**Osnove jQueryja**

**00:00:00**

Jedinica: 30 od 30

**+Rezime**

jQuery je jedna od najpopularnijih JavaScript biblioteka koju je kreirao *John Resig* 2006. godine, pod sloganom *Write less, do more*. I upravo je to i osnovna odlika jQuery biblioteke, s obzirom da ona pojednostavljuje realizaciju praktično svakog aspekta JavaScript programiranja. Neke od glavnih osobina jQuery biblioteke su:

* mogućnost lake manipulacije DOM strukturom, korišćenjem mehanizma za selektovanje otvorenog koda, pod nazivom *Sizzle*.
* elegantan način za *hvatanje* velikog broja najrazličitijih događaja
* uprošćen, nezavisan način za upućivanje sinhronih i asinhronih zahteva
* mnoštvo ugrađenih animacija
* malo zauzeće resursa – jQuery biblioteka zauzima 19 KB
* podrška za sve moderne web pregledače



*Slika 30.1 – jQuery logo*

**Implementacija jQuery biblioteke u stranicu**

Za implementiranje jQuery biblioteke u stranicu, postoji dva načina:

* preuzimanje fajlova biblioteke sa sajta jQuery.com, i njihovo uključivanje u stranicu sa lokalnog servera
* uključivanje jQuery biblioteke sa eksternih servera koji se nazivaju [CDN](https://www.link-elearning.com/linkdl/opisPojma.php?id=144321)

**Preuzimanje jQuery biblioteke**

Za preuzimanje sa jQuery.com web sajta dostupne su dve verzije jQuery biblioteke:

* produkciona verzija
* razvojna verzija

Produkciona verzija je ona koja se preporučuje za korišćenje na sajtovima koji su potpuno funkcionalni i dostupni korisnicima. Drugim rečima, ovu verziju je potrebno koristiti na živim, objavljenim sajtovima. Produkciona verzija jQuery biblioteke je kompresovana i umanjena ([minified](https://www.link-elearning.com/linkdl/opisPojma.php?id=144322)).

Razvojna verzija jQuery biblioteke namenjena je testiranju i razvoju, te zbog toga nije kompresovana niti umanjena, što znači da je njen kod lako čitljiv.

Preporučuje se korišćenje razvojne verzije jQuery biblioteke u fazi razvoja aplikacije i obavezno korišćenje produkcione verzije, kada sajt postane *živ*.

Kada se jednom preuzme fajl koji predstavlja jQuery biblioteku, bilo da je reč o produkcionoj ili razvojnoj verziji, potrebno je implementirati neophodne fajlove ove biblioteke na stranicama na kojima će se takve funkcionalnosti koristiti. Implementiranje podrazumeva korišćenje <script> tagova, baš kao i prilikom uključivanja eksternih fajlova sa JavaScript kodom. Na taj način, kod za implementaciju biblioteke je sledeći:

|  |  |  |
| --- | --- | --- |
| 1 | | <script src="/scripts/jquery-1.9.1.js"></script> |
| **Napomena**    *Primer podrazumeva da se biblioteka jQuery nalazi u folderu scripts.* | |

**Korišćenje jQuery biblioteke sa CDN-a**

Drugi način za uključivanje jQuery biblioteke u stranicu jeste korišćenje eksternih servera. Naime, nekoliko velikih korporacija hostuje jQuery biblioteku na svojim serverima, odakle se ova biblioteka slobodno može koristiti na HTML stranicama. Mreža servera na kojima se jQuery i slične biblioteke hostuju naziva se *Content Delivery Network*.

Dve velike korporacije koje hostuju jQuery biblioteku na svojim serverima su Google i Microsoft. Ukoliko se koristi Google CDN, uključivanje jQuery biblioteke u stranicu izgleda ovako:

|  |  |
| --- | --- |
| 1 | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script> |

Uključivanje jQuery biblioteke korišćenjem Microsoft CDN-a:

|  |  |  |
| --- | --- | --- |
| 1 | | <script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.4.min.js"></script><br><br> |
| **Napomena**    *U primerima uključivanja jQuery biblioteke korišćenjem CDN-a, unutar putanje, verzija se može razlikovati.* |

Korišćenje CDN-a za uključivanje jQuery biblioteke, donosi brojne prednosti. Biblioteke uključene na ovaj način biće keširane na klijentskim računarima samo jednom i to prvi put kada korisnik poseti neki sajt koji koristi ovako uključenu jQuery biblioteku. Pošto je takvih sajtova zaista puno, velike su šanse da klijent već poseduje keširanu verziju biblioteke, te na taj način neće morati da je ponovo preuzima, što značajno ubrzava učitavanje stranice. Takođe, malopre je rečeno da je CDN zapravo mreža servera. Ovo praktično znači da će klijent biti opslužen sa servera koji je najbliži lokaciji klijenta, što će dodatno ubrzati učitavanje.

**jQuery selektori**

Jedna od glavnih oblasti koju jQuery značajno pojednostavljuje jeste selektovanje elemenata.

Da bi se rukovalo HTML elementima, njih je prethodno potrebno locirati. To je u dosadašnjim lekcijama realizovano direktno korišćenjem JavaScript jezika, selektovanjem elemenata DOM stabla koje browser kreira prilikom parsiranja stranice. Biblioteka jQuery značajno pojednostavljuje proces selekcije elemenata, korišćenjem takozvanih selektora, koji oponašaju CSS selektore.

jQuery selektori počinju dollar ($) karakterom, nakon koga slede zagrade - $(). Na taj način se mogu kreirati tri osnovne grupe selektora, prikazane tabelom 30.1.

|  |  |  |
| --- | --- | --- |
| **Kriterijum selekcije** | **Primer** | **Opis** |
| Naziv taga | $("h1") | selektuje sve elemente sa definisanim nazivom taga |
| ID taga | $("#menu") | Selektuje element iz DOM strukture koji ima definisanu id vrednost |
| Naziv klase | $(".some-class") | Selektuje sve elemente sa navedenom klasom |

*Tabela 30.1 – jQuery selektori*

Sve prikazane selektore je moguće kombinovati.

|  |
| --- |
| **Napomena**    *Ugrađena jQuery funkcija $() je ekvivalentna funkciji jQuery(), i one se mogu ravnopravno koristiti.* |

|  |
| --- |
| **Primer**    Ukoliko je potrebno pročitati vrednost koju je korisnik uneo u polje na stranici koje ima id vrednost *username*i takvu vrednost prikazati u zasebnom dijalogu, dovoljno je uraditi sledeće:    var username = $("#username").val(); alert(username);  U prikazanom primeru, u pozadini, jQuery vrši pretragu DOM stabla kako bi pronašao element koji ima id vrednost *username*. Ukoliko takav element bude pronađen, izvršava se metoda val koja obavlja čitanje vrednosti, a pritom sam jQuery vrši čitanje vrednosti automatski na osnovu tipa elementa. Ukoliko se element sa navedenom vrednošću id-ja ne pronađe, metoda val se ne izvršava, a vrednost promenljive username će biti undefined. Ključna reč undefined označava da promenljivoj nije dodeljena vrednost. |

**Koji se karakter koristi kao prefiks unutar jQuery selektora ukoliko je potrebno selektovati elemente po nazivu klase?**

 .

 :

 #

 $

**Obrada događaja korišćenjem biblioteke jQuery**

Obrada događaja je još jedan aspekt programiranja koji je znatno olakšan korišćenjem jQuery biblioteke. Sve što je potrebno znati da bi se izvršilo registrovanje event handlera za određeni događaj jesu nazivi događaja i elementa koji izaziva događaj.

Nazivi nekih od najkorišćenijih događaja prikazani su u tabeli:

|  |  |  |  |
| --- | --- | --- | --- |
| **Miš** | **Tastatura** | **Forma** | **Document/Window** |
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave |  | blur | unload |

*Tabela 30.2 – jQuery događaji*

Registrovanje event handlera za klik na jedan HTML element se može postići na sledeći način:

|  |  |
| --- | --- |
| 1  2  3 | $("#message").click(function () {      alert("The element was clicked.");  }); |

U prikazanom kodu metodi click je prosleđena anonimna funkcija koja će biti pozvana prilikom klika na element koji ima id vrednost *message*.

Registrovanje event handlera za različite događaje može veoma lako da se postigne i korišćenjem jQuery metode on. Tako se upravo prikazani primer može transformisati na sledeći način:

|  |  |
| --- | --- |
| 1  2  3 | $("#message").on("click", function () {      alert("The element was clicked.");  }); |

Upravo korišćena metoda on ima sledeću sintaksu:

|  |  |
| --- | --- |
| 1 | $(selector).on(event,childSelector,data,function,map) |

*event* – obavezni parametar; definiše jedan ili više događaja za koje će za selektovani element biti registrovan event hander

*childSelector* – opcioni parametar; definiše da će se event handler logika aktivirati samo za decu elemente selektovanog elementa

*data* – opcioni parametar; definiše podatke koji će biti prosleđeni funkciji

*function* – obavezni parametar; definiše funkciju koja će se izvršiti kada dođe do pojave događaja

*map* – definiše mapu događaja koja omogućava registrovanje niza događaja i pripadajućih event handlera

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primer**    HTML struktura će biti sledeća:     |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11 | <html>  <head>      <title>JavaScript Demo</title>      <script src="jquery-1.9.1.min.js"></script>      <script src="script.js"></script>  </head>  <body>      <h2>JavaScript Demo</h2>      <p id="message">Welcome to <i>JavaScript</i>...</p>  </body>  </html> |   HTML strana sadrži naslov i jedan paragraf teksta. Prilikom klika na tekst, on će naizmenično menjati boju između plave i crne. Unutar head odeljka u HTML stranicu implementirana je jQuery biblioteka, ali i eksterni fajl script.js sa JavaScript kodom sledeće sadržine:   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | $(document).ready(function () {        var counter = 0;        $("#message").on("click", function (e) {            if (counter % 2 == 0) {                $(this).css('color', 'blue');            }          else {              $(this).css('color', 'black');          }            counter++;      });  }); |     Kako bi se dobila dojava o završetku kreiranja DOM strukture, u primeru se koristi ready funkcija koja prihvata anonimnu funkciju koja će se pozvati kada se učitavanje DOM-a završi. Funkcija ready se poziva nad instancom korenog elementa DOM strukture koji se selektuje korišćenjem jQuery funkcije.    Element sa atributom id *message* takođe se selektuje korišćenjem jQuery funkcije. Registrovanje event handlera za klik događaj na paragraf teksta se postiže korišćenjem on funkcije, koja prihvata naziv događaja i kod anonimne funkcije koja će se izvršiti prilikom *okidanja* eventa.    U telu anonimne funkcije, koja se poziva prilikom okidanja događaja, vrši se naizmenično bojenje teksta paragrafa. Za tako nešto koristi se jQuery funkcija css, koja je zadužena za manipulaciju CSS svojstvima elemenata. Do reference na element se dolazi korišćenjem ključne reči this. Na taj način se dolazi do elementa koji je trenutno u kontekstu, a u primeru je to element nad kojim je pozvana funkcija on. |

**jQuery Ajax**

Biblioteka jQuery značajno pojednostavljuje i proces upućivanja asinhronog zahteva serveru. Ne samo da jQuery ovaj proces pojednostavljuje, već obavlja i normalizaciju ove funkcionalnosti, što omogućava da se ona koristi i da ima efekat identičan na svim web browserima.

Osnovna jQuery metoda koja se koristi za slanje asinhronog zahteva serveru je metoda ajax. Ona ima sledeću sintaksu:

|  |  |
| --- | --- |
| 1 | $.ajax({name:value, name:value, ... }) |

Metoda ajax prihvata jedan ili više parova ključeva i vrednosti. Neki od najkorišćenijih parova ključeva i vrednosti su prikazani tabelom 30.3.

|  |  |
| --- | --- |
| **Ključ** | **Opis** |
| async | definiše da li se zahtev upućuje asinhrono ili ne |
| contentType | tip sadržaja koji se upućuje serveru |
| data | podaci/sadržaj koji se šalje serveru |
| error(xhr,status,error)  success(result,status,xhr) | funkcija koja se izvršava u slučaju greške  funkcija koja se poziva nakon uspešno poslatog zahteva i prijema podataka |
| type | tip HTTP zahteva (GET, POST) |
| url | URL na koji se upućuje zahtev |

*Tabela 30.3 – Parametri metode ajax*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primer**    HTML:   |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | <html>  <head>      <title>JavaScript Demo</title>      <script src="jquery-1.9.1.min.js"></script>      <script src="script.js"></script>  </head>  <body>      <h2>JavaScript Demo</h2>      <p id="message">Welcome to <i>JavaScript</i>...</p>      <button type="button" id="getBtn">Get Content</button>      <div id="contentContainer"></div>  </body>  </html> |   Primer  JavaScript:     |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $(document).ready(function () {        $("#getBtn").click(function(){          $.ajax({              url: "content.txt",              type: "GET",              dataType: "text",              success: function (result) {                  $("#contentContainer").html(result);              }          });      });    }); | |

|  |
| --- |
| **Napomena**    *Svi potrebni fajlovi za realizovanje prikazanog primera nalaze se u*[*prilogu*](https://www.link-elearning.com/linkdl/coursefiles/1166/jQuery%20Ajax.zip)*.* |

**jQuery Object**

U dosadašnjem toku lekcije prikazano je kako se za formiranje jQuery selektora koristi karakter dollar ($) ili ključna reč jQuery:

|  |  |
| --- | --- |
| 1 | var message = $("#message"); |

Ono što do sada nije rečeno jeste to da se na ovaj način formira jedan jQuery objekat. To je specijalna vrsta JavaScript objekta, koji je oplemenjen dodatnim svojstvima i metodama. jQuery rukuje isključivo jQuery objektima.

U dosadašnjem toku lekcije, navedena je bila i jedna ovakva linija:

|  |  |
| --- | --- |
| 1 | $(this).css('color', 'black'); |

Ovakva linija je bila definisana unutar event handler metode za click event. Značenje rezervisane reči this u jeziku JavaScript poznato je iz prethodnih lekcija. Jednostavno, ključna reč this se odnosi na objekat koji je vlasnik trenutnog konteksta. Takođe, ključna reč this je entitet JavaScript jezika.

U prikazanom primeru, može se videti da je ključna reč this smeštena između zagrada kojima prethodi karakter $. Drugim rečima, ključna reč this je parametar jQuery funkcije. S obzirom da, sama po sebi, ključna reč this referencira JavaScript objekat, takav objekat je neophodno prevesti u jQuery objekat kako bi bilo moguće koristiti sve ugrađene funkcionalnosti jQuery biblioteke.

Tako se može zaključiti da se na sledeći način formira jedan jQuery objekat:

|  |  |
| --- | --- |
| 1 | var jqueryObject = $(this); |

Nakon izvršenja prikazane linije, promenljiva jqueryObject referenciraće jedan jQuery objekat. Identična je situacija i u sledećem primeru:

|  |  |
| --- | --- |
| 1 | var message = $("#message"); |

Nakon izvršenja prikazane linije, promenljiva message referenciraće jedan jQuery objekat.