

Събития /events/

Димитър Митев

JS модел на събития /event model/

- DOM дървото предоставя собствен модел на събитията /event model/
 - Така потребителя може да взаимодейства с браузъра и да получи доста по-динамичен и добър UX
 - Моделът на събитията се състои от
 - Самите събития /events/
 - И нещо което ги прихваща /event listener/

Типове събития

- DOM предоставя съвкупност от събития които могат да бъдат групирани
 - Mouse events
 - Touch events
 - Form events
 - Keyboard events
 - DOM events
- [Пълен списък на всички типове събития](#)
- Могат да се създават и т.нар. Custom Events, т.е. дефинирани от програмиста

Общи типове събития /common event types/

Mouse Events	Keyboard Events	UI Events	Focus events	Touch Events
click	keypress	load	blur	tap
hover	keydown	abort	focus	touchstart
mouseup	keyup	select	focusin	touchend
mousedown		resize	focusout	touchmove
mouseover		change		touchcancel
mouseout		input		touchenter

Регистрация на събития. Прихващане и обработка на събития. /event handling/

- Събитие от конкретен тип може да бъде прихванато и обработено
- Регистрацията на самото събитие може да бъде чрез:
 - HTML атрибути
 - Характеристиките на DOM елемента
 - Т.нар. Event handler

Обработка на събития чрез закачване на HTML атрибути

- Event handler може да бъде закачен директно към HTML атрибута
 - Неговата стойност ще бъде чист JS, но не е задължително да бъде функция

```
<button onclick="buttonClickFunction()">Click Me</button>
```

```
function buttonClickFunction() {  
    console.log("You click the Button");  
}
```

Обработка на събития чрез характеристиките на DOM елемента

- Стандартните DOM елементи имат конкретни събития като характеристики
 - Можем да ги зададем като функции

```
<button id="click-button">Click me</button>
```

```
var button = document.getElementById("click-button");  
button.onclick = function onButtonClick() {  
    console.log("You clicked the button");  
}
```

Обработка на събития чрез Event Listeners

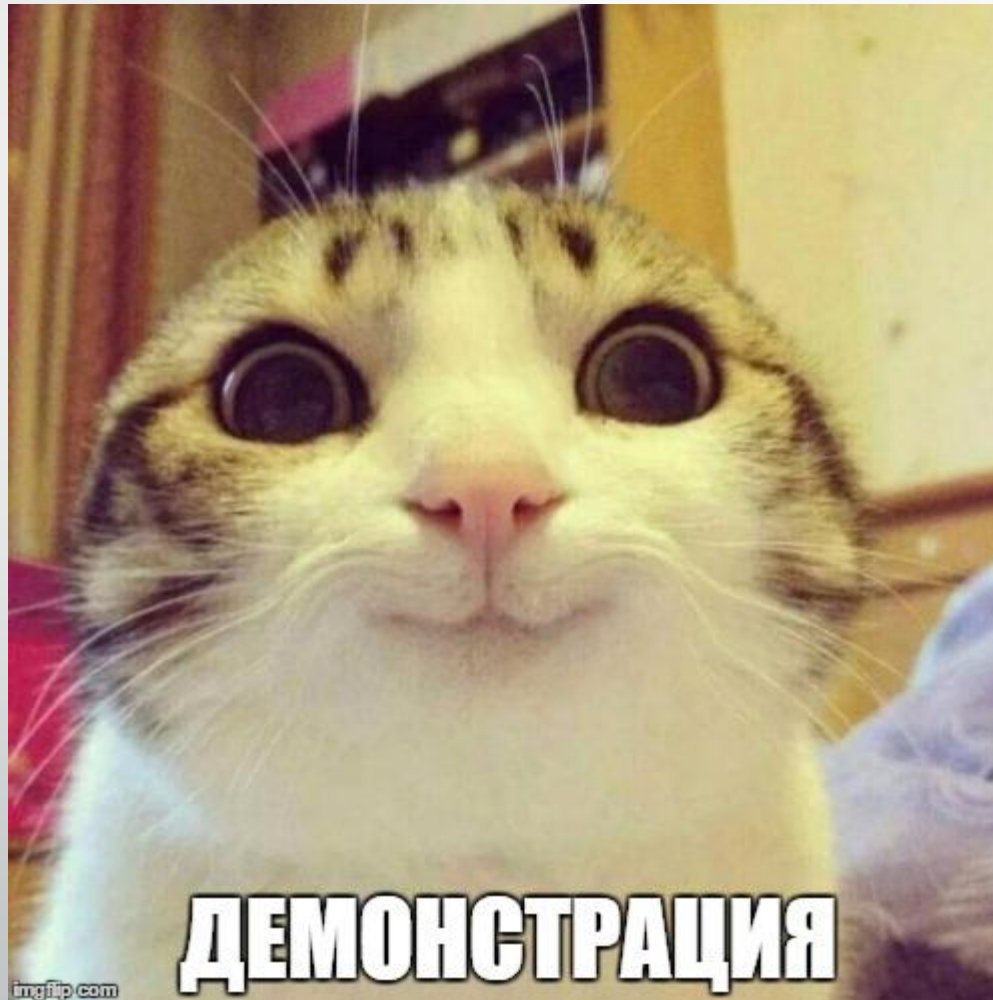
- **Правилният начин!**

- Стандартен синтаксис :

- *domElement.addEventListener(eventType, eventHandler, isCaptureEvent)*

```
var button = document.getElementById("click-button");
button.addEventListener("click", function () {
    console.log("You clicked me");
}, false);
```

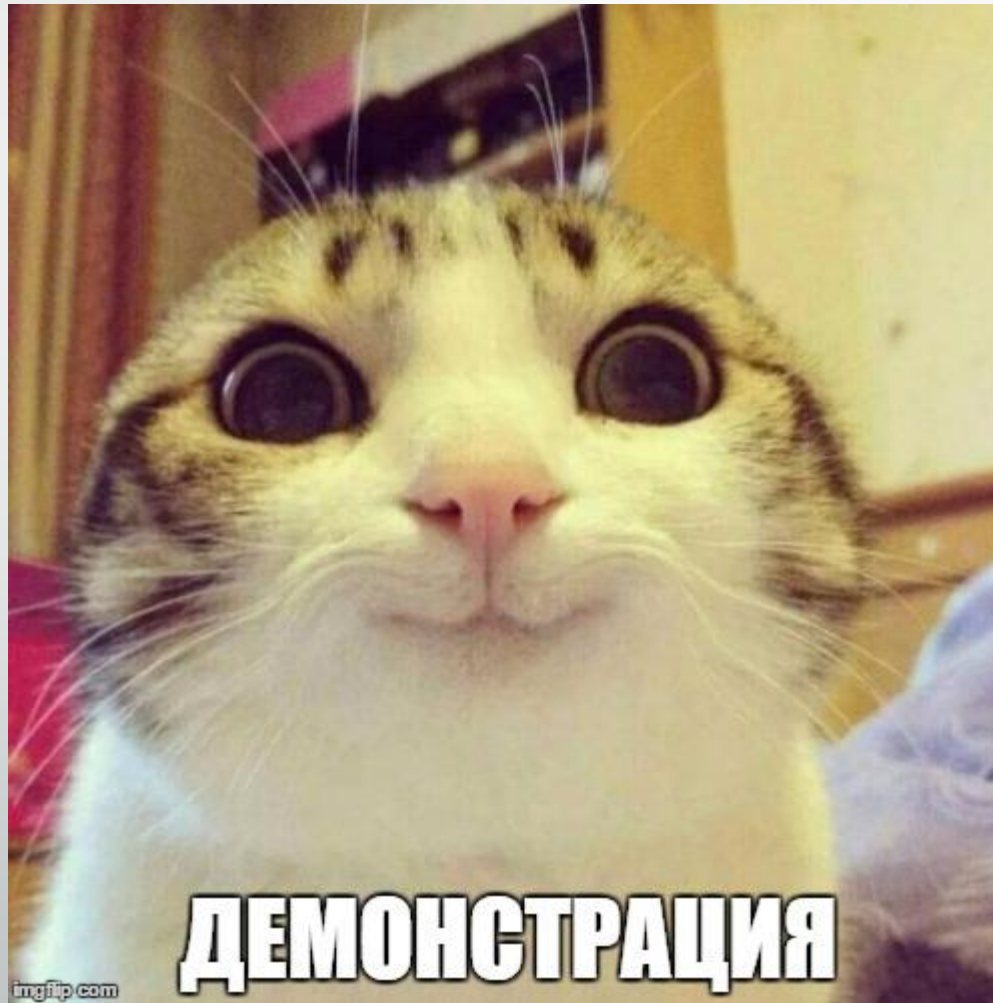

Обработка на събития



Обекта *Event*

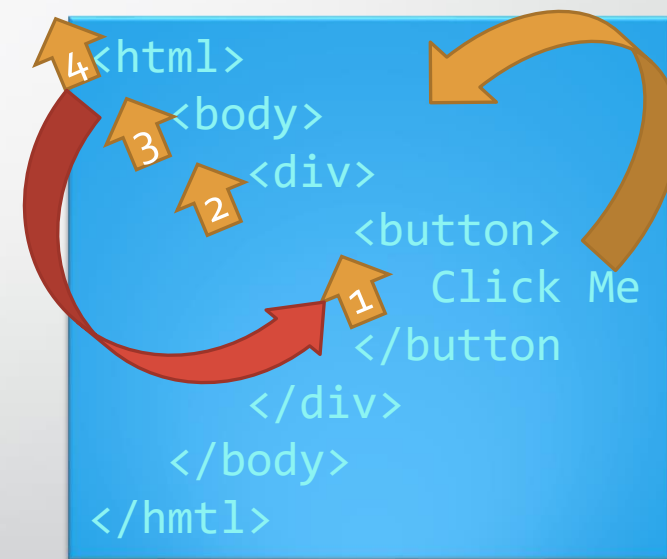
- EventHandler-ите имат достъп до Event обекта, който се подава като параметър на функцията
 - Този обект съдържа информация за
 - Типа на събитието
 - Целта на събитието, т.е. елемента върху, който се е случило
 - Клавиша, който е натиснат, ако е събитие от клавиатурата
 - Кой бутон на мишката е натиснат
 - Позицията на курсора на мишката спрямо екрана

Обекта *Event*

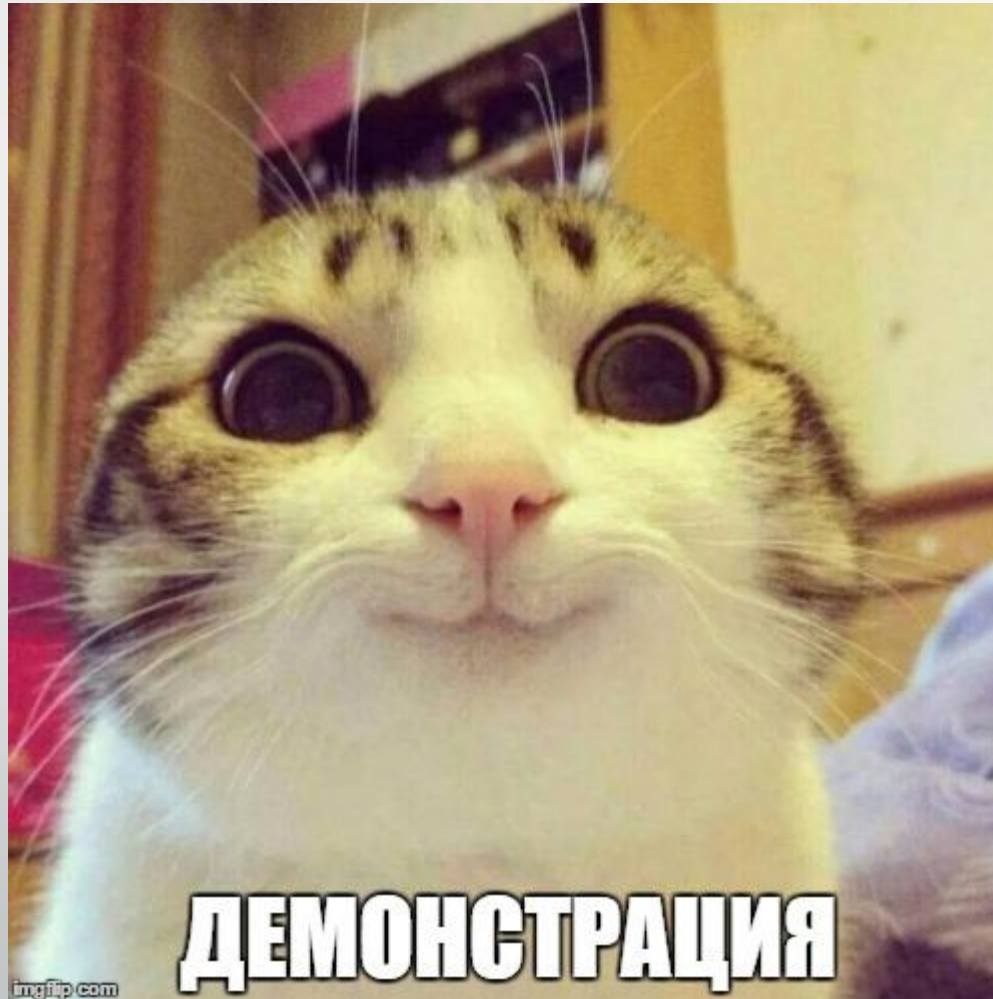


Редица от събития /event chain/. Capturing & Bubbling

- Въпреки това, че потребителя клика върху 1 определен елемент, то събитието се случва и върху всички негови родители
 - Образува се редица от събития
- Съществуват 2 възможности
 - **Bubbling** -> 1вия изпълнен Event Handler е този на конкретния елемент и след това се изпълняват подред тези на родителите
 - **Capturing** -> точно обратното на Bubbling, последен се изпълнява Event Handler-а на елемента върху, който се е случило събитието



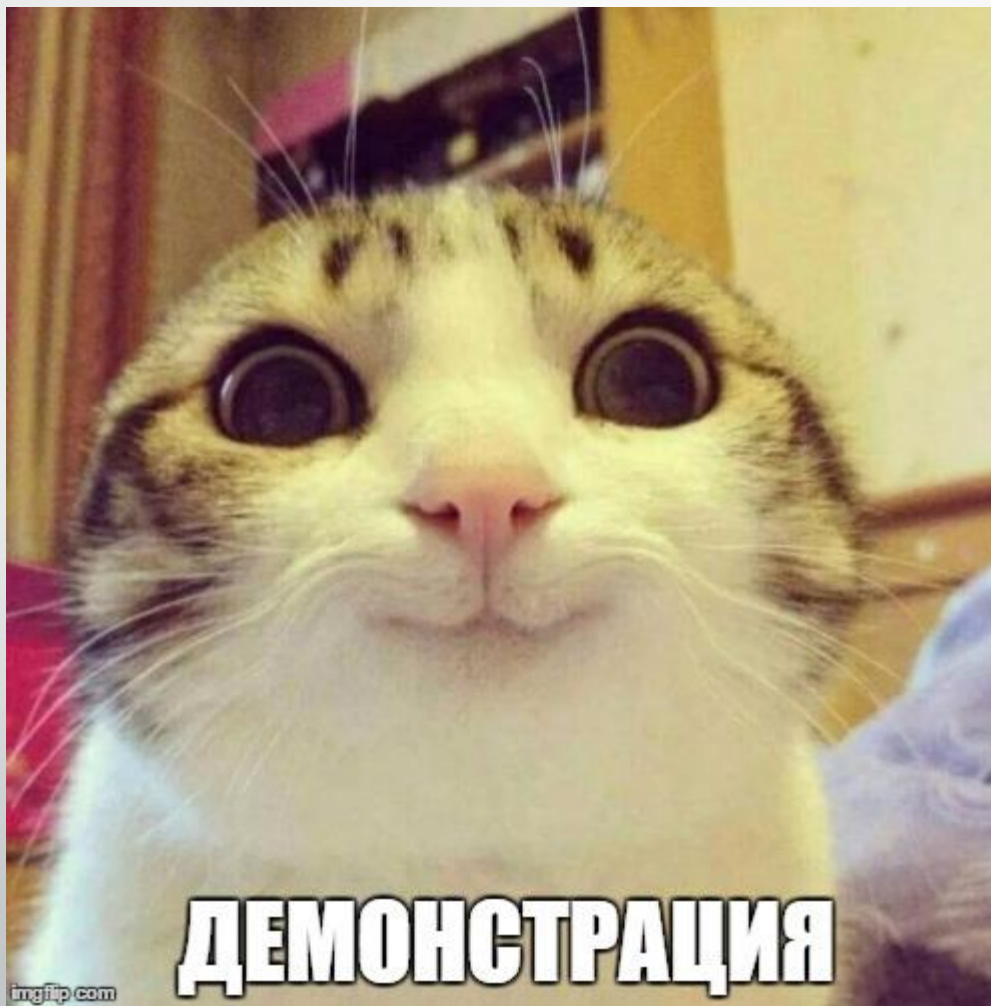
Редица от събития /event chain/. Capturing & Bubbling



Custom Events

- За да създадем наши събития, трябва да използваме функцията конструктор *CustomEvent(eventType)*
 - *eventType* е параметър, който ние подаваме, напр. tripleClick
 - Закачаме се по нормалния начин към новото събитие, чрез помощта на *addEventListener()*

Custom Events



НЯКОЙ ИМА ЛИ



ВЪПРОСИ?

Домашна работа

1. Напишете JS код, който по съществуващ HTML
 - Да намира бутон с id="clickable" и да се закача за събитието 'click'
 - При клик на този бутон да се сменя цвета на фона на 1вия <div /> елемент
 - Да се добавят още 2 бутона + и -, като при клик на + да се увеличава шрифта на текста, а при клик на – да се намалява шрифта на текста
2. Да се напише JS код, който да прави валидация на дадена HTML форма и да извежда на страницата коректни съобщения. Формата има полета за име и години
 - Името може да е само малки или главни латински букви, други символи не са допустими
 - Годините са цяло число в интервала 18-54