مقدمة عن لغة الجافا سكريب

صفحة الويب

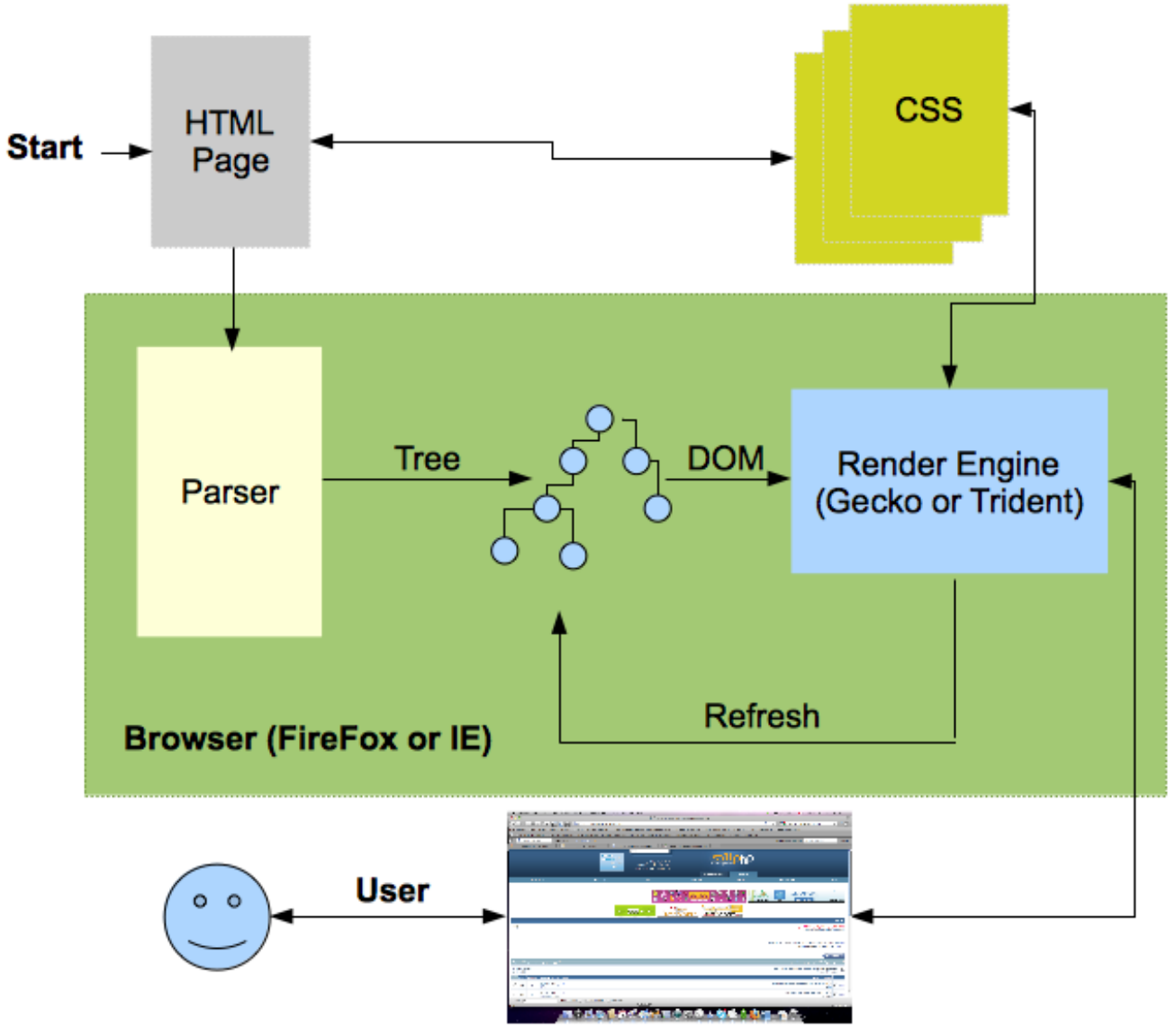
صفحة الويب هي عبارة عن صفحة مكونة من ثلاثة طبقات (Layers) كالتالي:

طبقة المحتويات (Content Layer)

طبقة العرض (Presentation Layer)

طبقة التفاعل(Behavior Layer)

ما هو الـ DOM ؟


Start
HTML Page
CSS
Parser
Tree
DOM
Render Engine (Gecko or Trident)
Refresh
Browser (FireFox or IE)
User

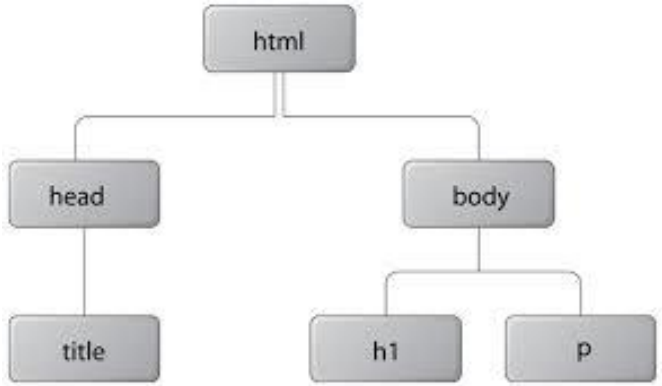
انشاء الشجرة

a) **Start:** هنا نقصد بداية طلب المستخدم للصفحة أي عند كتابته لعنوان معين فيه صفحةHTML .

b) **Parser** : هو عبارة عن برنامج موجود داخل المستعرض وظيفته الأساسية هي تحويل شفرة HTMLإلى شكل شجري أي أن مدخلات هذا البرنامج هي كود HTML ويقوم بإخراج شكل شجري يمثل هذا الكود و للتوضيح افرض أن لدي الكود التالي:

```
<html>
<head>
    <title>Welcome</title>
</head>
<body>
    <h1> Hi Mohammed </h1>
    <p> www.facebook.net </p>
</body>
</html>
```

- الآن وبعد أن يدخل هذا الكود إلى الـ Parser فإنه يتحول إلى Tree أو شكل شجري.



PARSER

- أي باختصار شديد نقول أن برنامج الـ Parser وظيفته هي تحويل الكود إلى شكل شجري لأن المستعرض لا يفهم كود HTML المكتوب وإنما يفهمه على الشكل الشجري وهذا الشكل الشجري يطلق عليه اسم DOM اختصاراً لـ Document Object Model وهي الشجرة التي ينتجها الـParser .

a) **Render Engine**هو برنامج يقوم بقراءة الشجرة أو الـ DOM التي ينتجها الـ Parser ويقوم برسم الرموز الموجودة في تلك الشجرة على شاشة المستعرض أي باختصار هو يقرأ الشجرة ويعرف معنى كل رمز ويقوم برسمه ولكن نريد التنبيه إلى شيء معين وهو لاحظ أن الرسم الموجود بالأعلى يضع سهماً بين Render Engine و CSSوالسبب هو أنه محرك الرسم يقوم أولاً بالتأكد من وجود ملفات CSSلكي يغير طريقة رسم الرموز وإن لم تكون موجودة فإنه يرسم بالطريقة الافتراضية, بعد ذلك وعند انتهاء المحرك من عملية الرسم فإن الشكل يظهر للمستخدم في المستعرض.

b) **Refresh** :هنا هي النقطة المهمة وهي عملية تحديث الصفحة والجميع يعرف انه عندما تريد تحديث الصفحة فإنك تضغط على زر تحديث او F5 ولكن لاحظ الرسم الموجود بالأعلى عند عملية التحديث فإن العملية التي تحدث هي الرجوع إلى الشجرة و قراءتها وليس الرجوع لبداية الكود ومن ثم رسم الكود من جديد.

ماهي لغة JAVASCRIPT

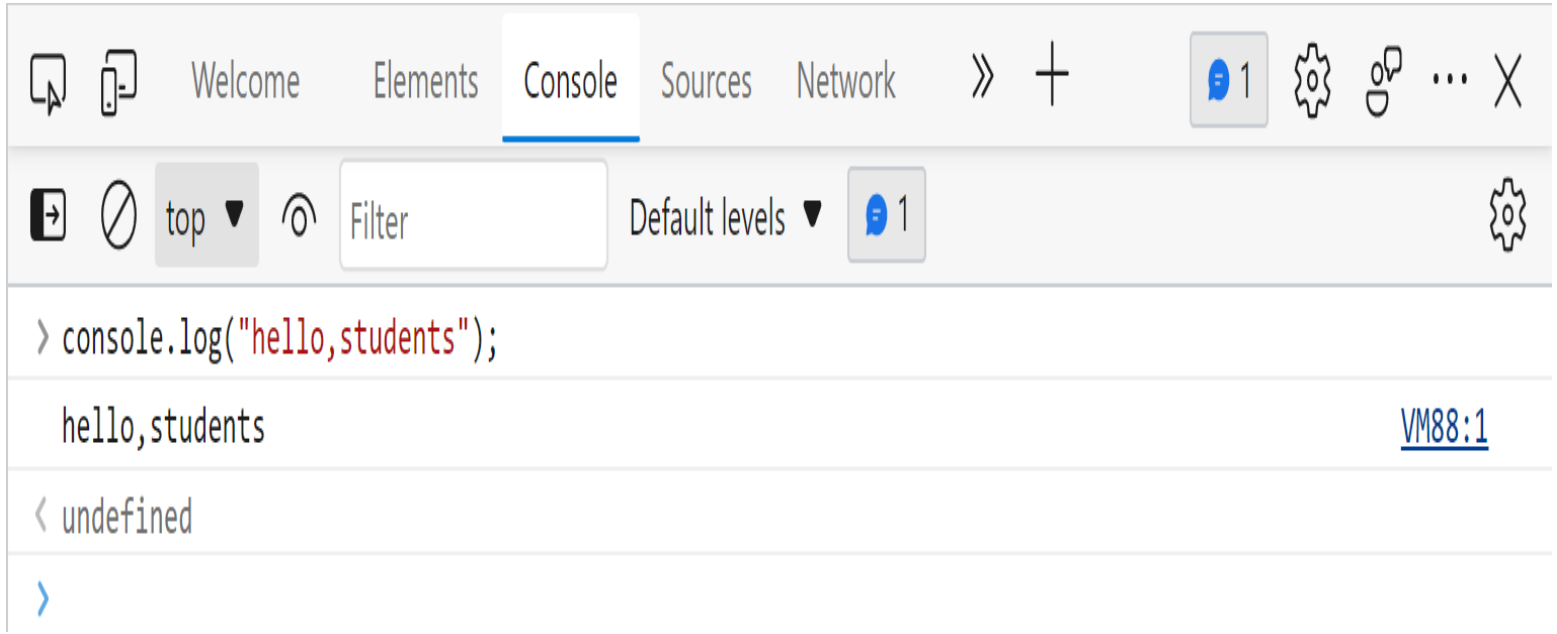
- لغة برمجة مترجمة خفيفة ومتعددة المنصات تُعرف أيضًا باسم لغة البرمجة النصية لصفحات الويب. تشتهر بتطوير صفحات الويب ، يمكن استخدام JavaScript للتطويرات من جانب العميل وكذلك للتطويرات من جانب الخادم، وتحتوي على مكتبة قياسية من الكائنات ، مثل Array و Date و Math ومجموعة أساسية من عناصر اللغة مثل عوامل التشغيل وهياكل التحكم والبيانات.

- لغة برمجة عالية المستوى تلعب دور حيوي وفعال في صفحات الويب من خلال القيام بوظائف قد تكون خارجية أو داخلية وهي مرنة لدرجة أنها تجعلك تتحكم بكل جزء من أجزاء صفحة الويب كأن تستخدمها في النماذج أو كنوافذ تخرج للمستخدم لتخبره بأمر معين أو للتنبيه وغيرها من الاستخدامات .

مميزاتها

1. الجافا سكربت تختلف عن أغلب لغات البرمجة الأخرى في كونها سهلة التحكم .
2. التكامل التام مع HTML / CSS.
3. مدعومة من قبل جميع المتصفحات الرئيسية وتمكينها افتراضيًا.
4. توفر التعامل مع الأحداث.
5. تعمل من خلال جميع أنظمة التشغيل .
6. تفاعل أقل مع الخادم server.
7. ردود فعل فورية للزوار وزيادة التفاعل.

تدعم جميع المتصفحات الحديثة كتابة الرسائل الى وحدة التحكم باستخدام دالة log لغرض تصحيح الأخطاء. يمكن فتح ال console الخاص بالجافا سكريبت من المتصفح وكتابة أكواد جافا سكريبت والامثلة التالية توضح ذلك:



ارسال متغيرات وطباعة القيم

```
> var id=24;
< undefined
> id
< 24
> let id2=23,name="Mohammed",avg=88.65;
< undefined
> console.log("std_num:",id,"std_name:",name,"std_avg:",avg);
  std_num: 24 std_name: Mohammed std_avg: 88.65
```

موضع كتابة شفرة JAVASCRIPT

- في رأس صفحة الـ html في وسم رأس الصفحة <head>

```
<head>
    <title>first page</title>
    <script type="text/javascript">
        //يتم هنا كتابة الشفرة
    </script>
</head>
```

1. استخدام ملف خارجي بامتداد js .
   - يتم استخدام الخاصية src في الوسم <script> للإشارة إلى ملف الـ javascript .

```
<html>
<head>
    <title>first page</title>
    </head>
<body>
    //------------
</body>
<script type="text/javascript" src="javascriptcode.js" ></script>
</html>
```

❑ المتغيرات في لغة جافا سكريبت باستخدام VAR KEYWORD

- نحتا ج المتغيرات VARIABLESلكي نقوم تخزين البيانات او المعلومات.
- المتغيرات VARIABLESهي عبارة عن حاوية CONTAINERيتم تخزين بداخلها قيمة معينة VALUEعن طريق ال NAMESالأسماء ثم نقوم باسترجاعها في أي وقت ويمكننا تغيير القيمة عن طريق لغة JAVASCRIPTاثناء التشغيل.
- المتغيرات تحتوي على قيم. هذا مهم جداً ان تفهمه. المتغيرات ليست القيم نفسها ؛ إنها حاويات للقيم. يمكنك التفكير في أنها مثل الصناديق الكرتونية الصغيرة التي يمكنك تخزين الأشياء فيها او وحدات الادراج التي تخزن بداخلها اوراق مهمه وهذه الاوراق هي القيم وتكون في كل مره اوراق مختلفة.
- يتم تخزين في المتغيرات المعلومات او البيانات الخاصة مثلاً بالمستخدمين USERSاو الخاصة بعملية الشراء عبر الانترنت كسعر منتج معين واسم المنتج الخ.

امثلة علي اسماء متغيرات متطابقة مع الشروط العلوية.

```
var age = 22;
var username = "closetag";
var myName = "Ahmed Aly";
var color = "blue";
var user_Name = "Mohamed Adly";
var password = 258622;
var email = "info@closetag.com";
var emailAddress = "info2@closetag.com";
var email_Address = "info3@closetag.com";
```

1. طباعة جملة (أمر الطباعة) في المستند:

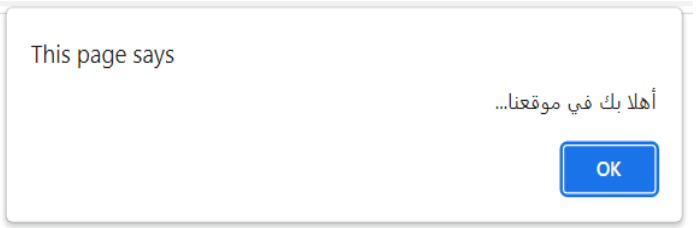
```
<script type="text/javascript">
        //أمر الطباعة بطرق مختلفة
  document.write("يتم كتابة النص المراد طباعته هنا..");
  document.write("<h1> html كتابة نص يقع عليه تأثير وسم من أوسمة الـ</h1>");
  document.write("<h1 style='color:red;'>css طباعة نص يقع عليه تأثير خصائص الـ</h1>");
        //طباعة قيمة متغير
        var name = "Ahmed";
        document.write(name);
        //طباعة قيمة متغير يسبقه جملة نصية
        document.write("welcome " + name);
 </script>
```

أوامر WINDOW

- خروج نافذة للمستخدم لعرض التنبيهات

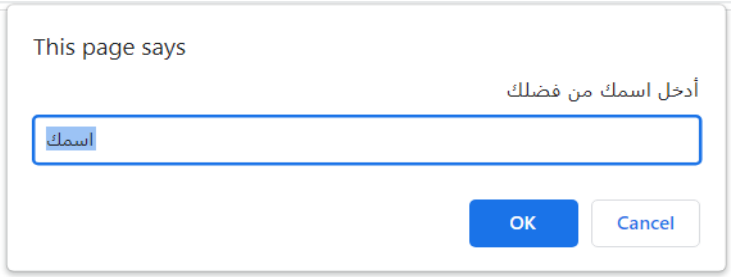
```
<script type="text/javascript">

window.alert("أهلا بك في موقعنا...");
 </script>
```



خروج نافذة للمستخدم نافذة قراءة من لوحة المفاتيح نافذة prompt

```
<script type="text/javascript">
var name = window.prompt("أدخل اسمك من فضلك", "اسمك");
 </script>
```

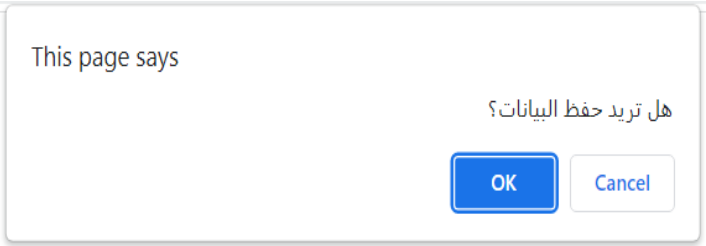


أوامر Window

```
<script type="text/javascript">

 var con = window.confirm("هل تريد حفظ البيانات؟");

</script>
```



- دالة confirm ترجع قيمة trueعند الضغط على زر حسنا (ok) وترجع قيمة false عند الضغط على زر الغاء(cancel).

❖ طريقة الوصول الى عناصر HTML:

✓الوصول للعناصر باستخدامID :

- أسهل طريقة للوصول إلى عنصر محدد في DOM هو المعرف الخاص بها ID(Identification)حيث يمكن الوصول إلى العنصر من خلال ID باستخدام التابع getElementById()مع العنصر document .

✓الوصول للعناصر باستخدامclass :

- تستخدم خاصية class للوصول إلى عنصر أو أكثر في DOM وذلك من خلال التابع getElementByClassName() .

✓الوصول للعناصر باستخدامTag :

- طريقة الوصول للعنصر من خلال اسم الوسم html tag name هو أقل الطرق تخصيصًا حيث يمكننا الوصول إلى العديد من العناصر باستخدام تابع getElementsByTagName()

✓محددات الاستعلامQuery Selector :

- من خلال استخدام التوابع querySelector() و querySelectorAll()يتم استدعاء العناصر باستخدام المحددات طريقة شبيه للتعامل مع العناصر في css.

بعض الخصائص المهمة المستخدمة في اللغة للتعامل مع محتوى الاوسمة ولإضافة خصائص CSS

1. innerHTML: لاستبدال محتوى html لاحد العناصر في الصفحة.
2. textContent: لاستبدال النصوص في عناصر html.
3. value: لقراءة واسناد قيم لأدوات الادخال في النماذج.
4. style: استخدام خصائص تقنية css.
5. className: استخدام الكلاسات المعرفة في ملف css .

❖ **الأحداث Events:**

- الأحداث هي الإجراءات أو الوقائع التي تحدث في النظام الذي تم برمجته عندما يتفاعل المستخدمين مع تطبيقات الويب، وهذا التفاعل يتم عن طريق:
  - احداث المستخدمين مع mouse .
  - احداث المستخدمين مع keyboard .
  - احداث المستخدمين مع screen .
  - احداث المستخدمين مع scrolling .
  - تحميل صفحة الويب.

احداث الجافاسكريبت JAVASCRIPT EVENT

الحدث	الوظيفة
onclick	النقر بزر mouse اليسار.
ondblclick	النقر مرتين بزر mouse اليسار.
onmouseout	عند الذهاب خارج العنصر بمؤشرmouse .
onmouseover	عندما يتم تحريك مؤشر mouse على عنصر أو احد أبناء هذا العنصر.
onscroll	عندما يقوم المستخدم بتحريك زر mouse الدائري الموجود بين الزر الأيمن الأيسر لأعلي أو لأسفل.
onkeyup	يقع الحدث عندما يقوم المستخدم بترك مفتاح لوحة المفاتيح.
onkeydown	يقع الحدث عندما يضغط المستخدم على مفتاح لوحة المفاتيح.
onload	عندما يتم تحميل الصفحة بالكامل.
onunload	عندما يتم إلغاء تحميل الصفحة.
onresize	عندما يتم تكبير أو تصغير نافذة صفحة الويب.
onblur	في أول لحظة عندما يقوم المستخدم بإلغاء التأشير على input محدد.
onchange	عندما يتم تغيير القيمة داخل input محدد.
onsubmit	عندما يتم إرسال البيانات من form إلي السيرفر.
onfocus	عندما يقوم المستخدم بنسخ المحتوى عند التأشير على الأداة.

- event attribute بداخل علامة open tag للعنصر عن طريق خصائص جافا سكريبت.
- event methods عن طريق دوال مجهزة في لغة جافا سكريبت.

JAVASCRIPT DOM GET ELEMENTS
البحث علي عنصر داخل صفحة HTMLفي لغة جافا سكريبت

- يمكن البحث علي العناصر داخل شجرة DOMبستة طرق مختلفة:
- ID name البحث عن عنصر واحد فقط له .ID
- class nameالبحث عن مجموعة عناصر تحمل classواحد.
- tag nameالبحث عن مجموعة عناصر تحمل اسم tagواحد.
- nameالبحث عن مجموعة عناصر تحمل اسم واحد.
- CSS selectorلتحديد عنصر واحد.
- CSS selector allلتحديد مجموعة عناصر.

البحث عن عنصر بواسطة IDفي جافا سكريبت

- يمكن البحث عن عنصر عن طريق id Nameوذلك عن طريق دالة مُجهزة ومُدمجة مع لغة جافا سكريبت وهي getElementById ()نقوم بتمرير بداخلها اسم IDويجب أن يكون الاسم فريداً في صفحة HTMLبمعني أن يكون الاسم غير مُكرر.
- البحث عن عنصر بواسطة idوحفظة بداخل متغير يحتوي المتغير علي جميع المعلومات الخاصة بالعنصر واصبح كائن به جميع الخصائص

```
<p>Click the button to display the date.</p>
<button onclick="displayDate()">The time is?</button>
<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>
<p id="demo"></p>
```

Result:

Click the button to display the date.

The time is?

Mon Feb 17 2020 06:57:56 GMT-0800 (توقيت المحيط الهادي الرسمي)

Onchange حدث

```
<head>
<script>
function myFunction() {
  var x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
Enter your name: <input type="text" id="fname“ onchange="myFunction()">
<p>When you leave the input field, a function is triggered which transforms the input text to upper case.</p>
```

أحداث Onmouseover وOnmouseout

```
<body>
<div onmouseover="mOver(this)" onmouseout="mOut(this)"
style="background-color:#D94A38;width:120px;height:20px;padding:40px;">
Mouse Over Me</div>
<script>
function mOver(obj) {
  obj.innerHTML = "Thank You"
}
function mOut(obj) {
  obj.innerHTML = "Mouse Over Me"
}
</script>
</body>
```

طريقة تغيير وتعديل محتوى عنصر HTML هي باستخدام خاصية innerhtml.

- **مثال:**

- **Var myElement = document.getElementById ("intro");**
- **إذا تم العثور على العنصر ، فستقوم الطريقة بإرجاع العنصر ككائن (في myElement).**
- **إذا لم يتم العثور على العنصر ، فسيحتوي myElement على قيمة خالية.**

- <body>
- <h2>Finding HTML Elements by Tag Name</h2>
- Hello World! This example demonstrates the getElementsByTagName method.
-
- <script> var x = document.getElementsByTagName("p");
- document.getElementById("demo").innerHTML ='The text in first paragraph (index 0) is: ' +

x[0].innerHTML; </script>

- **Result**

- **Finding HTML Elements by Tag Name**
- Hello World!
- This example demonstrates the **getElementsByTagName** method.
- The text in first paragraph (index 0) is: Hello World!

تغيير خصائص عنصر HTML

- `document.getElementById(id).style.property = new style`
- <!DOCTYPE html>
- <html>
- <body>
- Hello World!
- Hello World!
- <script>
- document.getElementById("p2").style.color = "blue";
- document.getElementById("p2").style.fontFamily = "Arial";
- document.getElementById("p2").style.fontSize = "larger";
- </script>
- The paragraph above was changed by a script.
- </body>
- </html>
- Result

//الاخفاء والعرض باستخدام خاصية Display

```
<script type="text/javascript">
      function show()
        {
           var id2 = document.getElementById('div1');
            id2.style.display ="block";

        }
      function hide()
        {
            var id2 = document.getElementById('div1');
            id2.style.display = "none";

        }
    </script>
<body >
<form id="form1" action="#"  method="get" >
    <input type="button" value="show" onclick="show();" />
     <input type="button" value="hide" onclick="hide();" />
      <div id="div1" style="background-color:aqua; width:70%; height:50%;
display:none;">
    welcome in my site....
    </div>
    </form>
```

DOM UPDATE ELEMENTS
تعديل العناصر في صفحة HTML

- يمكن التعديل علي محتوي أي عنصر عن طريق ثلاث خصائص وهم:
- **innerText:** وهي خاصية تمكنك من قراءة وتعديل المحتوي النصي لأي عنصر داخل ملف HTMLعن طريق لغة جافا سكريبت.
- **innerHTML:** وهي خاصية تمكنك من قراءة وتعديل محتوي HTMLأي يشمل علامات HTMLلأي عنصر داخل ملف HTMLعن طريق لغة جافا سكريبت.
- **textContent** : وهي خاصية تمكنك من قراءة وتعديل المحتوي النصي لأي عنصر داخل ملف HTMLعن طريق لغة جافا سكريبت وتشمل جميع العناصر المتفرعه من هذا العنصر حتي وأن كانت غير مرئية علي المتصفح.

الفرق بين خاصية Innertext وخاصية Innerhtml

innerText	innerHTML
تقوم بقراءة وتعديل النصوص فقط لأي عنصر	تقوم بقراءة وتعديل محتوي [HTML] لأي عنصر
لا تستطيع أن تضيف عناصر [HTML] بواسطتها	تستطيع أن تضيف عناصر [HTML] بواسطتها
تتجاهل المسافات والسطور الأضافية	يمكنك أن تضيف مسافات وسطور أضافية
تقوم بأرجاع المحتوي وعدم قراءة وتلفذ أوامر HTML	تقوم بأرجاع المحتوي وتقراء وتلفذ أوامر HTML

قراءة محتوي العناصر بلغة جافا سكريبت

- مقارنة بين خاصية textContent وinnerText
- سوف تقرأ خاصية textContent جميع العناصر الداخلية شامله العناصر التي لا تعرض علي المتصفح.

```
<html>
<head>
<title>get element by id JavaScript</title>
</head>
<body>
    <h1>Welcome To Dom</h1>
    <p id="apple"> Get element by id name <span
style="display:none"> this text hidden </p>
    <p> Get elements by class name </p>
    <p> Get elements by tag name</p>
    <p> Get elements by name</p>

     <script>
       let apple = document.getElementById('apple');
       document.write(apple.innerText);
       document.write("<br>");
       document.write(apple.textContent);
     </script>
```

تحديث المحتوى النصي بلغة جافا سكريبت

الاحتفاظ بالمحتوي السابق واضافة المحتوي الجديد بعدة عن طريق خاصية innerText.

```
<body>
<p id="myParagraph">هذه بعض النصوص الافتراضية.</p>
<button onclick="addMoreText()"> إضافة المزيد من
النصوص</button>
<script>
function addMoreText() {
  const paragraph = document.getElementById("myParagraph");
  paragraph.innerText += " هذه جملة جديدة.";
}
</script>
```

عنصر بلغة جافا سكريبت HTMLتحديث محتوي

```
<body>
<div id="myDiv">هذا هو المحتوى الأصلي.</div>
<button onclick="updateContent()">تحديث المحتوى</button>

<script>
function updateContent() {
  document.getElementById("myDiv").innerHTML = "<b>هذا هو المحتوى الجديد!</b>";
}
</script>
```

دالة DOM Getattribute() في Javascript

دالة getAttribute () في الشجرة DOM في لغة جافا سكريبت

تستخدم دالة getAttribute () في لغة جافا سكريبت بقراءة قيمة Value أي خاصية Attribute لأي عنصر في صفحة HTML عن طريق جافا سكريبت.

```
body>
<a id="myLink" href="https://www.example.com">رابط إلى مثال</a>

<script>
  const link = document.getElementById("myLink");
  const linkURL = link.getAttribute("href");
  alert("عنوان الرابط هو: " + linkURL);
</script>
</body
```

Example

```
<html>
<head>
<title>getAttribute JavaScript</title>
</head>
<body>
  <h1>Welcome To Dom</h1>
  <p id="apple" class="memory"> Get element by id name</p>
  <p  class="text"> Get elements by class name</p>
  <p  class="text"> Get elements by tag name</p>
  <p  class="text"> Get elements by name</p>
  <hr>
   <script>
       let apple = document.getElementById('apple');
       apple.getAttribute('class')
       document.write("Element class value : " + apple.getAttribute('class'));
       document.write("<br>");
       document.write("Element id value : " + apple.getAttribute('id'));
   </script>
```

SetAttribute() DOM Method In JavaScript

- تستخدم دالة setAttribute () في لغة جافا سكريبت لأضافة خصائص Attributes جديدة أو التعديل علي الخصائص Attributes الحالية لأي عنصر في صفحة HTML بلغة جافا سكريبت.
- setAttribute() method Syntax
- كيف تكتب الدالة ؟

```
<img id="myImage" src="old_image.jpg" alt="صورة قديمة">
<button onclick="changeImage()">تغيير الصورة</button>
<script>
function changeImage() {
  const image = document.getElementById("myImage");
  image.setAttribute("src", "new_image.jpg");
  image.setAttribute("alt", "صورة جديدة");
}
</script>
```

Example

```
<!DOCTYPE html>
<html>
<head>
<title>setAttribute JavaScript</title>
</head>
<body>
   <h1>Welcome To Dom</h1>
   <p id="apple"> Get element by id name</p>
   <p  class="text"> Get elements by class name</p>
   <p  class="text"> Get elements by tag name</p>
   <p  class="text"> Get elements by name</p>
   <hr>
    <script>
        let apple = document.getElementById('apple');
        document.write("get class value before add : " + apple.getAttribute('class'));
        document.write("<br>");
        apple.setAttribute('class','memory');
        document.write("New class value : " + apple.getAttribute('class'));
    </script>
</body>
</html>
```

إنشاء عناصر جديدة داخل صفحة HTMLبلغة جافا سكريبت

تستخدم دالة createElement () في لغة جافا سكريبت لأضافه عناصر جديدة في صفحة HTML عن طريق لغة جافا سكريبت ولكن يجب عليك تتبع هذه العملية بإضافة محتوي لهذا العنصر عن طريق دالة innerText أو innerHTML أو textContent ثم تتبع هذه العملية ايضاً بدالة appendChild () لألحاق هذا العنصر بأب paraent متفرع من شجرة DOM الخطوات:

- انشاء عنصر جديد عن طريق دالة createElement()
- اضافة محتوي نصي أو محتوي HTML عن طريق هذه الدوال دالة innerText أو innerHTML أو textContent.
- ألحاق العنصر الجديد داخل أب Parent في شجرة DOM عن طريق دالة appendChild.()

Example

```
<div id="myDiv">
بعض المحتوى هنا...
</div>
<script>
// إنشاء الفقرة
const newParagraph = document.createElement("p");
newParagraph.textContent = "هذه فقرة جديدة!";
const parentElement = document.getElementById("myDiv");
parentElement.appendChild(newParagraph);
// إنشاء الرابط
const newLink = document.createElement("a");
newLink.href = "https://www.example.com";
newLink.textContent = "رابط إلى مثال";
document.body.appendChild(newLink);
</script>
```

Result

```
<div id="myDiv">
بعض المحتوى هنا ...
  <p>هذه فقرة جديدة!</p>
</div>

<a href="https://www.example.com">رابط إلى مثال</a>
```

إنشاء عناصر جديدة داخل صفحة HTMLبلغة جافا سكريبت

يمكن اضافة عنصر جديد عن طريق دالة createElement () ودالة prepend()ت ضيف العنصر في parentقبل جميع العناصر.

```
<ul id="myList">
  <li> عنصر1</li>
  <li> عنصر2</li>
</ul>
<button onclick="addNewItem()">إضافة عنصر جديد</button>
<script>
function addNewItem() {
  const newListItem = document.createElement("li");
  newListItem.textContent = "عنصر جديد";
  const myList = document.getElementById("myList");
  // إضافة العنصر الجديد في البداية باستخدام prepend
  myList.prepend(newListItem);
}
</script>
```

```
<ul id="myList">
  <li>عنصر جديد</li>
  <li> عنصر1</li>
  <li> عنصر2</li>
</ul>
```

انشاء عنصر بعد الضغط علي زر الفأرة MOUSEفي لغة جافا سكريبت

من الضروري إظهار أجزاء عند تفاعل المستخدم من صفحة الويب عندما يقوم بالنقر علي جزء معين وذلك عن طريق جافا سكريبت .Events

Add element after click

اضافة عنصر بعد النقر علي زر الفأرة mouseداخل صفحة HTMLفي لغة جافا سكريبت.

```
<body>
<button id="addButton">أضف مربعًا</button>
<div id="container"></div>

<script>
  const addButton =
document.getElementById('addButton');
  const container = document.getElementById('container');
  addButton.addEventListener('click', () => {
    const newDiv = document.createElement('div');
    newDiv.style.width = '100px';
    newDiv.style.height = '100px';
    newDiv.style.backgroundColor = 'blue';
    newDiv.style.margin = '10px';
    container.appendChild(newDiv);
  });
</script>
```