المحاضرة السادسة أساسيات لغة JavaScript

برمجة ويب1 – BWP401

SLASH TEAM

برنامج الهندسة المعلوماتية – الجـــــامعة الافتراضية السورية

تقديم	بتول بيان
وصف	شرح أساسيات لغة جافا سكريبت
عدد الصفحات	14

لغة جافا سكريبت JAVASCRIPT

توصيف المحاضرة

التوصيف:

إعطاء أساسيات لغة البرمجة جافا سكريبت.

طريقة الدراسة:

قراءة الملف بشكل جيد والتمرين على كتابة الأكواد بالإضافة الى التدرب على تمارين الموجودة ضمن موقع: https://www.w3schools.com/js/default.asp

مقدمة عن JAVASCRIPT

ماهي لغة جافا سكريبت:

هي لغة برمجة تم تطويرها بشكل أساسي لجعل صفحات الويب تفاعلية فمن خلالها يمكنك تحديث ما سيحدث عند النقر على الأزرار، فحص ما يدخله المستخدم في مربعات النصوص، حذف عناصر في الصفحة....الخ.

جافا سكريبت غير محصورة في برمجة المواقع فقط حيث أنها تستخدم أيضا في تطبيقات الهاتف، تطوير تطبيقات الديسك توب وبرمجة الألعاب.

تختصر جافا سكريبت JavaScript بالحرفين JS. قبل البدء بتعلم JavaScript عليك اتقان CSS,HTML

قواعد كتابة الكود في جافا سكريبت

مبادئ كتابة الكود في جافا سكريبت:

عند كتابة كود جافا سكريبت يجب مراعاة القواعد المتعارف عليها في كتابة الكود حيث أن مهما كان المجال الذي نستخدم ضمنه هذه اللغة فإن طريقة الكتابة هي ذاتها.

أساليب كتابة الكود في جافا سكريبت:

كود جافا سكريبت هو عبارة عن مجموعة أوامرStatements يتم تنفيذها بالترتيب إذا كانت مكتوبة بشكل صحيح أما إذا وجد أمر مكتوب بشكل خاطئ فإنه لن يتم تنفيذه ولن يتم تنفيذ أي كود بعده.

حساسية الأحرف: في جافا سكريبت يوجد حساسية للأحرف (case sensitivity) وهذا يعني أنها تميز بين الأحرف الكبيرة والصغيرة. مثال: batool تختلف عن Batool

أسماء المعرفات: جافا سكريبت تعتمد أسلوب camel case في كتابة الأسماء. أي عند اختيار أسماء للمتغيرات أو للدوال يجب كتابة الاسم بأحرف صغيرة أو إذا كان الاسم يتألف أكثر من كلمة نبدأ أول كلمة بأحرف صغيرة وكل كلمة بعدها نبدأها بحرف كبير. مثال: batooBayyan

الفاصلة المنقوطة: يعتبر وضع فاصلة منقوطة في جافا سكريبت هو أمر اختياري ولكن يفضل وضعها من أجل ترتيب الكود ومعرفة نهاية كل تعليمة.

المسافات الفارغة: لا يسبب وجودها مشكلة ولكن يفضل استخدامها فقط لترتيب الكود.

التعليقات: تستخدم لوضع ملاحظات على الكود حيث أنها لا تؤثر إطلاقا على الكود المكتوب.

لكتابة التعليق إذا كان من سطر واحد نستخدم // مثال:

```
// this is a one-line comment      تعليق يتألف من سطر واحد
```

إذا كان يتألف من أكثر من سطر نستخدم /* نكتب هنا التعليق */ مثال:

```
/*
    This is multiline comment
    هذا التعليق يتألف من عدة أسطر
*/
```

الأحرف المستخدمة في جافا سكريبت: أي اسم نضعه يسمى identifier حيث أن كل عنصر نريد تعريفه علينا إعطاؤه اسم خاص غير مكرر.

في حال قمت بتعريف شيء جديد وكنت بالفعل قد استخدمت الاسم نفسه في تعريف شيء آخر، فإن الشيء الجديد سيحل مكان الشيء القديم الذي تم تعريفه بنفس الاسم.

قواعد الزامية عند إعطاء الأسماء:

- يجب أن يبدأ بحرف كبير بين A-Z أو حرف صغير بين a-z أو رمز الدولار $ أو ال under score _.
- يمنع أن يبدأ برقم.
- يمنع استخدام القيم الموجودة في جافا سكربت Literals كأسماء، أي يمنع استخدام القيمة true والقيمة false.
- يمنع استخدام أي كلمة من الكلمات المحجوزة Keywords.

الكلمات المحجوزة: يوجد الكثير من الكلمات المحجوزة إذا أردت الاطلاع عليها من موقع:

https://harmash.com/tutorials/javascript/reserved-keywords

أمثلة لأسماء يمكن استخدامها: _name, $password, num, Sum

أمثلة لأسماء لا يمكن استخدامها: -name, 1st ,@user

كتابة أول كود جافا

أين نكتب كود الجافا سكريبت:

يوجد طريقتين :

1. كتابة ضمن ملف Html قبل تاغ إغلاق ال body ضمن تاغ <script></script> نكتب كود جافا سكريبت.

```
<body>
  <h1>welcome</h1>
  <script>
    // write here JavaScript code
  </script>
</body>
```

2. انشاء ملف بامتداد js. وتضمينه داخل ملف الhtml المراد من خلال إضافة الخاصية src للتاغ script ووضع مسار الملف كقيمة للخاصية.

```
<body>
  <h1>welcome</h1>
  <script src="code.js"></script>
</body>
```

أوامر أساسية في جافا سكريبت:

أوامر نتعامل معها كثيرا تتيح رؤية مخرجات الأوامر التي تتعلمها.

1. الامر document.write: يتيح هذا الأمر طباعة أي شيء نريده في صفحات الويب.

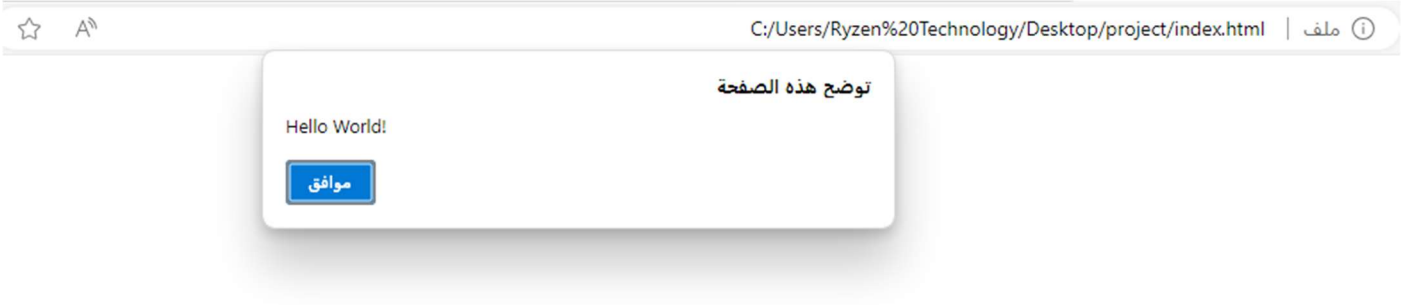
```
<body>
    <h1>welcome</h1>
    <script>
        document.write('Added by JavaScript!');
    </script>
</body>
```

الخرج:

2. الأمر alert(): يظهر لنا نافذة تظهر رسالة من اختيارنا.

```
<h1>welcome</h1>
<script>
    alert('Hello World!');
</script>
```

الخرج:



3. الامر document.getElementById: يتيح هذا الامر الوصول الى عنصر موجود داخل صفحات الويب من خلال المعرف id الذي قمت بإعطائه للعنصر. بعد الوصول للعنصر نستطيع التحكم به.
4. الخاصية innerHtml: أي عنصر في صفحات الويب نصل له بواسطة جافا سكريبت يمكن إعطاؤه محتوى جديد بواسطة هذه الخاصية .حيث أن المحتوى ممكن أن يكون نص أو يكون كود html مما يعني أننا يمكننا إضافة عناصر للصفحة.

مثال:

```
<body>
        <h2>JavaScript innerHTML Property</h2>
        <p id="demo"></p>
        <script>
            document.getElementById('demo').innerHTML = 'Added by <b>JavaScript</b>';
        </script>
</body>
```

الخرج:

5. الخاصية style: أي عنصر يمكنك ان تصل له في جافا سكريبت يمكنك تعديل تصميمه بواسطة هذه الخاصية.

مثال:

```
<body>
        <h2>JavaScript Alert</h2>
        <p id="demo">JavaScript changed my color to red.</p>
        <script>
            document.getElementById('demo').style.color = 'red';
        </script>
</body>
```

الخرج:

6. الأمر confirm() أو widow.confirm(): لإظهار نافذة منبثقة تطلب فيها من المستخدم الموافقة على أمر ما.

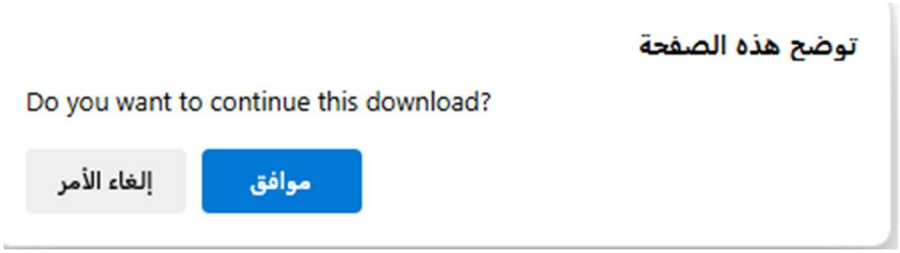
عند استدعائها يجب تمرير النص الذي نريد إظهاره في النافذة بين القوسين.

عندما يتم إغلاق النافذة فهناك حالتين: الأولى المستخدم ضغط على زر الموافقة وهنا سترجع القيمة true وإذا ضغط على زر الرجوع فإنه سترجع القيمة false.

مثال:

```
<script>
        var question =
        confirm("Do you want to continue this download?");
</script>
```

الخرج:



7. الأمر prompt: إظهار نافذة لإدخال قيمة prompt() أو window.prompt()

يمكنك استخدام الدالة prompt() أو window.prompt() لإظهار نافذة منبثقة تطلب فيها من المستخدم إدخال قيمة ما. عند استدعائها يجب أن نمرر لها بين أقواسها النص الذي سنخبره فيه بأن عليه إدخال قيمة، ويمكننا تمرير نص آخر لها ليكون القيمة الافتراضية التي سيتم وضعها في مربع النص. عندما يتم إغلاق هذه النافذة فهنا يوجد حالتين:

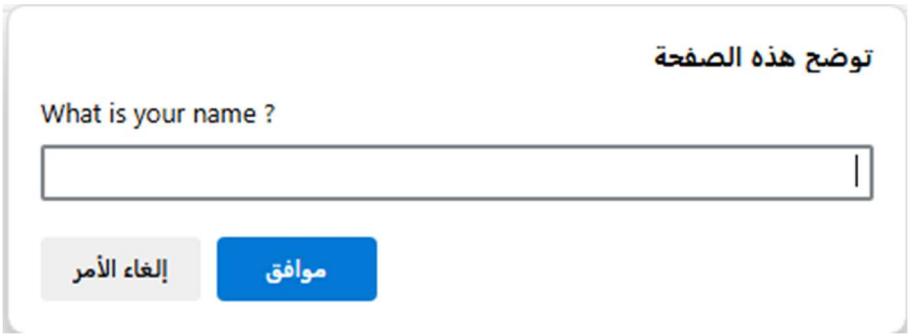
الأولى: المستخدم نقر على زر الموافقة وهنا سترجع القيمة التي أدخلها كنص.

الثانية: المستخدم نقر على زر الرجوع وهنا سترجع القيمة null.

مثال:

```
a = prompt("What is your name ?", "");
```

الخرج:



أساسيات JAVASCRIPT

المتغيرات:

المتغيرات هي عبارة عن حاويات تستعمل لتخزين القيم بشكل مؤقت مهما كان نوعها.

يوجد أربع طرق لتعريف المتغيرات:

1. التعريف التلقائي.
2. التعريف باستخدام كلمة var.
3. التعريف باستخدام كلمة let.
4. التعريف باستخدام كلمة const.

المتغيرات تحذف بحال الخروج من صفحة الويب.

مثال عن العريف متغير بشكل تلقائي:

```
<script>
        // هنا قمنا بتعريف المتغيرات و إعطاءها قيم
        width = 100;
        height = 50;
        area = width * height;
        // area هنا قمنا بطباعة القيمة الموجودة في المتغير
        document.write('Area = ' + area + 'cm');
</script>
```

الخرج:

Area = 5000cm

ملاحظة: اختر أسماء مناسبة للمتغيرات تعبر عن غرضها.

مثال تعريف متغيرات باستخدام الكلمة المحجوزة var:

```
<script>
    // هنا قمنا بتعريف المتغيرات و إعطاءها قيم
    var name = 'batool';
    var age = 22;
    var isWorking = true;
    // هنا قمنا بطباعة جميع القيم الموجودة فيها
    document.write(name + '<br>');
    document.write(age + '<br>');
    document.write(isWorking);
</script>
```

الخرج:

batool
22
true

إذا أردت تعلم لغة جافا سكريبت بالتفصيل أبحث عن باقي طرق تعريف المتغيرات.

أنواع المتغيرات الأساسية:

أي نوع نتعامل معه في جافا سكريبت له نوع محدد، في جافا سكريبت تنقسم أنواع الأنماط الى ثلاث أنواع:

فقط للاطلاع:

1. بيانات بدائية: مثل النصوص (strig) الأرقام (number) القيم المنطقية (boolean).
2. بيانات مرجعية: مثل الكائنات (object) الدوال (function) المصفوفات (array).
3. بيانات خاصة وهي القيمة undefined والقيمة null.

أنواع البيانات البدائية:

- القيم العددية (number): القيم العددية هي الأعداد بكافة أشكالها (الأعداد الصحيحة والعشرية والأسية....الخ)

مثال:

```
<script>
    // هنا قمنا بتعريف متغيرات وضعنا فيها أعداد صحيحة
    let x = 3;
    let y = 5;
    // هنا قمنا بطباعة قيم المتغيرات التي قمنا بتعريفها
    document.write('x = ' + x + '<br>');
    document.write('y = ' + y);
</script>
```

الخرج:

x = 3
y = 5

```
<script>
    // هنا قمنا بتعريف متغيرات وضعنا فيها أعداد عشرية
    let x = 10.5;
    let y = 2.44;
    // هنا قمنا بطباعة قيم المتغيرات التي قمنا بتعريفها
    document.write('x = ' + x + '<br>');
    document.write('y = ' + y);
</script>
```

الخرج:

x = 10.5
y = 2.44

- القيم النصية (string): هي الأحرف الكلمات الجمل النصوص حيث يمكن البحث فيها، تقطيعها، دمجها...

القيم النصية يمكن وضعها بين '... ' أو"..."

مثال:

```
<script>
    // هنا قمنا بتعريف متغيرات وضعنا فيها نصوص
    var web1 = 'batool';
    var web2 = 'bayyan';
    // هنا قمنا بطباعة قيم المتغيرات التي قمنا بتعريفها
    document.write('web1 = ' + web1 + '<br>');
    document.write('web2 = ' + web2);
</script>
```

الخرج:

web1 = batool
web2 = bayyan

مثال دمج النصوص:

```
<script>
    var firstName = 'batool';
    var lastName = 'bayyan';
    // firstName وضعنا فيه قيمة المتغير fullName المتغير
    // lastName يليه مسافة فارغة، يليه قيمة المتغير
    let fullName = firstName + ' ' + lastName;
    // fullName هنا قمنا بطباعة قيم المتغير
    document.write('fullName = ' + fullName);
</script>
```

الخرج:

fullName = batool bayyan

القيم المنطقية (boolean): تستخدم حين يكون هناك احتمالين فقط (true or false).

```
<script>
    // هنا قمنا بتعريف متغيرات وضعنا فيها نصوص
    var isWorking = true;
    var isMarried = false;
    // هنا قمنا بطباعة قيم المتغيرات التي قمنا بتعريفها
    document.write('isWorking = ' + isWorking + '<br>');
    document.write('isMarried = ' + isMarried);
</script>
```

الخرج:

```
isWorking = true
isMarried = false
```

العمليات الحسابية:

	رمزه	مثال	الشرح
الاسناد	=	a=b	إعطاء a قيمة b
الجمع	+	a+b	جمع القيمتين
الطرح	-	a-b	طرح القيم
الضرب	*	a*b	ضرب القيمتين
القسمة	/	a/b	قسمة a على b
باقي قسمة	%	a%b	الحصول على باقي قسمة a علىb
زيادة	++	a++	لإضافة 1 على القيمة a تستخدم في الحلقات التكرارية
نقصان	--	b++	لإنقاص 1 على القيمة a تستخدم في الحلقات التكرارية
أضافة وأسناد	+=	a+=b	أضف قيمة a على قيمة b وخزن الناتج في a وكذلك في باقي العمليات الطرح الضرب القسمة.
Type of	typeof	typeof a	معرفة نوع القيمة المخزنة ضمن المتغير a

التعامل مع الأرقام:

كما ذكرنا سابقا أن في جافا سكريبت لا نحتاج لتعريف للعدد الصحيح نوع وللعدد العشري نوع وانما اللغة توفر معرفة نوع المتغير من القيمة المخزنة ضمنه على عكس باقي لغات البرمجة.

حيث يوجد للأعداد في جافا سكريبت مجموعة من الخصائص منها:

, POSITIVE_INFINITY, NEGATIVE_INFINITY, PI. MAX_VALUE, MIN_VALUE, NaN

حيث أن Number.MIN_VALUE هذا الثابت يحتوي على أدنى عدد يمكن تخزينه في جافا سكربت.

Number.POSITIVE_INFINITY هذا الثابت يمثل عدد تجاوز الحد الأقصى للعدد الذي يمكن تخزينه في جافاسكربت. إذاً قيمته تساوي القيمة Infinity

Number.MAX_VALUE هذا الثابت يحتوي على أعلى عدد يمكن تخزينه في جافاسكربت.

بالتأكيد يوجد خصائص أخرى لمعرفتها ابحث عنها لأنها لا تنتهي .

مفهوم القيمة NAN:

مترجم جافا سكريبت في حال وجد أن العملية الحسابية المراد تنفيذها غير ممكنة أي غير منطقية فإنه يعطيه نتيجة هذه العملية NAN وهي اختصار ل not a number أي انه ليس رقم ليتمكن من تفيذ العملية.

مثال:

```
 <script>
let x = 10 - 'batool';  // NaN
document.write('x = ' + x);
        </script>
```

الخرج:

x = NaN

الطريقة الصحيحة لفحص القيمة NAN:

في حال قمت بفحص نوع المتغير بواسطة العامل typeof وكان هذا المتغير يحتوي على NaN كقيمة، فإن نوع المتغير سيكون number.

أما لو أردت فحص قيمة المتغير لتعرف ما إن كان يمكن تمثيلها أو التعامل معها كعدد، فهنا يجب أن تفحصها بواسطة الدالة isNaN المخصصة لذلك.

مثال:

```
<script>
    var x = 10 - 'batool';     // NaN
    document.write(typeof x);  // number => هكذا نعلم أنها في الأصل تمثل جواب لعملية حسابية
    document.write(isNaN(x));  // true   => هكذا نعلم أنها قيمة عددية لا يمكن تمثيلها
</script>
```

الخرج:

numbertrue

التعامل مع النصوص:

كما ذكرنا مسبقا أن السلسة النصية ليس لها حجم محدد ممكن أن تكون حرف أو نص حيث يمكنك تعريف النص بواسطة " " أو ' ' ففي حال قمنا بتعريف النص باستخدام " " نستطيع استخدام ' ' بداخله والعكس صحيح

مثال:

```
<Script>
        var str1 = 'She said "Hey"';
        var str2 = "Don't eat sugar";
</Script>
```

في حال أردنا استخدام الرمز نفسه نضع قبله \ لكي يفهم المترجم (compiler) أنك تريد اعتباره أنه محرف عادي.

مثال:

```
<Script>
         str1 = "She said \"Hey\"";
         str2 = 'Don\'t eat sugar';
</Script>
```

إذا أردنا كتابة \ يجب وضع قبله \ لكي يفهم ال compiler أنها محرف عادي وبالتالي نفهم أن مهمة ال \ هي تعطيل عمل الرمز الأساسي.

مثال:

```
<script>
     str = "Here, backslash is used as \\Normal Charater\\";
     document.write(str);
</script>
```

الخرج:

Here, backslash is used as \Normal Charater\

تعريف قالب نصي في JS:

وهي طريقة نقوم بتعريف متحول واعطائه نص معين وعند الحاجة لهذا المتحول نستخدمه (بحال كان هذا النص سيتكرر ضمن الموقع) وهي أشبه بجمع النصوص.

مثال:

```
<script>
     username = "batool";
     totalMessages = 10;
document.write(text = `Good morning ${username}, you have ${totalMessages}
messages.`);
```

```
</script>
```

الخرج مع ملاحظة أننا استخدمنا عبارة الطباعة في الموقع مباشرة عند تعريف المتحول:

Good morning batool, you have 10 messages.

جمع السلاسل النصية: تستخدم (+) لجمع سلسلتين أو أكثر.

مثال:

```
<script>
            var x = "batool";
             x = x + "bayyan";
             // x now is batoolbayyan
            document.write( x, "<br />");
</script>
```

الخرج:

batoolbayyan

مصطلحات مهمة حول التعامل مع النصوص:

- عدد أحرف النص يسمى length وهذه الخاصية لمعرفة طول النص ونقوم بالعد بشكل عادي أي المحرف الأول هو 1
- رقم أي خانة في النص يسمى index وهو فهرس النص ويبدأ من الصفر مثل باقي لغات البرمجة.
- إذا قمنا بأخذ جزء من النص يسمى substring

مثال:

```
<script>
            let str = 'Welcome to SVU';
            document.write('Total characters = ' + str.length);
</script>
```

الخرج:

Total characters = 14

الدوال الخاصة بالنصوص:

يوجد الكثير من الدوال الخاصة بالنصوص لكن سنذكر بعض منها وهي:

- charAt(index): ترجع الحرف الموجود على الخانة التي مررنا رقمها ضمن البارميتر في النص الذي قمنا باستدعائه.

مثال:

```
<script>
      let str = 'batool by';
      document.write(str.charAt() + '<br>');
      document.write(str.charAt(0) + '<br>');
      document.write(str.charAt(1) + '<br>');
      document.write(str.charAt(2) + '<br>');
      document.write(str.charAt(3) + '<br>');
      document.write(str.charAt(4) + '<br>');
      document.write(str.charAt(5) + '<br>');
      document.write(str.charAt(6) + '<br>');
</script>
```

الخرج:

b
b
a
t
o
o
l

- indexOf(str,indexStart): تبحث في النص الذي قمنا باستدعائه عن أول خانة ابتداء من عندها نفس النص الذي نمرره لها مكان البارامیتر str وترجعه. أما البارامیتر indexStart فهو اختياري يمكن من خلاله تحديد رقم الخانة التي نريد البحث من عندها.

حال لم تجد ما نبحث عنه ضمن النص ترجع 1-

مثال:

```
<script>
    let str = 'live your life. the life is simple.';
    // هنا سيتم طباعة الرقم 3 لأنه رقم أول خانة تم فيها
    // 'e' إيجاد الحرف
    document.write(str.indexOf('e'));
    document.write('<br>');
    // هنا سيتم طباعة الرقم 13 لأنه رقم أول خانة تم فيها
    // 4 عند بدء البحث من الخانة رقم 'e' إيجاد الحرف
    document.write(str.indexOf('e', 4));
    document.write('<br>');
    // هنا سيتم طباعة الرقم 10 لأنه رقم أول خانة تم عندها
    // 'life' إيجاد الكلمة
    document.write(str.indexOf('life'));
    document.write('<br>');
    // هنا سيتم طباعة الرقم 20 لأنه رقم أول خانة تم عندها
    // 20 عند بدء البحث من الخانة رقم 'life' إيجاد الكلمة
    document.write(str.indexOf('life', 20));
```

```
    document.write('<br>');

    // 'X' هنا سيتم طباعة الرقم 1- لأن النص لا يحتوي على الحرف
    document.write(str.indexOf('X'));
</script>
```

الخرج:

3
13
10
20
-1

- substring(indexStart,indexEnd): تستخدم للحصول على جزء معين من النص الذي قمنا باستدعائه.

البارامتر الأول نمرر رقم الindex الذي نريد البدء من عنده.

البارامتر الثاني هو بارامتر اختياري نمرر رقم ال index الذي نريد انهاء السلسلة المقتطعة من الأصلية.

مثال:

```
<script>
    let str = 'JavaScript is easy to learn.';
    // هنا سيتم طباعة النص الموجود إبتداءاً من الخانة رقم 0 وصولاً إلى الخانة رقم 9
    document.write(str.substring(0, 10));
    document.write('<br>');
    // هنا سيتم طباعة النص الموجود إبتداءاً من الخانة رقم 14 وصولاً إلى الخانة رقم 17
    document.write(str.substring(14, 18));
    document.write('<br>');
    // هنا سيتم طباعة النص الموجود إبتداءاً من الخانة رقم 14 و حتى آخر حرف
    document.write(str.substring(14));
</script>
```

الخرج:

JavaScript
easy
easy to learn.

- toLowerCase(): استرجاع النص الذي قمنا باستدعائه بأحرف صغيرة كلها.

toUpperCase(): استرجاع النص الذي قمنا باستدعائه بأحرف كبيرة كلها.

كما ذكرنا يوجد الكثير من الدوال المفيدة عند التدرب على اللغة بشكل جيد والبحث المستمر سنقوم بمعرفة أشياء أكثر.

الغرض DATE:

يمكن تعريف متغير يأخذ قيمة التاريخ والوقت الحالي.

```
var today= new Date();
```

يوجد الكثير من الدوال التابعة له عند حاجتها قم بالبحث عنها لانهم غير مهمين حالياً.

الغرض MATH:

الدالة وتعريفها
Math.abs(x): ترجع القيمة المطلقة للعدد الذي نمرره كبارامتر
Math.cbrt(x): ترجع الجذر التكعيبي للقيمة التي قمنا بتمريرها.
Math.floor(x): ترجع العدد الصحيح الأصغر أو العدد نفسه أن كان عدد صحيح.
Math.max(x1,x2.x3...): ترجع العدد الأكبر بين هذه الأعداد.

وبالتأكيد هنالك دوال كثيرة خاصة بالغرض math وهذه بعض منها.