

برمجة صفحات الويب جهة العميل

أمثلة واكواد مشروحة في لغات تصميم الويب (html5, css3, JavaScript)

Client-side web-pages programming (html5, css3, JavaScript)

د. فهد آل قاسم

أستاذ مساعد في الحاسبات والمعلومات - جامعة العلوم والتكنولوجيا اليمنية

fhdahmd18@yahoo.com

مقدمة

الحمد لله وحده والصلاة على من لا نبي بعده..

يسرنني كثيرا كثيرا العودة الى عالم الكتب الالكترونية، الذي تتبادل فيه خدماتنا المجانية للمعلوماتية باللغة العربية، باعتبارنا هواة أو محترفي أو محبي أو متخصصي عالم الحاسبات والمعلومات الكبير والواسع، صحيح ان الساحة قد تغيرت بشكل كبير، حتى اسم التخصص الذي كان علوم الحاسوب (علوم الحاسب الآلي) او الرياضيات والحاسوب، اصبحت له تنوعات كثيرة كالانفورماتيك ونظم المعلومات وتكنولوجيا المعلومات وهندسة البرمجيات وتقنية المعلومات وووو.. وغيرها من المسميات، التي يجمعها في بداية ومنتصف ونهاية الأمر: كتابة كود برمجي غالبا بالحروف اللاتينية، ويعمل على جهاز حاسوبي أو محوسب، ايا كان حجم شاشته إلا انه يظل حاسوبا، بمعالج وذاكرة معالجة وذاكرة تخزين، كما تعلمنا من دراستنا لنموذج نيومان، العالم العظيم الذي سطر الف باء الحوسبة.

أقدم اليوم كتبيا مبسطا في لغات تصميم الويب جهة العميل، تلك الصفحات الساكنة، التي يمكن ان تتحدث وتقول الكثير حتى بدون نت!!، والتي ايضا، لا غنى لها مطلقا عن البرمجة جهة المخدم، حيث يتم تبادل اكثر الحوارات الرقمية اهمية في عالم اليوم، باستخدام بروتوكولات كثيرة اهمها http بروتوكول نقل النص التشعبي.

هذا الكتيب لا يشرح مفاهيم ولا مبادئ اساسية بقدر ما يقدم امثلة، يفترض ان قارئة اما شخص ملم باساسيات برمجة الويب، او أن لديه القدرة على البحث عن المعلومات الاساسية المتاحة دائما. وكخدمة لقارئ المعلوماتية باللغة العربية، قمت بترجمة عدد من الامثلة باللغات الثلاث، بعض هذه الامثلة منقولة حرفيا من الكتاب الاساسي في المراجع، وبعضها عملت عليه وطورته بشكل مثال مبسط بنفسي. في النهاية يريد منك الكتاب هذا ان تقوم كمبرمج حاسوبي بقراءة وفهم ما استطعت فهمه ثم تنطلق الى ادواتك البرمجية وتبدأ في تطوير اكوادك الخاصة .

اقدم هذا الجهد المجاني المبسط ، عاشر اخوته، جنبا الى جنب مع تسعة كتيبات من سنين سابقا، كنت قد رفعتها الى الموقع الرائد :كتب kutub.info، هذه المنصة المتميزة التي قدمت لنا هذا الاستثمار الذكي، املا ان يكون القائمون عليه قد استفادوا منه ايضا، بقدر ما استفاد منه الاف القراء العرب، سائلا من الله التوفيق، وهوالمستعان في كل وقت وحين.

المؤلف: د. فهد آل قاسم - اليمن - إب

- لو كان هناك من إهداء لهذا الكتيب فهو لاساتذتي ولزملائي في التخصص فهم النجوم السامقة التي تملأ سماء قلبي ، واني لاتمنى سرد اسمائهم واحدا واحدا مع الاهداء الذي يليق به، ولكني على ثقة عالية بأنهم يعرفون انفسهم ويعرفون مكانتهم في قلبي، فلهم خالص الاهداء وجزيل الاعتذار.

المحتوى

1. تعريفات مهمة حول لغتي تصميم الصفحات الساكنة في الويب.
2. مدخل بسيط وسريع للـ html5 بمثال مشروح.
3. مثال عملي مشروح في لغة ترميز النص التشعبي html5 مع لغة صفحات التنسيق المتتالي css3.
4. مثال حول النماذج Forms وعناصرها في لغة ترميز النص التشعبي html الاصدار الخامس
5. لغة الجافا سكريبت : مقدمة.
6. جافا سكريبت: المجموعة الاولى من الامثلة.
7. جافا سكريبت: المجموعة الثانية من الامثلة.

1. بعض التعريفات المهمة حول لغتي تصميم وتنسيق الصفحات الساكنة في الويب

- **HTML file**: هو ملف بامتداد *.html مكتوب بلغة html وهي اختصار للغة ترميزالنص التشعبي hypertext markup language وتستخدم لتصميم ملفات نصية تنتقل عبر بروتوكول http يتم عرضها على متصفحات الانترنت internet browsers مثل المتصفح كروم او فايرفوكس موزيلا او سفاري او غيرها ، والاصدار الخامس هو اخر اصدار معتمد من منظمة w3c للغة ال html، وتتكون الصفحة من رأس head وجسم body نضع في الرأس المعلومات الرئيسية للصفحة وفي الجسم نضع العناصر والنصوص التي يتم عرضها في الصفحة.
- **CSS file**: هو ملف امتداده *.css يستخدم لكتابة قيم لخصائص عناصر صفحة ال html هذه الخصائص تحدد تنسيق عناصر الصفحة من خطوط والوان وهوامش وخصائص اخرى متعلقة بعناصر الصفحة كالعرض والارتفاع والازاحة وشكل العرض وغيرها ... والكلمة CSS هي اختصار لجملة صفحات التنسيق المتتابع cascade style sheets ويتم الربط بين صفحة التنسيق وصفحة ال html باستخدام العنصر link الذي يتم فيه تحديد مسار صفحة التنسيق وانها صفحة تنسيق متتابع stylesheet.
- **Article element**: هو احد عناصر الاحتواء الجديدة في لغة html5 التي تسمى بالعناصر الدلالية semantic elements، ويستخدم بالوسم <article> كحاوية رئيسية للعناصر الموجودة في الصفحة وغالبا ما يكون لدينا عنصر واحد او اثنين على الاكثر منه، بحيث تكون باقي الحاويات (عناصر الاحتواء) الاخرى داخله.
- **Paragraph element**: هو عنصر الفقرة الواحدة ويعتبر اصغر عناصر الاحتواء في صفحة ال html ويستخدم بالوسم <p> ومثله مثل اي عنصر احتواء يبدأ دائما بسطر جديد،

ورغم كونه من اقدم العناصر المستخدمة إلا انه لاهميته لايزال مستخدما ومهما حتى الان.

- **section element**: هو احد عناصر الاحتواء الجديدة في لغة html5 التي تسمى بالعناصر الدلالية semantic elements، ويستخدم بالوسم <section> كحاوية فرعية ضمن الحاوية الرئيسية article وموقعه في العادة وسط الصفحة سواء كان افقيا او عموديا ، ويمكن استخدام عناصر كثيرة منه في الصفحة الواحدة.

- **header element**: هو احد عناصر الاحتواء الجديدة في لغة html5 التي تسمى بالعناصر الدلالية semantic elements، ويستخدم بالوسم <header> كحاوية ترويسة الصفحة حيث يستخدم كترويسة لجسم الصفحة ويمكن ان نستخدم منه اكثر من واحد اذا كانت الصفحة تحتاج اكثر من ترويسة، كأن تكون مثلا مكونة من صفحات كتاب له رأس وتذييل.

- **href attribute**: احد اهم الخصائص في صفحة ال html وتنطق (إتش رف) وهي اختصار للعبارة مرجع رابط تشعبي hyperlink reference وتستخدم كخاصية تابعة لأي من العنصرين ال anchor وال link ويتم اسناد مسار معين لها قد يكون مسار ملف مجاور للملف الحالي او في مجلد داخلي او خارجي او رابط url على الانترنت او عنوان بريد الكتروني، وفي ال anchor تستخدم ايضا كرابط وصول لعنصر ضمن الصفحة نفسها عبر استدعاء الخاصية id التابعة له باستخدام الرمز #.

- **title attribute**: خاصية تتبع اغلب العنصر والهدف منها اسناد جملة نصية لها تظهر كبالون توضيحي حول العنصر وذلك عن تمرير المؤشر إلى مكان العنصر.

- **division element**: عنصر الاحتواء العام ، وكان يعتبر في الاصدار الرابع من ال html عنصر الاحتواء الرئيسية الذي يستخدم دائما، ولكن لانه كان يحتاج إلى خاصية معينة

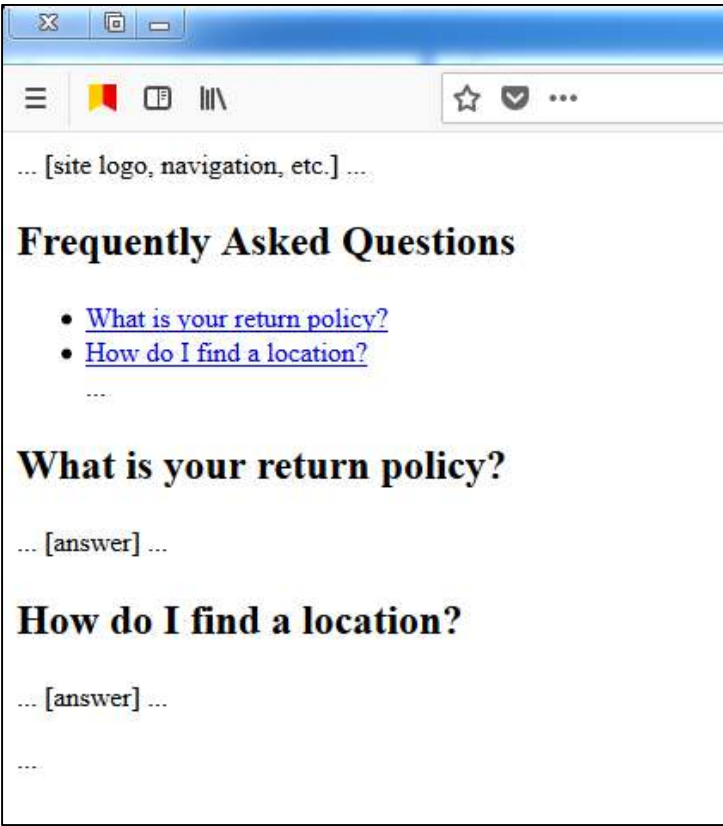
تحدد دلالة استخدامه كل مرة فقد تم إضافة عناصر دلالية في الاصدار الخامس من اللغة لكي تحل محله، ولكن مازال له استخدام واسع حتى الان.

- **footer element**: هو احد عناصر الاحتواء الجديدة في لغة html5 التي تسمى بالعناصر الدلالية semantic elements، ويستخدم بالوسم <footer> كحاوية للمحتوى اسفل الصفحة اي تذييل للصفحة وتكون مرادفة في العادة للترويسة header وتستخدم عادة لعرض عبارات التذييل المعتادة كحقوق الطبع او العنوان او خريطة الموقع او ما شابهها .

2. مدخل بسيط وسريع للـ html5 بمثال مشروح.

يقدم هذا القسم درسا مبسطا حول استخدام الوسوم الاساسية في لغة html5 الكود مع شرح في المثال التالي.

- سنقوم بتصميم الصفحة التي ستظهر بالشكل التالي:



بالنقر بالزر الايمن داخل الصفحة واختيار امر عرض المصدر سيظهر لنا الكود التالي، وقد قمنا بتوضيح اهم مكوناته:

السطر الاول امر يخبر المتصفح ان الكود التالي هو كود

```
<!DOCTYPE html>
```

هذا السطر فيه وسم بداية الصفحة والذي ينغلق في نهايتها اضافة الى خاصية تشير الى ان المحتوى سيكون باللغة الانجليزية

```
<html lang="en">
```

السطر القادم يحدد الجزء الخاص برأس الصفحة بشكل عام

<head>

السطر التالي يشير الى نوع الترميز الذي كتبت به الصفحة

<meta charset="UTF-8" />

السطر التالي يوضح عنوان الصفحة في المستعرض

<title>Example of page with two header elements</title>

الامر التالي هو امر اغلاق وسم رأس الصفحة

</head>

السطر التالي فيه امر بداية جسم الصفحة

<body>

قد تتكون من عدد من الصفحات HTML هناك رأس اخر خاص داخل الصفحة حيث ان صفحة ال لكل واحدة رأس خاص بها، ورأس الصفحة السابق اجباري على عكس هذا فيكون يتم توضيح وظيفة هذا الرأس. role اختياري... والخاصية

<header role="banner">

هنا يتم اضافة المحتوى النصي او الصور او غيرها.. حسب المصمم..

السطر التالي هو اغلاق الرأس الخاص.

</header>

الوسم التالي يحدد بداية الفقرات الاساسية للصفحة وكأننا نكتب مثلا مقالا حيث ان ارتيكل تعني مقال باللغة الانجليزية

<article>

ويمكن اضافة رأس خاص بكل مقال بالوسم التالي

<header>

الخاصية هنا تعرض ما بينها باعتباره عنوانا رئيسيا يربط عريض

<h1>Frequently Asked Questions</h1>

فيما يلي الوسم الذي يخبر المستعرض ان لدينا مجموعة كبيرة من الروابط

<nav>

... كما ان هناك اوسم unordered list اختصار ل ul الوسم التالي يحدد ان لدينا قائمة غير مرتبة و للقوائم المرتبة. ol اخر هو

بمعنى list item هي اختصار ل li الوسم التالي يحدد اول عنصر في القائمة غير المرتبة، والكلمة عنصر قائمة..

href الذي يحدد رابطا عبر الخاصية a ولاحظ ضمن القائمة يوجد الوسم

What is your return

policy?

نفس الامر مرة اخرى مع السطر التالي:

How do I find a

location?

ويفترض الاستمرار بالقوائم الى أي عدد يريده المصمم.

...

اغلاق وسم القائمة غير المرتبة

اغلق وسم الروابط

</nav>

اغلق وسم الرأس الخاص بالارتیکل

</header>

السطر التالي يوضح لنا تعليقا وهو رسالة بين المبرمجين ، وعلى المستعرض ان يتجاهل ما بين <!-- وحتى الوسم -->

<!-- the header links point to these -->

وهو خاصية مهمة لمعرفة هوية id ملاحظ انه تم البدء بارتیکل جديد قبل اغلاق الاول وايضا الخاصية الوسم فيما بعد عند البرمجة وتم استخدامها هنا في الرابط اعلاه مسبوقه برمز الهاش لكي يصل الرابط اليها.

<article id="answer1">

السطر التالي فيه عنوان رئيسي اصغر من السابق

<h2>What is your return policy?</h2>

السطر التالي يوضح فقرة كاملة واغلاقها

<p> ... [answer] ... </p>

والان يتم اغلاق وسم الارتیکل الاخير.

</article>

idفتح ارتیکل جديد لكي يصل اليه الرابط اعلاه عبر خاصية ال

<article id="answer2">

<h2>How do I find a location?</h2>

<p> ... [answer] ... </p>

</article>

...

</article> <!-- end parent article -->

وسم اغلاق جسم الصفحة

</body>

</html> واخير وسم اغلاق الصفحة كاملة.

3. مثال مطول مشروح في لغة ترميز النص التشعبي html5 مع لغة صفحات التنسيق المتتالي css3

المثال الأول : الصفحة الرئيسية

- في الشكلين الموضحين ادناه الصفحة الرئيسية والتي يمكن تلخيص ما نراه بالتالي:
 - عنوان الصفحة في راس المتصفح غير واضح تماما بسبب صغر الصورة مكتوب فيه اسم المقرر.
 - جسم الصفحة وهو الخلفية الظاهرة باللون الرمادي الغامق.
 - الحاوية الرئيسية للصفحة وهي بلون اخضر فسفوري.
 - ترويسة الصفحة وفيها عنوان الموقع وبالخلفية صورة ونص مكتوب بنمط عريض.
 - شريط جانبي عمودي فيه عناوين الصفحات المتفرعة من الصفحة الرئيسية (هناك من يختار ان يكون التفرع الى العناوين عبر شريط اعلا الصفحة غالبا ما يكون اسفل ترويسة الصفحة وبشكل افقي، والقرار يعتمد على تصميم الصفحات الفرعية وحجم المحتوى المطلوب عرضه ضمنها).
 - مجموعة مقاطع نصية مرتبة افقيا كل جزء فيه موضوع قصير او مدخل مختصر لموضوع في صفحة فرعية.
 - واخيرا هامش الصفحة وهو بنفس عرض ترويسة الصفحة.



الشكل التالي يوضح ترويسة الصفحة وقد تم تصميم صورة بنفس الحجم لكي تظهر بدون تكرار.



الشكل التالي يوضح الشريط الجانبي (شريط التنقل navigation bar):



فيما يلي سنقوم بشرح الكود المستخدم لتصميم الصفحة:

أولاً: كود الـ html5:

كما هو موضح في الشكل التالي الاسطر من 1 الى 19، وتتضمن كود رأس الصفحة في الاسطر التسعة الاولى وبعدها وسم جسم الصفحة في السطر الحادي عشر ثم وسم الحاوية الرئيسية لجسم الصفحة وهي هنا باستخدام عنصر article مع خاصية محددة لاستخدامها لاحقاً، ثم عنصر ترويسة الصفحة وفيه نصين كعناوين رئيسية من الدرجة الاولى والثانية.

وإذا تساءلت من اين اتت الصورة في خلفية الترويسة مع انها غير مذكورة في الكود ادناه والاجابة هي CSS.

```

1 <!doctype html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5 <meta charset = "utf-8" />
6
7 <link rel="stylesheet" href="css/mycss.css" />
8 <title> مقدمة في تصميم الويب </title>
9 </head>
10
11 <body>
12 <article class="main">
13
14 <header id="frstheadr">
15
16 <h1 > مقدمة في تصميم الويب </h1>
17 <h2> www.Intro2DesingWeb.com </h2>
18
19 </header>

```

نحتاج لكتابة السطر رقم واحد لكي نشير اننا نتعامل مع الاصدار الخامس من لغة html، والسطر الثاني فيه وسم الصفحة الاساسي اضافة الى خاصية تحدد لغة المحتوى وخاصة توضح اتجاه الكتاب من اليمين لليساار اختصار لـ right to left.

السطر الرابع يبدأ فيه جزء رأس الصفحة ويختلف عن ترويسة الصفحة ادناه في ان رأس الصفحة وسم اساسي وعام للصفحة ككل، ويظهر مرة واحدة فقط ، بينما ترويسة الصفحة عنصر اختياري ويمكن ان يوجد من اكثر من واحد.

السطر الخامس يحدد نمط الترميز للصفحة، بينما السطر السابع يحدد صفحة مسار ال CSS التي تتحكم بتنسيقات الصفحة الحالية، بعد ذلك وسم العنوان، ثم في السطر التاسع اغلاق لوسم راس الصفحة.

ابتداء من السطر ال 11 يبدأ جسم الصفحة وهو الجسم الظاهر للمستخدم، وضمنه حاوية اخرى هي حاوية عامة لعناصر الصفحة من نوع article ثم فتح واغلاق عنصر ترويسة الصفحة.

```

26 <div>
27 </div>
28 <nav>
29 <ul>
30 <li><a href="index.html" > الصفحة الرئيسية (المثال الاول) </a></li>
31 <li><a href="cvpage.html" > HTML5 </a></li>
32 <li><a href="page-two.html" > CCS3 </a></li>
33 <li><a href="index.html" > JAVASCRIPT </a></li>
34 <li><a href="second_eg.html" > المثال الثاني </a></li>
35 <li><a href="third_eg.html" > المثال الثالث</a></li>
36 <li><a href="fourth_eg.html" > المثال الرابع</a></li>
37 </ul>
38 </nav>
39
40
41 <section>
42 <h2> لغة HTML5 ترميز النص التشعبي </h2>
43 
44 <ul>
45 <li> المستوى الثاني تقنية معلومات
46 html5? ما هي لغة
47 </li>
48
49
50 </ul>
51 .....
52 .....
53 <b>
54 <a href="cvpage.html"> ... يتبع .... </a>
55 </b>
56 </section>

```

الاسطر التالية مستمرة بعرض كود الصفحة وتبدأ بالسطين 26 و 27 يعرضان عنصرا قديما من لغة html4 يمكن بالطبع استخدامه حيث انه كان اهم العناصر المستخدمة وقتها، ولكننا في الاصدار الخامس من اللغة فضلنا استخدام العناصر الدلالية لانها صممت خصيصا لمواقع محددة في الصفحة كما يدل عليه اسم كل عنصر. مثلا العنصر nav قمنا باستخدامه في الكود كحاوية لقائمة غير مرتبة (قائمة نقطية) وهذه القائمة تحتوي على روابط التنقل عبر الموقع ، وكلمة navigation تعكس لنا وظيفة العنصر حيث يستخدم كحاوية لأكبر من عدد من الروابط المتجاورة. الروابط استخدمنا لها اهم عناصر صفحات الترميز وهو المرساة anchor كما في الاسطر من 30 حتى 36 وفيه العنصر <a > مع الخاصية href والمستخدمه لتحديد موقع التنقل المقصود. كل رابط موضوع في قائمة نقطية unordered list وكل عنصر من هذه القائمة مشار اليه بالعنصر .

الحاوية التالية هي section وهي حاوية مشابهة للـ div والـ article ويفضل استخدامها كحاوية جزئية ضمن الـ article، وقد تكرر هذا العنصر عدة مرات وذلك لاحتواء الصفحة على عدد من العناصر المتتابعة افقيا كما يتضح من الاسطر من 41 حتى 130.

```

<section>
<h2> لغة صفحات التنسيق المتتالي CSS3 </h2>

<p>
<li> المستوى الثاني تقنية معلومات
html5? ما هي لغة
</li>

</p>

.....
.....
<b>
<a href="crs_1st_term2018.html"> ... يتبع .... </a>
</b>
</section>

<section>
<h2> لغة جافا سكريبت </h2>


<p>
ما هي لغة جافا سكريبت
</p>

.....
.....
<b>
<a href="mybooklets.html"> ... يتبع .... </a>
</b>

```

يتم في الاسطر السابقة اضافة عنوان رئيسي من الدرجة الثانية h2 وهو اصغر من ذلك الموجود في الترويسة.

ثم اضافة صورة باستخدام العنصر img ويتم تحديد المصدر source باستخدام الخاصية src ويشير الى مسار الصورة المطلوبة.

وفي الاسطر التالية خاتمة الكود حيث يتم استخدام عنصر هامش سفلي (حاشية) الصورة، ويحتوي فقط على نص بسيط في حالتنا هذه، ويشتهر عنصر الهامش باستخدامه لعرض عناوين التواصل مع اصحاب الموقع او عرض عبارات شهيرة كعبارة حقوق الطبع محفوظة وغير ذلك من الافكار التي يمكنك متابعتها على صفحات النت المختلفة.

```
131 <footer>
132
133 www.Intro2DesingWeb.com
134
135 </footer>
136 </article>
137 </body>
138 </html>
```

فيما يلي لازلنا مع المثال الأول : الصفحة الرئيسية ولكن:

ثانيا: كود الـ CSS

فيما يلي كود صفحة التنسيق المتتالي التي تم ربطها الصفحة اعلاه ، وكما يتضح من الكود القادم ادناه ان لدينا دائما عنصر مختارا وقوسين معقوفين داخلهما خصائص تنسيق تطبق على ذلك العنصر.

ستجد عناصر تتكرر اكثر من مرة كما ستجد استخداما للخاصية كلاس وفي مثال قادم استخدامات اخرى متنوعة، وفي النهاية ايا كان المقصود بما قبل القوسين هناك دائما خصائص يتم تغيير قيمها داخل كل مقطع، وفيما يلي سنشرح اهم الخصائص والتنسيق المطبق من قيمها المكتوبة.

```

body
{
width:700px; margin:auto;
background-color: gray;
}

header, section, nav, article, footer
{

display:block;
background-color:#8f8;
font-weight:bold;
border: solid green;
line-height: 40px;
text-align:center;
padding: 10px ;
}

article.main
{
width: 700px;
background-color: #5f5 ;
padding: 150 ;
}
    
```

عنصر جسم الصفحة سيكون:
عرضه 700 بيكسل والهوامش تلقائية
اما لون الخلفية فهو رمادي.

عناصر متعددة هي الترويسة والقسم
والابحار والمقال والحاشية، كلها
ستتميز بالخصائص التالية:

خاصية الظهور display ستكون على
شكل مقطع block مستقل وهي
خاصية مهمة نستخدمها القيمة
لامذكورة لنضمن عدم تطبيق القيمة
inline التي تعني على السطر كما
في تنسيق ملفات MS-word.

بعد ذلك نحدد خاصية لون الخلفية لكل
العناصر وان يكون الخط سميكاً، وان
يكون ارتفاع خط الكتابة اي التباعد 40
بيكسل، كما ان النص سيكون في
مركز العنصر واخيراً فان هناك مسافة
الحشو عن الطرف لكل عنصر هي 10
بيكسل.

العنصر الرئيسي الذي فنته main،
سيكون عرضه 700 بيكسل وحددنا
معها لون الخلفية ومسافة الحشو
بينه وبين العنصر الرئيسي له، في


```

header
{
color: white;
width: 96%;
text-align: center ;
background-image: url(../img/Landscape_Tree_3.jpg);
}
nav
{
float:right; width:20%; margin-bottom:10px;
}
section
{
float:left; width:70%; margin-bottom:10px;
}
footer
{
clear:both; color: red;
}

```

```

h2
{
font-weight:bold; margin-top:0;
}
img
{
border: solid green;
float: left;
}

div
{
border: solid #5f5 10px;
background-color:#5f5;
}

```

حالتنا هذه الجسم `body`.

عنصر ترويسة الصفحة سيكون:

لون الكتابة فيه ابيض، والعرض قريبا جدا من عرض العنصر الحاضن، والنص في المركز، واخيرا صورة الخلفية ستكون حسب المسار المرفق، لاحظ النقطتين قبل اسم المجلد تعني ان ملف ال CSS يقع داخل مجلد للمجلد الذي فيه الصورة، لهذا نخرج منه ونذهب للمجلد المطلوب.

عنصر `nav` فيه خاصية مهمة وهي خاصية `float` وهي تعني ان العنصر سوف ينزاح تلقائيا لليمين لهذا راعينا ان يكون عرضه + عرض العنصر المجاور له اقل من العرض الكلي للعنصر الحاضن، وكما في الكود ادناه جعلنا العنصر الذي يتبعه ينزاح للييسار وهكذا اصبحامتجاورين مع انهما كان قبل التنسيق متتابعان. لكل منهما ايضا الخاصية هامش سفلي تساوي عشرة بيسكل وهي خاصية مهمة للعناصر التي يطبق عليها عنصر الازاحة، بالتجربة ستفهم القصد اكثر. العنصر حاشية سفلية فيه خاصية مهمة معاكسة لخاصية الازاحة وهي

clear وتعني ان الموقع ثابت لكلا both
الجهتين اليسار واليمين، اما لون
الكتابة فيه فهو احمر.

تنسيق الصورة سيكون حدود كل
صورة تظهر في الصفحة (مصممتا) اي
غير شفاف ولا منقط وستنزاح الصور
تلقائيا لليساار ضمن العنصر الحاضن
لها.

العنصر div موجود لكنه لم يستخدم
فعليا للأسف.

المثال الثاني: صفحة بأكثر من عمود

في الصفحة التي تظهر المثال الثاني لم يتم تغيير شيء في الصفحة الرئيسية عدا حذف محتوى
ال sections مع الاحتفاظ بها، والهدف كان بسيطا .. حيث يتم التعديل في صفحة التنسيق من
اجل عرض الاقسام المذكورة متجاورة عموديا بدلا عن كونها كانت متجاورة افقيا.



وهذا يعني اننا لن نغير في كود ال html سوى جزء استدعاء صفحة التنسيق بتغيير اسم الملف * .css الى اسم اخر سيكون الملف الجديد هو نفس الملف السابق ما عدا اننا سنغير خصائص العنصر section كما هو موضح في الكود التالي:

```
section
{
display:inline;
float : right ; width:20%; margin-bottom:10px;
border: solid 5px green;
padding: 5px;
margin-right: 5px;
}
```

قمنا اعلاه بتغيير الخاصية display الى القيمة inline وهذا يعني ان الاقسام ستكون مجاورة تماما كالكلمات في هذه الصفحة على نفس السطر، وغيرنا خاصية الازاحة الى لايمين .. اما عرض كل قسم فقد أنقصناه لكي تظهر بالشكل الموضح في الصورة، هناك بالطبع احتياج في بعض الاحيان لعمل تنسيق للصفحة بعدة اعمدة وذلك حسب المحتوى المطلوب عرضه.

وهذا هو كل شيء!!!.

المثال الثالث : تحويل روابط الشريط الجانبي الى اشكال ثلاثية البعد

الصفحة التي تعرض المثال الثالث تعرض لنا كيفية تغيير النص الموضوع ضمن الروابط التي تعرض بشكل قائمة غير مرتبة ، والمقصود هنا محتوى الشريط الجانبي اي عنصر التنقل nav، حيث سيتم تنسيق عناصر القائمة بشكل ثلاثي الابعاد وكما في المثال السابق فان الفكرة ستعتمد على اضافة بضع اسطر الى ملف الـ CSS السابق وتغيير اسمه بالطبع وربط هذه الصفحة به.



وكود الـ CSS الذي سيتم اضافته هو الموضح في الشكل التالي:

<pre> nav ul { float: left ; } /* set general side button styles nav li { margin-bottom: 3px; text-align: center; list-style-type:none; width:125px; } /* set general anchor styles */ nav li a { display: block; color: white; font-weight: bold; text-decoration: none } </pre>	<p>العنصر <code>ul</code> القائمة الموجود ضمن العنصر <code>nav</code> سيكون منزاحا لليساار.</p> <p>والعنصر الابن <code>li</code> للـ <code>ul</code> وكلاهما ضمن <code>nav</code> سيكون الهامش السفلي له 3 بيكسل، والنص في المركز/وسط السطر، واي تنسيق للقائمة سيتم الغاؤه، والعرض سيكون 125 بيكسل.</p> <p>اما العنصر <code>a</code> الموجود ضمن عنصر القائمة المحتوى ضمن عنصر الابحار <code>nav</code> فسيكون عرضه بشكل قالب (block)، ولون الكتابة فيه ابيض والخط سميكاً ولا توجد عليه اي تنسيقات نصية.</p>
---	--

الشكل التالي يوضح اهم جزء في الموضوع وهي تنسيقات الازرار ثلاثية البعد، وفيه سيتم التعامل مع عنصر الرابط `a` مباشرة اولا ثم مع حدث مرور الماوس ومن ثم مع حدث النقر. مباشرة سيتم تغيير لون الخلفية ولون الكتابة والحدود ومع الحدود سيتم اضافة سمك الحدود الى القيمة 5 بيكسل ولون الحد وان يكون الحد خارجيا `outset`، فيما سيأتي سيتم تحديد الحدود الداخلية للشكل ثلاثي الابعاد عبر القيمة `inset` لخاصية الحد `border`.

يتم استخدام a: hover لتنسيق النص في الرابط عندما يمر الماوس عليه، وذلك بلون معين ولون اخر للحد على ان يكون سمك الحد 5 بيكسل ولاحظ استخدام outset هنا ايضا للدلالة على الحد الخارجي للشكل ثلاثي الابعاد.

بعد ذلك يتم وضع قيم لخصائص العنصر a في حالة النقر عليه mousedown وفيه يتم تغيير لون الخلفية ثم جعل سمك الحد الداخلي inset قيمته 5 بيكسل.

```
nav li a
{ background:#946055; color: white; border: 5px outset #C96E6B;}

/* mouseover */
nav li a:hover { background: #9F7562; border: 5px outset #C96E6B;}

/* onmousedown */
nav li a:active { background:green; border: 5px inset maroon }
```

المثال الرابع : عارض الصور المبسط

المثال هنا يعرض الصور بشكل مصغر، لقد قمنا باستخدام نسخ مصورة من الصور مع انه بإمكانك استخدام نفس الصور وتحديد الحجم الذي تريده، التنسيق هنا يحدث بأن يتم عرض الصورة التي يضع المستخدم المؤشر عليه.



الكود التالي هو الذي قمنا عبره بادراج ست صور، نسختين من كل صورة واحدة كبيرة وواحدة صغيرة، الصور الصغيرة يجمعها الكلاس thumbs والكبيرة يجمعها الكلاس hid ، واستخدمنا لعرضها نفس الفكرة المستخدمة اعلاه وهي فكرة تغيير تنسيق الرابط في العنصر a.

```
<section>
<ul id = "col-1">
<a href="#"><li><img class="thumbs" src = "img/thumb_Koala.jpg" alt="صورة كوالا" title="كوالا"> </li>
<li><img class = "hid" src = "img/Koala.jpg" alt=" "></li> </a>
<a href="#"><li><img class="thumbs" src = "img/thumb_Jellyfish.jpg" title="قنديل" alt="قنديل البحر"></li>
<li><img class = "hid" src = "img/Jellyfish.jpg" alt=" "></li></a>
<a href="#"><li><img class="thumbs" src = "img/thumb_Lighthouse.jpg" title="فنار" alt="فنار"></li>
<li><img class = "hid" src = "img/Lighthouse.jpg" alt=" "></li> </a>
</ul>
</section>
```

هذا بالنسبة لملف الصفحة ال html.

اما ملف التنسيق فبالطبع سنستخدم نسخة من ذلك المستخدم في المثال الثاني، ونضيف الجزء المطلوب والشكل التالي يوضح الكود المضاف فقط:

```

li
{
margin: 10px; list-style-type:none;
}
/* MouseOut state (default) - create a space for the larger images and hide them*/

#col-1 a img.hid
{
width: 260px; height:390px; position: absolute; top: 70px; left: 190px; visibility: hidden;
}

/* MouseOver state (hover)- reveal larger image */
#col-1 a:hover { background: white;
}
#col-1 a:hover img.hid { visibility:visible;}
img.thumbs
{
width: 120px ; height: 120px;
}

```

لاحظ ال id الذي تم تحديده لعنصر القائمة غير المرتبة ul وهو col-1، حيث يتم تحديد خصائص عناصر الصورة التي كلاس/فئة الصور هي hid والموجودة ضمن العنصر الذي هويته col-1، والخصائص هي: العرض والطول (الارتفاع)، الخاصية Position موقع العنصر وقيمتها اما مطلقة او نسبية، وقد جعلناها مطلقة حتى لا تتأثر بموقعها السابق بعد الصورة المصغرة كما في كود ال html اعلاه، وايضا غيرنا لجميع الصور من نفس الكلاس الخاصيتين top و left وهي خصائص الموقع وتشبهان الاحداثيات س و ص ، بحيث تظهر اي صورة مكبرة في نفس المكان، واخيرا الخاصية visibility خاصة الظهور اسندنا لها القيمة hidden اي اخفاء وذلك لاختفاء جميع الصور المكبرة.

بعد ذلك وضمن خصائص العنصر a حسب حدث مرور الماوس hover وللصور من الفئة hid التي تكون فقط داخل العنصر a، بالطبع ستظهر الصورة التي سيقوم المستخدم بالمرور بالماوس على الرابط الذي فيه الصورة المصغرة الخاصة بها، حيث تم اضافة القيمة visible لخاصية الرؤية visibility .

واخيرا: يتم تحديد طول وعرض الصور المصغرة وهي الصور التي فيها الخاصية كلاس تساوي thumbs كما في كود ال html المنضد اعلاه.

4. النماذج Forms وعناصرها في لغة ترميز النص التشعبي html الاصدار

الخامس

فيما يلي الملف (~chapter-16/form.html) ، من المرجع [1] ، والنموذج (أي ال form) هو طريقة مشهورة في البرمجة المرئية حيث يقوم المبرمج/المصمم بعرض مجموعة من ادوات التحكم بالادخال والايخارج داخل النموذج لكي تظهر للمستخدم فيقوم بتعبئتها ببيانات معينة ثم يعيد ارسالها إلى المخدم غالبا، طريقة ارسال البيانات هي الضغط على زر امر يكون في الغالب موجود اسفل ادوات/عناصر النموذج، ويمكن بالطبع ان لا يتم ارسال بيانات النموذج الى المخدم في حال كان الهدف من البيانات اجراء معالجات في جهاز العميل/المستخدم في هذا الحالة نستخدم نفس الكود باستثناء خصائص النموذج المتعلقة بالتعامل مع المخدم. والشكل ادناه يوضح لنا تصميم الصفحة، وفيما يلي سنقوم بشرح كود الصفحة واهم عناصرها:

Create a New Account

ACCOUNT

First Name:

Last Name:

Email:

Password:

Re-enter Password:

ADDRESS

Street Address:

City:

State:

ZIP Code:

PUBLIC PROFILE

Picture:
Microsoft Word, PDF, GIF, PNG

Screen Name:

Web:

How to message or contact this user:
E-mail:

Gender: Male Female

EMAILS

It is okay to email me with messages from other users.

It is okay to email me with occasional promotions about our other products.

لا يوجد ما يقال حول مقطع كود رأس الصفحة ادناه، غير انها موجهة للغة الانجليزية وان صفحة التنسيق المتتالي التي تتحكم في تنسيقها مرفقة في مجلد بجوار الصفحة بنفس المسار.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Creating a Form</title>
<link rel="stylesheet" href="css/form.css" />
</head>
```

جسم الصفحة حسب المقطع التالي، يوضح لنا ان الحاوية الرئيسية في جسم الصفحة هو العنصر div واول عنصر فيه عنوان رئيسي وبعد مباشرة يأتي عنصر النموذج <form> وهو العنصر الرئيسي لكل عناصر النموذج داخله، ويحتوي في الكود على ثلاثة خصائص: الخاصية method التي تحقق اسلوب ارسال البيانات الى صفحات المخدم وهو post ومعنى ذلك ان الهدف من النموذج ان يقوم المستخدم بتعبئته وارساله مدخلاته إلى المخدم، اما الصفحة التي ستعالج هذا ال post كاملا فهي صفحة بلغة php اسمها showform.php، والخاصية الثالثة enctype فمتعلقة بطريقة ارسال مدخلات المستخدم إلى صفحات المخدم.

بعد ذلك العنصر <fieldset> المستخدم كحاوية عامة لعناصر النموذج.

تلاحظ ان عناصر النموذج موضوعة داخل قائمة نقطية (غير مرتبة unordered list)، اول عناصر القائمة يتضمن داخله العنصر label المستخدم كمرجع للتسمية، لمن هذه التسمية، يتم تحديد ذلك عبر الخاصية for باسناد الخاصية id المتعلقة بالعنصر المقصود بالتسمية.

العنصر input يستخدم لعرض عدد كبير من ادوات التحكم في الادخال ويحدد نوع الاداة الخاصة type ، والوضع الافتراضي له هي القيمة text، فمثلا لو كتبنا السطر:

```
<input type="text" />
```

بدون ذكر خاصية type فإن أداة التحكم التي ستظهر على المتصفح هي المربع النصي، ولكن يفضل دائما ذكر الخاصية التي تحدد ذلك.

الخاصية id في العنصر المذكور استخدمت قبل ذلك في عنصر مربع التسمية، والخاصية name لها وظيفة شبيهة id مع اختلافات معينة، والاخيرة هي المستخدمة اكثر في الكتب الحديثة.

<body>

```
<div id="wrapper">
```

```
<h1>Create a New Account</h1>
```

```
<form method="post" action="showform.php" enctype="multipart/form-data">
```

```
<!-- START ACCOUNT FIELDS -->
```

```
<fieldset>
```

```
<h2 class="account">Account</h2>
```

```
<ul>
```

```
<li>
```

```
<label for="first_name">First Name:</label>
```

```
<input type="text" id="first_name" name="first_name" class="large"
required="required" placeholder="Enter your first name" />
```

```
</li>
```

```
<li>
```

```
<label for="last_name">Last Name:</label>
```

```
<input type="text" id="last_name" name="last_name" class="large" />
```

```
</li>
```

ونشير إلى اننا لسنا مضطرين لوضع عناصر النموذج ضمن عنصري القائمة، ولكن هذا خيار جيد للمصمم لكي تظهر العناصر متتالية بشكل جيد. المقطع التالي يحتوي على العنصر input ولكن

بالخاصية password وهذا يجعله مربعا نصيا ولكن من النوع الذي لا يظهر للمستخدم القيمة المكتوبة داخله، فيكون خاصا بادخال كلمات المرور.

```
<input type="password" id="password2" name="password2" />
```

```
</li>
```

```
</ul>
```

```
</fieldset>
```

لاحظ بعد عرض القسم الاول من العناصر في المقطع السابق يتم اغلاق عنصر fieldset الذي يحتوي على مجموعة من الادوات وهي المنسقة بعنوان رئيسي مستقل كما في الشكل اعلاه. بعد ذلك يتم الابتداء بحاوي جديد مع جملة تعليق تسبقه توضح الهدف منه:

```
<!-- START ADDRESS FIELDS -->
```

```
<fieldset>
```

القسم الثاني فيه عناصر مربع التسمية ومربع النصوص ومن ثم يأتي العنصر التالي:

```
<li>
```

```
<label for="state">State:</label>
```

```
<select id="state" name="state">
```

```
<option value="AL">Alabama</option>
```

```
<option value="AK">Alaska</option>
```

```
<option value="CA">California</option>
```

```
</select>
```

```
</li>
```

العنصر اعلاه يستخدم وسم ال select للتصريح عن البدء بقائمة منسدلة، وداخل هذا العنصر يتم استخدام عدد من العناصر option كل واحد منها يتضمن نصا هو الذي سيتم ادراجه للعرض في

القائمة المنسدلة. أما الخاصية value الموضحة اعلاه، فتستخدم من اجل القيمة التي سوف يتم ارسالها عبر ال post إلى صفحات المخدم، في حال قام المستخدم باختيار ذلك الاختيار (ال option) المعروض.

الكود التالي متعلق بالقسم الثالث للنموذج، حيث يتم عرض نوع آخر للعنصر input وهو النوع ملف (type="file")، هذا العنصر يحتوي على مربع تسمية زائدا زر امر، ووظيفة زر الامر فتح استعراض للملفات داخل الحاسب المحلي، لكي يقوم المستخدم باختيار ملف ثم الموافقة عليه، عندئذ يظهر اسم الملف في مربع التسمية المجاور.

لاحظ بعد هذا العنصر هناك فقرة تعطي تعليمات للمستخدم بنوع ملفات الصور المقبول ارسالها للمستخدم.

```
<label for="picture">Picture:</label>
```

```
<input type="file" id="picture" name="picture" />
```

```
<p class="instructions">Maximum size of 700k. JPG, GIF, PNG.</p>
```

```
</li>
```

وفي العنصر input الذي له الخاصية (type = "url")، يتم عرض مربع نصي عادي ولكنه لن يقبل اي نصا عاديا، فبمجرد الضغط على زر الامر submit في اخر النموذج، فإن النموذج يفحص قيمة المربع اذا لم تكن رابطا لموقع على النت، فإنه يوقف ارسال النموذج ويعطي تلميحا للمستخدم بانه هناك خطأ، جرب ذلك بنفسك؟.

```
<li>
```

```
<label for="bio">Bio:</label>
```

```
<textarea id="bio" name="bio" rows="8" cols="50"></textarea>
```

```
</li>
```

الكود اعلاه يحتوي على العنصر textarea وهو مربع نصي كبير يحتاج الى خصائص تحدد عدد الكلمات والاعمدة والصفوف داخله، وإلا فيتم اعتماد القيم الافتراضية، جرب ذلك؟.

في الكود التالي العنصر input مع الخاصية (type = "radio") ، والتي تعرض زر اختيار ، وهو زر يعرض اختيارا محدد في حال نقر عليه المستخدم، واذا كان هناك زر اختيار آخر وقام المستخدم باختياره، يتم ازالة الاختيار من على الاول والعكس بالعكس.

اما الخاصية "checkbox" فتشبه زر الاختيار، إلا انها تعرض لنا مربع الاختيار، وهو يختلف عن زر الاختيار، في الشكل وفي الوظيفة، فشكله مربعا (وزر الاختيار دائرة) ووظيفة مربع الاختيار ان يتم التاشير عليه او ازالة التاشير بشكل مباشر فهو يقوم بدور زر الاختيار معا.

```
<input type="radio" ...../>
```

```
<input type="checkbox" id="email_ok_msg_from_users" name="email_signup[]" value="user_emails" />
```

مربعات الاختيار سالفة الذكر موجودة في القسم الرابع، وبنهاية القسم الرابع تأتي اخر الاسطر في كود الصفحة كما يلي:

```
<fieldset>
```

```
<input type="submit" class="create_profile" value="Create Account" />
```

```
</fieldset>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

يتم وضع العنصر input في قسم خامس، وهو يعرض عنصرا له الخاصية type تساوي القيمة submit التي تعني ارسال، وهذا يعطينا واحدا من اهم عناصر النموذج وهو زر الامر، وزر الامر يستخدم من اجل تنفيذ امر واحد او عدة وظائف ويمكن ان نستخدم لعرضه الوسم button الذي له نفس الهدف، إلا ان القيمة submit لخاصية النوع تعني امرا اخر بالاضافة لكونه زر امر، فهي تعني ان كل البيانات المعبئة في النموذج ، أي تلك التي ملأها المستخدم، ستتحوّل إلى قائمة من القيم values وسوف يتم ارسالها الى صفحة المخدم المحددة في الخاصية action التابعة للعنصر form.

لزر الامر وظائف اخرى طبعا ويمكن برمجته بلغة javascript لعمل وظائف مختلفة، ولكنها وظائف متعلقة بجهة الزبون او المستخدم، وغالبا لا يكون لها علاقة بالمخدم.

ملاحظة: هناك عناصر أخرى لم تظهر في هذا المثال يرجى الرجوع إليها في الكتاب.

5. لغة الجافا سكريبت : مقدمة.

ما هو الجافا سكريبت? What is JavaScript?

- هي لغة برمجة نصية JavaScript is a scripting language.
- واللغة النصية هي لغة برمجة خفيفة الوزن A scripting language is a lightweight programming language
- تتكون الجافا سكريبت من اسطر لأكواد الكمبيوتر القابلة للتنفيذ A JavaScript consists of lines of executable computer code
- يتم تضمين الجافاسكريبت في صفحات الـ .html . A JavaScript is usually embedded directly into HTML pages
- تم تصميم لغة الجافا سكريبت لكي تضيف تفاعلا لصفحات الـ اتش تي ام ال JavaScript was designed to add interactivity to HTML pages
- لغة الجافا سكريبت مجانية بمعنى ان اي احد يستطيع استخدامها بدون ترخيص JavaScript is free. Everyone can use JavaScript without a license

البرمجة من جهة المستخدم/العميل/الزبون Client-side Scripting

عن طريق الجافاسكريبت يتم برمجة سلوك behavior او تصرفات المتصفح (المستعرض)، وهذا يسمى بالبرمجة النصية لجهة المستخدم client-side scripting (أوالبرمجة النصية للمتصفح (browser scripting).

بينما البرمجة النصية جهة السرفر/المخدم تتعلق ببرمجة سلوك او تصرفات المخدم على سبيل المثال لغة ASP أو لغة PHP.

يتم اضافة كود الجافا سكريبت داخل صفحة الـ html بهذا الشكل:

<html>


```
<head><title>سكرت الجافا </title>
```

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

-

به المرغوب الكود هنا يكتب

-

```
</script>
```

```
</head>
```

```
<body></body>
```

```
</html>
```

١- أمر الطباعة :

الطريقة الأولى : طباعة جملة لا يقع عليها تأثير اي وسم من وسوم ال (html) ...

```
document.write(" هنا الجملة المراد طباعتها");
```

الطريقة الثانية : طباعة جملة يقع عليها تأثير وسم ال (html) ...

```
document.write("<h1> هنا الجملة المراد طباعتها </h1>");
```

الطريقة الثالثة : طباعة جملة يقع عليها تأثير وسم ال (htm) و ال (style) ...

```
document.write("<h1 style='color : red'> هنا الجملة المراد طباعتها </h1>");
```

الطريقة الرابعة : طباعة قيمة متغير ...

```
document.write( sum );
```

الطريقة الخامسة : طباعة قيمة متغير يسبقه جملة نصية ..

```
document.write(" + sum ) هو الناتج هو ");
```

٢- خروج نافذة للمستخدم نحدد نحن ما يكتب بها وهي خاصة في كائنات النوافذ **Window Object**.

كهذه النافذة :



وتسمى هذه النافذة بالجافا سكربت (**alert**) ويكتب كودها بهذه الطريقة ...

```
window.alert(" أهلا بك في موقعنا ");
```

* بعض الاوامر التي تستخدم داخل كود ال (**alert**):

الامر	وصف الأمر	مثال ..
\n	سطر جديد	window.alert("hi \n all");
\t	لترك مسافة بين كل كلمة وكلمة بحيث تظهر كجدول	window.alert("hi \t all");
\r	لوضع كل كلمة بسطر ولكن بإختلاف ال position	window.alert("hi \r all");
\\	لكتابة رمز (\)	window.alert("hi \\ all");
\"	لكتابة رمز (")	window.alert("hi \" all");
'	لكتابة رمز (')	window.alert("hi \' all");

٣- خروج نافذة للمستخدم يحدد هو ما يكتب بها " اي نافذة قراءة من لوحة المفاتيح " وهي خاصة بكائنات النوافذ **Window Object** .

كهذه النافذة :



وتسمى هذه النافذة بالجافا سكريبت (**prompt**) ويكتب كودها بهذه الطريقة ...

```
window.prompt ("اسمك","من فضلك أدخل اسمك");
```

ملاحظة :

يأخذ كود الـ (**prompt**) بداخله قيمتين الاولى يكتب بها ما سوف يخرج للمستخدم في أعلى النافذة .. والقيمة الثانية يكتب بها قيمة ابتدائية بداخل صندوق الكتابة وهذه القيمة ليس لها اي قيمة فعلية فهي ما تلبث حتى ينتهي دورها بمجرد ان يكتب المستخدم مكانها اسمه او اي شيء يريد ... جرب ان تضع هذا الامر وانظر ماذا يخرج لك والذي يعتبر ايضا من كائنات النوافذ **Window Object**

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

وهي عبارة عن نافذة تخرج للمستخدم لسؤاله عن قبول أو رفض شيء معين أنت تحدده .

■ **المتغيرات :**

نظرة سريعة على المتغيرات :

كما تعلم في أي لغة برمجة تعتبر المتغيرات ذو فائدة كبيرة فبواسطتها نستطيع التحكم ببرامجنا بكل سهولة ...

ففي لغة الجافا سكريبت تعرف المتغيرات بـ (**var**) .

مثال :

```
var name ;
```

ويجدر التنبيه هنا انه اذا اردنا أن يكون المتغير قيمة رقمية (integer Or float) " أعداد صحيحة أو أعداد ذات أرقام عشرية " فإننا نقوم بتحويلها بهذه الطريقة :

1- parseInt (أسم المتغير هنا) ;

وهذا سوف يحول العدد الى عدد صحيح (integer) أي يكون كتابة الكود كالتالي :

```
var number = 55 ;  
parseInt (number) ;
```

2- parseFloat (أسم المتغير هنا) ;

وهذا سوف يحول العدد الى عدد عشري (float) أي يكون كتابة الكود كالتالي :

```
var number = 5.4 ;  
parseFloat (number) ;
```

ملاحظة :

عليك الإلتزام بشكل الحرف أي الحرف الصغير والكبير (parseFloat) .
..وللحديث بقية سوف يذكر في وقته .

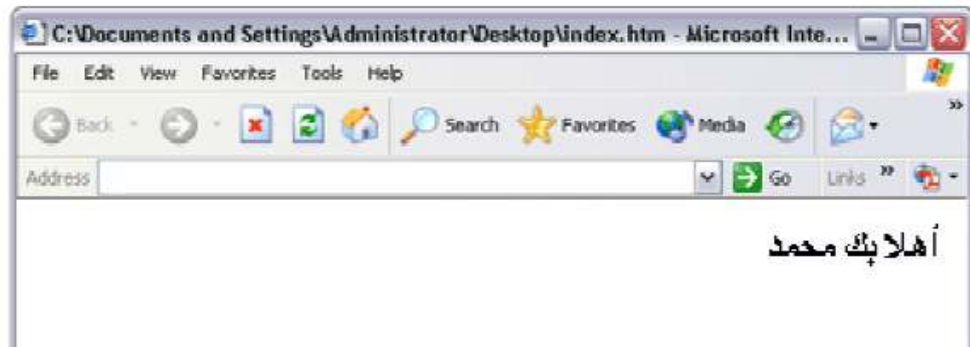
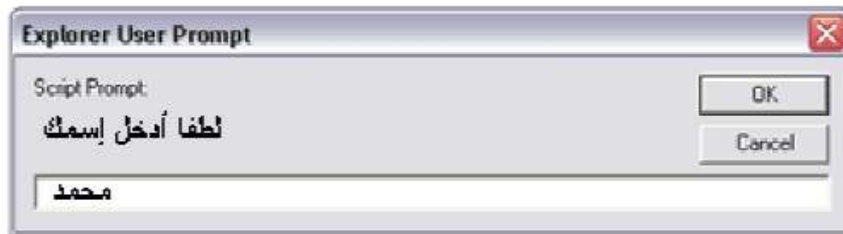
لنأخذ أمثلة على ما سبق ونطبقها ...

لنأخذ أمثلة على ما سبق ونطبقها ...

مثال ١ :

```
<html dir="rtl">
  <head><title>الجافا سكريبت</title>
  <script>
    var name ;
    name = window.prompt ( " لطفًا أدخل إسمك " , " إسمك " ) ;
    document.write( " أهلا بك " + name ) ;
  </script>
</head>
<body> </body>

</html>
```



=====

وباستثناء طرق الادخال والاخراج فإن لغة الجافا سكريبت تتشابه مع اي لغة مستقاة من الـ C++ والتي تسمى (شكل سي C-style) في باقي الجمل، كالجمل الشرطية (if .. else) او الجمل الشرطية (switch....case) او الجمل التكرارية (for statment) او (while statment) وغير ذلك من طرق البرمجة ككتابة الدوال والمصفوفات والتعامل مع الكائنات والطرق وغيرها.

الكود التالي يعرض صفحة html فيها حلقة تكرارية كمثال لما سبق:

```
<html >
  <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">
    for (var i =10 ; i>=0 ; i--)
      {
        document.write( i ) ;
      }
    document.write ( " <br /> بالأرقام بشكل تنازلي ")
  </script>
</head>
<body></body>
</html>
```

نفذ الكود السابق كتمرين ولاحظ مخرجاته.

وفيما يلي مثال يبين لنا تعريف دالة لطرح عددين، واستدعائها :

مثال :

```
<html>
  <head><title >الجافا سكريبت</title>
  <script type = "text/javascript">
    var x = 4 , y = 2 ;
    subtract ( x , y ) ;
    document.write ( " <h3> هذا هو الناتج </h3> " ) ;

    function subtract ( x , y )
      {
        var sub = 0 ;
        sub = x - y ;
        document.writeln ( sub ) ;
      }
  </script>
</head>
<body></body>
</html>
```

اما المثال التالي فيوضح كيفية استدعاء الدوال خارج العنصر script:

لاحظ استخدام الحدث بالخاصية onload والذي يعني عند التحميل، فعند تحميل جسم الصفحة سيتم استدعاء الدالة output() التي ستقوم بدورها باستدعاء الدالة cube، وهكذا يتم طباعة ناتج تكعيب الرقم المدخل.

ولكن لاحظ عدم تنفيذ اول سطر ضمن العنصر script، لماذا؟

```
<html>
<head>
<script>
document.write("<h1 style='color:red'>بِسْمِ اللَّهِ نَبْدَأُ</h1>");
function output ( )
{
var x = window.prompt (x); ;
var t = cube ( x ) ;
document.writeln ( t ) ;
}
function cube ( y )
{
return y*y*y ;
}
</script>
</head>
<body onload = " output ( ) ">
</body>
</html>
```

المصفوفات:

المثال التالي يبين تعريف مصفوفة وتعبئة بياناتها عبر حلقة for:

```
<html>
  <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">
    var i ;
    var number=new Array() ;
    for (i=0 ; i<4 ; i++)
      {
        number[i] = i ;
        document.writeln(number[i] + "<br>");
      }
  </script>
</head>
<body></body>
</html>
```

كائنات اللغة:

توجد في اللغة كائنات جاهزة للاستخدام مثل كائن الرياضيات math والمثال التالي يوضح بعض دواله:

```
<html>
  <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">

    document.write(Math.sqrt(9));
    document.write("<br />");
    document.write(Math.max(10,2));
    document.write("<br />");
    document.write(Math.sin(1.0));

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

اما كائنات السلسلة النصية string ففيها دوال كثيرة وكمثال عليها:


```
<html dir="rtl">
  <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">

    var a ="samialr";
    document.write("<p> اطبع الحرف الموجود في الموقع صفر " + a.charAt(0) );

  </script>
</head>
<body></body>

</html>
```

ويوجد ايضا كائن الوقت Date وهذا المثال يوضح بعض وظائفه:

```
<html>
  <head><title>الجافا سكريبت</title>
  <script type = "text/javascript">

    var d = new Date() ;
    document.writeln("<p> طباعة رقم اليوم من الشهر " + d.getDate());
    document.writeln("<p> طباعة رقم الشهر " + d.getMonth());
    document.writeln("<p> طباعة السنة لنا بأربع أرقام " + d.getFullYear());

  </script>
</head>
<body></body>

</html>
```

مثال :

وهو تعبئة نموذج من قبل المستخدم وبعد التعبئة يضغط على أيقونة موافق فيطبوع بياناته في صفحة الإنترنت

```

<html dir="rtl">
  <head><title> java script </title>
  <script type="text/javascript">
    function information()
    {
      var name=info.name.value;
      var job=info.job.value;
      var add=info.address.value;
      var comm=info.command.value;
      document.writeln("<table cellpadding="+"0"
      cellspacing="+"5" border="+"1" width="+"80%"
      align="center" dir="rtl" bgcolor="+"#eaeaea"
      bordercolor="+"#000000">");
      document.writeln("<caption><h3>
      المعلومات التي سجلتها لدينا
      </h3></caption>");
      document.writeln("<tbody>");
      document.writeln("<tr><td width="+"30"
      ><b> : الأسم </b>
      </td><td>" + name + "</td></tr>");
      document.writeln("<tr><td width="+"30"
      ><b> المهنة
      :</b></td><td> " + job + "</td></tr>");
      document.writeln("<tr><td width="+"30"
      ><b> : العنوان</b></td><td> " + add + "</td></tr>");
      document.writeln("<tr><td width="+"30"
      ><b> : ملاحظات</b></td><td> " + comm + "</td></tr>");
      document.writeln("</tbody>");
      document.writeln("</table>");
    }
  </script>

```

```

</head>
<body >
<form name="info" action="">
<table cellspacing="0" cellpadding="5" border="0" width="80%" align="center">
<tbody>
<tr>
<td>الاسم :</td>
<td><input type="text" name="name" value="" size="31"></td>
</tr>
<tr>
<td>العنوان:</td>
<td><input type="text" name="address" value="" size="45"></td>
</tr>
<tr>
<td>المهنة:</td>
<td><input type="text" name="job" value="" size="20"></td>
</tr>
<tr>
<td>ملاحظات:</td>
<td><textarea name="command" rows="4" cols="50"></textarea></td>
</tr>
<tr>
<td colspan="2" align="center">
<br>
<input type="button" name="show" value="عرض"
onclick="information()"><input type="reset" name="del" value="مسح" >
</td>
</tr>
</tbody> </table></body></html>

```

ومخرجات هذا المثال هي:





الامثلة السابقة في الجافا سكريبت، في هذا القسم فقط، مأخوذة من هذا الكتيب الموجود على النت، فشكرا لمؤلفه مع كل التقدير .

تعليم الجافا سكريبت
JAVASCRIPT LERNNING

إعداد :
م . سامي الربيعية

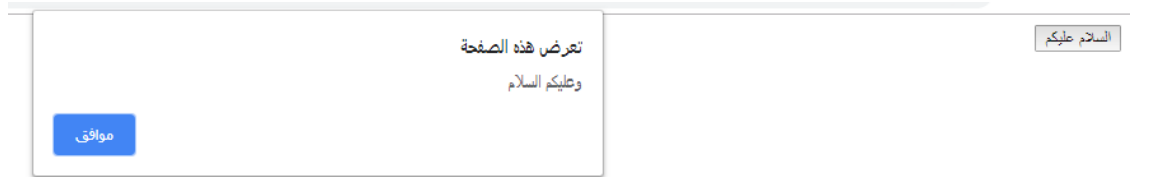
6. جافا سكريبت: المجموعة الثانية من الأمثلة

أمثلة متنوعة في استخدام لغة الجافا سكريبت مع المتصفح

هناك ثلاثة طرق لاضافة كود جافا سكريبت الى المتصفح، بشكل مشابه لصفحات التنسيق المتتالي، مع اختلافات يسيرة خاصة بلغة الجافاسكريبت، وهذه الطرق هي:

1. كتابة الكود مباشرة ضمن العنصر script، وهذا يمكن ان يظهر في جزء راس الصفحة او جسم الصفحة، ويمكن كتابة كود مباشر ولكن يفضل كتابته على شكل دوال، ويتم استدعاء الدوال اما في بداية العنصر script او
2. استدعاء دوال مكتوبة ضمن اي عنصر نريد، وذلك في خاصية تستدعي حدثا معيناً، مثل حدث النقر onclick او حدث مرور الماوس onmouseover او الضرب على لوحة المفاتيح onkeydown وهناك احداث كثيرة كل حدث يناسب عنصرا معيناً، فمثلا النقر يناسب زر الامر الضرب على مفتاح يناسب مربع النص وهكذا.
- مع انه بالامكان كتابة سطر برمجي او اكثر ضمن الخاصية بدلا من كتابة اسم دالة مكتوبة مسبقاً، وهذا لا يجعل الكود سهل القراءة، لهذا يستخدم في اضيق الحدود.
3. كتابة ملف نصي امتداده *.js فيه دوال متعددة واستدعاؤها داخل صفحة الـ html، بشرط ان يتم ربط الملف بالصفحة عبر نفس العنصر script ولكن باستخدام الخاصية src.

المثال الاول:



هذا المثال يعرض الرسالة عند الضغط على الزر حسب الشكل الموضح اعلاه، بالكود التالي:

<html>	الكود التقليدي لراس الصفحة
<head><title>مثال</title></head>	
<body dir="rtl">	

<code><script type="text/javascript"></code>	عنصر ارفاق دالة جافا سكريبت داخل صفحة html
<code>function salam() {</code>	التصريح عن الدالة باسمها و قوس بداية جسم الدالة
<code>alert("وعليكم السلام");</code>	جملة طباعة الرسالة الموضحة بالشكل السابق
<code>}</code>	انتهت الدالة
<code></script></code>	اغلاق عنصر ارفاق الدالة
<code><button onclick="salam();"> السلام </button></code>	استخدام الخاصية التابعة لعنصر زر الامر لاستدعاء الدالة باسمها
<code></body></code>	
<code></html></code>	

=====

المثال الثاني:

المثال التالي والموضح بالشكل أدناه، فيه زر امر واحد لاضاءة المصباح On والثاني لاغلاق النور Off، والكود ادناه يوضح كيف يتم ذلك:



<code><!DOCTYPE html></code>	راس الصفحة
<code><html></code>	
<code><body></code>	

<p><h2>What Can JavaScript Do?</h2></p>	
<p><p>JavaScript can change HTML attribute values.</p></p>	
<p><p>In this case JavaScript changes the value of the src (source) attribute of an image.</p></p>	
<p><button onclick="document.getElementById('myImage').src = 'pic_bulbon.gif' " > Turn on the light </button></p>	<p>يمكن كتابة كود الجافا سكريبت ضمن خاصية حدث النقر. الكود المرفق هناك</p>
<p></p>	<p>يستدعي عنصر الصورة حسب الخاصية id، ويقوم بتغيير الخاصية src التابعة لها وبالتالي يغير الصورة. من الصورة التي المصباح فيها منطفئ الى الصورة المنورة.</p>
<p><button onclick="document.getElementById('myImage').src = 'pic_bulboff.gif' " > Turn off the light </button></p>	<p>هنا يتم تغيير الصورة إلى الصورة غير المضاءة.</p>

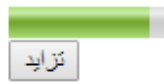
<pre></body> </html></pre>	
--	--

=====

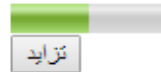
المثال الثالث :

تزايد مقياس التدرج بالنقر على زر الامر وكل مرة يتم اضافة كلمة الى الحاوية التي قبله، وذلك حسب الشكلين التاليين:

تزايد fahdfahdfahdfahd



تزايد



<pre><!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>Index</title> </head> <body>
 <div> <p id="dv">: مقياس التقدم ادناه </p> <meter id="mrt" value="0.5"> يبدأ من نصف قيمته </meter>
</pre>	
<pre><button onclick="mtr()"> تزايد </button> </div></pre>	<p>الكود هنا يستدعي الدالة التي تزايد قيمة المقياس</p>

<pre><script> function mtr() { mt=document.getElementById("mrt"); dv = document.getElementById("dv"); dv.textContent += "fahd"; if (mt.value == 1) mt.value -=0.5; mt.value +=0.10; } </script></pre>	<p>الدال التالية</p> <p>تحمل اسما لكي يتم استدعاؤها به</p> <p>يتم اخذ عنصر المقياس إلى متغير حسب الخاصية id، ونفس الامر بالنسبة لعنصر الـ div اضافة محتوى نصي إلى عنصر div الاحتوى</p> <p>عندما يصل المقياس الى قيمته العظمى يعود يعود الى النصف وإلا فإنه يتزايد بمقدار العشر كل مرة يتم فيها استدعاء الدالة، اي الضغط على زر الامر اعلاه.</p>
<pre></body></pre>	
<pre></html></pre>	

=====

المثال الرابع:

المثال التالي يغير قيمة خاصية الـ id للحاوية div وبهذا فإنه يغير خاصية الاخفاء والاطهار له حسب الزر الذي تنقر عليه:

<pre> <!DOCTYPE html> <html dir="rtl"> <head> <meta charset="UTF-8"> <title>Index</title> </head> <body> <div id="dv1"> <p >فقرة واحد</p> <p >فقرة اثنين</p> </div> <p >فقرة رقم ثلاثة</p> <div id="dv2"> <input > مربع نصي </input>
 <button onclick="fhd()"> إظهار </button> </div> <style> #dv1 { background:#225588; display: none; } div </pre>	<p style="text-align: center;">فقرة رقم ثلاثة</p> <p>مربع نصي <input type="text"/></p> <p style="text-align: right;">إظهار</p> <p style="text-align: right;">إخفاء إخفاء</p> <p style="text-align: right;">فقرة رقم ثلاثة</p> <p style="text-align: right;">إخفاء إخفاء</p> <p style="text-align: right;">فقرة واحد</p> <p style="text-align: right;">فقرة اثنين</p> <p style="text-align: right;">فقرة رقم ثلاثة</p> <p>مربع نصي <input type="text"/></p> <p style="text-align: right;">إظهار</p> <p style="text-align: right;">إخفاء إخفاء</p> <p style="text-align: right;">الحاوية التي سيتم اخفاؤها</p>
--	--

<pre> { display: block } </style> <script > function fhd() { var x =document.getElementById("dv1"); x.id="dv2"; } function dhf() { var x =document.getElementById("dv2"); x.id="dv1"; } </script>
 <input type="button" value = إخفاء " onclick="dhf()"> إخفاء </input> </body> </html> </pre>	<p>الفقرة خارج الحاوية لن تختفي !!</p> <p>الحاوية الثانية سيتم اخفاؤها ايضا</p> <p>زر اظهار الحاوية الاولى</p> <p>مقطع ال CSS</p> <p>خاصية العرض وقيمتها لا شيء فتؤدي الى عدم العرض</p> <p>اي حاوية ستكون نوع العرض لها مربع فستكون ظاهرة غير مخفية باستثناء تلك التي عرضها لا شيء كما سبق.</p> <p>الجزء البرمجي بالجافا سكريبت الدالة الاولى</p> <p>اسناد العنصر/الحاوية التي هويتها dv1 إلى</p>
--	---

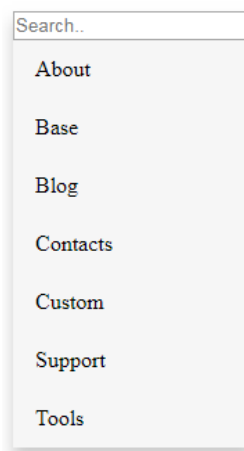
	<p>المتغير X</p> <p>تغيير هوية المتغير/العنصر إلى هوية أخرى</p> <p>الدالة الثانية</p> <p>تبدل أسماء الهويات فتتغير خاصية الاظهار والاختفاء</p>
--	--

=====

المثال الخامس:
البحث في قائمة كلمات:

Search/Filter Example

مثال للبحث داخل قائمة كلمات



الكود التالي يوضح طريقة تنسيق هذه القائمة والكود المستخدم لتنفيذها، المهم فيه هو دالة الفلتر، التي يتم استدعاؤها عند حدث الكتابة على مربع النص، وهي مشروحة مع التنسيق في المثال السادس بالتفصيل:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.dropdown-content
```

```
{ position: absolute; background-color: #f6f6f6; box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); }
```

```
.dropdown-content a
```

```
{ color: black; padding: 12px 16px; text-decoration: none; display: block; }
```

```
.dropdown a:hover
```

```
{ background-color: #ddd }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div >
```

```
<h2>Search/Filter Example</h2>
```

```
</div>
```

```
<div>
```

```
<h2>كلمات قائمة داخل للبحث مثال</h2>
```

```
</div>
```

```
<div class="dropdown">
```

```
<div id="myDropdown" class="dropdown-content">
```

```
<input type="text" placeholder="Search.." id="myInput"
```

```
onkeyup="filterFunction()">
```

```
<a href="#about">About</a>
```

```
<a href="#base">Base</a>
```

```
<a href="#blog">Blog</a>
```

```
<a href="#contactc">Contacts</a>
```

```
<a href="#custom">Custom</a>
```

```
<a href="#support">Support</a>
```

```
<a href="#tools">Tools</a>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
function filterFunction() {
```

```
var input, filter, div, a, i;
```

```
input = document.getElementById("myInput");

filter = input.value.toUpperCase();
div = document.getElementById("myDropdown");
a = div.getElementsByTagName("a");
for (i = 0; i < a.length; i++) {
    if (a[i].innerHTML.toUpperCase().indexOf(filter) > -1) {
        a[i].style.display = "";
    } else {
        a[i].style.display = "none";
    }
}
}
}
</script>
</body>
</html>
```

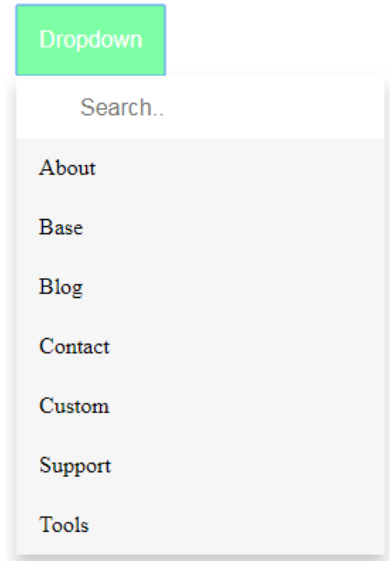
=====

المثال السادس:

قائمة منسدلة بلغة تنسيق متتالي وتظهر عند الضغط على المربع الاخضر، وداخلها مربع النص للبحث عن احد عناصر القائمة، مصممة بالجافا سكريبت كما هو موضح في الكود ادناه:

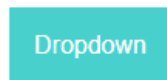
Search/Filter Dropdown Ex

Click on the button to open the dropdown m



Search/Filter Dropdown Example

Click on the button to open the dropdown menu, and use the input field to search for a sp



```
<!DOCTYPE html>
<html>
<head>
<style>
.dropbtn
{ background-color: #48d1cc; color: white; padding: 16px; font-size: 16px;
border: none; cursor: pointer; }
.dropbtn:hover, .dropbtn:focus
{ background-color: #7effa5; }
#myInput
{ border-box: box-sizing; background-image:url('search.png');
background-position: 14px 12px; background-repeat: no-repeat; font-size:
16px; padding: 14px 20px 12px 45px; border: none; }
.dropdown
{ position: relative; display: inline-block; }
```



```

.dropdown-content
{ display: none; position: absolute; background-color: #f6f6f6; min-width:
230px; overflow: auto; box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); }
.dropdown-content a
{ color: black; padding: 12px 16px; text-decoration: none; display: block; }
.dropdown a:hover
{background-color: #ddd}
.show {display:block;}
</style>
</head>
<body>
<h2>Search/Filter Dropdown Example</h2>
<p>Click on the button to open the dropdown menu, and use the input field to
search for a specific dropdown link.</p>
<div class="dropdown">
<button onclick="myFunction()" class="dropbtn">Dropdown</button>
<div id="myDropdown" class="dropdown-content"> <input type="text"
placeholder="Search.." id="myInput"onkeyup="filterFunction()">
<a href="#about">About</a>
<a href="#base">Base</a> <a href="#blog">Blog</a>
<a href="#contact">Contact</a>
<a href="#custom">Custom</a>
<a href="#support">Support</a>
<a href="#tools">Tools</a>
</div>
</div>

```

```
<script>
function myFunction()
{
document.getElementById("myDropdown") .classList.toggle("show");
}
function filterFunction()
{
var input, filter, ul, li, a, i;
input = document.getElementById ("myInput");
filter = input.value.toUpperCase();
div = document.getElementById ("myDropdown");
a = div.getElementsByTagName("a");
for (i = 0; i < a.length; i++)
{
if (a[i].innerHTML.toUpperCase() .indexOf(filter) > -1)
{
a[i].style.display = "";
}
Else
{
a[i].style.display="none";
}
}
}
}
</script>
</body>
```

</html>

بالنسبة للكود السابق ، لدينا عدد قليل من عناصر التنسيق المتتالي الجديدة مثلا: cursor: pointer، وهي تقوم بتغيير شكل مؤشر الماوس عند المرور على العنصر المحدد. أما الاختيار dropbtn:focus فيجري تغيير في التنسيقات عند حدوث عملية نقل التركيز الى العنصر المحدد، بينما في الاختيار dropbtn:hover فيجري التغييرات في التنسيقات عند مرور الماوس على العنصر المحدد.

الجدول التالي يلخص بعض خصائص التنسيق المهمة:

background-repeat: no-repeat;	عدم السماح بتكرار صورة الخلفية في حال
display: none;	حجم الصورة صغيرة.
box-shadow: 0px 8px 16px 0px	عدم عرض العنصر المحدد.
rgba(0,0,0,0.2);	عمل ظل ثلاثي الابعاد بالاحجام لجهات الاربع
text-decoration: none;	واللون المذكور.
	الغاء تنسيقات النص المحدد.

بعد ذلك استدعاء دالة الجافاسكريبت myFunction() عند النقر على زر الامر التالي:

```
<button onclick="myFunction()" class="dropbtn">Dropdown</button>
```

ثم استدعاء للدالة filterFunction() في الحدث onkeyup التابع لعنصر مربع النص، وهو حدث رفع المستخدم اصابعه من احد ازرار لوحة المفاتيح اثناء الكتابة على مربع النص.

```
<input type="text" placeholder="Search.." id="myInput" onkeyup ="filterFunction()">
```

والحدث onkeydown هو الحدث السابق له اي حدث الضغط على زر من لوحة المفاتيح.

عند الوسم <script> يبدأ محتوى دوال الجافا السكريبت المستخدمة في هذا الكود، وهما الدالتان:

1. myFunction() : وفيها سطر واحد يستدعي العنصر الذي لديه الهوية Id = myDropdown، ويستدعي الخاصية classList التي عن طريقها يتم استدعاء الدالة toggle ويمرر لها القيمة .show

2. filterFunction(): وفي اول سطر من الدالة يتم تعريف ستة متغيرات ، لكننا بعد ذلك لن نحتاج سوى اربعة منهم، وهذا خطأ غير مقصود من المبرمج، المتغيرات الاربعة يتم اسناد قيم لها بعد ذلك كالتالي:

```
var input, filter, ul, li, a, i;
input = document.getElementById ("myInput");
filter = input.value.toUpperCase();
div = document.getElementById ("myDropdown");
a = div.getElementsByTagName("a");
```

لاحظ الاسنادات اعلاه، حيث يتم اسناد المتغير input بحيث يأخذ العنصر الذي له الهوية myInput، وهو مربع النص الخاص بالبحث.

اما المتغير filter فيأخذ القيمة التي تم كتابتها في مربع النص المذكور ويحولها الى كتابة كبيرة (كابتل) وذلك لكي تكون المقارنة موحدة فلن يختلط الامر بين الحروف الصغيرة والكبيرة.

المتغير div يأخذ عنصر الاحتواء div الذي في الصفحة، وهو الذي يعرض القائمة.

والمتغير a هو مصفوفة وليس متغيرا مفردا فهو يأخذ جميع العناصر التي لها الاسم a ، لانه لا يحدد هوية ولكن اسم الوسم tagname، فسيكون عبارة عن مصفوفة من القيم، بالعودة الى كود ال html سنعرف ان حجمه سيكون ستة عناصر لان لدينا في الصفحة ستة عنصر لها نفس الوسم.

الحلقة التالية تمر على جميع عناصر المصفوفة a حسب طولها، يعني ست مرات، وكل مرة تقوم بالبحث عن القيمة المدخلة في مربع النص التي ستخزن في المتغير filter ، وكما ذكرنا فإن المقارنة ستتم بتحويل جميع الحروف الى كابتل، وبعد المقارنة اذا طابقت الكتابة المحتوى النصي ، او جزء من المحتوى النصي لاحد العناصر، فإنه يبقى وسيتم بعدها حذف العناصر الاخرى من عنصر الاحتواء div، وهذا خلاصة الحلقة والجملة الشرطية ادناه.

```
for (i = 0; i < a.length; i++)
{
if (a[i].innerHTML.toUpperCase().indexOf(filter) > -1)
{
a[i].style.display = "";
```

```

}
Else
{
a[i].style.display="none";
}
}
}
}

```

ولمزيد من التوضيح للنظر في السطر التالي الخاص بالجملة الشرطية اعلاه:

```
if (a[i].innerHTML.toUpperCase().indexOf(filter) > -1)
```

تم التعامل مع a كمصفوفة دليلها i الذي يتزايد كل مرة.

تم استخدام الكائن innerHTML الذي نصل عن طريقه للمحتوى داخل العنصر المذكور، وبالمحتوى نقصد كل شيء بين وسم الفتح ووسم الاغلاق، اما الدالة indexOf فتبحث عن نص داخل نص، وسبتحت عن ما كتبه المستخدم وتم تخزينه داخل المتغير filter داخل محتوى العنصر المقصود بالمصفوفة a ، وسوف يتم اعادة رقم يحدد موقع النص المبحوث عنه في حال كان موجودا، اما اذا كان غير موجود فالنتيجة سوف تكون -1، وفي حال تحقق الشرط فسوف يظهر فقط العنصر المحدد، اي العنصر الذي داخله مكتوب النص الذي بحثنا عنه.

```
</script>
```

وبختم دوال الجافا سكربت يتم بعدها ختم المستند كاملا بوسوم الاغلاق المعروفة.

=====

تمرين:

قم بجمع كل الدوال السابقة في ملف *.js واحد واربطه بمستند html ثم قم باستدعاء جميع الامثلة اعلاه في صفحة واحدة؟

7. جافا سكريبت: المجموعة الثالثة من الامثلة:

المثال الاول:

مثال يوضح كيفية الحصول على الصورة وعرضها، عبر زر امر اختيار الملف:



استخدمنا في هذا المثال طريقة استعراض صورة وجلب مسارها، وهي مستخدمة عند تعبئة بيانات نموذج form وارسالها عبر بروتوكول http كطلب post لمعالجتها في المخدم، ولكننا في هذا المثال وضحنا كيفية استخدام المسار الذي تم تحديده عبر دالة في الجافا سكريبت وعرضها في عنصر عرض الصور التابع للغة الترميز التشعبي.

الكود التالي يوضح هذه الفكرة عبر زري امر احدهما يحمل الصورة عبر عرضها مربع حوار اختيار ملف واختيار صورة بواسطة المستخدم والعودة بمسارها، والثاني يعرضها في جسم الصفحة. لاحظ في دالة الجافا سكريبت ادناه، قمنا اولا بتعريف متغير يأخذ لنا العنصر الذي يجلب الصورة، وبعده عرفنا متغير اخر يقوم بدور عنصر الصورة الذي في جسم الصفحة.

بعدها سوف نعرض مربع رسالة للمستخدم نعرض فيه اسم الصورة، باستخدام الخاصية name التابعة لكائن الملفات التابع لعنصر فتح الملف.

ثم نستخدم عنصر الصورة لعرض الصورة عبر تمرير كائن الصورة files باعتبار ان الصورة المطلوبة هي اول صورة في الكائن، وهذا الكائن يحتوي على مصفوفة من الملفات بافتراض ان المستخدم اختار اكثر من ملف، ولاننا اخترنا ملف واحد فقد عرفنا ان الملف الذي اخترناه اول عنصر في مصفوفة كائن الملفات files.

بعد ذلك قمنا باستدعاء الدالة المذكورة عبر الحدث onclick المستخدم كخاصية لعنصر زر الامر . button

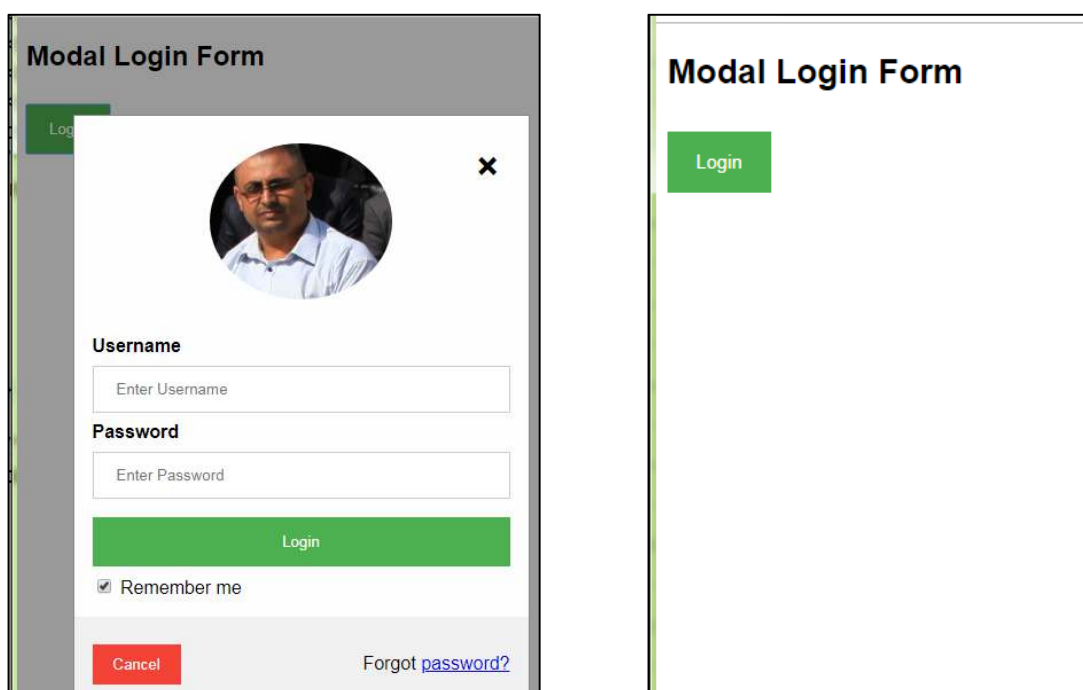
```
<html dir="rtl">
<head>
<script>
function showimg()
{
var x = document.getElementById("filet");
var y = document.getElementById("myimg");
window.alert(x.files[0].name);
y.src = window.URL.createObjectURL(x.files[0]);
}
</script>
</head>
<body>
<input id="filet" type="file" />
<input type="button" onclick=" showimg()" value=" اظهر" />
<p>
<img id="myimg" />
</p>
</body>
</html>
```

هناك مثال متكامل يستعرض باقي خصائص الكائن files ولا يحتاج لزر امر عرض الصورة، لانه يستخدم اسلوب الاستماع الى حدث تحميل الصورة، وبمجرد حدوث ذلك، يقوم بعرضها مع مجموعة من الخصائص التابعة لها، يمكنك دراسة المثال على الرابط التالي:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file>

المثال الثاني:

مثال يوضح كيفية تصميم نموذج تسجيل الدخول Login Form ، بحيث يعرض زر امرlogin، عندالضغط عليه يظهر للمستخدم نموذج ادخال بيانات تسجيل الدخول وهي اسم المستخدم وكلمة المرور ومربع تأشير يطلب (أو لا يطلب) تذكر المستخدم .



الكود التالي يعرض لنا استخداما للغات جهة العميل client-side الثلاث، فالمحتوى هو html5 كما تعودنا، ثمة كود لغة CSS فيه افكار تنسيقات جميلة، واخيرا مقطع نصي صغير بلغة JavaScript . وسنعرض فيما يلي الكود كاملا، ثم نناقش المقاطع المهمة التي قدرنا انها تحتاج الى توضيح.

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">


```
<style>
body {font-family: Arial, Helvetica, sans-serif;}
input[type=text], input[type=password] {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    box-sizing: border-box; }
button {
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%; }
button:hover {
    opacity: 0.8; }
.cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336; }
.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
    position: relative; }
```

```
img.avatar {
    width: 40%;
    border-radius: 50%; }

.container {
    padding: 16px; }

span.psw {
    float: right;
    padding-top: 16px;}

.modal {
    display: none; /* Hidden by default */
    position: fixed; /* Stay in place */
    z-index: 1; /* Sit on top */
    left: 0;
    top: 0;
    width: 100%; /* Full width */
    height: 100%; /* Full height */
    overflow: auto; /* Enable scroll if needed */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
    padding-top: 60px;}

.modal-content {
    background-color: #fefefe;
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and
centered */
    border: 1px solid #888;
    width: 80%; /* Could be more or less, depending on screen size */ }

.close {
```

```
position: absolute;
right: 25px;
top: 0;
color: #000;
font-size: 35px;
font-weight: bold;}
.close:hover, .close:focus {
  color: red;
  cursor: pointer; }
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s; }
@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)} }
@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)} }
@media screen and (max-width: 300px)
{  span.psw {
    display: block;
    float: none; }
  .cancelbtn {
    width: 100%; }
}
</style>
</head>
```

```

<body>
<h2>Modal Login Form</h2>
<button          onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Login</button>
<div id="id01" class="modal">
  <form class="modal-content animate" action="/action_page.php">
    <div class="imgcontainer">
      <span          onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
      
    </div>
    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname" required>
      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>
      <button type="submit">Login</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>
    <div class="container" style="background-color:#f1f1f1">
      <button          type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>

```

```

</form>
</div>
<script>
var modal = document.getElementById('id01');
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>
</body>
</html>

```

السطر التالي :

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

تحدد لنا انه يكون عرض محتوى صفحة المصفح، حسب عرض جهاز/ شاشة العرض.
اما هذا المقطع:

```
input[type=text], input[type=password] {.....
```

فيحدد خصائص عنصر الادخال بشرط اذا كان نوعه مربع نص او مدخلات كلمة مرور.
بينما يقوم المقطع التالي:

```
button {.....
```

بتحديد خصائص تنسيق جميع ازرار الامر التي ستظهر في الصفحة بدون تخصيص.
وعند مرور مؤشر الماوس يتم انقاص شفافية ازار الامر قليلا، كما يوضح هذا المقطع:

```
button:hover {
```

```
opacity: 0.8;
```

باستخدام كلاس خاص بالنموذج المعروض ، { modal. ، يتم تحديد خصائص مهمة فيه منها:
اخفاء تلقائي للنموذج عند فتح الصفحة:

```
display: none;
```

تثبيت الموقع حتى يظهر في نفس الموقع المحدد لها في كود html، بغض النظر عن عدم ظهوره.

```
position: fixed;
```

جعل ترتيب العرض له اعلى الجميع:

```
z-index: 1;
```

وغير ذلك من الخصائص المهمة.

أما المقطع التالي فيحدد موقع العنصر الذي يظهر لنا اشارة X لاغلاق النموذج:

```
.close {
  position: absolute;
  right: 25px;
  top: 0;
```

....

في المقطع السابق يكون الموضع مطلقا absolute ، وليس نسبيا relative ، ويبعد عن اليمين بمسافة بسيطة فقط 25 بيسكل، لكي يظهر في الركن الايمن للنموذج.

بعد ذلك كما في الكود التالي، تحديد خصائص الاشارة X عندما ينتقل التركيز focus لها يمر المستخدم بؤشر الماوس عليها hover .

```
.close:hover, .close:focus { .....
```

الكود التالي يحدد تنسيقات حركة النموذج عندما يتم عرضه، فعند النقر على زر الامر، سيظهر النموذج بطريقة Animation تحريك معين بسبب التنسيقات التالية.

فيما يلي تحديد زمن 0.6s كجزء من الثانية لعملية التحريك للعنصر ذي الكلاس المحدد:

```
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}
```

ولاحظ في الكود اعلاه ان التنسيق نفسه مذكور بخاصيتين واحدة مختصة بالمستعرضات -webkit- والخاصية الثانية باي مستعرض آخر، كالمستعرضات -moz- او -o- او غيرها.

المقطع التالي يحدد مدى التكبير في انتقال الحركة في حال تم تنفيذ الخاصية الاولى اعلاه، اي تلك المتعلقة بالمتصفحات -webkit- وذلك من اجل تجنب القيم الافتراضية التي تستطيع اختبارها اذا حذفت هذا المقطع من الكود.

```
@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)} }
```

والمقطع التالي هو نفس الكود السابق ولكن لاجل باقي المتصفحات.

```
@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)} }
```

في المقطع التالي يتم تغيير التنسيق للعنصرين المحددين، وذلك للتغيير الظهور في حال كان عرض الشاشة اقل 300px، اي الشاشات الصغيرة.

```
@media screen and (max-width: 300px)
```

```
{.....
```

وبهذا نكون قد انتهينا من التنسيقات المتتالية، وبعدها سوف يأتي محتوى لغة الترميز التشعبي ، فيها السطر التالي يحدد وظيف زر الامر الذي يعرض نموذج تسجيل الدخول بالشكل التالي:

```
<button onclick = " document. getElementById('id01').style .display = 'block' " style="width:auto;">Login</button>
```

وهو كود بلغة جافا سكربت يجعل الضغط على زر الامر المحدد، يغير الخاصية display للعنصر الحاوي لعناصر النموذج، وبتغييرها يتم عرض النموذج حيث ان القيمة المسبقة هي none فجعلته مخفيا.

الكود التالي يعرف حاوية صغيرة من نوع span تعرض رمز X في الركن الايمن لحاوية النموذج، وبالضغط عليها يتم اغلاق النموذج، كل هذا مكتوب بالسطر التالي:

```
<span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
```

ومن المهم ان نشير الى الكلمة المحجوزة required في عنصر تعريف مربعات النص، لكي تجعل ادخال البيانات اجباريا، لان العنصر مطلوب:

```
<input type="text" placeholder="Enter Username" name="uname" required>
```

وسنختم بالمقطع التالي:

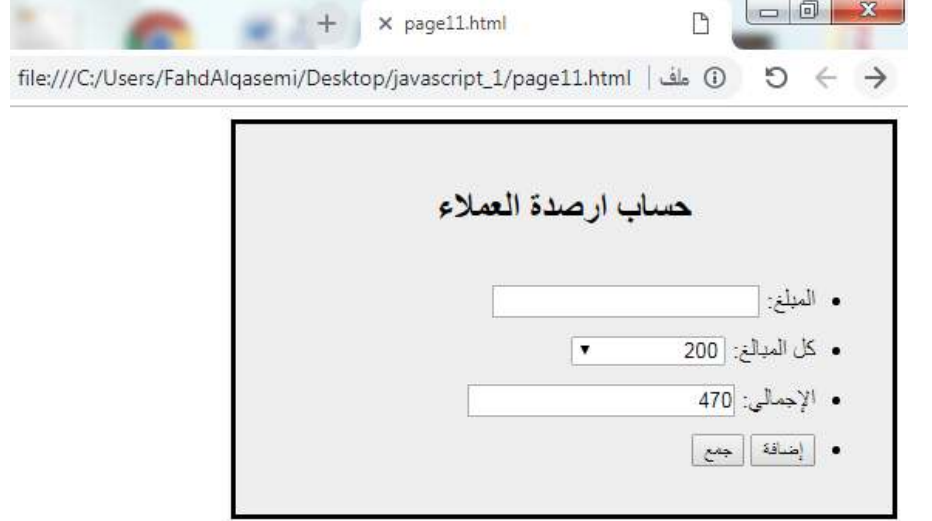
```
<script>
var modal = document.getElementById('id01');
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none"; } }
</script>
</body> </html>
```

وهو بالفعل اخر مقطع بكود الصفحة، ويقوم بتعريف مقطعا بلغة جافاسكربت، بحيث ينفذ في نهاية عرض الصفحة.

لاحظ معي انه لا توجد دالة محددة في السكريبت، لكي يتم استدعاؤها خارج العنصر <script>، ولكن يتم استخدام كائن نافذة المتصفح window حيث نختبر عبر هذا الكائن حدث النقر عليه onclick ، وفي حال تم النقر على نافذة المتصفح، يتم تنفيذ دالة بدون اسم، لان محتواها مكتوب مباشرة ومسند الى خاصية الحدث onclick.

هذه الدالة سوف تختبر حصول الحدث، وفي حال كان عنصر حاوية النموذج ظاهرا، العنصر الذي هويته هي id01، فإنه يتم تنفيذ سطر يستدعي خاصية جعل عرض العنصر تساوي لا شيء none ، من اجل اخفاء النموذج، والهدف من ذلك انه اذا نقر المستخدم على مكان في الصفحة خارج النموذج الظاهر، فإنه يتم اغلاق النموذج.

المثال الثالث: برنامج مبسط لحساب مجموع ارصدة العملاء



يقوم البرنامج وفق الشكل اعلاه بحساب مجاميع المبالغ التي يدخلها المستخدم، بحيث يتم اضافتها اولا إلى مربع القائمة المنسدلة، ثم يتم تجميعها ووضع الاجمالي في مربع نصي، وتقوم ازرار الامر الثلاثة ادناه بتنفيذ العمليات المكتوبة بلغة جافا سكريبت، كما هو موضح في كود الملف: page11.html، والذي يستدعي ملف الجافا سكريبت: account.js، المرفق ادناه، الكود التالي لملف ال HTML:

```
<!DOCTYPE html>
<html dir="rtl">
<head>
<script src=" account.js "> </script>
</head>
<body >
<section style="width:30%;padding:10px; line-height:2; background:#eee;
border:solid;">
```

```

<h2 style="text-align:center;"> حساب ارصدة العملاء </h2>
<ul>
<li> المبلغ: <input id="txt1" autofocus /> </li>
<li> كل المبالغ:
<select id="mySelect" style="width:100;"> </select>
</li>
<li> الإجمالي: <input id="txt2"/> </li>
<li>
<button onclick="addopt()"> إضافة </button>
<button onclick="sum()"> جمع </button>
<button onclick="clropt()"> محو البيانات </button>
</li>
</ul>
</section>
</body>
</html>

```

اما كود ملف الجافا سكربت account.js ، ففيه الدالتين التي تم استدعاؤهما اعلاه كالتالي:

```

var s = 0;
function addopt()
{
var x = document.getElementById("mySelect");
var txt = document.getElementById("txt1");
var option = document.createElement("option");
option.text = txt.value;
x.add(option);
var ops = Number(txt.value);
s += ops;

```

```
txt.value = "";  
}  
function sum()  
{  
var txt = document.getElementById("txt2");  
txt.value = s;  
}
```

شرح المقاطع المهمة:

نقل التركيز عند تحميل الصفحة الى العنصر المحدد:

```
<li> المبلغ: <input id="txt1" autofocus /> </li>
```

لاحظ العنصر select فارغا من محتوى ال option لانه سيتم انشاء ذلك في دالة الجافاسكربت، التي سيتم استدعاؤها في هذا السطر:

```
<button onclick="addopt()"> إضافة </button>
```

اما زر الامر هذا فخاص بعملية تجميع القيم المدخلة الى القائمة:

```
<button onclick="sum()"> جمع </button>
```

ملف الجافا سكربت يبدأ بتعريف عنصر عام، عرفناه خارج الدالتين لاننا سوف نحتاجه في كليهما:
var s = 0;

وقد تم استخدامه لتجميع القيم التي نضيفها كل مرة.

الدالة addopt() تقوم بالحصول على عنصر النص وعنصر القائمة ومن ثم تنشئ عنصر option كما في:

```
var option = document.createElement("option");
```

هذا العنصر سوف يأخذ القيمة في مربع النص كما في:

```
option.text = txt.value;
```

ثم بعد ذلك يتم اضافته إلى القائمة هكذا:

```
option.text = txt.value;
```

الخطوة التالية هي اخذ القيمة التي في مربع النص لتجميعها، وقد بدأنا بتعريف متغير يأخذ القيمة في مربع النص بعد تحويلها الى رقم:

```
var ops = Number(txt.value);
```

ثم سنقوم بتجميع المتغير، وبعدها تفرغ مربع النص من قيمته الحالية:

```
s += ops;
```

```
txt.value = "";
```

اما الدالة sum() فتحتوي على سطرين فقط احدهما يأخذ عنصر النص الثاني، الذي سيبطهر فيه المجموع:

```
var txt = document.getElementById("txt2");
```

والثاني تضع في عنصر مربع النص قيمة المتغير العام التي قمنا بتجميعها في الدالة السابقة:

```
txt.value = s;
```

=====

تمرين:

هل هناك من طريقة للحصول على البيانات المدخلة حتى بعد اعادة تحميل الصفحة؟؟؟
لاجابة هذا السؤال قمنا بتصميم الصفحة التالية، وافترضنا ان هناك طريقة لتخزين البيانات في المتصفح، بحيث يقوم المستخدم بادخال بيانات العميل مع المبلغ، ومن ثم يعرض اسماء العملاء الذين ادخلهم والمبالغ مقترنة بكل واحد، وبعدها يغلِق المتصفح ويفتحه مرة اخرى ويضغط على زر عرض ليتأكد ان البيانات لم تزل موجودة:

حساب ارصدة العملاء

- اسم العميل:
- مبلغ الدين:
- حفظ
- عرض اسماء العملاء ومديرتهم

المبلغ	العميل
4000	احمد
40000000	احمد محمود علي
3000	عندة نجاش
400	علي المراري
10000000	فانم مرعي
400000	فانم مسعود

مثال على التخزين المحلي في الجافاسكريبت

باستخدام الكائن: localStorage

برمجة: د. فهد آل قاسم

الواجهة بعد ادخال عدد من السجلات:

حساب ارصدة العملاء

اسم العميل:

مبلغ الدين:

عرض اسماء العملاء ومديونتهم

العميل	المبلغ	
ابوبكر سالم	140000	حذف
ايوب طارق	2000	حذف
عبد الحلیم حافظ	2100	حذف
علي عنيه	3000	حذف
كاظم الساهر	12500	حذف
محمد مرشد ناجي	3020	حذف

وعند النقر على رابط الحذف:

تعرض هذه الصفحة
تم حذف العميل عبدالحميم بنجاح

موافق

اسم العميل:

مبلغ الدين :

عرض

عرض اسماء العملاء ومديوناتهم

العميل	المبلغ	
ابوبكر سالم	140000	حذف
ايوب طارق	2000	حذف
عبد الحليم حافظ	2100	حذف
عبدالحميم	4032	حذف
علي عتيه	3000	حذف
كاظم الساهر	12500	حذف
محمد مرشد ناجي	3020	حذف

الكود ادناه غير مشروح بالتفصيل، وقد اعتمدت على انه سيكون بطبيعة الحال كودا مفهوما لمن قرأ وطبق اكواد هذا الكتيب منذ بدايته، ويتكون كود هذا المشروع المصغر من ملفين الملف الرئيسي وهو ملف *.html، والثاني ملف جافاسكربت *.js، ويحتوي كود ال html على اكواد التصميم للصفحة الظاهر اعلاه، اضافة الى اكواد ال CSS التي اعطتنا هذا المنظر المميز. ويحتوي كود الجافا سكربت على ثلاثة دوال مهمة، اولها دالة الحفظ ثم دالة العرض واخيرا دالة الحذف.

يجدر بالذكر الاشارة الى ان دالتي الحفظ والعرض تستخدمان الكائن localStorage وهو الكائن الخاص بحفظ البيانات واسترجاعها من والى المتصفح، وهي عملية شبيهة بالكوكيز، ظهرت مع html5 وهي مثل الكوكيز مرتبطة بالمتصفح الذي استخدمته حتى تقوم بحذف جميع المحفوظات.

اما دالة الحذف فتطبع جملة الحذف في العمود الثالث من كل صف، ثم تنتظر نقرة من المستخدم حتى اذا نقر المستخدم فحست اي صف هو لتقوم بالحذف حسب القيمة المعروضة فيه.

وفيما يلي كامل الكود للتطبيق والتطوير.

أولا: كود ملف ال html:

```
<html dir="rtl">
<head>
```

```
<style>
section
{
width:50%; padding:10px; margin: auto; line-height:2; background:#eee;
border: gray solid ;
}
a
{
font-weight:bold;
}
a:hover
{
text-decoration-line:underline;
color: blue;
cursor: pointer;
}
body {
width:60%; background:gray; margin: auto;
}
div
{
display: none;
}
button
{
font-size: large;
font-weight:normal;
width: 5em;
margin-left:30px;
margin-right:50px;
}
ul {
list-style-type:none;
```

```

}
table
{
border-collapse: collapse;
border: solid 1px;
text-align: center;
padding: 5px;
}
td, th
{
border: solid 1px;
width: 100px;
height: 20px;
}
</style>
<script src="account_12.js"> </script>
</head>
<body >
<section style="width:50%; height:5%; padding:10px; margin: auto; line-
height:2; background:#eee; ">
</section>
<section >
<h2 style="text-align:center;"> حساب ارصدة العملاء </h2>
<ul>
<li> <strong>اسم العميل</strong><input id="txt1" autofocus /> </li>
<li> <strong> مبلغ الدين : </strong> <input id="txt2" /> </li>
<p> </p>
<li>
<button onclick="save()"> حفظ </button>
<button onclick="show()"> عرض </button>
</li>
</ul>
<div id="datagrid">
عرض اسماء العملاء ومديونتهم

```



```

<table id="tbl">
</table>
</div>
</section>
</body>
</html>

```

ثانياً: كود الملف `account_12.js`:

```

var flag = 0;
function save()
{
    var txt1 = document.getElementById("txt1");
    var txt2 = document.getElementById("txt2");
    if (txt2.value == "") txt2.value = "0";
    localStorage.setItem(txt1.value, txt2.value);
    txt1.value = "";
    txt2.value = "";
    show();
    txt1.focus();
}

function show()
{
    dv = document.getElementById("datagrid");
    dv.style.display = "block";
    var tbl = document.getElementById("tbl");
    var str = "";

    str += "<tr><th >" + " العميل " + " </th><th>" + " المبلغ " + " </th><td></td></tr>";
    for (var i = 0; i < localStorage.length; i++)
    {
        var key = localStorage.key(i);
        var value = localStorage[key];
    }
}

```

```

    str += "<tr><td>" + key + " </td><td >" + value + "</td> <td><a >
حذف</a></td> </tr>";
}
tbl.innerHTML = str;
del();
}

```

```

function del() {
    var tble = document.getElementById("tble");
    var rindx, clindx ;
    for (var i = 0; i < tble.rows.length; i++) {
        for (var j = 0; j < tble.rows[i].cells.length; j++) {
            tble.rows[i].cells[j].onclick = function () {
                rindx = this.parentElement.rowIndex;
                clindx = this.cellIndex;
                if (clindx == 2) {
                    var skey = tble.rows[rindx].cells[0].textContent;
                    localStorage.removeItem(skey.trim());
                    alert(" بنجاح " + skey + " تم حذف العميل ");
                    flag = 1;
                    if (flag == 1)
                        show();
                }
            }
        }
    }
}

```

المراجع

1. Elizabeth Castro and Bruce Hyslop, HTML5 and CSS3, Seventh Edition: Visual QuickStart Guide, Seventh Edition.
2. سامي ربايعه، تعليم الجافا سكريبت، من النت.
3. موقع مدرسة w3: www.w3school.com
4. موقع كودبروجت الشهير: codeproject.com
5. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file>
6. مواقع متنوعة أخرى ع النت.

خاتمة

أتمنى ان يكون هذا الكتيب قد نال رضى المبرمج العربي، وقدم له الفائدة المرجوة، كما اشير الى ان ايميلي السابق، في الكتيبات السابقة، قد ضاع مني في زحمة مشاغلي وتنقلت في البريد الالكتروني الى الايميل الموضح أدناه، بمعنى اني مرحب بكل ملاحظة او تصويب او مناقشة او سؤال من جميع الدارسين والمهتمين بعالم الحاسبات والمعلومات الجميل.

نبذة عن المؤلف:



فهد احمد آل قاسم، من مواليد الجمهورية اليمنية، محافظة إب، بريد الكتروني: fhdahmd18@yahoo.com. دكتوراة في الحاسبات والمعلومات - جامعة المنوفية - مصر - 2018م. أستاذ مساعد في كلية الحاسبات وتكنولوجيا المعلومات - جامعة العلوم والتكنولوجيا اليمنية.

كتيباتي السابقة على موقع كتب وعدد من المواقع المشابهة هي:

1. مبادئ معالجة الصور باستخدام الماتلاب.
2. ملخص أساسيات تحليل وتصميم النظم.
3. مفاهيم في تكنولوجيا المعلومات المتقدمة.
4. مفاهيم قواعد البيانات العلائقية.
5. بالنمط المنفصل: مثال تطبيقي متكامل لبرمجة قواعد البيانات.
6. مبادئ قواعد البيانات الموزعة.
7. البرمجة الهدفية لغة سي ++
8. أمنية البيانات والتشفير عربي.
9. مدخل الى عالم الذكاء الاصطناعي.

ما بعد الخاتمة

- ثمة اعتذار جدير به كل من قرأ احد كتيباتي الرقمية البسيطة، وهو اعتذار عن الاخطاء، ولا يوجد بشر لا يخطأ، الاخطاء الطباعية غير المقصودية والاختفاء العلمية إن وجدت غير المقصودة ايضا، واي نوع من الاخطاء .. وكفى بالمرء نبلا انه بشر يصيب ويخطئ.
- في ختام الختام رجاء ان لا تغمطوا الناس حقوقهم الفكرية، فهذه الكتب مجانية النشر لكنها محفوظة الحق الفكري، وهذا شكر عاطر مني لكل من التزم بهذه المبادئ الاخلاقية وتحلى بها وما اكثرهم.