

تقنيات الانترنت المتقدمة

HTML

C++

المقدمة

HTML ليست إصدارًا جديدًا من لغة ترميز قديمة انما هي مفهوم جديد لبناء مواقع الويب والتطبيقات في عصر الأجهزة المحمولة والحوسبة السحابية والشبكات.

على مدى السنوات الماضية توصل المبرمجين و مصممي صفحات الويب من جميع انحاء العالم الى اضافات و طرق جديدة للتغلب على المشاكل التي واجهت اجهزة الحاسوب القديمة و برامجها منها لغة النصوص المركبة HTML والتي استخدمت الجافا سكريبت و ال CSS لتمكين المصممين من القيام بأكبر قدر ممكن من العمليات على صفحات الويب .

HTML5 هي في الواقع تحسين للhtml،تقترح HTML5 معايير لكل جانب من جوانب الويب وأيضًا غرضًا واضحًا لكل علامة .

توفر HTML العناصر الهيكلية لصفحة الويب،بينما تركز CSS على كيفية تحويل هذا الهيكل إلى شيء مرئي جذاب وقابل للاستخدام،و Javascript تستخدم لإجراء العمليات الحسابية و المنطقية اللازمة لعمل صفحات الويب بأفضل طريقة ممكنة .

الوحدة الأولى

مستندات HTML5

المكونات الأساسية

يوفر HTML5 ثلاث ميزات أساسية:

- الهيكل: باستخدام البنية و الأوامر النصية للـHTML
- أسلوب العرض (التنسيق): باستخدام الـCSS
- الوظائف و العمليات: باستخدام الجافا سكريبت

الهيكل العام لمستند HTML:

يتم تنظيم مستندات HTML بدقة. يتم تمييز كل جزء من المستند على حدى وذلك بارفاق علامة او tag .

أولاً ، نحتاج إلى تحديد نوع المستند الذي نقوم بإنشائه وهذا بسيط للغاية:

• **<!DOCTYPE html>**

استخدام عنصر <!DOCTYPE html>

لتعريف المستند بان النص المكتوب تحت العلامة هو نص html.

ملاحظة:

يجب أن يكون هذا السطر هو السطر الأول من ملفك ، دون أي مسافة أو سطر قبله، وذلك لتنشيط الوضع القياسي وإجبار المتصفحات على تفسير النص بأنه نص

.HTML5

طبعي:

يمكنك البدء في كتابة الكود الآن في ملف HTML الخاص بك وإضافة كل عنصر جديد ستتم دراسته لاحقاً.

• <html>

بعد الإعلان عن نوع المستند ، يتعين علينا بناء هيكل شجرة HTML. جذر هذه الشجرة هو عنصر <html> حيث يضم هذا العنصر كود HTML. يجب اغلاق أي علامة (tag) تم فتحها. العلامات: <html>... </html>.

ملاحظة:

HTML5 مرنة للغاية فيما يتعلق بالبنية والعناصر التي تستخدم في بنائها. يمكن تضمين عنصر <html> بدون أي سمات أو حتى تجاهله تماما.

• <head>

استخدام عنصر <head>:

ضمن علامات <head> سنحدد عنوان صفحة الويب الخاصة بنا ، يمكنك ادراج ترميز الأحرف وتقديم معلومات عامة حول المستند ودمج ملفات خارجية ذات أنماط أو نصوص أو حتى صور مطلوبة لعرض الصفحة.

• <body>

يعتبر جسم المستند هو النص الأساسي والجزء المرئي من منه ويتم تحديده بعلامة <body>. لم تتغير هذه العلامة مقارنة بالإصدارات السابقة من HTML:

```
<!DOCTYPE html>  
<head>  
</head>  
<body>  
</body>  
</html>
```

رأس مستند الـ HTML5

عناصر الـ html5 التي توضع عادة داخل رأس المستند:

• <meta>

توضع هذه العلامة في رأس المستند. هذه الخاصية هي احدى التغييرات التي طرأت على html وتحدد هذه العلامة تشفير أحرف المستند و كيف ستظهر على الشاشة.

```
<!DOCTYPE html>
<head>
<meta charset="utf-8">
</head>
<body>
</body>
</html>
```

ملاحظة:

يمكن تغيير utf-8 للترميز الذي تفضل استخدامه، كما يمكن اضافة علامات وصفية و كلمات رئيسية كما هو موضح في المثال التالي:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, Javascript">
</head>
<body>
</body>
</html>
```

ملاحظة:

نوع و وصف الـ <meta> لا يظهر على الشاشة انما يؤثر في باقي العناصر المكتوبة داخل المستند.

• <title>

علامة <title> ، هي علامة توضع داخل ال <head> و تحدد ببساطة عنوان المستند.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
</head>
<body>
</body>
</html>
```

النص الموجود بين علامات <title> هو عنوان المستند بأكمله.
عادةً ما يتم عرض هذا النص بواسطة المتصفحات في أعلى النافذة.

• <link>

عنصر آخر مهم في رأس المستند هو <link>.

هذا العنصر يستخدم لدمج الأنماط أو البرامج النصية أو الصور أو الرموز من الملفات الخارجية داخل المستند.

أحد الاستخدامات الأكثر شيوعًا لـ <link> هو دمج الأنماط التي تحتوي على ملف CSS خارجي:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
```

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

نحتاج إلى سمتين لتضمين ملف الأنماط:

rel و href.

السمة rel تعني العلاقة بين المستند والملف الذي ندرجه.

في هذه الحالة السمة rel هي التي تخبر المتصفح أن الملف mystyles.css هو ملف CSS به الأنماط المطلوبة لعرض الصفحة .

جسم مستند الـHTML5

هيكل جسم مستند html <body>:

بنية الجسم (الكود بين علامتي <body>) هو الجزء المرئي من المستند بمعنى انه يمثل محتوى صفحة الويب.

تقدم HTML أنواعاً مختلفة من الطرق لبناء المعلومات وتنظيمها.

تنظيم جسم الـHTML:

كان أحد العناصر الأولى المقدمة لهذا الغرض هو <table>.

• الجداول <table>:

سمحت الجداول للمصممين بترتيب البيانات والنصوص والصور والأدوات في صفوف وأعمدة مكونة من خلايا.

في الأيام الأولى للويب ، كانت الجداول ثورة ، وخطوة كبيرة إلى الأمام في تنسيق

المستند وتحسين تجربة المستخدمين، في وقت لاحق ، تدريجياً ، عناصر أخرى

استبدلت وظيفة الجداول ، مما يوفر طريقة مختلفة للقيام بنفس الشيء بأقل

عدد من الأوامر النصية وأسرع وقت مما يسهل انشاء صفحات الويب وإمكانية نقلها و صيانتها.

• التقسيم <div>:

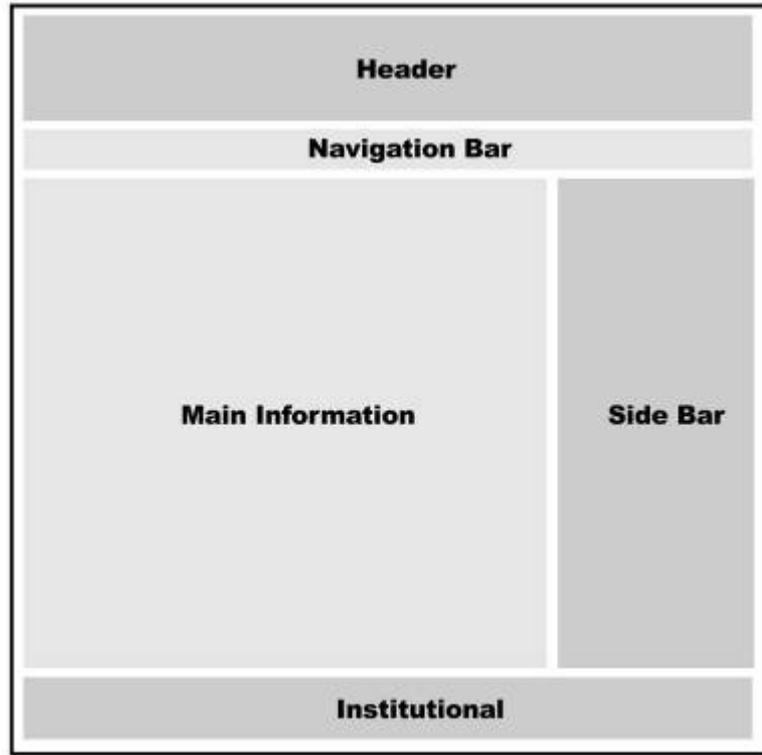
في السنوات الأخيرة بدأ عنصر <div> بالسيطرة على مجال تنظيم و تقسيم صفحات الويب.

الكلمة الأساسية div فقط تحدد انقساماً في الجسم ، مثل الخلية الموجودة في الجدول ، ولكنه لا يوجب تصنيف الاقسام وشرح ما بداخلها بخلاف الجداول.

الطريقة القياسية لتنظيم صفحات الويب:

يمثل الشكل ١-١ تخطيطاً منتظماً موجوداً في معظم مواقع الويب في الوقت الحالي. على الرغم من حقيقة أن كل مصمم يستطيع تصميم صفحته بطريقة الخاصة إلا أن هناك طريقة قياسية عادة ما يتم استخدامها.

بشكل عام سنكون قادرين على تقسيم صفحات الويب للأقسام التالية:



الشكل ١-١: تمثيل مرئي لتخطيط نموذجي لصفحة الويب

• الرأس:

الجزء العلوي ، الموصوف بالعنوان ، هو المكان الذي عادة ما يكون فيه شعارك واسمك ، الترجمة ، والأوصاف القصيرة لصفحتك على الويب.

• شريط التنقل:

أدناه ، يمكنك رؤية شريط التنقل الذي يقدم فيه كل مطور تقريباً قائمة اختيارية لتوجيهك داخل موقع الويب.

يتم توجيه المستخدمين من هذا الشريط إلى صفحات مختلفة عادة في نفس الموقع.

يتم وضع المحتوى الأكثر صلة بالصفحة في منتصف الصفحة و يقدم معلومات و روابط مهمة يتم تقسيمها إلى عدة صفوف وأعمدة.

• المحتوى الرئيسي و جانب الصفحة:

في مثال الشكل ١-١ يمكنك رؤية عمودين احدهما للمعلومات الرئيسية و الاخر هو الشريط الجانبي ، ولكن هذا القسم مرن للغاية و عادةً ما يقوم المصممون بتكليفه وفقاً لاحتياجاتهم عن طريق إدخال المزيد من الصفوف و تقسيم الأعمدة إلى كتل أصغر أو إنشاء مجموعات و توزيعات مختلفة.

المحتوى المقدم في هذا الجزء من التخطيط عادة ما يكون أولوية قصوى. في نموذج التخطيط ، يمكن أن تحتوي المعلومات الرئيسية على قائمة بالمقالات أو أوصاف المنتج أو إدخالات المدونة أو معلومات مهمة أخرى ، ويمكن أن يعرض الشريط الجانبي قائمة بالارتباطات التي تشير إلى كل من هذه العناصر.

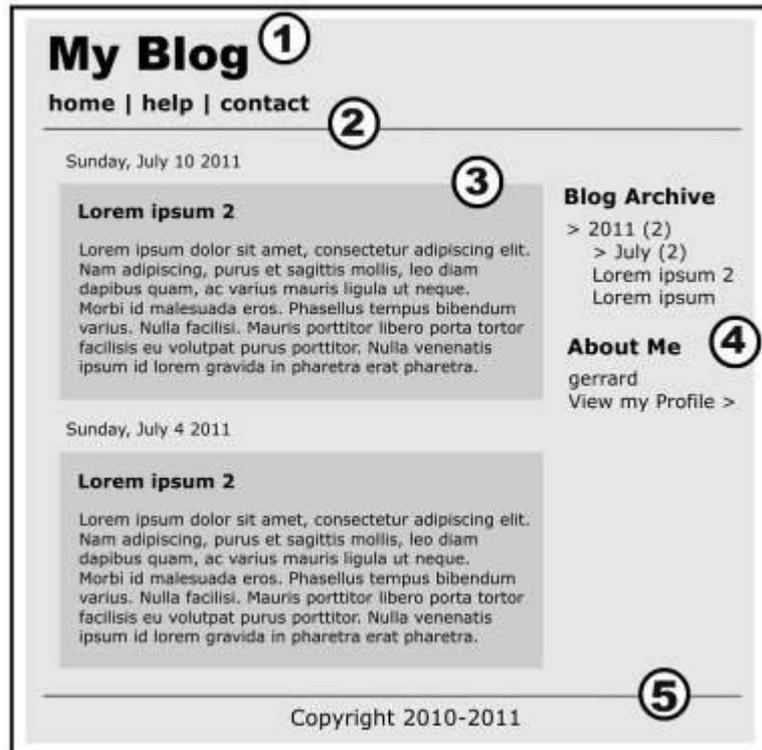
في مدونة ، على سبيل المثال ، سيقدم هذا العمود الأخير قائمة بالارتباطات التي تشير إلى كل التدوينات التي تمكت كتابتها ، معلومات عن المؤلف ، إلخ.

• التذييل و الشريط المؤسسي:

في الجزء السفلي من التصميم النموذجي لدينا شريط آخر يسمى شريط المؤسسات.

و نسميها بذلك لأن هذه المنطقة تحتوي عادة على معلومات عامة حول موقع الويب أو المؤلف أو الشركة ، بالإضافة إلى الروابط المتعلقة بالقواعد والشروط والبيانات الإضافية التي يعتبرها المطور تستحق المشاركة.

الشريط المؤسسي هو مكمل للرأس ، وهو جزء مما يعتبر هذه الايام هيكل أساسي لصفحة الويب.



الشكل ١-٢: تمثيل مرئي لتخطيط نموذجي للمدونة

الشكل ١-٢ مثال على مدونة ذات تخطيط نموذجي في هذا المثال ، يمكنك بسهولة تحديد كل

جزء من التصميم الذي كنا نفكر فيه من قبل.

١- رأس

٢- شريط التنقل

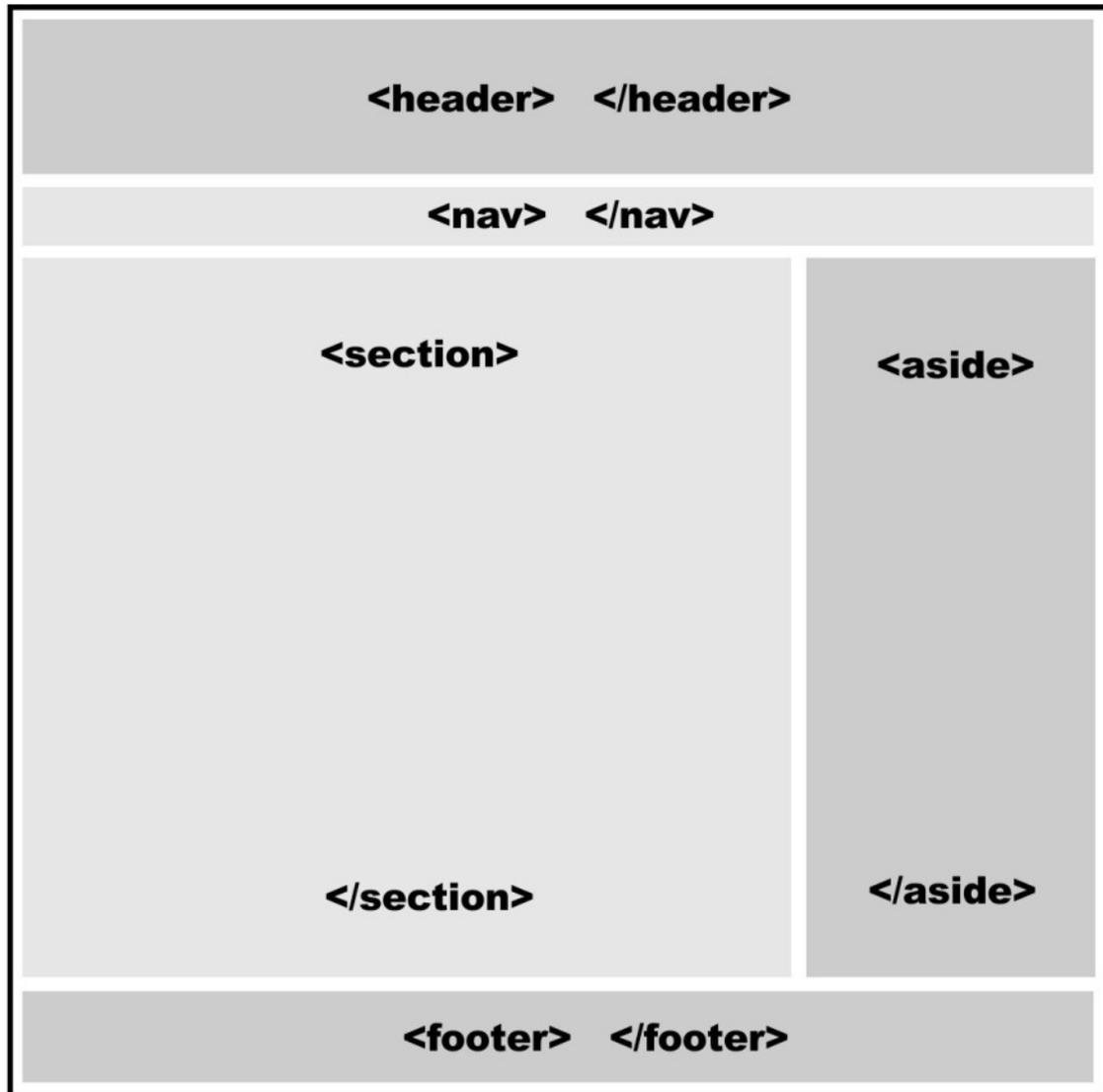
٣- قسم المعلومات الرئيسية

٤- جانب الصفحة

٥- التذييل أو الشريط المؤسسي

يمكن أن يساعدنا هذا التمثيل البسيط للمدونة في فهم أن كل قسم محدد موقع الويب له غرض. أحياناً لا يبدو هذا الغرض واضحاً ولكنه موجود ، وستكون قادرًا على رؤية الأقسام الموضحة أعلاه في كل موقع أو صفحة ويب .

يأخذ HTML5 في الاعتبار هذا الهيكل الأساسي والتخطيط ، ويوفر عناصر جديدة لصفحة الويب
ويمكننا من تقسيم صفحات الويب لأقسام واضحة:



الشكل ١-٣: تمثيل مرئي لترتيب القسم باستخدام علامات HTML5

يوضح الشكل ١-٣ التخطيط النموذجي الذي استخدمناه من قبل ، ولكن هذه المرة في صفحة ويب
معدة باستخدام HTML5 .

• الرأس <header>:

أحد العناصر الجديدة المدمجة في HTML5 هو <header>. يجب ألا يتم الخلط بين <header> وبين علامة <head> المستخدمة من قبل لبناء رأس المستند.

بنفس الطريقة التي يعمل بها <head> ، فإن الغرض من <header> هو تقديم المعلومات (مثل العناوين أو العناوين الفرعية أو الشعارات) ، لكن العلامتين تختلفان في النطاق (مكان الاستخدام).

توضع علامة <head> في بداية المستند و الغرض منها هو توفير معلومات حول المستند بأكمله، بينما توضع علامة <header> داخل الجسم والغرض منها هو توفير معلومات عن جسم أو محتوى الصفحة.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
</body>
</html>
```

طبعي:

إذا اتبعت الإرشادات من بداية هذا الفصل ، فيجب أن يكون لديك بالفعل ملف نصي يحتوي على الرموز التي تمت دراستها حتى الآن وجاهزًا للاختبار. أو فقومي بنسخ الكود الموجود في القائمة ١٠-١ في ملف نصي فارغ باستخدام أي محرر نصوص (مثل Windows 'Notepad) واحفظي الملف باسم من اختيارك وامتداد .html. لمشاهدة البرنامج النصي يعمل ، افتح الملف من ملف متصفح متوافق مع HTML5. يمكنك القيام بذلك من داخل المتصفح باستخدام الملف القائمة أو ببساطة انقر نقرًا مزدوجًا فوق الملف من مستكشف الملفات.

ملاحظة:

بين علامات <header> توجد عناصر ربما لا تعرفين بوجودها كالعنصر <h1> هو عنصر HTML قديم يستخدم لوضع عنوان.

يشير الرقم إلى أهمية العنوان ومحتواه. العنصر <h1> هو الأعلى من حيث الأهمية وله أكبر حجم خط ثم <h2> يليه <h3> حتى <h6> و هو الأقل أهمية ، لذلك سيتم استخدام <h1> لإظهار العنوان الرئيسي وسيتم استخدام علامات العنوان المتبقية للعناوين الفرعية أو ترجمة العناوين. سنرى كيف تعمل هذه العناصر في HTML5 لاحقاً.

• شريط التنقل <nav>:

القسم التالي هو شريط التنقل. يتم إنشاء هذا الشريط في HTML5

بعلامة <nav>.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
</body>
</html>
```

استخدام عنصر `<nav>`: داخل الجسم (علامة `<body>`) و لكنه يأتي بعد اغلاق الرأس (`</header>`) هذا بسبب ان شريط التنقل `<nav>` ليس جزءًا من الرأس ، ولكنه قسم جديد.

البنية والترتيب الذي نختار استخدامه مع HTML5 متروك لنا.

هذا يعني أن HTML5 متعدد الاستخدامات للغاية ، حيث أنه يعطي المصمم حرية تصميم موقعه أو صفحته مع امداده بكافة العناصر التي يحتاجها لذلك، ولكن كيفية استخدام هذه العناصر هو خيار متروك للمصمم.

مثال على ذلك هو أنه يمكن إدراج علامة `<nav>` داخل عنصر `<header>` أو في أي عنصر قسم آخر من جسم الـ.html.

ومع ذلك نوصي باتباع المعايير وإبقاء صفحة الويب واضحة قدر الإمكان.

الغرض من العنصر `<nav>`:

الغرض من العنصر `<nav>` هو القيام بإدراج خيارات بالصفحات التي سيريد المستخدم التنقل إليها داخل موقع أو صفحة الويب الخاصة بنا.

بعد علامة `<nav>` ، هناك عنصران يتم استخدامهما لإنشاء ملف قائمة هما:

- `` وتستخدم لفتح قائمة بعناصر يمكن الانتقال لها ويجب اغلاقها عند الانتهاء من سرد العناصر.

- `` و تستخدم لتعريف كل عنصر أو صفحة يراد الانتقال لها حيث انه لكل خيار في القائمة يجب فتح `` و اغلاقه `` على حده. كما هو موضح في المثال في السابق.

المحتوى الرئيسي `<section>`:

القسم التالي في التصميم القياسي هو ما نسميه شريط المعلومات الرئيسي و الذي يحتوي كما وضعنا من قبل على المعلومات التي يراد من الموقع أو الصفحة عرضها.

مثال:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
</section>
</body>
</html>
```

استخدام عنصر `<section>` مثل شريط التنقل ، يعد شريط المعلومات الرئيسي قسمًا منفصلاً. لذلك فإنه يوضع أسفل علامة الإغلاق `<nav/>`.

طبقي:

طبقي المثال السابق و قارني بين المخرج و تخطيط الشكل ٣-١ .

• جانب الصفحة <aside>:

في تخطيط موقع الويب النموذجي (الشكل ١-١) ، يوجد شريط يسمى الشريط الجانبي بجانب قسم المحتوى الرئيسي. هذا العمود أو القسم يحتوي عادةً على بيانات مرتبطة بالمحتوى الرئيسي ولكنها ليست بتلك الأهمية.

في مثال تخطيط المدونة القياسي (الشكل ٢-١) ، يحتوي الشريط الجانبي على قائمة بالروابط (رقم ٤) ، حيث كانت الروابط تشير إلى كل تدوينة تمك كتابتها مسبقاً ومعلومات إضافية عن مؤلف المدونة. في HTML5 ، يمكننا التفريق بين هذا النوع الثانوي من المعلومات بامتداد عنصر <aside>.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
</section>
<aside>
<blockquote>Article number one</blockquote>
<blockquote>Article number two</blockquote>
</aside>
</body>
</html>
```

استخدام عنصر <aside>:

يمكن وضع العنصر <aside> في الجانب الأيمن أو الأيسر في نموذج الصفحة لدينا حيث أنه ليس له موضع محدد مسبقاً.

قد يكون العنصر <aside> موجودة في أي جزء من التخطيط .

على سبيل المثال ، يمكننا استخدام عنصر <aside> داخل ملف <section> ، أو حتى ضمن المحتوى الرئيسي اذا اردنا وضع رابط لنص مرجعي فيه .

• التذييل <footer>:

لإنهاء بناء قالب أو البنية الأولية لمستند HTML5 ، نحن بحاجة إلى إغلاق التصميم وإنهاء نص المستند. html5 يوفر عنصرًا محددًا لهذا الغرض يسمى <footer>:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
</section>
<aside>
<blockquote>Article number one</blockquote>
```

```
<blockquote>Article number two</blockquote>
</aside>
<footer>
Copyright &copy; 2010-2011
</footer>
</body>
</html>
```

استخدام عنصر <footer>:

في تخطيط صفحة الويب النموذجي (الشكل ١-١) ، الشريط المؤسسي سيتم تعريفه بعلامات <footer>. ويشيع استخدام هذا الجزء من صفحة الويب لمشاركة الملفات العامة معلومات حول المؤلف أو الشركة التي تقف وراء المشروع ، مثل حقوق النشر والشروط ، إلخ.

عادةً ما يمثل عنصر <footer> نهاية نص المستند ، ويمكن استخدامه عدة مرات داخل الجسم لتمثيل نهاية أقسام مختلفة .

خصائص إضافية لجسم الـ HTML

تم الانتهاء من دراسة الهيكل الأساسي للموقع ، لكن لا يزال علينا العمل على المحتوى.

العناصر المستخدمة لإضافة محتوى للصفحة:

• <article> أو المقالة:

التخطيط في (الشكل ١-١) هو الهيكل الأكثر شيوعًا والأساسي لمواقع الويب على الإنترنت هذه الأيام ، ولكنه يمثل أيضًا مدى أهمية المحتوى الذي يظهر على الشاشة.

و نستخدم <article > لإضافة المحتوى الرئيسي للصفحة.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
<article>
This is the text of my first post
</article>
<article>
This is the text of my second post
</article>
</section>
```

```
<aside>
<blockquote>Article number one</blockquote>
<blockquote>Article number two</blockquote>
</aside>
<footer>
Copyright &copy; 2010-2011
</footer>
</body>

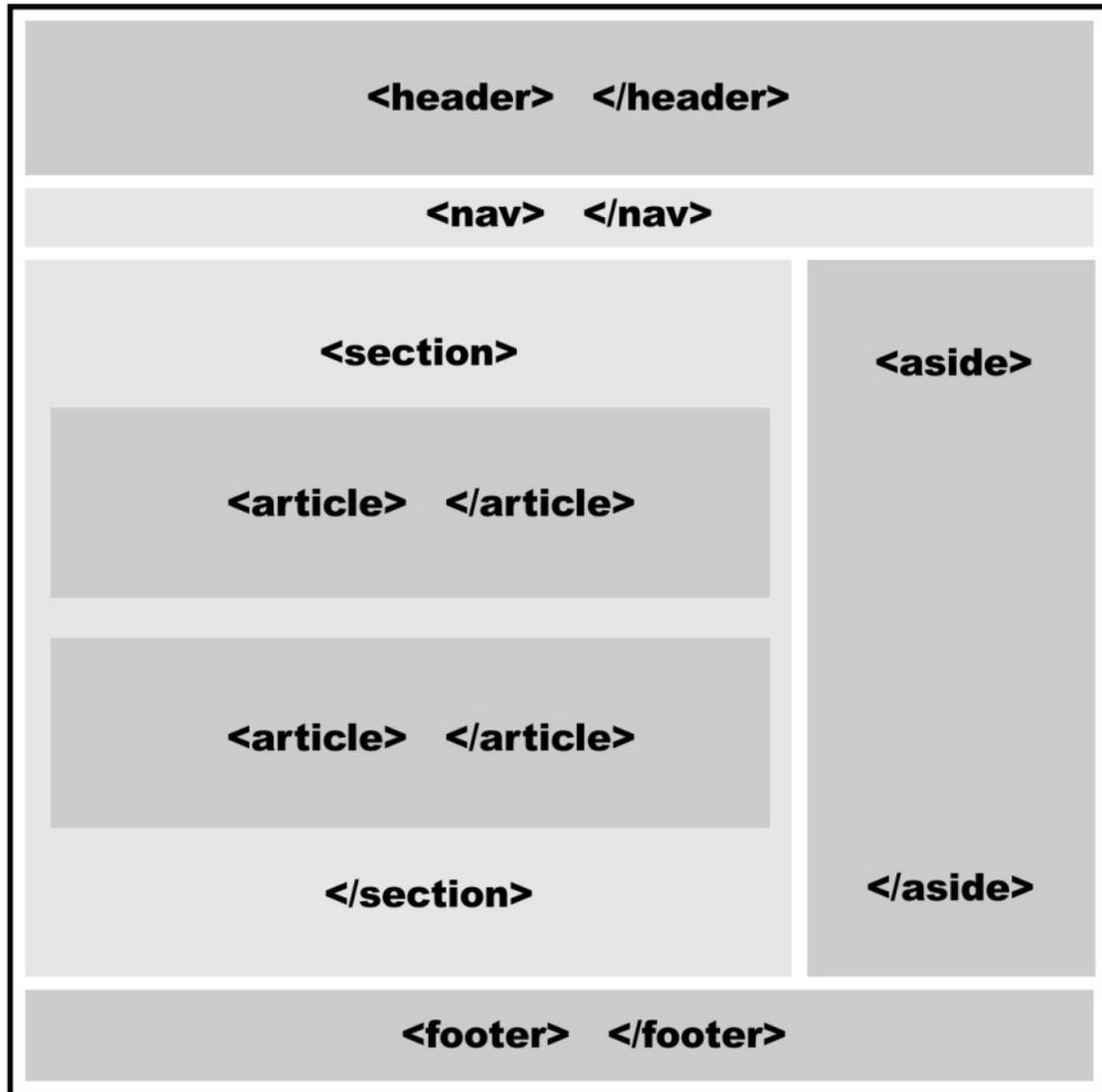
</html>
```

استخدام عنصر <article>:

كما ترى في الكود فإن علامات <article> توضع بين علامات فتح و اغلاق <section>.

راجع الأساسيات: كما قلنا من قبل ، يمكن وصف بنية HTML على أنها شجرة ،

مع عنصر <html> باعتباره جذره. طريقة أخرى لوصف العلاقات بين العناصر هو تسميتها كأبوين أو أطفال أو أشقاء وفقاً لموقفهم في هيكل الشجرة. على سبيل المثال ، في مستند HTML نموذجي يكون العنصر <body> ابن لعنصر <html> وشقيق عنصر <head>.



الشكل ١-٤: تمثيل مرئي لعلامات `<article>` داخل القسم الذي تم إنشاؤه للاحتفاظ به

المعلومات ذات الصلة لصفحة الويب

عنصر `<article>` او «مقالة» غير مقيد باسمه — لذلك لا يقتصر على المقالات الإخبارية بل يمكن أن تتضمن مشاركة في المنتدى ، أو مقالة في مجلة ، أو إدخال مدونة ، أو تعليق المستخدم ، وما إلى ذلك.

محتوى كل عنصر `<article>` سيكون له هيكل مستقل خاص به. لتحديد هذا الهيكل يمكننا الاستفادة من تعدد استخدامات علامتي `<header>` و `<footer>` التي تمت دراستها من قبل. حيث انه يمكن استخدام هذه العلامات ليس فقط في الجسم ، ولكن أيضًا في كل قسم من المستند.

مثال:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
<article>
<header>
<h1>Title of post One</h1>
</header>
This is the text of my first post
<footer>
<p>comments (0)</p>
</footer>
</article>
<article>
<header>
<h1>Title of post Two</h1>
</header>
This is the text of my second post
<footer>
<p>comments (0)</p>
</footer>
</article>
```

```

</section>
<aside>
<blockquote>Article number one</blockquote>
<blockquote>Article number two</blockquote>
</aside>
<footer>
Copyright &copy; 2010-2011
</footer>
</body>
</html>

```

• <hgroup> او مجموعة الرؤوس :

عنصر <header> يمكن يحتوي أكثر من عنوان و لتحقيق ذلك يمكننا الاستفادة من باقي العلامات <h1> <h2> <h3> ، <h4> و <h5> و <h6>.

ولكن لأغراض المعالجة الداخلية ، ولتجنب إنشاء ملفات متعددة اثناء قراءة صفحة الويب للمستند، يجب أن تجمع هذه العلامات معا.

للقيام بهذا HTML5 ، يوفر عنصر <hgroup>.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>

```

```
</nav>
<section>
<article>
<header>
<hgroup>
<h1>Title of post One</h1>
<h2>subtitle of the post One</h2>
</hgroup>
<p>posted 12-10-2011</p>
</header>
This is the text of my first post
<footer>
<p>comments (0)</p>
</footer>
</article>
<article>
<header>
<hgroup>
<h1>Title of post Two</h1>
<h2>subtitle of the post Two</h2>
</hgroup>
<p>posted 12-15-2011</p>
</header>
This is the text of my second post
<footer>
<p>comments (0)</p>
</footer>
</article>
</section>
<aside>
<blockquote>Article number one</blockquote>
<blockquote>Article number two</blockquote>
</aside>
<footer>
Copyright &copy; 2010-2011
</footer>
</body>
</html>
```

• <figure> and <figcaption> صورة أو شرح صورة

استخدام عنصري <figure> و <figcaption> لادراج صورة (

.(<”src = “http://minkbooks.com/content/myimage.jpg

تسمح لنا العلامات <figure> بارفاق الصور او مقاطع الفيديو ،و يتم ارفاق المعلومات المرتبطة بالصور أو مقاطع الفيديو على شكل نص قصير أدنى الصورة باستخدام علامات <figcaption>.

مثال:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="This is an HTML5 example">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>This text is the title of the document</title>
<link rel="stylesheet" href="mystyles.css">
</head>
<body>
<header>
<h1>This is the main title of the website</h1>
</header>
<nav>
<ul>
<li>home</li>
<li>photos</li>
<li>videos</li>
<li>contact</li>
</ul>
</nav>
<section>
<article>
<header>
<hgroup>
<h1>Title of post One</h1>
<h2>subtitle of the post One</h2>
</hgroup>
<p>posted 12-10-2011</p>
```

```
</header>
This is the text of my first post
<figure>

<figcaption>
This is the image of the first post
</figcaption>
</figure>
<footer>
<p>comments (0)</p>
</footer>
</article>
<article>
<header>
<hgroup>
<h1>Title of post Two</h1>
<h2>subtitle of the post Two</h2>
</hgroup>
<p>posted 12-15-2011</p>
</header>
This is the text of my second post
<footer>
<p>comments (0)</p>
</footer>
</article>
</section>
<aside>
<blockquote>Article number one</blockquote>
<blockquote>Article number two</blockquote>
</aside>
<footer>
Copyright &copy; 2010-2011
</footer>
</body>
</html>
```

الوحدة الثانية

تنسيق صفحات الويب باستخدام الـ CSS

CSS3

صفحات التنسيق المتتالية (cascading style sheet) أو css هي نصوص جاهزة تتم اضافتها لمستندات الHTML لتنسيق خلفياتها، وضع تأثيرات على النصوص ، التحكم بالحدود و غيرها من خيارات التنسيق المختلفة.

اغراض استخدام CSS:

- ١- تستخدم لتحديد طريقة عرض العناصر بشكل منفصل عن هيكلها.
- ٢- تستخدم لتنسيق و تقديم المعلومات.
- ٣-تستخدم للتحكم بمظهر موقع الويب.

مدقق CSS:

<http://jigsaw.w3.org/css-validator/>

خصائص الـcss3:

- ١- تُستخدم للتحكم في كيفية عرض العناصر في صفحة الويب.
- ٢- تستخدم مع صيغ مختلفة مثل HTML / XHTML.
- ٣- تستخدم في المتصفحات المرئية الشائعة (Internet Explorer و FireFox و Opera).
- ٤- عند استخدامها بشكل صحيح ، يمكن أن تبسط التصميم المرئي وإدارة الموقع وصيانة المحتوى بشكل كبير.

استخدام الـ CSS:

يمكن تعيين الأنماط في ورقة أنماط أو في عنصر نمط في الرأس أو في سمة نمط

```
/* melville.css style sheet */
body { font-family: verdana, helvetica, sans-serif;
      background: #f1f1f1;
    }
```

← Stylesheet

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>Moby Dick, or, The Whale</title>
    <link rel="stylesheet" href="melville.css"
        type="text/css" />
    <style type="text/css">
      <!--
        h1 { font-style: italic; }
      -->
    </style>
  </head>
  <body>
    <div style="text-align: center;
      width: 70%; border: solid black 1px;">
      <h2>About "Moby Dick, or, The Whale"</h2>
      <p>This story of a obsessive sea captain's quest to
      <span style="color: red;">find and destroy</span>
        an extraordinary whale that had taken his leg in a
        previous encounter is sometimes seen as an allegory
        for all of humanity's obsessive pursuits.
      </p>
    </div>
    <h1>Moby Dick, or, The Whale</h1>
    <p>Call me Ishmael. Some years ago -
      never mind how long precisely - having
      little or no money in my purse, and nothing
      particular to interest me on shore, I thought
      I would sail about a little and see the watery
      part of the world.</p>
  </body>
</html>
```

← Style element

← Style attribute

ما هو المقصود بالتصميم أو التنسيق :

"التنسيق" هو أمر تقوم بتعيينه لإخبار المتصفح بالشكل الذي يجب أن يبدو عليه قسم معين من صفحة الويب الخاصة بك. يمكنك استخدام التنسيق على العديد من "عناصر" HTML (مثل `<p>` `<table>` `<h1>` إلخ).

قواعد التنسيق:

إضافة تنسيق لعنصر معين ينقسم لجزأين هما:

- المحدد (selector): عنصر الـ html الذي تريد تنسيقه.
- التعريف (declaration): نص تنسيق العنصر و الذي يتكون من صفة (property) و قيمة (value).

Selector { **declaration**; }

Declaration = { **property**: **value**; }

الصفة: الصفة التي تريد تغييرها مثل الحجم الخط اللون
القيمة : ضبط الصفة مثل لون أحمر خط بحجم ٢٠
مثل:

Selector {property: value;}

h1 {color: red;}

تغيير لون الرأس ١ للون الأحمر.

```
p {color: blue;}
```

This is the text in this paragraph.

لون خط الفقرة ازرق.

```
a {color:red;
background-color: yellow;}
```

This is my link

```
Selector {property: value;}
```

```
h1 {color: blue; background-color: green;}
```

لون خط الرأس ازرق بينما لون الخلفية اخضر

أنواع صفحات التنسيق:

- ١- تنسيقات مضمنة .
- ٢- صفحة تنسيق داخلية .
- ٣- صفحة تنسيق خارجية.

مثال على التنسيقات المضمنة :

```
<html>
<head><title>My Wonderful Example</title>
</head>
<body>
  <p>What was I thinking?</p>
</body>
</html>
```

الكود الاصيلي

تضمين التنسيق :

```
<body>
  <p style="text-align: center; font-weight: bold; color:
  blue;">What was I thinking?</p>
</body>
```

What was I thinking?

في المنتصف ، عريض، ازرق.

صفحة تنسيق داخلية :

```
<head><title>My Wonderful Example</title>
<style type="text/css">
body {
text-align: left;
font-family: trebuchet, verdana;}
</style>
</head>
```

نعرف نوع التنسيق قبل جسم المستند، ثم نحدد التنسيقات داخل الجسم.

تنسيقات الخلفية:

يمكن اعطاء خلفية لصفحة ال html اما بلون او صورة باستخدام الcss.

الوحدة الثالثة

واجهات المستخدم

واجهة المستخدم

إن واجهة المستخدم في مجال التصميم الصناعي لتفاعل الإنسان مع الآلة، هي مجموعة من الوسائل التي يتفاعل بها الأشخاص (المستخدمون) مع منتج أو خدمة رقمية كإستخدام آلة، أو جهاز، أو برنامج حاسوبي، أو أي أداة معقدة أخرى (النظام). ويتم الوصول إلى هذه الوسائل بعدة طرق مثل الأزرار والرموز.

هدف هذا التفاعل هو السماح بعمليات فعالة والتحكم الكبير للآلات من قبل الإنسان في الوقت الذي تقوم به الآلة بتزويد معلومات تساعد المشغل على عملية اتخاذ القرار. تتضمن الأمثلة على هذا المفهوم الواسع النطاق لواجهات المستخدم مظاهر التفاعل في أنظمة تشغيل الحاسوب وأدوات التحكم باليد وأدوات التحكم في المعدات الثقيلة.

مع انتشار الحواسيب الشخصية وتراجع معرفة وخبرة المجتمع بالمعدات الثقيلة، فمصطلح واجهة المستخدم يشير عادة إلى واجهة المستخدم الرسومية، بينما لوائح التحكم الصناعية ونقاشات تصميم تحكم الآلات يشار إليهم غالباً بواجهات إنسان_آلة.

مصطلحات أخرى لواجهة المستخدم تشمل:

واجهة إنسان_حاسوب وواجهة رجل-آلة (MMI)

العمليات على واجهة المستخدم :

- الإدخال: تسمح للمستخدمين بالتعامل مع النظام.
- الإخراج: تسمح للنظام بالتجاوب مع تعاملات المستخدم.

أهمية واجهة المستخدم:

- لواجهة المستخدم دوراً كبيراً في توجيه المستخدم خلال تعامله مع أنظمة الحاسوب المختلفة.
- تقوم واجهة المستخدم بتوجيه المستخدم في إتمام الوظائف المختلفة. والواجهة الجيدة هي التي من خلالها لا يبذل المستخدم الكثير من الوقت والجهد في استخدامها.
- تسهل واجهات المستخدم التفاعل مع البرامج. في واجهة المستخدم الرسومية يكون الاعتماد فيها على الرموز والنقر بعكس واجهة الأوامر النصية التي تعتمد على إدخال الأوامر وبذلك تساعد واجهات المستخدم على إدخال البيانات أو الأوامر المطلوبة.

تاريخ موجز لواجهة المستخدم (UI):

لم تكن الواجهات الرسومية موجودة تجارياً، ولكن إذا أردت استخدام جهاز كمبيوتر عليك كتابة أسطر البرمجة. وعليك كتابة سطور غير محدودة إذا ما أردت التواصل مع مستخدمين آخرين وعبر لغات البرمجة وذلك لإنجاز مهمة بسيطة.

تم تطوير أول واجهة مستخدم رسومية (GUI) من قِبَل علماء الكمبيوتر في شركة Xerox PARC بحلول الثمانينات من القرن الماضي. وبذلك يستطيع المستخدمون إرسال الأوامر بصرياً عبر الأزرار والرموز والقوائم للتفاعل مع أجهزة الكمبيوتر الشخصية الخاصة بهم. وكان أول كمبيوتر منزلي ناجح تجارياً يستخدم هذا النوع من الواجهات هو جهاز Macintosh الذي أصدرته Apple Computer في عام ١٩٨٤. أصبحت تكنولوجيا تصميم الواجهات تكنولوجيا متنامية، وتطور دور مصمم واجهة المستخدم وهذا ما فتح المجال للمصممين لإظهار إبداعهم.

في عصرنا الحالي، أصبحت أجهزة الكمبيوتر جزءاً مهماً من الحياة اليومية وهذا ما يبين أهمية الاستمرار في تطوير الواجهات لجعل استخدام المنتجات الرقمية فعالاً.

المهام الأساسية لمصمم واجهة المستخدم (UI):

- الشكل والشعور
- بحوث التصميم.
- تطوير العلامة التجارية والرسوم البيانية.
- دليل الاستخدام.
- الاستجابة والتفاعل.
- تجاوب التصميم مع الأجهزة المختلفة.
- نماذج واجهة المستخدم.
- التفاعل مع الرسوم المتحركة.

أنواع واجهات المستخدم:

مثال لواجهة المستخدم الرسومية (GUI)

هناك عدة أنواع لواجهات المستخدم، ولكن أبرزها نوعان: واجهة المستخدم الرسومية وواجهة الأوامر النصية.

١. واجهة المستخدم الرسومية (بالإنجليزية): (واجهة مستخدم رسومية (GUI)): واجهة المستخدم الرسومية عبارة عن عرض رسومي، بياني، في نافذة واحدة أو عدة نوافذ، يتكون من عدة مكونات. من هذه المكونات: زر الضغط، زر الاختيار. تتيح هذه المكونات للمستخدم إنجاز مهام ضمن الحاسب الآلي.

٢. واجهة الأوامر النصية (بالإنجليزية): (واجهة سطر الأوامر (CLI)): تعتبر هذه الواجهة بين الحاسب الآلي والمستخدم، حيث يسمح للمستخدم أن ينفذ الأوامر بطريقة فعالة جداً، ويسمح أيضاً للمستخدم التعامل مع النظام عن طريق طباعة الأوامر باستخدام لوحة المفاتيح وتخرج النتيجة على شكل نص مطبوع على الشاشة.

تصميم واجهة المستخدم

تصميم واجهة المستخدم (بالإنجليزية: User Interface Design) أو هندسة واجهة المستخدم (بالإنجليزية: User Interface Engineering) هو تصميم لواجهات المستخدم التي تُستخدم في الأجهزة والبرمجيات كالحاسوب وأجهزة الجوال وغيرها من الأجهزة الإلكترونية، التي تُركز على خبرة المستخدم والقابلية على الاستخدام. فالهدف من تصميم واجهة المستخدم هو جعل تفاعل المستخدم مع الجهاز بسيط وكفاءة عالية قدر المستطاع للوصول إلى أهداف المستخدمين.

واجهات المستخدم الجيدة هي التي يتم فيها إنهاء المهام بسهولة وبدون الحاجة لتعقيدات لا لزوم لها. فالتصميم الجرافيكي وفن الطباعة وإخراج المحتوى تدعم الواجهات بشكل كبير من حيث المساعدة على تحسين التصميم والإخراج النهائي لجذب المستخدمين ولكن الاهتمام بتصميم الجملاليات إما قد يزيد أو يُنقص من استخدام كافة الوظائف والعمليات في الواجهة. لذلك لا بُد من الحفاظ على التوازن ما بين الوظائف التقنية الفعلية والتصاميم الجمالية في الواجهة، للوصول إلى نظام لا ينفذ العمليات فحسب إنما أيضاً قابل للاستخدام والتكيف مع احتياجات المستخدمين.

تصميم الواجهات ذات نطاق واسع فهي في العديد من المشاريع من أنظمة الحاسوب إلى السيارات والطائرات التجارية، فكل هذه المشاريع تعتمد على ذات القواعد الأساسية من حيث تفاعل الإنسان ولكنها قد تتطلب أيضاً مهارات وخبرات متفردة. لذلك يميل المصممون للتخصص في مجال محدد من المشاريع لتركيز مهاراتهم وخبراتهم فيه للوصول إلى البراعة فيه، سواء أكان ذلك في تصميم البرمجيات أو تصميم المواقع أو التصميم الصناعي أو غيرها.

مراحل تصميم واجهة المستخدم:

تصميم واجهات المستخدم يتطلب فهم جيد لاحتياجات المستخدمين. فهناك عدة مراحل لتصميم واجهة المستخدم بعضها أكثر طلباً من الآخر وذلك يعتمد على المشروع ذاته. (ملاحظة: في هذا الجزء استخدام كلمة نظام تدل على أي مشروع سواء كان موقع أو تطبيق أو جهاز).

جمع المتطلبات - عمل قائمة بجميع المتطلبات والوظائف التي سيقوم بها النظام لخدمة المستخدم وتحديد الهدف من هذا المشروع وتوضيح احتياجات المستخدم المحتملة.

تحليل المهام والمستخدمين - هو شكل من أشكال البحوث حيث يتم فيه تحليل المستخدمين ومهام النظام من خلال عمل دراسات حول كيفية أداء المهام وعمل مقابلات مع المستخدمين لتوضيح أهدافهم. وقد تشمل الأسئلة مايلي:

- ماذا يريد المستخدم من النظام أن يفعل؟
- كيف يتناسب النظام مع أنشطة المستخدم اليومية؟
- ما مدى معرفة المستخدم وماهي البدائل المشابهة التي يستخدمها الآن؟

هيكلية المعلومات -تطوير العمليات (و ذلك من خلال عمل مخطط شجري يوضح خطوات سير النظام، كرس مخطط لموقع وتوضيح تسلسل الصفحات بشكل هرمي.)

عمل نماذج أولية - و ذلك إما أن تكون على الورق أو عمل شاشات تفاعلية بسيطة. في هذه النماذج يتم التركيز على المحتوى في الواجهة.

فحص القابلية للاستخدام - فحص مدى قابلية استخدام الواجهة. بشكل عام هذه المرحلة تُعد أقل تكلفة من مرحلة اختبار الاستخدام (في المراحل أدناه) فهي يمكن استخدامها في المراحل الأولية من عملية التطوير لأنها تتم على النماذج الأولية من النظام بينما لا يُمكن اختبارها على المستخدمين . و تشمل بعض عمليات فحص قابلية الاستخدام : التقييم المعرفي الذي يُركز على البساطة في تنفيذ المهام مع النظام بالنسبة للمستخدمين الجدد، التقييم الاستدلالي الذي يتم فيه جمع عدد من الاستدلالات التي تُساهم في حل مشكلات تصميم الواجهات، التقييم العددي الذي يتم من خلال تحديد مجموعة مختارة من المستخدمين لينفذوا سيناريو المهمة ويحددوا مدى قابلية استخدامها .

اختبار الاستخدام - اختبار النماذج الأولية على مستخدمين فعليين - و ذلك من خلال استخدام تقنية التفكير بصوت عال وفيها يقوم المستخدمون بالتعبير عن أفكارهم من خلال التجربة. اختبار تصميم الواجهات يسمح للمصمم بفهم التصميم من وجهة نظر المستخدم وبالتالي يسهل إنشاء تطبيقات ناجحة .

تصميم واجهات المستخدم - ففي هذه المرحلة يتم التصميم الفعلي النهائي لواجهة المستخدم. ربما يعتمد التصميم على النتائج التي تم تطويرها من خلال البحوث التي تمت على المستخدمين، وإعادة حل للمشكلات التي تم إيجادها في نتائج اختبار الاستخدام (أي الخطوات السابقة). و ذلك يعتمد على نوع الواجهات التي تم إنشاؤها، ففي هذه المرحلة تتضمن العمليات بعض البرامج للتأكد من صحة النماذج والروابط و تنفيذ الإجراءات المطلوب بالشكل الصحيح .

صيانة البرمجيات - بعد نشر الواجهات الجديدة - ربما تدعو الحاجة لصيانة بعض الأخطاء، وتغيير بعض المميزات أو ترقية النظام كاملاً. فبمجرد اتخاذ قرار لترقية وتطوير الواجهات سيخضع النظام القديم لإصدار آخر حديث وتبدأ بعدها عملية تكرار للخطوات التي تمثل دورة حياة الواجهات.

تم بحمد الله