

Javascript v praxi

Zdeněk Vlach



SEZNAM.CZ
...najdu tam, co hledám





Co nás čeká?

- Core versus frameworky
- DOM – kolekce
- Události – fáze zpracování události
- Detekce (browser x feature)
- Prototypy
- Čas na dotazy

Core versus Frameworky

- Je fajn vědět na čem stavím
- Framework – rychlost
- Core – vím co dělám!
- Takže co je lepší?

DOM (Document Object Model)

- Programové rozhraní v Javascriptu pro práci s HTML a XML dokumenty
- Reprezentace dokumentu je strukturovaná tvořená ulzy a objekty, které mají vlastnosti a metody

```
var div = document.getElementsByTagName("div");  
alert(div[0].nodeName);
```

DOM – pole

- Jsou součástí jádra Javascriptu

```
var pole = new Array(1,2,3);  
pole.slice(1,2); // vrati 2
```

- Odbočka: procházení pole cyklem for

```
for(var i=0; i<pole.length;i++)  
    alert(pole[i])
```

```
Array.prototype.kuk = function() { //neco }  
for(var i in pole) {alert(i)} // chyba
```

DOM – nodeList (kolekce)

- Jsou součástí DOM API
- Repräsentují výsledek, který vrací metody jako:
`document.getElementsByTagName()`,
`document.getElementsByClassName()`

DOM – nodeList

- Co vlastně kolekce je?

```
var nodes = document.getElementsByTagName("div")
nodes.length; // vrati nejake cislo
nodes[0]; // pristoupi k prvni polozce

nodes.slice(1,2); // ha tohle nefunguje
nodes.item(0); // a tohle zase ano
```

DOM – nodeList

```
var nodes = document.getElementsByTagName("div");
var pole = [1,2,3];
nodes.constructor.toString();
    // "[object NodeListConstructor]"
pole.constructor.toString();
    // "function Array() {
        [native code]
    }"
```

DOM – nodeList

- „Žije“
- Může být výhodnější převést kolekci na obyčejné pole

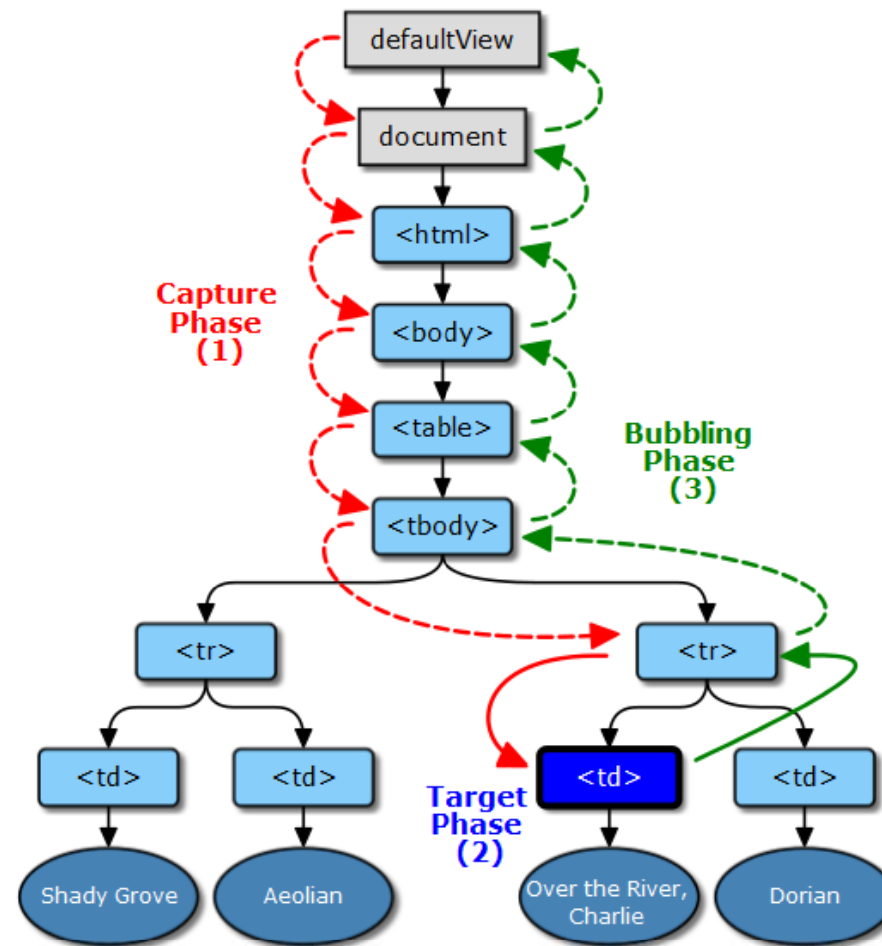
```
var n = document.getElementsByTagName( "div" );  
for (var i=0; i<n.lenght; i++) {  
n[i].parentNode.removeChild(n[i]);  
}  
// nevymaze vse  
// reseni preved na pole nebo zmen cyklus
```

Události

- API úrovně 0 – ošklivé pomocí atributů
- API úrovně 2 – standard používaný v současnosti
- Fáze zpracování události
 - Zachytávání (true)
 - Target
 - Bublání (false)

```
node.addEventListener("click", show, true/false)
```

Události – model ZB



Události – příklad

```
document.addEventListener("click", show1, true);  
document.getElementById("header").addEventListener  
("click", show2, true);  
// naveseno v zachytavaci fazi
```

- Jaké bude pořadí volání posluchačů?
- Jak to ovlivní funkce stopPropagation?

Detekce

- Co chci detekovat?
 - Prohlížeč
 - Vlastnost kterou chci použít
 - Používané a podporované DOM moduly

Detekce - prohlížeč

- Možnosti:
 - Vlastnost specifickou pro daný prohlížeč
 - UserAgent řetězec (navigator.userAgent)

```
if (window.opera) {  
    return "opera";  
} else if (window.chrome) {  
    return "chrome";  
else if (this._agent.indexOf("KHTML")) {  
    if (this._vendor == "KDE") {  
        return "konqueror";  
    } else {  
        return "safari";  
    }  
}
```


Detekce - feature

- Pokud používám feature dostupnou pouze v některých prohlížečích – nebudu používat k detekci UserAgent řetězec, ale:

```
if (document.addEventListener) {  
    elm.addEventListener(type, action, capture);  
} else if (document.attachEvent) {  
    elm.attachEvent('on'+type, action);  
}
```

Detekce - hasFeature

- Umožňuje detekovat, který DOM modul a v jaké verzi prohlížeč podporuje
- Moduly: HTML, XHTML, Core, CSS, CSS2
- Odpověď true nemusí být výhrou

```
console.log(document.implementation.hasFeature("HTML", "2.0"))
```

Prototypy

- Javascript je netřídní jazyk (class je jen rezervované slovo)
- Bez silné typové kontroly
- Trošku obskurní, ale objektový a s prototypovou dědičností na místo třídní

Prototypy – co je to?

- Každý objekt v JS má svůj vlastní prototypový objekt
- Všechny vlastnosti prototypového objektu se jeví jako vlastnosti každého objektu, jehož je prototypem
- To znamená že, každý objekt **dědí** vlastnosti svého prototypu

Prototypy – idea

Prototype objekt pro
T
(sdílené vlastnosti)

```
T.prototype.p = 10;  
T.prototype.a = [];  
T.prototype.f. =  
  function () {};
```

//setrime pamet

Instance t2 new T()

```
alert(t2.p); // 10
```

```
alert(t2.a.length);
```

Instance t1 new T()

```
alert(t1.p); // 10  
t1.p = 20;  
alert(t1.p); // 20
```

```
T1.a.push(1);  
alert(t1.a.length);
```

Prototypy – ukázka

```
var Vozidlo = function () {} // funkce konstrukturu
Vozidlo.prototype.jed = function () {}

console.log(Vozidlo.prototype.constructor) // fce Vozidlo

var vozidlo = new Vozidlo(); //nová instance

// zkusim vytvorit analogii tridniho dedeni
var Tank = function () {} // funkce konstrukturu
Tank.prototype = vozidlo; //bez frameworku musim takto
Tank.prototype.constructor = Tank; // jelikoz si prepisu cely
prototypovy objekt
Tank.prototype.vystrel = function () {}

var tank = new Tank() // bude mit metody jed i vystrel
```

Otázky a odpovědi



Děkuji za pozornost...

Zdeněk Vlach



SEZNAM.CZ

SEZNAM.CZ

...najdu tam, co hledám