

Webové technológie 9

Aplikácie Webových technológií

Peter Bednár

Knižnica jQuery

- Jednoduchá knižnica jazyka JavaScript, ktorá zjednodušuje prácu s DOM, obsluhu udalostí a AJAX volania
- Kompatibilná s hlavnými prehliadačmi
 - Firefox, Chrome, Safari, Internet Explorer, Edge
 - Zjednotené rozhranie, ktoré ošetruje odlišnosti v implementácii medzi prehliadačmi
- Dokumentácia
 - <https://api.jquery.com>
- CDN URL pre pridanie knižnice
 - <https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js>
 - <https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js>



Rozhranie jQuery

- Vytvára sa volaním funkcie `jQuery(arg)`
 - Skrátený zápis volania `$(arg)`
- **Argument:**
 - Reťazec s CSS selektorom pre výber elementov
 - DOM objekt/pole objektov ku ktorým chceme pristupovať cez jQuery rozhranie
- **Návratová hodnota:**
 - jQuery objekt, ktorý reprezentuje kolekciu elementov
 - Indexovanie `$(arg)[index]` od 0
 - Počet elementov `$(arg).length`

Prístup k elementom (1)

- V jQuery sa primárne pristupuje k elementom pomocou CSS selektorov, napr.:
 - `$("#id")` – vráti kolekciu, ktorá bude obsahovať iba element s daným ID
 - `$("elm")` – vráti kolekciu elementov daného typu, napr. všetky elementy ``: `$("span")`
 - `$("elm1 > elm2")` – vráti kolekciu elementov typu `elm2`, ktoré sú vnorené do elementov typu `elm1`
 - `$(".class")` – vráti kolekciu elementov patriacich do CSS triedy `class`

Prístup k elementom (2)

- Dodatočne je možné kolekciu elementov odfiltrovať
- `jq.filter(selector)` – vráti podmnožinu elementov z kolekcie `jq` ktoré vyhovujú zadanému selektoru
- `jq.find(selector)` – vráti kolekciu všetkých potomkov elementov v kolekcii `jq`, ktoré vyhovujú zadanému selektoru

Prístup k elementom (3)

- Príklad:

```
// kolekcia nadpis1 obsahuje element s ID "nadpis1"  
var nadpis1 = $("#nadpis1");  
// kolekcia podnadpisy obsahuje potomkov elementu  
// nadpis1, ktoré sú zaradené do triedy  
// "podnadpis"  
var podnadpisy = nadpis1.find(".podnadpis");
```

- Zreťazený zápis:

```
var podnadpisy = $("#nadpis1").find(".podnadpis");
```

Potomkovia, rodičia a súrodenci (1)

- Pre všetky nasledujúce metódy je možné uviesť ako parameter voliteľný selektor pre filtrovanie
- `jq.children()` – vráti kolekciu všetkých priamych potomkov elementov v kolekcii `jq`
- `jq.parent()` – vráti kolekciu všetkých priamych predkov elementov v kolekcii `jq`
- `jq.parents()` – vráti kolekciu všetkých predkov elementov v kolekcii `jq`

Potomkovia, rodičia a súrodenci (2)

- `jq.next()` – vráti kolekciu súrodencov, ktoré priamo nasledujú za elementami z kolekcie `jq`
- `jq.nextAll()` – vráti kolekciu všetkých súrodencov, ktoré nasledujú za elementami z kolekcie `jq`
- `jq.prev()` – vráti kolekciu súrodencov, ktoré priamo predchádzajú pred elementami z kolekcie `jq`
- `jq.prevAll()` – vráti kolekciu všetkých súrodencov, ktoré priamo predchádzajú pred elementami z kolekcie `jq`

Obsah elementov

- `jq.html()` – vráti reťazec s HTML obsahom prvého elementu v kolekcii `jq`
- `jq.html("obsah")` – nastaví HTML obsah všetkých elementov v kolekcii `jq`
- `jq.text()` – vráti reťazec ktorý vznikne spojením textov všetkých elementov v kolekcii `jq`
- `jq.text("text")` – nastaví textový obsah všetkých elementov v kolekcii `jq`

- Príklad:

```
$("#info").html("Zmenený <span>obsah</span>");  
// objekty sa dynamicky zmenia, nastavíme obsah  
// elementu <span> na "text"  
$("#info").children().text("text");
```

Zmena atribútov elementu

- `jq.attr(attr)` – vráti reťazec s hodnotou atribútu `attr` pre prvý element v kolekcii `jq`, ak hodnota neexistuje, vráti `undefined`
- `jq.attr(attr, val)` – nastaví atribút `attr` všetkých elementov v kolekcii `jq` na hodnotu `val`
- `jq.removeAttr(attr)` – odstráni atribút `attr` zo všetkých elementoch v kolekcii `jq`
- Hodnoty prvkov formulára (atribút `value`)
 - `jq.val()` – vráti reťazec s hodnotou pre prvý element v kolekcii `jq`, ak hodnota neexistuje, vráti `undefined`
 - `jq.val(val)` – nastaví hodnoty prvkov formulára všetkých elementov v kolekcii `jq` na hodnotu `val`

Zmena CSS

- `jq.css(prop)` – vráti reťazec s vypočítanou hodnotou CSS vlastnosti `prop` pre prvý element v kolekcii `jq`
- `jq.css(prop, val)` – nastaví CSS vlastnosť `prop` všetkých elementov v kolekcii `jq` na hodnotu `val`
- Nasledujúce metódy menia všetky elementy v kolekcii `jq`
 - `jq.addClass(class)` – pridá CSS triedu (`class` môže byť aj zoznam tried oddelených medzerou, napr. "nadpis nav")
 - `jq.removeClass(class)` – odstráni CSS triedu
 - `jq.toggleClass(class)` – pridá CSS triedu ak nebola elementu priradená, resp. odstráni ak bola
- `jq.hasClass(class)` – vráti `true` ak niektorí z elementov kolekcie `jq` je zaradený do CSS triedy `class`

Vytvorenie, vloženie a zmazanie elementu

- `$("#<typ/>")` – vytvorenie nového elementu daného typu, napr. pre ``:

```
var span = $("#<span/>").text("text elementu");
```

- `jq.before(elms)` – pridá všetky elementy z kolekcie `elms` ako súrodencov pred všetky elementy v kolekcii `jq`
- `jq.after(elms)` – pridá všetky elementy z kolekcie `elms` ako súrodencov za všetky elementy v kolekcii `jq`
- `jq.append(obsah)` – pridá HTML obsah na koniec obsahu každého elementu v kolekcii `jq`
- `jq.prepend(obsah)` – pridá HTML obsah na začiatok obsahu každého elementu v kolekcii `jq`
- `jq.remove()` – odstráni z dokumentu všetky elementy v kolekcii `jq`

Obsluha udalostí

- `jq.on("typ", f)` – nastaví obsluhu udalostí daného typu na danú funkciu pre všetky elementy v kolekcii `jq`
- `jq.off("typ", f)` – odoberie funkciu z obsluhy udalostí daného typu pre všetky elementy v kolekcii `jq`
- Príklad

```
// štandardné rozhranie
var elms = document.querySelectorAll("*");
for (var i = 0; i < elms.length; i++) {
    elms[i].addEventListener("click", onClick);
}
// jQuery
$("*").on("click", onClick);
```

AJAX v jQuery

- AJAX volanie sa vytvára volaním funkcie `$.ajax(nastavenia)`
- Nastavenia – objekt s vlastnosťami:
 - `type` – názov metódy HTTP (napr. `"GET"`)
 - `url` – URL adresa servera
 - `dataType` – formát odosielaných alebo prijímaných dát, možné formáty `"json"`, `"xml"`, `"html"`, `"text"`
 - `data` – odosielané dáta pre PUT a POST metódu
 - `async` – `true` pre asynchrónne volanie (prednastavená hodnota), `false` pre synchrónne

AJAX v jQuery – obslužné funkcie (1)

- Obsluha po úspešnom volaní:
 - Registruje sa metódou `$.ajax(nastavenia).done(f)`
 - Dáta sú predané ako argument `data` – hodnota je načítaná podľa nastavenia `dataType` (objekt JavaScript-u pre JSON, DOM objekt pre HTML, alebo XML, alebo reťazec pre ostatné formáty)

- Príklad:

```
function onSuccess(data) {  
    // pre JSON formát jQuery automaticky skonvertuje  
    // prijaté dáta na JavaScript objekt  
    alert("Ahoj " + data.firstName);  
}
```

AJAX v jQuery – obslužné funkcie (2)

- Obsluha pri chybe:
 - Registruje sa metódou `$.ajax(nastavenia).fail(f)`
 - Argument `response` - objekt odpovede AJAX rozhrania z ktorého je možné zistiť chybový kód, alebo textový popis chyby
- Príklad:

```
function onError(response) {  
    // spracovanie chyby  
    alert("Došlo k chybe: " + response.statusText);  
}
```


Príklad AJAX v jQuery – GET

```
function onSuccess(data) {  
    // spracovanie dát (data je JavaScript objekt)  
}  
function onError(response) {  
    // spracovanie chyby  
}  
$.ajax({  
    type: "GET",  
    url: "http://www.example.org/data/",  
    dataType: "json",  
}).done(onSuccess).fail(onError);
```

Príklad AJAX v jQuery – PUT

```
function onError(response) {  
    // spracovanie chyby  
}  
var data = { firstName: "Peter", lastName: "Bednár" };  
$.ajax({  
    type: "PUT",  
    url: "http://www.example.org/data/",  
    dataType: "json",  
    data: data,  
}).fail(onError);
```

Pre JSON formát je možné nastaviť odosielané dáta priamo na JavaScript objekt

jQuery – prehľad API (1)

- Prístup k elementom
 - `$()`
 - `children()`, `parent()`, `parents()`, `next()`, `nextAll()`, `prev()`, `prevAll()`
- Obsah elementov
 - `html()`, `text()`
- Zmena atribútov
 - `attr()`, `removeAttr()`, `val()`
 - `css()`, `addClass()`, `removeClass()`, `toggleClass()`, `hasClass()`

jQuery – prehľad API (2)

- Vytvorenie, vloženie a zmazanie elementov
 - `$("<typ/>")`
 - `before()`, `after()`, `append()`, `prepend()`, `remove()`
- Obsluha udalostí
 - `on()`, `off()`
- AJAX
 - `$.ajax()`

Knižnica Bootstrap

- Knižnica CSS štýlov pre návrh responzívnych stránok
 - Štýly pre zjednotenie typografie, formátovania a rozvrhovania stránky
 - Knižnica komponentov používateľského rozhrania
- Dokumentácia
 - <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- Štýly CSS
 - [bootstrap.min.css](#)
- Voliteľné JavaScript knižnice potrebné pre niektoré komponenty používateľského rozhrania
 - [jquery-3.3.1.slim.min.js](#)
 - [popper.min.js](#)
 - [bootstrap.min.js](#)



Zjednotenie typografie a formátovania

- Zjednotenie formátovania pre rôzne prehliadače pomocou `reboot.css`
- Nastavenie fontu písma podľa systému prehliadača (tzv. *native font stack*)
- Nastavenie formátovania elementov
 - Nadpisy
 - Odstavce
 - Zoznamy
 - Tabuľky
 - Prvky formulárov

Rozvrhovanie stránok

- Rozvrhovanie je založené na rovnomernom rozdelení stránky na 12 stĺpcov ktorých šírka sa prispôsobuje podľa aktuálnej šírky viewportu
 - Implementácia je zložená na flexbox zobrazení
- Obsah je rozvrhovaný pomocou `<div>` elementov, ktorým sú priradené CSS triedy:
 - `row` – rozdeľujú stránku vertikálne na riadky s maximálnou šírkou
 - `col` – definujú horizontálne stĺpce

Šírka stĺpcov

- **col-šírka**
 - Napr. **col-3** definuje stĺpec, ktorý zaberá $3/12=1/4$ viewportu
- **col-veľkosť zobrazenie-šírka**
 - Napr. **col-md-3** definuje šírku stĺpca $1/4$ ak je stránka zobrazená na stredne veľkom okne
- Veľkosti

Prefix	Veľkosť viewportu
col-	< 576 px
col-sm	≥ 576 px
col-md	≥ 768 px
col-lg	≥ 992 px
col-lx	≥ 1200 px

Rozvrhovanie stránky – príklad (1)

```
<div class="row">  
  <div class="col-6 col-md-4">prvý</div>  
  <div class="col-6 col-md-8">druhý</div>  
</div>
```

- Zobrazenie pre viewport ≥ 768 px

prvý	druhý
------	-------

- Zobrazenie pre malé okno < 768 px

prvý	druhý
------	-------

Rozvrhovanie stránky – príklad (2)

```
<div class="row">  
  <div class="col-md-3">prvý</div>  
  <div class="col-md-6">druhý</div>  
  <div class="col-md-3">tretí</div>  
</div>
```

- Zobrazenie pre viewport ≥ 768 px

prvý	druhý	tretí
------	-------	-------

prvý
druhý
tretí

- Zobrazenie pre malé okno < 768 px

Bootstrap komponenty (1)

- Správy pre používateľa
 - [Alerts](#)
- Navigačné riadky
 - [Breadcrumb](#), [Navs](#), [Navbar](#)
- Informačné bannery
 - [Carousel](#), [Jumbotron](#)
- Karty
 - [Card](#)
- Tlačidlá
 - [Buttons](#), [Button group](#)
- Formuláre
 - [Forms](#), [Input group](#)

Bootstrap komponenty (2)

- Modálne okná
 - [Modal](#)
- Vyskakovacie okná
 - [Popovers](#)
- Stránkovanie
 - [Pagination](#)
- Ukazovatele priebehu
 - [Progress](#)
- ...

Príklady

- TODO list v jQuery
- Archív súborov:
 - [todo-jquery.zip](#)
- HTML stránka
 - [index.html](#)
- Skript
 - [js/todo-jquery.js](#)