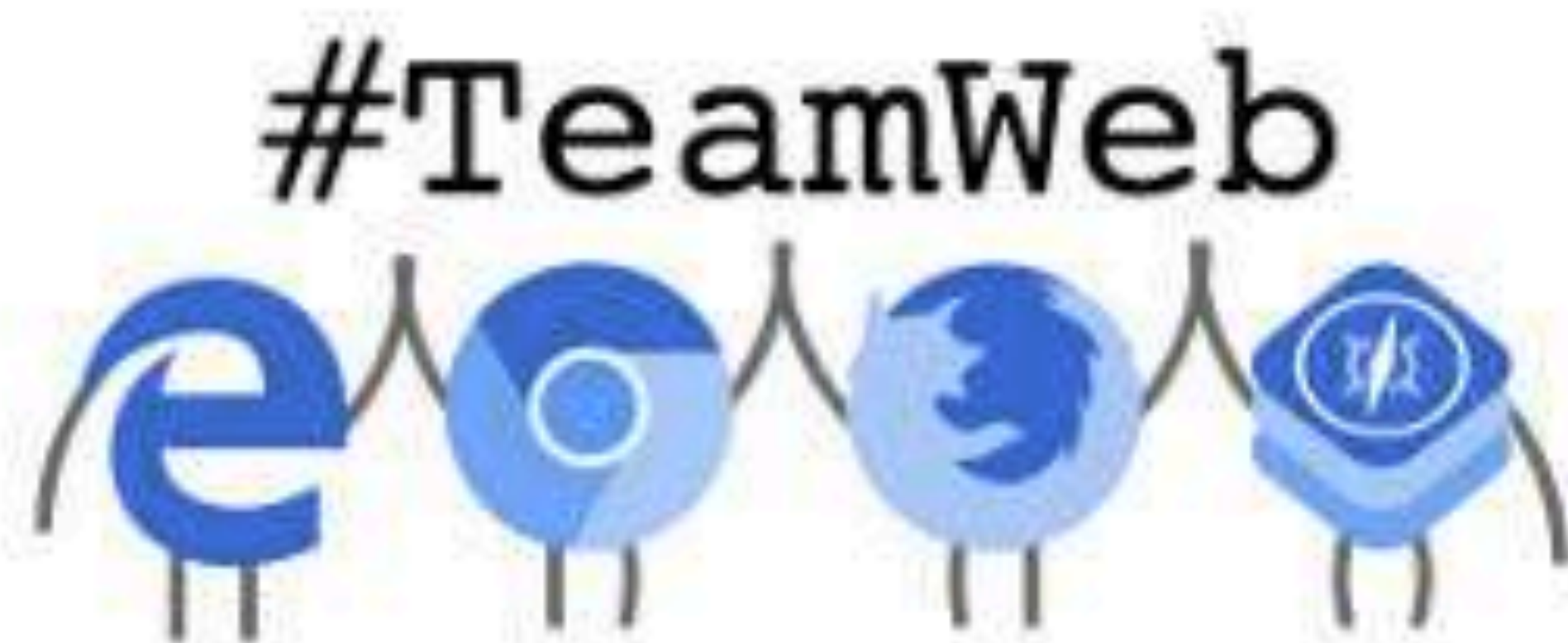




# LAYOUT THE WEB WITH CSS GRID

and the rest of team layout



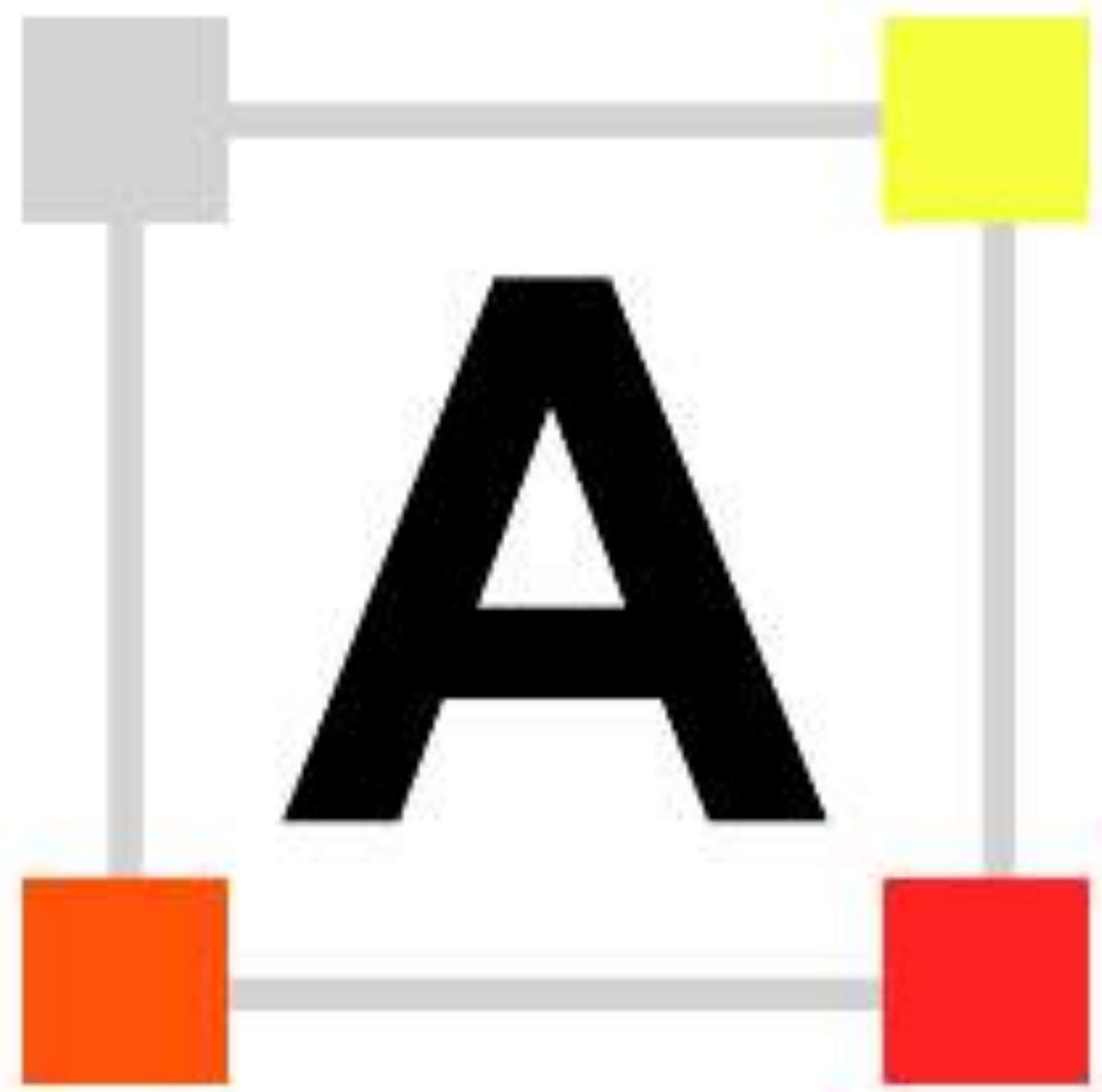
Original image by the amazing [Lin Clark](#)



# General agenda

- Introduction and set up
- Pre-grid techniques and concepts
- Flexbox basics
- Real-world application: Image gallery (part 1)
- Real-world application: Configuration page
- Grid basics
  - Defining a grid
  - Auto-placement
  - Flexible sizing
  - Manual placement
- Real-world application: Image gallery (part 2)
- Real-world application: Responsive dashboard
- Feature queries
- Real-world application: Artist profile page







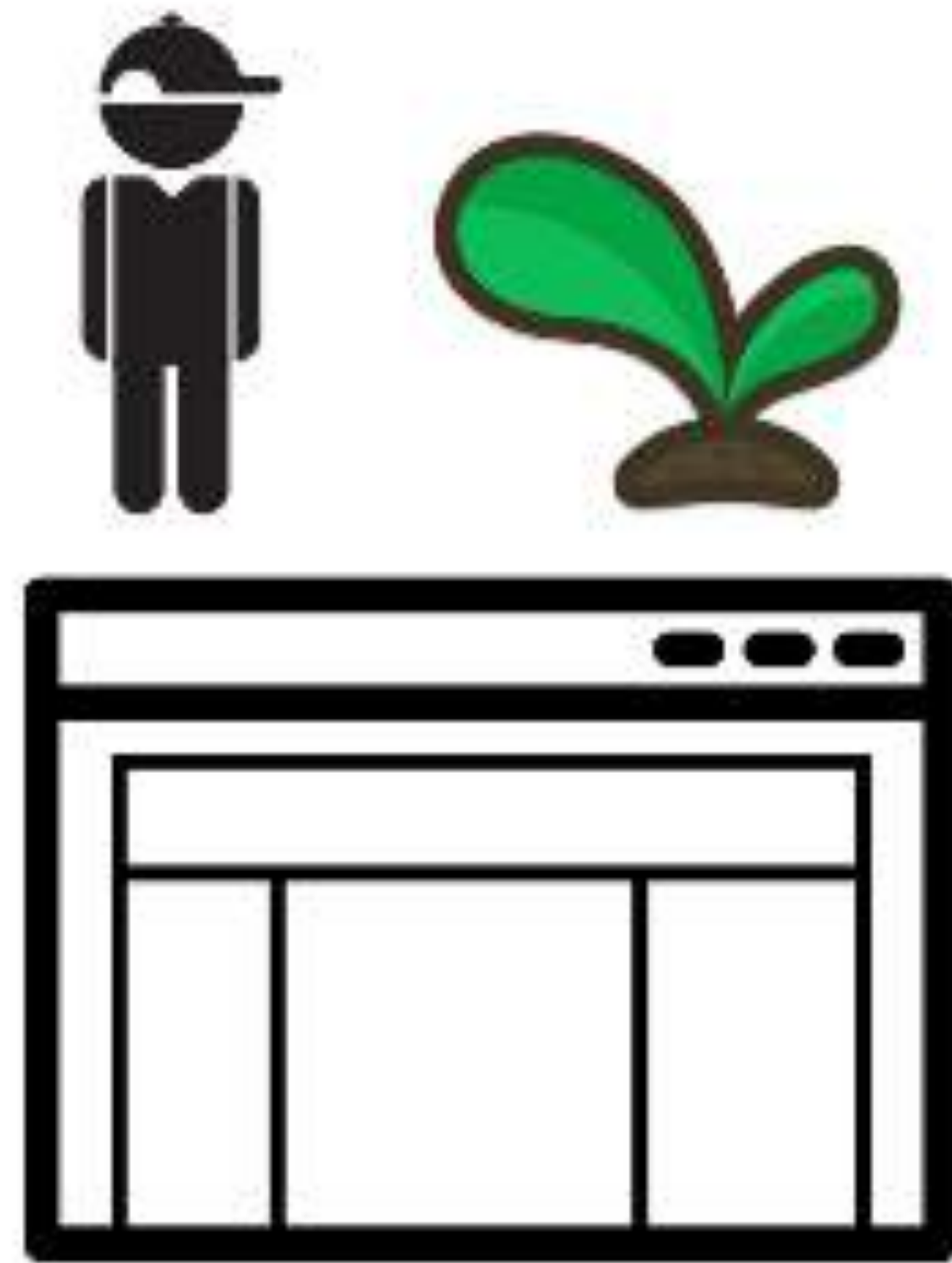
# Evolution of browsers



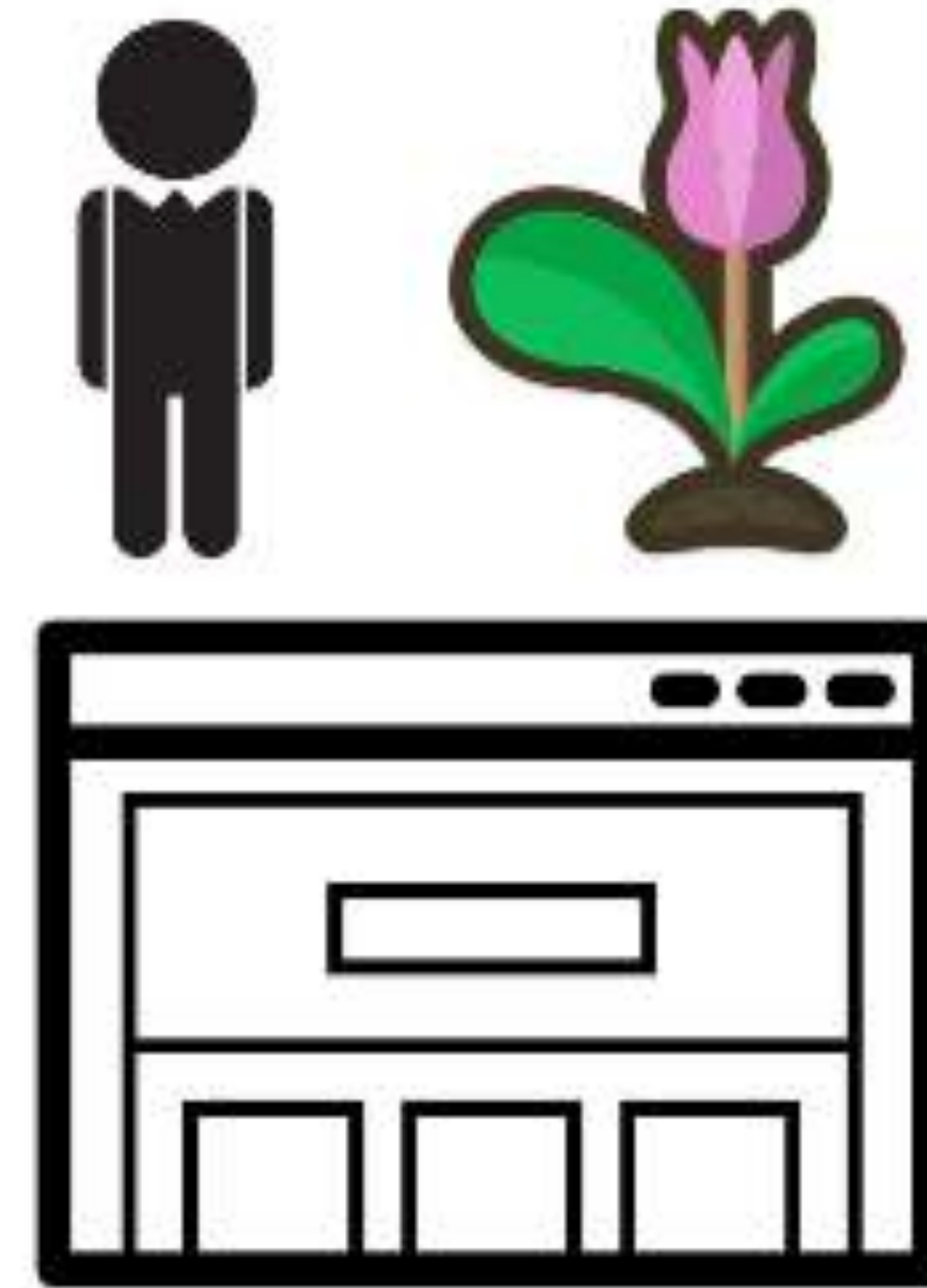
No layout



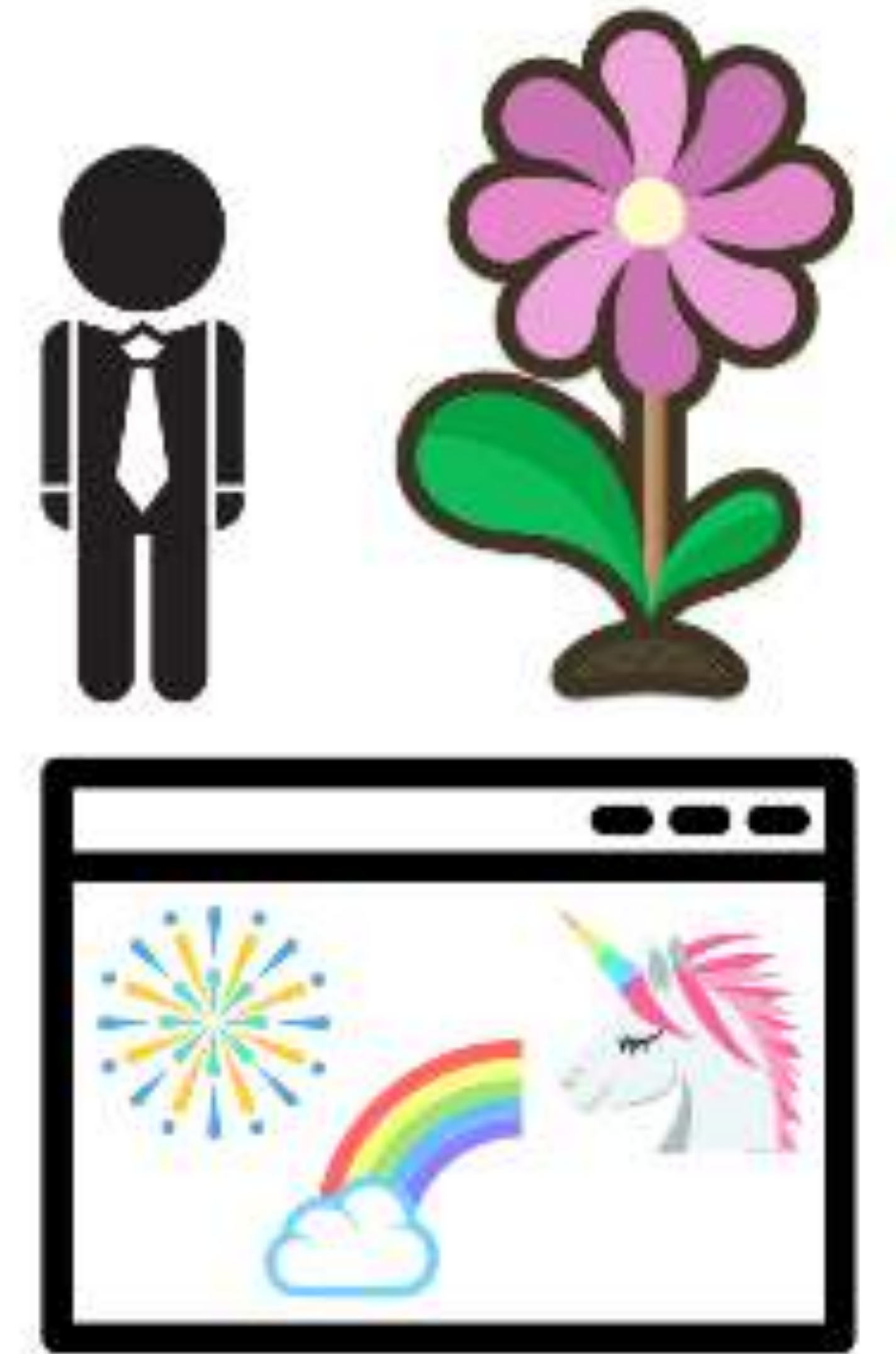
HTML Tables



CSS Floats



Frameworks



Grid and beyond



# Pre-grid techniques

HTML Result EDIT ON CODEPEN

```
<h2>Inline-block</h2>

<div class="inlineblock">
  <div class="inlineblock_item"><p>Item A</p></div><div class="inlineblock_item"><p>Item B</p></div><div
class="inlineblock_item"><p>Item C</p></div><div class="inlineblock_item"><p>Item D</p></div><div
class="inlineblock_item"><p>Item E</p></div><div class="inlineblock_item"><p>Item F</p></div>
</div>

</div>

<h2>Floats</h2>

<div class="float">
  <div class="float_item">
    <p>Item A</p>
  </div>
  <div class="float_item">
    <p>Item B</p>
  </div>
  <div class="float_item">
    <p>Item C</p>
  </div>
  <div class="float_item">
    <p>Item D</p>
  </div>
  <div class="float_item">
    <p>Item E</p>
  </div>
  <div class="float_item">
    <p>Item F</p>
  </div>
</div>
```

## Inline-block

Item A
Item B
Item C
Item D
Item E
Item F

## Floats

Item A
Item B
Item C

<https://codepen.io/huijing/pen/KG0mLZ/>



# Flexbox basics

The screenshot shows a CodePen editor with two tabs: 'HTML' and 'CSS'. The 'HTML' tab is active, displaying the following code:

```
<div class="wrapper">
  <div class="box">1一</div>
  <div class="box">2二</div>
  <div class="box">3三</div>
  <div class="box">4四</div>
  <div class="box">5五</div>
  <div class="box">6六</div>
  <div class="box">7七</div>
  <div class="box">8八</div>
  <div class="box">9九</div>
  <div class="box">10十</div>
  <div class="box">11十一</div>
  <div class="box">12十二</div>
  <div class="box">13十三</div>
  <div class="box">14十四</div>
  <div class="box">15十五</div>
  <div class="box">16十六</div>
  <div class="box">17十七</div>
  <div class="box">18十八</div>
  <div class="box">19十九</div>
  <div class="box">20二十</div>
</div>
```

The 'Result' tab shows the rendered output, which is a vertical list of 20 items, each consisting of a number followed by its corresponding Chinese numeral character (e.g., '1一', '2二', etc.). The items are arranged in a single column, demonstrating a flexbox layout with a vertical direction.


<https://codepen.io/huijing/pen/NEPwoj/>



# Image gallery

HTML SCSS Result EDIT ON CODEPEN

```
<ul>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>  
<li></li>
```



<https://codepen.io/huijing/pen/dQPZmL/>



# Configuration page

HTML SCSS Result EDIT ON CODEPEN

```
<header>
  <h1>Configuration</h1>
</header>



<main>
  <div class="config-toggles">
    <h2>Toggle mood</h2>
    <input id="happy" type="radio" name="mood" checked>
    <label for="happy" class="config-toggle">
      <span class="emoji" role="img" aria-label="grinning face with smiling eyes">&#x1F601;</span>
    <span>Happy</span>
    </label>
    <input id="angry" type="radio" name="mood">
    <label for="angry" class="config-toggle">
      <span class="emoji" role="img" aria-label="pouting face">&#x1F621;</span> <span>Angry</span>
    </label>
  </div>

  <div class="config-toggles">
    <h2>Adjust weather</h2>
    <input id="sunny" type="radio" name="weather" checked>
    <label for="sunny" class="config-toggle">
      <span class="emoji" role="img" tabindex="0" aria-label="sun">&#x2600;&#xFE0F;</span>
    <span>Sunny</span>
    </label>
    <input id="stormy" type="radio" name="weather">
    <label for="stormy" class="config-toggle">
      <span class="emoji" role="img" tabindex="0" aria-label="cloud with lightning and
rain">&#x26C8;&#xFE0F;</span> <span>Stormy</span>
    </label>
  </div>



  <button type="button">Done</button>
</main>
```

## Configuration

### Toggle mood

 Happy   Angry

### Adjust weather

 Sunny   Stormy

<https://codepen.io/huijing/pen/GwgMJx/>



# Defining a grid

HTML SCSS Result EDIT ON CODEPEN

```
<h2>Defining a grid</h2>
<div class="grid">
  <div class="grid_item">
    <p>Item A</p>
  </div>
  <div class="grid_item">
    <p>Item B</p>
  </div>
  <div class="grid_item">
    <p>Item C</p>
  </div>
  <div class="grid_item">
    <p>Item D</p>
  </div>
  <div class="grid_item">
    <p>Item E</p>
  </div>
  <div class="grid_item">
    <p>Item F</p>
  </div>
</div>
```

Defining a grid
Item A
Item B
Item C
Item D
Item E
Item F

<https://codepen.io/huijing/pen/mzKoNj/>



# The repeat() function

HTML SCSS Result EDIT ON CODEPEN

```
<h2>The <code>repeat()</code> function</h2>
<div class="grid">
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
  <div class="grid_item">
    <p>Item</p>
  </div>
</div>
```

The repeat ( ) function

Item
Item
Item
Item
Item
Item
Item
Item
Item
Item

<https://codepen.io/huijing/pen/XxveJe/>



# auto-fill versus auto-fit

HTML SCSS Result EDIT ON CODEPEN

```
<h2><code>auto-fill</code> versus <code>auto-fit</code></h2>

<div class="grid">
  <div class="grid_item">
    <p>Item A</p>
  </div>
  <div class="grid_item">
    <p>Item B</p>
  </div>
  <div class="grid_item">
    <p>Item C</p>
  </div>
  <div class="grid_item">
    <p>Item D</p>
  </div>
  <div class="grid_item">
    <p>Item E</p>
  </div>
  <div class="grid_item">
    <p>Item F</p>
  </div>
</div>
```

auto-fill versus auto-fit

Item A
Item B
Item C
Item D
Item E
Item F

<https://codepen.io/huijing/pen/GYVyMX/>



# grid-auto-row and grid-auto-column

HTML SCSS Result EDIT ON CODEPEN

```
<h2><code>grid-auto-row</code> and <code>grid-auto-column</code></h2>

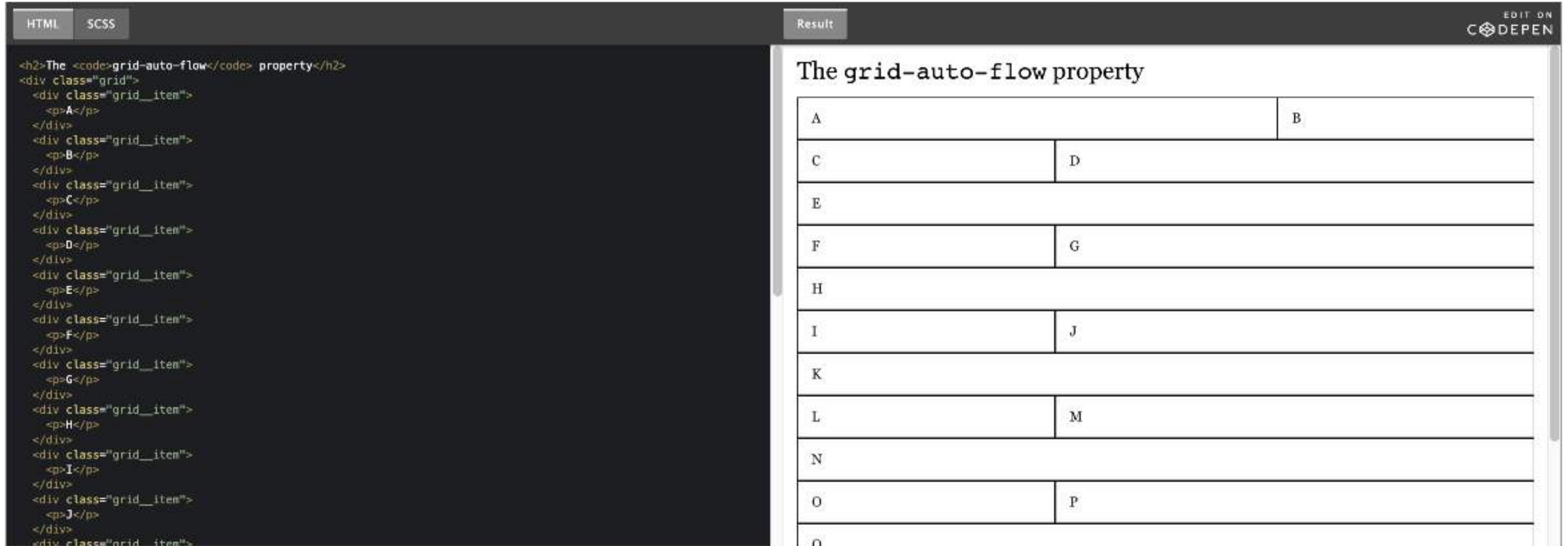
<div class="grid">
  <div class="grid_item">
    <p>A</p>
    <p>A</p>
  </div>
  <div class="grid_item">
    <p>B</p>
  </div>
  <div class="grid_item">
    <p>C</p>
  </div>
  <div class="grid_item">
    <p>D</p>
  </div>
  <div class="grid_item">
    <p>E</p>
  </div>
  <div class="grid_item">
    <p>F</p>
  </div>
  <div class="grid_item">
    <p>G</p>
  </div>
  <div class="grid_item">
    <p>H</p>
  </div>
  <div class="grid_item">
    <p>I</p>
  </div>
  <div class="grid_item">
    <p>J</p>
  </div>
</div>
```

A A
B
C
D
E
F
G
H
I
J

<https://codepen.io/huijing/pen/ePqyMz/>



# The grid-auto-flow property



The screenshot shows a CodePen editor with the following HTML code in the left pane:

```
<h2>The grid-auto-flow property</h2>
<div class="grid">
  <div class="grid_item">
    <p>A</p>
  </div>
  <div class="grid_item">
    <p>B</p>
  </div>
  <div class="grid_item">
    <p>C</p>
  </div>
  <div class="grid_item">
    <p>D</p>
  </div>
  <div class="grid_item">
    <p>E</p>
  </div>
  <div class="grid_item">
    <p>F</p>
  </div>
  <div class="grid_item">
    <p>G</p>
  </div>
  <div class="grid_item">
    <p>H</p>
  </div>
  <div class="grid_item">
    <p>I</p>
  </div>
  <div class="grid_item">
    <p>J</p>
  </div>
  <div class="grid_item">
    <p>K</p>
  </div>
  <div class="grid_item">
    <p>L</p>
  </div>
  <div class="grid_item">
    <p>M</p>
  </div>
  <div class="grid_item">
    <p>N</p>
  </div>
  <div class="grid_item">
    <p>O</p>
  </div>
  <div class="grid_item">
    <p>P</p>
  </div>
  <div class="grid_item">
    <p>Q</p>
  </div>
</div>
```

The right pane, titled "Result", shows the rendered grid. The grid consists of 13 rows and 2 columns. The items are arranged as follows:

A	B
C	D
E	
F	G
H	
I	J
K	
L	M
N	
O	P
Q	

<https://codepen.io/huijing/pen/LgwegQ/>



# The fr unit

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
}

.grid_item {
}

// General styles
body {
  padding: 1em;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}

code {
  font-family: monospace;
  font-size: 1.25em;
}

[class$="item"] {
  border: 1px solid;
  padding: 1em;
}
```

## The fr unit

Item A
Item B
Item C

<https://codepen.io/huijing/pen/WaVdLN/>



# The minmax() function

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
}

.grid_item {
}

// General styles
body {
  padding: 1em;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}

code {
  font-family: monospace;
  font-size: 1.25em;
}

[class$="item"] {
  border: 1px solid;
  padding: 1em;
}
```

The minmax( ) function

Item A
Item B
Item C

<https://codepen.io/huijing/pen/jegYob/>



# Content-based sizing

HTML SCSS Result EDIT ON CODEPEN

```
<h2>Content-based sizing</h2>
<div class="grid">
  <div class="grid_item">
    <p>If you don't have any shadows, you're not in the light.</p>
  </div>
  <div class="grid_item">
    <p>You laugh at me because I'm different, I laugh at you because you're all the same.</p>
  </div>
  <div class="grid_item">
    <p>Don't judge anyone but don't trust everyone.</p>
  </div>
</div>
```

## 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)

HTML SCSS Result EDIT ON CODEPEN

```
<ul>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>

```



<https://codepen.io/huijing/pen/dQPZmL/>



# Line-based placement

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, calc(100% / 3));
  grid-template-rows: repeat(3, calc(100% / 3));
}

.grid_item {
}

img {
  max-width: 100%;
  max-height: 100%;
}


// General styles
html {
  height: 100%;
}

body {
  padding: 1em;
  height: 100%;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}
```

Line-based placement



<https://codepen.io/huijing/pen/pxMaYp/>



# grid-column and grid-row

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
  grid-template-columns: repeat(6, calc(100% / 6));
  grid-template-rows: repeat(6, calc(90vh / 6));
}

.grid_item:nth-child(1) {
}

.grid_item:nth-child(2) {
}

.grid_item:nth-child(3) {
}







.grid_item:nth-child(4) {
}

.grid_item:nth-child(5) {
}

.grid_item:nth-child(6) {
}

// General styles
html {
  height: 100%;
}
```

grid-column and grid-row

					
---	---	---	---	---	---

<https://codepen.io/huijing/pen/zmgRgM/>



# The span keyword

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
  grid-template-columns: repeat(5, 20%);
  grid-template-rows: repeat(5, 20vh);
}

.grid_item {
}

img {
  max-width: 100%;
  max-height: 100%;
}

// General styles
html {
  height: 100%;
}

body {
  padding: 1em;
  height: 100%;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}
```

The span keyword

<https://codepen.io/huijing/pen/NOQYMM/>



# Using the grid-area shorthand

HTML SCSS Result EDIT ON CODEPEN

```
.grid {
  display: grid;
  grid-template-columns: repeat(5, 20%);
  grid-template-rows: repeat(3, 30vh);
}

.grid_item {
}

img {
  max-width: 100%;
  max-height: 100%;
}


// General styles
html {
  height: 100%;
}

body {
  padding: 1em;
  height: 100%;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}
```

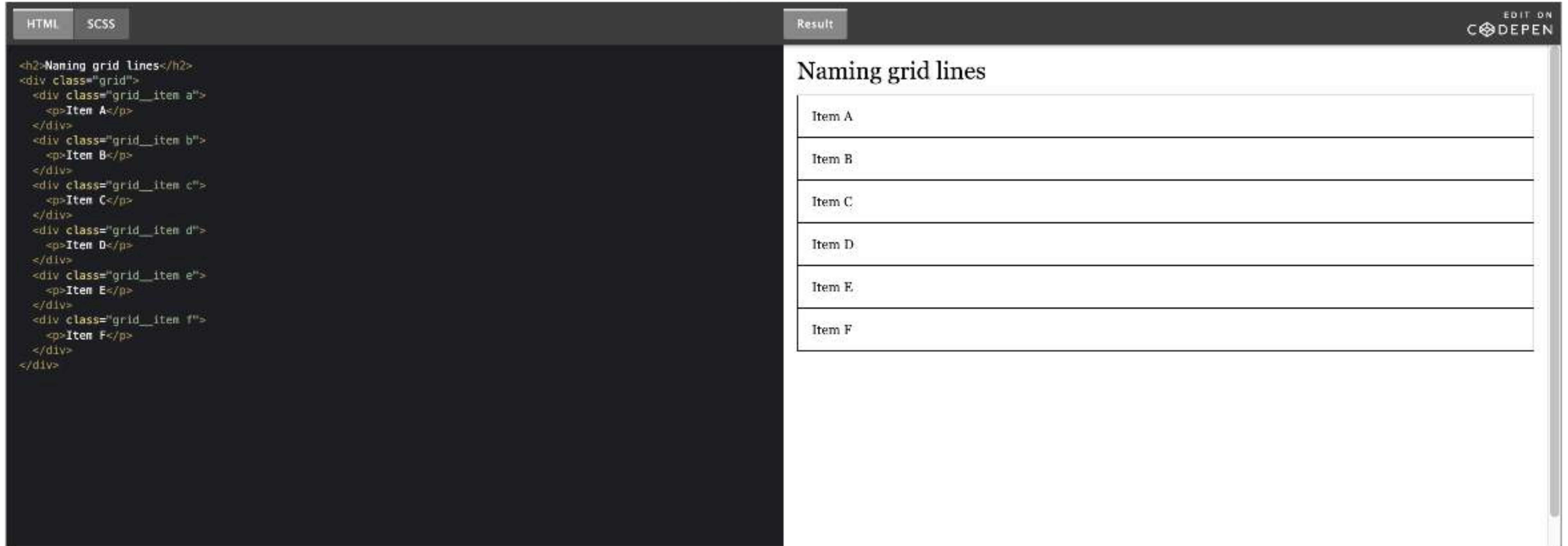
Using the grid-area shorthand



<https://codepen.io/huijing/pen/vVoRzx/>



# Naming grid lines



The image shows a CodePen editor interface. On the left, the HTML code is displayed in a dark theme. On the right, the rendered result is shown in a light theme. The rendered result is a vertical stack of six horizontal lines, each containing the text 'Item A' through 'Item F' respectively.

```
<h2>Naming grid lines</h2>
<div class="grid">
  <div class="grid_item a">
    <p>Item A</p>
  </div>
  <div class="grid_item b">
    <p>Item B</p>
  </div>
  <div class="grid_item c">
    <p>Item C</p>
  </div>
  <div class="grid_item d">
    <p>Item D</p>
  </div>
  <div class="grid_item e">
    <p>Item E</p>
  </div>
  <div class="grid_item f">
    <p>Item F</p>
  </div>
</div>
```

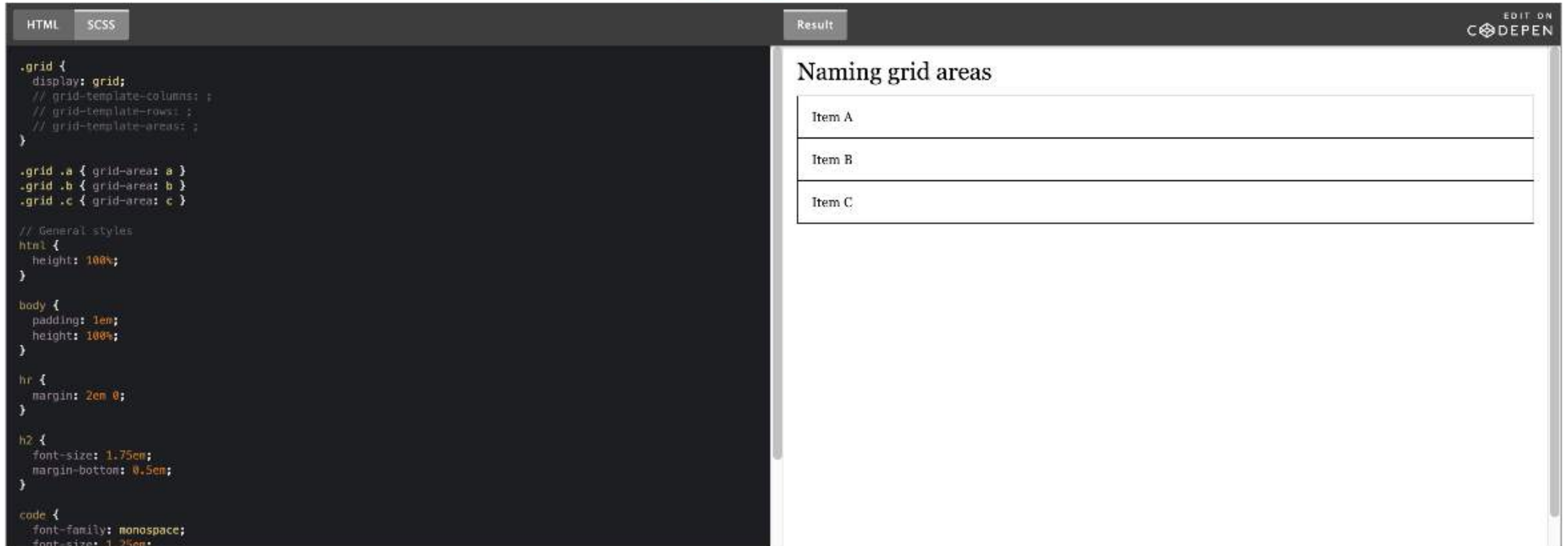
Naming grid lines

Item A
Item B
Item C
Item D
Item E
Item F

<https://codepen.io/huijing/pen/xyvWBZ/>



# Naming grid areas



The image shows a CodePen editor with two tabs: 'HTML' and 'SCSS'. The 'SCSS' tab is active, displaying the following code:

```
.grid {
  display: grid;
  // grid-template-columns: ;
  // grid-template-rows: ;
  // grid-template-areas: ;
}

.grid .a { grid-area: a }
.grid .b { grid-area: b }
.grid .c { grid-area: c }

// General styles
html {
  height: 100%;
}

body {
  padding: 1em;
  height: 100%;
}

hr {
  margin: 2em 0;
}

h2 {
  font-size: 1.75em;
  margin-bottom: 0.5em;
}

code {
  font-family: monospace;
  font-size: 1.25em;
}
```

The 'Result' tab shows the rendered output. It features a heading 'Naming grid areas' followed by three horizontal rectangular boxes, each containing the text 'Item A', 'Item B', and 'Item C' respectively, demonstrating a vertical grid layout.

<https://codepen.io/huijing/pen/EdqLxP/>



# Responsive dashboard

HTML SCSS Result EDIT ON CODEPEN

```
<div class="title">
  <h1>Some dashboard</h1>
</div>

<div class="score">
  <p>Score: 100</p>
</div>

<div class="stats">
  <h2>Some statistics</h2>
  <p>👍 rating: 65</p>
  <p>👎 rating: 100</p>
  <p>👤 rating: 75</p>
</div>

<div class="board">
  <h2 class="board_title">I dunno...use your imagination</h2>
  <div class="board_text">
    <p>Okay, fine...this is supposed to be a game board so let's centre the content of the board.</p>
    <p>This is sort of a trick question, actually...🤔</p>
    <p>Also, let's vertically centralise the title, score and statistics. Then centre the controls both ways.</p>
  </div>
</div>

<div class="controls">
  <p><span>Up</span> <span>Up</span> <span>Down</span> <span>Down</span> <span>Left</span> <span>Right</span> <span>Left</span> <span>Right</span> <span>B</span> <span>A</span></p>
</div>
```

## 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/xQbXK0/>



# Feature queries

```
.selector {  
  /* Styles that are supported in old browsers */  
}  
  
@supports (property:value) {  
  .selector {  
    /* Styles for browsers that support the specified property */  
  }  
}
```



# Musician profile page

HTML SCSS JS Result EDIT ON CODEPEN

```
// // Settings
@font-face {
  font-family: 'Galatia SIL';
  src: url('https://www.chenhuijing.com/demos/grids-tycho/fonts/galsilb.eot');
  src: url('https://www.chenhuijing.com/demos/grids-tycho/fonts/galsilb.eot?#iefix') format('embedded-opentype'),
  url('https://www.chenhuijing.com/demos/grids-tycho/fonts/galsilb.woff2') format('woff2'),
  url('https://www.chenhuijing.com/demos/grids-tycho/fonts/galsilb.woff') format('woff');
  font-weight: normal;
  font-style: normal;
}

$header-font: 'Galatia SIL', Georgia, serif;
$body-font: 'Avenir Book', 'Franklin Gothic Medium', Helvetica, Arial, sans-serif;


$main: #fff;
$accent: #fe320f;
$text: #282828;

// General styles
html {
  box-sizing: border-box;
  height: 100%;
}

*,
*::before,
*::after {
  box-sizing: inherit;
  margin: 0;
  padding: 0;
}
```

**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.



<https://codepen.io/huijing/pen/bQNoe0/>



# 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](#)
- [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](#)
- [What Happens When You Create A Flexbox Flex Container?](#)
- [Everything You Need To Know About Alignment In Flexbox](#)
- [Use Cases For Flexbox](#)
- [Grid “fallbacks” and overrides](#)



# Thank you!



<https://www.chenhuijing.com>



@hj\_chen



@hj\_chen



@huijing

Font used is [Mission Gothic](#), by James T. Edmondson and Trevor Baum