MODERN LAYOUTS WITH CSS GRID

REVOLUTIONISING VISUAL DESIGN ON THE WEB

mozilla
Me trying to Write CSS

CSS FLOATS
Y U NO CLEAR YOURSELF?!?

Tricky CSS inheritance issue on Monday

Tricky CSS inheritance issue on Friday

CHALLENGE ACCEPTED

Important
all the things!
Web layouts over the years
2017: the year of CSS grid

March 2017

<table>
<thead>
<tr>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>

October 2017

<table>
<thead>
<tr>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evolution of CSS Specifications

CSS1
Recommendation: 17 Dec 1996

CSS2
Recommendation: 12 May 1998

CSS2.1
Recommendation: 7 Jun 2011

CSS2.2
Working draft: 12 Apr 2016

CSS3
Decision to modularise: 14 Apr 2000
(26 modules)

CSS Snapshot 2017
(88 modules)
Team Layout

Display
Box Alignment
Flexible Box
CSS Grid
Ruby Layout
Inline Layout
Positioned Layout
Writing Mode
CSS Shapes
CSS Transforms

And more...
Grid vs. flexbox?
Grid AND Flexbox
Flexbox ➔ single dimension

- `flex-direction: row`
- `flex-direction: column`
What is CSS grid?

Defines a **two-dimensional** grid-based layout system, optimized for user interface design.
Basic terminology

- Grid line
- Grid track
- Grid area
- Grid cell
- Grid gap
Flexbox and Grid

Based on the container-child relationship
"Grid works from the **container** in, other layout methods start with the **item**"

—Rachel Andrew
# Layout technique: inline-block

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Item C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item D</td>
<td>Item E</td>
<td>Item F</td>
</tr>
</tbody>
</table>

```css
.inlineblock_item {
  display: inline-block;
  width: calc(100% / 3);
}
```
## Layout technique: float

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Item C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item D</td>
<td>Item E</td>
<td>Item F</td>
</tr>
</tbody>
</table>

```css
.float__item {
  float: left;
  width: calc(100% / 3);
}
```
## Layout technique: flex

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Item C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item D</th>
<th>Item E</th>
<th>Item F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```css
.flexbox {
  display: flex;
  flex-wrap: wrap;
}

.flexbox__item {
  flex: 1 1 calc(100% / 3);
}
```
“Grid is the only layout technique that establishes a **relationship** between rows and columns of grid items.”
CSS grid basics

Define your grid.

Place items in the grid.
Properties on the Grid container
Defining a grid

```html
<div class="grid1">
  <div class="grid1__item">
    <p>Item A</p>
  </div>
  <div class="grid1__item">
    <p>Item B</p>
  </div>
  <div class="grid1__item">
    <p>Item C</p>
  </div>
  <div class="grid1__item">
    <p>Item D</p>
  </div>
  <div class="grid1__item">
    <p>Item E</p>
  </div>
  <div class="grid1__item">
    <p>Item F</p>
  </div>
</div>
```

```css
.grid1 {
  display: grid;
  grid-template-columns: 5em 5em 5em;
  grid-template-rows: 5em 5em;
  grid-gap: 1em;
}
```
"Cede control of your designs to the browsers that render them."
The `fr` unit

Represents a fraction of the free space in the grid container.

```
.grid4 {
  display: grid;
  grid-template-columns: 150px 1fr 2fr;
}
```
Fluid CSS grid

```css
.container {
  display: grid;
  grid-template-columns: repeat(3, 3fr 2fr);
}
```
The `minmax()` function

Defines a size range for columns or rows in the grid.

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Item C</th>
</tr>
</thead>
</table>

```css
.grid5 {
  display: grid;
  grid-template-columns: minmax(200px, 1fr) 300px 300px;
}
```
The `repeat()` function

To specify a large number of columns or rows that follow a similar pattern

| Item | Item | Item | Item | Item | Item | Item | Item |

```css
.grid6 {
    display: grid;
    grid-template-columns: repeat(4, 75px 120px);
}
```
**auto-fill vs. auto-fit**

Allow browser to determine how many tracks to create depending on track size.

\[
\text{repeat(auto-fill, 100px);}
\]

\[
\text{repeat(auto-fit, 100px);}
\]

**auto-fit** collapses empty tracks.
auto-fill versus auto-fit

| A | B | C | D | E | F |

```css
.keyword {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
}
```
Responsive grid without media queries

```css
.container {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(10em, 1fr));
}
```
Explicit Grid and Implicit Grid

```css
.grid-container {
  display: grid;
  grid-template-columns: 150px 150px 150px; /* three columns */
  grid-template-rows: 150px 150px; /* two rows */
}

.item {
  grid-column: 5 / 6; /* grid-column-start: 5; grid-column-end: */
  grid-row: 2 / 3; /* grid-row-start: 2; grid-row-end: 3; */
}
```

The browser will create an implicit grid to hold items placed outside the grid.
Aligning your grid items
<table>
<thead>
<tr>
<th>Property</th>
<th>Axis</th>
<th>Aligns</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>justify-content</td>
<td>main/inline</td>
<td>content within element (effectively adjusts padding)</td>
<td>block containers, flex containers and grid containers</td>
</tr>
<tr>
<td>align-content</td>
<td>cross/block</td>
<td></td>
<td>block-level boxes, absolutely-positioned boxes and grid items</td>
</tr>
<tr>
<td>justify-self</td>
<td>inline</td>
<td>element within parent (effectively adjusts margins)</td>
<td>absolutely-positioned boxes and grid items</td>
</tr>
<tr>
<td>align-self</td>
<td>cross/block</td>
<td></td>
<td>flex-containers and grid-containers</td>
</tr>
<tr>
<td>justify-items</td>
<td>inline</td>
<td>items inside box (controls child items)</td>
<td>block containers and grid containers</td>
</tr>
<tr>
<td>align-items</td>
<td>cross/block</td>
<td></td>
<td>flex-containers and grid-containers</td>
</tr>
</tbody>
</table>

Source: CSS Box Alignment Module Level 3
justify/align-content

content-distribution properties

A

B

c

D

.content {
  justify-content: normal;
  align-content: normal;

  display: grid;
  grid-template-columns: repeat(3, 25%);
  grid-template-rows: repeat(3, 20%);
  grid-gap: 1em;
  grid-template-areas:
    "a b b"
    "a b b"
    "c c d";
}

.content_itemA {
  grid-area: a;
}

.content_itemB {

<table>
<thead>
<tr>
<th>Values</th>
<th>justify-content</th>
<th>align-content</th>
</tr>
</thead>
<tbody>
<tr>
<td>center</td>
<td>![center icon]</td>
<td>![center icon]</td>
</tr>
<tr>
<td>start</td>
<td>![start icon]</td>
<td>![start icon]</td>
</tr>
<tr>
<td>end</td>
<td>![end icon]</td>
<td>![end icon]</td>
</tr>
<tr>
<td>space-around</td>
<td>![space-around icon]</td>
<td>![space-around icon]</td>
</tr>
<tr>
<td>space-between</td>
<td>![space-between icon]</td>
<td>![space-between icon]</td>
</tr>
<tr>
<td>space-evenly</td>
<td>![space-evenly icon]</td>
<td>![space-evenly icon]</td>
</tr>
</tbody>
</table>
justify/align-self

self-alignment properties

A

B

class .self {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-gap: 1em;
  grid-auto-rows: calc(25% - 1em);
  grid-template-areas:
    "a a b b"
    "a a b b"
    "c c d d"
    "c c d d";
}
class .self_itemA {
  grid-area: a;
}
class .self_itemB {
  grid-area: b;
}
justify/align-items

defaults for justify/align-self

```css
.items {
  justify-items: normal;
  align-items: normal;

display: grid;
grid-template-columns: repeat(4, 1fr);
grid-gap: 1em;
grid-auto-rows: calc(25% - 1em);
grid-template-areas:
  "a a b b"
  "a a b b"
  "c c d d"
  "c c d d";
}
.items_itemA {
  grid-area: a;
}
```

Unter den Straßenzügen die ausgebauten Kiosken, Lichtglanz auf dem Wasserspiegel.
Self-alignment properties, maybe?

These are not the borders you are looking for
Flexbox to the rescue

That's more like it
Properties on the grid item
Grid placement with grid lines / named lines

grid-template-columns: [first sidebar-start] 250px [sidebar-end content-start] 1fr [content-end last]
grid-template-rows: [first header-start] 100px [header-end content-start] 1fr [content-end footer-start]
Using `grid-column` and `grid-row`
Grid placement with named areas

```css
.grid-container {
  display: grid;
  grid-template-areas: "logo stats" "score stats" "board board" "... controls";
}
.logo { grid-area: logo; }
.score { grid-area: score; }
.stats { grid-area: stats; }
.board { grid-area: board; }
.controls { grid-area: controls; }
```
Using grid areas

```css
.grid3 {
  display: grid;
  grid-template-columns: 150px 150px 150px;
  grid-template-rows: 100px 100px 100px;
  grid-template-areas: 'a . .'
                       'a . c'
                       'b . c';
}
.grid3 .a { grid-area: a; }
.grid3 .b { grid-area: b; }
.grid3 .c { grid-area: c; }
```
CONNECTING THE DOTS

Founded in 2005, Whitespace is a culturally savvy design agency providing creative direction, branding and interactive services across various mediums, platforms and time zones.
Braun HiFi
Steuergerät
audio 300

In der modernen Aufnahmotechnik ist keine Minute zu groß, kein Aufwand zu kostspielig, um auch allerletzte Feinheiten einer Interpretation zu erhalten.

Demgegenüber gibt es aber nur wenige HiFi Geräte, die es fertigbringen, den ganzen Nuancenreichtum wieder vollends hörbar zu machen.

Ein Gerät mit diesen Fähigkeiten ist das neue „audio 300“, eine Kompaktanlage mit ungewöhnlicher technischer Ausrüstung und einem formalen Konzept, das Braun höchste internationale Auszeichnungen eintrug.
In der modernen Aufnahmetechnik ist keine Mühle zu groß, kein Aufwand zu kostspielig, um auch allerletzte Feinheiten einer Interpretation zu erhalten.

Demgegenüber gibt es aber nur wenige Hifi-Geräte, die es fertigbringen, den ganzen Nuancenreichtum wieder vollends hörbar zu machen.

Ein Gerät mit diesen Fähigkeiten ist das neue "audio 300", eine Kompaktanlage mit ungewöhnlicher technischer Ausrüstung und einem formalen Konzept, das Braun höchste International-Auszeichnungen eintrug.
In der modernen Aufnahmetechnik ist heute Würde zu erlangen, die auf der Basis technischer Kenntnisse und praktischer Erfahrungen in einer Interpretation zu erlangen.

Braun Hifi Steuergerät audio 300

Ein Gerät mit einem Flächengewinn bei der neuen "Audio 300", eine Kompaktstrophe mit ungewöhnlicher technischer Ausführung und einem formellen Konzept, das Braun keinesfalls in seiner Ausführung vernachlässigt.

```
.steuergerät {
  grid-gap: 0.5em;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: calc((100vh - 3em) / 3) calc((100vh - 3em) / 3);
  grid-template-areas: "title title title" "text text text" "key key key" "detail1 detail2 detail3" "detail4 detail5 detail6";
}
@media screen and (min-aspect-ratio: 8/5) and (min-height: 36em) {
  .steuergerät {
    grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
    grid-template-rows: calc((100vh - 3em) / 3) calc((100vh - 3em) / 3);
    grid-template-areas: "title key key . ." "text detail1 key detail2 detail3" "text . . detail4 . .";
  }
```
With his bold, stripped-back typographic approach, Hanson Ho founded H55 in 1999. It is now one of the most recognised and celebrated graphic design studios in Singapore.
THINK GLOBAL GROW LOCAL


WE FIND THE OFFICES OF FIGTREE NETWORK TUCKED AWAY IN A QUIET BUT COOL STREET IN THE WANCHAI DISTRICT OF THE CITY. AFTER VISITING THE MONOCLE STORE ACROSS THE ROAD, WE POP IN TO CHAT WITH DAVID AND OUR CONVERSATION QUICKLY LEADS TO THE RISE OF THE CREATIVE SECTOR IN HIS NEW HOMELAND.

DAVID: I think there is a push going on, though not as strongly so in Singapore, which is a shame. In Hong Kong the government have identified the creative industries as one of six areas of growth as they try to expand beyond finance, property and shipping.

Sure, this is the big leap that Asian businesses, especially Chinese businesses need to make. We've had the experience of going through that process with HTC. Many businesses are not yet ready to make the leap so our first strategy when we came here was to help Asian businesses build global brands. But it's not as simple as just doing communications. They have to invest quite heavily in a long-term strategy, acquire new capabilities they never had before - innovation, marketing - it is a big jump for companies that are used to making things for others, and not designing things for themselves.

HTC was a bit further along because they designed stuff for mobile networks so they had all the competencies they needed and the scale to make it happen. But they had to do it with HTC's brand identity.

How did Figtree Network get started?

We are about seven years old. Orange was the foundation client. I joined shortly after we were incorporated. You need eight good clients for an agency. HTC is our most recent large one and we have a couple more big ones in the UK right now.

OK, so how many others does Figtree have?

Three, we've got London - which is our HQ - there are about forty people there. About three or four work in Paris, then eight of us here in Hong Kong.

Did you initially work in the London branch?

Yeah, back in the days when we were three guys in a garage, just like Google!

What brought you to Hong Kong?

Opportunity and change of scene. I was getting a bit bored with London; the weather; the bustle; the high-tax, low-growth. I found that I keep stepping over at Hong Kong when I was on my way to Taiwan to meet with HTC, I used to come here before flying back to London to catch up with some mates. I gradually fell in love with the place, so I thought: 'Well, what do we do now? We work here, let's come here.'
The Orange Issue

.container {
  display: grid;
  grid-template-columns: repeat(9, 1.25em);
  grid-template-rows: repeat(6, 1.25em);
}

h2 {
  grid-row: 3 / 5;
  grid-column: 1 / 10;
}

.container::before {
  grid-row: 1 / 7;
  grid-column: 1 / 7;
}

.container::after {
  grid-row: 1 / 7;
  grid-column: 4 / 10;
}
Building a grid layout

Earlier this year, San Francisco-bred artist and designer Tycho announced a new album on the way for 2016, but gave scant details about the project, leaving fans to speculate about just what the ambient master had in store. Fans need no longer worry as Tycho's fifth studio album, Epoch, has arrived in grand fashion.
Good ol' pencil and paper
Tycho

Artist Spotlight

Tycho is an American ambient music project led by Scott Hansen as primary composer, songwriter and producer. Hailing from San Francisco, California, he is known as ISO50 for his photographic and design works. His music is a combination of downtempo vintage-style synthesizers and ambient melodies.

Read more.
“Websites do **NOT** have to look the same in every browser.”
Tycho

ARTIST SPOTLIGHT

Tycho is an American ambient music project led by Scott Hansen as primary composer, songwriter and producer. Hailing from San Francisco, California, he is known as ISO50 for his photographic and design works. His music is a combination of downtempo vintage-style synthesizers and ambient melodies.

READ MORE

10.2 2010 TORONTO
THE DRAKE HOTEL

TOUR DATES

Fallback layout


CSS grid in action

```css
@supports (display:grid) {
  @media (min-width: 42em) and (min-height: 27em) {
    main {
      max-width: none;
      padding: 0;
      display: grid;
      grid-template-columns: 2fr minmax(10em, max-content) 1fr;
      grid-template-rows: 35vh 40vh 15vh 10vh;
    }

    h1 {
      grid-column: 3 / 6;
      grid-row: 1 / 2;
      z-index: 2;
      padding-left: 0.25em;
      margin-bottom: initial;
    }

    h2 {
      grid-row: 1 / -1;
      grid-column: 6 / 7;
      writing-mode: vertical-rl;
      margin-bottom: initial;
      color: $text;
    }

    hr {
      grid-column: 5 / 6;
      grid-row: 2;
    }
  }
}
```
Tycho

Tycho is an American ambient music project led by Scott Hansen as primary composer, songwriter and producer. Hailing from San Francisco, California, he is known as ISO50 for his photographic and design works. His music is a combination of downtempo vintage-style synthesizers and ambient melodies.

10.2.2010 TORONTO
THE DRAKE HOTEL

READ MORE
Fallback with feature queries

A conditional group rule whose condition tests whether the user agent supports CSS property: value pairs.
Anatomy of a feature query

```javascript
.selector {
    /* Styles that are supported in old browsers */
}

@supports (property:value) {
    .selector {
        /* Styles for browsers that support the specified property */
    }
}
```
# Browser support for @supports

CSS Feature Queries allow authors to condition rules based on whether particular property declarations are supported in CSS using the @supports at rule.

<table>
<thead>
<tr>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>14</td>
<td>56</td>
<td>62</td>
<td>10</td>
<td>48</td>
<td>10.2</td>
<td>4.4</td>
<td>4.4</td>
<td>4.4</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>57</td>
<td>63</td>
<td>10.1</td>
<td>49</td>
<td>10.3</td>
<td>4.4</td>
<td>4.4</td>
<td>4.4</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>58</td>
<td>64</td>
<td>11</td>
<td>50</td>
<td>11.2</td>
<td>all</td>
<td>62</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>59</td>
<td>65</td>
<td>11.1</td>
<td>51</td>
<td>11.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Global: 93.52% + 0% = 93.52%

Data from caniuse.com | Embed from caniuse.bitsofco.de
header::before {
  content: 'This example requires a browser that supports CSS flexbox or grid. If your browser does not support CSS grid or flexbox, you have seen something that resembles the image below, except recreated in HTML and CSS instead.';
  display: inline-block;
  background-color: rgba(253, 182, 182, 0.75);
  font-size: initial;
  max-width: 36em;
  color: #000;
  padding: 0.5em;
}

header * {
  display: none;
}

header {
  background-color: $background;
  background-image: url('./fallback.png');
  background-size: contain;
  background-repeat: no-repeat;
  background-position: bottom;
  overflow: hidden;
  max-height: 60vh;
  position: relative;
}
@supports (display: flex) {
    header::before {
        content: none;
    }
}

header {
    background-image: none;
}

.grid {
    display: block;
}

.title {
    color: $txt-light;
    display: flex;
    align-items: center;
    justify-content: flex-end;
    font-size: 2.5em;
    z-index: 1;
    padding-right: 0.25em;
}

.time {
    color: $txt-dark;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 0 0.5em;
}
@supports (display: grid) {
  .time {
    grid-row: 7;
  }

  .grid {
    display: grid;
    grid-template-columns: repeat(9, 1fr);
    grid-template-rows: repeat(3, 2em) repeat(3, 0.125em);
    position: relative;
  }

  @media screen and (min-width: 480px) {
    .grid {
      transform: rotate(-30deg) translateX(5vh);
      grid-template-rows: repeat(3, 2em) repeat(3, 0.125em);
    }
  }

  @media screen and (min-width: 640px) {
    .grid {
      transform: rotate(-30deg) translateY(-15vh);
    }
  }

  .grid_item:nth-of-type(1) {
    grid-row: 4;
    grid-column: span 3;
    width: 100%;
    height: 80px;
  }
}
“CSS isn't a programming language. It's a stylesheet language. We shouldn't expect it to behave like a programming language. It has its own unique landscape and structures, ones that people with programming language mental maps might not expect.”

—Danielle Huntrods
The CSS Layout Workshop

Learn CSS layout with straightforward and practical examples. A course designed to remove layout confusion for good!

Does this sound familiar?

I love designing a site, but then find I have to compromise the design when it comes to building it with CSS.

My approach to building layouts is to try things, reload and see what happens.

I use a framework to do my layouts. I’d love to know what it is really doing under the hood.

Once I get my CSS working I’m afraid to make changes in case it all falls apart again!

I’ve never really understood why margins collapse.

https://thecssworkshop.com/

http://labsjensimmons.com/
Demos and examples

- Space Jam website
- Regular grids
- Malerei, Fotografie, Film (pg. 126)
- Braun HiFi print posters
- Simple overlap
- Tycho artist profile
- Fallback with feature queries
- Wismut Labs website
- Demos source code on GitHub
Useful references

- CSS Grid Layout Module Level 1
- Codrops CSS Grid reference
- Grid by Example
- Learn CSS Grid
- Grid Auto-Placement Is Ready
- Automatizing the Grid
- Deep Dive into Grid Layout Placement
- CSS Grid Layout and positioned items
- The Story of CSS Grid, from Its Creators
- CSS Grid Layout is Here to Stay
- The New Layout Standard For The Web: CSS Grid, Flexbox And Box Alignment
THANK YOU!

https://www.chenhuijing.com

@hj_chen

@hj_chen

@huijing