

getElement

برای کار با هر المان ابتدا باید آن را انتخاب کنید تا بتوان ویژگی های آن را کنترل کرد .
برای این کار چند راه ابتدایی وجود دارد : (فقط همین 5 روش وجود داره و مورد استفاده)

Html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<h1 align="center" id="title">iman rad</h1>
<ul>
  <li class="list-item">About</li>
  <li class="list-item">Contact</li>
  <li class="list-item">servies</li>
  <li class="list-item">Home</li>
</ul>
<script src="script.js"></script>
</body>
</html>
```

script

1 :

```
let h1Elem = document.getElementById('title')
```

2:

```
let ulElem = document.getElementsByTagName('ul')
```

این روش همیشه برای شما یک آرایه بر می گرداند .

3:

```
let ulElem = document.getElementsByClassName('list-item')
```

این روش همیشه برای شما یک آرایه بر می گرداند .

4:

```
let firstListItem = document.querySelector('.list-item')
```

در این روش اولین المنت که با آن تگ ساخته شده باشد را انتخاب می کند .

در شیوه صدا زدن المنت خود دقت کنید ، یعنی مانند css اگر کلاس بود با (.) و اگر id بود با (#) .

5

```
let firstListItem = document.querySelectorAll('.list-item')
```

مانند قبل است اما یک آرایه بر می گرداند که متشکل از المنت با شناسه ذکر شده است

دسترسی به اتریبوت ها

برای خواندن و گرفتن صفت از `getAttribute` استفاده می کنیم .

```
console.log(h1Elem.getAttribute('align'))
```

```
console.log(h1Elem.getAttribute('id'))
```

```
console.log(h1Elem.getAttribute('class'))
```

برای تغییر در اتریبوت ها از چه روشی استفاده می شود ؟

```
console.log(h1Elem.setAttribute('class', 'iran'))
```

استایل دهی به المنت ها

شیوه استایل دهی به المنت ها با تنظیم استریبوت های ان متفاوت است .

برای تغییر استایل المنت ها از روش زیر استفاده می کنیم :

```
h1Elem.style.  
h1Elem.style.color = 'red'  
h1Elem.style.fontSize = '5rem';
```

```
let h1Elem : HTMLInputElement = document.getElementById( elementId: 'title');
```

```
h1Elem.style.
```

alignmentBaseline	string
alignSelf	string
animation	string
animationDelay	string
animationDirection	string
animationDuration	string
animationFillMode	string
animationIterationCount	string
animationName	string
animationPlayState	string
animationTimingFunction	string
appearance	string

Ctrl+Down and Ctrl+Up will move caret down and up in the editor [Next Tip](#)

نحوه دسترسی به محتوای html و تغییر آن ها

به بخش جذابی رسیدیم که بشدت مورد استفاده منه !

1 innerHtml

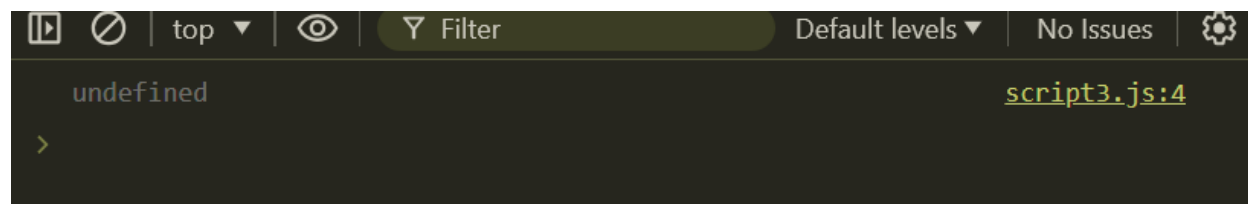
```
let h1Elem = document.getElementById('title');  
console.log(h1Elem.innerHTML)
```

با این روش محتوای داخل تگ مد نظر را می توان بررسی کنید .

اما اگر چنتا بودن چی ؟

```
let liItems = document.querySelectorAll('.title');  
console.log(liItems.innerHTML)
```

خروجی :



در صورتی که یک لیست به ما برگردد و ما محتوای آن را بخواهیم این پاسخ را دریافت می کنیم ، راحل صدا زدن دقیق تر به وسایه index ها است .

```
console.log(liItems[2].innerHTML)
```

2 innerText

```
console.log(liItems[2].innerText)
```

3 textContent

```
console.log(liItems[2].textContent)
```

خب خب خب ، حالا آگه بخوام تغییرش بدم چی؟

```
liItems[2].innerHTML='iman'
```

اماااا این وروش روی برخی تگ ها جواب نمی ده مثلا input ها برای اونا از روش دیگری استفاده می شود :

```
<body>
<h1 align="center" id="title">iman rad</h1>
<ul>
  <li class="list-item">About</li>
  <li class="list-item">Contact</li>
  <li class="list-item">servies</li>
  <li class="list-item">Home</li>
</ul>
<hr>
<input type="text" id="inp" value="cantent">
<script src="script3.js"></script>
</body>
```

```
let inpuetElem = document.getElementById('inp');
inpuetElem.value='hi iman';
```

در واقع برای دسترسی به تگ هایی که قالبان در فرم ها استفاده می شوند از value استفاده می کنیم .

نحوه ایجاد یک المان با dom

خب خیلی سادس از createElement استفاده می کنیم ، یعنی چی ؟ خب ببین ...

```
let newH1Elem = document.createElement('h1');
```

و میتونیم بهش اتریبوت و استایل بدیم :

```
let newH1Elem = document.createElement('h1');
let username = prompt('Enter your name :')

newH1Elem.innerHTML=username;
newH1Elem.setAttribute('id','new-h1');
newH1Elem.setAttribute('class','title');
newH1Elem.style.color= 'green';
```

اما باید ان را به جایی افزود دو سوال چه المنتی ؟ به کدام المنت ؟

چه المنتی رو یادگرفتید بریم سراغ کجا افزودنش ... یه روش ساده از append

```
ulElem.append(newH1Elem);
```

یا می تونید از appendChild استفاده کنید .

کلیک 😊

برای این حالت نیاز به یک تابع است ، که یک المنت را به آن متصل کنیم . باید برای المنت مد نظر یک رویداد تعریف شود که اگر آن رویداد خاص اتفاق افتاد تابع خود را اجرا کند .

تعریف رویداد :

```
document.getElementById('btn').addEventListener('click',click);
```

تعریف تابعی که در صورت اتفاق آن رویداد باید انجام شود :

```
function click() {  
    btn.style.background = 'red';  
}
```

در نتیجه و در نگاه کلی :

```
let btn = document.getElementById('btn');  
  
function click() {  
    btn.style.background = 'red';  
}  
  
document.getElementById('btn').addEventListener('click',click);
```

با چیز های که تا اینجا یادگرفتیم دوتا پروژه زدیم که آگه بری توی فایل ها چک کنی می تونه کمکت کنه .

استفاده از رویداد keyboard

چند حالت وجود دارد

حالت اول که به ازای هر تایپ است :

onkeypress

```
<input id="password" onkeypress="keypress('password')">
```

```
function keypress(state) {  
    alert('hi')  
}
```

Event focus – blur

زمانی که بر روی یک اینپوت سلکت می کنیم focus اجرا می شود و همچنین زمانی که از آن خارج می شویم blur اجرا می شود .

```
<input type="text" id="input" onfocus="focusInput()">  
<input type="text" id="input" onblur="blurInput()">
```

در واقع برای مدیریت ورود و خروج کاربر از اینپوت ها استفاده می شود .

dblClick

کاملاً یک چیز چرته ! همون onclick خودمونه فقط با این تفاوت که می گه باید دوبار کلیک کنی ☺
باشه اوکی ! بریم بعدی ...

addEventListener

تا الان دیدیم که می تونیم در ساختار html تابع ها رو به المنت ها متصل کنیم ، اما راه بهر و اصولی تری است .
بهتر است که این کار را با یک اسکریپت و در فایل js انجام دهیم مانند مثال زیر :

```
document.getElementById('input').addEventListener('typeEvent', your_fun);
```

خبیب اینجا اول مشخص کردیم کدوم المنت رو می خوایم و بعدشش امدیم گفتیم اگه این ایونت برات رخ داد این تابع را اجرا کن .
نکته در روش قبل دیدیم که باید موقع صدا زدن تابع در html جلوش پرانتز بزاریم ، ولییی اینجا بهتر نزارید ! چرا؟
چون اگر بزارید اون تابع یک بار اجرا می شه مثل do while

classList

حالا می خوایم به یک المنت با js کلاس بدیم ! خیلی سادس

```
ulElem.classList.add('ul')
```

می تونید بگید اگر فلان اتفاق افتاد این المنت ظاهرش تغییر کند .

```
function click () {  
let ulElem = document.getElementById('ul')  
ulElem.classList.add('ul')  
}  
document.getElementById('input').addEventListener('click', click);
```

البته این برای افزودن بود برای حذف از remove استفاده می شود .

```
ulElem.classList.remove('ul')
```

toggle

تاگل اینجا یعنی چک کنه اگر اون کلاسو داشت حذف و اگه نداشت اضافه کنه .

```
ulElem.classList.toggle('ul')
```

onchange

وقتی یک انپوت مقدار داخلش تغییر کنه این رویداد اجرا می شود .

3 تفاوت اصلی append و appendChild

فرض کنید یک li به وسیله createElement ساختیم خب چطور به ul که در html است اضافه کنیم ؟

روش اول استفاده از append هست که در ظاهر با appendChild شاید یکسان است اما

اگر به append یک متن خالی بدید اون رو به شکل li اضافه می کنه ولی appendChild به هیچ عنوان چیزی جز المنت دریافت نمی کنه ! این اولی

این دلیل به این خاطر بود که appendChild با نود ها کار می کنه و برای کار با نود و افزودن به ان باید المنت داشته باشد .

دوم ، append چند ورودی می گیر اما appendChild خیر ، فقط یک ورودی می گیرد .

سوم اگر المنت اضافه شده در append را در یک متغیر ذخیره کنید و ان را log بگیرید تعریف نشده دریافت خواهید کرد ، اما اگر این کار را در appendChild انجام دهید المنت اضافه شده را دریافت خواهید کرد .