

مقرر
مبادئ صفحات الانترنت

اعداد
وجدان الشريف

الفصل الأول

لغات تطوير الويب

العناوين:

١. نبذة عن لغات تطوير الويب
٢. الجافا سكريبت و استخداماتها

لغات تطوير الويب

لغات تطوير الويب هي لغات برمجة تستخدم اوامر نصية لتمكين صفحة الويب من فعل أمر ما و تنقسم الى قسمين:

- لغات العميل
- لغات الخادم

الفرق بين القسم الأول و الثاني:

القسم الأول لغات العميل:

يتم معالجة الكود الخاص به من خلال متصفح الزائر – فأنت تعرف أن أي لغة برمجة تحتاج معالج يترجم شفراتها (compiler) – .

القسم الثاني لغات الخادم:

فيتم معالجة أكواده من خلال الخادم (server) ثم اعطاء الناتج بعد ذلك الى متصفح الزائر .
و لعلك تتساءل لماذا لا يستخدم أحدهما كبديل للآخر طالما أن القسمين متخصصين في تطوير الويب ؟

الاجابة أن لغات العميل مثل الجافا سكريبت (java script) و (vbscript) الهداف منها اضافة المؤثرات و ادخال الحيوية الى الصفحة فمثلا تستطيع تغيير لون الخلفية باختلاف وقت الزيارة، و اعطاء الزائر رسالة تختلف باختلاف موطنه و لكن ماذا اذا ارت ان تجعل الزوار يقيموا موقعك أو أن يبديوا اراءهم في موضوع معين ، هنا ستظهر الحاجة الى استخدام لغة تقوم بتخزين و استرجاع البيانات من الخادم وهذا ما تعجز عنه لغات الجانب الشخصي و تنجح في تنفيذه لغات الخادم (server) ومن هذه اللغات ASP, PHP .

لغات الخادم	لغات العميل
تخزين و استرجاع البيانات	- تنسيق صفحات الويب و اعطاءها حيوية - اضافة التأثيرات و الحركات على صفحة الويب	استخدامها
ASP - PHP -	JavaScript- Vbscript-	مثال

و الان بعد أن تعرفنا على الفرق بين لغات العميل و لغات الخادم سنتطرق الى ما يخصنا في هذا المنهج وهو لغات الجانب الشخصي أو لغات العميل و منها الجافا سكريبت.

الأساس في تصميم صفحات الويب هو لغة الـ HTML ولكنها تنتج صفحات ثابتة لا تتفاعل مع الزائر (static pages) و من هنا جاءت الحاجة لانتاج لغة تضيف الحيوية لصفحات الويب و تم انتاج لغة الجافا سكريبت و الجدير بالذكر أن لغة الجافا سكريبت تختلف كثيرا عن لغة الجافا فلغة الجافا لغة متكاملة أي يمكن بناء برامج ذاتية التشغيل بواسطتها أما لغة الجافا سكريبت فهي خاصة بصفحات الانترنت فقط و يتم ترجمة أكوادها من خلال المتصفح و بذلك لا يمكنها بناء برامج ذاتية التشغيل و لكنها تشبه لغة الجافا في أسلوب الكتابة حيث أن الاثنان مشابهتان للغة السي C.

استخدامات لغة الجافا سكريبت عديدة جدا في صفحات الويب وذلك لقدرتها على التحكم بصفحة الويب و التعامل معها بسهولة و هذا ما ستجده بنفسك عند التعمق في دراسة هذه اللغة .

و استخدام لغة الجافا سكريبت لا يتاثر بالخادم الذي يستضيف موقعك لأنها كما قلنا سابقا يتم ترجمة أكوادها من خلال متصفح الزائر لذلك فهي تعتمد على نوع المتصفح مثل متصفح Microsoft Internet Explorer.

لغة الجافا سكريبت هي أكثر لغات البرامج النصية شيوعا لانها مدعومة من قبل أشهر برامج التصفح للويب : مثل Internet Explorer و Netscape كما يمكن تمكينها في متصفح Google Chrome ، ورغم ذلك يمكن استخدام لغات اخرى لاضافة التفاعل مع المستخدم مثل Vbscrit ولكنها مدعومة من قبل متصفح Internet Explorer فقط.

ادخال عبارات الجافا سكريبت في حقل العناوين داخل المستعرض :

تتكون لغة الجافا سكريبت من عبارات Statements التي تقوم بمهام معينة عند تنفيذها على سبيل المثال تعرض عبارة

```
document.write("hello")
```

كلمة hello على الشاشة داخل نافذة المستعرض.

تستطيع المستعرضات أن تفسر و تنفذ عبارات جافا سكريبت و يمكنك أن تكتب هذه العبارات داخل صفحة الويب أو في حقل العناوين داخل نافذة المستعرض مباشرة.

لكي تجعل مستعرض الويب ينفذ عبارة جافا سكريبت ، اكتب كلمة

JavaScript:

في حقل العناوين ثم اكتب العبارة التي تريدها ثم اضغط مفتاح الادخال Enter لتنفيذها.

يجب أن تضع الفاصلة المنقوطة (;) في نهاية عبارة الجافا سكريبت و من الممكن أن تكتب عدة عبارات ليتم تنفيذها دفعة واحدة ، لعمل ذلك ، افصل بين كل عبارة و أخرى بالفاصلة المنقوطة . (;)

على سبيل المثال اكتب العبارة

```
JavaScript:document.write("hello");
```

في حقل العناوين داخل المستعرض ثم اضغط مفتاح الادخال ستجد أن المستعرض عرض كلمة hello داخل النافذة .

إذا نسيت أن تكتب كلمة JavaScript فان المتصفح سيعتقد أن النص الذي كتبتة هو عنوان صفحة ويب و بالتالي سيحاول العثور عليها و بالطبع سيفشل في ذلك.

تضمين عبارات جافا سكريبت داخل مستند HTML :

يمكنك تضمين عبارات جافا سكريبت داخل صفحة الويب ضمن علامات <script> ، وسيتم تنفيذ عبارات جافا سكريبت عندما يتم تحميل صفحة الويب داخل المتصفح.

يمكن أن تضع علامات `<script>` في أي مكان داخل علامة `<head>` او علامة `<body>` و يمكن لصفحة الويب الواحدة أن تحتوي على عدة علامات `<script>` .

لا تحتاج برنامج خاص لكتابة صفحة ويب باستخدام HTML حيث أنه يمكنك استخدام تطبيقي المفكرة أو الدفتر الموجودين في جهاز الحاسب الخاص بك، و لكنه يجب عليك أن تحفظ المستند باسم المستند ثم ترفقه بالامتداد `.html`. حتى يحفظ كصفحة ويب.

مثال ١ انشاء صفحة ويب :

```
<html>
<head>
</head>
<body>
<script language="javascript">
</script>
</body>
</html>
```

بعد الانتهاء من كتابة النص البرمجي سنحفظه باسم `ex1.html` مثلا .

النص البرمجي السابق سيمكنني من انشاء صفحة ويب باسم `ex` ، و لكن صفحة الويب هذه ستكون فارغة نظرا لعدم وجود أوامر تنفيذية ظاهرة في النص البرمجي.

لكن لنفصل كل سطر في النص السابق لآخذ فكرة أفضل عن كيفية عمل نصوص الـ `html` :

١- `<html>`

نكتب هذا السطر لتعريف المتصفح باننا سنستخدم الـ `html`

٢- `<head>`

الرأس: هذا السطر يفترض به أن يحتوي على عنوان صفحة الـ `html`

٣- `</head>`

اغلق الرأس: بهذا السطر نغلق الرأس أو العنوان

٤- `<body>`

الجسم: هنا نستطيع أن نضيف العمليات و الفقرات النصية و النماذج .

-٥ <script language= "javascript">

تعريف الجافا سكريبت: بهذا السطر أخبر صفحة الويب أنني أريد استخدام الجافا سكريبت

-٦ </script>

بهذا السطر أوقف تعريف الجافا سكريبت.

-٧ </body>

اغلق الجسم: بهذا السطر نغلق الجسم

-٨ </html>

بهذا السطر أنهى النص البرمجي لصفحة الويب.

.....

مثال ٢ انشاء صفحة ويب بلون خلفية أزرق:

```
<html>
<head>
</head>
<body>
  <script language="javascript">
    document.bgColor="blue";
  </script>
</body>
</html>
```

النص البرمجي في مثال ٢ مطابق للنص في مثال ١ و لكن بزيادة سطر

```
document.bgColor="blue";
```

و هو السطر الذي يستخدم لتغيير لو الخلفية و الترجمة اللغوية له ستكون:

غير لون المستند الى اللون الأزرق

*نلاحظ أن حرف الـ c في كلمة `bgColor` حرف كبير الجافا سكريبت لغة حساسة لحالة الأحرف لذلك أتأكد أن اكتب الحروف بالشكل الصحيح فلو كتبت حرف الـ c صغير لكان هناك خطأ في النص البرمجي و بالتالي لن يتم تنفيذ الأمر.

*نلاحظ أيضا اننا استخدمنا ; الفاصلة المنقوطة في نهاية الأمر و ذلك لان أي امر يكتب داخل الجافا سكريبت يجب أن ينتهي بفاصلة منقوطة.

مثال ٣ اضافة نص لصفحة الويب داخل الجافا سكريبت:

```
<html>
<head>
</head>
<body>
  <script language="javascript">
    document.bgColor="blue";
    document.write("Computer Center");
  </script>
</body>
</html>
```

يزيد مثال ٣ عن المثالين السابقين بالسطر التالي:

```
document.write("Computer Center");
```

و هو الأمر المستخدم لإظهار نصوص على الشاشة باستخدام javascript

الترجمة اللغوية للسطر البرمجي هي :

اكتب Computer Center على الشاشة .

*نلاحظ أننا نستخدم أقواس التنصيص " " قبل و بعد النص .

و طبعا يجب حفظ المستند باستخدام الامتداد .html .

إضافة التعليقات الى أوامر الجافا سكريبت:

يمكنك أن تستخدم التعليقات لتشرح لنفسك أو لاي شخص يقرأ النص البرمجي الغرض من سطر او ميزة معينة اضفتها عليه ، التعليقات لا تظهر في صفحة الويب في المستعرض انما تظهر فقط في المستند الذي كتبت فيه النص البرمجي .

يمكنك اضافة سطر واحد من التعليقات بوضع // قبله.

و يمكنك أن تضيف عدة أسطر من التعليقات بأن تضع /* التعليق..... */ في بداية و نهاية التعليق، و سيتجاهل المستعرض أي عبارات أو نصوص بينها.

*التعليقات يمكن أن توضع في أي مكان بعد <script> الى </script> .

.....

أسئلة و تمرينات

(١) ما الفرق بين لغات العميل و لغات السيرفر؟

(٢) طبق التالي:

- افتح ملف جديد في برنامج المفكرة .
- اكتب عبارات جافا سكريبت لظهار اسمك و اسم المعهد على الشاشة بخلفية حمراء و خط عريض .
- احفظ الملف باسم ex.html

الفصل الثاني

المتغيرات

العناوين:

١. تعريف المتغيرات
 ٢. العمليات على المتغيرات
 ٣. العمليات الرياضية
 ٤. العمليات على المتغيرات النصية:
- المقارنة بين المتغيرات
 - العلاقات المنطقية

المتغيرات

المتغيرات: هي وعاء تخزيني للقيم.

كيفية استخدام المتغير:

نقوم بتعريف المتغير باسم معين و هكذا أصبح لدينا وعاء يمكننا اسناد (تخزين) أي قيمة اليه.

نسند قيمة لهذا المتغير .

عند الرغبة في استخدام هذا القيمة نشير اليها باسم المتغير.

و الخلاصة أن المتغير وعاء له اسم و يحتوي على قيمة ، لاستخدام هذه القيمة نشير اليها عن طريق الاسم.

مثال:

```
<script language="javascript">
Var mah;
mah=15;
Alert(mah);
</script>
```

١- تعريف الجافا سكريبت للـHTML

٢- تسمية المتغير

المتغير (Variable) يرمز له ب Var تمت تسميته ب mah

٣- اعطينا قيمة للمتغير mah

٤- صفحة الويب ستظهر انذار (alert) بقيمة المتغير mah .

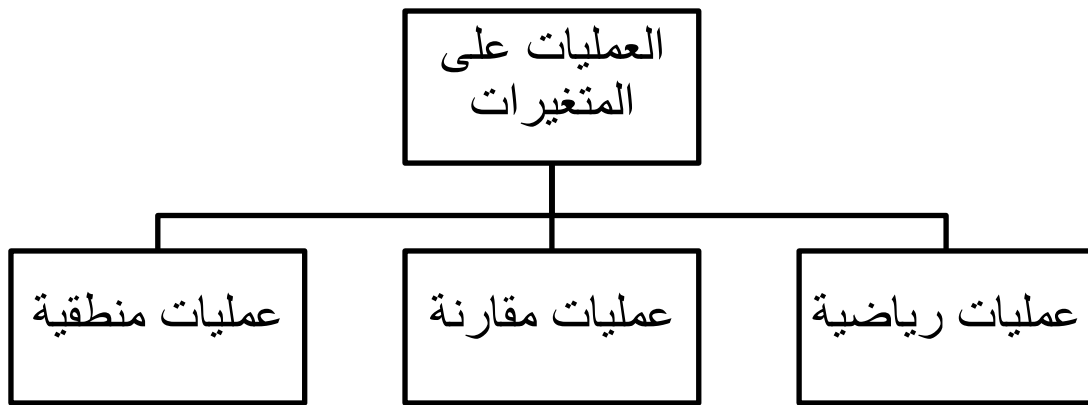
*نلاحظ هنا أننا لم نستخدم اقواس التنصيص لاننا نريد من صفحة الويب أن تظهر قيمة و ليس نص .

ملاحظات على أسماء المتغيرات :

- لا يمكن أن يبدأ اسم المتغير برقم و لكن يبدأ بحرف أو علامة (_) .
- اسم المتغير يتأثر بحالة الأحرف فمثلا mah يختلف عن MAH.
- لا يمكن أن تحتوي اسم المتغير على مسافة .
- لا يمكن استخدام الكلمات المستخدمة بواسطة اللغة (كلمات محجوزة) كأسماء متغيرات مثل if ,for ,while .

العمليات على المتغيرات :

تنقسم العمليات على المتغيرات الى ثلاثة أنواع وهي:



العمليات الرياضية:

بداية نحتاج أن نتعرف على كيفية اسناد قيمة لمتغير وذلك بوضع علامة = بعد اسم المتغير ثم وضع القيمة التي يراد اسنادها بعد ذلك كالتالي:

mah=15;

بهذا تكون قيمة المتغير mah تساوي 15.

و يمكن اسناد قيمة متغير الى متغير اخر كالتالي :

mah=15;

newvar=mah;

بهذا تكون قيمة المتغير الجديد newvar هي mah والذي يخزن القيمة 15 .

لا شك أن الجميع يعرف العمليات الحسابية الأساسية الجمع و الطرح و الضرب و القسمة و استخدام هذه العمليات في البرمجة مشابه الى حد كبير لاستخدامها في الواقع .

لجمع نستخدم علامة + و للطرح نستخدم - و للضرب * و للقسمة / .

مثال لاستخدام العمليات الرياضية :

```
mah=15;
Addvar=mah+20; // قيمة المتغير 35
Subvar=mah-5; // قيمة المتغير 10
Mulvar=mah*2; // قيمة المتغير 30
Div_var=mah/3; // قيمة المتغير 5
```

حتى الان لا يوجد أي اختلاف بين العمليات في البرمجة و في الواقع .

أيضا يمكن استخدام متغير كقيمة للجمع بدلا من القيم الثابتة .

```
var1=30;
vVar2=20;
Addvar=var1+var2; // قيمة المتغير 50
```

و اذا اردنا القيام بعملية على متغير موجود نستطيع أن نقوم بها كالتالي:

```
Mah=20;
Mah=Mah+5; // ستكون القيمة النهائية للمتغير Mah = 25
```

و ترجمة السطر البرمجي السابق لغويا هي اصف 5 للمتغير Mah و الذي قيمته 20.

أو للاختصار يمكن استخدام الطرق في الأسفل للحصول على نفس نتائج العمليات.

```
Mah=20;
Mah+=5; // تعني اضافة 5 الى القيمة الحالية في المتغير .
Mah-=5; // تعني طرح 5 من القيمة الحالية للمتغير .
Mah*=5; // تعني ضرب قيمة المتغير الحالية في 5 .
Mah/=5; // تعني قسمة قيمة المتغير الحالية على 5 .
```

لاضافة أو طرح 1 من قيمة المتغير يمكن ان تكتب بالطرق التالية :

Mah=15;

Mah=mah+1;

أضفنا 1 ال قيمة المتغير بما يعني أن قيمته اصبحت 16.

Mah+=1;

أضفنا للقيمة السابقة ١ و بالتالي اصبحت قيمة المتغير 17 .

Mah++;

اضفنا للقيمة السابقة 1 و بالتالي أصبحت قيمة المتغير 18.

نرى أن هناك نفس العملية مثل اضافة 1 يمكن أن تكتب بطرق متعددة.

نفس العمليات السابقة يمكن أن تطبق على عمليات الطرح و لكن بدلا من الشارة + سنستخدم الاشارة - .

*نتذكر ان استخدام ++أو - - يعني اضافة أو طرح العدد 1 من قيمة المتغير وليس هناك عمليات مماثلة يمكن استخدامها مع باقي الأعداد.

هناك عملية رياضية أخرى قد لا يكون تعاملت معها مسبقا وهي باقي القسمة (%) لاحظ أن العلامة هنا لا تعني النسبة المئوية و انما باقي القسمة .

mah=17;

Modu=mah%4; // ناتج العملية سيكون 1 لانه باقي قسمة 17 على 4 .

الأولوية في العمليات:

١- فك الأقواس.

٢- الأسس و الجذور .

٣- الضرب و القسمة .

٤- الجمع و الطرح.

إذا كانت هناك أكثر من عملية لها نفس الأولوية فان الحل يبدأ من الجهة اليسرى لليمنى.

مثال

نتاج العملية هو ٧ لأننا نقوم بعملية الضرب ثم الجمع حسب قانون ترتيب الأولويات .
 $Mah=3+2*2; //$
 نتاج العملية هو ١٧ .
 $R=2*1+3*5; //$

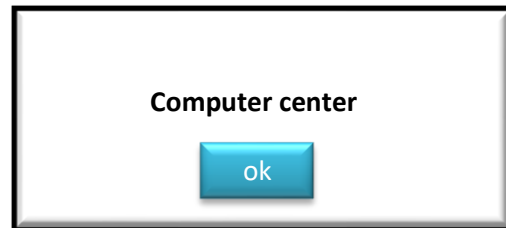
العمليات على المتغيرات النصية :

يمكنك فقط جمع المتغيرات النصية .

مثال:

```
<html>
<head>
</head>
<body>
<script language="javascript">
    Var1 = "computer ";
    Var2 = "center";
    Addvar= Var1+ Var2 ;
    alert(Addvar);
</script>
</body>
</html>
```

نتاج العملية سيظهر صفحة ويب فيها انذار كالتالي:

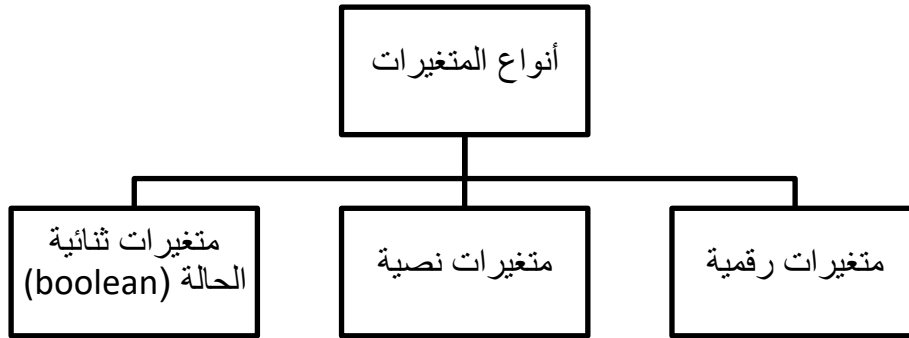


نلاحظ أنه تم جمع الكلمتين ووضع مسافة بينهما نتيجة جمع المتغير الأول بمساحة فارغة و من ثم جمع هذه المسافة بالكلمة الثانية .

*المساحة الفارغة يتعامل معها البرنامج على أنها أمر برمجي أيضا.

المقارنة بين المتغيرات :

أنواع المتغيرات كما تطرقنا لها سابقا هي نوعين :



المتغيرات ثنائية الحالة (Boolean) هي متغيرات تحتوي اما على صح true او خطأ false .

```
Mah=true;
```

```
R=false;
```

علاقات المقارنة :

احتمالات العلاقة بين القيم تتلخص في الآتي:

Var1 > var2	أكبر من
Var1 < var2	أصغر من
Var1 == var2	يساوي
Var1 >= var2	أكبر من أو يساوي
Var1 <= var2	أصغر من أو يساوي
Var1 != var2	لا يساوي

معنى المتغيرات ثنائية الحالة (Boolean) أنها تختبر هل العلاقة صحيحة أو خاطئة أي مثلا في السطر الأول `var1 > var2` سيختبر النص البرمجي هل `var1` فعلا أكبر من `var2` أو لا و يخرج نتيجة بصح أو خطأ عند طلبها.

مثال ١:

Var1=30;	نتائج Mah سيكون true أو
Var2=20;	صح لان var1 أكبر من var2
Mah=(var1>var2);	بالفعل .

مثال ٢:

Mah=(30>=30); //true
Mah=(30!=29); //true
Mah=(15==14); // false

العلاقات المنطقية :

العلامات التي تستخدم لإنشاء علاقات منطقية هي العلامة (&&) تعني (و) و العلامة (||) تعني (أو).

تستخدم العلاقات المنطقية مع علاقات المقارنة لعمل مقارنات أكثر تعقيدا من السابقة.
مثال:

Mah = ((50>30)&&(45==45))

بمعنى mah يساوي اما صح أو خطأ بناء على صحة عبارتي المقارنة المذكورة بين الأقواس .

المقارنة الأولى 50>30 ناتجها صح.

المقارنة الثانية 45==45 ناتجها صح.

و عليه فان ناتج صح و صح هو صح.

الجدول في الأسفل يختصر نتائج العلاقات المنطقية لـ && (و) :

العلاقة الأولى	العلاقة الثانية	&&	الناتج
true	true	True && true	true
true	false	True && false	false
false	true	False && true	false
false	false	False && false	false

الجدول في الأسفل يختصر نتائج العلاقات المنطقية لـ || (أو):

العلاقة الأولى	العلاقة الثانية	&&	الناتج
true	true	True && true	true
true	false	True && false	true
false	true	False && true	true
false	false	False && false	false

نفي العلاقات المنطقية (!):

```
Mah = !(24<28)
```

العلامة (!) تعني لا او ليس.
 العبارة تعني mah لا يساوي ناتج المقارنة.
 ناتج 24<28 هو صح.
 ليس (!) صح = خطأ.
 اذا الناتج النهائي لـ mah هو خطأ.

أمثلة على العلاقات المنطقية:

```
a1 = true && true // t && t returns true
a2 = true && false // t && f returns false
a3 = false && true // f && t returns false
a4 = false && (3 == 4) // f && f returns false

o1 = true || true // t || t returns true
o2 = false || true // f || t returns true
o3 = true || false // t || f returns true
o4 = false || (3 == 4) // f || f returns false
```

```
Var1="computer center";
Var2="computer center";
Check_var=(var1==var2) //true
Check_var=(var1 !=var2) //false
```

أسئلة و تمرينات

١- ماهي الملاحظات التي يجب مراعاتها عند تسمية المتغيرات

٢- ماهي قيمة المتغير mah بعد الانتهاء من تنفيذ كل من العمليات التالية :

a. mah=7;

mah+5;

mah++

b. mah=((30>30)&&(15==15))

c. mah=((20==20) || (25<10))

d. mah=(4+3)*2

e. mah=4+(3*2)

الفصل الثالث

التحكم في سير البرنامج

العناوين:

- الجمل الشرطية
- حلقات التكرار

الجمل الشرطية

الجمل الشرطية IF :

الجمل الشرطية if (إذا) و if-else (إذا - غيره) هي من أهم القواعد البرمجية لأنها تجعل البرنامج أكثر ذكاء بحيث تصبح له القدرة على تغيير الناتج وفقا لعوامل كثيرة يحددها المبرمج وهي ما تعني تفاعل أكثر مع المستخدم و تستخدم مع الجمل الشرطية علاقات المقارنة و العلاقات المنطقية السابق ذكرها.

الصورة الأساسية لعبارة if:

```

If( جملة الشرط )
{
    بعض العبارات المراد تنفيذها
}

```

جملة الشرط تكون علاقة مقارنة أو علاقة منطقية ، و ان صحت هذه العلاقة فان البرنامج سيبدأ في تنفيذ الاجراءات الملحقة مع الجملة و ان لم تصح فسيهمل البرنامج هذا الاجراءات .

مثال ١:

```

if (a > 0)
{
    result = 'positive' ;
}

```

مثال ٢:

```

if (x > 10)
{
    alert(x) ;
}

```


الصورة الأساسية للعبارة if – else :

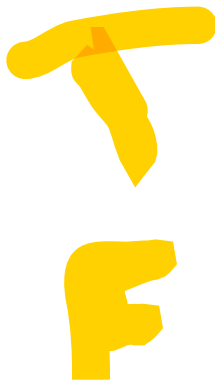
```

If(          )
{
    بعض العبارات المراد تنفيذها
}
else
{
    عبارة أخرى
}
    
```

مثال ١ :

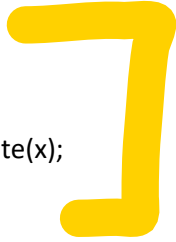
```

If(5>10 )
{
    alert("true");
}
else
{
    alert("false");
}
    
```



مثال ٢:

```
if (x==5)
{
  ++x ;
  document.write(x);
}
else
{
  document.write(x);
}
```



طبقي المثال السابق في مستند html و ادرسي مخرجه.

الجمل الشرطية

الجمل الشرطية Switch:

الجمل الشرطية switch تمكنني من وضع قيم مختلفة لنفس المتغير حيث يختار البرنامج أو صفحة الويب القيمة المناسبة للمتغير بناء على شرط.

الصيغة الأساسية للعبارة الشرطية switch:

```
switch ( ) {  
    case value1:  
        // العبارة المطلوب اداؤها عند تحقيق الشرط لهذه القيمة  
        break;  
    case value2:  
        // العبارة المطلوب اداؤها عند تحقيق الشرط لهذه القيمة  
        break;  
    ...  
    case valueN:  
        // العبارة المطلوب اداؤها عند تحقيق الشرط لهذه القيمة  
        break;  
    default:  
        // العبارة المطلوب اداؤها عند تحقيق الشرط لهذه القيمة  
        break;  
}
```

مثال ٢:

```
<html>
<head>
</head>

<body>
<script language="javascript">

var day = 4;
var dayString;

switch (day) {
case 1:

dayString = "Sunday";
break;

case 2:
dayString = "Monday";
break;

case 3:
dayString = "Tuesday";
break;

case 4:
dayString = "Wednesday";
break;

case 5:
dayString = "Thursday";
break;

case 6:
dayString = "Friday";
break;

case 7:
dayString = "Saturday";
break;

default:
dayString = "Invalid day";
}

document.write("The day is: " + dayString);

</script>
</body>
</html>
```

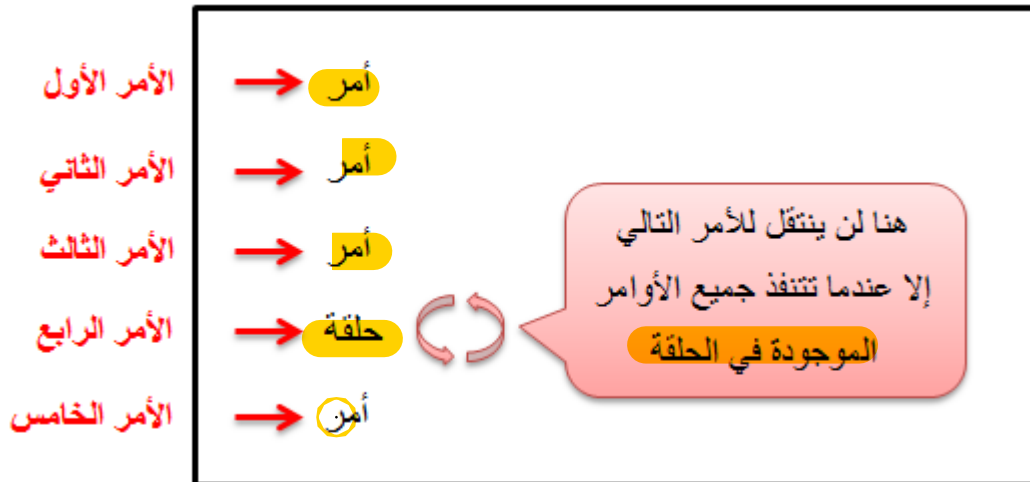
طبقي النص البرمجي السابق في مستند html و ادرسي النتيجة .

حلقات التكرار

ستخدم الحلقات (**Loops**) بهدف تكرار نفس الكود عدة مرات. إذاً أي كود نريده أن يتنفذ عدة مرات، نقوم بكتابته داخل حلقة فتقوم هي بإعادة تنفيذ الكود قدر ما شئنا ضمن شروط معينة نقوم نحن بتحديدتها.

طريقة تنفيذ الأوامر و الحلقات في الذاكرة:

الأوامر في العادة تتنفذ بتسلسل وراء بعضها، و لكن الحلقات تجعل سهم تنفيذ الأوامر يقف عندها فيقوم بتنفيذ الأوامر التي بداخلها عدة مرات، و بعد أن يخرج من الحلقة يعود و يكمل تنفيذ باقي الأوامر الموجودة بعدها كالتالي.



عندما تنفذ الحلقة فإن الأوامر الموجودة فيها تنفذ بشكل منفصل عن باقي الأوامر الموجودة في البرنامج، أي يمكنك اعتبار أن جميع أوامر الحلقة توضع في مكان خاص في الذاكرة، هذا المكان يسمى **scope**.

بعد أن تنفذ جميع أوامر الحلقة في هذا الـ **scope**، يتم مسح الـ **scope** كلياً من الذاكرة و هذه العملية تسمى **Destroy**.

دواعي الإستخدام	إسم الحلقة
تستخدم الحلقة for في حال كان عدد المرات التي سيعاد فيها تنفيذ الكود معروفاً.	For Loop
يفضل إستخدام الحلقة while في حال كان عدد المرات التي سيعاد فيها تنفيذ الكود غير معروف.	While Loop
يفضل إستخدام الحلقة while do في حال كان عدد المرات التي سيعاد فيها تنفيذ الكود غير معروف و بنفس الوقت يجب أن يتنفذ مرة واحدة على الأقل.	Do While Loop

الصيغة الأساسية للـ for loop:

```
For( مقدرا الزيادة، حدود العد ، بداية العد )
{
    بعض الاجراءات للتنفيذ
}
```

مثال ١:

```
for (i=0;i< 9; i++)
{
    document.write(i);
}
```

طبقي النص البرمجي في صفحة html وناقشني المخرج.

الصيغة الأساسية للـ while loop:

```
while( الشرط )
{
    بعض الاجراءات للتنفيذ
}
```

مثال ١:

```
x= 2;
b=3;
v=x+b;
v++
while (v==6)
{
    Document.write("v=6");
}
```

طبقي النص البرمجي في صفحة html وناقشي المخرج.

الصيغة الأساسية لـ do – while loop :

```
do
{
    بعض الاجراءات للتنفيذ
}
While( الشرط )
```

مثال ١ :

```
X=0
do
{
    Document .write(x);
}
While( x=1 )
```

طبعي النص البرمجي في صفحة html وناقشي المخرج و الفرق بين الـ while و do-while مع معلمتك و زميلاتك.

أسئلة و تمرينات

١- اكتبى برنامج يقوم بتعريف قيمة متغير ثم يقوم باختبار هذه القيمة هل هي سالبة أم موجبة أم أنها تساوي الصفر و يظهر رسالة باسم القيمة .

٢- اكتبى برنامج يقوم بطباعة الأعداد الفردية من ١ الى ٢٠ من الأكبر الى الأصغر.

٣- باستخدام جملة while اكتبى برنامج يقوم بطباعة اسمك ١٠ مرات على الشاشة .

الفصل الرابع

الدوال و الأحداث

العناوين:

- الدوال و استخداماتها
- الأحداث الأساسية في الجافا سكريبت

الدوال

دالة: تعني **Method** او **Function** في اللغة الإنجليزية، و هي عبارة عن مجموعة أوامر مجمعة في مكان واحد و تنفذ عندما نقوم باستدعائها.
كما أن جافا تحتوي على مجموعة كبيرة جداً من الدوال الجاهزة التي يمكنك إستعمالها مباشرةً و التي سنتعرف عليها في دروس لاحقة.

- إسم الدالة.
- قائمة بارامترات الدالة، محصورة بين قوسين ويفصل بينها بفواصل.
- تعليمات الجافا سكربت التي تُعرف الدالة، داخل الأقواس المتعرجة {}.

الصورة الأساسية للدوال:

```
function (معامل ) اسم الدالة
{
عمليات للتنفيذ
}
```

مثال ١:

```
function square(number)
{
return number * number;
}
```

الدالة بالاسم square لعدد معين ستخرج ناتج ضرب العدد في نفسه.

مثال ٢:

```
function f (x) {
x=10;
x++;
return x;
}
```

الدالة سينتج عنها قيمة if بعد زيادة واحد.

مثال ٣:

```
function m (s,d) {
s=3;
d=2;
if(s>d)
{
Document.write("s is greater than d");
}
Else
{
Document.write("s is not the greater that d");
}
}
```

استدعاء الدوال

إستدعاء الدوال

تعريف الدالة يعني ببساطة إعطاء إسم لها وتحديد ما ستقوم به عندما يتم إستدعائها. أما إستدعاء الدالة فيقوم فعليا بتنفيذ الإجراءات مع البرامترات المحددة. على سبيل المثال، إذا قمت بتعريف الدالة square، فستقوم بإستدعائها كما يلي :

استدعاء الدالة في مثال ١ :

```
square(5);
```

استدعاء الدالة في مثال ٢ :

```
f (5);
```

استدعاء الدالة في مثال ٣ :

```
m(4,7);
```

طبقي أمثلة الدوال في مستندات html و ناقشي مخرجاتها.

الأحداث

الأحداث Events هي سمة مهمة فبدونها لا تستطيع أن تجعل صفحات الويب تفاعلية. حيث أنه من خلال الأحداث Events يمكن تنفيذ دالة معينة عندما يحدث شيء مثل أن يكتمل تحميل صفحة الويب أو شيء يفعله المستخدم مثل الضغط على عنصر في الصفحة.

الحدث	الوصف
load	يحدث عندما يكتمل تحميل صفحة أو صورة
click	يحدث عندما يضغط المستخدم على لك أو صورة أو أي عنصر من عناصر الصفحة
mouseover	يحدث عند دخول الماوس فوق عنصر
mouseout	عند إبتعاد الماوس من فوق عنصر
keydown	عند الضغط على أي زر في لوحة المفاتيح

مثال ١:

```
<html>
  <input type="button" value="اضغط هنا" onClick="alert('تم الضغط')">
</html>
```

طبقي النص البرمجي و ناقشي المخرجات.

الفصل الخامس

النماذج

العناوين:

- نماذج الـ html اعدادها و تنسيقها.

النماذج

مفهوم النموذج

النموذج (Form) في العادة عبارة عن مجموعة عناصر توضع في الصفحة بهدف جعل المستخدم يدخل بيانات فيها. من الأمثلة الشائعة التي نتعامل فيها مع نماذج، هي صفحات تسجيل الدخول، صفحات إنشاء حساب جديد، إضافة تعليق، إضافة منشور إلخ..

النموذج بحد ذاته ليس شيئاً يتم عرضه في الصفحة بل هو بمثابة حاوية للوسوم التي نضعها فيه. الفكرة الأساسية من النموذج هي تجميع المعلومات التي يدخلها المستخدم في العناصر الموضوعة فيه و إرسالها دفعة واحدة للمكان الذي سنحفظها فيه.

في هذا الدرس سنتعرف على الخصائص التي يمكن وضعها في النموذج و دور كل منها.

النماذج في الـ html تستخدم لنقل معلومات مطلوبة لخدم .

نماذج الـ HTML يمكن أن تحتوي عناصر ادخال مختلفة كمربع ادخال نص، أزرار ،قائمة تحديد، مساحة نصية

الصورة الأساسية لنماذج html:

```
<html>
<head>
</head>
<body>
<form name ="form">
محتوى النموذج
</form>
</body>
</html>
```

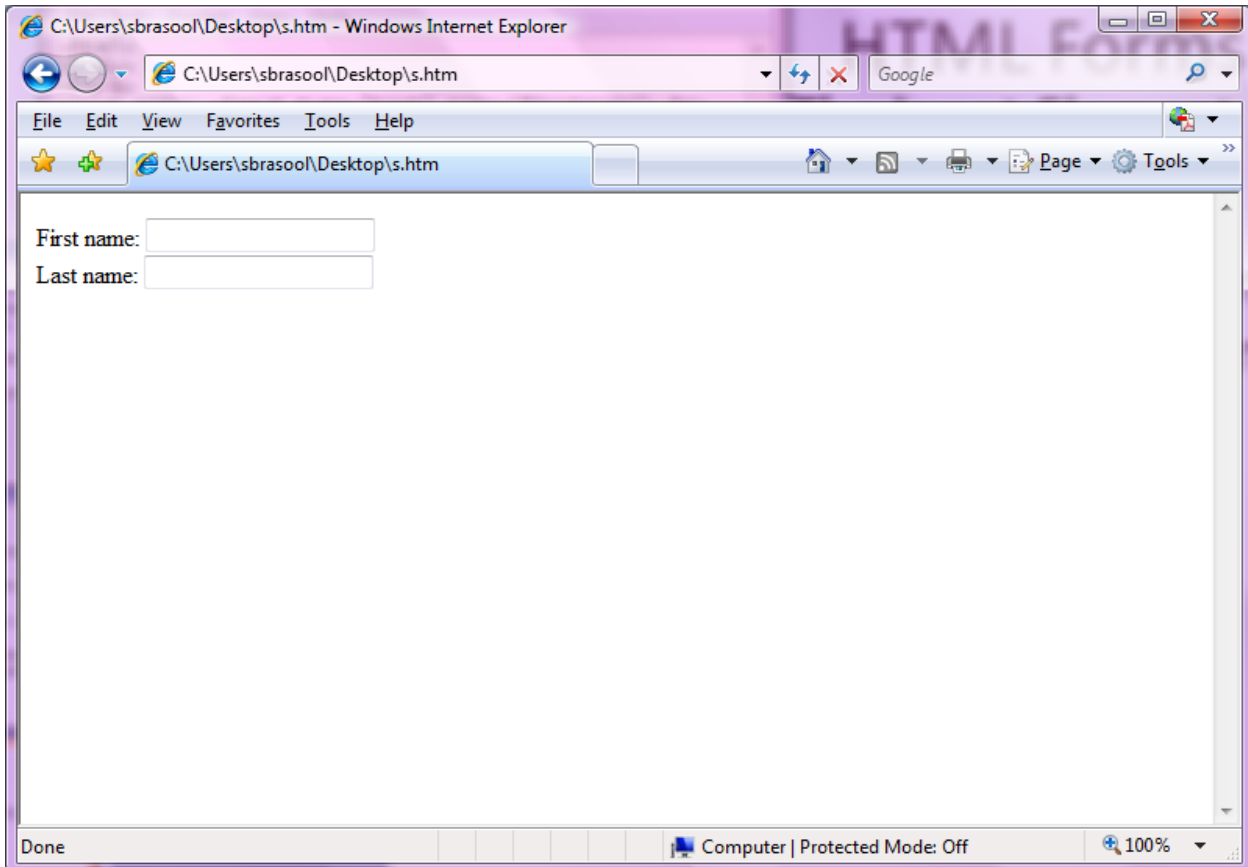

جدول معرفات النماذج:

tag	الوصف
<form>	يعرف المسند كنموذج
<input>	يعرف خاصية التحكم بمدخل
<textarea>	يعرف خاصية التحكم بمدخل متعدد الحروف
<label>	يمكن من اعطاء تصنيف لعنصر مدخل
<fieldset>	يجمع العناصر المرتبطة في النموذج
<legend>	يعطي شرح عنصر في النموذج
<select>	يمكن من استخدام قائمة منسدلة
<optgroup>	يحدد مجموعة خيارات مترابطة في قائمة منسدلة
<option>	يحدد خيار في قائمة منسدلة
<button>	يعرف زر قابل للضغط
<datalist>	يحدد مجموعة من الخيارات المعدة مسبقا للتحكم بالمدخلات
<output>	يعطي مخرج أو ناتج عملية

مثال ١ استخدام مربع نص في نموذج:

```
<html>
<head>
<head/>
<body>
<"form name="form">
<"br>First name: <input type="text" name="firstname">
<"br>Last name: <input type="text" name="lastname">
</form>
</body>
</html>
```

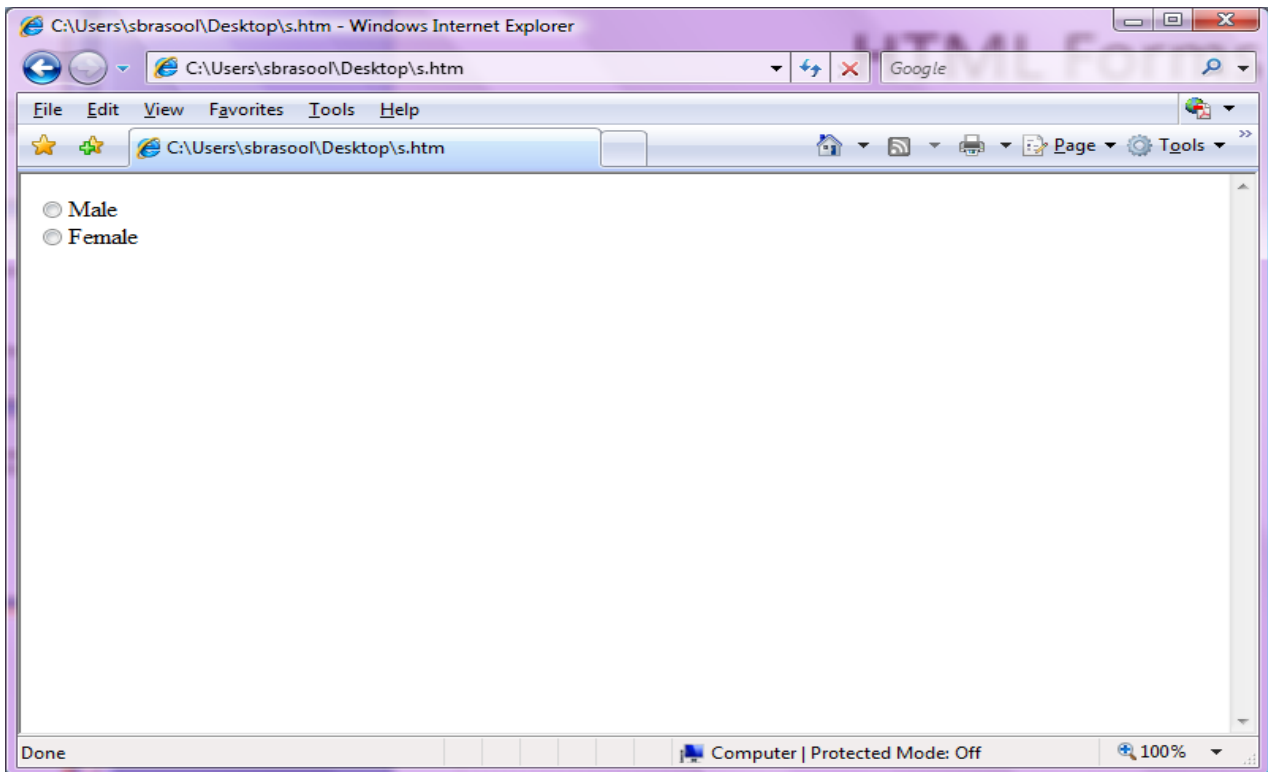
مخرجات مستند html السابق :



مثال ٢ استخدام زر راديو (اختيار واحد من متعدد) في نموذج:

```
<html>
<head>
<head/>
<body>
<form name="form">
<br> Sex:
<br><input type="radio" name="sex" value="male">Male
<br><input type="radio" name="sex" value="female">Female
</form>
</body>
```

مخرج مستند html السابق :



*ملاحظة:
 يستخدم للنزول لسطر جديد.

مثال ٣ استخدام check list (اختيار متعدد) في النماذج:

```
<html>

<head>

<head/>

<body>

<"form name="form">

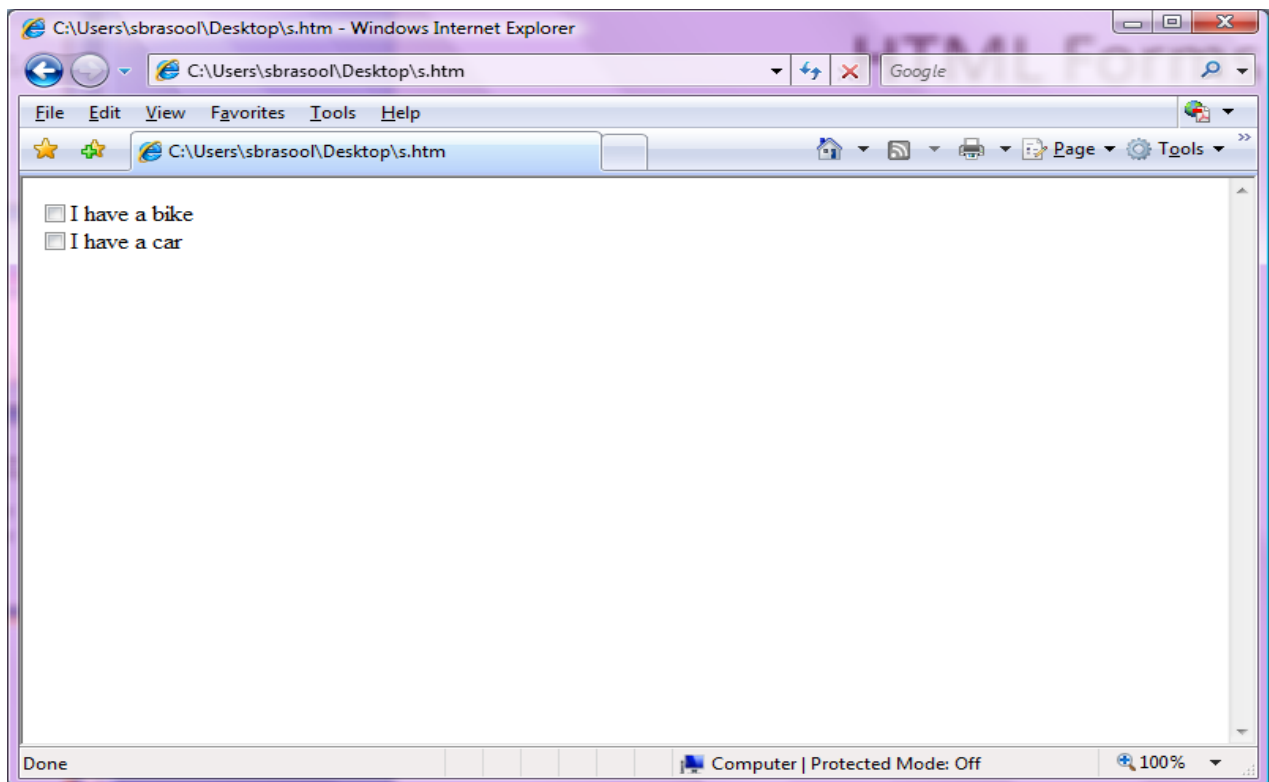
<input type="checkbox" name="vehicle" value="Bike">I have a bike <br>
<input type="checkbox" name="vehicle" value="Car">I have a car

</form>

</body>

</html>
```

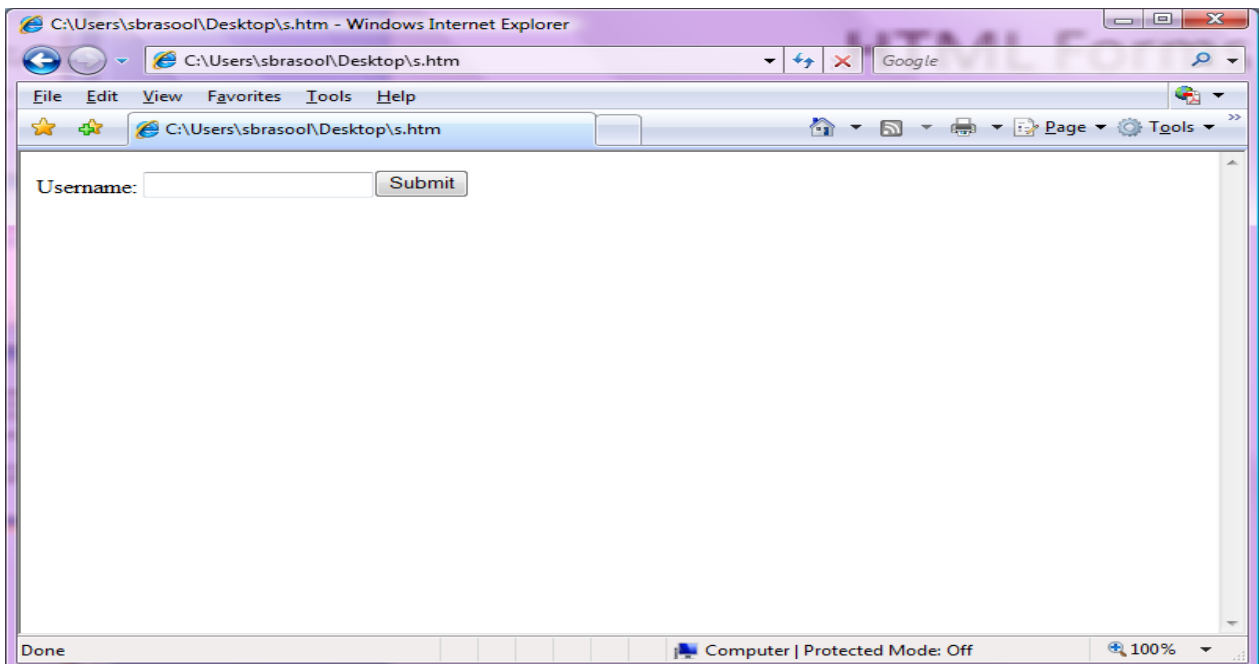
مخرج مستند الـhtml السابق:



مثال ٤ : إضافة زر ارسال لمستند html:

```
<html>
<head>
<head/>
<body>
<"form name="form">
Username: <input type="text" name="user">
<input type="submit" value="Submit"></form>
</body>
</html>
```

مخرج مستند الـhtml السابق:



أسئلة و تمرينات

١- طبق ي النص البرمجي في الأسفل و ناقشي مخرجاته:

```
<html>
  <head><b> <i>form
</head/>
<body>
  <"form name="form">
  <"br>First name: <input type="text" name="firstname">
  <"br>Last name: <input type="text" name="lastname">
  <br> Sex:
  <br><input type="radio" name="sex" value="male">Male
  <br><input type="radio" name="sex" value="female">Female
  <br> hopyy:<br>
  <input type="checkbox" name="hopyy" value="drawing">drawing<br>
  <input type="checkbox" name="hopyy" value="reading">reading<br>
  <input type="checkbox" nmae="hopyy" value="swiming">swiming<br>
  <input type="checkbox" name="hopyy" value="writing">writing<br>
  <br> <input type="button" name="b" value="hello"
  onclick="document.form.b.value='welcome'";
  ondblclick="document.form.b.value='bey'";>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

٢- اضيفي خانة جديدة في النموذج ليتحدث مرسل النموذج عن نفسه بشكل مختصر.

والله ولي التوفيق
انتهى