Hodina 27.3. JS DOM

Na probuzení  
začnu zeširoka – koukala jsem na minulá videa a dnes bude náročná hodina,   
zezačátku si zopakujeme, co znáte, abychom na tom mohli stavět

kvíz datové typy

otázky???

Opakování   
dosud jste se učili hlavně HTML a CSS – tvořili jsme statickou stránku   
reagovala možná na změnu okna zalomením textu, ale to dělá CSS

HTML je obsah, CSS je styl – napojení CSS na HTML,

Se Sergejem jste si zkoušeli konzoli a padla otázka – ono to ve webu nic neudělá  
JS dokáže vzít ten HTML obsah, upravit ho měnit, přidávat dynamicky   
reagovat na akci uživatele (scrollování, výběr) a ukázat třeba část obsahu nebo animace  
<https://www.youtube.com/watch?v=vQ0ufLz4wgY&t=192s>

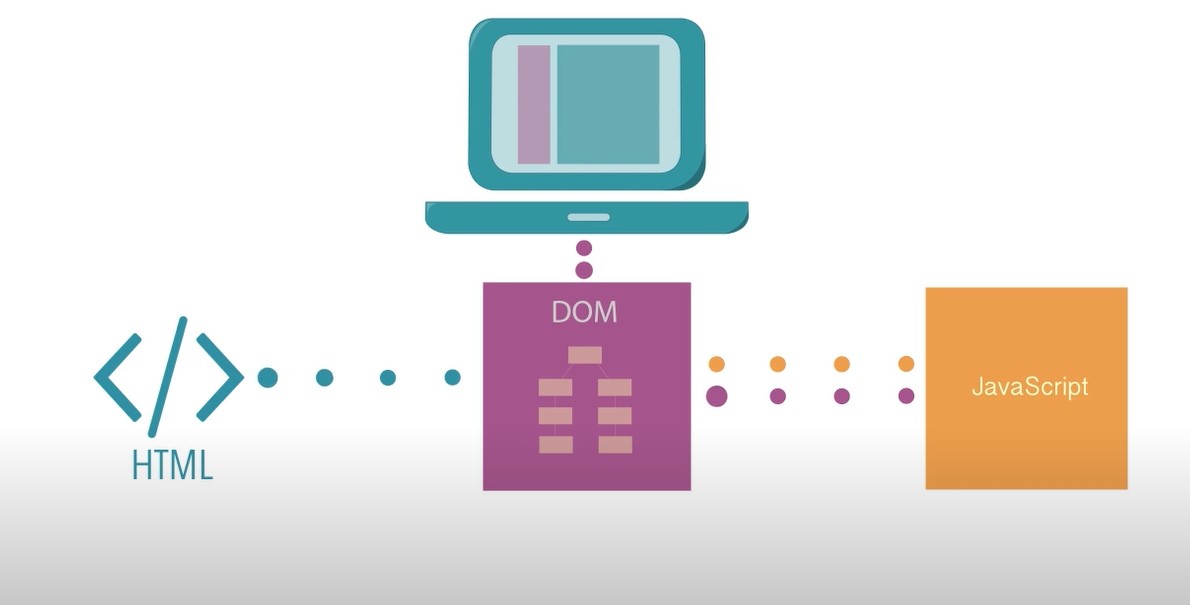
Pokud už na stránce budeme používat javascript - programování,

* vymýšlíme nějakou logiku, jak se stránka bude chovat a co bude dělat.

JS byl spíš zvlášť, do stránky jste vypsali text pomocí document.write, ale teď to spojíme   
  
než začneme kódovat, tak je potřeba si vysvětlit ještě jeden termín   
DOM   
Document object model – viruální reprezentace naší stránky jako strom objektů  
pomocí DOMu můžeme tvořit dokument, navigovat v jeho struktuře, přidávat, modifikovat nebo mazat elementy nebo obsah

Používáme ho na interakci s webovými stránkami,

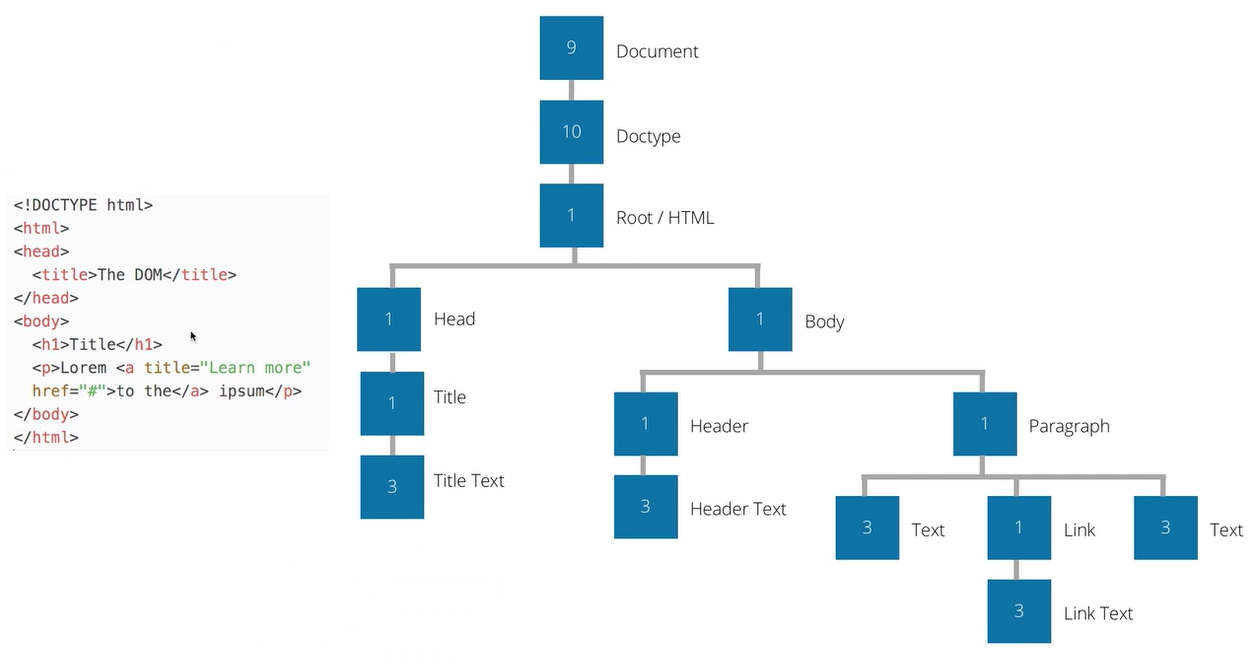
Přidávat, odebírat, měnit obsah html   
  
Document – wepage – většinouo myslíme HTML dokuemnt   
Objekt – každý element je objekt (head, body, header, paragraph)

A my už víme, že s objekty se dá skvěle pracovat   
Model – ukazuje nám, jak jsou tyto objekty mezi sebou rozmístěny, jak na sebe navazují   
můžeme měnit elementy, obsah, atributy  


CO, PROČ, JAK, DEMO   
co vidíme na stránce závisí na DOMu, ne jen na HTML  
Reprezentuje strukturu HTML jako tree structure   
dom viewer online   
Document – stránka –

Object – elements

Model – struktura, layout, rozložení   
používá se k tomu, abychom se dostali k obsahu stránky a mohli ho číst, přistoupit k němu a měnit – strukturu, styl, obsah

  
Objektová reprezentace stránky, která může nýt modifikovaná skriptovacím jazykem, jako je JS   
můžeme přistoupit k elementům pomocí querySelector  
  
DEVTOOLS – vidíme tu DOM reprezentaci a můžeme si prohlížet elementy – to znáte

Do ted se používala console.log   
document.write

* JS buď můžeme psát do našeho souboru index.js . musíme to propojit v body
* Nebo do konzole – zmizí po refreshi   
  otázky???

HTML, CSS

Dnes se naučíme ty komplexnější výrazy, které se stránkou interagují   
Javascript obsahuje spoustu užitečných funkcí – Math.random, Math.floor etc.

-length

.trim .slice …

My si ukážeme další document   
odpoledne se naučíte tvořit vlastní fuknce, ale stále budeme používat přednastavené v JS   
pauza před cvičením

  
budeme dělat nejenom věci v konzoli   
script je na konci, aby mohl interagovat s dokumentem   
holky kódují společně s tím   
html si spustí

Používaly jste si document.write (nadřazený objekt, reprezentuje stránku)   
naučíme se užitečné věci s tím   
 proměnné – kam se dá ukládat   
můžeme potom tyto dál používat

Atribut defer v index.js

S Martinem jste psali do stránky pomocí document.write   
jestli vám to dává smysl, nebo jestli můžu něco líp vysvětlit

Dokument je objekt, který má nějaké vlastnosti a metody   
jelikož je to jedinečný, specifický typ – DOM, tak má na sobě nějaké vlastnosti a metody – přes tečku se k nim dostaneme

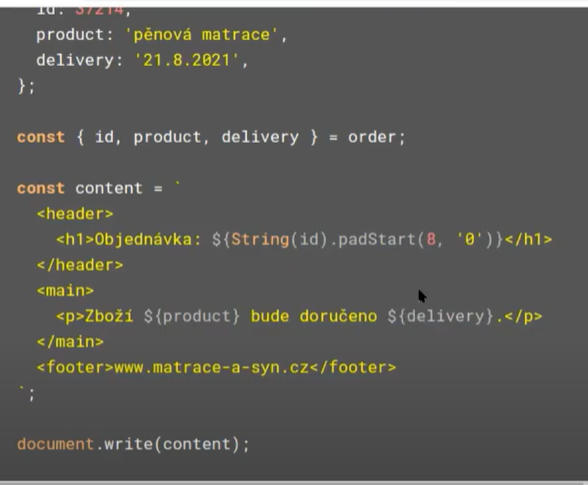
Vlastnost vs metoda v objektu

Vlastnost – obsah, obsahuje hodnotu – tu nám vrací

Metoda – funkce – něco to dělá , VOLá se závorkami

Psaní uvnitř index.js vs v conzoli

Interpolace řetzců – to budete používat pořád

Dostali jste se i celkem hluboko do něčeho, co v budoucnu poznáte jako používání props.   


Naučili jste se BOOLEAN hodnoty   
podmínky

Operátory

Složité podmínky, if, else