

مقدمه‌ای سریع بر جاوااسکریپت

سید کاوه احمدی

مقدمه

- جاوا اسکریپت یکی از زبان‌های مهم برنامه‌نویسی وب و سمت کلاینت است که در سال ۱۹۹۵ معرفی شده است و وظیفه ابتدایی آن فقط معتبرسازی عناصر فرم بود.
- از لحاظ `syntax`، جاوااسکریپت مثل `C` می‌باشد.
- اما به طور کلی با آن فرق دارد.
- در جاوا اسکریپت، توابع می‌توانند به عنوان داده به کار روند: به تابع‌های دیگر به عنوان پارامتر ارسال شوند و یا مقدار `return` یک تابع باشند.
- بازگشت یک تابع به عنوان مقدار `return` تابعی دیگر، سبب ایجاد مبحث فوق‌العاده حرفه‌ای به نام `closure` می‌شود.
- یکی از کارهایی که کلاژرها برای ما انجام می‌دهد، بالا بردن قابلیت‌های جاوااسکریپت در برنامه‌نویسی شی‌گرا می‌باشد.

- می‌توانید کلیه کدهای جاوااسکریپت خود را درون یک فایل خارجی با پسوند `.js` ذخیره نمایید و آن‌را در صفحاتی که نیاز دارید، داخل نمایید.

– `<script type="text/javascript" src="scripts.js"></script>`

- استفاده از فایل‌های جاوااسکریپت خارجی شدیداً توصیه می‌شود.

کلیات اصلی زبان

- حساسیت نسبت به حروف

– جاوااسکریپت نسبت به حروف بزرگ و کوچک حساس است.

- فضاهاى سفید

– جاوااسکریپت از فاصله‌ها (`space`)، تب‌ها (`tab`)، و خطوط جدید (`newline`)،

صرف نظر می‌کند. شما مختارید هر طور که می‌خواهید کدتان را فرمت کنید.

- عبارات جاوااسکریپت، توسط یک سمی کولن (;) به پایان می‌رسند.
- اگر هر خط از کد جاوااسکریپت، درون یک خط جداگانه قرار دارد، می‌توان از این سمی کولن صرف‌نظر کرد.
- صرف‌نظر کردن از سمی کولن‌ها، کار خوبی نیست.

توضیحات

- `// slashslash line comment`
- `/*
 slashstar
 block
 comment
*/`

- قبل از استفاده از متغیرها، بهتر است آن‌ها را تعریف کنیم. این کار توسط `var` انجام می‌شود.
- متغیر بدون نوع هستند و در زمان‌های مختلف می‌توانند مقادیر مختلفی داشته باشند
- `var a1 = 'Hello Word!';`
- `a1 = 55;`

- اولین کاراکتر یک متغیر می‌تواند حرف، `_` یا `$` باشد
- سایر کاراکترها علاوه بر بالایی‌ها می‌تواند عدد نیز باشد.
- `var test;`
- `var $test;`
- `var $1;`
- `var _test2;`

■ در جاوا اسکریپت پنج نوع اصلی وجود دارد:

– **undefined, null, boolean, number, string**

نوع داده **undefined**

■ این نوع فقط شامل یک مقدار می‌شود **undefined**. متغیری که اعلان می‌شود ولی مقداردهی اولیه نمی‌شود به صورت پیش فرض از نوع است.

- عملگر `typeof` یک پارامتر می‌گیرد (یک متغیر یا یک مقدار) و نوع آن را باز می‌گرداند

```
var oTemp = 5;

alert (typeof oTemp) ; //outputs number
```

نکته

- متغیری که اعلان می‌شود و مقدار نمی‌گیرد با متغیری که اصلاً اعلان هم نشده است کاملاً متفاوت است. هر چند که عملگر `typeof` بین این دو تفاوتی قائل نمی‌شود. و برای هر دو متغیر مقدار `Undefined` را باز می‌گرداند.

```
var oTemp ;
alert(typeof oTemp) ; // outputs "Undefined"
alert(typeof oTemp2) ; // outputs "Undefined"
```

- استفاده از `oTemp2` در هر جایی به جز `typeof` منجر به رخ دادن خطا می‌شود.

```
//make sure this variable isn't defined
//var oTemp2;
//try outputting
alert(oTemp2 == undefined); //causes error
```

- مقدار `undefined` زمانی که یک تابع مقداری را برنگرداند هم بازگردانده می‌شود.

```
function Testfunc () {
    // leave the function blank
}
alert(TestFunc() == undefined); //outputs "true"
```

نوع داده `null`

- یک مقدار ویژه می‌گیرد: `null`

- از نظر جاوااسکریپت نوع `undefined` یکی از مشتقات `null` است:

```
- alert(null == undefined); //outputs "true"
```

- اگر چه این دو معادل یکدیگرند اما معانی خاصی دارند:

- `undefined` موقعی به یک متغیر نسبت داده می‌شود که اعلان شود ولی مقداردهی نشود. در حالی که متغیری وقتی از نوع `null` است که شامل شیء ای باشد که وجود ندارد.

- انواع اصلی `boolean`، `number` و `string` اشیای کاذب هستند، به این معنی که که دارای خصوصیت‌ها و متدهای مشترک و منحصر به فردی می‌باشند.

– به عنوان مثال برای بدست آوردن طول یک رشته می‌توان از خاصیت `length` استفاده کرد

```
var sColor = "blue" ;  
alert (sColor.length) ; //outputs "4"
```

تبدیل به رشته

- سه نوع داده‌ی `boolean`، `number` و `string` متدی به نام `toString()` برای تبدیل به رشته دارند.
- این متد برای متغیرهای از نوع `boolean` یکی از مقادیر `"true"` یا `"false"` را بسته به مقدار متغیر بر می‌گرداند.
- برای متغیرهای از نوع `number` رشته‌ای حاوی آن عدد را بر می‌گرداند.

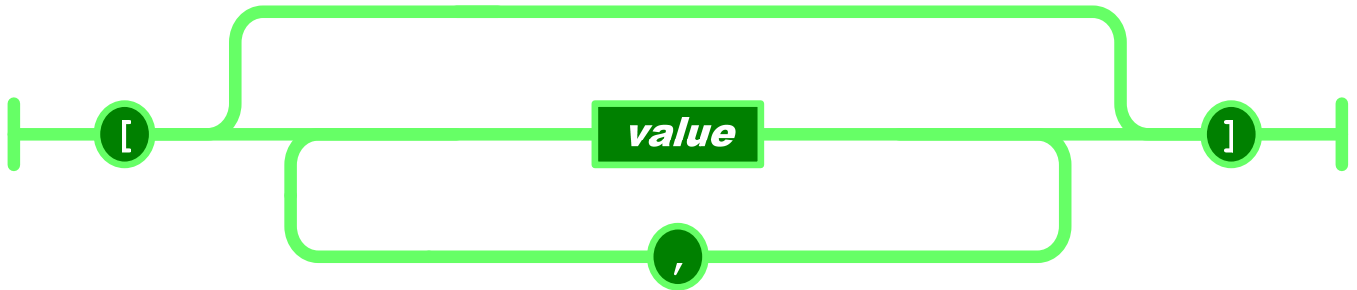
- **parseInt()**
- **parseFloat()**

```
- var iNum1 = parseInt("1234blue"); //1234
- var iNum3 = parseInt("22.5"); //22
- var iNum4 = parseInt("blue"); //NaN

- var fNum1 = parseFloat("1234blue"); //1234.0
- var fNum3 = parseFloat("22.5"); //22.5
- var fNum4 = parseFloat("22.34.5"); //22.34
- var fNum6 = parseFloat("blue"); //NaN
```

روش دیگر تبدیل نوع

- **Boolean()**
- **Number()**
- **String()**



■ تعریف آرایه

- `var aValues = new Array();`
- `var bValues = new Array(20);`

```
- var aColors = new Array();  
- aColors[0] = "red";  
- aColors[1] = "green";  
- aColors[2] = "blue";
```

یا: ■

```
- var aColors = new Array("red", "green", "blue");
```

یا: ■

```
- var aColors = ["red", "green", "blue"];
```

کار با آرایه

■ به دست آوردن طول آرایه

```
- aColors.length
```

■ تبدیل رشته به آرایه

```
- var sColors = "red,green,blue";
```

```
- var aColors = sColors.split(",");
```

```
var aColors = ["red", "green", "blue"];  
alert(aColors.toString()); // "red,green,blue"  
alert(aColors.valueOf()); // "red,green,blue"  
alert(aColors.toLocaleString()); // "red,green,blue"
```

join

```
var aColors = ["red", "green", "blue"];  
alert(aColors.join(", ")); // "red,green,blue"  
alert(aColors.join("-spring-")); // "red-spring-green-spring-blue"  
alert(aColors.join("] [")); // "red][green][blue"
```

```
var aColors = ["red", "green", "blue"];
var aColors2 = arr.concat("yellow", "purple");

// "red,green,blue,yellow,purple"
alert(aColors2.toString());

// "red,green,blue"
alert(aColors.toString());
```

مرتب‌سازی آرایه

```
var aColors = ["red", "green", "blue"];
aColors.reverse();
alert(aColors.toString()); // "blue,green,red"

var aColors = [3, 32, 2, 5]
aColors.sort();
alert(aColors.toString()); // "2,3,32,5"
```

حذف و درج در میانه‌های آرایه

حذف دو عنصر اول آرایه `arr`

```
- arr.splice(0, 2);
```

درج بدون حذف

درج عناصر در موقعیت ۲ آرایه

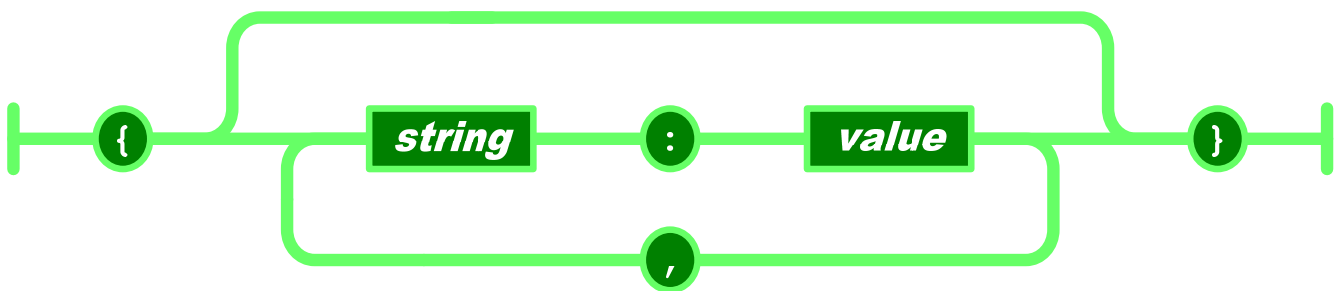
```
- arr.splice(2, 0, "red", "green")
```

درج همراه با حذف

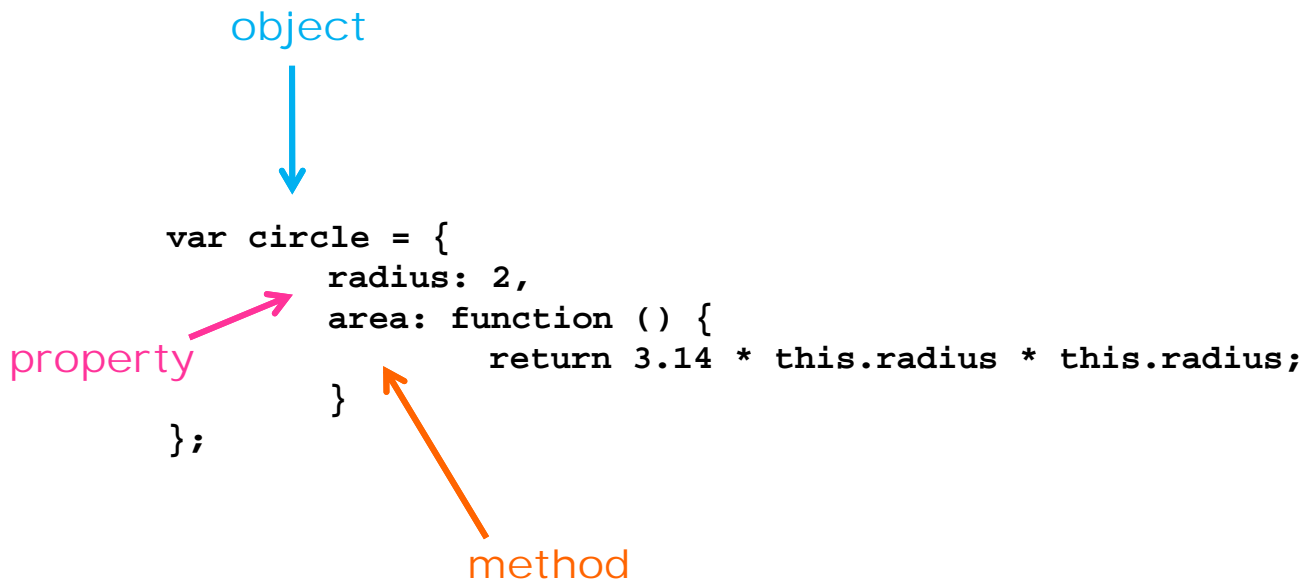
حذف عنصر موقعیت ۲ و درج از همان موقعیت

```
- arr.splice(2, 1, "red", "green")
```

شی



اشیا: خصوصیت‌ها، متدها و `this`



اشیا: خاصیت‌ها، متدها و `this`

- `alert(circle.radius); // 2`
- `alert(circle.area()); // 12.56`

- `if/else if/else`
- `switch`
- `while`
- `do/while`
- `for`
- `for/in`
- `break`
- `continue`
- `return`
- `throw`
- `try/catch`
- `with`

for

- توسط این حلقه، می‌توان به عناصر یک آرایه دسترسی پیدا کرد.

```
for (var i = 0; i < array.length; i += 1) {  
  // within the loop  
  // i is the index of the current member  
  // array[i] is the current element  
}
```

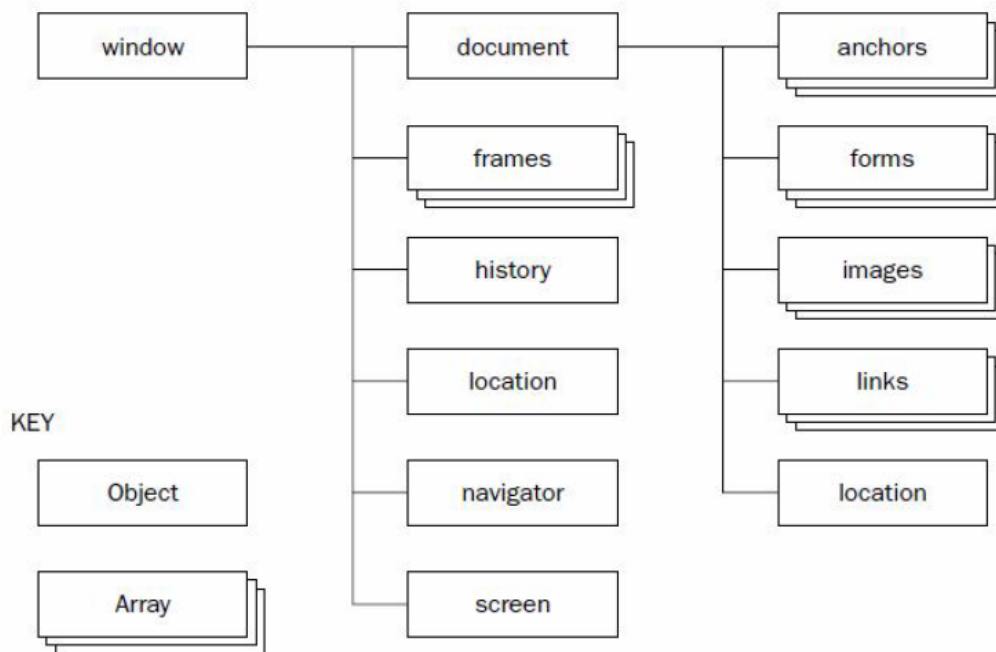

- توسط این حلقه، می‌توان به خاصیت‌های یک شی دسترسی پیدا کرد.

```
for (var property in object) {  
    // within the loop  
    // property is the key of current member  
    // object[property] is the current value  
}
```

```
var person={fname:"John",lname:"Doe",age:25};  
  
for (x in person) {  
    txt=txt + person[x];  
}
```

BOM (Browser Object Model)

- BOM اشیایی که با پنجره ی مرورگر ارتباط و تعامل مستقیم دارند را فراهم می کند



شی window

- پنجره ی مرورگر را شامل می شود اما نه لزوما محتوایی که در آن نمایش داده می شود.
- از این شی برای جابجایی، تغییر اندازه و دیگر اثرات بر روی پنجره ها استفاده می کنیم
- `moveBy(dx,dy)`: جابجایی مرورگر نسبت به موقعیت فعلی
- `moveTo(x,y)`: جابجایی نقطه بالا-چپ مرورگر به موقعیت مشخص شده
- `resizeBy(w,h)`: تغییر اندازه پنجره مرورگر نسبت به اندازه فعلی
- `resizeTo(w,h)`: تغییر اندازه پنجره مرورگر با مقادیر مشخص شده

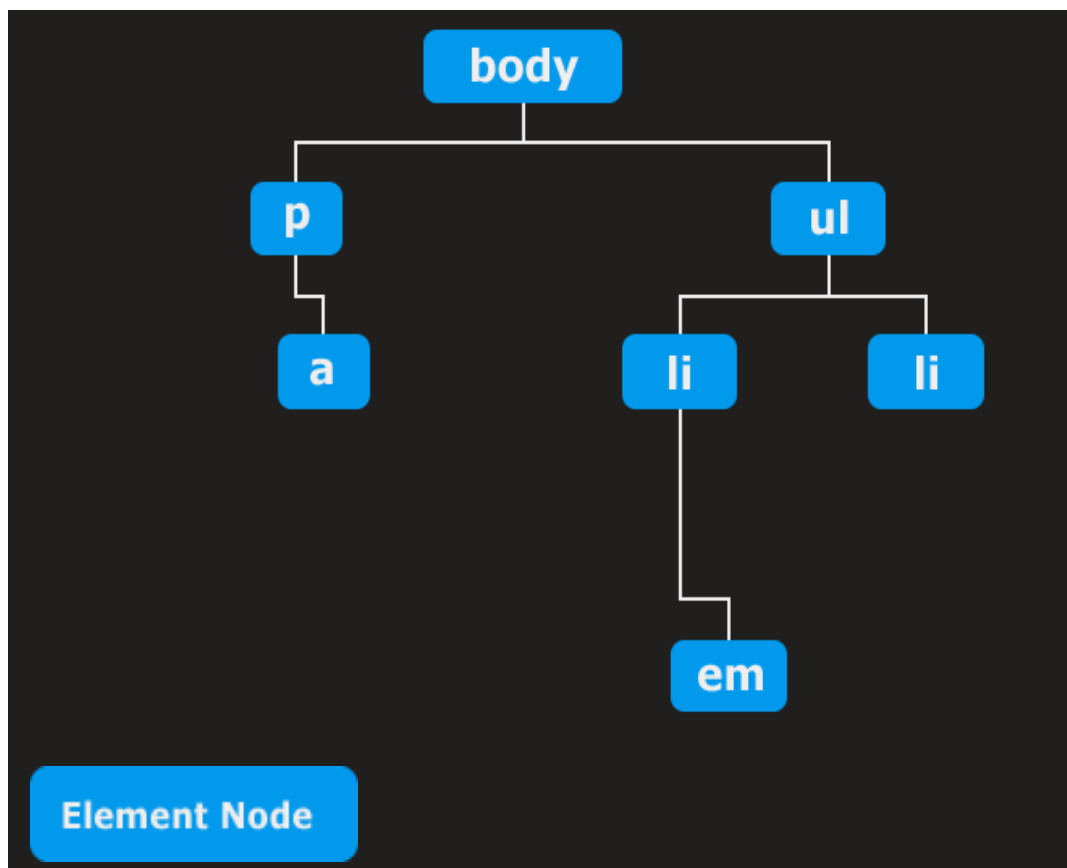
■ شی window دارای ۵ شی مهم و اصلی است که با کمک این اشیا می توان تمام فرآیندها را در صفحه کنترل کرد:

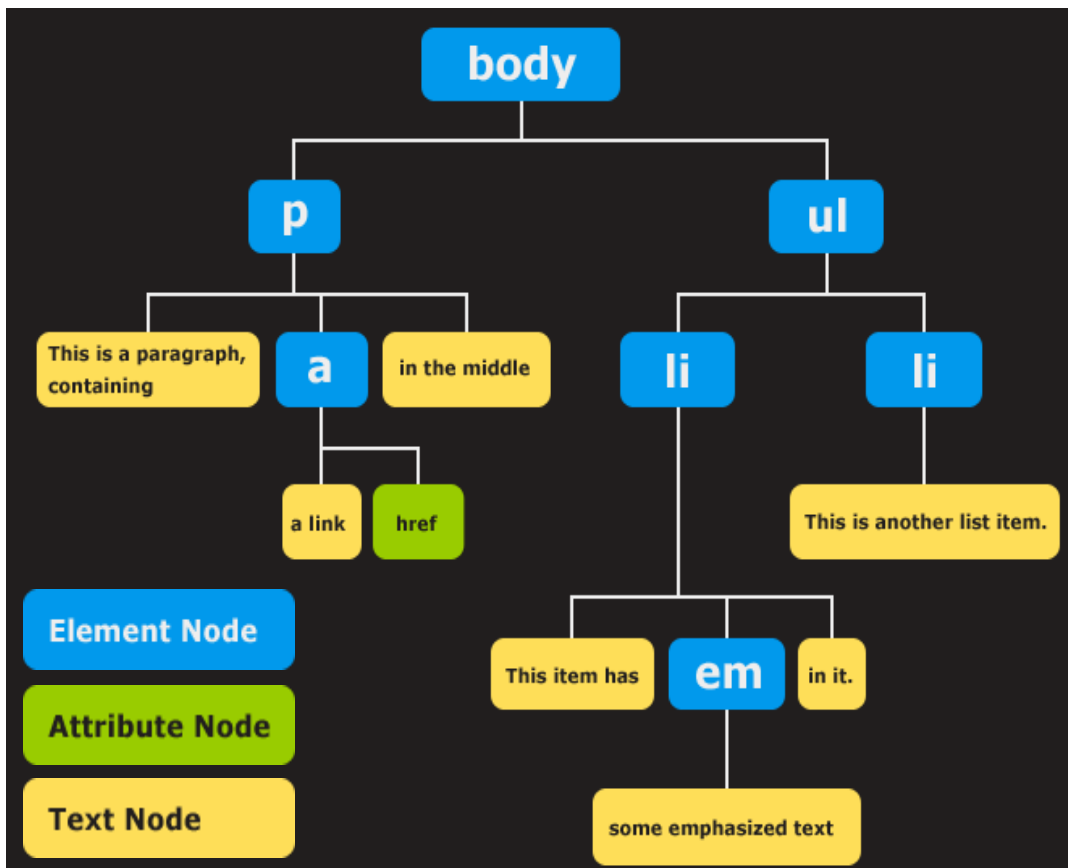
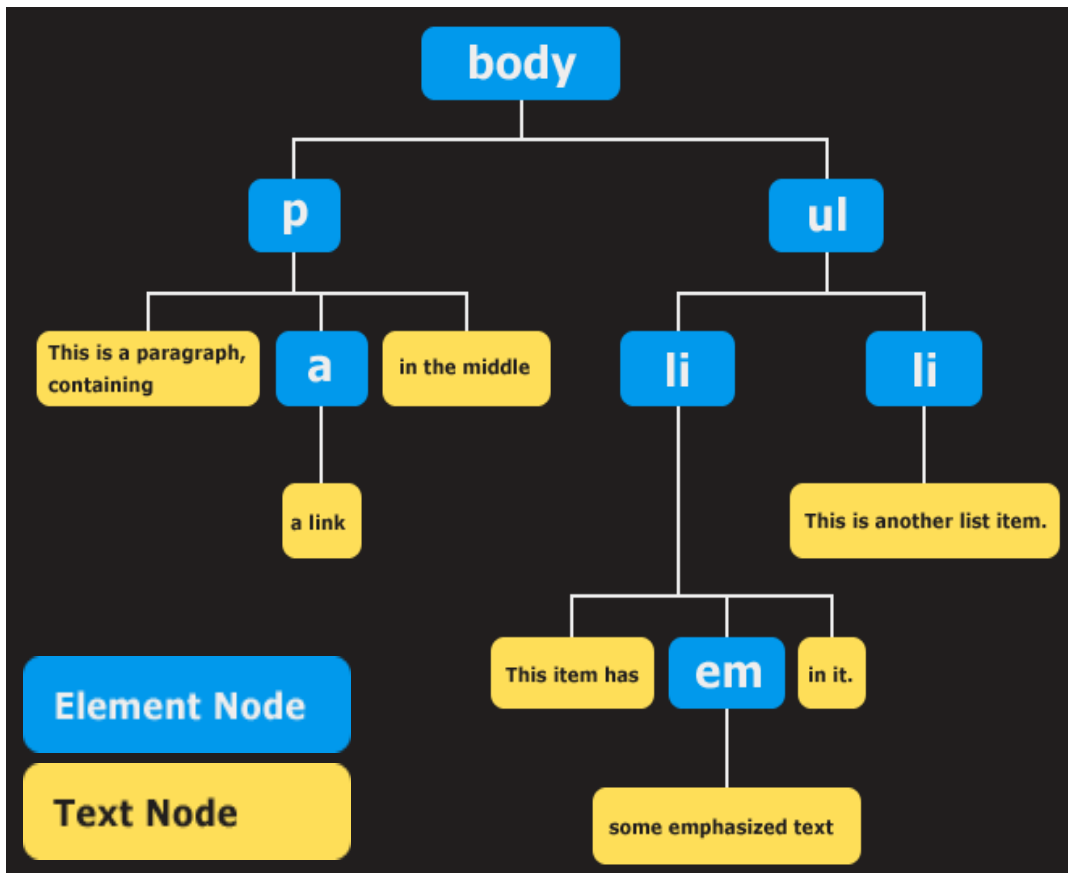
- Document
- Location
- History
- Navigator
- Screen

DOM

- هر سند اچ تی ام ال، در جاوا اسکریپت، یک مدل درختی دارد.
- این مدل درختی را DOM می نامند.
- توسط DOM، می توان المان های موجود در اچ تی ام ال را ویرایش کرد.

```
<body>
  <p>
    This is a paragraph, containing
    <a href="#">
      a link
    </a>
    in the middle.
  </p>
  <ul>
    <li>
      This item has
      <em>
        some emphasized text
      </em>
      in it.
    </li>
    <li>
      This is another list item.
    </li>
  </ul>
</body>
```





■ تنها شی مشترک بین مدل‌های DOM و BOM

– این شی به ارائه اشیا و خصوصیات و متدهایی برای کار کردن با تمام عناصر سند ها، مانند فرم ها و پیوند ها و کلید ها و ... می پردازد

■ این شی دارای یک مجموعه از خصوصیات مجموعه‌ای برای دسترسی به انواع عناصر داخل

صفحه است

| توضیحات | مجموعه |
|----------------------------------|---------|
| دسترسی به لینک های صفحه | anchors |
| دسترسی به تمامی عناصر embed صفحه | embeds |
| دسترسی به تمامی فرم های صفحه | forms |
| دسترسی به تمامی عناصر عکس صفحه | images |
| دسترسی به تمامی لینک های صفحه | links |

■ هر مجموعه می‌تواند بوسیله‌ی عدد یا نام، **index** گذاری شوند. به این معنی

که شما می‌توانید به صورت زیر به اولین عنصر عکس صفحه دسترسی داشته باشید:

– `document.images[0];`

– `document.images['image-name'];`

■ می‌توانیم به آدرس تصاویر نیز دسترسی داشته باشیم:

– `document.images[0].src`

■ برای چاپ یک متن

- `document.write('Hello!');`
- `document.writeln('Hello!');`

دسترسی به المان‌ها

- `document.getElementById`
- `node.getElementsByTagName`
- `node.getElementsByClassName`

document.getElementById

■ از این متد، برای دستیابی به المانی با یک `id` خاص، استفاده می‌کنیم.

```
– var e = document.getElementById('element_id');
```

document.getElementsByTagName

■ این متد، برای دستیابی به تمامی المان‌هایی از یک نوع خاص، براساس نام تگ، مورد استفاده قرار می‌گیرد.

■ المان‌های به‌دست آمده، به صورت یک آرایه هستند.

```
■ var es = document.getElementsByTagName('element_tagname');
```


با المان‌ها چه کار کنیم؟

- پرسش مهم این است: با این المان‌ها که در جاوااسکریپت به دست آوردیم، چه کار کنیم؟

- جواب بسیار ساده است: هر کاری که دلتان می‌خواهد می‌توانید با آن‌ها انجام دهید.

- حذف آن‌ها

- کپی کردن آن‌ها

- دسترسی به خاصیت‌ها

- تنظیم استایل‌ها

- دسترسی به محتوای آن‌ها

innerHTML

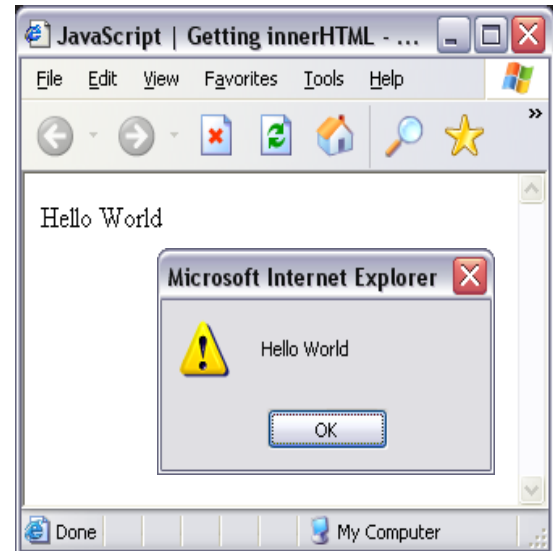
- از این خاصیت، برای دسترسی به محتوای یک المان استفاده می‌کنیم.

- می‌توان محتوای المان را نیز، تغییر داد.

- این خاصیت جزء استانداردهای DOM نیست

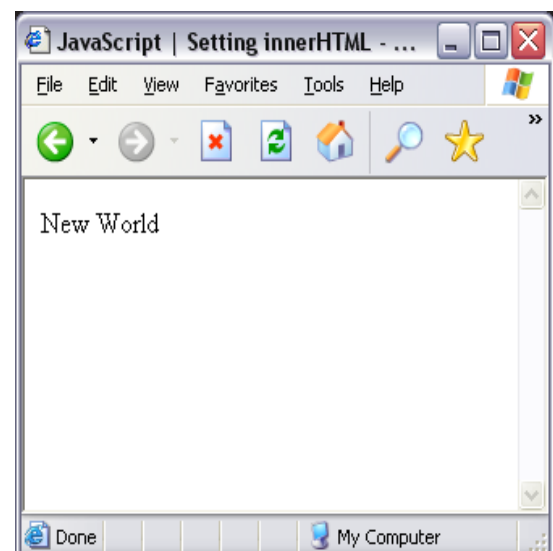
```
<body>
  <p id="para">Hello World</p>

  <script type="text/javascript">
    var e = document.getElementById('para');
    alert(e.innerHTML);
  </script>
</body>
```



```
<body>
  <p id="para">Hello World</p>

  <script>
    var e = document.getElementById('para');
    e.innerHTML = 'New World';
  </script>
</body>
```



```
<div>
  <a href="http://google.com/">Google</a>
</div>
<div>
  <a href="http://yahoo.com/">Yahoo!</a>
</div>
```

```
var links = document.getElementsByTagName('a');
for (var i = 0; i < links.length; i++) {
  alert(links[i].innerHTML);
}
```

```
var links = document.getElementsByTagName('a');
```

links[0]

the **a** element for "Google"

links[1]

the **a** element for "Yahoo!"

دسترسی به خاصیت‌های یک المان

element.property

className

htmlFor

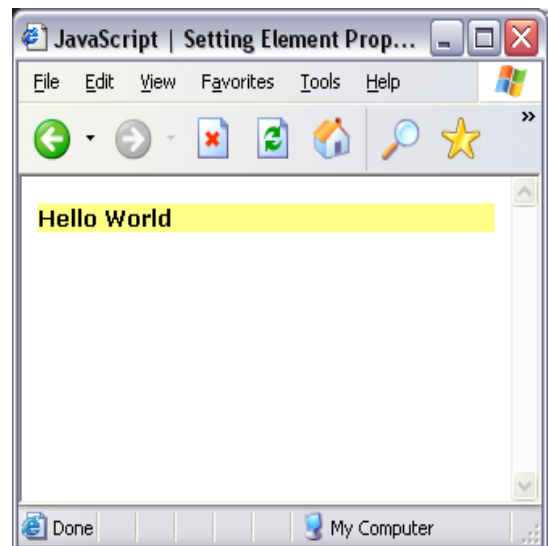
```
<p>
  <a href="http://yahoo.com/" id="ya">
    fantazio
  </a>
</p>

<script>
  var link = document.getElementById('ya');
  alert(link.href);
</script>
```

```
<style type="text/css">
  #special {
    background: #ff8;
    font: bold 10pt Verdana;
  }
</style>
```

```
<p id="para">Hello World</p>
```

```
<script>
  var p = document.getElementById('para');
  p.id = 'special';
</script>
```



■ آشنایی با DOM

– <http://www.w3schools.com/html/dom/>

■ مرجع کامل جاوااسکریپت و DOM

– <http://www.w3schools.com/jsref>