

## ما هي ال Layouts؟

- هي عملية بناء الصفحة وتقسيمها.
- هي الطريقة لوضع النصوص والصور وغيرها من العناصر واختيار أماكنهم وترتيبهم في الصفحة.
- يوجد نوعين من ال Layout هما:

### Page layout

طريقة ترتيب وتنظيم العناصر داخل الصفحة نفسها.

### Component layout

طريقة ترتيب وتنظيم العناصر داخل ال Component نفسها

-----

## Floats

- يجعل العنصر خارج ال Normal flow.
- يجعل النصوص وال Inline elements تلتف حول العنصر.
- الأب (Parent) لن يشعر بالعنصر ولن يكون قادر على ضبط ارتفاع (height) العنصر.

Normal Flow	Absolute Positioning	Floats
<ul style="list-style-type: none"> <li>-Default positioning</li> <li>-Element is "in flow"</li> <li>-Elements are simply laid out according to their order in the HTML code</li> </ul>	<ul style="list-style-type: none"> <li>-Element is removed from the normal flow: "out of flow"</li> <li>-No impact on surrounding elements, might overlap them</li> <li>-We use top, bottom, left, or right to offset the element from its relatively positioned container</li> </ul>	<ul style="list-style-type: none"> <li>-Element is removed from the normal flow: "out of flow"</li> <li>-Text and inline elements will wrap around the floated element</li> <li>-The container will not adjust its height to the element</li> </ul>
<code>position: relative</code>	<code>position: absolute</code>	<code>float: left</code> <code>float: right</code>

## Box Model

### Element width

left border + left padding + width + right padding + right border

### Element height

top border + top padding + height + bottom padding + bottom border

**box-sizing: content-box;**

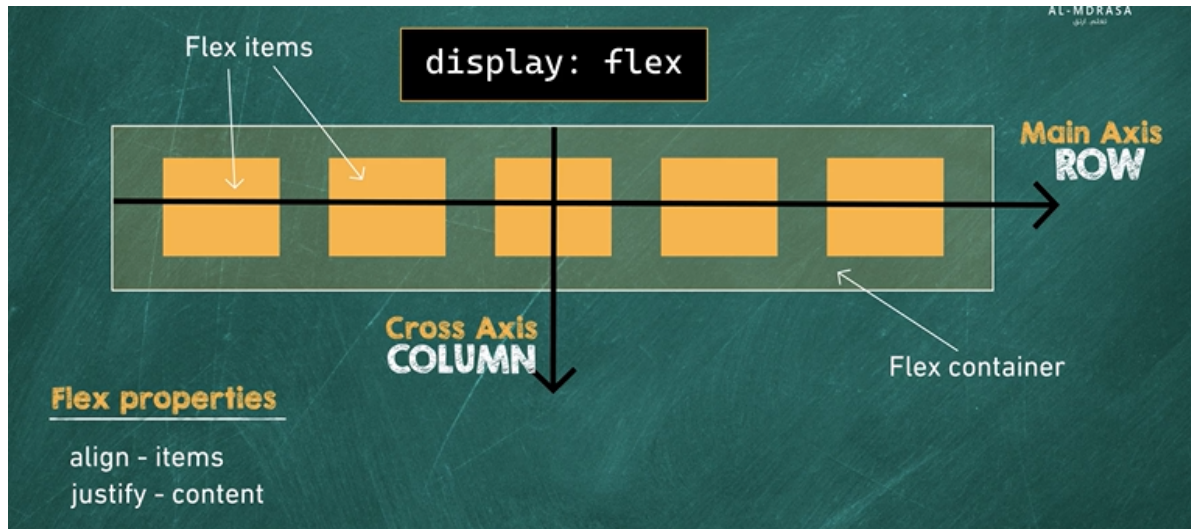
هذه هي القيمة الافتراضية (Default) والتي تجعلنا لا نستطيع التحكم بشكل كامل في حجم العنصر لأنها تعمل على جمع border و padding مع عرض (Width) وطول (Height) العنصر.

**box-sizing: border-box;**

- يعتمد فقط على قيمة عرض (Width) وطول (Height) العنصر.
- أي أن عرض (Width) وطول (Height) العنصر يكون من ضمنهم border و padding

## Flexbox

- هو مجموعة من CSS properties تساعدنا في بناء One Dimensional Layout.
- يجعلنا نتحكم في Empty space.
- تسهيل عملية التوسيط الرأسي (Vertical centering).
- لاستخدام Flexbox نكتب display: flex;



## أشهر Flex properties

Flex container	Flex items
<p><b>gap: 0   &lt;length&gt;</b> To create space between items, without using margin</p> <p><b>justify-content: flex-start flex-end</b>   center   space-between   space-around   space-evenly To align items along main axis (horizontally, by default)</p> <p><b>align-items: stretch</b> flex-start   flex-end   center   baseline To align items along cross axis (vertically, by default)</p> <p><b>flex-direction: row</b>   row-reverse   column   column-reverse To define which is the main axis</p> <p><b>flex-wrap: nowrap</b>   wrap   wrap-reverse To allow items to wrap into a new line if they are too large</p> <p><b>align-content: stretch</b>   flex-start   flex-end   center   space-between   space-around Only applies when there are multiple lines (flex-wrap: wrap)</p>	<p><b>align-self: auto</b>   stretch   flex-start   flex-end   center   baseline To overwrite align-items for individual flex items</p> <p><b>flex-grow: 0   &lt;integer&gt;</b> To allow an element to grow (- means no, +1 means yes)</p> <p><b>flex-shrink: 1   &lt;integer&gt;</b> To allow an element to shrink (- means no, +1 means yes)</p> <p><b>flex-basis: auto   &lt;length&gt;</b> To define an item's width, instead of the width property</p> <p><b>flex: 0 1 auto &lt;int&gt; &lt;int&gt; &lt;len&gt;</b> Recommended shorthand for flex-grow, -shrink, -basis.</p> <p><b>order: 0   &lt;integer&gt;</b> Controls order of items. 1- makes item first, 1 makes it last</p>

AL-MDRASA  
مدرسة المدرس

تحدثنا في الدرس السابق على أن Flexbox ليس خاصية واحدة بل مجموعة من الخواص المختلفة.

نستطيع استخدام Flexbox وخواصه عن طريق إضافة `display: flex;` إلى Container

الذي بداخله مجموع العناصر.

من خواص Flexbox

**`align-items: center;`**

تقوم بتوسيط العناصر بشكل رأسي

**`justify-content: center;`**

تقوم بتوسيط العناصر بشكل أفقي

**`align-self: center;`**

نستخدمها لتوسيط عنصر معين دون باقي العناصر.

**`order: 2;`**

نستخدمها لوضع عنصر معين في مكان معين على حسب ترتيب العناصر وهو من الخواص

الخاصة بال child element

**`gap: 10px;`**

نستخدمها لوضع مسافات بين عناصر ال flex.

## The flex Property

### flex-grow

- القيمة الافتراضية (Default) لهذه الخاصية هي 0.
- تحدد هل عناصر Flex مسموح لهم بإشغال المساحة الفارغة في الأب (Parent) أم لا.

### flex-shrink

القيمة الافتراضية (Default) لهذه الخاصية هي 1

### flex-basis

- القيمة الافتراضية (Default) لهذه الخاصية هي auto.
- نستخدمها لوضع عرض (Width) محدد للعنصر، وتشبه Width property مع بعض المميزات.

- إذا كانت قيمة flex-shrink هي القيمة الافتراضية (Default) وهي 1 فإنه مهما كانت قيمة flex-basis فإن العنصر سوف يحترم الأب (Parent) والمساحة الموجودة.

**flex: 1;**

هذا Shorthand الذي يجمع الثلاث خصائص السابقة ويجعلهم متساويين في الحجم وسوف تكون قيمهم هي

flex-grow: 1;

flex-shrink: 1;

flex-basis: 0;

## CSS Grid

- يتكون من Grid container و Grid items.
- للاستفادة من ال CSS Grid نكتب display: grid بداخل Grid container.

- يساعدنا على بناء Two Dimensional Layout.
- لتحديد عدد وحجم الأعمدة (Columns) نستخدم

**grid-template-columns**

- لتحديد عدد وحجم الصفوف (Rows) نستخدم

**grid-template-rows**

- لوضع مسافة بين عناصر ال Grid نستخدم

**gap: row column;**

**ما هو CSS Grid؟**

هو مجموعة من CSS properties تساعدنا في بناء Two Dimensional Layout.

**ما هي مميزات CSS Grid؟**

- يساعدنا على بناء Layouts معقدة دون الحاجة للمكتبات.
- سهل الفهم، ويجعل شكل الكود منظم وبسيط.

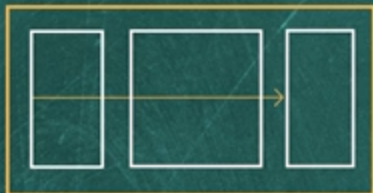
**هل CSS Grid يستبدل Flex Box؟**

- الإجابة هي لا، لأنهم يكملون بعضهم البعض.



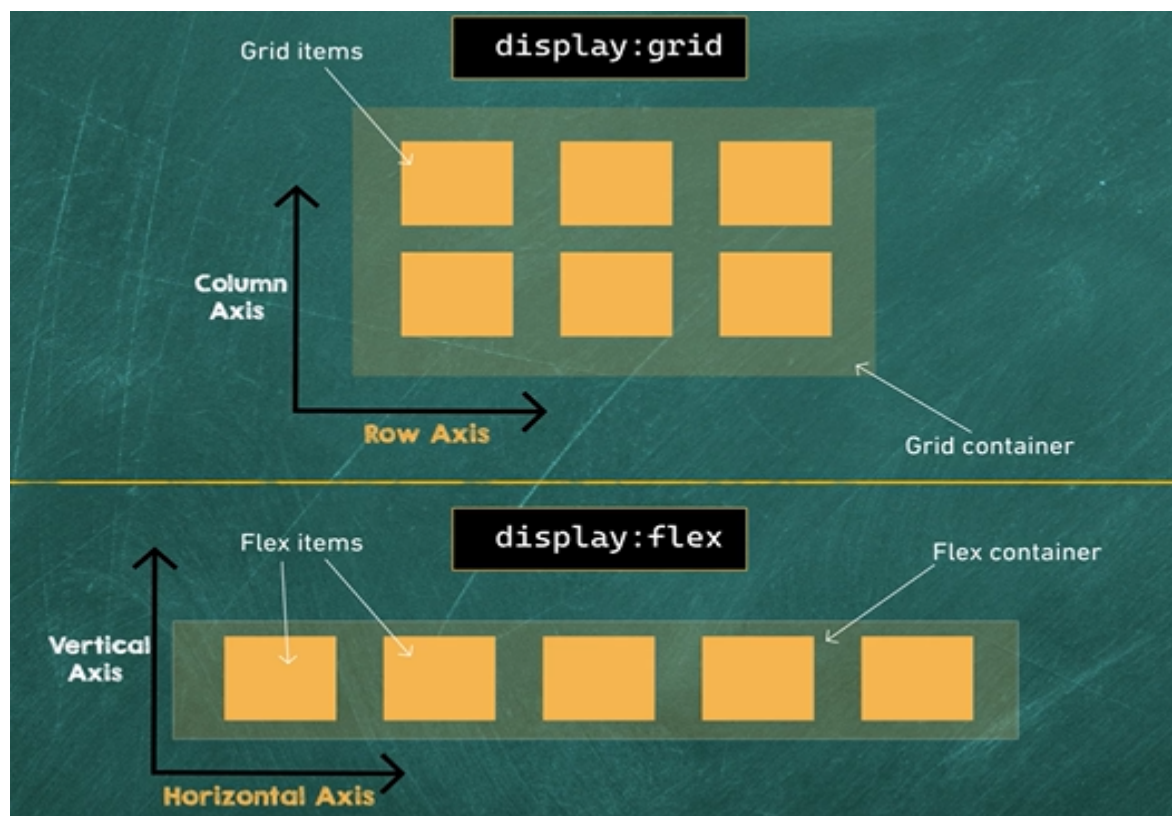
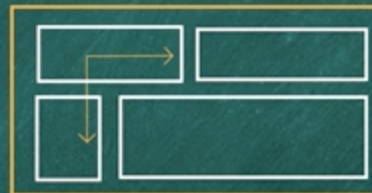
## Flex Box

مناسب جدا في ال one dimensional  
( Component Layout )



## CSS Grid

مناسب جدا في ال Two dimensional  
( Page Layout )



## خصائص CSS Grid

Grid container	Grid items
<p><b>grid-template-rows:</b> &lt;track size&gt;*</p> <p><b>grid-template-columns:</b> &lt;track size&gt;*</p> <p>To establish the grid row and column tracks. One length unit for each track. Any unit can be used, new fr fills unused space</p> <p><b>row-gap:</b> 0   &lt;length&gt;</p> <p><b>column-gap:</b> 0   &lt;length&gt;</p> <p>To create empty space between tracks</p> <p><b>justify-items:</b> stretch   start   center   end</p> <p><b>align-items:</b> stretch   start   center   end</p> <p>To align items inside rows / columns (horizontally / vertically)</p> <p><b>justify-content:</b> start   start   center   end  </p> <p><b>align-content:</b> start   start   center   end   .</p> <p>To align entire grid inside grid container. Only applies If container is larger than the grid</p>	<p><b>grid-column:</b> &lt;start line&gt; / &lt;end line&gt;   span &lt;number&gt;</p> <p><b>grid-row:</b> &lt;start line&gt; / &lt;end line&gt;   span &lt;number&gt;</p> <p>To place a grid item into a specific cell, based on line numbers. span keyword can be used to span an item across more cells</p> <p><b>justify-self:</b> stretch   start   center   end</p> <p><b>align-self:</b> stretch   start   center   end</p> <p>To overwrite justify-items / align-items for single items</p>