THE DEPTHS

in CSS Grid

الأعْمَاقُ فِي تَخطِيطِ شَبَكَةِ أوُرَاقِ الأَنمَاطِ المُتَعاقِبَة

لوُرَنس أحمَد عُثمَان

لوُرَنس أحمَد عُثمَان

مهتم بالتقنية | مطور مواقع ويب | مبرمج

الإيميل Lorans.ev@gmail.com

ترخيص الكتاب

الأعْمَاق

4

بوح نتألم كثيراً عندما نضطر مجبرين للاعتماد على المصادر الأجنبية في رحلة التعلم، فهي عدا عن محتواها الغزير بالمعلومات نلاحظ فيها نوع من الاهتمام بالشكل، فالحلوى تُصبح أطيب عندما تُغلف كهدية، فأحببت أن أقدم لكم كتاب «الأعماق» تحت شعار «المعلومة أولاً، وقليل من التنسيق لن يضر» وتحت هدف واحد «لعله، عِلمٌ يُنتَفَعُ بِه».

المُحْتوَياتْ

إهداء

إلى النضال..

إلى كل شيء..

إلى أمي ، أما بعد:

إلى السيد أسامة محمد السيد - الزيرو ، العطاء لديك لا يفنى

ولا يُخلق من العدم بل يتحول من شكل لآخر.

إليك أنت أيها المناضل في دروب العلم..

إلى كل المساهمين في إثراء المحتوى العربي..

محبتي الخالدة.

مُقَدِّمَة

لم تكن مواقع الإنترنت في التسعينات نفسها التي نراها اليوم، فالمواقع كانت عبارة عن صفحات تحتوي نصوص وصور مرتبطة بصفحات أخرى، أما مواقع اليوم اختلفت بشكل كبير فالمواقع اليوم أشبه بصالات عرض للمنتجات والأمر أشبه بمقارنة عربة تنتقل بالدفع مع سيارة تسلا، كلاهما تملكان أربعة إطارات وتسيران على الطريق، لكن الإمكانيات التي تقدمها كلٍ منهما مختلفة كلياً، فالمواقع لم تعد تقتصر على عرض المحتوى بل أصبحت قابلة للتفاعل والتخصيص وتقديم خدمات أكثر، ولذلك كان على مطوري الويب مواكبة هذه التطورات، ومن أبرز القضايا التي دفعت مجتمع تطوير الويب هو تنسيق المحتوى، فلا يكفي أن يكون لدي محتوى مهم للعملاء بل يجب عرضه بطريقة تليق بهم أيضاً، كانت البدايات بتقسيم مساحة الصفحة عن طريق الجداول، ثم انتقلت للإطارات، ظهرت بعدئذ خرائط الصور، ثم ظهرت الحاويات التي تقسم الصفحة لأقسام منطقية وحلت بديلاً عن الجداول، إلى أن جاء تخطيط جديد يعتمد على العناصر العائمة Float، وبعد ظهور أجهزة الجوال الذكية و ظهور الحاجة للمواقع المتجاوبة مع مختلف أجهزة العرض مما أدى لنشوء استعلامات الوسائط Media Queries ، وبعد ذلك ظهرت تقنية Flexbox التي تتعامل مع العناصر ضمن بعد واحد عمودي أو أفقي، مما مهد لظهور التقنية التي تتعامل مع العناصر ضمن بعدين في وقت واحد وهي تقنية الشبكة Grid التي سنتناولها في هذا الكتاب.

كعربي ، أقدمه كمساهمة في إثراء المحتوى العربي.

كمسلم ، أقدمه للجميع علّه يكون علم يُنتفع به.

إن أصبت فمن اللّه ، و إن أخطأت فمن نفسي ، واللّه الموفق.

لورنس أحمد عثمان

19-2-2020

إدلب، سورية

ماهي HTML

HTML

يشار بها اختصاراً إلىHyperText Markup Language ، و تعني لغة ترميز النص الفائق، ولكن ما هو النص الفائق؟

لنتخيل أنك تقرأ كتاب، بالكتاب لن تستطيع الانتقال من الصفحة الأولى إلى الصفحة الثامنة مثلاً حتى تمر بكل الصفحات بينهما، هذا هو النص الخطي، أي أنك تسير في خط واحد وللانتقال من نقطة لنقطة فأنت مقيد بالمرور بجميع النقاط بينهما، بينما بالنص الفائق تستطيع الانتقال من نقطة إلى نقطة دون المرور بباقي النقاط، كما يحدث في صفحات الويب فأنت تنتقل من نقطة لنقطة مباشرة دون المرور بباقي المكونات ، تقوم HTML بوصف النص الفائق لبرامج التصفح فقد يكون صورة أو جدول أو فقرة أو عنوان وذلك عن طريق الوسوم tags، فكل وسم يصف نوع مختلف، إذاً هي لغة أو طريقة لوصف بنية الصفحة لبرامج التصفح لكي تقوم بعرضها بشكل صحيح، وأصبحت HTML اللغة الرئيسية التي يتكون منها أي موقع أو صفحة ويب على شبكة الإنترنت اليوم، بينما يعود تاريخ ظهور هذه اللغة إلى سنة 1991 على يد مؤسس الشبكة العنكبوتية تيم بيرنرز ليس Tim Berners less، أما النسخة التالية HTML v2.0 فيعود تاريخ تأسيسها إلى سنة 1995، وتوالت التحديثات عليها حتى آخر نسخةٍ منها في عام 2016.

ماهي CSS

CSS

يشار بها اختصاراً إلى Cascading Style Sheets ، وتعني أوراق الأنماط المُتعاقبة، تحدّد لغة HTML هيكل الصفحة وتُخبر المتصفح بوظيفة كل عنصر في الصفحة (مثل رابط لصفحة أخرى أو عنوان رئيسي) في حين تقدّم لغة CSS تعليمات للمتصفح حول كيفية عرض هذا العنصر ضمن الصفحة من حيث التصميم والمسافة والموضع، ولو افترضنا أن HTML تقوم على بناء هيكل المنزل فإن CSS تقوم بطلاء المنزل و تصميم الديكور الخاص به، ويتم ذلك عن طريق مجموعة من التعليمات التي تحدد من و ما يجب تنسيقه على عناصر HTML ، كما تحتوي تلك التعليمات على خاصيات مثل الألوان وحجم الخطوط ونوع الخط، كما تستطيع عزل التنسيق (الألوان - الخطوط - الأزرار...) عن محتوى المستند المكتوب بـ HTML .

يعود تاريخ ظهور لغة CSS لأول مرة إلى سنة 1994 على يد المبرمج هاكون فيوم لاي Hakon Wium Lie، حيث تعاون مع فريق متكامل من المبرمجين للخروج في نهاية المطاف بأوراق نمطية لتصميم صفحات الويب بكل كفاءة، وتم التوصل إلى ضرورة إصدار الوثيقة الأولى W3C CSS سنة 1996، فكان ذلك الإصدار الأول لها وتم إصدار النسخة التالية بناءً على اقتراحات بيرت بوس، ازدادت أهمية لغة CSS بالتزامنِ مع تطوير لغة HTML لتكمل كلٍ منهما الأخرى، وقد ساهم دمجهما بجعل عرض الصفحات عبر الشبكة العنكبوتية أمرا أكثر وضوحاً وسهولة وأهمية للمستخدمين، وفي كل إصدار من إصدارات هذه اللغة كان فريق العمل القائم على تطويرها حريصاً على معالجة الثغرات التي يعاني منها الإصدار السابق وإضافة المزيد من السمات والخصائص الإضافية للخروج بإصدار جديد، وتشير المعلومات إلى أن هناك إصدارات ما زالت خاضعة للتطوير منذ عام 2014.

مُصْطَلحَات تخْطِيطِ

Grid

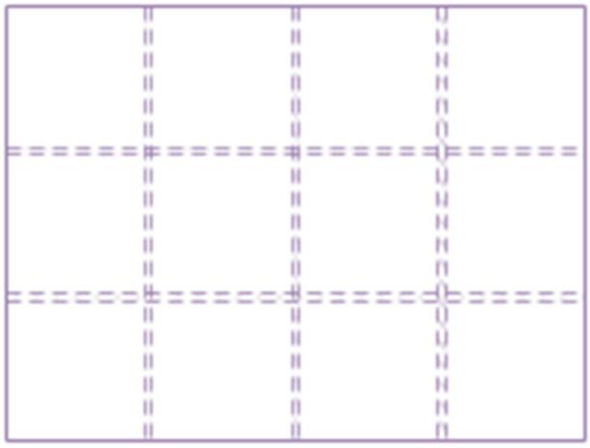
تعددت تقنيات لغة CSS التي تسمح للمطورين بإنشاء صفحات ويب متجاوبة مع مختلف أجهزة العرض، وفي عام 2017 ظهرت تقنية جديدة هي تقنية الشبكة Grid أو تم اعتمادها رسمياً من قبل المتصفحات الرئيسية في ذلك الوقت، سنقوم في هذا الكتاب برحلة لمعرفة هذه التقنية وكشف روعتها.

تخطيط Grid :

تخطيط يُقدم نظام بشبكة ثنائية الأبعاد تُستعمل لتقسيم صفحة أو عناصر HTML إلى مجموعة من المساحات و تحديد العلاقة من حيث المقاس والموقع والترتيب فيما بينها.

شبكة Grid :

شبكة ناتجة عن تقاطع صفوف (مسارات أفقية) مع أعمدة (مسارات عمودية) ينتج عنه خلايا وخطوط شبكة و يمكن الفصل بين هذه المسارات بفواصل، تُستعمل لوضع عناصر HTML ضمنها.



تتمتع شبكة Grid بعدة خصائص:

- مقاسات المسارات يمكن أن تكون ثابتة أو مرنة: يمكن إنشاء شبكة بمسارات ذو مقاسات ثابتة باستعمال px مثلاً ، أيضاً يمكن أن تكون مقاسات المسارات مرنة باستعمال النسبة المئوية ٪ أو واحدة fr المصممة للمقاسات المرنة .

- التحكم بموقع العناصر: يمكن ضبط العناصر لأن تكون في موقع محدد على الشبكة اعتمادا على أرقام أو أسماء خطوط الشبكة أو الإشارة للمنطقة من الشبكة التي سيتبع لها العنصر. كما تتضمن شبكة Grid خوارزمية للتحكم بمواقع العناصر التي لم يتم ضبط موقعها على الشبكة بشكل صريح.

- التحكم بتموضع العناصر: تمتلك شبكة Grid خاصيات للتحكم بتموضع العناصر حيث تضبط العلاقة بين حدود العنصر وحدود الموقع أو المنطقة من الشبكة التي ضُبِط العنصر إليها، مثلا ضبط أحد العناصر ليكون في منتصف الموقع المُسند له.

- توليد مسارات إضافية لاحتواء العناصر: في تخطيط Grid يمكن التعريف عن شبكة Grid بشكل صريح وذلك بتعريف المسارات العمودية والأفقية بشكل صريح من حيث العدد والمقاس، ولكن عندما يكون هناك عناصر أكثر مما تستوعبه الشبكة الصريحة أو تم ضبط موقع أحد العناصر أو تمدد لخارج حدود الشبكة الصريحة فتخطيط Grid مرن بما فيه الكفاية لتوليد مسارات جديدة لاحتواء جميع العناصر.

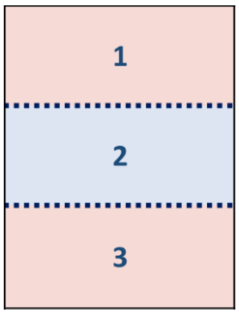
- التحكم بتداخل العناصر: يمكن للعناصر أن تتراكب فوق بعضها البعض إذا صرحنا لأكثر من عنصر بأن يكون لهم الموقع نفسه، يمكن التحكم بهذا التداخل من حيث أولوية الظهور بخاصية z-index .

حاوية الشبكة (Grid Container)

يتم إنشاء حاوية Grid بإسناد إحدى القيمتين grid أو inline-grid لخاصية display لأحد العناصر، وبذلك يصبح هذا العنصر حاوية شبكة وكل أبناؤه المباشرين يصبحون عناصر شبكة Grid Items .

لن تلاحظ أي فرق بصري عند تحويل العنصر إلى تخطيط الشبكة مبدئياً لأن الشبكة ستُنشئ مسار عمودي واحد وعدة مسارات أفقية على عدد العناصر الأبناء.

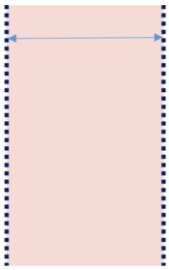
```
<div class="grid-container" style="display: grid;">
  <div>1</div>
  <div>2</div>
  <div>3</div>
 </div>
```



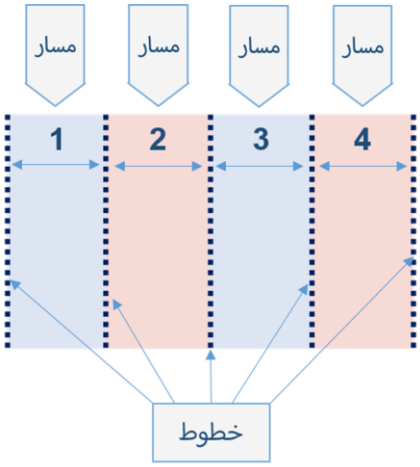
مسارات الشبكة (Grid Tracks)

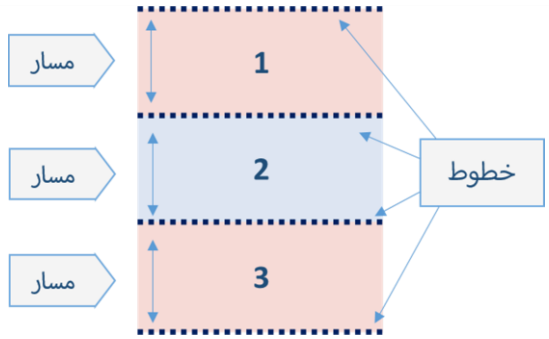
يمكن تقسيم حاوية الشبكة إلى أي عدد من المسارات الأفقية والعمودية، و يتم إنشاء المسارات إما بشكل صريح أو ضمني.

عند ضبط قيمة مقاس أي مسار فإننا عملياً نضبط المسافة بين خطين، خط يمثل البداية وخط يمثل النهاية.



إذاً المسار هو المسافة بين خطين، هذه الخطوط هي خطوط الشبكة، يمكن أن يكون خط النهاية لأحد المسارات هو خط البداية للمسار التالي.





نلاحظ من الأشكال السابقة أن عدد الخطوط أكثر من عدد المسارات بواحد، فثلاثة مسارات أفقية أنتجت أربع خطوط أفقية في الشكل الأول، و أربعة مسارات عمودية أنتجت خمس خطوط عمودية في الشكل الثاني.

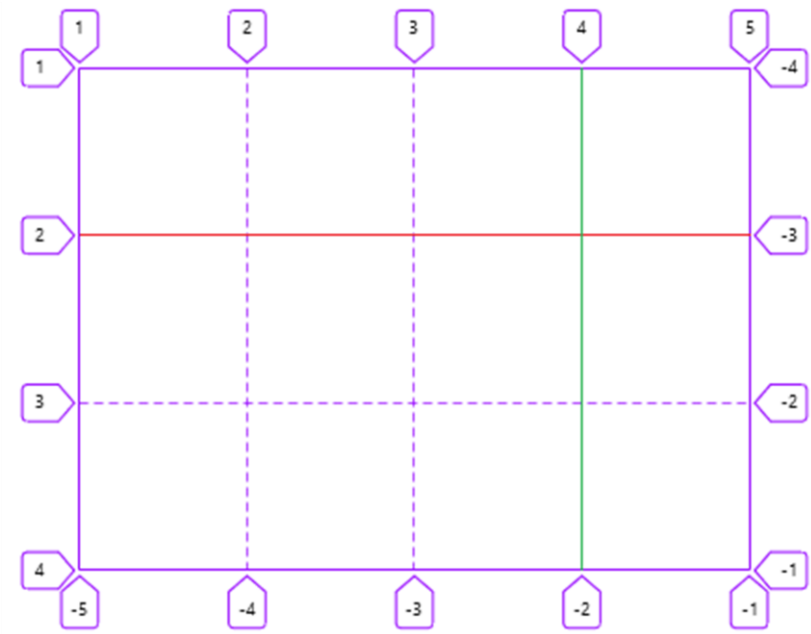
هنا لدينا شبكة من ثلاثة مسارات أفقية وأربع مسارات عمودية، أنتجت أربع خطوط شبكة أفقية مع خمس خطوط شبكة عمودية.

خطوط الشبكة (Grid Lines)

هي خطوط مُكونَة نتيجة لتكوين المسارات، هي خطوط إرشادية تُستعمل لتحديد موقع وتموضع العناصر ضمن الشبكة، يتم ترقيم الخطوط آلياً كما يمكن تسميتها يدوياً.

ما يجب ملاحظته أنه عند التعريف عن الشبكة نحن نصرح عن المسارات ولا نصرح عن الخطوط، الخطوط تُنتج تلقائياً فهي حد البداية وحد النهاية لأي مسار يتم تكوينه.

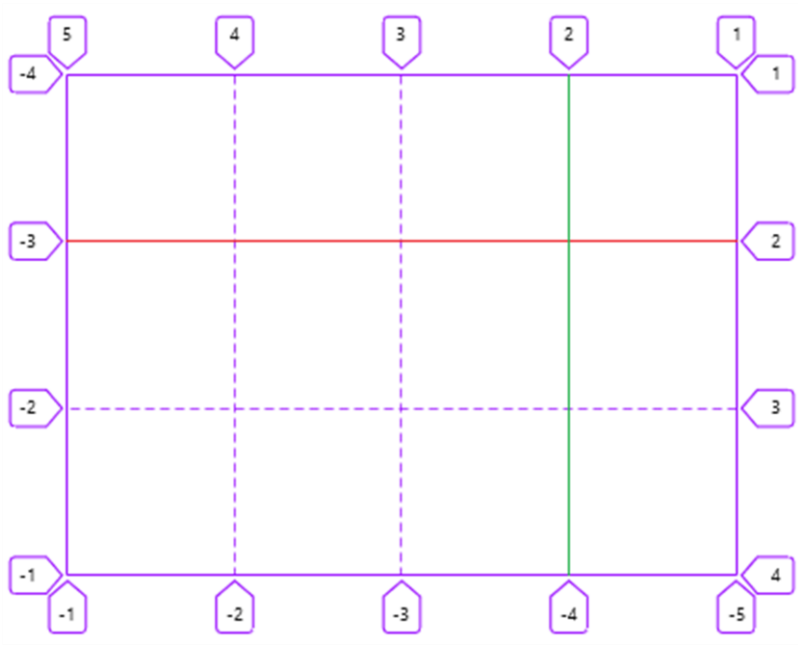
- يتم ترقيم الخطوط آلياً بحسب اتجاه العنصر بحيث يكون كل خط عمودي بالأعداد الموجبة مع اتجاه العنصر وبالأعداد السالبة عكس اتجاه العنصر، و كل خط أفقي بالأعداد الموجبة بالاتجاه من الأعلى للأسفل و بالأعداد السالبة من الأسفل للأعلى.



اتجاه العنصر من اليسار لليمين LTR

الخط الأخضر العمودي هو الرابع مع اتجاه العنصر وسالب اثنان عكس اتجاه العنصر

الخط الأحمر الأفقي هو الثاني من أعلى لأسفل وسالب ثلاثة من أسفل لأعلى

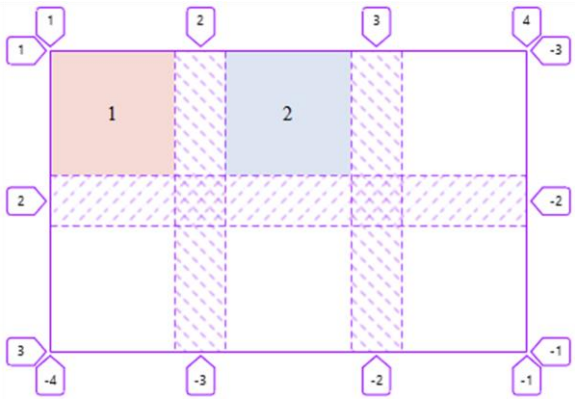


اتجاه العنصر من اليمين لليسار RTL

الخط الأخضر العمودي هو الثاني مع اتجاه العنصر وسالب أربعة عكس اتجاه العنصر

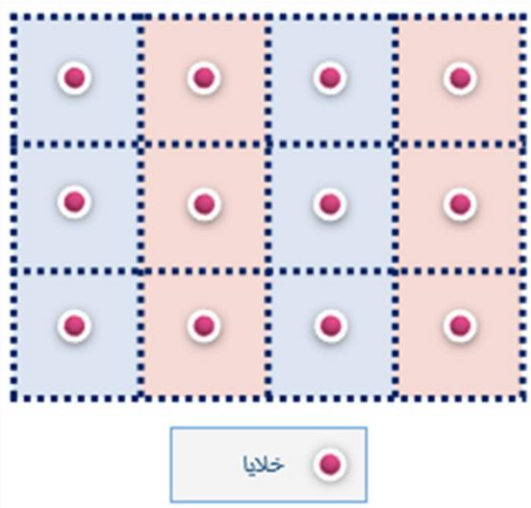
الخط الأحمر الأفقي هو الثاني من أعلى لأسفل وسالب ثلاثة من أسفل لأعلى

- **يمكن تسمية خطوط الشبكة يدوياً للإشارة لأسماء الخطوط بدلاً من استعمال الأرقام، سنتعرف على ذلك في الفصول اللاحقة.**
- **عند وجود فواصل بين المسارات نلاحظ أن الخط الواحد يكون على جانبي الفاصل، في الشكل التالي نلاحظ أن الخط العمودي الثاني يمثل نهاية العنصر الأول وبداية العنصر الثاني في نفس الوقت على الرغم من وجود الفاصل بينهما، سنتحدث عن الفواصل لاحقاً في هذا الفصل.**



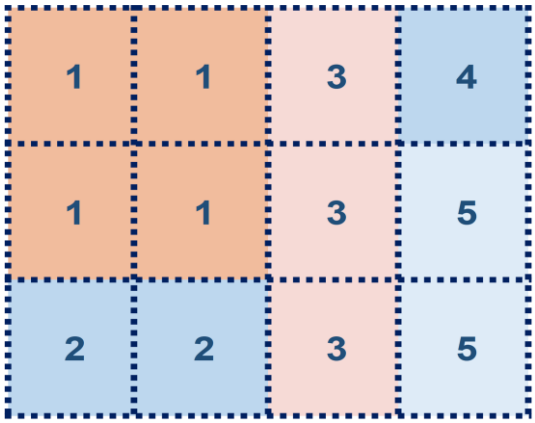
خلايا الشبكة (Grid Cells)

أصغر وحدة في الشبكة تمثل أصغر مسافة محاطة بأربع خطوط، خطين متجاورين أفقيين وخطين متجاورين عموديين، بمعنى آخر أصغر وحدة ناتجة عن تقاطع مسار أفقي مع مسار عمودي، من حيث المبدأ تشبه خلية في أحد الجداول.



مناطق الشبكة (Grid Areas)

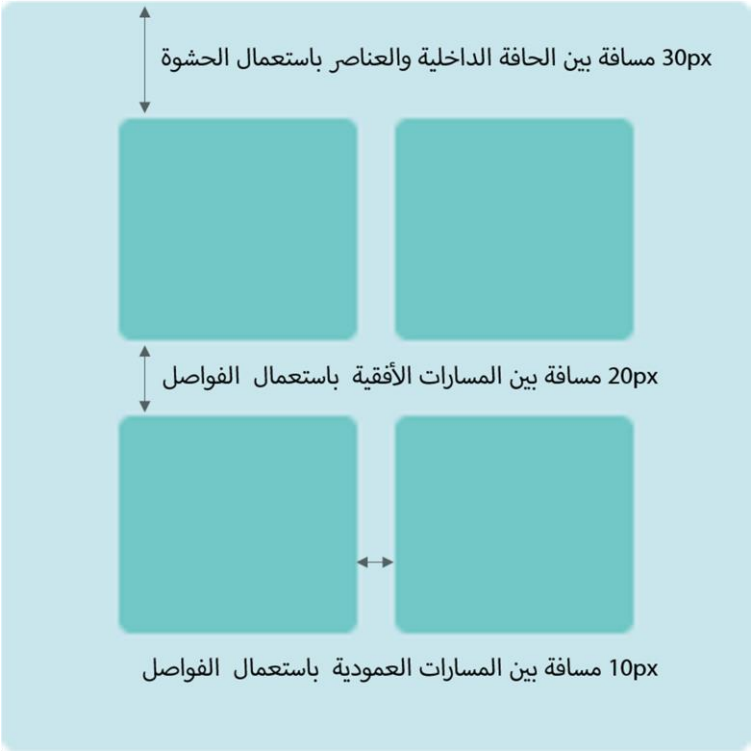
هي وحدة محاطة بأربع خطوط، خطيين أفقيين إما أن يكونا متجاورين أو لا، وخطين عموديين إما أن يكونا متجاورين أو لا، بمعنى آخر هي وحدة تمثل خلية واحدة أو تمتد لتشمل أكثر من خلية أفقياً أو عمودياً أو معاً.



حسب آخر مواصفة للشبكة Grid specification من الضروري أن تكون المناطق على شكل مربع أو مستطيل، أي لا يمكن إنشاء منطقة على شكل L مثلاً.

فواصل الشبكة (Grid Gabs)

هي مسافات تفصل بين المسارات، يتم ضبطها لتفصل بين المسارات العمودية أو الأفقية أو الإثنين معاً، لا يمكن وضع أي محتوى داخل الفواصل، كما لا يمكن الفصل بين حواف حاوية الشبكة الداخلية والعناصر الأبناء بواسطة الفواصل، استعمل الحشوة padding بدلاً من ذلك.



عناصر الشبكة (Grid Items)

هي أي عنصر HTML يكون ابن مباشر لحاوية الشبكة، العنصر قد يشغل خلية أو مسار أو منطقة.

يجب الانتباه عند وجود أي محتوى ضمن حاوية الشبكة غير مُغلف بعنصر HTML ستعتبره الشبكة عنصر طبيعي وتعامله بخوارزمية التموقع التلقائي، لا يمكن التعديل على موقع هذا العنصر لأنه لا يمكن استهدافه لتطبيق الخصائص عليه لذلك تسمى عناصر الشبكة المجهولة Anonymous Grid Items.

```
<div class="grid_container">

  أنا نص غير مغلف إذاً أنا عنصر مجهول
  أرث خصائصي من أبي

  <div> العنصر الأول</div>
  <h1> ترويسة</h1>
  <p> فقرة</p>
  <ul class="list">
   <li> لائحة1</li>
   <li> لائحة2</li>
   <li> لائحة3</li>
  </ul>

  وأنا عنصر مجهول أيضاً،
  لا يمكنك تطبيق الخصائص علي لأنه لا يمكنك استهدافي

 </div>
```

29

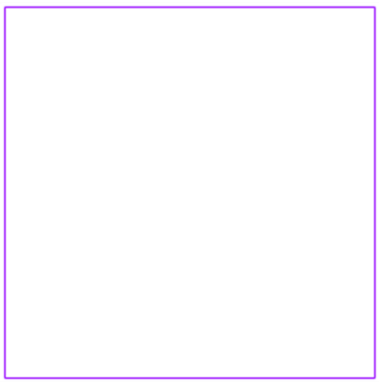
الحَاوِيَةَ

Grid Containre حاوية الشبكة

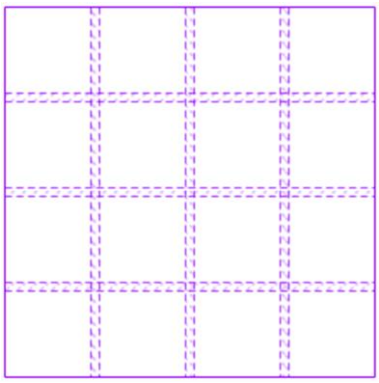
قبل أن نبدأ

لقد قلنا أنَّ الشبكة ناتجة عن تقاطع صفوف (مسارات أفقية) مع أعمدة (مسارات عمودية) ينتج عنه خلايا وخطوط شبكة و يمكن الفصل بين هذه المسارات بفواصل، إذاً ماذا يجب أن نفعل لنحصل على الشبكة؟

بدايةً نحتاج عنصر يحمل الشبكة ويُمثلها، ثم نُنشئ المسارات وهذا بدوره يؤدي إلى تشكيل خلايا وخطوط، نستطيع بعد ذلك وضع فواصل بين المسارات إذا أردنا، وبعد أن يتم ضبط الشبكة نستطيع تقسيم الشبكة إلى مناطق توافق الرؤية التصميمية بحيث مثلاً الشعار وجزء التنقل يحتلان كامل خلايا الصف الأول، كما ونستطيع ضبط تموضع المسارات نسبة إلى حدود الشبكة ونستطيع ضبط تموضع العناصر نسبة إلى حدود الخلية أو المنطقة التي أُسند إليها.



عنصر يُمثل الشبكة



إنشاء المسارات والفواصل وهذا بدوره يؤدي إلى تشكيل خلايا وخطوط

نستطيع بعد ذلك تقسيم الشبكة إلى مناطق توافق الرؤية التصميمية، ونستطيع ضبط تموضع المسارات والعناصر، في الشكل على الجانب تم ضبط تموضع العناصر لتكون في المنتصف.

سنتناول في هذا الفصل ضبط حاوية الشبكة بالتفصيل أما الآن سنسرد أهم النقاط الأساسية في عملية الضبط ليكون لدينا تصور كامل وخريطة ذهنية لكامل العملية.

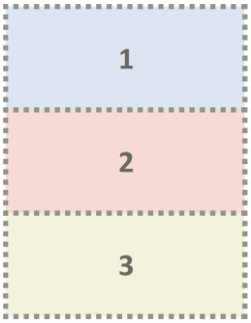
الخريطة الذهنية لضبط حاوية الشبكة:

1. **التصريح عن الشبكة.**
2. **إنشاء الشبكة**
   - **إنشاء المسارات: الصريحة والضمنية.**
   - **إنشاء المناطق.**
   - **إنشاء الفواصل.**
   - **الخاصيات المُختصرَة.**
3. **التحكم بالتموضع**
   - **تموضع المسارات.**
   - **تموضع العناصر.**

التصريح عن الشبكة

يتم التصريح عن حاوية الشبكة بإسناد إحدى القيمتين grid أو inline-grid لخاصية display لأحد العناصر، وبذلك يصبح هذا العنصر حاوية شبكة وكل أبناؤه المباشرون يصبحون عناصر شبكة Grid Items .

```
/////HTML/////

<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

/////CSS/////

.grid-container {
 display: grid;
}

///OR

.grid-container {
 display: inline-grid;
}
```



الفرق بين القيمتين grid و inline-grid هو أن grid تضبط صندوق عرض العنصر ليكون ذو تخطيط كتلي block-level أي أن العنصر سيأخذ جميع العرض المتوفر إذا لم يتم ضبط العرض لمقاس محدد، أما القيمة inline-grid تضبط صندوق عرض العنصر ليكون ذو تخطيط سطري inline-level أي أن العنصر سيأخذ أقل قيمة من العرض تسمح بإظهار كامل المحتوى، كما لن تلاحظ أي فرق بصري عند تحويل العنصر إلى تخطيط الشبكة مبدئياً لأن الشبكة ستُنشئ مسار عمودي واحد وعدة مسارات أفقية على عدد العناصر الأبناء.

ستتأثر العناصر الأبناء عند تحويل العنصر الأب لشبكة grid مثل الخاصيات التي تؤثر على صندوق عرض العنصر مثل float و inline و block و inline-block و table-cell و vertical-align و column لن يكون لها تأثير على عناصر الشبكة Grid Items ، كما أن ظاهرة انهيار الهامش margin-collapse لن تؤثر على عناصر الشبكة Grid Items .

إنشاء الشبكة

بعد التصريح عن الشبكة نلاحظ أنه تم إنشاء مسار عمودي واحد وعدة مسارات أفقية على عدد العناصر الأبناء آلياً، هذه الشبكة التي تم إنشاءها آلياً تسمى شبكة ضمنية Implicit Grid وذلك لأننا لم نقم بتعريف المسارات بشكل صريح بعد.

نستطيع تعريف وإنشاء المسارات بشكل صريح وذلك بضبط عددها ومقاساتها إلى خاصيات إنشاء المسارات الصريحة، وتسمى الشبكة الناتجة شبكة صريحة Explicit Grid وذلك لأننا حددنا عدد المسارات ومقاساتها، نستطيع أيضاً ضبط الفواصل ما بين المسارات، كما نستطيع تعريف المناطق.

بعد إنشاء الشبكة الصريحة قد تحدث معنا عدة حالات مثل زيادة عدد العناصر عن ما يمكن للشبكة الصريحة أن تستوعبه، ستقوم الشبكة آلياً بتوليد مسارات إضافية جديدة لاحتواء العناصر، تسمى هذه المسارات مسارات ضمنية لأننا لم نقم بتعريفها بشكل صريح ضمن الشبكة الصريحة، مقاسات هذه المسارات الضمنية معتمد على محتوى العناصر، كما و يمكن التحكم بمقاسات المسارات الضمنية المُوَلَّدَة واتجاه التوليد.

إذاً فشبكة Grid تتكون من شبكة صريحة هي الشبكة التي تم تعريف عدد ومقاسات مساراتها بشكل صريح، وشبكة ضمنية هي الشبكة التي يتم توليد مساراتها آلياً بناءاً على عدة عوامل سنتعرف عليها لاحقاً في هذا الفصل.

إنشاء المسارات الصريحة

يتم تعريف المسارات الأفقية والعمودية المكونة للشبكة بشكل صريح بإسناد لائحة المسارات لخاصيات إنشاء المسارات الصريحة.

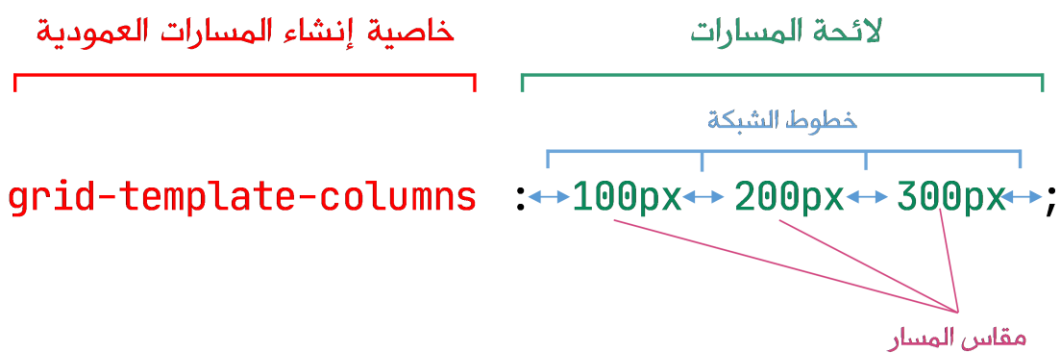
- **خاصيات إنشاء المسارات الصريحة:**

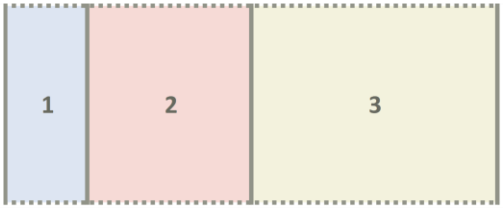
| **خاصية إنشاء المسارات العمودية grid-template-columns**

| **خاصية إنشاء المسارات الأفقية grid-template-rows**

- **لائحة المسارات (Track list) :**

هي لائحة من القيم المفصولة بمسافة، عدد القيم يُمثل عدد المسارات التي سيتم إنشاؤها، كل قيمة من هذه اللائحة تُمثل مقاس المسار الذي سيتم إنشاؤه، وكل مسافة على جانبي كل قيمة تُمثل خط شبكة.





نلاحظ أنه تم إنشاء ثلاث مسارات عمودية صريحة و أربع خطوط شبكة عمودية ، وتم إنشاء مسار أفقي ضمني واحد آلياً مع خطي شبكة أفقيين.

ضبط لائحة المسارات (Track list)

بواسطة لائحة المسارات نستطيع ضبط عدد المسارات ، مقاس كل مسار ، تسمية خطوط الشبكة.

• عدد المسارات

هو عدد قيم مقاسات المسارات المضبوطة، لإضافة مسار جديد نضيف قيمة مقاسه، ولحذف مسار نحذف قيمة مقاسه.

مسار عمودي واحد	`grid-template-columns: 100px ;`
مسارين أفقيين	`grid-template-rows: 100px 200px ;`
ثلاث مسارات عمودية	`grid-template-columns: 100px 200px 300px ;`

• مقاس المسار

يتم التحكم بمقاس المسار بعدد من الطرق.

- length
- %
- auto
- fr
- min-content
- max-content
- minmax() function
- fit-content() function
- repeat()function

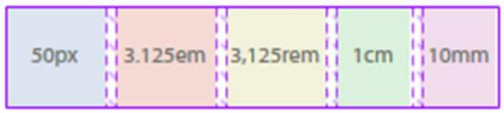
• تسمية خطوط الشبكة

يمكن تسمية خطوط الشبكة يدوياً بحيث يتم إعطاء اسم لكل خط شبكة، سنتحدث عن ذلك لاحقاً في هذا الفصل.

ضبط مقاس المسار

1. Length : ضبط مقاس المسار لنوع البيانات Length الذي يُستعمل لتمثيل القيم الطولية، تتألف القيم الطولية من قيمة عددية <number> يتبعها رمز الواحدة (مثل px أو em أو rem أو cm أو mm ...إلخ)، وكما في جميع واحدات CSS لا يجوز وضع فراغ بين رمز الواحدة والقيمة العددية، إلا إذا كانت القيمة العددية هي 0 (في هذه الحالة يجوز عدم وضع رمز الواحدة بجانب القيمة 0).

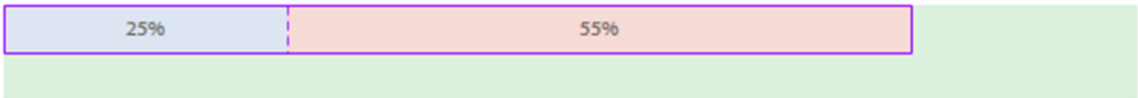
```
grid-template-columns: 50px  3.125em  3.125rem  1cm  10mm ;
grid-template-rows: 50px;
gap: 5px;
```



تشكيل 5 مسارات عمودية مقاس كل مسار مُحدد بواحدة مختلفة، ومسار أفقي مقاسه 50 بيكسل، بالإضافة لفواصل كل فاصل 5 بيكسل

2. النسبة المئوية ٪ : ضبط مقاس المسار نسبةً إلى أبعاد الكتلة الحاوية وهنا حاوية الشبكة، سيتم احتساب مقاسات المسارات العمودية نسبةً لعرض حاوية الشبكة ومقاسات المسارات الأفقية نسبة لارتفاع حاوية الشبكة، عندما يكون مجموع المقاسات أقل من 100بالمئة من الشبكة سوف تظهر خلفية حاوية الشبكة.

```
width: 400px;
grid-template-columns: 25% 55% ;
height: 50px;
grid-template-rows: 50% ;
```



تشكيل مسارين عموديين مقاس المسار الأول ربع عرض الحاوية والثاني أعلى من نصفها بقليل، ومسار أفقي بنصف ارتفاع الحاوية، والباقي مساحة فارغة من حاوية الشبكة.

في حالة كانت أبعاد حاوية grid غير مضبوطة وتعتمد على أبعاد ومحتوى المسارات الموجودة فيها، فيجب أن تُعامل النسب المئوية مثل auto.

3. auto : يتم حساب مقاس المسار أو المسارات المضبوطة إلى auto بحيث يتم اقتطاع (مقاسات المسارات المحددة لقيم ثابتة إن وجدت، واقتطاع مقاسات الفواصل إن وجدت، واقتطاع المقاس الأدنى للمسارات المضبوط لها auto) من مقاس حاوية الشبكة، ثم تقسيم المسافة المتبقية بالتساوي على العناصر المضبوط لها auto.

```
width: 400px;
grid-template-columns: 25%  30px  auto  auto ;
gap: 5px;
```

على الرغم من أن آخر مسارين لهما القيمة auto نفسها إلا أننا نجد أن الثالث أصغر من الرابع، لذلك لنقم ببعض الحسابات.

عرض الحاوية 400 بيكسل سنقتطع مقاسات المسارات المحددة لقيم ثابتة، إذاً سنقتطع 100 بيكسل عن المسار الأول و30 بيكسل عن المسار الثاني، ثم سنقتطع مقاسات الفواصل، لدينا ثلاث فواصل عمودية كل واحد 5 بيكسل إذاً سنقتطع 15 بيكسل، بقي لدينا 255 بيكسل، سنقتطع المقاس الأدنى للعناصر المضبوط لها auto، الأول 22px والثاني 71px بقي لدينا 162 بيكسل سنقسمها بالتساوي على العناصر المضبوط لها auto، إذاً الأول سيصبح 103=81+22والثاني 152=81+71

المقاس الأدنى: هو مقاس المحتوى، في مثالنا المقاس الأدنى للمسار الثالث هو مقاس حروف كلمة auto ، والمقاس الأدنى للمسار الرابع هو مقاس حروف جملة auto track size ، والمقاس الأدنى يتأثر بعدة عوامل منها حجم الخط والحشوة والفراغات بين الكلمات والفراغات بين الحروف.

عند ضبط قيمة المسارات أو أحد المسارات إلى auto نلاحظ أنه:

- لن تبقى مساحات فارغة من حاوية الشبكة لأنه سيتم تقسيمها بين المسارات التي لها القيمة auto.
- إذا كانت المسارات التي لها القيمة auto مختلفة بالمقاس الأدنى أو المحتوى لن تبدو متساوية بالمقاس، أن كان لها نفس المقاس الأدنى ستبدو متساوية.
- لا تستطيع تقسيم المسافة المتبقية نسبياً بين المسارات التي لها auto مثلاً أول مسار يأخذ ربع المسافة المتبقية والثاني يأخذ الباقي لذلك جاءت القيمة fr.

4. fr : يتم حساب مقاس المسار أو المسارات المضبوطة إلى fr بحيث يتم اقتطاع (مقاسات المسارات المحددة لقيم ثابتة إن وجدت، واقتطاع مقاسات الفواصل إن وجدت، واقتطاع المقاس الأدنى للمسارات المضبوط لها auto ، واقتطاع المقاس الأدنى للمسارات المضبوط لها fr فقط إذا كانت حصته من fr أقل من مقاسه الأدنى) من مقاس حاوية الشبكة، ثم تقسيم المسافة المتبقية على مجموع واحدات fr لنحصل على حصة كل fr ثم نعوض كل مسار بعدد واحدات fr المُسندة له.

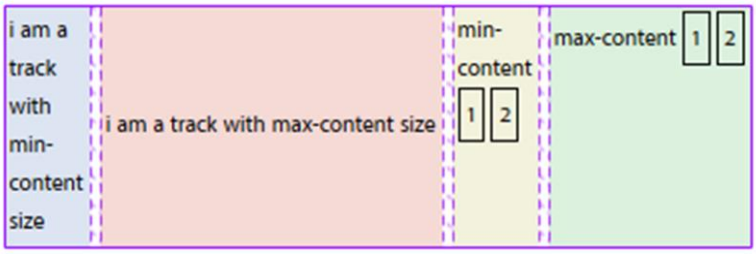
عند وجود مسار له fr فإن كل المسارات المضبوط لها auto لن تتمدد ولن تحصل على حصة من المسافة المتبقية وستظهر دائماً بالمقاس الأدنى.


```
width: 600px;
grid-template-columns: 10% 30px auto 1fr 2fr ;
gap: 5px;
```

هنا لدينا عرض الحاوية 600 بيكسل سنقتطع مقاسات المسارات المحددة لقيم ثابتة، إذاً سنقتطع 60 بيكسل عن المسار الأول و30 بيكسل عن المسار الثاني، ثم سنقتطع مقاسات الفواصل، لدينا أربعة فواصل عمودية كل واحد 5 بيكسل إذاً ستقتطع 20 بيكسل، بقي لدينا 490 بيكسل، سنقتطع المقاس الأدنى للعناصر المضبوط لها auto، سنقتطع 22px عن المسار الثالث، بقي لدينا 468 بيكسل سنقسمها على مجموع واحدات fr لنحصل على حصة كل fr، لدينا 3 واحدات fr واحدة عن المسار الرابع واثنتين عن المسار الأخير 156=3/468 لكل واحدة fr، إذاً الرابع سيصبح 156 والخامس 312=2*156

5. Max-content: ضبط المسار لأصغر مقاس لعرض المحتوى بدون التفاف warp .
6. Min-content : ضبط المسار لأصغر مقاس لعرض المحتوى مع التفاف warp وبشرط أن لا يُحدِث overflow للمحتوى.

```
grid-template-columns: min-content max-content min-content max-content ;
```

7. function Minmax (min , max) : دالة لها معاملين تقوم بضبط مقاس المسار ليكون ضمن نطاق أكبر أو يساوي المعامل الأول وأصغر أو يساوي المعامل الثاني.
- المعاملان : min القيمة الأدنى للنطاق ، max القيمة الأقصى للنطاق.
- سلوك الدالة : الدالة ستقوم بضبط مقاس المسار للقيمة الأقصى أولاً، وفي حالة عدم توافر المقاس المناسب ستسعى نحو القيمة الأدنى.

- كلا المعاملين يمكن أن يكون :

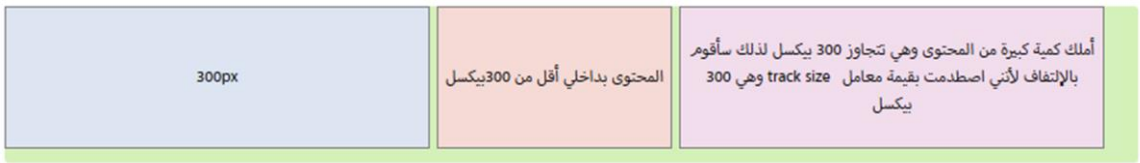
Length , ٪ , auto , fr , min-content , max-content باستثناء min التي لا يمكن أن تُسند للقيمة fr.

- إذا كانت قيمة max أقل من قيمة min سيتم تجاهل max وستعامل الدالة قيمة min للمعاملين.
- لا يمكن استعمال دالة minmax داخل دالة minmax أخرى.
- الصيغة minmax(min-content , max-content) تجعل مقاس المسار مُعتمد على المحتوى بداخله بحيث لن يتقلص لأقل من أصغر مقاس لعرض المحتوى مع التفاف ، ولن يتمدد لأكثر من أصغر مقاس لعرض المحتوى بدون التفاف warp.يعني نحصل على حسنات min-content وحسنات max-content في صيغة واحدة.

8. function (track size) fit-content : دالة معتمدة على المحتوى لها حدين ومعامل.

- الحدين : الحد الأول هو القيمة الأدنى لمقاس المسار ويأخذ القيمة min-content ، الحد الثاني هو القيمة الأقصى لمقاس المسار ويأخذ القيمة max-content.
- معامل track size عند ضبطه إلى قيمة معينة مقاس المسار لن يتجاوزها مهما زاد المحتوى، وفي حالة زيادة المحتوى سيحصل له التفاف wrap.
- سلوك الدالة : الدالة ستقوم بضبط مقاس المسار للحد الأدنى أولاً، وفي حالة زيادة المحتوى ستسعى نحو إحدى القيمتين(قيمة معامل track size أو قيمة الحد الأقصى) وهنا لدينا عدة حالات.
  - قيمة معامل track size أكبر من قيمة الحد الأقصى، الدالة ستسعى نحو قيمة الحد الأقصى و مقاس المسار لن يتمدد لأكثر من ذلك.
  - قيمة معامل track size أصغر من قيمة الحد الأقصى، الدالة ستصطدم بقيمة معامل track size وسيحدث التفاف للمحتوى.

```
width: 800px;
grid-template-columns:300px fit-content(300px) fit-content(300px) ;
```



المسار الثاني لم يسعى لـ300 بيكسل لأن الحد أقصى أقل من قيمة معامل track size وهي 300بيكسل.

المسار الثالث اصطدم بقيمة معامل track size لأنها أقل من الحد الأقصى، هنا الحد الأقصى أصبح أكبر لأن المحتوى زاد.

9. دالة repeat() : دالة تكرار بُنيتها على الشكل (المعامل الثاني،المعامل الأول) repeat

لها ثلاث صيغ مختلفة :

- الصيغة الأولى (لائحة المسارات ، عدد التكرار) repeat
- الصيغة الثانية (مقاس المسار ، repeat (auto-fit
- الصيغة الثالثة (مقاس المسار ، repeat (auto-fil

تتفق الصيغ الثلاثة في أنها يمكن أن تمثل بمفردها لائحة المسارات أو يمكن أن تمثل جزء من لائحة المسارات، وأنه لا يمكن وضع دالة التكرار ضمن دالة تكرار أخرى.


```
grid-template-columns:100px repeat(6,50px);

grid-template-columns:50px repeat(auto-fit,50px);

grid-template-columns:50px repeat(auto-fill,50px);
```

في جميع هذه العبارات مثلت دالة التكرار جزء من لائحة المسارات، يعني لائحة ضمن لائحة.

```
grid-template-columns: repeat(6,50px);

grid-template-columns: repeat(auto-fit,50px);

grid-template-columns: repeat(auto-fill,50px);
```

في جميع هذه العبارات مثلت دالة التكرار لائحة المسارات.

الصيغة الأولى : (لائحة المسارات ، عدد التكرار) repeat

تُكرر هذه الصيغة لائحة المسارات لأي عدد من المرات.

```
grid-template-columns: repeat(2,50px 1fr 100px);
```

تم تكرار لائحة المسارات 50px 1fr 100px لمرتين.

الصيغة الثانية : (مقاس المسار ، auto-fit) repeat

الصيغة الثالثة : (مقاس المسار ، auto-fil) repeat

ماذا سيحدث لو أن مجموع مقاسات المسارات أكبر من مقاس حاوية الشبكة يعني overflow، وماذا سيحدث لو أن مجموع مقاسات المسارات أصغر من مقاس حاوية الشبكة، تعالج صيغتا التكرار هاتين الحالتين عن طريق إنشاء مسارات ضمنية جديدة ولكن تختلفان في التعامل مع هذه المسارات الجديدة.

الحالة الأولى مجموع مقاسات المسارات أكبر من مقاس حاوية الشبكة overflow :

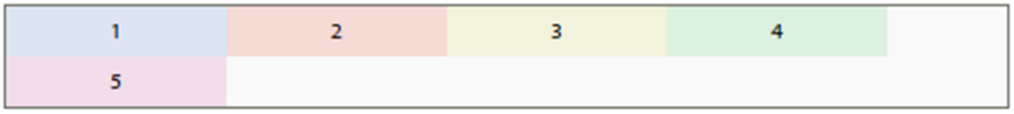
ليكن لدينا مقاس حاوية الشبكة 500px و 5 عناصر شبكة grid items تم إنشاء مسار عمودي لكل عنصر بحيث يكون مقاس المسار الواحد 110px بالصيغة

repeat (auto-fit , 110px) أو بالصيغة repeat (auto-fill , 110px)

نلاحظ أنه سيتم تقسيم مقاس حاوية الشبكة بحيث تتسع لأكبر عدد من المسارات دون overflow و عناصر الشبكة الزائدة سيتم إنشاء مسار أفقي ضمني جديد لتنزلق فيه أي سيحدث لها التفاف wrap ، تتفق الصيغتان على هذه السلوك.

```
/////HTML
<div class="grid_container">
   <div>1</div>
   <div>2</div>
   <div>3</div>
   <div>4</div>
   <div>5</div>
 </div>
```

```
/////CSS
.grid_container{
  width: 500px;
  grid-template-columns:repeat(auto-fit , 110px) ;
}

///OR
.grid_container{
  width: 500px;
  grid-template-columns:repeat(auto-
fill , 110px) ;
}
```



في المثال السابق تم تقسيم مقاس حاوية الشبكة بحيث تتسع لأكبر عدد من المسارات دون overflow يعني 110/500= 4.5454 أي أن الشبكة تتسع لـ 4 مسارات عمودية في كل منها عنصر شبكة أي أول 4 عناصر، والعنصر الأخير سيتم إنشاء مسار أفقي ضمني جديد لينزلق فيه.

يجب التنويه أنه لوكان يوجد مسار أفقي صريح آخر لانزلقت العناصر فيه ولما تم إنشاء مسار ضمني أفقي.

الحالة الثانية مجموع مقاسات المسارات أصغر من مقاس حاوية الشبكة:

ليكن لدينا مقاس حاوية الشبكة 500px و 5 عناصر شبكة grid items تم إنشاء مسار عمودي لكل عنصر بحيث يكون مقاس المسار الواحد 80px بالصيغة

repeat (auto-fit , 80px) أو بالصيغة repeat (auto-fill , 80px)

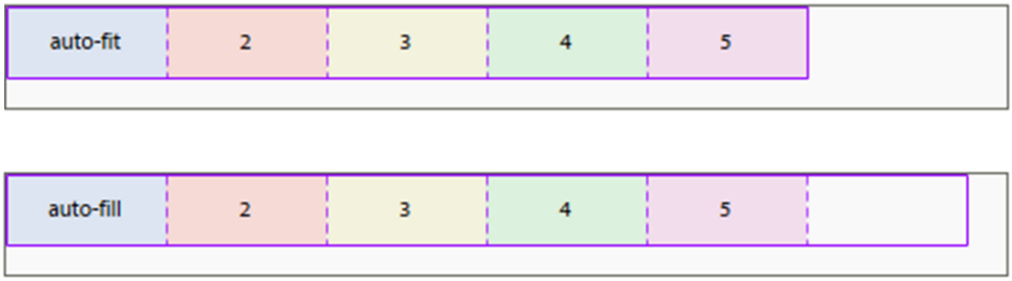
نلاحظ أنه سيتم تقسيم مقاس حاوية الشبكة بحيث تتسع لأكبر عدد من المسارات دون overflow هنا لدينا حالتين هل لدينا عناصر شبكة grid items لتشغل هذه المسارات أم لا ؟

auto-fit كل مسار فارغ أو لم يجد عنصر شبكة ليشغله سيكون مقاس مساره 0px .

auto-fill كل مسار سيُحجز له مكان حتى لو كان فارغاً أو لم يجد عنصر شبكة ليشغله .

```
/////HTML
<div class="grid_container">
   <div>1</div>
   <div>2</div>
   <div>3</div>
   <div>4</div>
   <div>5</div>
 </div>
```

```
/////CSS
.grid_container{
  width: 500px;
  grid-template-columns:repeat(auto-fit , 80px) ;
}

///OR
.grid_container{
  width: 500px;
  grid-template-columns:repeat(auto-fill , 80px) ;
}
```

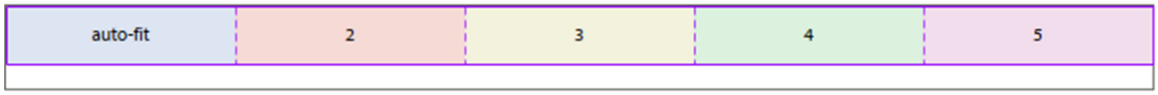


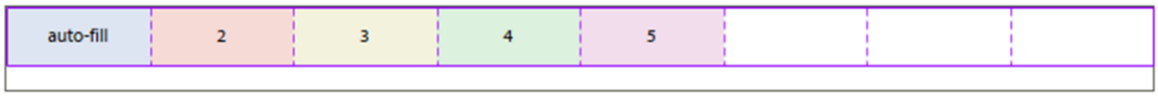
في المثال السابق تم تقسيم مقاس حاوية الشبكة بحيث تتسع لأكبر عدد من المسارات دون overflow يعني 80/500= 6.25 أي أن الشبكة تتسع لـ 6 مسارات عمودية في أول 5 مسارات وُضِعَ عنصر شبكة في كل مسار، أما المسار السادس بقي فارغاً، auto-fit ستجعل مقاس هذا المسار الفارغ 0px، بينما auto-fill ستحجز له مكان حتى لو كان فارغاً.

سيبدو هذا السلوك أوضح إذا جعلنا مقاس حاوية الشبكة 700px و مقاس المسار الواحد minmax(80px,1fr) بالصيغة repeat (auto-fit , minmax(80px,1fr))

أو بالصيغة repeat (auto-fill , minmax(80px,1fr))

```
/////CSS
.grid_container{
  width: 700px;
  grid-template-columns:repeat(auto-fit , minmax(80px ,1fr)) ;
}

///OR
.grid_container{
  width: 700px;
  grid-template-columns:repeat(auto-fill , minmax(80px ,1fr)) ;
}
```





تم تقسيم مقاس حاوية الشبكة بحيث تتسع لأكبر عدد من المسارات دون overflow يعني 80/700= 8.75 أي أن الشبكة تتسع لـ 8 مسارات عمودية في أول 5 مسارات وُضِعَ عنصر شبكة في كل مسار، أما المسارات الثلاثة الأخيرة بقيت فارغة

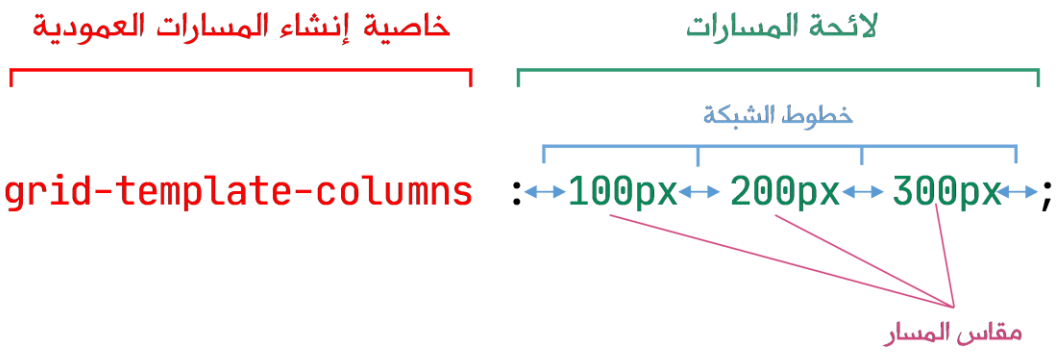
auto-fit ستجعل مقاس هذا المسارات الفارغة 0px، بينما auto-fill ستحجز لها مكان حتى لو كانت فارغة.

تكلمنا عن صيغتي التكرار مع المسارات العمودية تنطبق هذه الشروط أيضاً على المسارات الأفقية أيضاً مع مراعاة اتجاه توليد المسارات الجديدة في خاصية grid-auto-flow سنتحدث عنها بالفصول القادمة.

تسمية خطوط الشبكة

لقد قلنا في قسم ضبط لائحة المسارات أنه بواسطة لائحة المسارات نستطيع ضبط عدد المسارات ، مقاس كل مسار ، تسمية خطوط الشبكة.

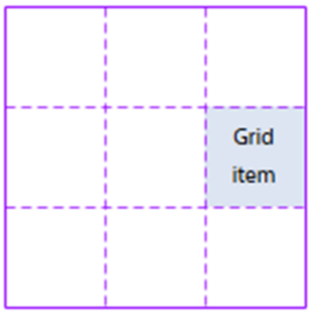
لقد تكلمنا عن ضبط عدد المسارات وضبط مقاس كل مسار سنتكلم الآن عن تسمية خطوط الشبكة، لنعد أولاً للائحة المسارات حيث قلنا أنها لائحة من القيم المفصولة بمسافة، عدد القيم يُمثل عدد المسارات التي سيتم إنشاؤها، كل قيمة من هذه اللائحة تُمثل مقاس المسار الذي سيتم إنشاؤه، وكل مسافة على جانبي كل قيمة تُمثل خط شبكة.



إذا تُركت هذه المسافات فارغة سيتم ترقيم الخطوط آلياً بحسب اتجاه العنصر بحيث يكون كل خط عمودي بالأعداد الموجبة مع اتجاه العنصر وبالأعداد السالبة عكس اتجاه العنصر، وكل خط أفقي بالأعداد الموجبة بالاتجاه من الأعلى للأسفل و بالأعداد السالبة من الأسفل للأعلى، تحدثنا عن الترقيم الآلي في فصل مصطلحات الشبكة.

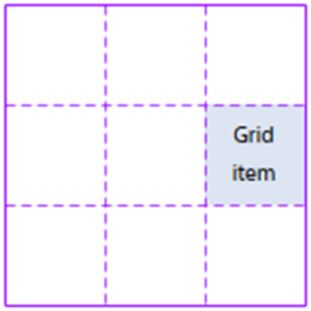
تستعمل خطوط الشبكة كخطوط إرشادية لتعيين موقع العنصر على الشبكة كأن نقول أن العنصر يقع بين الخطين العموديين الثالث والرابع، و بين الخطين الأفقيين الثاني والثالث.

```
.item1{
  grid-column-start: 3;
  grid-column-end: 4;
  grid-row-start: 2;
  grid-row-end: 3;
}
```



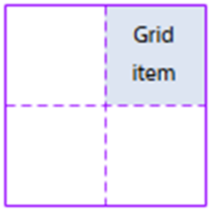
يمكن تسمية خطوط الشبكة يدوياً للإشارة إليها بدلاً من الأرقام وذلك بوضع اسم الخط ضمن أقواس مربعة[ ] في لائحة المسارات.

```
.grid_container{
  grid-template-columns:[col1start]50px [col1end] 50px[col2end] 50px [col3end] ;
  grid-template-rows: [row1start]50px [row1end] 50px[row2end] 50px [row3end];
}
.item1{
  grid-column-start: col2end;
  grid-column-end: col3end;
  grid-row-start: row1end;
  grid-row-end: row2end;
}
```



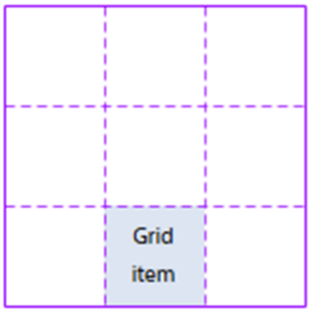
نلاحظ من المثال السابق أن الخط المُسمى [col1end] هو خط مشترك بين المساريين العموديين الأول و الثاني فهو يمثل نهاية المسار الأول وبداية المسار الثاني لذلك نستطيع تسمية الخط الواحد بأكثر من اسم وذلك بوضع مسافة بين الاسمين ضمن الأقواس المربعة [firstName secondName] .

```
.grid_container{
  grid-template-columns:[col1start]50px [col1end col2start] 50px[col2end] ;
  grid-template-rows: [row1start]50px [row1end row2start] 50px[row2end] ;
}
.item1{
  grid-column-start: col2start;
  grid-column-end: col2end;
  grid-row-start: row1start;
  grid-row-end: row1end;
}
```



يمكن تكرار أسماء الخطوط بدالة التكرار أيضاً و تتم الإشارة إليها باسم الخط وبجانبه رقم ترتيب المسار.

```
.grid_container{
  grid-template-columns:repeat(3,[colstart] 50px [colend]) ;
  grid-template-rows: repeat(3,[rowstart] 50px [rowend]) ;
}
.item1{
  grid-column-start: colstart 2;
  grid-column-end: colend 2;
  grid-row-start: rowstart 3;
  grid-row-end: rowend 3;
}
```



إنشاء المسارات الضمنية

نستطيع تعريف وإنشاء المسارات بشكل صريح وذلك بضبط عددها ومقاساتها إلى خاصيات إنشاء المسارات الصريحة، وتسمى الشبكة الناتجة شبكة صريحة Explicit Grid بعد إنشاء الشبكة الصريحة قد تحدث معنا حالات مثل زيادة عدد العناصر عن ما يمكن للشبكة الصريحة أن تستوعبه، ستقوم الشبكة آلياً أيضاً بتوليد مسارات إضافية جديدة ضمنية لاحتواء العناصر، مقاسات هذه المسارات الضمنية معتمد على محتوى العناصر، يمكن التحكم بمقاسات المسارات الضمنية المُوَلَّدَة واتجاه التوليد، وتسمى الشبكة الناتجة الشبكة الضمنية Implicit Grid .

- متى يتم توليد المسارات الضمنية ؟

- عندما لا يتم التصريح عن المسارات الصريحة .
- احتواء العناصر الزائدة عن استيعاب الشبكة الصريحة .
- ضبط موقع أحد العناصر خارج حدود الشبكة الصريحة .
- تمدد أحد العناصر لخارج حدود الشبكة الصريحة.

- يتم التحكم بمقاسات المسارات الضمنية المُوَلَّدَة بالخاصيتين :

| خاصية تتحكم بمقاس المسارات العمودية Grid-auto-columns
| خاصية تتحكم بمقاس المسارات الأفقية Grid-auto-rows

بحيث يمكن أن يكون مقاس المسار :

Length , ٪ , autu(default) , fr ,min-content , max-content , minmax

fit-content, يعني كل المقاسات ماعدا دالة التكرار .

- يتم التحكم باتجاه توليد المسارات الجديدة بالخاصية Grid-auto-flow فإما أن يتم توليد مسارات أفقية وتأخذ القيمة row وهي القيمة الافتراضية، أو يتم توليد مسارات عمودية وتأخذ القيمة column، بالإضافة يمكن لهذه الخاصية أن تضبط طريقة التعبئة التلقائية في خوارزمية التموقع التلقائي Auto-Placement Algorithm التي سنتحدث عنها لاحقاً، ولكن يمكن القول إلى أنها يمكن أن تكون إحدى القيمتين sparse أو dense حيث sparse هي القيمة الافتراضية ويمكننا الآن الإشارة فقط إلى أن سلوك هذه القيمة يتمثل بالشعار التالي "الحفاظ على ترتيب العناصر هو الأهم" بينما القيمة dense تسلك سلوك يتمثل بالشعار "يجب ملئ الفراغات" ، و يجب التنبيه أيضاً إلى أن القيمة sparse هي القيمة الافتراضية، و بحسب آخر مواصفة لا يمكن كتابتها أو إسنادها للخاصية Grid-auto-flow يدوياً ، أي يمكن إسناد القيمة dense للخاصية Grid-auto-flow وإذا أردت التغيير إلى القيمة sparse احذف القيمة dense .

لتلخيص ما سبق يمكن القول أن الخاصية Grid-auto-flow يمكن أن تأخذ القيم :

Grid-auto-flow	One Keyword	row(default)
		column
		dense
	Two Keyword	row dense
		column dense

بعيداً عن المسارات الضمنية فإن ضبط الخاصية Grid-auto-flow لإحدى القيمتين row أو column له وظيفة أخرى أيضاً وهي طريقة تدفق العناصر، سنتوسع فيها بالصفحة المقبلة، وسبب عدم ذكرنا لهذه الوظيفة بالبداية صراحةً يرجع لسببين أن خاصيات الضابطة للشبكة تتداخل بشدة فيما بينها وهذا يُصعب عملية الشرح، و أيضاً ستجد من خلال تصفحك لمختلف المصادر التي تشرح الشبكة أنه بالإمكان شرح الشبكة بأكثر من طريقة مختلفة تماماً بالطرح، ولكن جميعها تصطدم بالحقيقة أنها ستصل إلى مرحلة يصبح فيها من الصعب المحافظة على التسلسل المنطقي للأفكار.

Grid-auto-flow

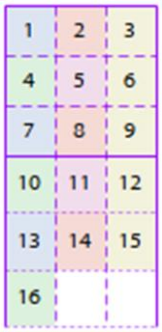
يمكن القول أن هذه الخاصية تضبط ثلاث وظائف، وظيفتين يتم تحديدهما بـ row أو column ، و وظيفة مختصة بطريقة التعبئة التلقائية في خوارزمية التموقع التلقائي يتم تحديدها بـ sparse أو dense سنتحدث عنها لاحقاً.

الوظيفتين اللتين يتم تحديدهما بـ row أو column .

عند ضبط Grid-auto-flow لإحدى القيمتين row أو column فأننا نضبط في نفس الوقت طريقة ملئ الخلايا و اتجاه توليد المسارات الجديدة.

Grid-auto-flow : row

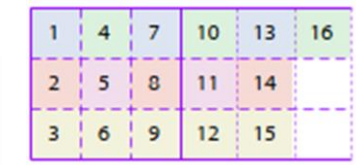
```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;
  grid-auto-flow: row;
}
```



نلاحظ أننا أنشأنا 3 مسارت عمودية صريحة و 3 مسارات أفقية صريحة، مما يعني أنه لدينا 9 خلايا متاحة، لكن بالمقابل لدينا 16 عنصر فماذا فعلت الشبكة ؟ بما أن الخاصية Grid-auto-flow مضبوطة إلى row ستقوم الشبكة بوظيفتين أولاً ستقوم بملء المسارات الأفقية أولاً لذلك نجد أن المسار الأفقي الأول فيه العناصر 1 و 2 و 3 ثم انتقلت للمسار الأفقي الثاني ونجد فيه 4 و 5 و 6 ثم انتقلت للمسار الأفقي الأخير ونجد فيه 7 و 8 و 9 هنا انتهت المسارات الصريحة لذلك ننتقل للوظيفة الثانية لخاصية Grid-auto-flow وهي توليد المسارات وبما أنها مضبوطة إلى row ستقوم بتوليد مسارات أفقية ضمنية جديدة حتى استيعاب كل العناصر.

Grid-auto-flow : column

```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;
  grid-auto-flow: column;
}
```



نلاحظ أننا أنشأنا 3 مسارت عمودية صريحة و 3 مسارات أفقية صريحة، مما يعني أنه لدينا 9 خلايا متاحة، لكن بالمقابل لدينا 16 عنصر فماذا فعلت الشبكة ؟ بما أن الخاصية Grid-auto-flow مضبوطة إلى column ستقوم الشبكة بوظيفتين أولاً ستقوم بملء المسارات العمودية أولاً لذلك نجد أن المسار العمودي الأول فيه العناصر 1 و 2 و 3 ثم انتقلت للمسار العمودي الثاني ونجد فيه 4 و 5 و 6 ثم انتقلت للمسار العمودي الأخير ونجد فيه 7 و 8 و 9 هنا انتهت المسارات الصريحة لذلك ننتقل للوظيفة الثانية لخاصية Grid-auto-flow وهي توليد المسارات وبما أنها مضبوطة إلى column ستقوم بتوليد مسارات عمودية ضمنية جديدة حتى استيعاب كل العناصر.

إنشاء المناطق

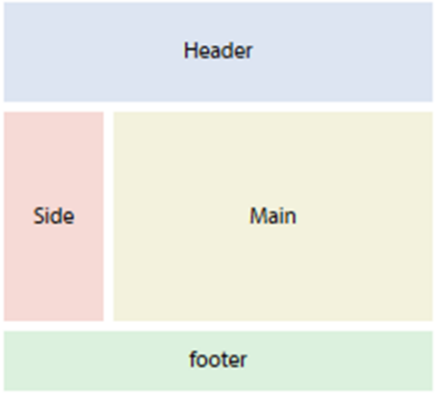
بعد إنشاء المسارات يمكننا ضبط موقع وتمدد كل عنصر من العناصر على الشبكة و ذلك بالإشارة إلى خطوط الشبكة التي يقع ضمنها، مثلاً العنصر الأول يقع بين الخطين العموديين الأول و الثالث والخطيين الأفقيين الثاني والثالث، بينما يقع العنصر الثاني بين الخطين العموديين الثالث والرابع و الخطيين الأفقيين الرابع والخامس.

```
.grid_container{
  grid-template-columns:repeat(3,50px) ;
  grid-template-rows:repeat(4,50px) ;
}
.item1{
  grid-column-start:1 ;
  grid-column-end:3 ;
  grid-row-start:2 ;
  grid-row-end:3 ;
}
.item2 {
  grid-column-start:3 ;
  grid-column-end:4 ;
  grid-row-start:4 ;
  grid-row-end:5 ;
}
```

هذا يسمى التموقع المُعتمد على خطوط الشبكة، يوجد أيضاً طريقة أخرى وهي بإنشاء قالب للشبكة عن طريق كتابة أسماء مناطق مرجعية تتم الإشارة إليها بالأبناء، بحيث يتم التصريح عن القالب بالأب(حاوية الشبكة) ومن ثم ربط كل عنصر باسم المنطقة الموافقة.

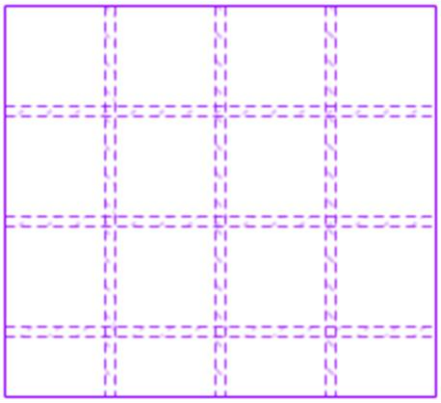
1.التصريح عن القالب

ليكن لدينا التصميم التالي ونريد أن نُصرح عن قالبه.



بداية لنُصرح عن المسارات و الفواصل .

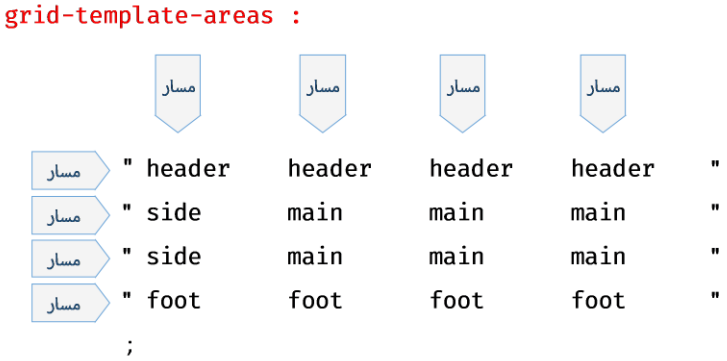
```
.grid_container{
  grid-template-columns:repeat(4,50px) ;
  grid-template-rows:repeat(3,50px) 30px ;
  gap:5px;
}
```



ثم نُصرح عن القالب أو أسماء المناطق ويتم ذلك عبر الخاصية Grid-template-areas.

```
grid-template-areas:
  "header header header header"
  "side   main   main   main  "
  "side   main   main   main  "
  "foot   foot   foot   foot  "
  ;
```

توفر هذه الصيغة تصور بصري للشبكة.



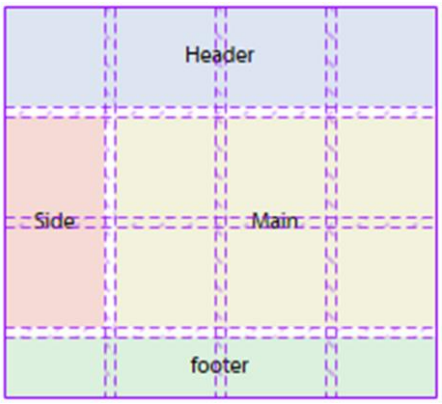
قواعد يجب مراعاتها عند استخدام القالب:

- **كل اسم يمثل خلية.**
- **تكرار الاسم يعني أن المنطقة تمتد على أكثر من خلية و العنصر الموافق سيتمدد ليغطي كامل المنطقة.**
- **وضع نقطة (.) بدلاً من اسم منطقة يعني أن هذه الخلية ستكون فارغة، و ستعامل النقاط الموضوعة بجانب بعضها البعض مهما كان عددها كنقطة واحدة طالما لا يفصل بينها مسافة مثلا (...) أو (.....)**
- **المناطق دائماً مستطيلة أو مربعة، لا يمكن إنشاء منطقة على شكل (L)مثلا.**
- **كل سطر بين علامتي التنصيص " " يجب أن يكون بنفس عدد الأسماء و إلا ستتعطل الشبكة.**

2.الربط مع الأبناء(عناصر الشبكة)

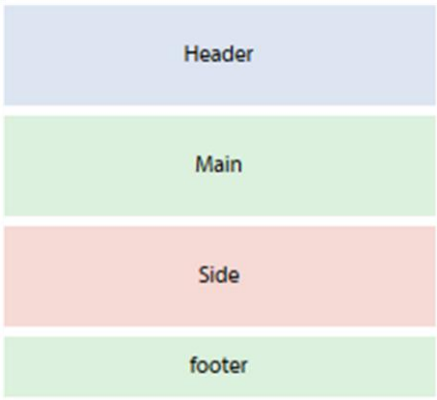
يتم ربط كل عنصر مع المنطقة الموافقة بالإشارة إلى اسم المنطقة الموافقة بخاصية grid-area في الابن.


```
.grid_container{
  grid-template-columns:repeat(4,50px) ;
  grid-template-rows:repeat(3,50px) 30px ;
  gap:5px;
  grid-template-areas:
    "header header header header"
    "side   main   main   main  "
    "side   main   main   main  "
    "foot   foot   foot   foot  ";
}
.item1 {
  grid-area: header;
}
.item2 {
  grid-area: side;
}
.item3 {
  grid-area: main;
}
.item4 {
  grid-area: foot;
}
```

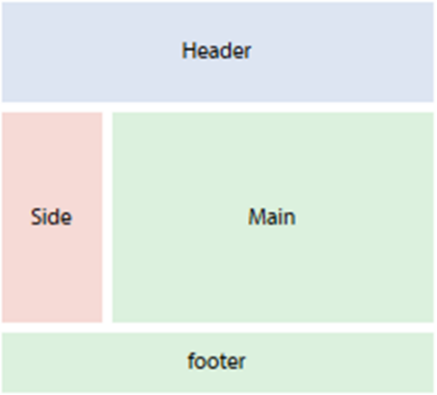


تفيد هذه الطريقة في ضبط مواقع العناصر في استعلامات الوسائط(Media Queries) بكل سهولة فكل ما عليك فعله هو تغيير ترتيب القالب في كل استعلام مختلف.

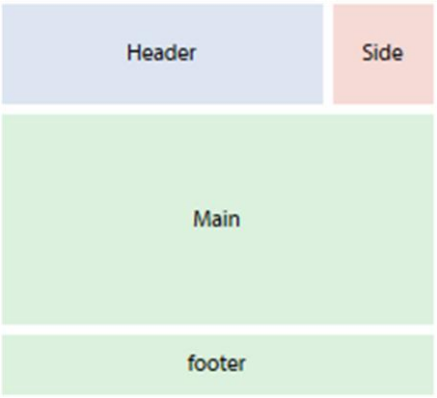
```
.grid_container{
 grid-template-columns:repeat(4,50px) ;
 grid-template-rows:repeat(3,50px) 30px ;
 gap:5px;
 grid-template-areas:
  "header  header  header  header "
  "main    main    main    main   "
  "side    side    side    side   "
  "foot    foot    foot    foot   "
}
.item1 {
 grid-area: header;
}
.item2 {
 grid-area: side;
}
.item3 {
 grid-area: main;
}
.item4 {
 grid-area: foot;
}
```




```
@media screen and (min-width: 769px) {
  .grid_container {
   grid-template-areas:
   "header  header  header  header "
   "side    main    main    main   "
   "side    main    main    main   "
   "foot    foot    foot    foot   "
   ;
  }
 }
```




```
@media screen and (min-width: 992px) {
  .grid_container {
   grid-template-areas:
   "header  header  header  side  "
   "main    main    main    main  "
   "main    main    main    main  "
   "foot    foot    foot    foot  "
   ;
  }
```



إنشاء الفواصل

هي مسافات تفصل بين المسارات، يتم ضبطها لتفصل بين المسارات العمودية أو الأفقية أو الإثنين معاً، لا يمكن وضع أي محتوى داخل الفواصل كما لا يمكن الفصل بين حواف حاوية الشبكة الداخلية والعناصر الأبناء بواسطة الفواصل استعمل الحشوة padding بدلاً من ذلك.

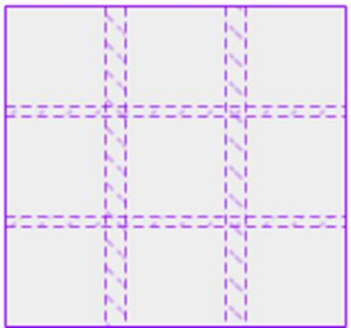
يتم ضبط مقاس الفاصل بالخاصيات:

- **الفاصل بين المسارات العمودية column-gap.**
- **الفاصل بين المسارات الأفقية row-gap.**
- **الفاصل بين المسارات الأفقية والعمودية معاً gap.**

و يمكن أن يأخذ مقاس الفاصل القيم التالية : ٪ , length

إذا أسندت gap لقيمة واحدة فهي تمثل الفاصل بين المسارات الأفقية والعمودية معاً، أما إذا أسندت لقيمتين فالأولى هي الفاصل بين المسارات الأفقية والثانية الفاصل بين المسارات العمودية.

```
width: 170px;
height: 160px;

column-gap: 10px;
row-gap: 5px;

///OR
gap: 5px 10px;

///OR
gap: 3.125%  5.882%;
```



وجب التنويه أن الخاصيات row-gap , column-gap , gap كانت سابقاً grid-gap grid-column-gap , grid-row-gap وتم تجديدها في آخر مواصفة .

خاصية grid المُختصرَة

هي التي تضبط جميع الخاصيات التي تُحدّد خصائص الشبكة الصريحة (أي grid-template-rows و grid-template-columns و grid-template-areas)، وجميع الخاصيات التي تُحدِّد خصائص الشبكة ضمنيًا (أي grid-auto-rows و grid-auto-columns و grid-auto-flow)، والخاصيات التي تحدد الفواصل (أي column-gap و row-gap)، وكل ذلك في قاعدة واحدة.

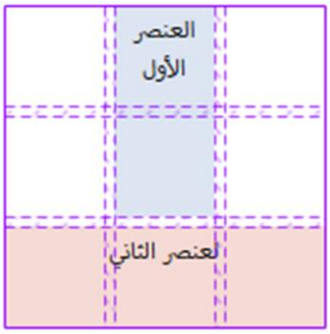
يمكن تحديد القيم التي تُحدِّد خصائص الشبكة الصريحة (explicit) أو التي تُحدِّدها ضمنيًا (implicit) في قاعدة grid ، والخاصيات التي لا تُحدِّد قيمتها ستُضبَط إلى القيمة الابتدائية (كما هو معتاد في الخاصيات المختصرة).

التحكم بالتموضع

قبل البدء بالتحكم بالتموضع يجب التفريق بين مفهومي التحكم بالموقع والتحكم بالتموضع بحسب الشبكة، فالتحكم بالموقع هو ضبط عنصر الشبكة Grid Item للمكان الذي سيظهر فيه على الشبكة اعتماداً على خطوط الشبكة أو بالإشارة إلى أسماء المناطق، أما التحكم بالتموضع فهو ضبط العلاقة بين حدود العنصر وحدود الموقع الذي أُسند إليه، كأن نضبط تموضع العنصر ليكون في منتصف الموقع الذي أُسند إليه.

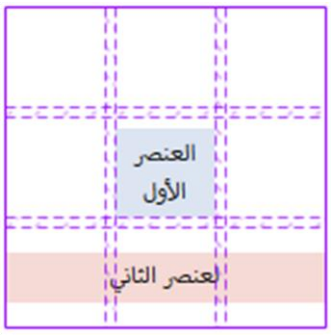
ليكن لدينا عنصرين نريد ضبط موقع العنصر الأول اعتماداً على خطوط الشبكة ليكون بين الخطيين العموديين الثاني والثالث وبين الخطين الأفقيين الأول والثالث، ونريد ضبط موقع العنصر الثاني بالإشارة لاسم المنطقة حيث سيأخذ موقع المنطقة المسماة two .

```
.grid_container{
  grid-template-columns:repeat(3,50px) ;
  grid-template-rows:repeat(3,50px) ;
  grid-template-areas:
    "header header header "
    "side   main   main   "
    "two    two    two    "
    ;
  gap:5px;
}
.item1 {
  grid-column-start: 2;
  grid-column-end: 3;
  grid-row-start: 1;
  grid-row-end: 3;
}
.item2 {
  grid-area: two;
}
```



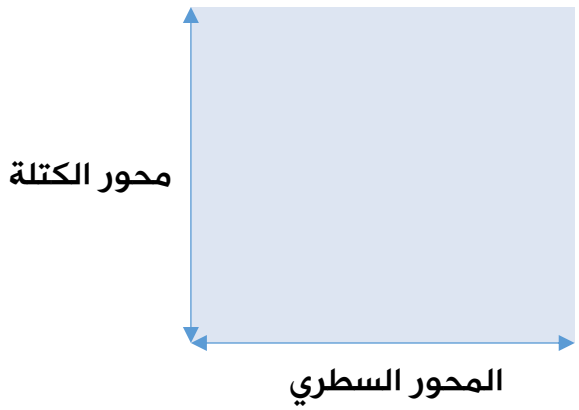
بعد أن ضبطنا موقعي العنصرين لنضبط تموضعهما ضمن المواقع المسندة إليها، حيث سيكون تموضع العنصر الأول في نهاية الموقع المسند إليه من الأسفل، بينما سيكون العنصر الثاني في منتصف الموقع المسند إليه.

```
.item1 {
  align-self: end;
}
.item2 {
  align-self: center;
}
```



نعلم أن تخطيط Grid يقدم شبكة ثنائية الأبعاد تتعامل مع الصفوف والأعمدة في نفس الوقت، لذلك عند التحكم بالتموضع يجب الانتباه إلى محورين يجب التعامل معهما:

محور الكتلة(Block-axis) ويقال له أيضاً محور العمود(Column-axis)، والمحور السطري(Inline-axis) ويقال له محور الصف(Row-axis).



عند التكلم عن ضبط التموضع لدينا نوعين من الخاصيات، خاصيات تضبط تموضع المسارات نسبةً إلى حاوية الشبكة، وخاصيات تضبط تموضع العناصر نسبةً للمواقع المسندة إليها.

خاصيات ضبط تموضع المسارات :

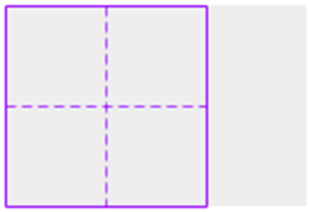
- ضبط تموضع المسارات على محور الكتلة align-content .
- ضبط تموضع المسارات على المحور السطري justify-content .
- ضبط تموضع المسارات على المحورين معاً place-content .

خاصيات تضبط تموضع العناصر:

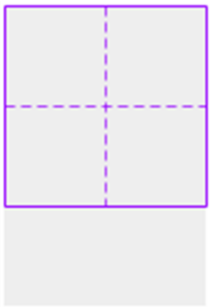
- ضبط تموضع العناصر على محور الكتلة align-items , align-self .
- ضبط تموضع العناصر على المحور السطري justify-items , justify-self .
- ضبط تموضع العناصر على المحورين معاً place-items , place-self .

ضبط تموضع المسارات

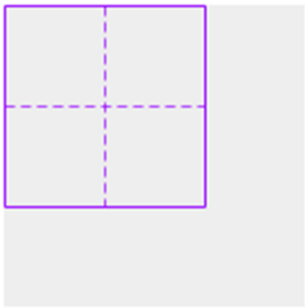
خاصيات ضبط تموضع المسارات تضبط تموضع المسارات بالنسبة لحدود حاوية الشبكة ونستطيع التحكم بتموضع المسارات عندما يكون مجموع مقاسات المسارات لا تتوافق مع أبعاد حاوية الشبكة مثلاً:



عرض حاوية الشبكة 150بيكسل، ومجموع مقاسات المسارات العمودية 100بيكسل.



ارتفاع حاوية الشبكة 150بيكسل، ومجموع مقاسات المسارات الأفقية 100بيكسل.

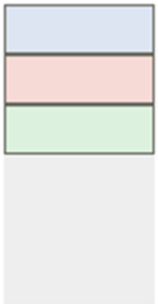


ارتفاع وعرض حاوية الشبكة 150بيكسل، ومجموع مقاسات المسارات الأفقية 100بيكسل، ومجموع مقاسات المسارات العمودية 100بيكسل.

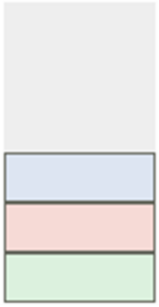
• ضبط تموضع المسارات على محور الكتلة align-content •

تأخذ هذه الخاصية عدة قيم .

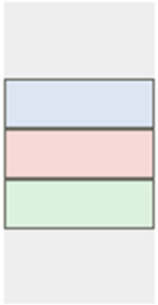
start (default)



end

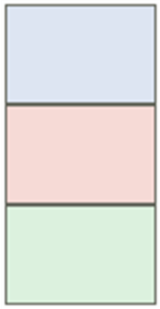


center



stretch

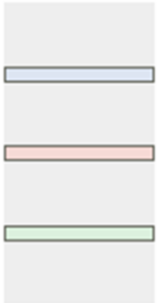
لن تعمل إلا إذا كان
مقاس المسارات
مضبوط إلى auto



Space-around

Space-between

Space-evenly

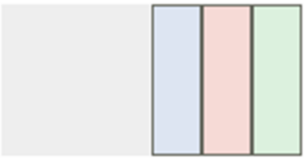


- ضبط تموضع المسارات على المحور السطري justify-content

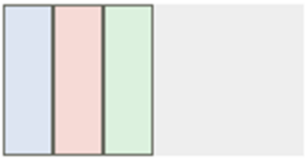
تأخذ هذه الخاصية عدة قيم .

center

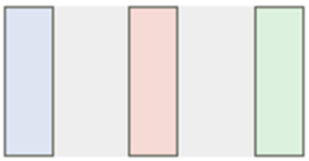
end



start (default)

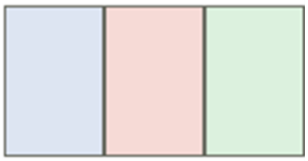


Space-between



Space-around

stretch



Space-evenly

• ضبط تموضع المسارات على المحورين معاً place-content •

تضبط هذه الخاصية المُختصرَة تموضع المسارات على المحورين معاً وتأخذ الشكل:

```
Place-content : align-content / justify-content
Place-content : start center
```

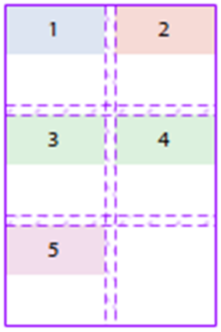
في حال تجاهل القيمة الثانية ستعتبر القيمة الأولى للخاصيتين align-content و justify-content معاً.

ضبط تموضع العناصر

خاصيات ضبط تموضع العناصر تضبط تموضع العناصر بالنسبة لحدود الموقع من الشبكة الذي أسند إليه كل عنصر و للتحكم بتموضع العناصر هنالك طريقتين، فنستطيع أن نضبط تموضع العناصر بخاصيات التموضع في تعريف الكتلة الحاوية والتموضع سيُطبق على جميع العناصر، ونستطيع ضبط موضع كل عنصر بشكل فردي في تعريف كل عنصر على حدة، مع ملاحظة أن ضبط التموضع الفردي سيتجاوز override ضبط التموضع العام، و بشكل عام الخاصية التي في نهايتها self– هي للضبط الفردي.

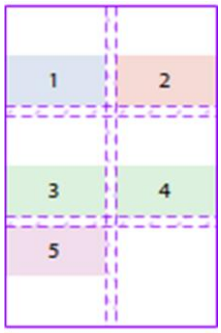
- ضبط تموضع العناصر على محور الكتلة align-items , align-self

start



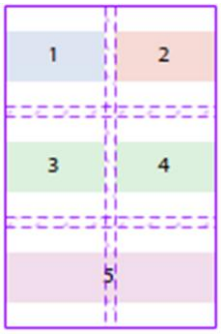
end

```
.item5 {
 align-self: start;
}
```



center

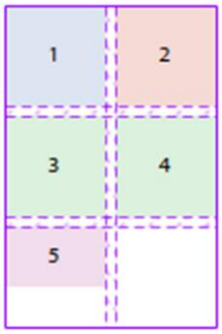
```
.item5 {
  grid-column-start: 1;
  grid-column-end: 3;
}
```



stretch

لن تعمل إلا إذا كان مقاس العناصر مضبوط إلى auto

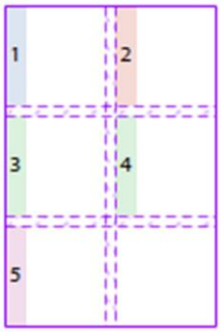
```
.item5 {
   height: 30px;
```



- نلاحظ أن القيم ماعدا stretch ستُظهر العناصر بالمقاس الأدنى.
- Stretch هي القيمة الافتراضية، وستعمل على أن تتمدد العناصر لتشغل كامل الخلية، إلا إذا كانت أبعاد العنصر مضبوطة لقيم محسوبة.
- إذا تمدد العنصر على أكثر من خلية سيتم تطبيق التموضع على ما يمتد عليه.

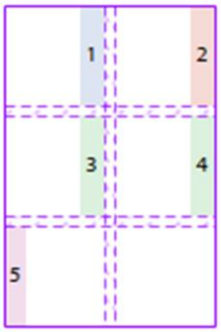
• ضبط تموضع العناصر على المحور السطري justify-items , justify-self •

start

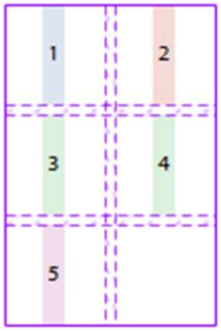


end

```
.item5 {
justify-self: start;
}
```



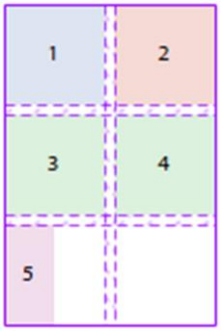
center



stretch

لن تعمل إلا إذا كان مقاس العناصر مضبوط إلى auto

```
.item5 {
 width: 25px;
```



• ضبط تموضع العناصر على المحورين معاً place-items , place-self •

place-items

خاصية مُختصرَة، تكتب في الحاوية وتؤثر على تموضع جميع عناصر الشبكة على المحورين معاً، وتأخذ الشكل

place-items : align-items / justify-items

خاصية مُختصرَة، تكتب في كل عنصر على حدة، تقوم بتجاوز override قيمة place-items ، وتؤثر على تموضع كل عنصر على المحورين معاً، وتأخذ الشكل

place-self : align-self / justify-self

عناصر الشبكة Grid Items

الأبْنَاء

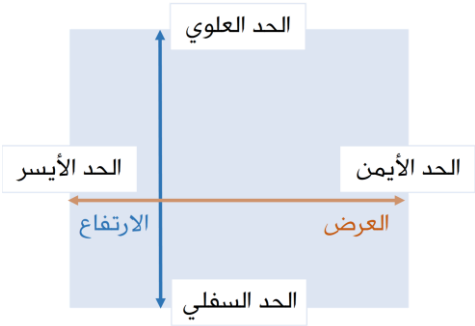
قبل أن نبدأ

في آخر مواصفة للشبكة CSS Grid Specification لا يمكن استهداف العناصر المكونة للشبكة مثل الخلية أو المسار أو المنطقة لتنسيقها لأنها غير موجودة في شجرة DOM أي أننا لا نستطيع أن نُنسق الخلية الثانية أو المسار العمودي الثالث مثلاً لتكون لهما خلفية حمراء، لكن ما نستطيع فعله هو أن نضع عنصر HTML ليشغل الخلية أو المسار أو المنطقة ثم نستطيع استهدافه للنُنسِقه، فيجب التفريق بين العناصر المُكونة للشبكة مثل(الخلايا والمسارات والمناطق)وعناصر HTML التي ستشغل تلك العناصر، والتي تسمى Grid Items.

إذاً Grid Items هي أي عنصر HTML يكون ابن مباشر لحاوية الشبكة، العنصر قد يشغل خلية أو مسار أو منطقة.

يجب الانتباه عند وجود أي محتوى ضمن حاوية الشبكة غير مُغلف بعنصر HTML ، ستعتبره الشبكة عنصر طبيعي وتعامله بخوارزمية التموقع التلقائي، لا يمكن التعديل على موقع وتنسيق هذا العنصر لأنه لا يمكن استهدافه لتطبيق الخصائص عليه لذلك تسمى عناصر الشبكة المجهولة Anonymous Grid Items ، تحدثنا عن ذلك في فصل مصطلحات الشبكة.

بالإضافة أننا نعلم أن لكل عنصر HTMLصندوق عرض، مستطيل أو مربع، وعرض العنصر width هو المسافة بين الحد الأيمن والحد الأيسر للعنصر، وارتفاع العنصر height هو المسافة بين الحد العلوي والحد السفلي للعنصر.



كما ستتأثر العناصر الأبناء عند تحويل العنصر الأب لشبكة grid مثل الخاصيات التي تؤثر

على صندوق عرض العنصر مثل float و inline و block و inline-block و table-cell و vertical-align و column لن يكون لها تأثير على عناصر الشبكة Grid Items ، كما أن ظاهرة انهيار الهامش margin-collapse لن تؤثر على عناصر الشبكة Grid Items.

سنتناول في هذا الفصل عناصر الشبكة Grid Items من حيث:

- ضبط الموقع.
- ضبط التمدد.
- الخاصيات المُختصرَة.
- الترتيب order.

ضبط الموقع

يتم ضبط موقع عناصر الشبكة Grid Items بثلاث طرق، بالتموقع المُعتمد على خطوط الشبكة، أو التموقع المُعتمد على المناطق، أو بخوارزمية التموقع التلقائي Auto-Placement Algorithm ، تحدثنا عن التموقع المُعتمد على المناطق في فصل حاوية الشبكة، وسنتحدث عن خوارزمية التموقع التلقائي في الفصول اللاحقة، أما الآن سنتحدث عن التموقع المعتمد على خطوط الشبكة.

التموقع المُعتمد على خطوط الشبكة يعني ضبط موقع كل عنصر من عناصر الشبكة بالإشارة إلى خطوط الشبكة التي يقع ضمنها هذا العنصر، مثلاً العنصر الأول يقع بين الخطين العموديين الأول والثالث والخطين الأفقيين الثاني والثالث، بينما يقع العنصر الثاني بين الخطين العموديين الثالث والرابع والخطين الأفقيين الرابع والخامس.

حتى نستطيع ضبط موقع كل عنصر بالإشارة إلى خطوط الشبكة يجب إسناد خطوط الشبكة التي يقع العنصر ضمنها لخاصيات التموقع في كل عنصر، وهي:

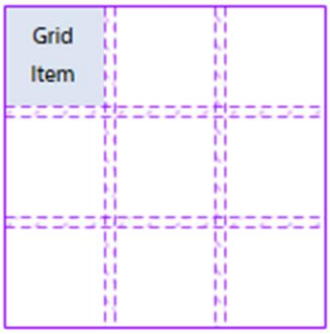
- grid-column-start : تحدد خط الشبكة العمودي الذي سيقع عليه الحد الأيسر للعنصر.
- grid-column-end : تحدد خط الشبكة العمودي الذي سيقع عليه الحد الأيمن للعنصر.

- grid-row-start : تحدد خط الشبكة الأفقي الذي سيقع عليه الحد العلوي للعنصر.
- grid-row-end : تحدد خط الشبكة الأفقي الذي سيقع عليه الحد السفلي للعنصر.

في ضوء ما سبق وبما أنه هناك عدة طرق مختلفة للإشارة لخطوط الشبكة، هذه الخاصيات يمكن أن تأخذ أنواع مختلفة من القيم:

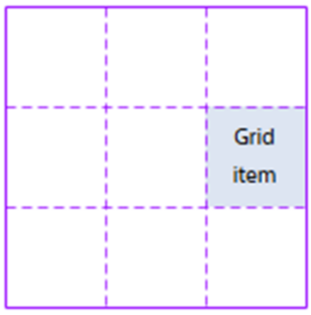
- Auto (default) : ترك تحديد الخط لخوارزمية التموقع التلقائي، وهي القيمة الافتراضية.
- Line Number : تحديد رقم خط الشبكة الذي سيقع عليه حد العنصر.
- Line Name : تحديد اسم خط الشبكة الذي سيقع عليه حد العنصر.
- LineName+Number : تُستعمل هذه الطريقة عندما تكون لائحة المسارات مُكونة عن طريق دالة التكرار، لذلك نضع اسم الخط وبجانبه ترتيب المسار.
- Span+Number : هذه القيمة لا تحدد خط الشبكة الذي سيقع عليه حد العنصر وإنما تحدد عدد المسارات التي سيتمدد عليها العنصر، سنتحدث عنها لاحقاً في قسم التمدد.

Auto (default)



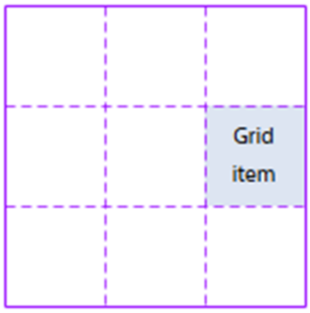
Line Number

```
.grid_item{
  grid-column-start: 3;
  grid-column-end: 4;
  grid-row-start: 2;
  grid-row-end: 3;
}
```



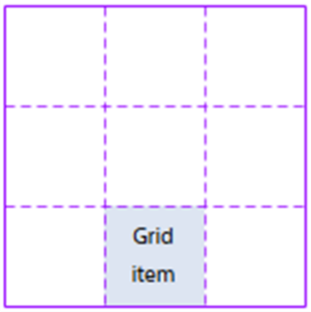
Line Name

```
.grid_container{
  grid-template-columns:[col1start]50px [col1end] 50px[col2end] 50px [col3end] ;
  grid-template-rows: [row1start]50px [row1end] 50px[row2end] 50px [row3end];
}
.grid_item {
  grid-column-start: col2end;
  grid-column-end: col3end;
  grid-row-start: row1end;
  grid-row-end: row2end;
}
```



LineName+Number

```
.grid_container{
  grid-template-columns:repeat(3,[colstart] 50px [colend]) ;
  grid-template-rows: repeat(3,[rowstart] 50px [rowend]) ;
}
.grid_item {
  grid-column-start: colstart 2;
  grid-column-end: colend 2;
  grid-row-start: rowstart 3;
  grid-row-end: rowend 3;
}
```



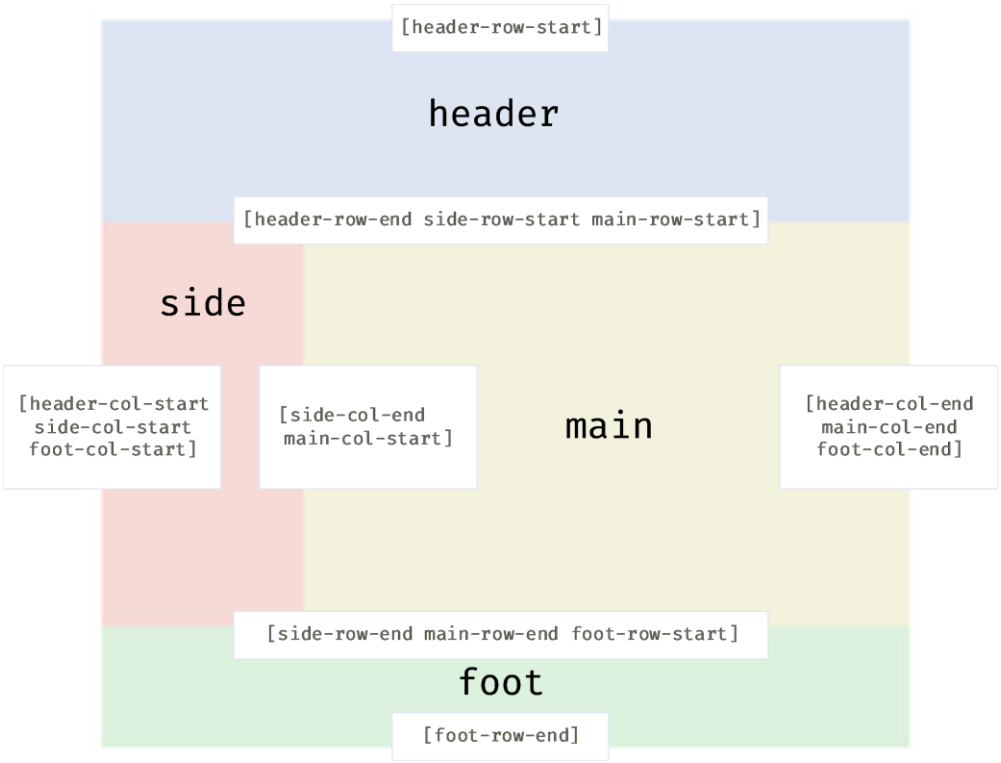
ملاحظات ضبط الموقع

الموقع المُعتمد على المناطق

وجب التنويه أن ضبط الموقع المُعتمد على المناطق يستعمل خطوط الشبكة أيضاً لضبط مواقع العناصر، حيث تقوم الشبكة آلياً بتسمية الخطوط المحيطة بكل عنصر اعتماداً على اسم المنطقة.

```
.grid_container{
 grid-template-columns:repeat(4,50px) ;
 grid-template-rows:repeat(3,50px) 30px ;
 gap:5px;
 grid-template-areas:
   "header header header header"
   "side   main   main   main  "
   "side   main   main   main  "
   "foot   foot   foot   foot  ";
}
```

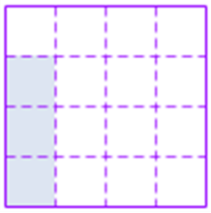
```
.item1 {
 grid-area: header;
}
.item2 {
 grid-area: side;
}
.item3 {
 grid-area: main;
}
.item4 {
 grid-area: foot;
}
```



التبديل التلقائي

في حالة وضع قيمة خط الشبكة اللاحق قبل خط الشبكة السابق ستقوم الشبكة آلياً بالتبديل بينهما، مثلا نريد العنصر أن يكون حده العلوي على خط الشبكة الثاني وحده السفلي على خط الشبكة الخامس، ولكن بالخطأ قمنا بالتبديل بينهما بحيث يقع حد العلوي على خط الشبكة الخامس وحده السفلي على خط الشبكة الثاني، في هذه الحالة ستقوم الشبكة آلياً بالتبديل بينهما لأنه لا يمكن أن يأتي الحد السفلي قبل الحد العلوي.

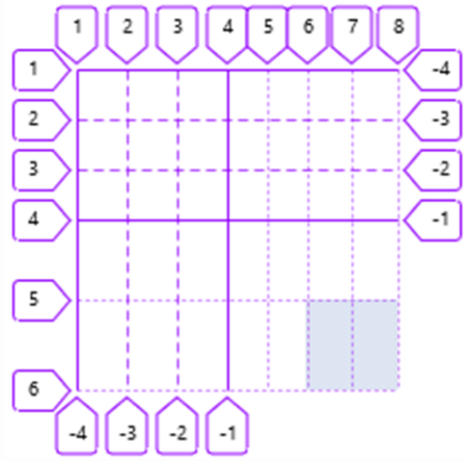
```
.grid_item{
  grid-row-start: 5;
  grid-row-end: 2;
}
```



موقع خارج الحدود

عند ضبط موقع العنصر لخطوط شبكة خارج حدود الشبكة الصريحة ستقوم الشبكة بتوليد مسارات ضمنية لاستيعاب العنصر.

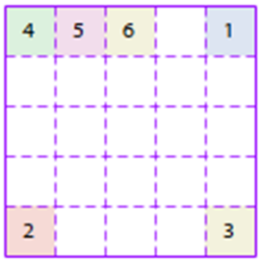
```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;
}

.item1 {
  grid-column-start: 6;
  grid-column-end: 8;
  grid-row-start: 5;
  grid-row-end: 6;
}
```



موقع في نهاية المسار

عند ضبط حد العنصر الأيمن أو السفلي إلى سالب واحد(1-) سيكون دائماُ في نهاية المسار، مهما كان هناك من مسارات أو مهما تم توليد مسارات.

```
.grid_container{
  grid-template-columns:repeat(5,25px) ;
  grid-template-rows:repeat(5,25px) ;
  grid-auto-flow: dense;
}

.item1 {
   grid-column-end: -1;
}

.item2 {
   grid-row-end: -1;
}
.item3 {
   grid-column-end: -1;
   grid-row-end: -1;
}
```



ضبط التمدد

التمدد هو أن يتوسع ويتمدد عنصر الشبكة Grid Item ليغطي أكثر من خلية، يمكن أن يتمدد أفقياً ليغطي أكثر من مسار عمودي، ويمكن أن يتمدد عمودياً ليغطي أكثر من مسار أفقي، ويمكن أن يتمدد أفقياً وعمودياً.

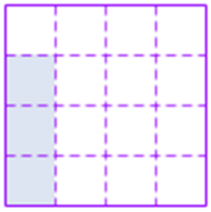
هناك نوعين للتمدد، التمدد الضمني، و التمدد بواسطة الكلمة span.

التمدد الضمني

يحدث عندما نضبط موقع حدود العنصر من الأعلى والأسفل معاً، أو من اليمين و اليسار معاً، أو من الجهات الأربعة.

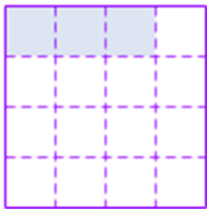
- التمدد الضمني من الأعلى والأسفل: كأن نقول أنّ حدود العنصر ستقع بين الخطين الأفقيين الثاني والخامس، سنجد أن العنصر سيتمدد تلقائياً بشكل عمودي ليغطي ثلاث مسارات أفقية.

```
.grid_item{
  grid-row-start: 2;
  grid-row-end: 5;
}
```



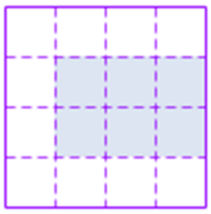
- التمدد الضمني من اليمين واليسار :كأن نقول أن حدود العنصر ستقع بين الخطين العموديين الأول والرابع ، سنجد أن العنصر سيتمدد ضمنياً بشكل أفقي ليغطي ثلاث مسارات عمودية.

```
.grid_item{
  grid-column-start: 1;
  grid-column-end: 4;
}
```



- التمدد الضمني من الجهات الأربع :كأن نقول أن حدود العنصر ستقع بين الخطين الأفقيين الثاني والرابع وبين الخطيين العموديين الثاني والخامس ، سنجد أن العنصر سيتمدد تلقائياً أفقياً وعمودياً ليغطي أكثر من مسار عمودي وأفقي ،ويجب أن ننوه إلى أن ضبط الموقع المُعتمد على المناطق يضبط حدود العنصر من الجهات الأربع.

```
.grid_item{
  grid-column-start: 2;
  grid-column-end: 5;
  grid-row-start: 2;
  grid-row-end: 4;
}
```

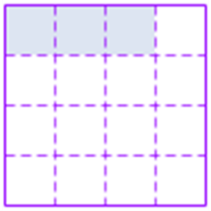


التمدد بواسطة الكلمة span

توضع هذه الكلمة كقيمة لخاصيات ضبط تموقع العناصر، وتحدد عدد المسارات التي سيتمددها العنصر، ولها قواعد .

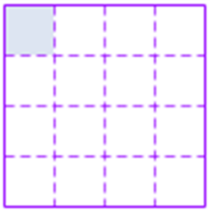
- تكتب الكلمة span و بجانبها رقم يدل على مقدار المسارات التي سيتمددها العنصر.

```
.grid_item{
  grid-column-start: span 3;
}
```



- إذا لم يكن هناك رقم ستعاملها الشبكة كرقم واحد، أي لن يتمدد العنصر.

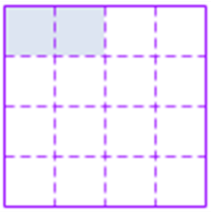
```
.grid_item{
  grid-column-start: span ;
}
```



- ضبط span لأحد خصائص التموقع سيمنعك من تحديد خط الشبكة الذي تحدده تلك الخاصية وسيأخذ خط الشبكة القيمة auto.

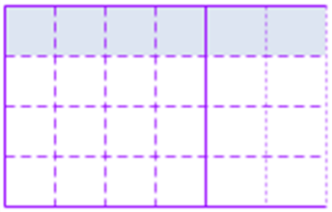
- لا تُجمَع مقادير التمدد في span لو ضبطت من جهتين، أي لو ضبطنا grid-column-start إلى span 2 وضبطنا grid-column-end إلى span 3 فلن يصبح المجموع span 5 و إنما تؤخذ قيمة الخاصية التي لاحقتها start .

```
.grid_item{
  grid-column-start: span 2;
  grid-column-end: span 3;
}
```



- اذا تمدد العنصر لخارج حدود الشبكة الصريحة ستقوم الشبكة بتوليد مسارات ضمنية .

```
.grid_item{
   grid-column-start: span 6;
}
```



الخاصيات المُختصَرَة

لدينا ثلاث خاصيات مُختصرَة تفيد في ضبط الموقع

- grid-column
- grid-row
- grid-area

grid-column

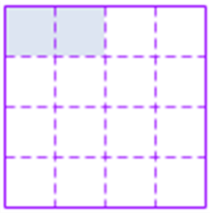
تحدد هذه الخاصية خط الشبكة الذي سيقع عليه الحد الأيسر للعنصر وخط الشبكة الذي سيقع عليه الحد الأيمن للعنصر في وقت واحد، أي أنها دمج للخاصيتين grid-column-start و grid-column-end معاً، وتأخذ الشكل:

```
grid-column : grid-column-start / grid-column-end
```

بما أنه هناك أكثر من طريقة للإشارة لخطوط الشبكة، فيمكن لهذه الخاصية أن تأخذ جميع القيم المحددة لخطوط الشبكة.

- Auto (default)
- Line Number
- Line Name
- LineName+Number
- Span+Number

```
.grid_item{
 grid-column: 1 / 3;
}
```



في حالة تجاهل قيمة grid-column-end ستضبط آلياً إلى القيمة auto.

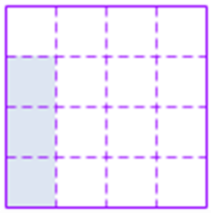
grid-row

تحدد هذه الخاصية خط الشبكة الذي سيقع عليه الحد العلوي للعنصر وخط الشبكة الذي سيقع عليه الحد السفلي للعنصر في وقت واحد، أي أنها دمج للخاصيتين grid-row-start و grid-row-end معاً، وتأخذ الشكل:

```
grid-row : grid-row-start / grid-row-end
```

ويمكن لهذه الخاصية أن تأخذ جميع القيم المحددة لخطوط الشبكة.

```
.grid_item{
 grid-row: 2 / 5;
}
```



في حالة تجاهل قيمة grid-row-end ستضبط آلياً إلى القيمة auto.

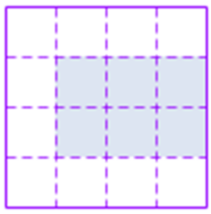
grid-area

لهذه الخاصية وظيفتين فإما تحدد اسم المنطقة التي سيتبع لها العنصر، أو تحدد هذه الخاصية جميع خطوط الشبكة التي ستقع عليها حدود العنصر في وقت واحد أي أنها دمج للخاصيتين المُختصرَتين grid-column و grid-row معاً، وتأخذ الشكل :

```
grid-area:grid-row-start/grid-column-start/grid-row-end/grid-column-end
```

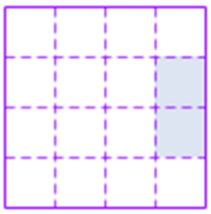
ويمكن لهذه الخاصية أن تأخذ جميع القيم المحددة لخطوط الشبكة.

```
.grid_item{
 grid-area: 2 / 2 / 4 / 5;
}
```



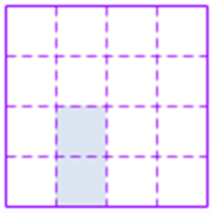
أي قيمة لا تريد ضبطها اضبطها للقيمة auto ، مثلاً لا نريد ضبط grid-column-start .

```
.grid_item{
 grid-area: 2 / auto / 4 / 5;
}
```



لو تم تجاهل آخر قيمة ستضبط آلياً إلى auto.

```
.grid_item{
 grid-area: 2 / 3 / 5;
}
```



لو تم تجاهل آخر قيمتين ستُضبطان آلياً إلى auto.

لو تم تجاهل آخر ثلاث قيم ستضبط آلياً إلى auto.

الترتيب order

تضبط هذه الخاصية ترتيب ظهور العناصر على الشبكة، تأخذ هذه الخاصية قيم عددية، القيمة الافتراضية 0 ، ستُظهر الشبكة العناصر التي لها القيمة الأقل أولاً ثم تنتقل للقيمة الأكثر وهكذا حتى نهاية العناصر.

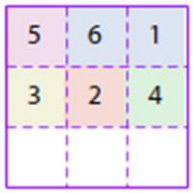
```
.grid_item2{
 order: 1;
}

.grid_item9{
 order: -1;
}
```

هنا جميع العناصر لها القيمة الابتدائية 0 ماعدا العنصر الثاني والتاسع، فالعنصر التاسع له القيمة 1- وهي أقل قيمة للخاصية order لذلك يظهر أولاً، بينما العنصر الثاني له القيمة 1 وهي أعلى قيمة لذلك يظهر أخيراً.

ولكن يتراود سؤال منطقي، هناك 7 عناصر متبقية لها القيمة 0 فكيف تم ترتيبها؟ يتم ترتيب العناصر المتساوية في قيمة الخاصية order بحسب ترتيبها في شجرة DOM ، أي بحسب وجودها في كود HTML .

```
/////HTML
<div class="grid_container">
   <div class="item1">1</div>
   <div class="item2">2</div>
   <div class="item3">3</div>
   <div class="item4">4</div>
    <div class="item5">5</div>
   <div class="item6">6</div>
  </div>
```

```
.item2{
 order: 1;
}
.item4{
 order: 1;
}
.item5{
 order: -1
}
.item6{
 order: -1
}
```



هنا لدينا عنصرين لهما القيمة 1- وهي أقل قيمة، لذلك سيظهران أولاً لكن العنصر

الخامس يسبق العنصر السادس في ترتيبه في HTML لذلك سيظهر العنصر الخامس يليه العنصر السادس، ننتقل للعنصرين الأول والثالث ولهما القيمة الافتراضية 0، لكن الأول

يسبق الثالث في HTML لذلك يظهر الأول يليه الثالث، ننتقل للعنصرين الثاني والرابع

ولهما القيمة 1 وهي القيمة الأعلى وسيظهران أخيراً لكن الثاني يسبق الرابع لذلك يظهر الثاني ويليه أخيراً الرابع.

يجب الانتباه أن هذه الطريقة تُرتب العناصر ظاهرياً فقط، مما يضر بالوصولية Accessibility .

خَوارِزْمِيَةُ التَمُوقعْ التِلْقَائِي

Auto-Placement Algorithm

مشاعر من الفرح والسعادة تغمرك وأنت جالس على جهاز الكومبيوتر تعمل على تخطيط Grid الأداة القوية الجديدة التي تجعلك تشعر بأنك عملاق تصميم المواقع الإلكترونية ولكن فجأة قمت بتغيير موقع أحد العناصر فقط ثم تحول تصميمك الأنيق إلى كتلة فوضوية ومبعثرة من التقسيمات والفواصل تجعل أصعب مراحل لعبة تيتريس Tetris تبدو سهلة مقارنةً بتصميمك.

بعض العناصر تنزلق في مسارات غير مساراتها!

بعض العناصر في مواقع غير متوقعة!

هناك خلايا فارغة لا يريد أي من العناصر أن يشغلها!

يتم توليد مسارات جديدة لا تريدها!

أن تكون ماهراً في تخطيط Grid لا يقتصر على حفظ بنية الخاصيات وكيفية تطبيقها وحسب، بل من الضروري أيضاً أن تفهم كيف المستعرض سيموقع العناصر ضمن الشبكة، لقد تحدثنا بالتفصيل كيف يمكنك ضبط مواقع العناصر يدوياً على الشبكة باستعمال التموقع المعتمد على المناطق أو المعتمد على خطوط الشبكة ولكن بقيت قطعة واحدة من السحر، خوارزمية تحسب أولوية تموقع العناصر وتضبط تدفق كل شيء إلى الشبكة، إنها خوارزمية التموقع التلقائي Auto-Placement Algorithm ، التي سنتعرف عليها في هذا الفصل.

خوارزمية التموقع التلقائي

هي خوارزمية مسؤولة عن ترتيب ظهور العناصر على الشبكة بالشكل النهائي، هذه وظيفتها بشكل عام.

بشكل أدق هي عملية ترتيب أولوية أخذ العناصر ثم وضعها ضمن الشبكة، أي أخذ العناصر ثم وضعها بالشبكة، أخذ ثم وضع.

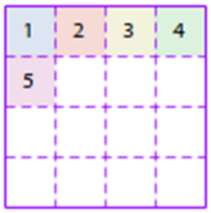
يمكن أن نلخص هذه العملية بثلاث مراحل:

1-عملية أخذ العنصر.

2- عملية وضع العنصر.

3-الانتقال للعنصر التالي وتكرار ما سبق حتى نهاية العناصر.

```
/////HTML

<div class="grid_container">
   <div>1</div>
   <div>2</div>
   <div>3</div>
   <div>4</div>
   <div>5</div>
  </div>

/////CSS

.grid_container{
  display: grid;
  grid-template-columns:repeat(4,25px) ;
  grid-template-rows:repeat(4,25px) ;
}
```



الخوارزمية أخذت العنصر الأول من HTML ثم تعاملت معه بوضعه بأول مكان مناسب، ثم انتقلت للعنصر التالي أخذته من HTML ثم تعاملت معه بوضعه في المكان المناسب، تُكرر هذه الخطوات حتى آخر عنصر.

ولكن بالنسبة لعملية الأخذ كيف تقرر الخوارزمية من هو أول عنصر ستأخذه من HTML؟ وكيف تقرر من العنصر التالي؟ إذن لابد من عملية تصفية(فلترة) للعناصر لنعرف ترتيب أخذها.

وبالنسبة لعملية الوضع هل سلوكها يختلف من عنصر لآخر ؟

ولكن قبل أن ننتقل إلى ذلك هل تعلم أن الخوارزمية هي خوارزمية عنصرية بعض الشيء.

خوارزمية التموقع التلقائي خوارزمية عنصرية!

تُقَسِّم الخوارزمية العناصر إلى ثلاث طبقات من حيث الأولوية، عناصر الموقع الصريح و العناصر المقفولة مع الاتجاه وأخيراً العناصر المتبقية، وبالتالي تُصبح لدينا مراحل الخوارزمية مُقسمة بحسب الطبقات.

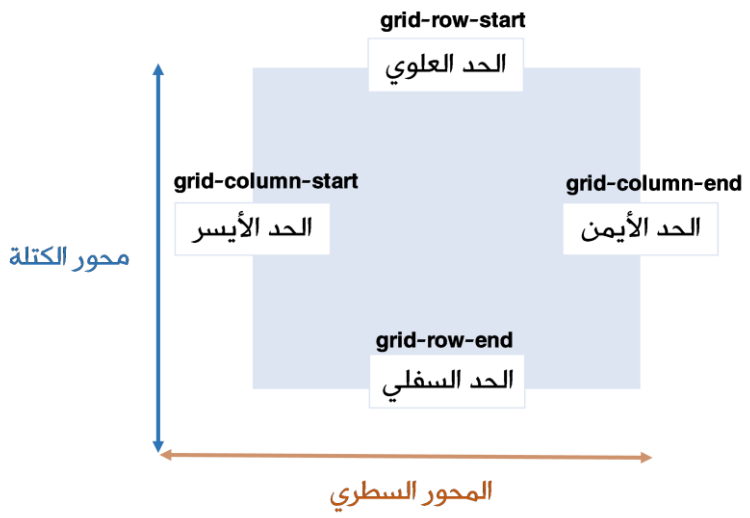
مرحلة عناصر الموقع الصريح	1. أخذ عناصر الموقع الصريح. 2. وضع عناصر الموقع الصريح في الشبكة. 3. تكرار ما سبق حتى نهاية عناصر الموقع الصريح. 4. الانتقال لمرحلة العناصر المقفولة مع الاتجاه.
مرحلة العناصر المقفولة مع الاتجاه	5. أخذ عناصر العناصر المقفولة مع الاتجاه. 6. وضع العناصر المقفولة مع الاتجاه في الشبكة. 7. تكرار ما سبق حتى نهاية العناصر المقفولة مع الاتجاه. 8. الانتقال لمرحلة العناصر المتبقية.
مرحلة العناصر المتبقية	9. أخذ العناصر المتبقية. 10. وضع العناصر المتبقية في الشبكة. 11. تكرار ما سبق حتى نهاية العناصر المتبقية.

ولكن كيف سنميز عناصر الموقع الصريح من عناصر الأولوية القصوى من العناصر المتبقية ؟

عناصر الموقع الصريح

هي أعلى العناصر من حيث الأولوية ستأخذها الخوارزمية أولاً وتضعها بالشبكة، و قبل الخوض في معرفة عناصر الموقع الصريح لنسترجع بعض المعلومات الأساسية:

- **نعلم أنه لدينا محورين للتعامل معهم، محور الكتلة والمحور السطري.**
- **نعلم أن لكل عنصر أربع حدود، حد علوي وسفلي وأيسر وأيمن.**
- **ونعلم أنه يمكننا تحديد موقع العنصر بدقة على الشبكة بتحديد خطوط الشبكة التي يقع عليها كل حد من حدود العنصر عن طريق الخاصيات grid-column-start و grid-column-end وgrid-row-start و grid-row-end أو بالخاصيات المختصرة grid-column و grid-row و grid-area.**

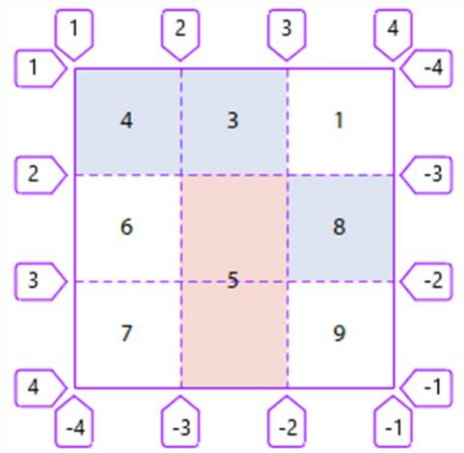


بناءاً على ما سبق فعناصر الموقع الصريح هي أي عنصر تم تحديد أحد حديه الموافقين للمحور السطري على الأقل مع أحد حديه الموافقين لمحور الكتلة على الأقل وبالعكس، بمعنى آخر هي أي عنصر تم تحديد:

- **حده العلوي مع الحد الأيمن أو الأيسر.**
- **أو حده السفلي مع الحد الأيمن أو الأيسر.**
- **أو حده الأيسر مع الحد العلوي أو السفلي.**
- **أو حده الأيمن مع الحد العلوي أو السفلي.**

- ثلاثة حدود من حدوده.
- كل حدوده.

```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;

}
.item2 {
  grid-row-start: 2;
  grid-column-start: 2;
}
.item3 {
  grid-row-end: 2;
  grid-column-start: 2;
}
.item4 {
  grid-row-end: 2;
  grid-column-end: 2;
}
.item5 {
  grid-area: 2 / 2 / 4 / 3;
}
.item7 {
  grid-row-start: 3;
}
.item8 {
  grid-area: 2 /3;
}
```



العناصر 2 و 3 و 4 و 5 و 8 هي عناصر موقع صريح لأن :

العنصر 2 تم تحديد موقع حده العلوي مع موقع حده الأيسر.

العنصر 3 تم تحديد موقع حده السفلي مع موقع حده الأيسر.

العنصر 4 تم تحديد موقع حده السفلي مع حده الأيمن.

العنصر 5 تم تحديد مواقع حدوده الأربعة.

العنصر 8 تم تحديد موقع حده العلوي مع موقع حده الأيسر.

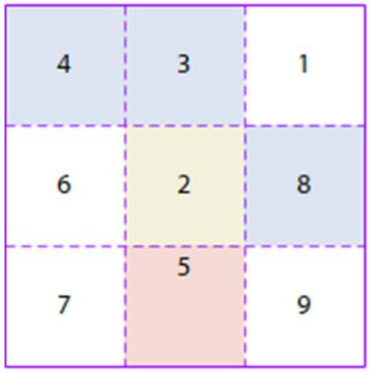
بينما العنصر 7 ليس عنصر موقع صريح لأنه تم تحديد حده العلوي فقط.

لحظة !

أين العنصر 2 ؟

العنصر 2 والعنصر 5 يقعان في نفس المنطقة لذلك تداخلا، العنصر 2 حالياً تحت العنصر 5 لتحديد أولوية الظهور نحدد قيمة z-index .

```
.item2 {
    z-index: 1;
}
```



العنصر 5 حالياً تحت العنصر 2 ولكن بما أنه يمتد على أكثر من مسار فإننا نرى جزء منه.

> **عناصر الموقع الصريح يمكن أن تتراكب فوق بعضها البعض اذا تم ضبطها لنفس الموقع، لتحديد أولوية الظهور نستعمل الخاصية z-index**

بتنا نعرف الآن عناصر الموقع الصريح و كيف تكون أولوية أخذهم وكيف تضعهم الخوارزمية على الشبكة، وبعد الانتهاء من مرحلة عناصر الموقع الصريح ستنتقل الخوارزمية لمرحلة العناصر المقفولة مع الاتجاه.

العناصر المقفولة مع الاتجاه

ما هو الاتجاه ؟ وما هو القفل ؟

اتجاه التدفق

هو القيمة التي ضُبطت للخاصية grid-auto-flow التي تحدد اتجاه توليد المسارات الجديدة وطريقة الملء ، فهي إما أن تكون row وهي الافتراضية أو column ، وعليه فإن اتجاه التدفق إما أن يكون row أو column .

تنويه لقد قلنا سابقاً أن خاصية grid-auto-flow تأخذ نوع آخر من القيم، إما sparse وهي الافتراضية أو dense ، حسناً نعم هذه النوع من القيم مختص بطريقة التعبئة، سنتحدث عنها لاحقاً في هذا الفصل فهي تقرر سلوك الخوارزمية في عملية وضع العناصر المتبقية على الشبكة لأنها لا تؤثر على عناصر الموقع الصريح ولا العناصر المقفولة مع الاتجاه.

الآن نعلم أن اتجاه التدفق إما أن يكون row أو column ، ولكن ما هو القفل ؟

القفل

بدايةً لدينا نوعين من القفل، قفل مع اتجاه التدفق وقفل عكس الاتجاه.

و قبل الخوض في القفل لنسترجع بعض المعلومات الأساسية:

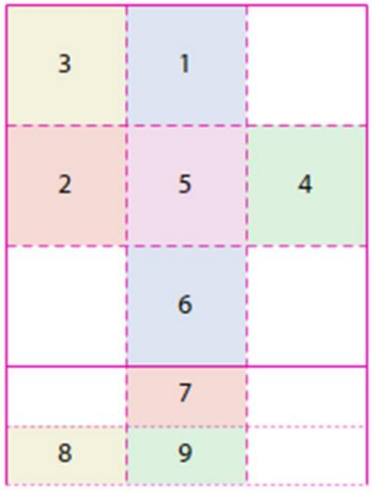
- **نعلم أن لكل عنصر أربع حدود، حد علوي وسفلي وأيسر وأيمن.**
- **ونعلم أنه يمكننا تحديد موقع العنصر بدقة على الشبكة بتحديد خطوط الشبكة التي يقع عليها كل حد من حدود العنصر عن طريق الخاصيات grid-column-start و grid-column-end وgrid-row-start و grid-row-end أو بالخاصيات المختصرة.**
- **ونعلم أن اتجاه التدفق إما أن يكون row أو column .**

بناءاً على ما سبق فإن العناصر المقفولة مع الاتجاه هي أي عنصر تم ضبط أحد حدوده على الأقل على خط الشبكة الموافق لاتجاه التدفق.

أي إذا كان اتجاه التدفق row فإي عنصر تم ضبط موقع حده العلوي أو حده السفلي أو كلاهما هو عنصر مقفول مع اتجاه التدفق row ، و أي عنصر تم ضبط موقع حده الأيسر أو حده الأيمن أو كلاهما فهو عنصر مقفول عكس اتجاه التدفق row ، للتنويه أي عنصر تم ضبط ثلاثة أو كل حدوده فهو عنصر من عناصر الموقع الصريح.

grid-auto-flow : row

```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;
  grid-auto-flow: row;
}
.item2 {
  grid-row-start: 2;
}
.item3 {
  grid-row-end: 2;
}
.item4 {
  grid-row: 2 / 3;
}
.item5 {
  grid-area: 2 / 2 / 3 / 3;
}
.item6 {
  grid-column-start: 2;
}
.item7 {
  grid-column-end: 3;
}
.item8 {
  grid-column:1 / 2 ;
}
```



العناصر 2 و 3 و 4 عناصر مقفولة مع اتجاه التدفق، لأن اتجاه التدفق هو row و تم ضبط أحد الحدين العلوي أو السفلي أو كلاهما .

و العنصر 5 هو الأعلى أولوية هنا لأنه تم ضبط كل حدوده(عناصر الموقع الصريح).

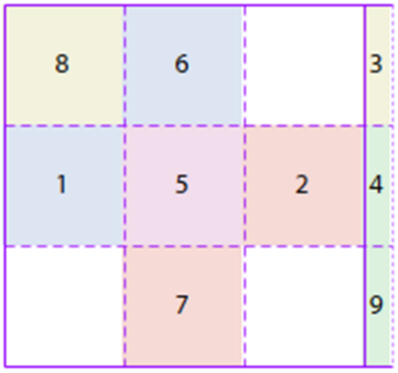
العناصر 6 و 7 و 8 عناصر مقفولة عكس اتجاه التدفق لأن اتجاه التدفق هو row و تم ضبط أحد الحدين الأيسر أو الأيمن أو كلاهما.

قد يبدو ترتيب العناصر بالصورة مشتتاً سنتحدث عن ذلك لاحقاً في هذا الفصل، أما الآن يكفي أن نميز العناصر المقفولة مع الاتجاه من العناصر المقفولة عكس الاتجاه .

و إذا كان اتجاه التدفق column فإي عنصر تم ضبط موقع حده الأيمن أو حده الأيسر أو كلاهما هو عنصر مقفول مع اتجاه التدفق column ، و أي عنصر تم ضبط موقع حده العلوي أو حده السفلي أو كلاهما فهو عنصر مقفول عكس اتجاه التدفق column ،

grid-auto-flow : column

```
.grid_container{
  grid-template-columns:repeat(3,25px) ;
  grid-template-rows:repeat(3,25px) ;
  grid-auto-flow: column;
}
.item2 {
  grid-row-start: 2;
}
.item3 {
  grid-row-end: 2;
}
.item4 {
  grid-row: 2 / 3;
}
.item5 {
  grid-area: 2 / 2 / 3 / 3;
}
.item6 {
  grid-column-start: 2;
}
.item7 {
  grid-column-end: 3;
}
.item8 {
  grid-column:1 / 2 ;
}
```



العناصر 6 و 7 و 8 عناصر مقفولة مع اتجاه التدفق، لأن اتجاه التدفق هو column و تم ضبط أحد الحدين الأيسر أو الأيمن أو كلاهما .

العنصر 5 هو الأعلى أولوية هنا لأنه تم ضبط كل حدوده(عناصر الموقع الصريح).

العناصر 2 و 3 و 4 عناصر مقفولة عكس اتجاه التدفق، لأن اتجاه التدفق هو column و تم ضبط أحد الحدين العلوي أو السفلي أو كلاهما.

يجب التنويه أن الكلمة span لا تُحسب عند تحديد الأولوية، أي عند ضبط حد العنصر للقيمة span لن يُحسب من العناصر المقفولة مع الاتجاه ولا عكسه، لأن span تفيد بالتمدد ولا تفيد بتحديد الموقع

إذاً الخوارزمية ستعطي الأولوية هنا للعناصر المقفول موقعها مع اتجاه التدفق ستقوم بأخذها و وضعها ضمن الشبكة، والعناصر المقفولة بعكس اتجاه التدفق سيتم اعتبارها من عناصر المرحلة التالية(العناصر المتبقية).

ملاحظات هامة حول العناصر المقفولة مع الاتجاه

الملاحظة الأولى :قد يتبادر إلى ذهنك سؤال : كيف ستختار الخوارزمية ترتيب العناصر المقفولة مع الاتجاه، بمعنى آخر : من العناصر المقفولة مع الاتجاه أي عنصر سيكون الأول ثم الثاني ثم الثالث إلى نهاية العناصر ؟

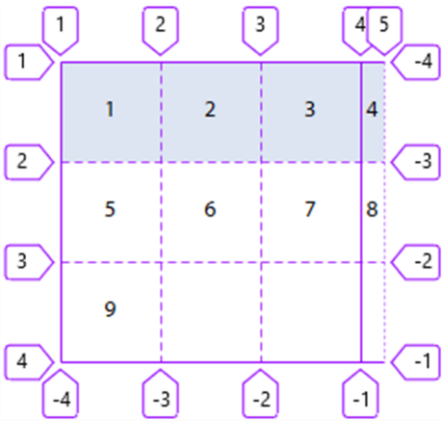
ستقوم الخوارزمية بترتيب العناصر بناءاً على عاملين:

1. order .

2. ترتيب وجود العنصر في DOM.

الملاحظة الثانية: أن العناصر المقفولة مع الاتجاه لا تتراكب فوق بعضها البعض كما تفعل عناصر الموقع الصريح.

```
.grid_container{
  grid-template-columns:repeat(3,50px) ;
  grid-template-rows:repeat(3,50px) ;
  grid-auto-flow: row;
}
.item1 {
  grid-row: 1 / 2;
}
.item2 {
  grid-row: 1 / 2;
}
.item3 {
  grid-row: 1 / 2;
 }
.item4 {
  grid-row: 1 / 2;
}
```



العناصر 1 و 2 و 3 و 4 مقفولة مع الاتجاه وتم ضبط موقع الحد العلوي والسفلي لجميع العناصر نفسه، لذلك نلاحظ أن الشبكة كونت مسار عمودي جديد لاستيعاب العنصر4 على الرغم من جهة توليد المسارات الجديدة هي row.

الآن خوارزمية التموقع التلقائي، ستقوم بأخذ عناصر الموقع الصريح وتضعها في أماكنها على الشبكة، وبعد الانتهاء من موقعة جميع عناصر الموقع الصريح تنتقل إلى موقعة العناصر المقفولة مع الاتجاه، وبعد الانتهاء ستنتقل إلى المرحلة الأخيرة مرحلة العناصر المتبقية.

العناصر المتبقية

تتألف من العناصر التي لم يتم تحديد أي حد من حدودها والعناصر المقفولة عكس الاتجاه، تتأثر هذه العناصر بالقيمتين sparse و dense اللتان تحددان طريقة التعبئة.

مرحلة العناصر المتبقية

1. أخذ العناصر المتبقية.
2. وضع العناصر المتبقية في الشبكة.
3. تكرار ما سبق حتى نهاية العناصر المتبقية.

أخذ العناصر المتبقية

ستقوم الخوارزمية بترتيب أخذ العناصر بناءاً على عاملين:

1. order .

2. ترتيب وجود العنصر في DOM.

وضع العناصر المتبقية في الشبكة

ستسلك الخوارزمية في هذه المرحلة سلوكين مختلفين بناءاً على طريقتي التعبئة sparse أو dense وعلى عدة عوامل مؤثرة أخرى.

العوامل المؤثرة في السلوكين:

1. هل العنصر المتبقي مقفول مع الاتجاه.
2. نوع التمدد : هل هو ناتج عن span أو تمدد ضمني (تحدث فقط في حالة كان العنصر مقفول عكس الاتجاه).

سلوك sparse

تستخدم الخوارزمية القيمة sparse التي تحافظ على ترتيب عناصر الشبكة حتى لو أدى ذلك إلى ترك فجوات بين العناصر لذلك ستبدو العناصر متناثرة، ولن تقوم هذه الخاصية بملء الفراغات بين عناصر grid، سلوكها يمكن تلخيصه بالشعار «ترتيب العناصر هو الأهم».

تتأثر بالعاملين:

- هل العنصر مقفول عكس الاتجاه.
- تمدد العنصر : من حيث نوع التمدد هل هو ناتج عن span أم تمدد ضمني (فقط في حالة كان مقفول عكس الاتجاه).

العنصر مقفول عكس الاتجاه

نعلم أن العنصر سيكون مقفول عكس الاتجاه إذا تم ضبط أحد حدوده المخالفة لاتجاه التدفق، أي إذا كان اتجاه التدفق row سيكون العنصر مقفول عكس الاتجاه إذا تم ضبط أحد حديه الأيمن أو الأيسر أو كلاهما، أما إذا كان اتجاه التدفق column سيكون العنصر مقفول عكس الاتجاه إذا تم ضبط أحد حديه العلوي أو السفلي أو كلاهما.

ستحترم sparse الحدود المضبوطة للعنصر المقفول بعكس الاتجاه أي أنها ستضعه على خطوط الشبكة المضبوط إليها عندما يأتي دوره حتى لو أدى ذلك لظهور خلايا شبكة فارغة ولن تقوم sparse بملء تلك الفراغات بعناصر تأتي بعد العنصر المقفول عكس الاتجاه لأن شعارها «ترتيب العناصر هو الأهم».

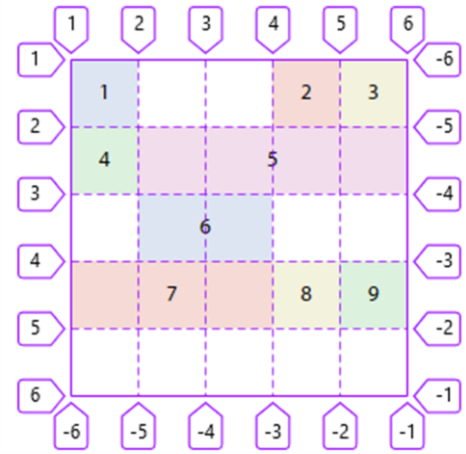
تمدد العنصر

عندما يأتي دور العنصر ليتم وضعه على الشبكة فقد يكون مقاسه (تمدده)أكبر من المكان الذي سيوضع فيه، هنا sparse تراعي حالتين إذا كان التمدد ناتج عن الكلمة span سينزلق العنصر لأقرب مكان يتسع فيه حتى لو عنى ذلك ظهور خلايا شبكة فارغة خلفه، أما اذا كان تمدده ضمني(تحدث فقط في حالة كان مقفول عكس الاتجاه) فستحترم sparse الحدود المضبوطة للعنصر المقفول بعكس الاتجاه وستقوم بتوليد مسارات جديده لكي يتسع فيها العنصر، وفي الحالتين لن تقوم sparse بملء الفراغات لأن شعارها «ترتيب العناصر هو الأهم».

طريقة التعبئة sparse

اتجاه التدفق والملء : row grid-auto-flow

```
.grid_container{
  grid-template-columns:repeat(5,35px) ;
  grid-template-rows:repeat(5,35px) ;
  grid-auto-flow: row;
}
.item2 {
  grid-column: 4 / 5;
}
.item5 {
  grid-column-start: span 4;
}
.item6 {
  grid-column:2 / 4 ;
}
.item7 {
  grid-column-start: span 3;
}
```



جميع هذه العناصر هي عناصر متبقية لكن العناصر 2 و 6 هي عناصر متبقية مقفولة عكس الاتجاه لذلك ستحترم sparse الحدود المضبوطة للعنصر المقفول بعكس الاتجاه حتى لو عنى ذلك ظهور خلايا فارغة.

لنفسر ما يحدث عنصر بعنصر :

بداية ستأخذ الخوارزمية العناصر بالترتيب لأنها كلها عناصر متبقية، كما أنه لا يوجد أي عنصر تم ضبط order له .

العنصر 1 : سيوضع في أول خلية متاحة.

العنصر 2 : عنصر مقفول عكس الاتجاه ستحترم sparse حدوده المضبوطة مما أدى لظهور خليتين فارغتين.

العنصر 3 : سيوضع في أول خلية متاحة بعد العنصر 2.

العنصر 4 : سيوضع في أول خلية متاحة بعد العنصر 3.

العنصر 5 : سيوضع في أول خلية متاحة بعد العنصر 4 لكنه يتمدد على 4 مسارات وهذا التمدد يتسع في المكان الذي أسند له.

العنصر 6 : عنصر مقفول عكس الاتجاه ستحترم sparse حدوده المضبوطة مما أدى لظهور خلية فارغة قبله.

العنصر 7 : سيوضع في أول خلية متاحة بعد العنصر 6 لكنه يتمدد على 3 مسارات والمكان يتسع لمسارين فقط، لذلك انزلق لأسفل تاركاً خلفه خلايا فارغة.
العنصر 8 و 9 سيوضعان في أول خلايا متاحة على الترتيب.

يجب التنويه أن العناصر المقفولة عكس الاتجاه مثل 2 و 6 حتى لو تمددت لأكثر من استيعاب الشبكة لن تنزلق لأسفل بل سيتم توليد مسارات جديدة لاحتوائها.

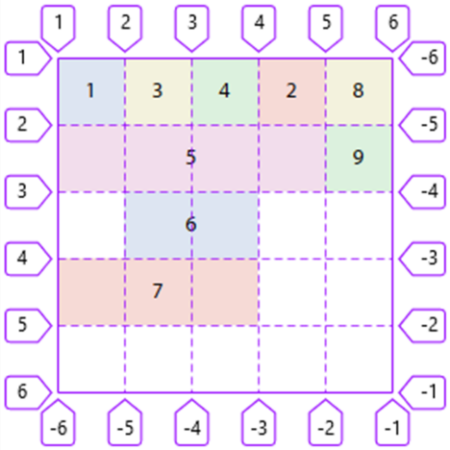
سلوك dense

سلوكها نفس سلوك sparse إلا أنه يجب ملء الفراغات المُتشكلة حتى لو كانت من عناصر تأتي لاحقاً، سلوكها يمكن تلخيصه بالشعار «يجب ملء الفراغات» ، ويجب التنويه إلى أنه من الممكن أن تتشكل فراغات في حال لم يوجد عناصر تشغل الفراغات أو كانت العناصر المتبقية تتمدد لأكثر مما يتسع له الفراغ.

طريقة التعبئة dense

grid-auto-flow : row اتجاه التدفق والملء

```
.grid_container{
  grid-template-columns:repeat(5,35px) ;
  grid-template-rows:repeat(5,35px) ;
  grid-auto-flow: dense;
}
.item2 {
  grid-column: 4 / 5;
}
.item5 {
  grid-column-start: span 4;
}
.item6 {
  grid-column:2 / 4 ;
}
.item7 {
  grid-column-start: span 3;
}
```



هنا جميع هذه العناصر هي عناصر متبقية لكن العناصر 2 و 6 هي عناصر متبقية مقفولة عكس الاتجاه لذلك ستحترم dense الحدود المضبوطة للعنصر المقفول بعكس الاتجاه حتى لو عنى ذلك ظهور خلايا فارغة، لكنها ستعمل على تعبئة هذه الفراغات من العناصر التي تأتي لاحقاً.

لنفسر ما يحدث عنصر بعنصر :

بداية ستأخذ الخوارزمية العناصر بالترتيب لأنها كلها عناصر متبقية، كما أنه لا يوجد أي عنصر تم ضبط order له .

العنصر 1 : سيوضع في أول خلية متاحة.

العنصر 2 : عنصر مقفول عكس الاتجاه ستحترم dense حدوده المضبوطة مما أدى لظهور خليتين فارغتين قبله.

العنصر 3 : سيوضع في أول خلية متاحة لذلك سيعود لملء الفراغ طالما يتسع فيه.

العنصر 4 : سيوضع في أول خلية متاحة لذلك سيعود لملء الفراغ بجانب العنصر 3 طالما يتسع فيه.

العنصر 5 : سيوضع في أول خلية متاحة أي سيوضع على يمين العنصر 2 لكنه يتمدد على 4 مسارات وهذا التمدد لا يتسع في المكان الذي أسند له لذلك سينزلق للأسفل مخلفاً خلية فارغة على يمين العنصر 2.

العنصر 6 : عنصر مقفول عكس الاتجاه ستحترم sparse حدوده المضبوطة ولن تضعه ليملأ الخلايا الفارغة السابقة، مما يؤدي لظهور خليتين فارغتين قبله، خلية في المسار السابق وخلية في المسار الحالي، وبالطبع لدينا خلية فارغة على يمين العنصر 2.

العنصر 7 : سيوضع في أول خلية متاحة لكنه يتمدد على 3 فلا يستطيع ملء الفراغات السابقة ولا يتسع في المكان بعد العنصر 6 لذلك انزلق لأسفل تاركاً خلفه خلايا فارغة.

العنصر 8 : سيوضع في أول خلية متاحة لذلك سيعود لملء أول فراغ متاح طالما يتسع فيه وهو الفراغ بجانب العنصر 2.

العنصر 9 : سيوضع في أول خلية متاحة لذلك سيعود لملء أول فراغ متاح طالما يتسع فيه وهو الفراغ بجانب العنصر 5.

كلـمة أخـيرة

المجتمع من حولنا اليوم لم يعد مجتمع معرفي كما كان سابقاً، بل أصبح مجتمع قائم على المهارات، فالمعرفة موجودة في كل مكان حولنا، في السابق أي درجة علمية من الجامعة كانت تكفي لكي تحصل على وظيفة مرموقة، أما الآن فلا أحد يهتم بما تعرف ومن أين حصلت على شهادتك، بل بما تستطيع عمله أي ما تملكه من مهارات، فامتلاك المعرفة ليس كافياً بل معرفة كيفية استخدامها، لذلك كان هذا الكتاب، آملاً أن يكون مساهمة متواضعة في إثراء المحتوى العربي وعِلْمَاً يُنتَفَعُ بِهِ.

انتهى في:

19-2-2020

" إذَا كَانَ الخَطُ حَسَنُ الوصْف، مَليحُ الرّصْف، مُفَتّح العيون، أملس المتون، كَثيرُ الائِتلاف، قَليلُ الاختِلاف، هشّتْ إِليّه النُفُوس وَاشْتَهَته الأَرْوَاح "

أحمد بن علي-القلقشندي

لطالما كان الخط من الصنائع الحضارية، وخزان العلوم الإنسانية، الخط هندسة صعبة وصناعة شاقة، فالخط لسان اليد، ينطق عن الساكت، ويترجم عن القلوب، في هذا السياق الشكر موصول للمصمم "محمد الحسن متالي " وهو مصمم موريتاني له باع طويل في مجال التصميم الرقمي والتيبوغرافيا والإنتاج الفني، ومبدع سلسلة خطوط Hacen التي اُستعمِل منها خط Hacen Liner في هذا الكتاب .

Hacen

www.hacen.net