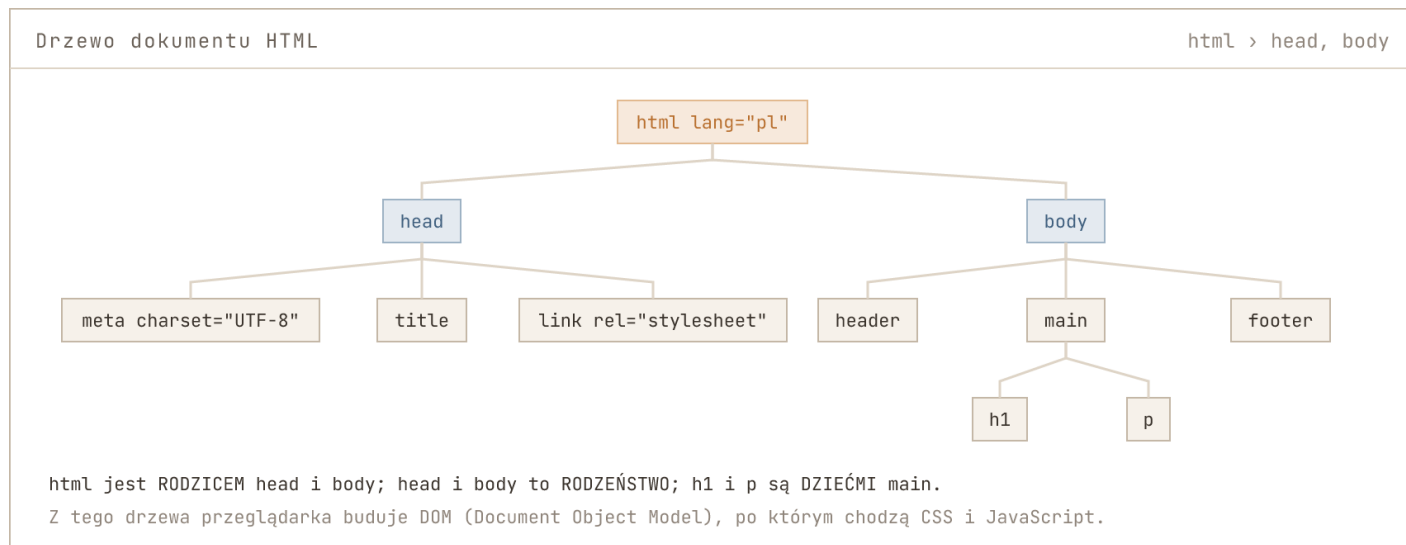


JavaScript i DOM: zmiana strony z poziomu skryptu (ściągą)

Drzewo dokumentu



Pobieranie elementu

Zapis	Zwraca
document.getElementById("x")	jeden element o identyfikatorze x, bez #
document.getElementsByTagName("p")	kolekcję wszystkich akapitów
document.getElementsByClassName("c")	kolekcję wszystkich elementów klasy c
document.querySelector("#x")	pierwszy element pasujący do selektora CSS
document.querySelectorAll(".c")	kolekcję wszystkich pasujących
kolekcja[0], kolekcja.length	element kolekcji (numeracja od zera), jej rozmiar

Zmiana elementu

Zapis	Skutek
el.innerHTML = "3"	treść ze znacznikami, znaczniki działają
el.textContent = "3"	czysty tekst, znaczniki widoczne dosłownie
el.setAttribute("src", "a.png")	ustawienie wartości atrybutu
el.getAttribute("alt")	odczyt wartości atrybutu
pole.value	tekst wpisany w pole formularza
el.style.color = "blue"	kolor tekstu
el.style.backgroundColor = "red"	kolor tła, nazwa CSS bez minusa
el.style.display = "none"	ukrycie elementu razem z jego miejscem
el.className = "a b"	podmiana całego atrybutu class
el.classList.add("a")	dołożenie klasy, bez ruszania reszty
rodzic.appendChild(el)	dopięcie węzła jako ostatniego dziecka
rodzic.removeChild(el)	usunięcie dziecka, wywołane na rodzicu

Lista z tablicy

```
const lista = document.getElementById("produkty");
const nazwy = ["Monitor 24", "Klawiatura", "Mysz"];
for (let i = 0; i < nazwy.length; i++) {
  const pozycja = document.createElement("li");
  pozycja.textContent = nazwy[i];
  lista.appendChild(pozycja);
}
```

Wynik: trzy w liście produkty.

Pułapki

- getElementById w liczbie pojedynczej, getElementsByTagName i getElementsByClassName w mnogiej; zapisy getElementByID i getElementByTagName nie istnieją.
- Kolekcja nie jest elementem: bez [0] albo pętli przypisanie nic nie zmienia.
- Skrypt w <head> dostaje null, bo <body> jeszcze nie powstało.
- style.background-color to błąd składni, style.font-color nie istnieje: kolor tekstu to color.
- Pole input nie ma treści: wpisany tekst leży w value, nie w innerHTML.
- removeChild wywołuje się na rodzicu i dostaje dziecko, nie odwrotnie.

Egzamin praktyczny: plik dołącza <script src="skrypt.js"></script> przed </body>; kolory strony zmienia document.body.style.backgroundColor i document.body.style.color, wynik wstawia innerHTML, a listę z tablicy pętla z createElement i appendChild.