BUILDING LAYOUTS WITH MODERN CSS

#TeamWeb

Original image by the amazing Lin Clark
Surname: Chen
First name: Hui Jing
@hj_chen
https://events.mozilla.org/techspeakers
General agenda

- Introduction and set up
- Pre-grid techniques and concepts
- Flexbox basics
  - Flex syntax
  - Aligning flex items
- Real-world application: Image gallery
- Grid basics
  - Auto-placement
  - Flexible sizing
  - Manual placement
- Real-world application: Responsive dashboard
Evolution of browsers

No layout
HTML Tables
CSS Floats
Frameworks
Grid and beyond
Pre-grid techniques

https://codepen.io/huijing/pen/KGOmLZ/
The **flex** syntax

- Brackets are for grouping
- `<flex-grow>` `<flex-shrink>`? || `<flex-basis>`
- Preceding type/word/group is optional
- Separates 2 options, one or more must occur, order doesn't matter
Flexbox basics

This is a sentence. A rather short sentence.

For the purposes of this example, we need a paragraph. She must have hidden the plans in the escape pod.

https://codepen.io/huijing/pen/NEPwoj/
<table>
<thead>
<tr>
<th>Flex shorthand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>flex: initial</strong></td>
<td><code>flex: 0 1 auto</code>, cannot grow but can shrink when there isn't enough space</td>
</tr>
<tr>
<td><strong>flex: auto</strong></td>
<td><code>flex: 1 1 auto</code>, can grow and shrink to fit available space</td>
</tr>
<tr>
<td><strong>flex: none</strong></td>
<td><code>flex: 0 0 auto</code>, cannot grow or shrink, AKA inflexible</td>
</tr>
<tr>
<td><strong>flex: &lt;positive-number&gt;</strong></td>
<td><code>flex: &lt;positive-number&gt; 1 0</code>, can grow and shrink, extent of growth depends on flex factor</td>
</tr>
</tbody>
</table>
“Authors are encouraged to control flexibility using the flex shorthand rather than with its longhand properties directly, as the shorthand correctly resets any unspecified components to accommodate common uses.”

—CSS Flexible Box Layout Module Level 1
Flex lines

nowrap

wrap

wrap-reverse
Auto-margins with Flexbox

https://codepen.io/huijing/pen/EOWedx/
Aligning flex items

https://codepen.io/huijing/pen/zMNqOo/
Image gallery

https://codepen.io/huijing/pen/dQPZmL/
Defining a grid

https://codepen.io/huijing/pen/mzKoNj/
The `repeat()` function

https://codepen.io/huijing/pen/XxveJe/
grid-auto-row and grid-auto-column

https://codepen.io/huijing/pen/ePqyMz/
auto-fill versus auto-fit

https://codepen.io/huijing/pen/GYVyMX/
The grid-auto-flow property

[Link to CodePen example](https://codepen.io/huijing/pen/LgwegQ/)
The fr unit

https://codepen.io/huijing/pen/WaVdLN/
The `minmax()` function

https://codepen.io/huijing/pen/jegYob/
Content-based sizing

- If you don't have any shadows, you're not in the light.
- You laugh at me because I'm different, I laugh at you because you're all the same.
- Don't judge anyone but don't trust everyone.

https://codepen.io/huijing/pen/bmXLPd/
Image gallery (part 2)

https://codepen.io/huijing/pen/dQPZmL/
Line-based placement

https://codepen.io/huijing/pen/pxMaYp/
grid-column and grid-row

https://codepen.io/huijing/pen/zmgRgM/
The span keyword

https://codepen.io/huijing/pen/NOQYMM/
Using the grid-area shorthand

https://codepen.io/huijing/pen/vVoRzx/
Naming grid lines

https://codepen.io/huijing/pen/xyvWBZ/
Naming grid areas

https://codepen.io/huijing/pen/EdqLxP/
Responsive dashboard

Some dashboard

Some statistics
* rating: 65
* rating: 100
* rating: 75

I dunno...use your imagination

Okay, fine...this is supposed to be a game board so let's centre the content of the board. This is sort of a kludge, actually. Also, let's vertically centre the title, score and statistics. Then centre the controls both ways.
Do we have enough time?
Some dashboard

Score: 100

Some statistics
- rating: 65
- rating: 100
- rating: 75

I dunno...use your imagination

Okay, fine...this is supposed to be a game board so let's centre the content of the board. This is sort of a trick question, actually...

Also, let's vertically centralise the title, score and statistics. Then centre the controls both ways.

Up Up Down Down Left Right Left Right B A

https://codepen.io/huijing/pen/xQbXKO/
Feature queries

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

@supports (property:value) {
    .selector {
        /* Styles for browsers that support the specified property */
    }
}
Card-based layout

https://codepen.io/huijing/pen/VVPjBL/
Useful references

- What Happens When You Create A Flexbox Flex Container?
- Everything You Need To Know About Alignment In Flexbox
- Use Cases For Flexbox
- 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
- CSS Logical Properties and Values in Chromium and WebKit
- Changes on CSS Grid Layout in percentages and indefinite height
- 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
- Grid “fallbacks” and overrides
- Cascading Web Design with Feature Queries
- Basic grid layout with fallbacks using feature queries
Thank you!

https://www.chenhuijing.com

@hj_chen

@huijing

Font used is ABeezee, by Anja Meiners