
Logika na straně klienta

Martin Klíma



Computer Graphics Group



UDÁLOSTI



Computer Graphics Group



Události v javascriptu

- Implicitní definice Event-handleru
- Explicitní definice Event-handleru
- Objekt event
- Životní cyklus události
- Probublávání události



Události

- Události jsou generovány v uživatelském rozhraní
- Máme možnost je odchytnout a napojit na nějaký vlastní kód
- Část programu, která ošetřuje události se nazývá **Event-Handler**



Napojení na události

- Inline registrace
- Tradiční registrace
- Pokročilá registrace



Inline registrace událostí

```
<A HREF="somewhere.html"  
      onClick="alert('I\'ve been clicked!') ">
```

```
<A HREF="somewhere.html" onClick="doSomething()" >
```

- Registrace je napsaná jako součást HTML kódu
- Není to moc elegantní
- Nedoporučuje se, protože se míchá html a javascript

kuk: [events_register_inline.html](#)



Ošetření událostí

- Některé události mají přiřazené implicitní akce
- Tyto akce jsou volány, pokud neřekneme jinak
- Příklad:
 - click na odkazu způsobí přechod na jinou stránku
 - click na tlačítko submit způsobí odeslání formuláře
- Jestliže definujeme vlastní akci, je pořadí vykonání
 - 1.vlastní definované akce
 - 2.implicitní akce



Zrušení default akce

- w3c: `event.preventDefault()`
- MS: `event.returnValue = false;`

nebo

`return false;`

kuk: [prevent_default_udalosti.html](#)



Defaultní akce - Vyjímka

- je mouseover a `window.status`
- Stornuje se pomocí
`return true;`

Nikdo neví proč :-(



Programová registrace událostí

■ Registrace

```
element.onclick = doSomething;
```

■ Zrušení registrace

```
element.onclick = null;
```



Programové vyvolání události

- je to normální funkce
- lze ji zavolat z programu

```
element.onclick()
```

- Pozor, špatná implementace v IE 5.5

```
element.fireEvent('onclick')
```

Spouštění funkce

- Handleru události předáme ukazatel na funkci

```
function doSomething() {  
    alert ("Stala se událost");  
}
```

```
element.onclick = doSomething
```

Toto je ukazatel na funkci,
žádné závorky!



Event handler ukázka

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <title>
      Mouse Capture
    </title>
    <script type="text/javascript">
      function action() {
        alert("udalost odchycena");
      }

    </script>
  </head>
  <body onload=
    "document.getElementById('div1').onclick = action;">
    <div id="div1" style=
      "height:100;width:200;background-color:red">
      Zde je klikaci prlocha
    </div>

  </body>
</html>
```



Jak registrovat více obsluh k jedné události?

- Co nefunguje:

```
element.onclick = doSomething;  
element.onclick = doSomethingElse;
```

Druhá registrace přepíše první ☹

- Co funguje:

- použijeme anonymní funkci

```
element.onclick =  
function() {doSomething(); doSomethingElse();}
```



Registrace Event-handlerů podle standardu W3C

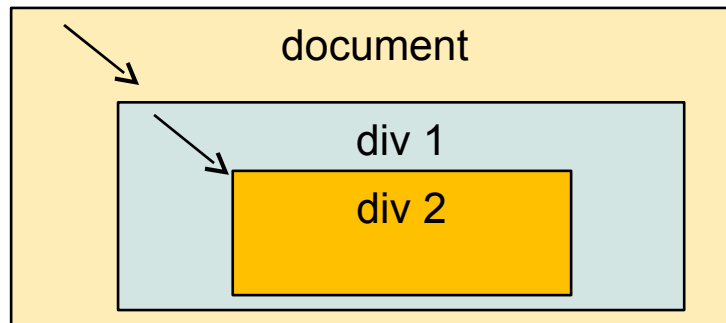
```
element.addEventListener('click',  
                        doSomething, false);  
  
element.addEventListener('click',  
                        doSomethingElse, false);
```

- Registrují se oba
- Poslední argument určuje, zda se událost má odchytit ve fázi capture nebo bubble (false=bubble)
- Odstranění Event-handleru

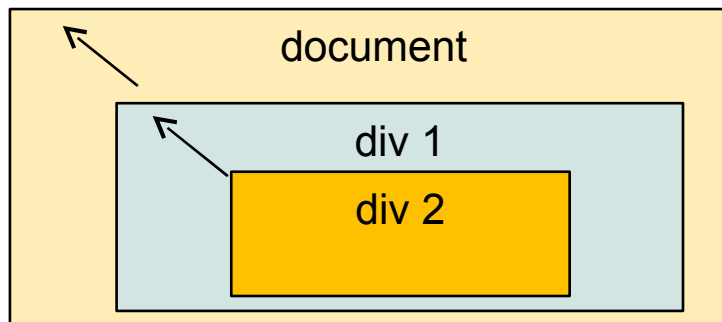
```
element.removeEventListener('click', doSomethingElse, false)
```

Způsoby zachytávání událostí

- Otázka: jestliže mám vnořený element který odchytává stejnou událost jako jeho nadřazený element, kdo to má odchytit první?
- Event Capture
 1. document
 2. div 1
 3. div 2
- Event Bubbling
 - div 2
 - div 1
 - document



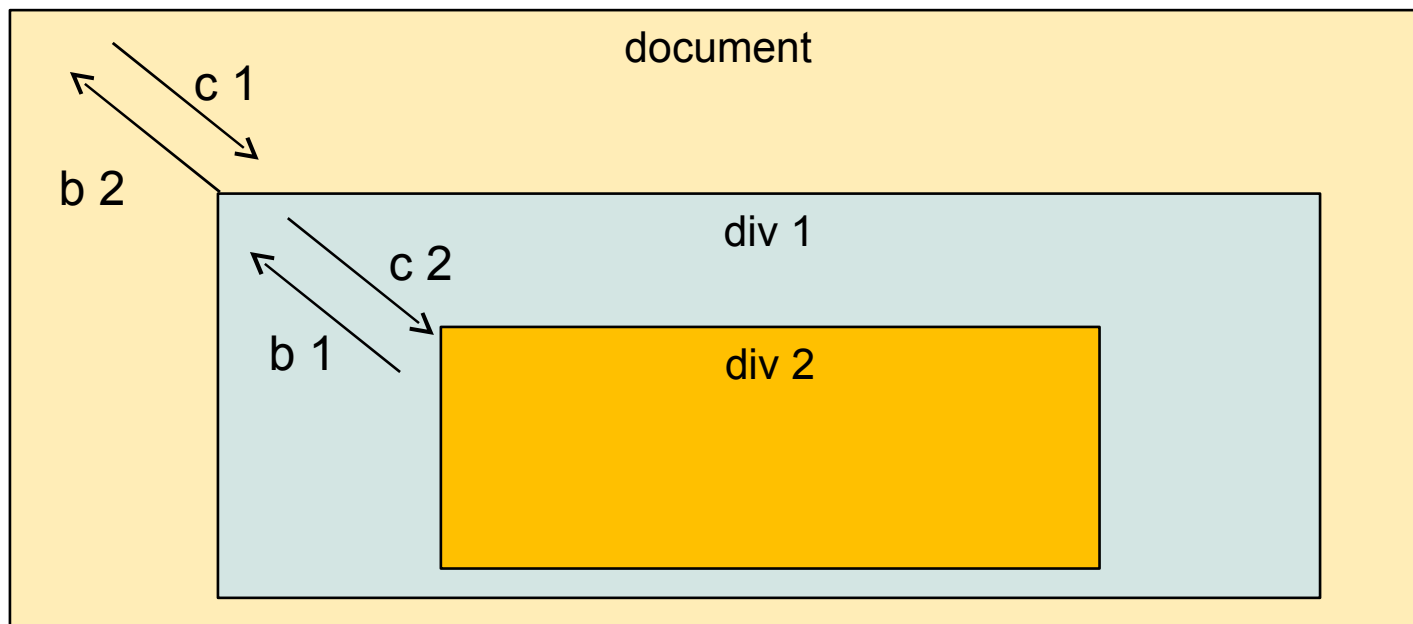
Netscape



Microsoft

Propagování události – W3C

- Kombinace capture a bubble propagace
- Nejprve capture (c), pak zpětné bubble (b)



kuk: capture.htm

Registrace událostí v IE

Registrace

```
element.attachEvent('click', doSomething)  
element.attachEvent('click', doSomethingElse)
```

Zrušení registrace

```
element.detachEvent('click', doSomething)
```

Zrušení probublávání

- Standard

```
event.cancelBubble = true;
```

- IE

```
event.stopPropagation();
```