

Web Programlama

11. Hafta

02.12.2024



Mühendislik
Fakültesi
Bilgisayar Mühendisliği

Hazırlayan: Dr. Ercan Ezin

GİRİŞ

JavaScript Devam, JQuery, AJAX

Not: Bu dersin içeriđi kaynakçada belirtilen materyallerde derlenerek üretilmiştir.

Credit: Content of the course has been picked from various sources listed in the credit section and has been generously shared by the authors.

JS : LIST (ARRAY) OPERATIONS

Metod	Açıklama
<code>list.push(<i>element</i>)</code>	Elemanı listeye ekler. Link
<code>list.unshift(<i>element</i>)</code>	Elemanı en başa ekler. Link

Metod	Description
<code>list.pop()</code>	En sondaki elemanı siler ve geri döner. Link
<code>list.shift()</code>	İlk sıradaki elemanı silip geri döner. Link

Metod	Description
<code>list.indexOf(<i>element</i>)</code>	Elemanın sırasını döner, eğer yoksa -1 döner. Link

W3 school web sitesinde bulunan burda yer almayan array operasyonlarını çalışmanız önerilir.

JS: ARRAY OP. SPLICE

Add/remove element at index: [splice](#)

`list.splice(startIndex, deleteCount, item1, item2, ...)`

Remove one element at index 3:

`list.splice(3, 1);`

```
const months = ['Jan', 'Feb ', 'April', 'June'];
months.splice(3, 1);
//Sonuç: Array ["Jan", "Feb", "April"]
```

Add **element** at index 2:

`list.splice(2, 0, element);`

```
const months = ['Jan', 'Feb ', 'April', 'June'];
months.splice(2, 0, 'March');
// Sonuç : Array ["Jan", "Feb", "March", "April", "June"]
```

JS : RANDOM

JavaScript için en popüler rasgele sayı üreten fonksiyon [Math.random\(\)](#) 'dur

- `Math.random()` fonksiyonu $[0, 1)$ (0 dahil 1 dahil değil) aralığında bir sayı üretir ve geri döner.

0-max bir sayı aralığında bir sayı üretmek için şunu kullanabilirsiniz :

```
Math.floor(Math.random() * max);
```

`Math.floor` fonksiyonu sayının virgülden sonraki kısmını siler.

Ör: `Math.floor(4.8);`

//Console'da görünen değer: 4

JS: THIS KEYWORD

JavaScript için farklı bağlamlarda kullanılan this kelimesi genelde o an üzerinde işlem yapılan elemanı tanımlar.

```
console.log(this); // window
```

```
function normalFonksiyon() {  
    console.log(this);  
}  
normalFonksiyon(); // window
```

```
const obje = {  
    ad: "Ali",  
    tanimla: function() {  
        console.log(this.ad);  
    }  
};  
obje.tanimla(); // "Ali"
```

```
button.addEventListener("click", function() {  
    console.log(this); // button  
});
```

DOM : NODES

Property	Description
<u>textContent</u>	Dom içerisindeki bir öğenin ve içeriğinin metin içeriği.
<u>childNodes</u>	Bir öğenin içerdiği alt öğelerin listesi.
<u>parentNode</u>	Bir öğenin içinde bulunduğu ebeveyn öğe için referans.

```
<body>
  <h1>Şanlıurfa'nın İlçeleri</h1>
  <section>
    <p>Siverek</p>
    <p>Bozova</p>
  </section>
</body>
```

SORU: `<section>` elemanının
Ebeveyni hangisidir?
Alt öğeleri nelerdir?

Cevap: Ebeveyn-body, Alt Öğeler: p tagları

DOM: ALL AND TEXT NODES

DOM öğelerden(Nodes) oluşur ve bir kaç tür öge-altı unsur vardır.

- Element: Dom içerisinde bulunan img veya span gibi elemanlar.
- Text: DOM içerisinde bulunan metin elemanlar. Boşluk karakterleri dahil.
 - Text Bu öge-altı unsurunun altında öge bulunmaz. DOM ağacında bitim noktalarıdır(Leaf).
 - Comment: HTML içerisinde eklenmiş `<!-- -->` arasına yazılan yorumlar.

Bir ögenin hangi unsurdan meydana geldiği nodeType özelliğinden ulaşılır. ÖR: `element.nodeType`.

Normal eleman ögesi = 1, Özellik ögesi=2, Text Ögesi = 3

JS: MULTIPLE EVENTS AND BUBBLING

```
<div id="outer">
  Click me!
  <div id="inner">
    No, click me!
  </div>
</div>
```

```
const outer = document.querySelector('#outer');
const inner = document.querySelector('#inner');
outer.addEventListener('click', onOuterClick);
inner.addEventListener('click', onInnerClick);
```

div id="outer"



div id="inner"

Bir elemana ve alt öğesine aynı olay türü tanımlanırsa, ikiside aynı anda tetiklenebilir. Böyle durumlarda içeriden dışarıya doğru tetiklenme hareket eder. Yani önce inner sonra outer olay çalışır.

JS : STOPPROPAGATION

Eğer bir event tetiklendikten sonra ebeveyn öğelere tanımlı olaylar tetiklensin istemiyorsak,

`event.stopPropagation()` kullanabiliriz.

```
function onInnerClick(event) {  
  inner.classList.add('selected');  
  console.log('Inner clicked!');  
  event.stopPropagation();  
}
```

Tanımlı diğer olayları durdurur.



JS : OTHER EVENTS

1. Klavye Etkinlikleri

- keydown**: Tuşa basıldığında tetiklenir.
- keyup**: Tuş bırakıldığında çalışır.
 - Kullanım: Form doğrulama, oyun kontrolleri.

2. Fare/işaretçi Etkinlikleri

- click**: Öğe üzerine tıklama.
- mousemove**: Fare hareketi algılar.
- mouseover**: Fare bir öğenin üzerine geldiğinde tetiklenir.
 - Kullanım: Dinamik menüler, sürükle-bırak.

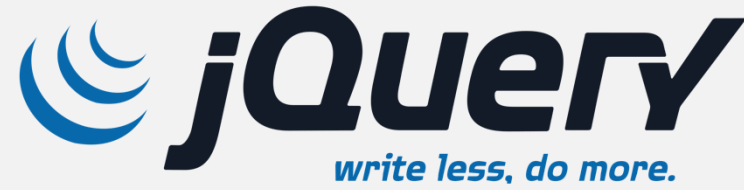
3. Animasyon Etkinlikleri

- animationstart**: Animasyon başladığında.
- animationend**: Animasyon tamamlandığında.
 - Kullanım: Görsel efektler ve animasyon geçişleri.

PRATİK-PRATİK-PRATİK

[MDN Web Docs](#)

[W3Schools](#)



jQuery, "az kod, çok iş" anlayışıyla tasarlanmış, hafif bir JavaScript kütüphanesidir. Amacı JavaScript'in web sitelerinde kullanımını kolaylaştırmaktır. Birçok satır JavaScript kodu yerine, tek satır jQuery ile aynı işlemi gerçekleştirebilirsiniz.

Öne Çıkan Özellikler:

- HTML/DOM Manipülasyonu:** DOM elemanlarını kolayca seçip düzenleyin.
- CSS Manipülasyonu:** Stilleri kolayca değiştirin.
- Etkinlik Yönetimi:** HTML olaylarını basit yöntemlerle yönetin.
- Efektler ve Animasyonlar:** Gelişmiş görsel geçişler ve efektler.
- AJAX:** Asenkron veri işlemlerini basitleştirir.
- Yardımcı Fonksiyonlar (Utilities):** Geliştirme sürecini hızlandıran araçlar.

jQuery, hemen her görev için kullanılacak zengin eklentilere sahiptir.

JQUERY : INSTALLATION

Web sayfanıza jQuery eklemek için gerekli adımlar:

[jQuery.com](https://jquery.com) adresinden indirin ve HTML <script> etiketi ile ekleyin:

```
<html>
  <head>
    <script src="jquery-3.7.1.min.js"></script>
  </head>
</html>
```

- 1. Canlı ortamlar için :** Canlı siteler için sıkıştırılmış versiyonunu eklemeniz önerilir.
- 2. Geliştirme Versiyonu:** Test ve geliştirme ortamlarında okunabilir olduğu için geliştirme versiyonu önerilir.

Alternatif olarak indirme yerine, Google gibi bir [CDN'den](#) doğrudan çağırabilirsiniz:

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  </head>
</html>
```

JQUERY : SYNTAX

jQuery için söz dizilimi HTML elemanlarını seçip, çeşitli operasyonlar yapmak için tasarlanmıştır.

Genel söz dizilimi şöyledir: **`$(selector).action()`**

- **`$`** işareti JQuery üzerindeki bir elemana erişim yapılacağını belirtir.
- **`(selector)`** bir elemanın bulunması için kullanılan alandır.
- **`action()`** ise eleman üzerinde yapılacak operasyonun yazıldığı yerdir.

Örnekler:

`$(this).hide()` - mevcut elemanı saklar(css özelliğini display=none yapar).

`$("p").hide()` - tüm p elemanlarını saklar.

`$(".test").hide()` - class="test" olan tüm elemanları saklar.

`$("#test").hide()` - id="test" id'si test olan elemanı saklar..

JQUERY : DOCUMENT READY

Sayfa tam yüklenmeden Javascript kodları çalışmaması için jQuery kodları genelde aşağıdaki gibi bir ready function içine yazılır. Soldaki ve sağdaki aynı şekilde çalışmakta, sadece sağdaki kullanım söz dizilimini biraz daha kısaltmıştır.

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

```
$(function(){  
  
    // jQuery methods go here...  
  
});
```

JQUERY : ELEMENT SELECTOR

Sayfada bulunan bir butona tıklandığında tüm paragraf elemanlarını kapatmak için veya id olarak test olan bir elemanı kapatmak için veya sınıfı test olan elemanı kapatmak için şunlar yapılır;

```
$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
});
```

```
$(document).ready(function(){  
  $("button").click(function(){  
    $("#test").hide();  
  });  
});
```

```
$(document).ready(function(){  
  $("button").click(function(){  
    $(".test").hide();  
  });  
});
```


MORE SELECTOR EXAMPLES

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code><p></code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code><p></code> element
<code>\$("ul li:first")</code>	Selects the first <code></code> element of the first <code></code>
<code>\$("ul li:first-child")</code>	Selects the first <code></code> element of every <code></code>
<code>\$("[href]")</code>	Selects all elements with an <code>href</code> attribute
<code>\$("a[target='_blank']")</code>	Selects all <code><a></code> elements with a <code>target</code> attribute value equal to <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Selects all <code><a></code> elements with a <code>target</code> attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code>
<code>\$("tr:even")</code>	Selects all even <code><tr></code> elements
<code>\$("tr:odd")</code>	Selects all odd <code><tr></code> elements

JQUERY-IMPORTANT FUNCTIONS

Bir elemanı soluklaştırıp yok etmek için.

```
$("#button").click(function(){
  $("#div1").fadeIn();
  $("#div2").fadeIn("slow");
  $("#div3").fadeIn(3000);
});
```

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

Bir elemana kayma efekti vererek yok etmek için.

```
$("#flip").click(function(){
  $("#panel").slideDown();
});
```

- slideDown()
- slideUp()
- slideToggle()

```
$("#button").click(function(){
  $("#div").animate({
    left: '250px',
    opacity: '0.5',
    height: '150px',
    width: '150px'
  });
});
```

Bir elemana
animasyon
uygulamak için
parametreler.

```
$("#p1").css("color", "red")
.slideUp(2000)
.slideDown(2000);
```

Chaining sayesinde
tek satırda birden
çok metod çağrılır

```
$("#button").click(function(){
  $("#p").hide("slow", function(){
    alert("The paragraph is now hidden");
  });
});
```

Bir efekt yada olay bittikten sonra çalışmasını
istediğimiz kodları yazmak için **Callback**
kullanırız.

JQUERY HTML EFFECTS

GET Fonksiyonu ile değer almak için.

```
$("#btn1").click(function(){
    alert("Text: " + $("#test").text());
});
$("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
});
```

SET Fonksiyonu ile değer atamak için.

```
$("#btn1").click(function(){
    $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
});
```

```
function appendText() {
    var txt1 = "<p>Text.</p>"; // Create element with HTML
    var txt2 = $("<p></p>").text("Text."); // Create with jQuery
    var txt3 = document.createElement("p"); // Create with DOM
    txt3.innerHTML = "Text.";
    $("body").append(txt1, txt2, txt3); // Append the new elements
}
```

jQuery ve diğer metodlar ile eleman oluşturma.

JQUERY HTML EFFECTS-2

Sınıfı test veya demo olan tüm elemanları DOM'dan silmek için

```
$("#p").remove(".test, .demo");
```

- **addClass()** - Seçilen elemanlara bir veya daha fazla sınıf ekler
- **removeClass()** - Seçilen elemanlardan bir veya daha fazla sınıfı kaldırır
- **toggleClass()** - Seçilen elemanlarda sınıf ekleme/kaldırma işlemini değiştirir.
- **css()** - Stil özelliğini ayarlar veya döndürür.

```
$("#button").click(function(){  
    $("#h1, h2, p").addClass("blue");  
    $("#div").addClass("important");  
});
```

```
$("#button").click(function(){  
    $("#h1, h2, p").removeClass("blue");  
});
```

```
$("#button").click(function(){  
    $("#h1, h2, p").toggleClass("blue");  
});
```

```
$("#p").css({"background-color": "yellow", "font-size": "200%"});
```

JQUERY IMPORTANT REMINDER

- ❖ jQuery kodlarınız her zaman farklı tarayıcılarda çalışmayabilir. O yüzden mutlaka test etmelisiniz.
- ❖ jQuery CDN olarak eklendiğinde dışa bağımlılık oluşmakta, bu yüzden dikkatli olmakta fayda var.
- ❖ jQuery eklentileri belli bir süre sonra yönetilmesi zor olan hallere girmekte. O yüzden JQuery ile çözülebilecek basit şeyler için yüklü eklentilerden kaçınin.
- ❖ jQuery hatalarını bulabilmek için console.log'dan faydalanın.
- ❖ jQuery bir yerde hata verdiğinde geri kalan tüm sayfaya etki edebilir. O yüzden herşeyi JQuery ile yazmamak iyi bir pratiktir.

jQuery için pratik yapıp hakimiyet sağlamak istiyorsanız bu [linkteki](#) örnekleri tekrar edip kendi projelerinize uygulayın.

JQUERY : AJAX

AJAX = Asynchronous JavaScript and XML.

Bir HTML sayfasının yenilenmesine gerek duymadan sunucu ile bilgi alışverişini sağlayan jQuery metodudur(jQuery kütüphanesi olmadan da çalışır ancak çok [kod](#) eklemek gerekir). Günümüzdeki modern uygulamaların hemen hemen hepsi kullanmaktadır.

```
$("#button").click(function(){  
  $("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){  
    if(statusTxt == "success")  
      alert("External content loaded successfully!");  
    if(statusTxt == "error")  
      alert("Error: " + xhr.status + ": " + xhr.statusText);  
  });  
});
```

JQUERY : AJAX (GET)

```
$("#button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

```
<html>
<script type="text/javascript" src="jQuery.js"></script>
<script type="text/javascript">
    $(document).ready(function(){
        $("#submit").click(function(){
            $.ajax({
                type: 'GET',
                url: "http://localhost/page1.html",
                success: function(data){
                    alert(data);
                }
            });
            return false;
        });
    });
</script>
</head>

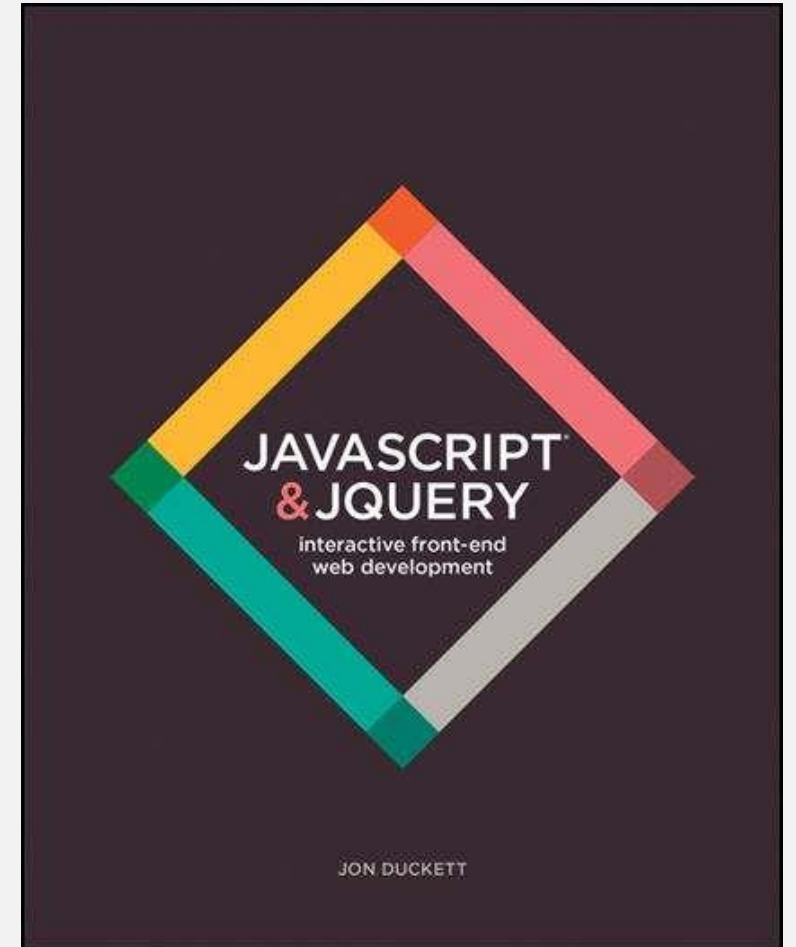
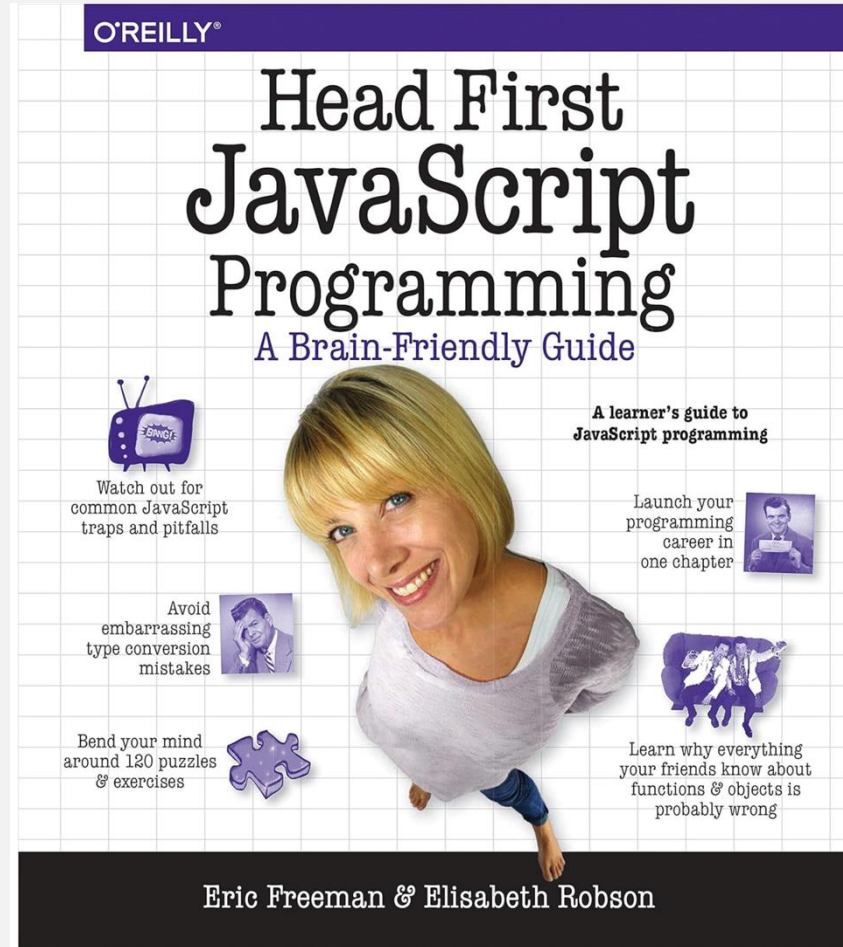
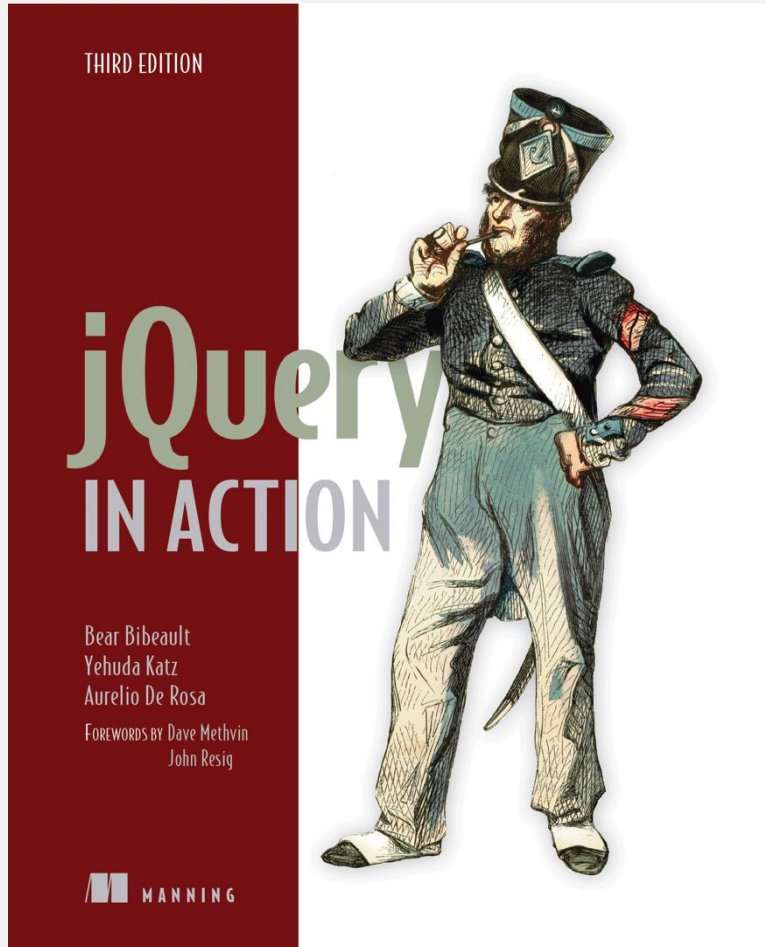
<body>
<input type="button" id="submit" value="submit" />
</body>
</html>
```

JQUERY: AJAX (POST)

```
$("#button").click(function(){
    $.post("demo_test_post.asp",
    {
        name: "Donald Duck",
        city: "Duckburg"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

```
$.ajax({
    url: "url",
    type: "POST",
    dataType: "json",
    contentType: "application/json; charset=utf-8",
    data: JSON.stringify({ name: 'value1', email: 'value2' }),
    success: function (result) {
        // when call is successful
    },
    error: function (err) {
        // check the err for error details
    }
}); // ajax call closing
```


JS AND JQUERY RESOURCES



JS ve jQuery konusunda yukarıdaki kaynaklardan pratikler yapmanız önerilir.

CHALLENGE

Tic-Tok-Toe oyununu jQuery kullanarak yapalım.



EOF

REFERENCES / CREDITS

1. Stepp M, Miller J, Kirst V. Web Programming Step by Step. Step by Step Publishing; 2012.
2. CS193X Web Programming Fundamentals Course Slides at Stanford Uni by Victoria Kirst
3. <https://www.w3schools.com/jquery/>