

16 paskaita



Code Academy

Programuok savo ateitį!

1. Naršyklės aplinka
2. DOM

Pradžioje **JavaScript** buvo skirtas tik interneto naršyklėms.

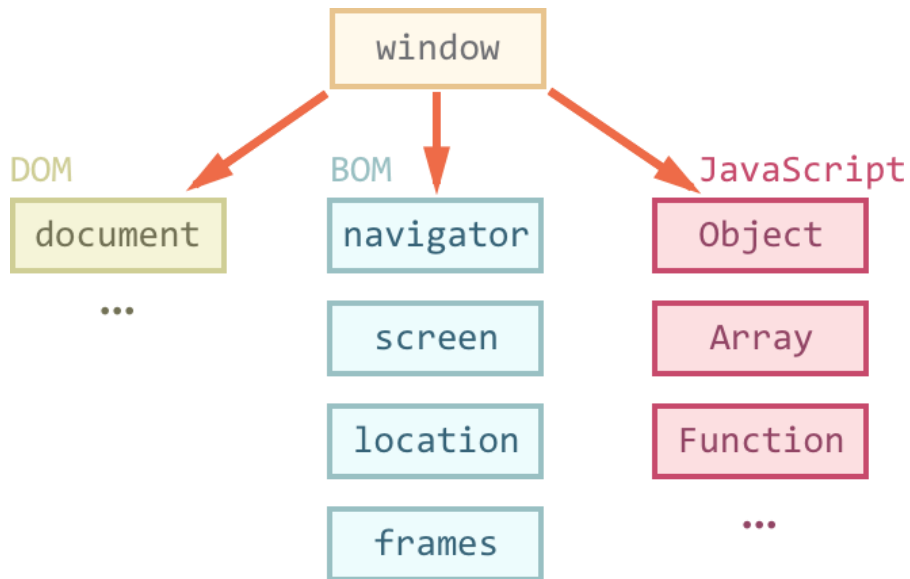
Dabar JavaScript galime sutikti skirtinguose aplinkose
- web serveris, mikrokompiuteris..

Ši aplinka vadinama ***host environment***.

Aplinka, kurioje veikia JavaScript mums suteikia jai specifinių objektų ir funkcijų.

Naršyklės leidžia kontroliuoti web puslapius, Node.js įgalina mus valdyti serverio funkcionalumą.

Štai ką mes turime kai JS veikia naršyklėje:



window - pagrindinis (root) objektas, kuris:

1. Veikia kaip pagrindinis objektas visam JS kodui
2. Atstovauja naršyklės langą ir suteikia mums metodus kaip jį valdyti.

DOM

Document object model

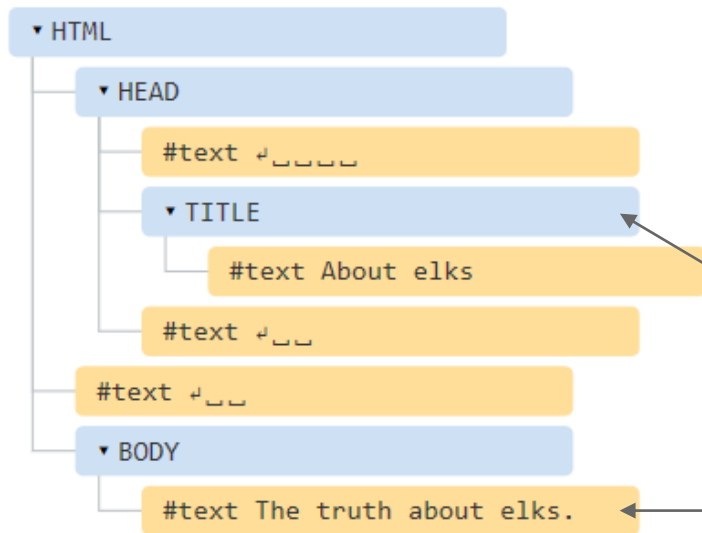
Dokumento objekto modelis:

- » programuojama **sąsaja** skirta HTML/XML dokumentams (sujungia puslapius su JS kalba)
- » ***document*** objektas leidžia mums keisti/kurti bet ką puslapyje;

DOM medis (tree)

- » Kiekviena **HTML** žymė yra objektas.
- » Žymės viduje esanti kita žymė vadinama vaiku (children).
- » Tekstas žymės viduje - taip pat objektas.


```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>About elks</title>
5 </head>
6 <body>
7   The truth about elks.
8 </body>
9 </html>
```



DOM atvaizduoja HTML medžio struktūros principu.

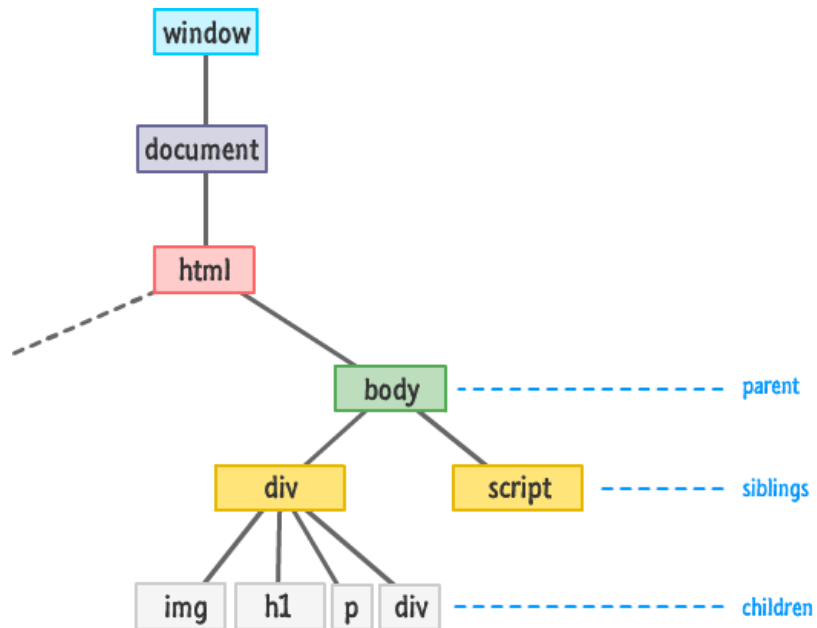
Element node

Text node

- » Jei naršyklė aptinka neteisingą HTML - ji automatiškai pataiso kurdama DOM medį.
- » Tag'ai automatiškai uždaromi, lentelėse atsiranda papildomas ***tbody*** elementas ir pan.

Inspect DOM

- » `<html>` = **`document.documentElement`**
- » `<body>` = **`document.body`**
- » `<head>` = **`document.head`**



<code 1>

Jei turime **DOM node**, galime surasti: *parentNode*,
childNodes, *firstChild*, *lastChild*, *previousSibling*,
nextSibling

Jei turime DOM **elemento** node, galime surasti:
parentElement, *children*, *firstElementChild*,
lastElementChild, *previousElementSibling*,
nextElementSibling

DOM elementų paieška:

- » **getElementById** arba **Id**
- » **getElementsByTagName**
- » **querySelector**
- » **querySelectorAll**

<code 2>

DOM elementų kūrimas:

» `document.createElement(tag)`

Elementų įterpimas į dokumentą:

- » parentElem.**appendChild**(node)
- » parentElem.**insertBefore**(node, nextSibling)
- » prepend/append/before/after
- » insertAdjacentHTML/Text/Element

<http://javascript.info/modifying-document>

code 3