**Lezione 27/06/2024**

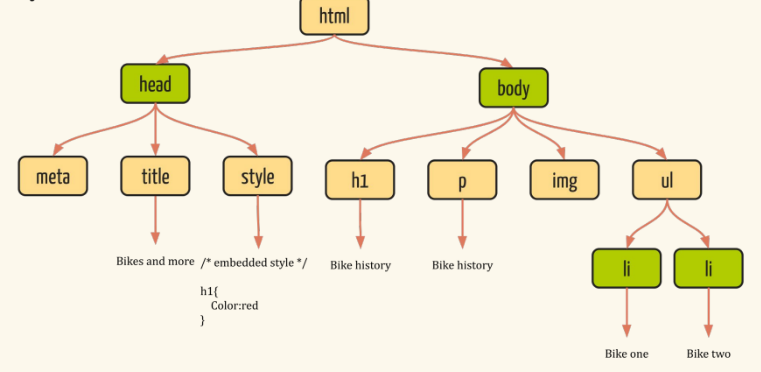
**web developer**

**JavaScript**

**DOM (Document Object Model)**

mappa documento html, struttura logica accessibile e manipolabile dal JS consentendo una modifica (aggiunta, alterazione e roba figa) degli elementi della pagina HTML dal JS

cit unità 5 html



**Il DOM si accede da Browser**

si **conserva come variabile** e si manipolano gli elementi all’interno del nodo

(cambio attributi, modifica stile, dare un nuovo html interno e aggiungere altri nodi DOM)

**Cosa è un nodo?** è un oggetto in JS che rappresenta un elemento HTML

<https://www.w3schools.com/js/js_htmldom_document.asp>

<https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model>

**Metodi di accesso**

*document.getElementById(id);* (cerca nodi DOM usando id=””)

*document.getElementsByClassName(className);* (cerca nodi DOM usando class=””)

*document.getElementsByTagName(tagName);* (cerca nodi DOM usando <tag>)

*document.querySelector(cssQuery);* (cerca il primo elemento per un selettore css)

*document.querySelectorAll(cssQuery);* (cerca il tutti elementi per un selettore css)

***nb:*** questi metodi di accesso ritornano dei valori

in blu: ritornano un valore (Element o Null)

in arancione: ritornano una collezione di valori stile array (HTMLCollection, elementi HTML, piena o vuota )

**nb:** document.querySelectorAll(cssQuery) ritorna una NodeList (oltre a essere live come HTMLCollection, è anche static e supporta metodo foreach, itera elementi + facilmente dell’HTMLCollection)

**NB: SEMBRA** che modifichiamol’HTML, ma in realtà con JS **si modifica la presentazione**

esempi di metodi di accesso

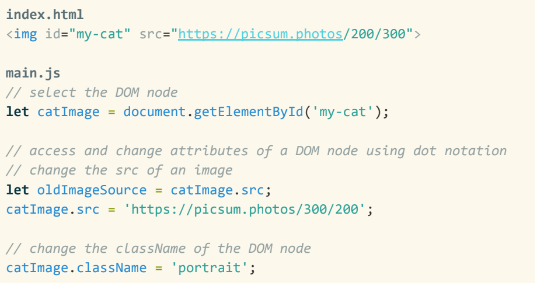


oppure

****

**Manipolazione del DOM**

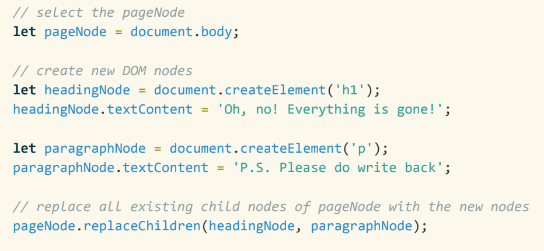
esempio manipolazione tag HTML con JS



esempio manipolazione selettori CSS con JS



**Manipolazione contenuto**

****

**NB:** è possibile modificare testo con i nodi DOM attraverso **textContent** oppure **innerHTML**

**textContent:** funziona per browser recenti, + veloce xché non richiede analisi dell’html, + sicuro xché previene esecuzione di codice

**innerHTML:** compatibilità vecchi browser, +potente xché manipola direttamente il codice html ma +pericoloso xché sensibili ad attacchi XSS (iniettare script dannosi)

occhio: si modifica il contenuto di un elemento HTML, NON l’HTML originale!!!

**Creazione di nodi**

*document.createElement('tagName');*

*document.createTextNode('text');*

*document.appendChild(childToAppend);*



**NB: Element.append()**

è un metodo usato per inserire un set di nodi o oggetti DOMString dopo l’ultimo figlio di Element, permette di appendere molteplici elementi alla volta

