (a === b); // false
```

SYNTAX JS

- **Podmienky:** Podobne ako v Pythone

```
let score = 85;

if (score >= 90) {
  console.log("Grade: A");
} else if (score >= 80) {
  console.log("Grade: B");
} else if (score >= 70) {
  console.log("Grade: C");
} else {
  console.log("Grade: F");
}
```

SYNTAX JS

- **Funkcie:** Podobne ako v Pythone
- **Deklarácia a volanie funkcií:**
  - Funkcie môžu byť deklarované pomocou kľúčového slova function.
  - Funkčné výrazy môžu byť priradené premenným.

```
1   function greet(name) {
2     return "Hello, " + name;
3   }
4
5   let message = greet( name: "Alice");
6   console.log(message); // "Hello, Alice"
7
8   let sum = function(a, b) {
9     return a + b;
10  };
11
12  console.log(sum( a: 5,  b: 3)); // 8
```

SYNTAX JS

- **Cykly:** Podobne ako v Pythone

```
1   // For cyklus
2   for (let i = 0; i < 5; i++) {
3     console.log(i);
4   }
5
6   // While cyklus
7   let j = 0;
8   while (j < 5) {
9     console.log(j);
10    j++;
11  }
```

HTML + JS

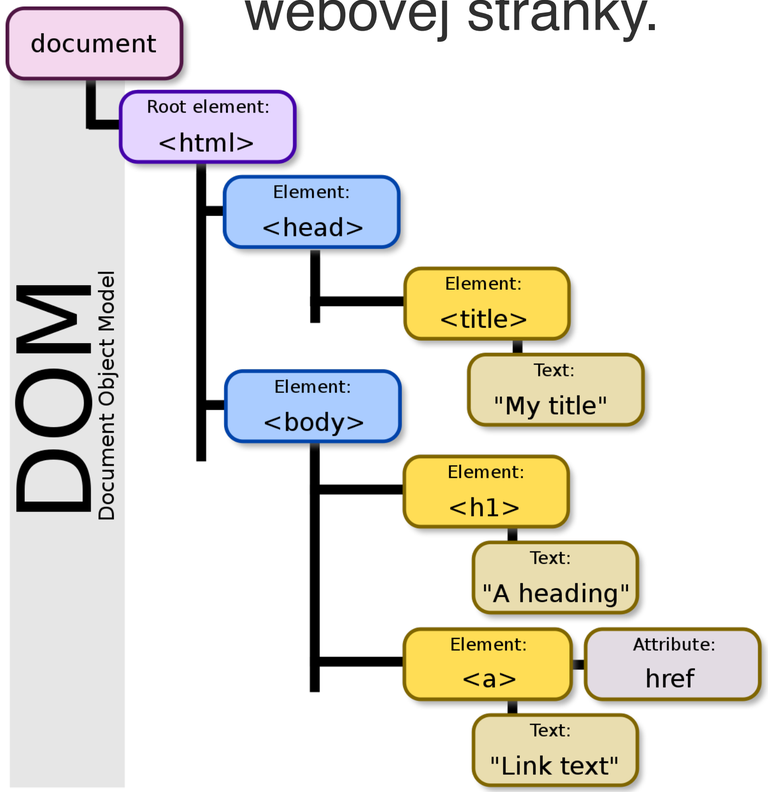
- Vo Flask frameworku sa JS súbory nachádzajú v adresári static

```
1  <!DOCTYPE html>
2  <html lang="sk">
3  <head>
4      <meta charset="UTF-8">
5      <title>Nejaka stranka</title>
6      <link rel="stylesheet" href="../static/styles.css">
7      <script src="../static/script.js"></script>
8  </head>
9  <body>
10     <header>
11         <h1>Nadpis stranky</h1>
```

DOM

DOM (Document Object Model) je programové rozhranie pre HTML a XML dokumenty. Predstavuje štruktúru dokumentu ako stromovú štruktúru, kde každý uzol predstavuje časť dokumentu (element, atribút, text).

Umožňuje JavaScriptu dynamicky manipulovať s obsahom, štruktúrou a štýlom webovej stránky.



MANIPULÁCIA S DOM

- **Metódy na výber elementov:**
  - *document.getElementById(id)*: Vyberie element s daným ID.
  - *document.querySelector(selector)*: Vyberie prvý element, ktorý zodpovedá CSS selektoru.
  - *document.querySelectorAll(selector)*: Vyberie všetky elementy, ktoré zodpovedajú CSS selektoru.

```
1  let elementById = document.getElementById( elementId: "myId");
2  let firstElement = document.querySelector( selectors: ".myClass");
3  let allElements = document.querySelectorAll( selectors: "p");
```

MANIPULÁCIA S DOM

- **Manipulácia s obsahom:**
  - *innerHTML*: Nastaví alebo vráti HTML obsah elementu.
  - *textContent*: Nastaví alebo vráti textový obsah elementu.

```
let myElement = document.getElementById( elementId: "myElement");
let myElement2 = document.getElementById( elementId: "myElement2");
myElement.textContent = "Nový textový obsah";
myElement2.innerHTML = "<h1>Test</h1>";
```

MANIPULÁCIA S DOM

- **Práca s atribútmi elementov:** Môže meniť, pridávať alebo odstraňovať atribúty elementov.
  - *getAttribute(attr)*: Získa hodnotu atribútu.
  - *setAttribute(attr, value)*: Nastaví hodnotu atribútu.
  - *removeAttribute(attr)*: Odstráni atribút.

```
let link = document.querySelector( selectors: "a");
link.setAttribute( qualifiedName: "href",  value: "https://example.com");
let hrefValue = link.getAttribute( qualifiedName: "href");
link.removeAttribute( qualifiedName: "target");
```

MANIPULÁCIA S DOM

- **Pridávanie a odstraňovanie elementov:**
  - *createElement(tag):* Vytvorí nový element s daným tagom.
  - *appendChild(node):* Pridá nový uzol ako posledné dieťa elementu.
  - *removeChild(node):* Odstráni konkrétny uzol z elementu.

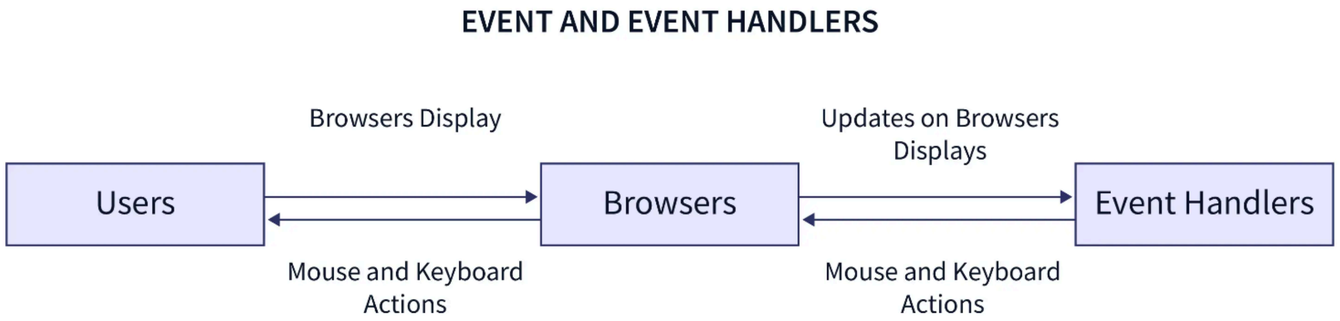
```
let newElement = document.createElement( tagName: "div");
newElement.textContent = "Toto je nový element";
document.body.appendChild(newElement);

let parent = document.getElementById( elementId: "parentElement");
let child = document.getElementById( elementId: "childElement");
parent.removeChild(child);
```

EVENT HANDLING

Udalosť (event) je akcia alebo výskyt, ktorý sa stane v prehliadači, ako napríklad kliknutie myšou, stlačenie klávesy, načítanie stránky alebo odoslanie formulára.

Udalosti umožňujú webovým stránkam reagovať na interakcie používateľov, čím sa stránky stávajú interaktívnymi.



EVENT HANDLING

- **Pridávanie Event Listenerov:**
  - a*ddEventListener:* Používa sa na pripojenie event listenerov k elementom.

```
document.getElementById("myElement").addEventListener("click", function() {
  alert("Button was clicked!");
});
```

EVENT HANDLING

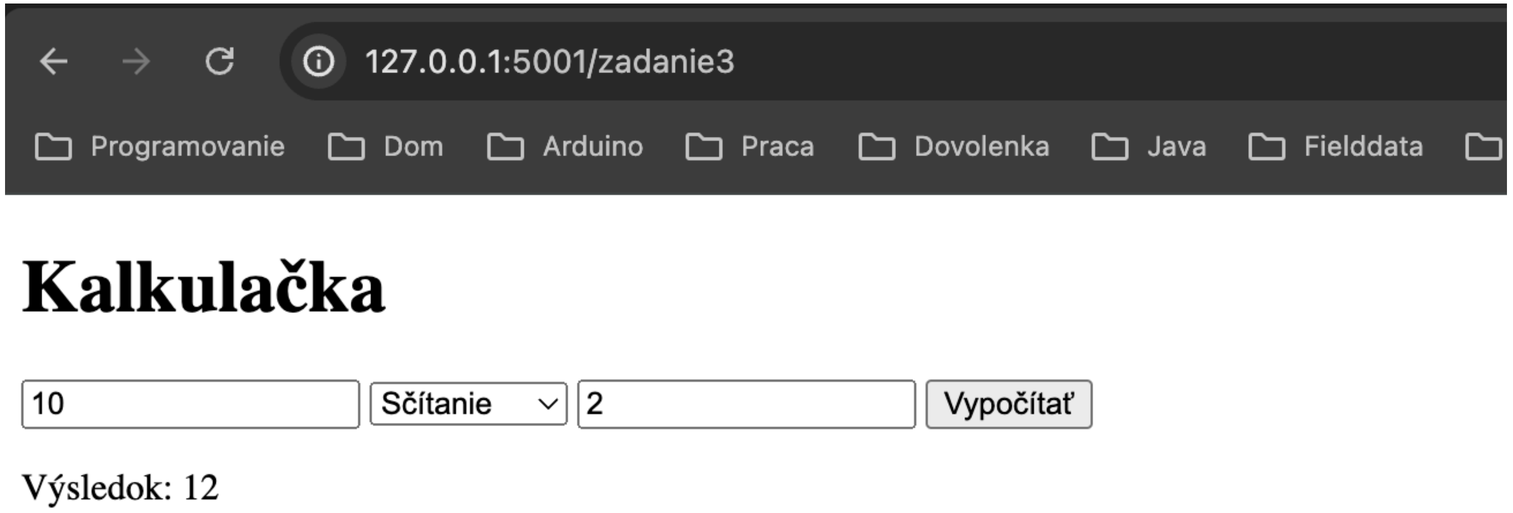
- **Typy udalostí:**
  - *Mouse Events:* 'click', 'dblclick', 'mousedown', 'mouseup', 'mouseover', 'mouseout', ‘mousemove'
  - *Keyboard Events:* 'keydown', 'keyup', ‘keypress'
  - *Form Events:* 'submit', 'reset', 'focus', 'blur', ‘change'
  - *Window Events:* 'load', 'resize', 'scroll', ‘unload'

```
window.addEventListener( type: "load",  listener: function() : void  {
  console.log("Page is fully loaded");
});

document.getElementById( elementId: "myInput").addEventListener( type: "focus",  listener: function() : void  {
  this.style.backgroundColor = "yellow";
});
```

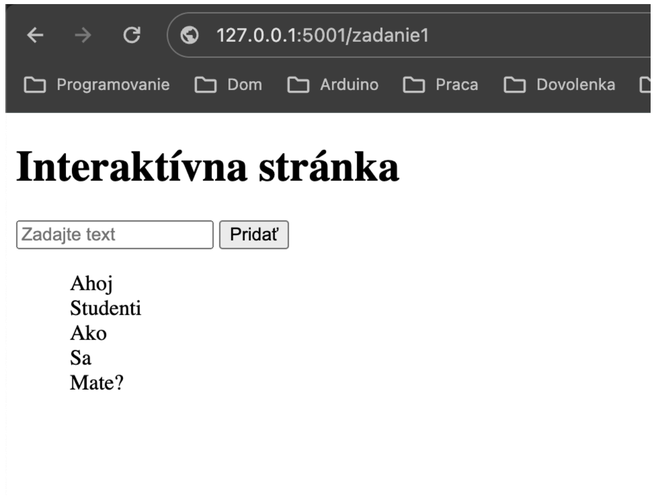
ZADANIE

Vytvorte jednoduchú kalkulačku, ktorá umožňuje používateľovi zadávať dve čísla a vybrať si matematickú operáciu (sčítanie, odčítanie, násobenie, delenie). Výsledok sa zobrazí po kliknutí na tlačidlo "Vypočítať".



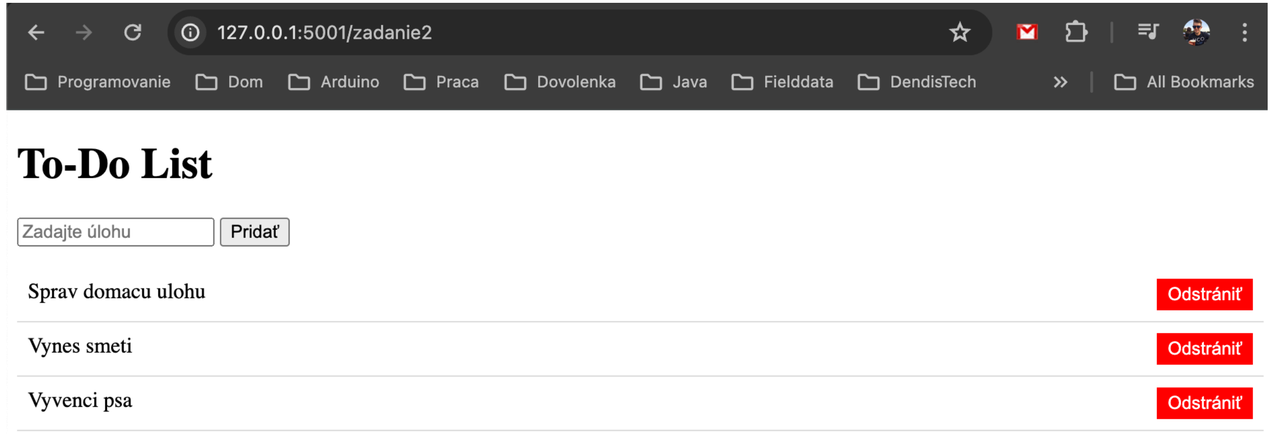
ZADANIE

Vytvorte jednoduchú webovú stránku, ktorá obsahuje formulár s textovým polo a tlačidlom. Po kliknutí na tlačidlo by sa mal text z textového políčka zobraziť v zozname pod formulárom. Ak sa klikne na text v zozname, mal by sa tento text odstrániť.



ZADANIE

Vytvorte interaktívny to-do list, kde môžu používatelia pridávať nové úlohy, označovať ich ako dokončené kliknutím na nich a odstraňovať ich kliknutím na tlačidlo "Odstrániť" vedľa každej úlohy.



HTML/CSS FRAMEWORK

HTML frameworky sú kolekcie predpripravených HTML, CSS a JavaScript súborov, ktoré poskytujú základnú štruktúru a štýl pre vývoj webových stránok. Tieto frameworky uľahčujú a zrýchľujú proces vývoja tým, že poskytujú hotové komponenty, ako sú navigačné lišty, tlačidlá, formuláre, modálne okná a ďalšie.

HTML/CSS FRAMEWORK

- **Rýchlejší Vývoj:** Frameworky poskytujú predpripravené komponenty, ktoré môžu vývojári okamžite použiť, čím sa zrýchľuje proces vývoja.
- **Responzivita:** Väčšina moderných frameworkov je navrhnutá s ohľadom na mobilné zariadenia a responzivitu, čo znamená, že stránky sa automaticky prispôsobia rôznym veľkostiam obrazoviek.
- **Štandardizácia:** Frameworky pomáhajú udržať konzistentný dizajn a štruktúru naprieč celým projektom.
- **Komunita a Podpora:** Populárne frameworky majú veľkú komunitu, ktorá poskytuje dokumentáciu, návody a podporu.

BOOTSTRAP

Bootstrap je open-source framework pre vývoj webových stránok, ktorý obsahuje HTML, CSS a JavaScript šablóny pre typické webové komponenty. Bol pôvodne vyvinutý vývojármi v Twitteri a teraz je jedným z najpopulárnejších frameworkov na svete.

https://getbootstrap.com/

ĎAKUJEM ZA POZORNOSŤ