

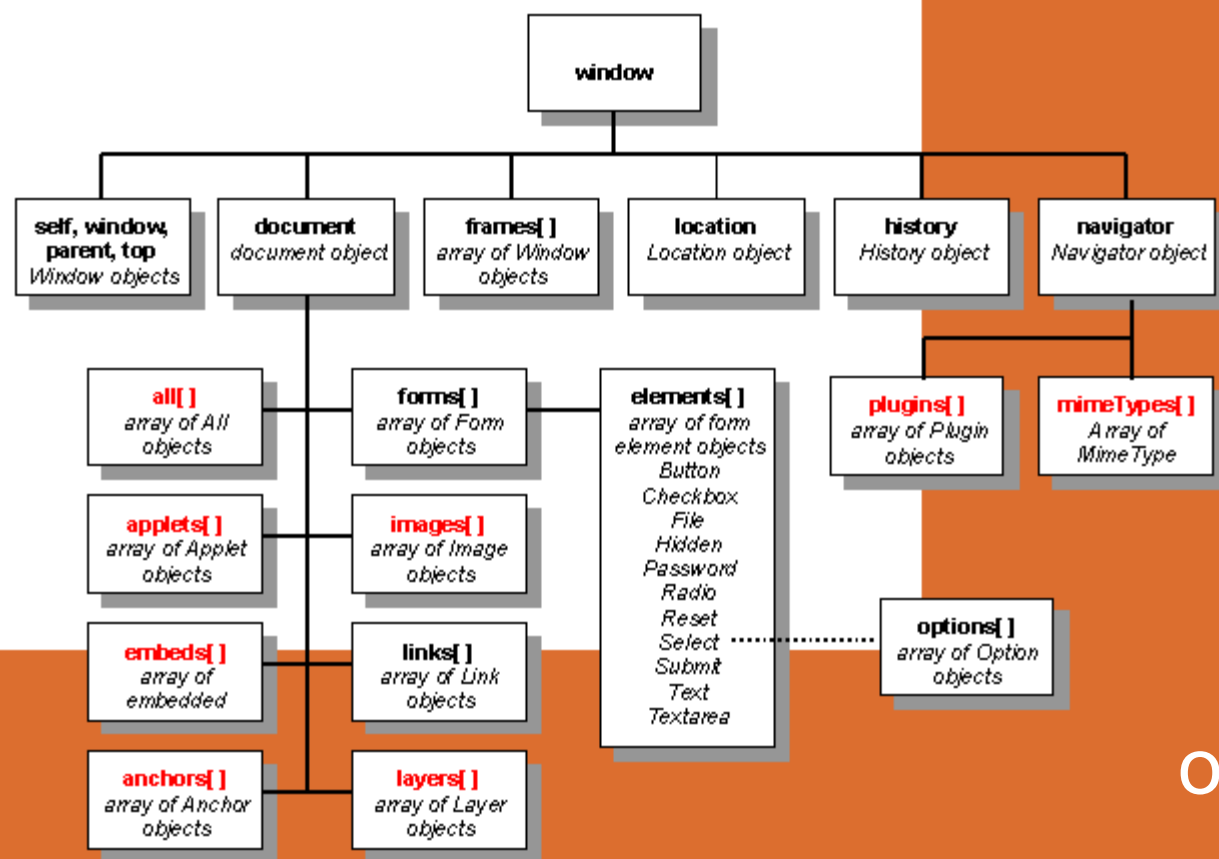


WEB разработка

JavaScript
DOM

Съдържание

- Вградени обекти в браузъра
- DOM -
 - DOM API
 - Селектиране на DOM елементи
 - Node, nodes, nodeLists(Live&Static)



Вградени
обекти в браузъра

Вградени обекти в браузъра

Браузърът ни предоставя данни чрез -

window

Елементът, на върха на DOM дървото

Представява порзореца на браузъра

document

Съдържа информация за текущия документ, зареден в браузъра

screen

Съдържа информация за потребителските настройки за визуализация

navigator

Съдържа информация за браузъра

Вградени обекти в браузъра

document object

Съдържа масиви с данни за текущата заредена в браузъра страница.

```
document.links[0].href = "yahoo.com";  
document.write("This is some <b>bold text</b>");
```

Вградени обекти в браузъра - demo

`document.location`

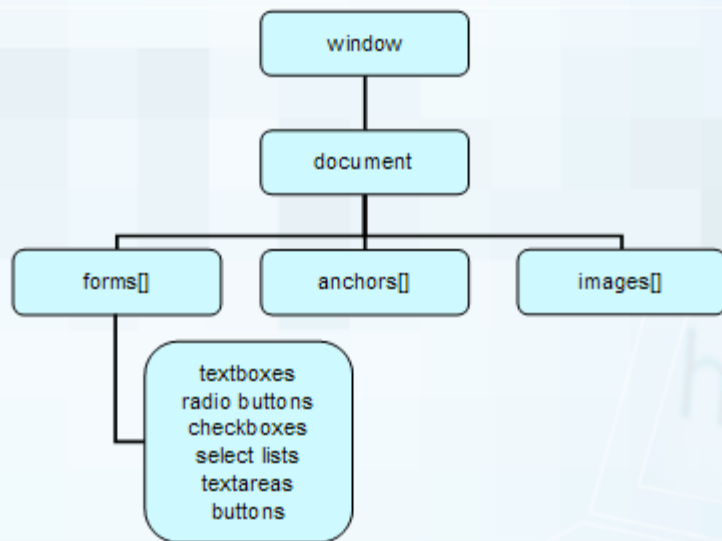
Текущия URL, зареден в браузъра или за пренасочване на браузъра

```
document.location = "http://www.yahoo.com";
```

DOM

*Document
Object
Model*

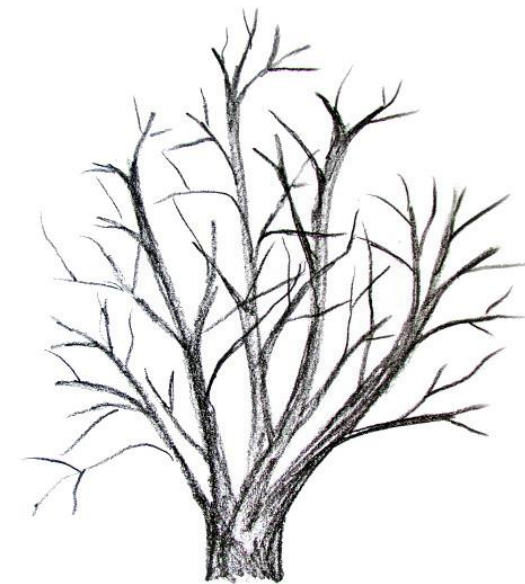
The HTML DOM



DOM

DOM - document object model, MDN

- DOM - структура на HTML
 - Представяне на HTML документа като **DOM - дърво**
 - Състоящо се от елементи, които имат деца-елементи,
 - Които на свой ред могат да имат деца елементи и т.н.
- DOM - съдържа [API](#) за обхождане/променяне на **DOM - дървото**
 - Позволява ни да променяме HTML документа динамично -
 - да махаме или добавяме нови елементи
 - Да променяме начина, по който изглеждат елементите
 - Добавяме или премахваме атрибути на HTML елементи



Как взаимодействаме с **DOM - дървото** ? Как го променяме?

- Чрез JavaScript

DOM API - demo

Всеки HTML елемент, чрез DOM API може да бъде достъпен като JS обект.

`document.documentElement` is the `<html>` element

`document.body` is the `<body>` element of the page

DOM API - *info*

document е специален обект -

Той представлява началото - входната точка, от която започва работа DOM API.

document е

корена/root на

DOM - дървото.

DOM API

Ако всеки HTML елемент е JS обект, то
Атрибутите на HTML елементите са свойствата/properties на JS обекта.

Атрибути<->Свойства

`id`, `className`, `style`, `value`, и т.н.

DOM API - demo

innerHTML

Съдържа стринг - съдържанието на елемента, без самия елемент

outerHTML

Стринг - съдържанието на елемента и елемента

innerText / textContent

Стринг - текста на елемента, беза таговете

[innerText vs. textContent](#)

DOM API - *info*

Всеки HTML елемент има съответен интерфейс - набор от методи и свойства в DOM API. Част от тези свойства наследяват от основни [HTMLElement](#) обект, други са специфични за елементите.

HTMLAnchorElement има **href** свойство

HTMLImageElement има **src** свойство

HTMLInputElement има **value** свойство

**Селектиране
на
HTML
елемент/и**

Selecting HTML elements

HTML елементите могат да бъдат достъпни и запазени в променливи с помощта на DOM API

- Избиране на един елемент или защо в страницата трябва да имаме само едно id с дадена стойност.

```
var email_2 = document.getElementById('email_address2');
```

```
var span = document.querySelector('#email_form span');///първия срещнат такъв елемент
```

Selecting HTML elements

- Селектиране на група елементи

```
var inputs = document.getElementsByTagName('input');  
var email = document.getElementsByName('email_address2');  
var classGroup = document.getElementsByClassName('className');  
var form = document.querySelectorAll('#email_form input');//всички срещнати
```

*Какво съхранява всеки един от примерите?**

Как открива елементите?

Selecting HTML elements

- Използване на дадена група от елементи

```
var links = document.links; // връща колекция от елементи
```

```
var forms = document.forms; // връща колекция от елементи
```

```
var form = document.forms[0]; // посочваме конкретен елемент
```

Използване на `getElementsBy` методи

getElementsByTagName Methods

- DOM API съдържа методи за селектиране на елементи, базирани на някои характеристики
 - **getElementById(id)**:
 - Резултатът е един елемент или null

getElementsByTagName Methods

- **getElementsByTagName(className):**
 - Връща група/масив от елементи

```
var posts = document.getElementsByTagName('post-item');
```

- **getElementsByTagName(tagName);**
 - Връща група/масив от елементи

```
var sidebars = document.getElementsByTagName('sidebar');
```

getElementsByTagName Methods

- **getElementsByTagName(name);**
 - Returns a **collection of elements**

```
var gendersGroup = document.getElementsByTagName('genders');
```

Използване на querySelector методи

querySelector methods

- DOM API притежава методи, използващи CSS селектори за откриване и селектиране на HTML елементи.
 - **querySelector(selector)**
 - і. Връща първия елемент, който селекторът открие.
 - **querySelectorAll(selector)**
 - і. Връща колекция/масив от елементи, които отговарят на селектора.

querySelector methods

- И двата метода приемат параметър стринг -
 - CSS селектора на елемента

```
var header = document.querySelector('#header');
```

```
var navItems = document.querySelectorAll('#main-nav li');
```


**Селектиране
на
вложени
елементи**

Селектиране на вложени елементи

Избиране на всички DIV-елементи, намиращи се в елемент с id = "wrapper",

```
var wrapper = document.getElementById('wrapper');
```

```
var divsInWrapper = wrapper.getElementsByTagName('div');
```

Node

Nodes

NodeList

NodeList

- NodeList - колекцията от елементи, резултат от метода със съответния DOM селектор:
 - `getElementsByTagName(tagName)`
 - `getElementsByName(name)`
 - `getElementsByClassName(className)`
 - `querySelectorAll(selector)`

NodeList

- NodeList прилича на масив, но не е:)
 - Обект - със свойства, подобни на масива
 - length и индекси
 - Ако го обхождаме с for-in може да получим неочаквани резултати:)

Static NodeList

&

Live NodeList

info

Static NodeList and Live NodeList

- Съществуват два вида NodeList-и
 - Методите getElementsByTagName - връщат LiveNodeList
 - Методите querySelectorAll - връщат StaticNodeList
- LiveNodeList следят какво се случва със селектираните елементи
 - Дори, когато са направени промени в DOM, след като е създаден NodeList-a, промените се отразяват в него.

Static NodeList and Live NodeList

- **StaticNodeList** пози елементите такива, каквито са били в момента на изпълнение на метода.
- LiveNodeList е по-бавен от обикновен масив.
 - Препоръчително е да се кешира, за да бъде по-бърз
 - По-добре да се конвертира към масив, когато ще бъде обхождан много пъти.