

Importante

25/03/2020 , 12:39:41

FORMAS PARA PONER GRID  
en navegadores antiguos de IE

es una comunidad dedicada al aprendizaje de est ndares web (HTML, CSS, y accesibilidad digital. Encontrar  tutoriales, foros, anuncios de trabajo, cuestionarios, concursos y mucho m s. [Cree su perfil](#) en segundos para participar o consulte nuestra [pol tica de protecci n de datos](#) .

[Ocultar](#)Nivel 

## CSS Grid Layout en producci n?

TUTORIAL

CSS

Publicado por [Rafael](#) en 12 de julio de 2018 (27,993 lecturas)[posicionamiento rejilla GridLayout](#)

**Flexbox** y **Grid Layout** se encuentran entre las especificaciones que m s han contribuido a la evoluci n de los **estilos CSS** en los  ltimos a os. Verdaderas bendiciones para el integrador, estos dos enfoques han revolucionado literalmente los m todos de dise o cl sicos y han enviado al olvido t cnicas antiguas y temblorosas.

Los primeros **borradores** de las **especificaciones** de **Flexbox** datan de **2009**, mientras que el **primer borrador** oficial de **Grid Layout** apareci  en **2012**. Aunque ambos fueron compatibles simult neamente desde el lanzamiento de Internet Explorer 10 en octubre de 2012, la adopci n de Grid Layout es todav a en su infancia debido a ... Internet Explorer.

 Cu les son las razones de esta reticencia y, sobre todo, c mo usar Grid Layout en la producci n actual?

## Compatibilidad aceptable ...

A mediados de 2018, las cifras ya eran muy elocuentes: 90% de apoyo en Francia y 88% en todo el mundo. La gran mayor a de los navegadores de escritorio y m viles reconocen esta especificaci n de varias versiones, como se muestra en la referencia esencial [ Puedo usar ?](#)

Solo ausente hasta la fecha: Opera Mini (0.14% de uso en Francia) y ... Internet Explorer versi n 9 y posteriores (0.1% de uso en Francia).

Tenga en cuenta que sitios como Slate, Financial Times, Stripe, Slack, Medium y Yandex utilizan **Grid**

**Layout en producci n:** <http://www.gridexamples.com/>

IE	Edge	Firefox	Chrome	Safari	iOS Safari	Opera Mini	Chrome for Android	UC Browser for Android	Samsung Internet
					9.3				
					10.2				
		52	49		10.3				
		60	66		11.2				6.2
11	17	61	67	11.1	11.4	all	67	11.8	7.2
	18	62	68	12	12				
		63	69	TP					
			70						

## Pero soporte parcial para Internet Explorer

Microsoft Edge 16, lanzado en septiembre de 2017, reconoce la última versión de la especificación Grid en su totalidad.

Por otro lado, las versiones anteriores de Internet Explorer están un poco más atrás: IE10, IE11 y las primeras ediciones de Edge admiten oficialmente Grid Layout, pero se refieren a una versión antigua de la especificación que es mucho menos completa, y muchas Por lo tanto, falta funcionalidad.

## Una cuestión de versión

La **implementación** de **Grid** por las diferentes versiones de **Internet Explorer** es **simple** y complicada al mismo tiempo:

- IE9 y a continuación: sin soporte
- IE10, IE11, Edge 12-15: especificación anterior (soporte parcial)
- Edge 16+: nueva especificación

*En el resto de este artículo, el término "Internet Explorer" se refiere a las versiones "IE10, IE11, Edge 12-15".*

El soporte parcial de Internet Explorer se explica por el hecho de que la especificación de diseño de cuadrícula ha evolucionado más rápido que ella, y se manifiesta principalmente por:

- Algunas **características que faltan** (que simplemente no existen en IE)
- De **diferentes características** (que se escriben diferente o de otro modo operan en IE que otros)
- Algunos **bugs**

## Funciones faltantes

Entre las principales deficiencias de estas versiones, está la falta de soporte para ciertas características principales que se detallan a continuación.

### Sin colocación automática

Primera característica notable en la integración de cuadrículas en Internet Explorer: no existe una noción de colocación automática de elementos en IE: **debe colocar cada elemento de forma individual y explícita en cada celda.**

Se deduce lógicamente que **IE no reconoce ninguna** de las siguientes **propiedades** estándar:

- `grid-auto-columns`
- `grid-auto-rows`
- `grid-auto-flow`
- `grid` (propiedad global acortada)

## Sin canales `gap`

No temas nada por tu hogar, las únicas canaletas que se verán privadas en Internet Explorer son aquellas que permiten `espaciar` las `filas` y las `columnas` de la `cuadrícula`.

Las `propiedades` `grid-row-gap`, `grid-column-gap` y  (y su más reciente equivalente `gap` corta) `SET` para `gestionar` fácilmente las `canaletas` en `Diseño` de `cuadrícula` están totalmente `ausentes` del vocabulario de `IE` y no pasará nada si cruza la vuelta de una línea de código

La única `salida` es `proporcionar` una `columna` (o fila) `adicional` que `actuará como` una `canaleta`:

```
.container {  
  display: -ms-grid; /* version IE */  
  display: grid;  
  grid-gap: 1rem;  
  -ms-grid-columns: 1fr 1rem 1fr; /* version IE */  
  grid-template-columns: 1fr 1fr;  
}
```

CSS

## No hay áreas nombradas

El `diseño` de `cuadrícula` ofrece un `método` de `construcción` alternativo: `zonas` y `líneas` con `nombre`.

Las `propiedades` `grid-template-areas` y `grid-area` permiten `crear` `zonas` con `nombre` y `ubicarse` allí ... pero `tampoco` son `compatibles` con `Internet Explorer`.

Este tipo de sintaxis, por lo tanto, no funcionará en IE:

```
.container {  
  display: grid;  
  grid-template-areas: "a b"  
                      "c d";  
}  
.child {  
  grid-area: a;  
}
```

CSS

## Varias otras infracciones

Finalmente, no olvidemos que `Internet Explorer` `ignora` totalmente:

- Valores `auto-fit` y `auto-fill` muy prácticos para la construcción de redes automáticas,
- La palabra clave `dense` utilizada para completar los cuadros que quedan libres en la cuadrícula.



## Características divergentes

Desde su versión borrador (adoptada rápidamente por IE10 en ese momento), ciertas propiedades o valores han cambiado ... sin que Internet Explorer siga al vagón.

### Diferentes propiedades ... y prefijado

Es un juego del que nos hubiera gustado prescindir: tenemos que adivinar qué propiedades tienen el prefijo de Internet Explorer y, sobre todo, sentir cuál es su pequeño nombre en este navegador porque puede ser diferente del resto del mundo.

Aquí está la lista de propiedades de cuadrícula reconocidas en IE pero bajo un nombre diferente de los estándares:

propiedad estándar	IE equivalente
<code>display: grid</code>	<code>display: -ms-grid</code>
<code>grid-template-columns</code>	<code>-ms-grid-columns</code>
<code>grid-template-rows</code>	<code>-ms-grid-rows</code>
<code>grid-row-start</code>	<code>-ms-grid-row</code>
<code>grid-column-start</code>	<code>-ms-grid-column</code>
<code>grid-row-end</code>	<code>-ms-grid-row-span</code> (más o menos equivalente)
<code>grid-column-end</code>	<code>-ms-grid-column-span</code> (más o menos equivalente)
<code>align-self</code>	<code>-ms-grid-row-align</code>
<code>justify-self</code>	<code>-ms-grid-column-align</code>

Ejemplo de código CSS estándar:

```
.container {  
  display: grid;  
  grid-template-rows: 20rem 100px 1fr;  
  grid-template-columns: 100px 1fr 10rem;  
}  
.child {  
  grid-column: 1 / 3;  
}
```

CSS

Equivalente en Internet Explorer:

```
.container {  
  display: -ms-grid;  
  -ms-grid-rows: 20rem 100px 1fr;  
  -ms-grid-columns: 100px 1fr 10rem;  
}  
.child {  
  -ms-grid-column: 1;  
  -ms-grid-column-span: 2;  
}
```

CSS

*Afortunadamente, la herramienta de corrección automática descrita en detalle a continuación administra muy bien estas propiedades sin que tenga que preocuparse por ellas.*

**repeat()**  
**repetir ()**

En las primeras especificaciones, la **función** de **valor** **repeat()** nació en una versión totalmente diferente de la que conocemos hoy. Internet Explorer está ahí para recordarnos esos días.

valor estándar	IE equivalente
repeat(10, 1fr 2rem)	(1fr 2em)[10]

Por lo tanto, es perfectamente posible **aplicar** este **valor** **para** **repetir** **patrones** de **track** pista, pero teniendo cuidado de respetar la sintaxis específica de IE.

Bugs

Ah, bueno, obviamente no podríamos cortarlo: Internet Explorer, como otros navegadores, a veces es un poco defectuoso cuando se trata de CSS, y más particularmente cuando una especificación es reciente. A veces tiene que esperar las actualizaciones del navegador para ver que algunos errores se resuelven o desaparecen.

Para ayudarlo a esperar, Rachel Andrew, papa de Grid Layout, ha inventado un espacio de colaboración en Github llamado **"Gridbugs"** y que compila un conjunto de errores (y soluciones) frecuentes sobre Grid Layout: [https://github.com/rachelandrew/\\_gridbugs](https://github.com/rachelandrew/_gridbugs)

No dudes ni un segundo en colocar este enlace en tus favoritos.

README.md

## GridBugs

Inspired by [Flexbugs](#) this list aims to be a community curated list of CSS Grid Layout bugs, incomplete implementations and interop issues. [Grid shipped into browsers](#) in a highly interoperable state, however there are a few issues - let's document any we find here.

While I'd like to focus on issues relating to the [Grid Specification](#) here, it is likely that related specs such as Box Alignment will need to be referenced. If you think you have spotted an issue and it seems to relate to Grid, [post it](#). We can work out the details together and make sure browser or spec issues get logged in the right place.

*Also, please raise an issue if I am wrong about any of these or you have more info or examples to add. Quite possibly I've pointed the finger at the wrong UA or missed a change in the spec. Help to make the list accurate appreciated!*

## Entonces, ¿qué hacemos concretamente?

En la práctica, tenemos tres soluciones para gestionar las deficiencias actuales de los navegadores:

- nuestro cerebro
- la herramienta Autoprefixer **PostCSS** , **PostCSS es una herramienta de desarrollo de software que utiliza complementos basados en JavaScript para automatizar las operaciones de rutina de CSS**
- la **regla CSS @supports ()** .

No entraré en los detalles de la primera opción, sino que me centraré en las dos siguientes.

### Autoprefixer

[Autoprefixer](#) es una **herramienta** (excelente) que le **permite agregar todos** los **prefijos CSS a sus propiedades de forma automatizada** y, sobre todo, solo **para los navegadores** que aún lo **requieren**.

**De forma predeterminada, el diseño de cuadrícula no está activado en Autoprefixer** . Debe activarlo en las opciones yendo **grid: false** a **grid: true** .

```
// Styles -> autoprefixer -> /dist
gulp.task('styles', function () {
  return gulp.src('./src/assets/css/styles.css')
    .pipe(autoprefixer({
      grid: true
    }))
    .pipe(gulp.dest('./dist/assets/css/'));
});
```

Otro punto importante: la última versión de Autoprefixer ha progresado enormemente en la administración de prefijos y funciones para Internet Explorer (desde la versión 8.6.4 precisamente), ahora es mucho más confiable y altamente recomendable en sus proyectos (no solo para el diseño de cuadrícula, por cierto).

Para comprender cómo funciona el nuevo Autoprefixer en detalle, lo invito a consultar este recurso de habla inglesa muy completo: <https://css-tricks.com/css-grid-in-ie-css-grid-and-the-new-autoprefixer/>

## ¿Qué trae Autoprefixer?

Como su nombre indica, Autoprefixer agrega automáticamente los prefijos ( `-webkit-`, `-moz-`, `-ms-` etc.) cuando sea necesario en una propiedad o un valor.

Pero va mucho más allá ya que es capaz de convertir completamente las propiedades para hacerlas compatibles.

Las siguientes propiedades, por ejemplo, no tienen equivalente en Internet Explorer, pero se reconocen gracias a Autoprefixer, que las transforma en "sintaxis compatible con IE":

- `grid-template-areas`
- `grid-template`
- `grid-row-end`
- `grid-column-end`
- `grid-row`
- `grid-column`
- `grid-area`
- `grid-row-gap`
- `grid-column-gap`
- `grid-gap`

Ejemplo de código estándar:

```
.container {  
  display: grid;  
  grid-gap: 1em;  
  grid-template: "a b" minmax(100px, 1fr)  
                 "c d" 20em / 1fr 1fr;  
}  
.child {  
  grid-area: a;  
}
```

CSS

Convertido usando Autoprefixer a:

```
.container {  
  display: -ms-grid;
```

CSS

```

display: grid;
grid-gap: 1em; 1ºfila 2ºfila 3ºfila
-ms-grid-rows: minmax(100px, 1fr) 1em 20em;
-ms-grid-columns: 1fr 1em 1fr;
grid-template: "a b" minmax(100px, 1fr)
               "c d" 20em / 1fr 1fr;
               Fila / columna
}
.child {
  -ms-grid-row: 1;
  -ms-grid-column: 1;
  grid-area: a;
}

```

¡Y salta, empackado se pesa! Es mucho más fácil administrar el caso de Internet Explorer gracias a este tipo de herramienta.

Tenga en cuenta que gracias a Autoprefixer, nada nos impide ahora usar el método de ubicación a través de áreas con nombre.



## @supports ()

La regla @supports, también llamada "regla condicional", detecta el reconocimiento de ciertas propiedades CSS dentro del navegador.

Introducida en las especificaciones del módulo condicional de CSS, al igual que Media Queries, la regla @supports está cerca de lo que una herramienta como Modernizr puede ofrecernos, es decir, detectar el soporte o no de una funcionalidad CSS en su visitante para proporcionar una alternativa si es necesario.

- @supports desde entonces ha sido reconocido por Microsoft Edge, por lo que IE11 y versiones posteriores lo ignoran por completo.



- la prueba (`display: grid`) solo se reconoce desde Edge 16

```
@supports (display: grid) {  
    /* will return true for any browser supporting the spec, Edge included */  
}
```

CSS

Esta **sintaxis** `@supports` nos permite mantener solo navegadores que sean totalmente compatibles con versiones estables de Grid Layout, excluyendo IE10, IE11 y Edge 15 que solo reconocen la versión prefijada.

## Cuales son mis opciones?

Finalmente, veamos qué opciones están disponibles para mí con respecto a las restricciones de compatibilidad de mi navegador.

### Soy libre ! (Borde mínimo)

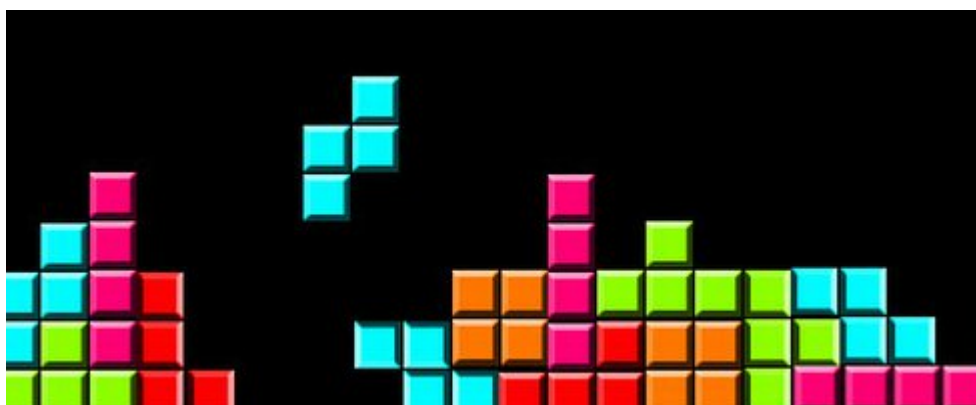
1. Aplicamos el diseño de cuadrícula solo para navegadores modernos (Edge +) a través de `@supports`
2. no es necesario reparar automáticamente

### Absolutamente debo tener en cuenta IE10 e IE11

1. Aplicamos el diseño de cuadrícula para todos los navegadores
2. activamos Autoprefixer para llenar los vacíos en IE
3. Usamos propiedades y funcionalidades que Autoprefixer puede tener en cuenta y tenemos cuidado donde ponemos los dedos para no ser pellizcados con demasiada fuerza.

### Absolutamente debo tener en cuenta IE9 (¡OMG!)

1. será necesario proporcionar una alternativa en posiciones ancestrales ( float/ inline-block). Coraje!
2. Aplicamos el diseño de cuadrícula solo para navegadores modernos (Edge) a través de la regla `@support`
3. no es necesario corregir automáticamente, pero tenemos que hacer el doble de trabajo (dos versiones de posicionamiento para mantener)



# Conclusión

Nada como un pequeño ejemplo concreto para ilustrar todo este artículo y concluir con estilo.

El objetivo es diseñar una plantilla de 3 elementos de ancho idéntico (1/3 del espacio disponible) y separados por una canaleta de 1em.

```
.container {  
  display: flex; /* 1 */  
  justify-content: space-between; /* 2 */  
}  
  
.container > * {  
  flex-basis: calc(100% / 3 - (1em * 2 / 3)); /* 3 + 4 */  
}  
  
@supports (display: grid) { /* 5 */  
  .container {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-gap: 1em;  
  }  
}  
  
/*  
Explications :  
1- l'alternative est ici Flexbox (prévoir Autoprefixer pour les navigateurs  
nécessiteux)  
2- on prévoit une "gouttière" de 1em  
3- on calcule la taille des 3 enfants par rapport à la gouttière  
4- on préfère flex-basis à width pour éviter de devoir écraser width dans Grid  
Layout  
5- seuls les navigateurs modernes (Edge) se reconnaîtront ici  
*/
```

CSS

## Comentarios

tzi

12 de julio de 2018 a las 2:35 p.m.

Hola raphaël

Gracias por este artículo Eso es genial :)

En otro artículo aconsejaste usar `@supports (grid-area: auto) {}` para detectar navegadores modernos.

Finalmente, ya no necesitamos este consejo?

Thomas

**Rafael**

*12 de julio de 2018 a las 4:03 p.m.*

@tzi: Hola :)

De hecho, finalmente fue un error de Codepen: pantalla: la cuadrícula funciona correctamente (= no se reconoce en el borde <16, según se desee)

<https://blog.goetter.fr/2017/08/26/a-story-of-grid-layout-of-microsoft-and-supports/#comment-650>

**Widgets**

*13 de julio de 2018 a las 12:00 a.m.*

Hola !

Gran artículo. Gracias. Es sorprendente lo que podemos aprender de este sitio y especialmente de sus artículos.

**Rafael**

*13 de julio de 2018 a las 9:59 a.m.*

@Bidules: Muchas gracias :)

**niuxe**

*14 de julio de 2018 a las 3:57 p.m.*

Tenga en cuenta IE y Edge, ¡qué herejía!

**Wonderfoule**

*19 de julio de 2018 a las 10:54 a.m.*

Merci Raphaël pour ce récap très complet. Il est encore difficile de se passer d'IE qui encore utilisé sur les sites institutionnels ou par nos clients eux-mêmes. Perso, j'ai encore recours aux commentaires conditionnels malgré l'utilisation d'Autoprefixer.

**eckri**

*08 Janvier 2019 à 10h00*

J'utilise flexbox qui marche avec tout et qui fait pareil que grid sans s'emmerder avec les navigateurs

je ne comprend toujours pas pourquoi flexbox n'a pas de gap mais je remplace par une bordure blanche ou transparente

Grid n'a pas des fonctions radicalement supérieures à flexbox

j'utiliserais peut-être Grid dans quelques années

merci pour le texte

**Crazy88**

*19 Janvier 2019 à 05h57*

@eckri: Grid est bi-dimensionnel alors que Flexbox est mono-dimensionnel. De plus Grid est orienté structure alors que Flexbox est orienté contenu. On peut tout à fait coupler les 2.

**zebruno**

*26 Janvier 2019 à 00h10*

Bonjour, c'est peut-être une question à la con, mais je la pose quand même.

Au final il n'est question que de la problématique des anciennes versions d'IE. Quid de Firefox et Chrome ?

Grid est implémenté par exemple à partir de la version 52 de Firefox.

Est-ce que la question n'est pas abordée car personne n'est censé être bloqué sur une vieille version de Firefox, ce dernier se mettant de toutes façons à jour sans considération de version de système d'exploitation ou autre ?

Merci

**tannoy**

*10 Décembre 2019 à 09h01*

Salut Raphaël,

Comment tu gères pour les versions d'iOS 9 ?

Merci

Alsacr ations es una comunidad dedicada al dise o de sitios web y aplicaciones web de calidad, gracias a los est ndares W3C, hojas de estilo CSS, lenguajes HTML y JavaScript, y accesibilidad. Realizado por la agencia web [Alsacreations.fr](http://Alsacreations.fr) · [Syazen hosting](http://Syazen hosting) · [Acerca de y avisos legales](#) · [Datos personales](#)

<https://www.alsacreations.com/>