

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

JavaScript HTML DOM Navigation

مدرس : مهندس افشین رفوآ

[دوره آموزش JQuery](#)

[دوره آموزش JavaScript](#)

JavaScript HTML DOM Navigation

پیمایش مدل DOM

در مدل DOM می توانید با استفاده از رابطه ی گره ها (node relationship)، درخت گره (node tree) را پیمایش کنید.

گره های DOM

مبتهی بر استاندارد مدل DOM هر عنصر، تگ، خاصیت و یا متن در سند HTML یک گره محسوب می شود. با تکیه بر این اصل تعاریف زیر ممکن می باشد:

کل سند یا صفحه ی HTML، خود یک گره سند (document node) تلقی می گردد.

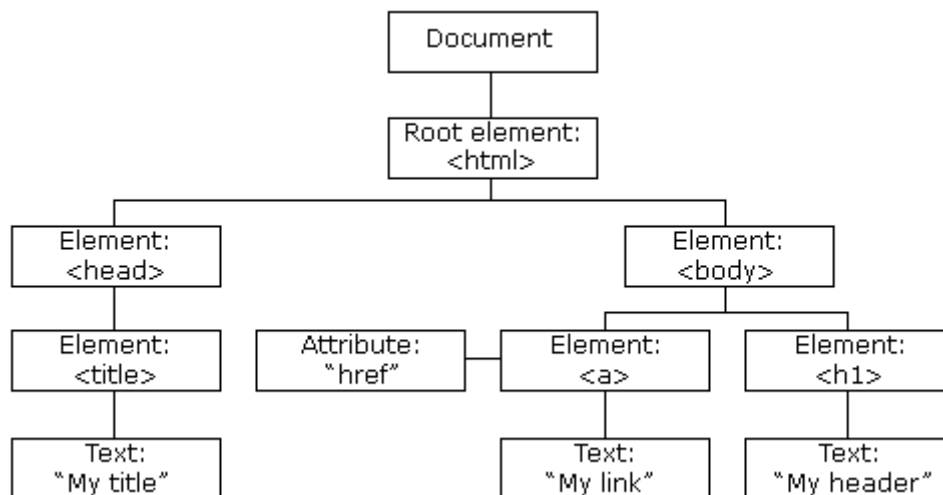
هر tag، خود یک گره عنصر (element node) محسوب می شود.

متن های موجود بین tag ها (در مواردی همچون پاراگراف و لینک) گره های متنی (text Node) محسوب می شوند.

آدرس آموزشگاه : تهران - خیابان شریعی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7

88146323 - 88446780 - 88146330

هر خاصیت در تگ ها و کنترل های HTML ، یک گره خصیصه (**attribute Node**) تلقی می شوند .
comment های موجود در سند های HTML ، گره های توضیحی (**comment Node**) محسوب می
 شوند.



در مدل **DOM** ، می توان با استفاده از جاوا اسکریپت به تمامی گره ها دسترسی پیدا کرد.

در این مدل، امکان ایجاد گره های جدید وجود دارد و همچنین تمامی گره ها را می توان در صورت لزوم اصلاح و یا حذف کرد.

ارتباط گره ها

گره های موجود در درخت گره ها (**node tree**)، یک رابطه ی سلسله مراتبی با یکدیگر دارند.

برای توصیف صحیح روابط گوناگون بین گره ها از واژه های **sibling**، **child** و **parent** استفاده می کنیم.

در **node tree** ، بالادست ترین گره، گره ی ریشه یا **root node** خوانده می شود.

به ازای هر گره یک **parent** وجود دارد، البته به استثنای گره ی ریشه که گره ی والد ندارد.

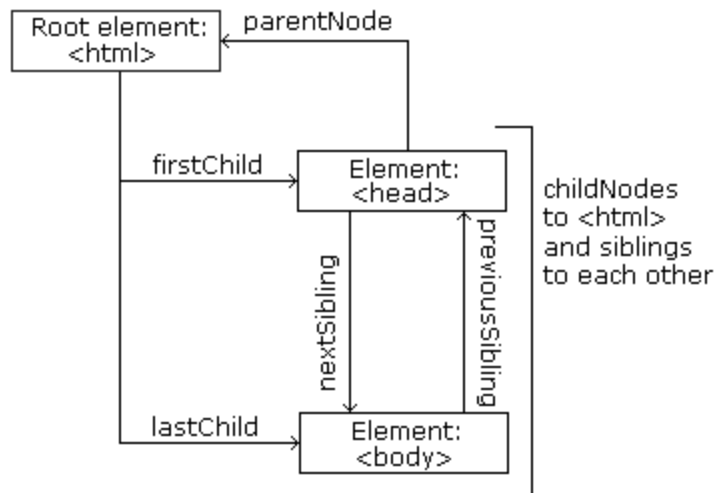
هر گره می تواند تعدادی فرزند (**child**) داشته باشد.

گره های که رابطه ی بین آن ها از نوع **sibling** است، گره هایی هستند که دارای والدی یکسان می باشند.

```

<html>
<head>
  <title>DOM Tutorial</title>
</head>
<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
</body>
</html>

```



از کد HTML فوق می توان نتیجه گرفت:

تگ **<html>** همان گره ی ریشه است

تگ **<html>** هیچ والدی ندارد

<html> والد هر دو تگ **<head>** و **<body>** محسوب می شود

<head> اولین فرزند المان **<html>** تلقی می گردد

<body> آخرین فرزند **<html>** می باشد

:9

<head> یک فرزند دارد و آن تگ **<title>** است

المان `<title>` خود یک فرزند (گره ی متنی) دارد: "DOM Tutorial"

`<body>` دارای دو گره ی فرزند است: `<h1>` و `<p>`

`<h1>` یک فرزند دارد که آن "DOM Lesson one" است

`<p>` یک فرزند دارد: "Hello world!"

`<h1>` و `<p>` با هم رابطه ی بدوی (خواهر و برادری) دارند

پیمایش بین گره ها

می توانید با استفاده از خاصیت های **node** زبان جاوا اسکریپت، بین گره ها پیمایش کنید:

parentNode

childNodes[*nodenumber*]

firstChild

lastChild

nextSibling

previousSibling

توجه:

یک اشتباه معمول در پردازش DOM این است که گره ی ریشه متن دربر می گیرد.

در این مثال، `<title>DOM Tutorial</title>`، گره ی عنصر `<title>` هیچ نوشته ای دربر ندارد، بلکه

دربردارنده ی یک گره ی ریشه است که مقدار "DOM Tutorial" را درخود نگه داشته.

می توان به وسیله ی خاصیت **innerHTML** گره ی مورد نظر یا **nodeValue**، به مقدار گره ی متنی (**text**)

node) دسترسی پیدا کرد.

گره های فرزند و مقادیر گره ها

آدرس آموزشگاه : تهران - خیابان شریعی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7

88146323 - 88446780 - 88146330

علاوه بر خاصیت **innerHTML**، می توان از خواص **childNodes** و **nodeValue** به منظور بازیابی محتویات یک المان کمک گرفت.

مثال زیر مقدار **node** المان **<h1>** را بدست آورده، سپس آن را در المان **<p>** جای گذاری می کند.

مثال:

```
<!DOCTYPE html>
<html>
<body>
  <h1 id="intro">My First Page</h1>
  <p id="demo">Hello World!</p>
  <script>
    var myText = document.getElementById("intro").childNodes[0].nodeValue;
    document.getElementById("demo").innerHTML = myText;
  </script>
</body>
</html>
```

در مثال فوق، **getElementById** یک متد است در حالی که **nodeValue** و **childNodes** هر دو خاصیت (**property**) محسوب می شوند.

در این سری آموزشی غالباً از خاصیت **innerHTML** بهره می گیرم. با این وجود توصیه می کنیم دو متد نام برده را برای درک بهتر ساختار درختی و پیمایش **DOM**، بیاموزید.

استفاده از خاصیت **firstChild**، نتیجه ای یکسان با **childNodes[0]** می دهد:

```
<!DOCTYPE html>
<html>
<body>
  <h1 id="intro">My First Page</h1>
  <p id="demo">Hello World!</p>
  <script>
    myText = document.getElementById("intro").firstChild.nodeValue;
    document.getElementById("demo").innerHTML = myText;
  </script>
</body>
</html>
```

گره های ریشه ی DOM

در مجموع دو خاصیت وجود دارند که امکان دسترسی به کل یک سند را فراهم می‌سازند. این دو خاصیت عبارتند از:

document.body – دسترسی به بدنه ی سند را امکان پذیر می‌سازد.

document.documentElement – دسترسی به کل یک سند را ممکن می‌سازد.

مثال:

```
<!DOCTYPE html>
<html>
<body>
  <p>Hello World!</p>
  <div>
    <p>The DOM is very useful!</p>
    <p>This example demonstrates the <b>document.body</b> property.</p>
  </div>
  <script>
    alert(document.body.innerHTML);
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <p>Hello World!</p>
  <div>
    <p>The DOM is very useful!</p>
    <p>This example demonstrates the <b>document.documentElement</b> property.</p>
  </div>
  <script>
    alert(document.documentElement.innerHTML);
  </script>
</body>
</html>
```

خاصیت nodeName

این خاصیت، اسم یک گره (node) را مشخص می‌کند.

nodeName فقط خواندنی و ثابت (read-only) می‌باشد.

nodeName یک گره ی عنصر (element node) با اسم tag یکسان می‌باشد.

- nodeName** گره ی خصیصه (**attribute node**)، اسم خصیصه است.
- nodeName** گره ی متنی (**text node**)، همیشه **#text** است.
- nodeName** گره ی صفحه (**document node**) همیشه **#document** است.

خاصیت **nodeValue**

خاصیت **nodeValue** مقدار یک گره را مشخص می کند.

- مقدار خاصیت **nodeValue** برای گره ی عنصر (**element node**)، **undefined** می باشد.
- nodeValue** برای گره های متنی (**text node**) خود متن می باشد.
- nodeValue** برای گره های خصیصه (**node attribute**) همان مقدار خصیصه (**attribute value**) می باشد.

خاصیت **nodeType**

خاصیت **nodeType**، نوع گره ی مورد نظر را برمی گرداند. خاصیت **nodeType**، فقط خواندنی و ثابت (**read-only**) است.

مهم ترین نوع های گره به ترتیب در جدول زیر ارائه شده:

Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9