Nội dung

[**Tạo lưới CSS đầu tiên của bạn** 2](#_Toc81148892)

[**Thêm cột bằng lưới-mẫu-cột** 3](#_Toc81148893)

[**Thêm hàng với lưới-mẫu-hàng** 4](#_Toc81148894)

[**Sử dụng đơn vị Lưới CSS để thay đổi kích thước của cột và hàng** 5](#_Toc81148895)

[**Tạo khoảng cách cột bằng cách sử dụng lưới-cột-khoảng cách** 6](#_Toc81148896)

[**Tạo Khoảng cách hàng bằng cách sử dụng lưới-hàng-khoảng cách** 7](#_Toc81148897)

[**Thêm khoảng trống nhanh hơn với khoảng cách lưới** 8](#_Toc81148898)

[**Sử dụng lưới-cột để kiểm soát khoảng cách** 9](#_Toc81148899)

[**Sử dụng hàng lưới để kiểm soát khoảng cách** 10](#_Toc81148900)

[**Căn chỉnh một mục theo chiều ngang bằng cách sử dụng justify-self** 11](#_Toc81148901)

[**Căn chỉnh một mục theo chiều dọc bằng cách sử dụng căn chỉnh tự** 12](#_Toc81148902)

[**Căn chỉnh tất cả các mục theo chiều ngang bằng cách sử dụng justify-items** 13](#_Toc81148903)

[**Căn chỉnh tất cả các mục theo chiều dọc bằng cách sử dụng căn chỉnh các mục** 14](#_Toc81148904)

[**Chia lưới thành một mẫu khu vực** 15](#_Toc81148905)

[**Đặt các mục trong vùng lưới bằng cách sử dụng thuộc tính vùng lưới** 16](#_Toc81148906)

[**Sử dụng vùng lưới mà không cần tạo mẫu vùng** 17](#_Toc81148907)

[**Giảm sự lặp lại bằng cách sử dụng chức năng lặp lại** 18](#_Toc81148908)

[**Giới hạn kích thước mặt hàng bằng cách sử dụng hàm minmax** 19](#_Toc81148909)

[**Tạo bố cục linh hoạt bằng cách sử dụng tự động điền** 20](#_Toc81148910)

[**Tạo bố cục linh hoạt bằng cách sử dụng tính năng tự động điều chỉnh** 21](#_Toc81148911)

[**Sử dụng truy vấn phương tiện để tạo bố cục đáp ứng** 22](#_Toc81148912)

[**Tạo lưới trong lưới** 23](#_Toc81148913)

# **Tạo lưới CSS đầu tiên của bạn**

Biến bất kỳ phần tử HTML nào thành một vùng chứa lưới bằng cách đặt thuộc tính của nó displaythành grid. Điều này cung cấp cho bạn khả năng sử dụng tất cả các thuộc tính khác được liên kết với CSS Grid.

**Lưu ý:** Trong CSS Grid, phần tử mẹ được gọi là vùng *chứa* và phần tử con của nó được gọi là *các mục* .

# **Thêm cột bằng lưới-mẫu-cột**

Đơn giản chỉ cần tạo một phần tử lưới không giúp bạn đi xa được. Bạn cũng cần xác định cấu trúc của lưới. Để thêm một số cột vào lưới, hãy sử dụng thuộc grid-template-columnstính trên một vùng chứa lưới như được minh họa bên dưới:

.container {

display: grid;

grid-template-columns: 50px 50px;

}

Điều này sẽ cung cấp cho lưới của bạn hai cột có chiều rộng mỗi cột là 50px. Số lượng tham số được cung cấp cho thuộc grid-template-columnstính cho biết số cột trong lưới và giá trị của mỗi tham số cho biết chiều rộng của mỗi cột.

# **Thêm hàng với lưới-mẫu-hàng**

Lưới bạn đã tạo trong thử thách cuối cùng sẽ đặt số hàng tự động. Để điều chỉnh các hàng theo cách thủ công, hãy sử dụng thuộc grid-template-rowstính giống như cách bạn đã sử dụng grid-template-columnstrong thử thách trước đó.

# **Sử dụng đơn vị Lưới CSS để thay đổi kích thước của cột và hàng**

Bạn có thể sử dụng các đơn vị tuyệt đối và tương đối như pxvà emtrong CSS Grid để xác định kích thước của hàng và cột. Bạn cũng có thể sử dụng những thứ này:

fr: đặt cột hoặc hàng thành một phần nhỏ của không gian có sẵn,

auto: tự động đặt cột hoặc hàng thành chiều rộng hoặc chiều cao của nội dung,

%: điều chỉnh cột hoặc hàng theo chiều rộng phần trăm của vùng chứa của nó.

Đây là mã tạo đầu ra trong bản xem trước:

grid-template-columns: auto 50px 10% 2fr 1fr;

Đoạn mã này tạo ra năm cột. Cột đầu tiên rộng bằng nội dung của nó, cột thứ hai là 50px, cột thứ ba là 10% vùng chứa của nó và đối với hai cột cuối cùng; không gian còn lại được chia thành ba phần, hai phần được phân bổ cho cột thứ tư và một phần cho cột thứ năm.

# **Tạo khoảng cách cột bằng cách sử dụng lưới-cột-khoảng cách**

Cho đến nay trong các lưới bạn đã tạo, tất cả các cột đã được gắn chặt với nhau. Đôi khi bạn muốn có một khoảng trống giữa các cột. Để thêm khoảng cách giữa các cột, hãy sử dụng thuộc grid-column-gaptính như sau:

grid-column-gap: 10px;

Điều này tạo ra 10px không gian trống giữa tất cả các cột của chúng ta.

# **Tạo Khoảng cách hàng bằng cách sử dụng lưới-hàng-khoảng cách**

Bạn có thể thêm khoảng cách giữa các hàng của lưới bằng cách sử dụng grid-row-gapgiống như cách bạn đã thêm khoảng cách giữa các cột trong thử thách trước.

# **Thêm khoảng trống nhanh hơn với khoảng cách lưới**

grid-gaplà một thuộc tính viết tắt cho grid-row-gapvà grid-column-gaptừ hai thử thách trước, thuận tiện hơn để sử dụng. Nếu grid-gapcó một giá trị, nó sẽ tạo ra khoảng cách giữa tất cả các hàng và cột. Tuy nhiên, nếu có hai giá trị, nó sẽ sử dụng giá trị đầu tiên để đặt khoảng cách giữa các hàng và giá trị thứ hai cho các cột.

# **Sử dụng lưới-cột để kiểm soát khoảng cách**

Cho đến thời điểm này, tất cả các thuộc tính đã được thảo luận đều dành cho các vùng chứa lưới. Các grid-columnbất động sản là một trong những đầu tiên để sử dụng vào các mục lưới mình.

Các đường ngang và dọc giả định tạo ra lưới được gọi là *các đường* . Các dòng này được đánh số bắt đầu bằng 1 ở góc trên cùng bên trái của lưới và di chuyển sang phải đối với các cột và xuống đối với các hàng, đếm lên trên.

Đây là những gì các đường trông giống như một lưới 3x3:

https://www.freecodecamp.org/learn/responsive-web-design/css-grid/use-grid-column-to-control-spacing

Để kiểm soát số cột mà một mặt hàng sẽ sử dụng, bạn có thể sử dụng thuộc grid-columntính kết hợp với số dòng mà bạn muốn mặt hàng đó bắt đầu và dừng lại.

Đây là một ví dụ:

grid-column: 1 / 3;

Điều này sẽ làm cho mục bắt đầu ở dòng dọc đầu tiên của lưới bên trái và kéo dài đến dòng thứ 3 của lưới, tiêu thụ hai cột.

# **Sử dụng hàng lưới để kiểm soát khoảng cách**

Tất nhiên, bạn có thể làm cho các mục tiêu thụ nhiều hàng giống như bạn có thể làm với các cột. Bạn xác định các đường ngang mà bạn muốn một mục bắt đầu và dừng lại ở việc sử dụng thuộc grid-rowtính trên một mục lưới.

# **Căn chỉnh một mục theo chiều ngang bằng cách sử dụng justify-self**

Trong CSS Grid, nội dung của mỗi mục nằm trong một hộp được gọi là *ô* . Bạn có thể căn chỉnh vị trí của nội dung trong ô của nó theo chiều ngang bằng cách sử dụng thuộc justify-selftính trên một mục lưới. Theo mặc định, thuộc tính này có giá trị là stretch, giá trị này sẽ làm cho nội dung lấp đầy toàn bộ chiều rộng của ô. Thuộc tính CSS Grid này cũng chấp nhận các giá trị khác:

start: căn chỉnh nội dung ở bên trái ô,

center: căn chỉnh nội dung ở giữa ô,

end: căn chỉnh nội dung ở bên phải ô.

# **Căn chỉnh một mục theo chiều dọc bằng cách sử dụng căn chỉnh tự**

Cũng giống như bạn có thể căn chỉnh một mục theo chiều ngang, cũng có một cách để căn chỉnh một mục theo chiều dọc. Để làm điều này, bạn sử dụng thuộc align-selftính trên một mục. Thuộc tính này chấp nhận tất cả các giá trị giống như justify-selftừ thử thách cuối cùng.

# **Căn chỉnh tất cả các mục theo chiều ngang bằng cách sử dụng justify-items**

Đôi khi bạn muốn tất cả các mục trong CSS Grid của mình chia sẻ cùng một căn chỉnh. Bạn có thể sử dụng các thuộc tính đã học trước đó và căn chỉnh chúng riêng lẻ hoặc bạn có thể căn chỉnh tất cả chúng cùng một lúc theo chiều ngang bằng cách sử dụng justify-itemstrên vùng chứa lưới của bạn. Thuộc tính này có thể chấp nhận tất cả các giá trị tương tự mà bạn đã học được trong hai thử thách trước, sự khác biệt là nó sẽ di chuyển **tất cả** các mục trong lưới của chúng tôi đến sự căn chỉnh mong muốn.

# **Căn chỉnh tất cả các mục theo chiều dọc bằng cách sử dụng căn chỉnh các mục**

Sử dụng thuộc align-itemstính trên một vùng chứa lưới sẽ thiết lập căn chỉnh theo chiều dọc cho tất cả các mục trong lưới của chúng tôi.

# **Chia lưới thành một mẫu khu vực**

Bạn có thể nhóm các ô của lưới lại với nhau thành một *vùng* và đặt tên tùy chỉnh cho vùng đó. Làm điều này bằng cách sử dụng grid-template-areastrên vùng chứa như thế này:

grid-template-areas:

"header header header"

"advert content content"

"advert footer footer";

Đoạn mã trên nhóm các ô của lưới thành bốn vùng; header, advert, content, Và footer. Mỗi từ đại diện cho một ô và mỗi cặp dấu ngoặc kép đại diện cho một hàng.

# **Đặt các mục trong vùng lưới bằng cách sử dụng thuộc tính vùng lưới**

Sau khi tạo mẫu khu vực cho vùng chứa lưới của bạn, như được hiển thị trong thử thách trước, bạn có thể đặt một mục trong khu vực tùy chỉnh của mình bằng cách tham chiếu tên bạn đã đặt cho nó. Để làm điều này, bạn sử dụng thuộc grid-areatính trên một mục như sau:

.item1 {

grid-area: header;

}

Điều này cho phép lưới biết rằng bạn muốn item1lớp đi trong vùng được đặt tên header. Trong trường hợp này, mục sẽ sử dụng toàn bộ hàng trên cùng vì toàn bộ hàng đó được đặt tên là vùng tiêu đề.

# **Sử dụng vùng lưới mà không cần tạo mẫu vùng**

Các grid-areatài sản mà bạn đã học được trong thử thách cuối cùng có thể được sử dụng theo cách khác. Nếu lưới của bạn không có mẫu khu vực để tham chiếu, bạn có thể tạo một khu vực nhanh chóng cho một mục sẽ được đặt như sau:

item1 { grid-area: 1/1/2/4; }

Đây là cách sử dụng các số dòng bạn đã học trước đó để xác định vị trí của khu vực cho mục này. Các số trong ví dụ trên đại diện cho các giá trị sau:

grid-area: horizontal line to start at / vertical line to start at / horizontal line to end at / vertical line to end at;

Vì vậy, mục trong ví dụ sẽ sử dụng các hàng giữa dòng 1 và 2 và các cột giữa dòng 1 và 4.

# **Giảm sự lặp lại bằng cách sử dụng chức năng lặp lại**

Khi bạn sử dụng grid-template-columnsvà grid-template-rowsđể xác định cấu trúc của lưới, bạn đã nhập một giá trị cho mỗi hàng hoặc cột bạn đã tạo.

Giả sử bạn muốn một lưới có 100 hàng có cùng chiều cao. Không thực tế lắm nếu chèn 100 giá trị riêng lẻ. May mắn thay, có một cách tốt hơn - bằng cách sử dụng repeathàm để chỉ định số lần bạn muốn cột hoặc hàng của mình được lặp lại, theo sau là dấu phẩy và giá trị bạn muốn lặp lại.

Đây là một ví dụ sẽ tạo lưới 100 hàng, mỗi hàng cao 50px.

grid-template-rows: repeat(100, 50px);

Bạn cũng có thể lặp lại nhiều giá trị bằng hàm lặp lại và chèn hàm vào giữa các giá trị khác khi xác định cấu trúc lưới. Đây là những gì trông giống như:

grid-template-columns: repeat(2, 1fr 50px) 20px;

Điều này được dịch thành:

grid-template-columns: 1fr 50px 1fr 50px 20px;

**Lưu ý:** Các 1fr 50pxđược lặp đi lặp lại hai lần tiếp theo 20px.

# **Giới hạn kích thước mặt hàng bằng cách sử dụng hàm minmax**

Có một chức năng tích hợp khác để sử dụng grid-template-columnsvà grid-template-rowsđược gọi minmax. Nó được sử dụng để giới hạn kích thước của các mục khi vùng chứa lưới thay đổi kích thước. Để làm điều này, bạn cần xác định phạm vi kích thước chấp nhận được cho mặt hàng của bạn. Đây là một ví dụ:

grid-template-columns: 100px minmax(50px, 200px);

Trong đoạn mã trên, grid-template-columnsđược thiết lập để tạo hai cột; quảng cáo đầu tiên có chiều rộng 100px và hình ảnh thứ hai có chiều rộng tối thiểu là 50px và chiều rộng tối đa là 200px.

# **Tạo bố cục linh hoạt bằng cách sử dụng tự động điền**

Chức năng lặp lại đi kèm với một tùy chọn được gọi là *tự động điền* . Điều này cho phép bạn tự động chèn bao nhiêu hàng hoặc cột có kích thước mong muốn tùy thuộc vào kích thước của vùng chứa. Bạn có thể tạo bố cục linh hoạt khi kết hợp auto-fillvới minmax, như sau:

repeat(auto-fill, minmax(60px, 1fr));

Khi vùng chứa thay đổi kích thước, thiết lập này sẽ tiếp tục chèn các cột 60px và kéo dài chúng cho đến khi nó có thể chèn một cột khác. **Lưu ý:** Nếu vùng chứa của bạn không thể chứa tất cả các mặt hàng của bạn trên một hàng, nó sẽ chuyển chúng xuống hàng mới.

# **Tạo bố cục linh hoạt bằng cách sử dụng tính năng tự động điều chỉnh**

auto-fithoạt động gần như giống hệt nhau auto-fill. Sự khác biệt duy nhất là khi kích thước của vùng chứa vượt quá kích thước của tất cả các mục được kết hợp, hãy auto-filltiếp tục chèn các hàng hoặc cột trống và đẩy các mục của bạn sang một bên, trong khi thu auto-fitgọn các hàng hoặc cột trống đó và kéo dài các mục của bạn để vừa với kích thước của vùng chứa .

**Lưu ý:** Nếu vùng chứa của bạn không thể chứa tất cả các mặt hàng của bạn trên một hàng, nó sẽ chuyển chúng xuống hàng mới.

# **Sử dụng truy vấn phương tiện để tạo bố cục đáp ứng**

CSS Grid có thể là một cách dễ dàng để làm cho trang web của bạn phản hồi nhanh hơn bằng cách sử dụng các truy vấn phương tiện để sắp xếp lại các vùng lưới, thay đổi kích thước của lưới và sắp xếp lại vị trí của các mục.

Trong bản xem trước, khi chiều rộng khung nhìn từ 300px trở lên, số lượng cột thay đổi từ 1 thành 2. Vùng quảng cáo sau đó chiếm hoàn toàn cột bên trái.

# **Tạo lưới trong lưới**

Việc chuyển một phần tử thành một lưới chỉ ảnh hưởng đến hành vi của các phần tử con trực tiếp của nó. Vì vậy, bằng cách biến một hậu duệ trực tiếp thành một lưới, bạn có một lưới trong một lưới.

Ví dụ: bằng cách đặt các thuộc tính displayvà grid-template-columnsthuộc tính của phần tử với item3lớp, bạn tạo một lưới trong lưới của mình.