

JAVASCRIPT

TIMEOUTS. DOM. XHR

СЪДЪРЖАНИЕ

- TIMEOUTS
- DOM
- XHR

TIMEOUTS



SETTIMEOUT И CLEARTIMEOUT

- Изпълнява дадена функция след определено време
- Времето за изпълнение на функцията не е гарантирано, а минимално

SETTIMEOUT

```
function test() {  
    console.log("Hello");  
}  
setTimeout(test, 1000);  
  
console.log("World");
```

```
function test() {  
    console.log("Hello");  
  
    setTimeout(test, 1000);  
}  
  
setTimeout(test, 1000);
```

CLEARTIMEOUT

```
function test() {  
    console.log("Hello");  
}  
var timerId = setTimeout(test, 1000);  
clearTimeout(timerId);  
  
console.log("World");
```

SETINTERVAL И CLEARINTERVAL

- Повтаря изпълнението на дадена функция на определен интервал от време

SETINTERVAL

```
function test() {  
    console.log("Hello");  
}  
  
setInterval(test, 1000);
```


CLEARINTERVAL

```
function test() {  
    console.log("Hello");  
}  
  
var timerId = setInterval(test, 1000);  
clearInterval(timerId);
```

SETINTERVAL VS SETTIMEOUT

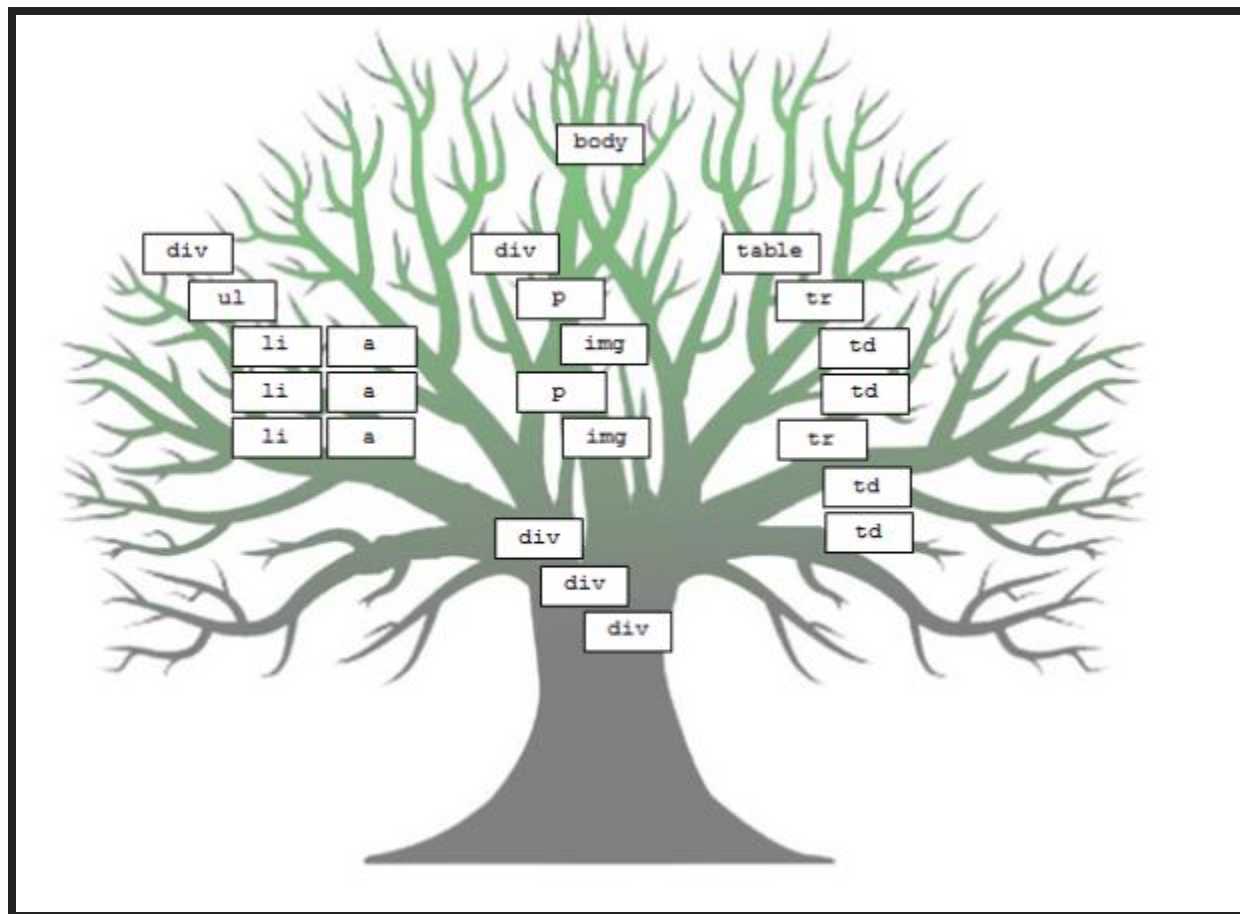
- Understanding timers: setTimeout and setInterval

ПОЛЕЗНИ ВРЪЗКИ

- [What the heck is the event loop anyway?](#)

QUIZ

DOM МАНИПУЛАЦИИ



КАКВО Е DOM?

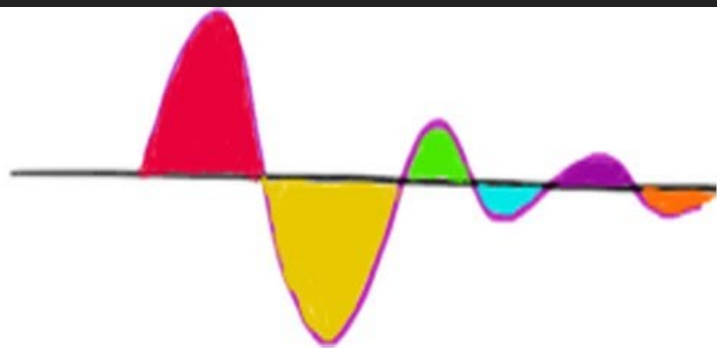
Document Object Model

Структурира се като дърво, което представя какво има
на HTML/XML/SVG документ

HTML ЕЛЕМЕНТ

- div, p, a, head, button, други?
- Тяхното поведение се дефинира чрез JavaScript

ДОСТЪПВАНЕ НА DOM API ПРЕЗ JAVASCRIPT



JavaScript

getElementById()

СЕЛЕКТОРИ

- `document.getElementById()`
- `document.getElementsByTagName()`
- `document.getElementsByClassName()`
- `document.querySelector()`
- `document.querySelectorAll()`

ПРИМЕР

```
<div id="my-div">
  <div id="inner-div">
    <ul>
      <li class="first-li">1.1</li>
      <li class="first-li">1.2</li>
      <li class="second-li">2.1</li>
      <li class="second-li">2.2</li>
    </ul>
  </div>
</div>
```

```
document.getElementById("my-div"); // връща първия див
document.getElementsByClassName("second-li"); // предполагате? :)
document.getElementsByTagName("li") // всички li
document.querySelector("li") // първото li в DOM
```



СЪЗДАВАНЕ И ДОБАВЯНЕ НА DOM ЕЛЕМЕНТИ В СТРАНИЦА



СЪЗДАВАНЕ НА DOM ЕЛЕМЕНТИ

- `document.createElement(tagName)`

```
var myDiv = document.createElement("div")  
console.log(myDiv);
```

ДОБАВЯНЕ НА DOM В СТРАНИЦА

- `element.appendChild()`
- `element.removeChild()`

```
var myDiv = document.createElement("div");  
document.body.appendChild(myDiv);
```



МАНИПУЛИРАНЕ НА HTML ЕЛЕМЕНТИ

- `element.style.propety()`
- `element.hasAttribute()`
- `element.toString()`
- `element.innerHTML`
- `element.textContent`
- `element.children`
- И още много други..



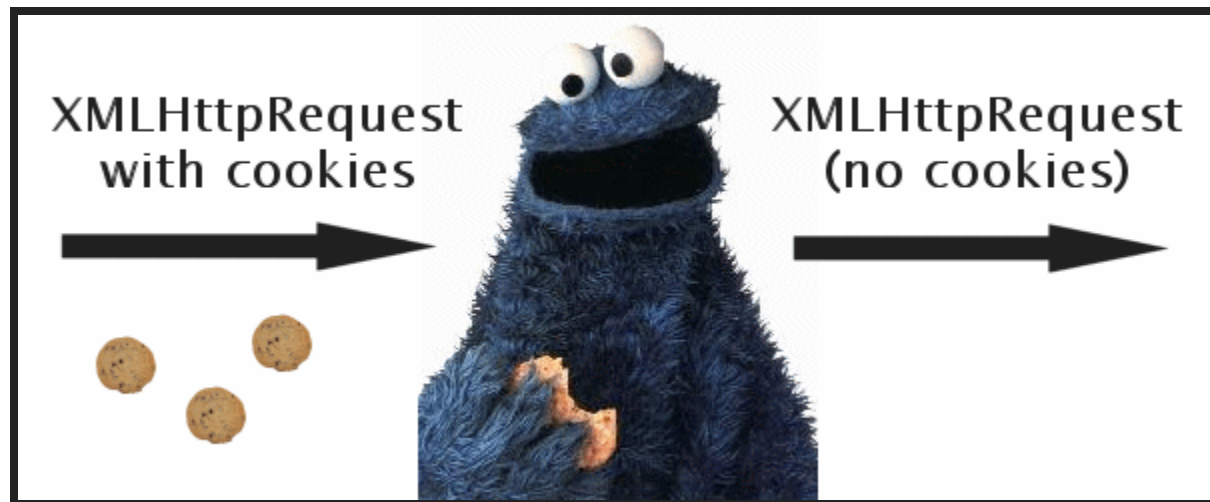
ДОБАВЯНЕ НА СЪБИТИЯ

- click
- addEventListener
- Информация за възможните събития
- scrollTop
- и други

ПРИМЕР

```
var myButton = document.createElement("Button");  
myButton.addEventListener("click", function() {  
    alert("Hi there!");  
});
```

XMLHttpRequest



КАКВО Е XHR

- Служи за комуникация между клиент и сървър
- Няма нужда от пълно презареждане на страницата
- Работи и с други неща освен с XML
- Използва се в AJAX програмирането
- Прави нещата "асинхронни"

XHR СИНТАКСИС

```
var myRequest = new XMLHttpRequest(); //прави обект от такъв тип
```

ОСНОВНИ МЕТОДИ И СВОЙСТВА НА XMLHttpRequest

- `open` - инициализира заявката
- `send` - изпраща заявката
- Всички методи и свойства на обекта може да прочетете [тук](#)



VAR THANK = "YOU!";

```
var tutors = [{  
  name: "Мартин Христов",  
  email: "martin.r.hristov@sap.com"  
}, {  
  name: "Адриан Бобев",  
  email: "adrian.bobev@sap.com"  
}];
```