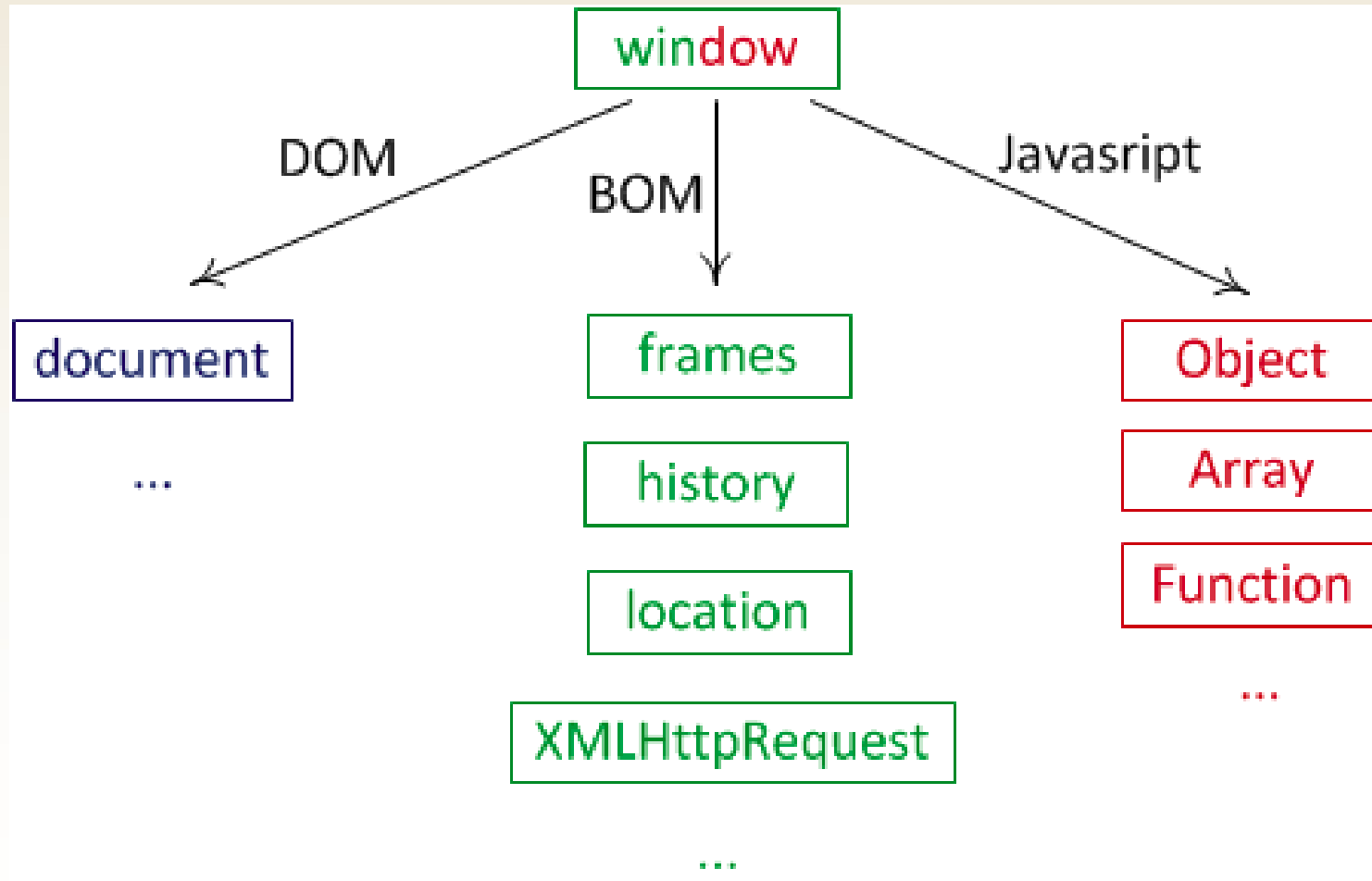


Documents & Events



Tarayıcı Ortamı



Tarayıcı Ortamı

- Tarayıcı içerisinde bir grup **nesne hiyerarşisi** mevcuttur
- Bu nesne hiyerarşisi uygulama geliştiriciler tarafından çalışma zamanında **manipülasyona** tabi tutulabilir
- Hiyerarşinin en üstünde **window** nesnesi mevcuttur
- Window **global object**'dir

Nesne Hiyerarşisinin Elemanları

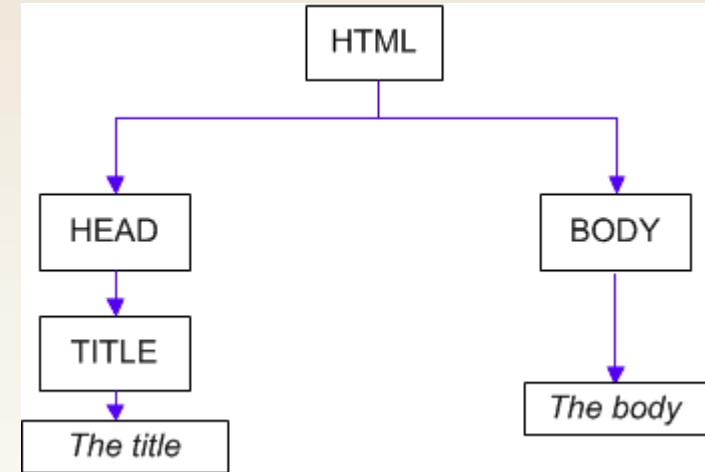
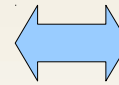
- Document Object Model (**DOM**)
 - Sayfa içeriğini represente eder
- Browser Object Model (**BOM**)
 - Tarayıcıyı kontrol etmeyi sağlar
- **Javascript** nesne ve fonksiyonları
 - Kendine has nesne ve fonksiyonlar sunar
 - DOM ve BOM nesnelerine erişmeyi, bunlar üzerinde işlem yapmayı sağlar

DOM

- Web dokümanının içeriğini bir **ağaç hiyerarşisi** şeklinde ele almayı sağlar
- Her bir node bir **HTML elemanına**, **text** veya **whitespace** node'a karşılık gelir
- Node'lar arasında **parent-child ilişkisi** vardır
- DOM'un ana amacı **herşeyi bir nesne** olarak ele almayı sağlamaktır
- Böylece herhangi bir node'a erişip bu node'un **özellikleri değiştirilebilir**

DOM

```
<html>  
  <head>  
    <title>The title</title>  
  </head>  
  <body>  
    The body  
  </body>  
</html>
```

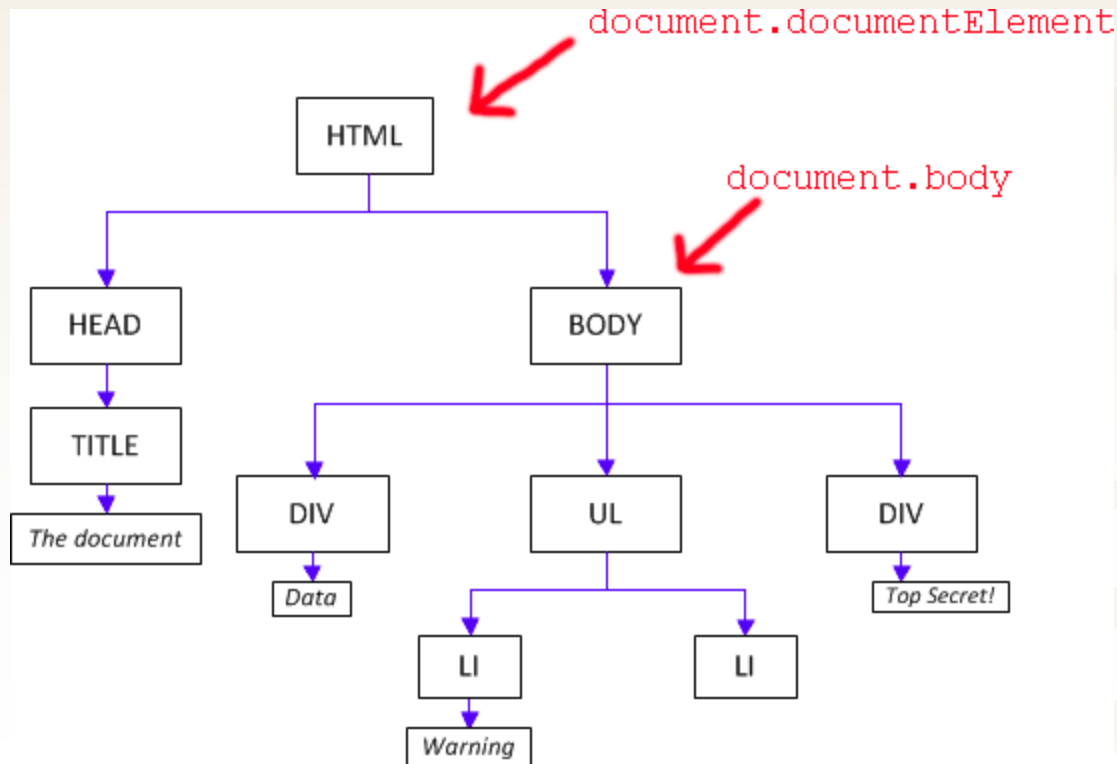


```
document.body.style.backgroundColor = 'red';
```

Ayrıca DOCTYPE, comment gibi node'larda mevcuttur
DOCTYPE, tarayıcıya html sayfanın hangi HTML versiyonu ile
Oluşturulduğunu vs bildirir

DOM Üzerinde Dolaşma

Document.documentElement her zaman için en tepedeki kök elemana ulaşmayı sağlar. Diğer bir başlangıç noktası `document.body`'dir. Ancak `<head>` elemanındaki javascript'lerde `document.body`'ye erişmeye çalışıldığı vakit NULL dönülecektir. Çünkü bu aşamada daha body yüklenmemiştir

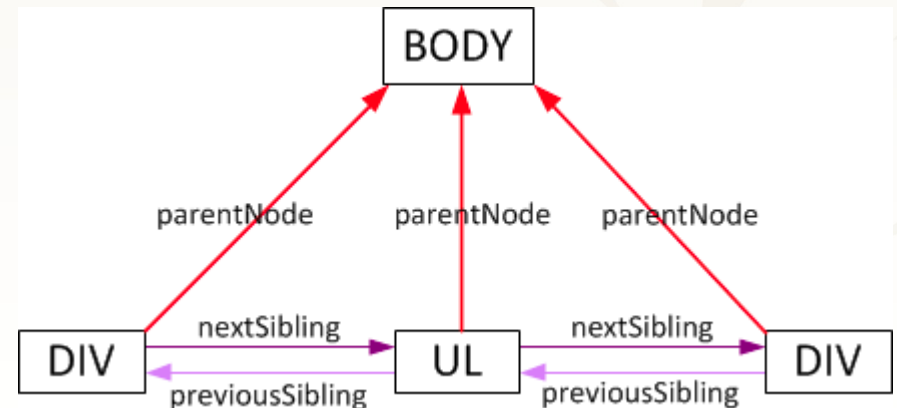
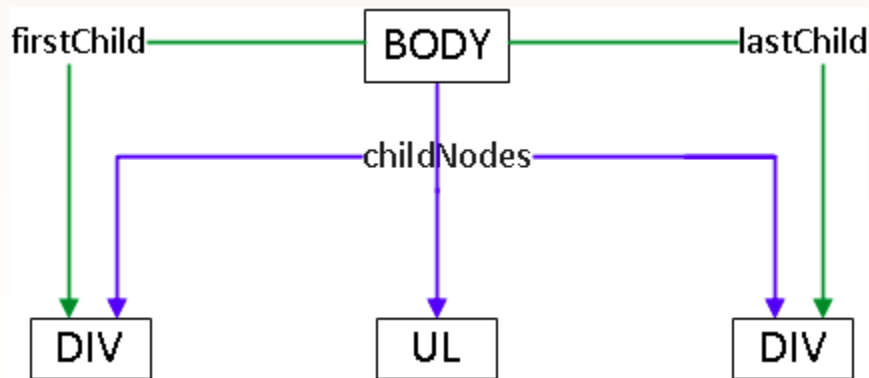


DOM Üzerinde Dolaşma

- **document.body.childNodes**
 - Whitespace node'larda dahil bütün node'lar üzerinde dolaşma imkanı sunar
 - IE < 9'da whitespace node mevcut değildir
- **document.body.children**
 - Sadece element node'lar üzerinde dolaşma imkanı sunar
 - Whitespace, text node'ları hariç tutar

DOM Üzerinde Dolaşma

- firstChild, lastChild
- parentNode, previousSibling, nextSibling
- gibi property'lerle diğer node'lar üzerinde dolaşma imkanı da vardır



DOM Üzerinde Dolaşma

- Tarayıcı node'lar arasındaki bu ilişkilerin **doğruluğunu garanti** eder
- Dolaşımı sağlayan bu property'lerin hepsi **salt-okunurdur**
- Eğer herhangi bir property **mevcut değil** ise **NULL** dönülür

DOM Node Properties

- Her bir **node**'un **özellikleri** mevcuttur
- Bu özelliklerden bazıları **salt-okunur**, bazıları ise **değiştirilebilirdir**
- **nodeType**
- **nodeName, tagName**
- **innerHTML**
- **nodeValue** temel özelliklerdir

nodeType

- 12 farklı node tipi vardır
- En sık kullanılanları ELEMENT_NODE ve TEXT_NODE'dur

```
interface Node {
    const unsigned short ELEMENT_NODE = 1;
    const unsigned short ATTRIBUTE_NODE = 2;
    const unsigned short TEXT_NODE = 3;
    const unsigned short CDATA_SECTION_NODE = 4;
    const unsigned short ENTITY_REFERENCE_NODE = 5;
    const unsigned short ENTITY_NODE = 6;
    const unsigned short PROCESSING_INSTRUCTION_NODE = 7;
    const unsigned short COMMENT_NODE = 8;
    const unsigned short DOCUMENT_NODE = 9;
    const unsigned short DOCUMENT_TYPE_NODE = 10;
    const unsigned short DOCUMENT_FRAGMENT_NODE = 11;
    const unsigned short NOTATION_NODE = 12;
}
```

```
var childNodes = document.body.childNodes
for(var i=0; i<childNodes.length; i++) {
    if (childNodes[i].nodeType != 1) continue;
    alert(childNodes[i]);
}
```

nodeName, tagName

- Her ikisi de **element node**'un ismini döner
- Dönen değer **büyük harflidir**
- Eğer tarayıcı **XML modunda** parse etmiş ise **case sensitive** dönebilir
- **Element node** için her ikisi de **aynı değeri** döner
- Diğer node tiplerinde **tagName undefined** veya comment node'da IE için **“!”** dir

innerHTML

- **HTML 5'in bir parçasıdır**
- Element node'ların **içeriğine text formatında** erişmeyi sağlar
- Herhangi bir element node'un innerHTML property'si üzerinden içeriği **değiştirilebilir**

```
alert( document.body.innerHTML );  
document.body.innerHTML = 'Hello World!';
```



Eğer script <head> elemanı içerisinde ise document.body NULL olduğu unutulmamalıdır

innerHTML

- **IE** için COL, COLGROUP, FRAMESET, HEAD, HTML, STYLE, TABLE, TBODY, TFOOT, THEAD, TITLE, TR elemanlarının innerHTML property'si **alt-okunur**
- elem.innerHTML += "New text" ile **içeriği append etmek** syntaktik olarak doğru olsa bile çalışma zamanında bu **içeriğin reload edilmesine** neden olmaktadır

nodeValue

- innerHTML sadece element node'larda mevcuttur
- **Diğer node tipleri** için ise nodeValue kullanılabilir
- Örneğin **text node**'un içeriği **nodeValue** ile değiştirilebilir

Node Attributes & Properties

- Herhangi bir DOM node **attribute** ve **property**'lere sahip olabilir
- Bu ikisi **birbirleri ile alakalı** olmasına rağmen **farklı şeylerdir**
- Öncelikle DOM node bir JS nesnesi olduğu için üzerinde **custom metot ve property tanımlanması** normaldir
- Custom metot ve property'ler **sadece JS için**'dir, HTML tarafını etkilemez

Node Attributes & Properties

- DOM node'ları ayrıca **HTML attribute**'larına erişim imkanı da sunar
- Sadece **String değer** içerirler
- Attribute **isimleri büyük harflidir**
- **element.attributes** property'si ile bir element node'un attribute'larına erişilebilir
- **innerHTML** ile de erişilebilirler

```
elem.hasAttribute('ID');  
elem.getAttribute('ID');  
elem.setAttribute('ID', 'myTable');  
elem.removeAttribute('ID');
```

Node Attributes & Properties

- Her bir DOM node tipinin kendine has **standart property'leri** mevcuttur
- Bu property'ler **attribute'lar** olarak **sync** edilir
- Fakat her bir property'de değerlerin **bire bir attribute** tarafında olması beklenmemelidir
- Property'ye göre **sadece attribute** mevcut olabilir, **değeri olmayabilir**, farklı olabilir

DOM Değişiklikleri

- Sayfaları dinamik yapmak için DOM üzerinde **çalışma zamanında modifikasyon** yapmak önemlidir

```
var div = document.createElement('div');
```

```
var text = document.createTextNode('Hello world!');
```

Normal bir element veya text node yaratmak mümkündür

```
div.cloneNode(true);
```

```
text.cloneNode(false);
```

Mevcut herhangi bir node clone'lanarak yeni bir kopyası oluşturulabilir. Flag, clone işleminin çocuk node'lara inip inmeyeceğini kontrol eder.

```
div.appendChild(text);
```

```
document.body.appendChild(div);
```

```
document.body.insertBefore(  
    document.createTextNode("How are you?"),div);
```

Bir node başka bir node'a eklenebilir

DOM Değişiklikleri

- Mevcut node'ları silmek veya başka bir node ile değiştirmek de mümkündür

```
document.body.removeChild(childNode);
document.body.replaceChild(newNode, oldNode);
```

- Text node create ettikten sonra bunu div elemana append etmek ile div elemanın innerHTML'ini set etmek arasında da ufak bir fark vardır

```
var text = document.createTextNode("<b>Hello world!</b>");
div.appendChild(text);
div.innerHTML= "<b>Hello world!</b>";
```

createTextNode
html tag'lerini escape
yapar. innerHTML'de
ise html tag'leri düzgün
biçimde render edilir

DOM Değişiklikleri

- **document.write**, text içeriğin doğrudan doküman içerisine yazılmasını sağlar
- Herhangi bir **HTML içerik** sayfanın içerisine eklenebilir
- HTML içeriğin **tam veya geçerli** olmasına da **gerek yoktur**
- Yazma işlemi **dokümanın yüklenmesi tamamlanmadan** gerçekleşmelidir
- Dokümanın yüklenmesi tamamlandıktan sonra yapılacak write **mevcut içeriğin silinmesine** neden olacaktır

DOM İçerisinde Arama

- DOM içeriğini **doğrudan sorgulamak ve elemanlara erişmek** de mümkündür
- DOM içinde arama **5 farklı metot** ile gerçekleştirilebilir

```
var table = document.getElementById("table1");
var tables = document.getElementsByTagName("table");
var nodeList = document.getElementsByName("myName");
var nodeList = document.querySelectorAll("UL > LI");
```

Sorgu sonucu birden fazla eleman dönülebilir

CSS 3 sorgusu ile çalışır. QuerySelector ilk elemanı döner.

Table Element

- DOM içerisinde table elemanına eriştikten sonra **table'in satırlarına, her bir satırın da hücrelerine** erişmek mümkündür

```
<table>
  <tr>
    <td>one</td><td>two</td>
  </tr>
  <tr>
    <td>three</td><td>four</td>
  </tr>
</table>

<script type="text/javascript">
  var table = document.body.children[0]
  alert(table.rows[0].cells[0].innerHTML) // "one"
</script>
```


Form Element

- Form doküman içerisinde **index** veya **ismi** ile erişilebilir
- form.elements[index/name]** ile Form elemanlarına da erişmek mümkündür

```
<form name="myForm">
    <input name="firstName" value="Kenan">
    <input name="lastName" value="Sevindik">
</form>

<script type="text/javascript">
    var form = document.forms.myForm // veya document.forms[0]

    var element = form.elements.firstName // veya form.elements[0]

    alert(element.value) // "Kenan"
</script>
```

Form Element

- Aynı isme sahip form elemanlarına da erişmek mümkündür
- Select form elemanının option'larına da erişilebilir
- Her bir form elemanının ait olduğu form'a bir referansı vardır

```
<form>
  <input type="radio" name="gender" value="male">
  <input type="radio" name="gender" value="female">
  <select name="specialty">
    <option name="radiology" value="radiology">Radyoloji</option>
    <option name="surgeon" value="surgeon">Ameliyat</option>
  </select>
</form>

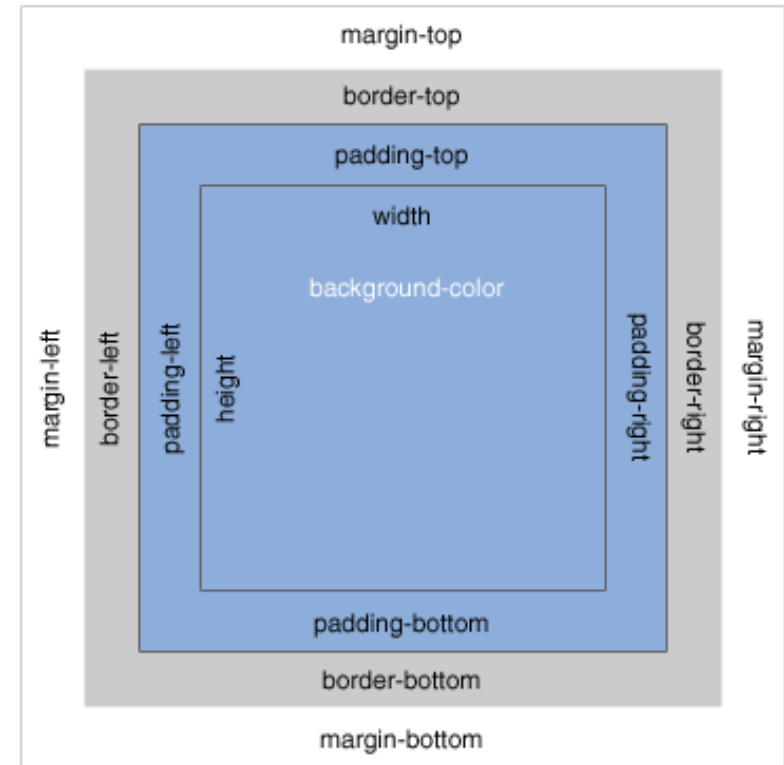
var form = document.forms[0]
var elements = form.elements.gender
alert(elements[0].value) // "male"
alert(form.elements['specialty'].options[0].value) // "radiology"
alert(elements[0].form == form);
```

View & Position



CSS Box Model

- Bütün HTML elemanları **dikdörtgen bir kutucuk** içerisinde yer alırlar
- Bu kutunun şekli, ebatları ve dokümanda nasıl konumlanacağı yönetilebilir



CSS Box Model

- Bu dikdörtgen kutunun ebatlarını ve lokasyonunu yönetmeyi sağlayan **CSS property**'ler mevcuttur
 - width
 - height
 - padding (left,right,top,bottom)
 - margin (left,right,top,bottom)
- CSS property'leri **element.style** ile set edilebilir, yada **getComputedStyle()/currentStyle** ile erişilebilir

CSS Box Model

- Javascript içerisinde kullanılan bütün metrik değerler “**pixel**” dir ve “px” son eki kullanılmaz
- Temel metrik CSS property'leri üzerinden çalışan **diğer property'ler** de mevcuttur
- **clientWidth/Height**: content area + padding – scroll area
- **scrollWidth/Height**: content area + padding + scroll area

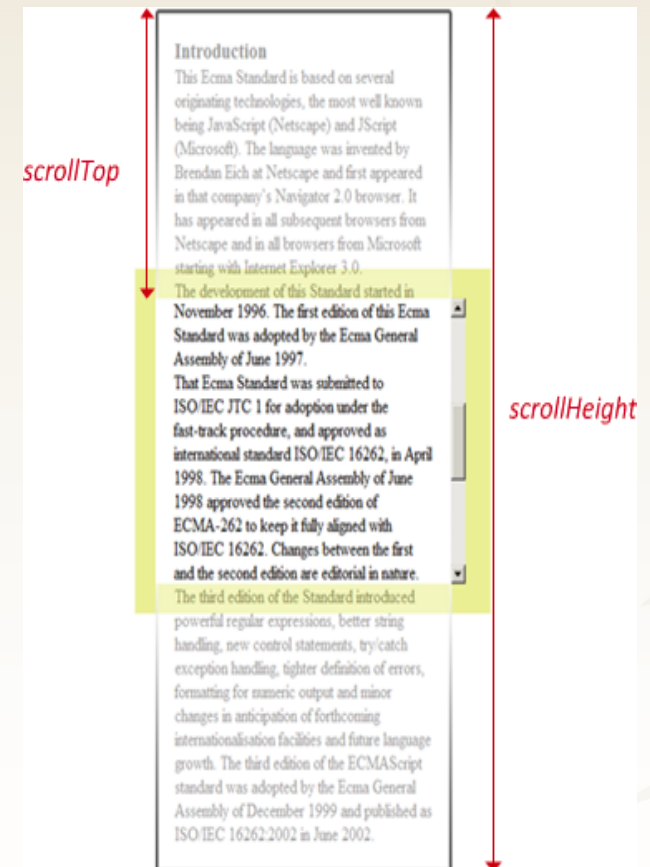
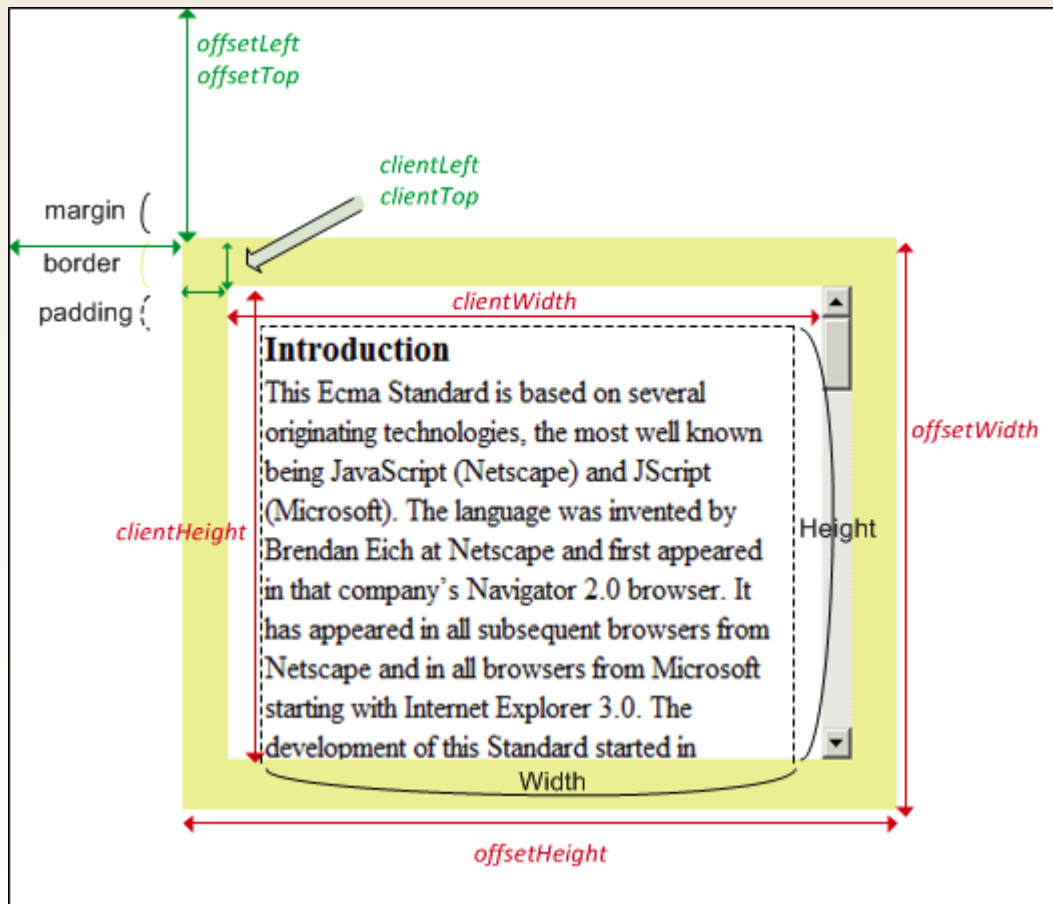
CSS Box Model

- **scrollTop/Left:** Elemanın scroll dışında kalan bölümlerini gösterir, diğer property'lerden farklı olarak değiştirilebilir
- **offsetWidth/Height:** elemanın tüm ebatıdır, border dahildir, fakat margin hariçtir
- **clientTop/Left:** Kutunun sol üst köşesinden uzaklıktır

CSS Box Model

- **offsetParent:** Elemanların sayfa içerisindeki layout'una göre en yakın parent elemanın ebatıdır
- **offsetTop/Left:** Bir elemanın parent'ından relatif olarak uzaklığını belirtir

CSS Box Model



Events

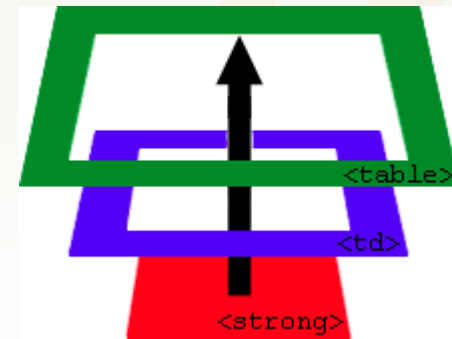


Event Delegation

- **Kabarcık (bubbling) tekniği** event'lerin handle edilmesini daha basit hale getirir
- Temel bir **Javascript örüntüsüdür**

```
table.onclick = function(event) {
    event = event || window.event
    var target = event.target ||
                  event.srcElement
```

```
while (target != table) {
    if (target.nodeName == 'TD') {
        toggleHighlight(target)
    }
    target = target.parentNode
}
}
```



Eğer event target <TD> elemanı değil ise bir üst node'a Geçilir. Bu target node table olana kadar devam eder.

Mouse Event'leri

- İki türüdür
- **Basit event'ler:** mousedown, mouseup, mouseover, mouseout, mouse move
- **Karmaşık event'ler:** click, contextmenu, dblclick
- Bir action, birden fazla mouse event'in aynı anda fırlatılmasına neden olabilir

```
document.getElementById('btn').onclick = function() {
    alert('click!')
}

document.getElementById('btn2').oncontextmenu = function() {
    alert('right click!')
}
```

Mouse Event'leri

- Örneğin bir button click sırasında sırası ile mousedown, mouseup, click event'leri meydana gelir
- Ya da double click sırasında click işleminde meydana gelen event'ler iki kere tekrarlar
- Event nesnesinde **target** ve **relatedTarget** property'leri vardır
 - **target**: mouse'un üzerinde olduğu elemanı gösterir
 - **relatedTarget**: mouse'un geldiği elemanı gösterir

Mouse Event'leri

- Event nesnesinde **which** ve **button** property'lerinde click event'in meydana geldiği mouse button'unun bilgisi yer alır
- W3C için button property'si değeri
 - 0 : sol tuş, 1 : orta tuş, 2 : sağ tuş
- IE için ise button property değeri bit pozisyonu ile mask'lanarak bulunur
 - $\text{button} \& 1$: sol tuş, $\text{button} \& 4$: orta tuş, $\text{button} \& 2$: sağ tuş

Mouse Koordinatları

- Window veya dokümana relatif biçimde mouse koordinatları elde edilebilir
- **clientX/clientY**: window'a relatif
 - Mouse pozisyonu değişmeden yapılan scroll değerlerini değiştirmez
- **pageX/pageY**: dokümana relatif
 - Scroll işlemi değerlerini değiştirir

Focus

- **Focus event**, bir **elemana odaklanıldığında** fırlatılır
- Bütün elemanlara odaklanmak mümkün değildir
- **INPUT** elemanı odaklanılabilir, fakat **DIV** elemanı odaklanılamaz
- Focus olabilen bir eleman **focus()** metodu ile programatik olarak da odaklanılabilir

Focus Örneği

```
<input type="text" value="E-mail" class="untouched" value="Enter Your E-mail" />

<style>
    .untouched {
        color: gray
    }
</style>

<script>
    var input = document.getElementsByTagName('input')[0];

    input.onfocus = function() {
        if (this.className == '')
            return ;

        this.className = '';
        this.value = '';
    }
</script>
```

Tabindex

- Odaklanılamayan bir elemana **tabindex=<number>** eklenerek bu kabiliyet kazandırılabilir
- **Tab tuşu** ile tabindex'i **küçükten büyüğe** doğru olan elemanlara odaklanılabilir
- **tabindex="0"** ise bu sıralamada elemanın sonuncu olduğunu belirtir
- **tabindex="-1"** ise elemanı odaklanılabilir yapar, ancak tab key bu elemanı pas geçer

Blur

- Focus işleminin tersidir
- Eleman **odağını kaybettiğinde** blur event fırlatılır
- **blur()** metodu ile eleman programatik olarak da blur yapılabilir
- Focus ve blur event'leri kabarcık tekniği ile ele alınamazlar
- Bunun yerine **focusin** kullanılmalıdır

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

