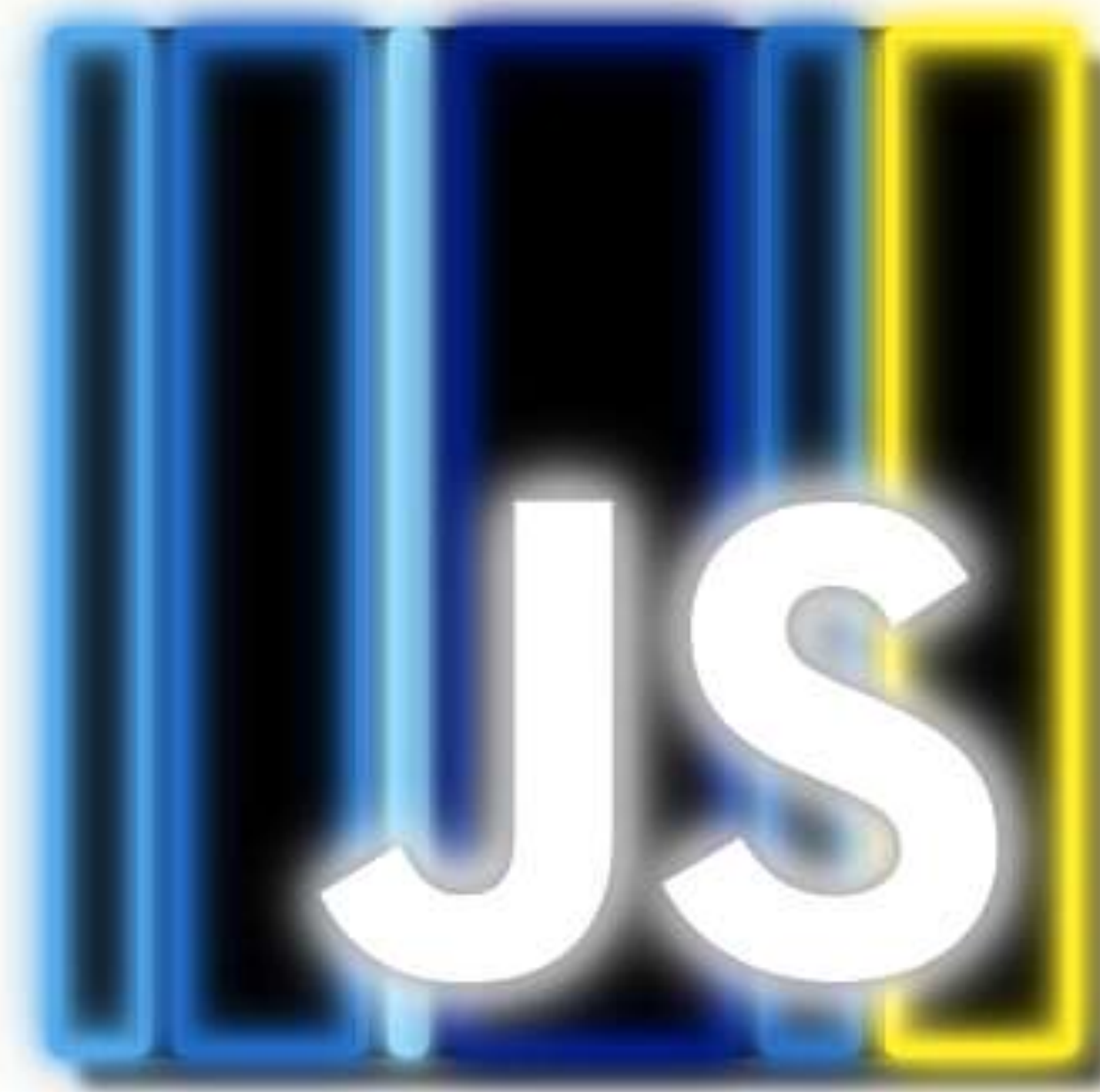


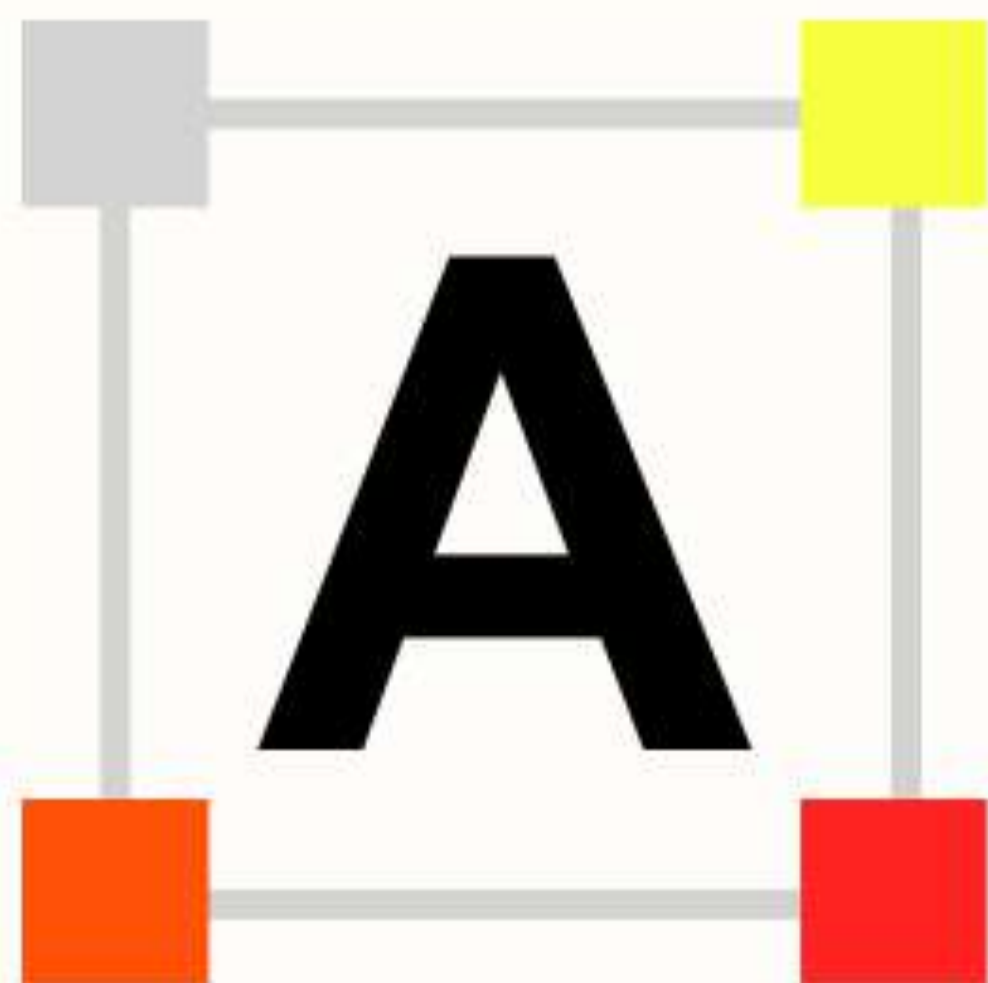
GRID IS THE NEW BLACK

REVOLUTIONISING VISUAL DESIGN ON THE WEB

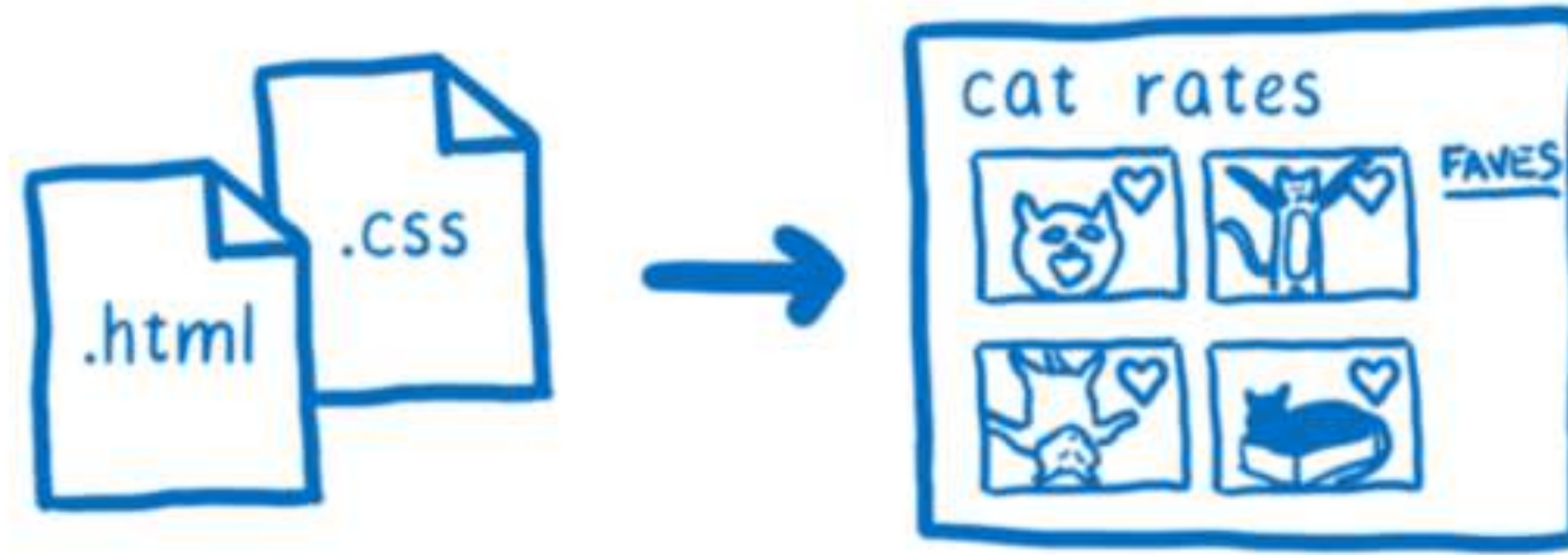


Starter files

<https://www.chenhuijing.com/slides/grid-workshop/files/grid-workshop-files.zip>



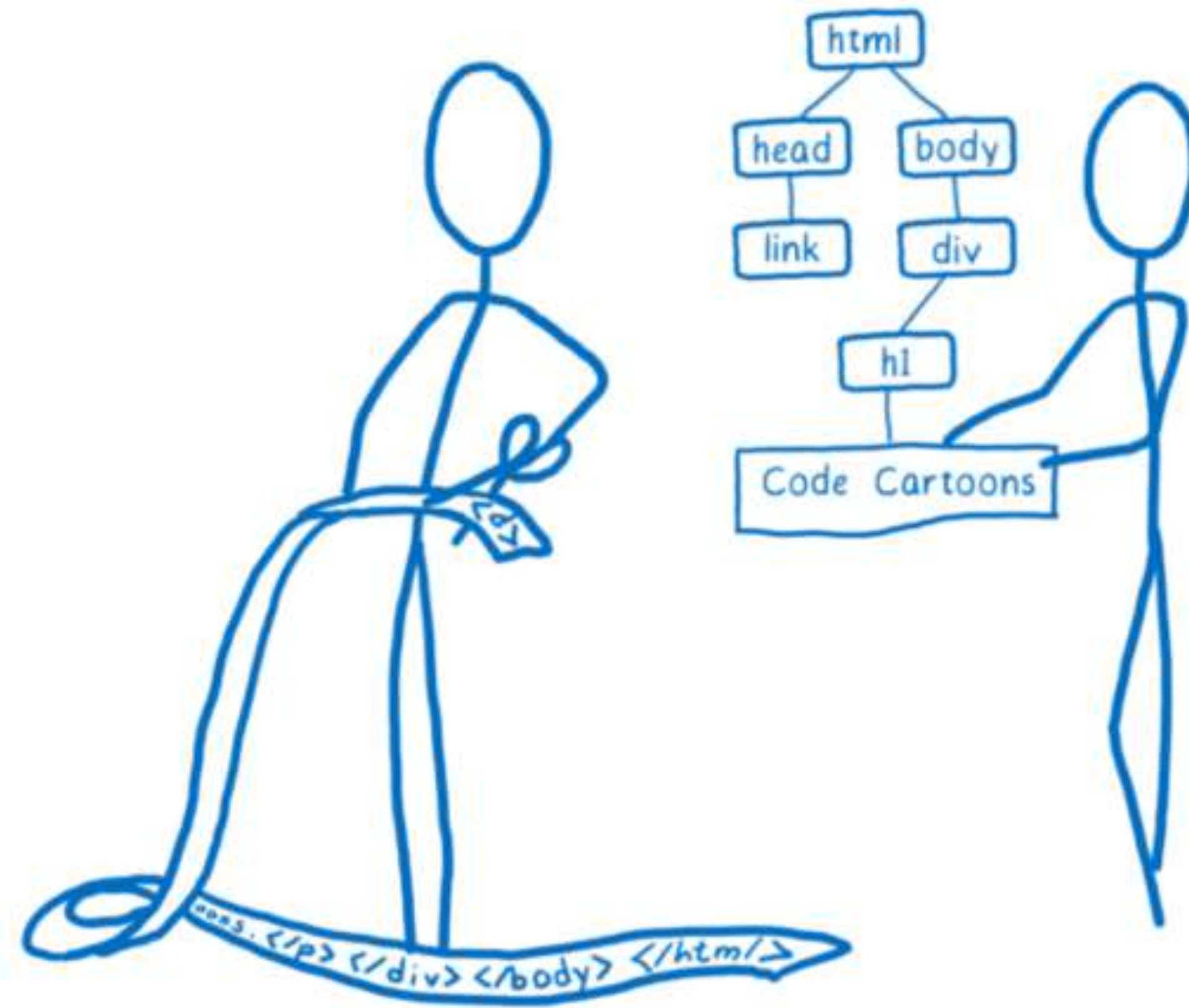
Rendering engines 101



Reads HTML and CSS files and turns them into pixels on the screen

Images and explanations by [Lin Clark](#)

PARSE



Parse the files into objects the browser can understand, including the DOM



STYLE

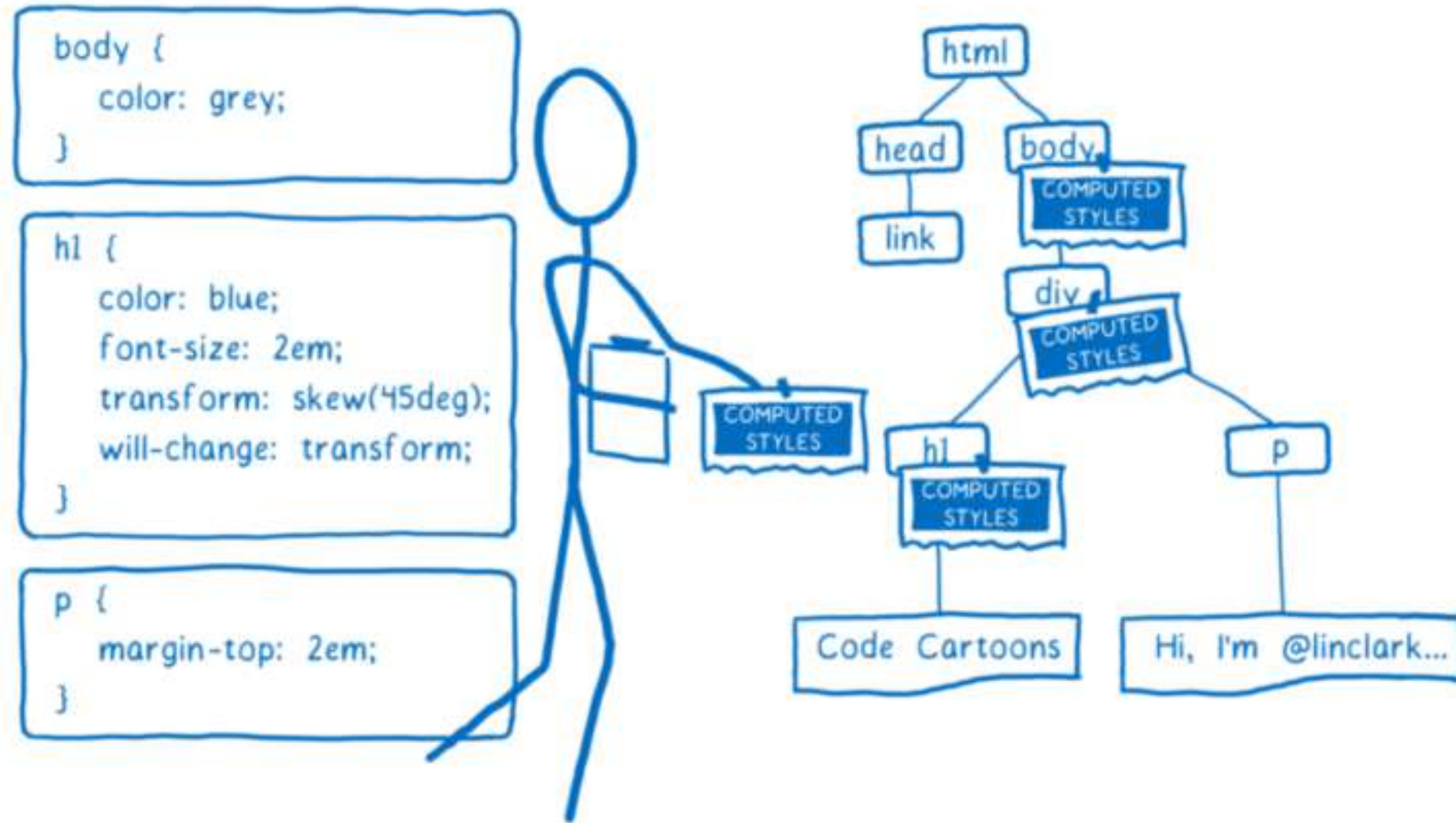


Figure out what the elements should look like



LAYOUT

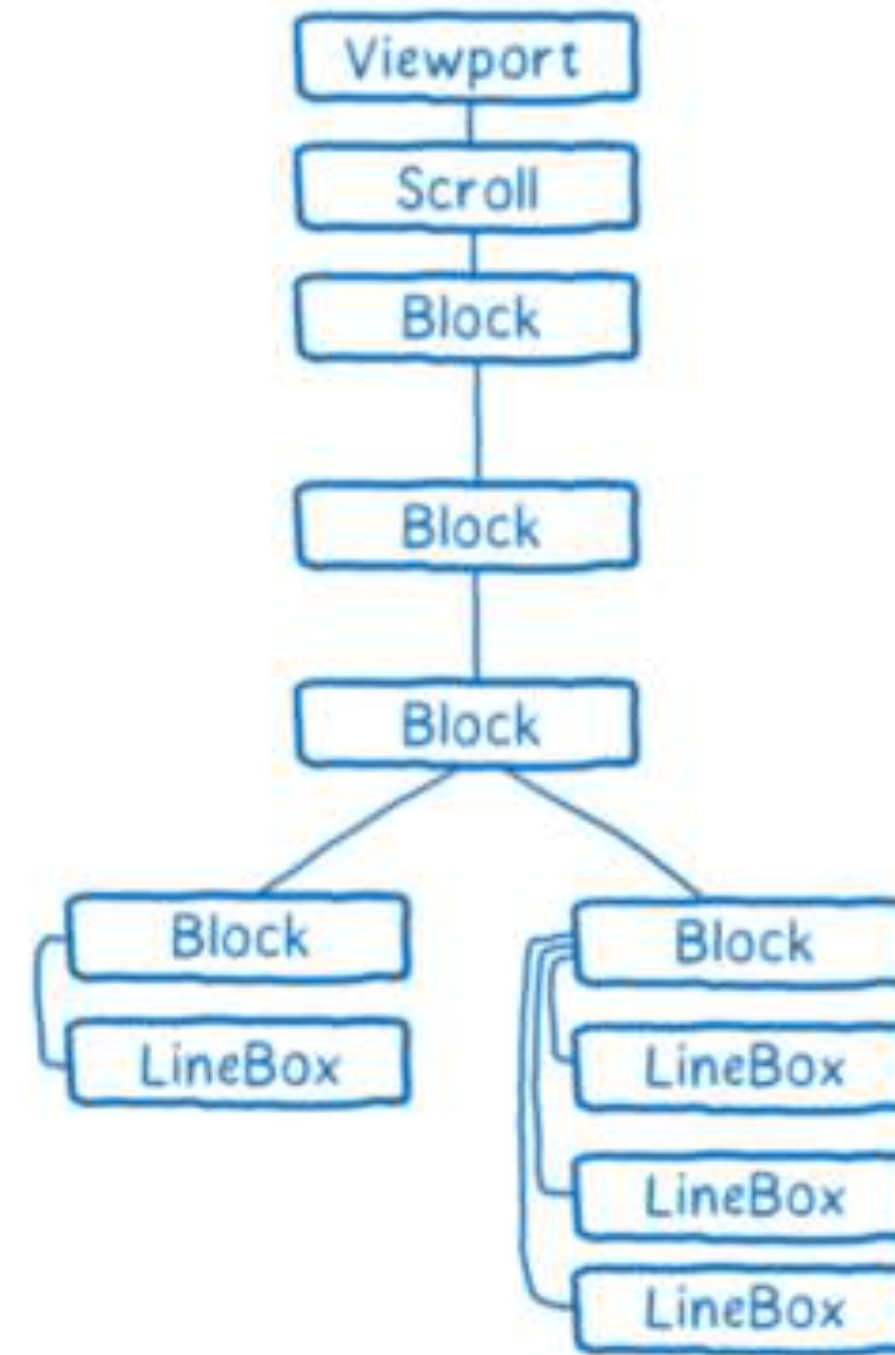
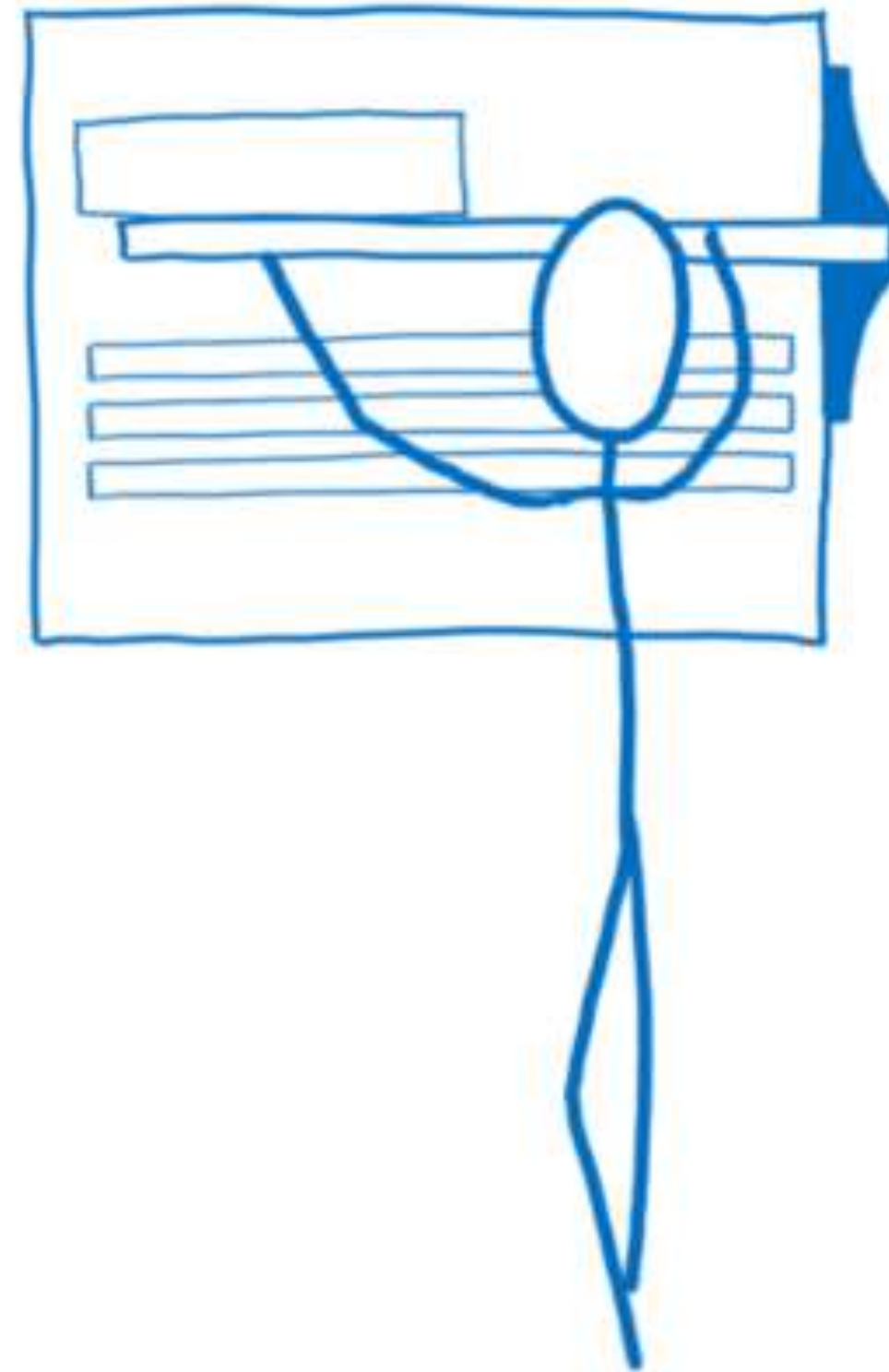
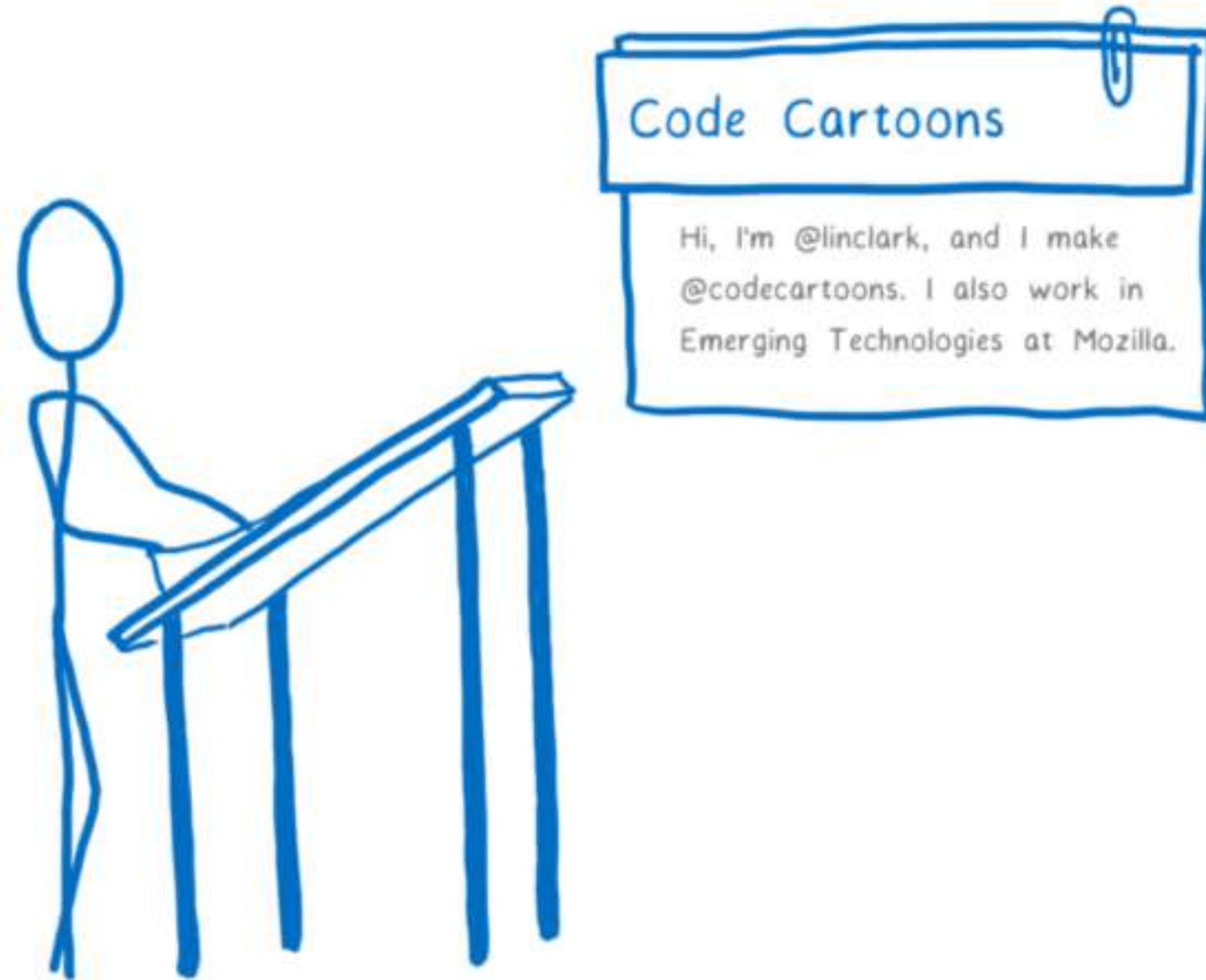


Figure out dimensions for each node and where it goes on the screen



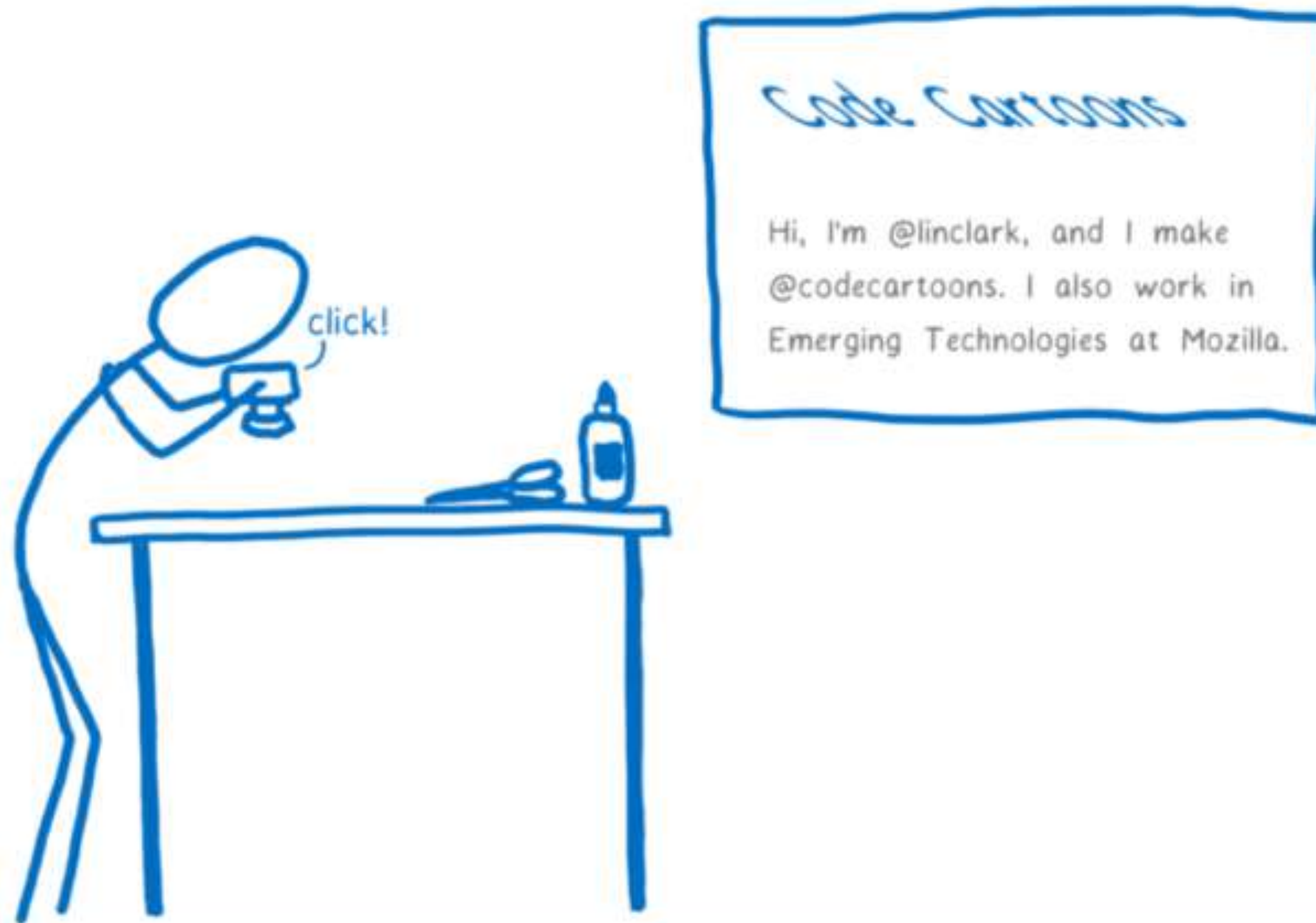
PAINT



Paint the different boxes



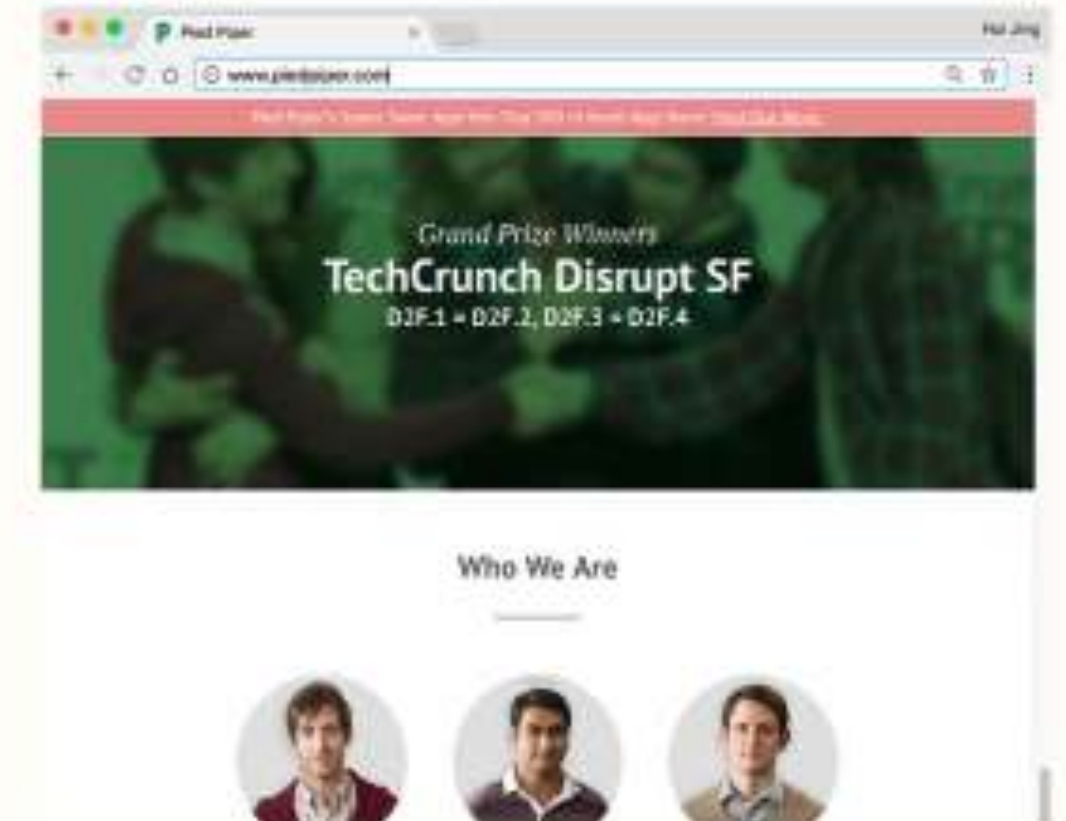
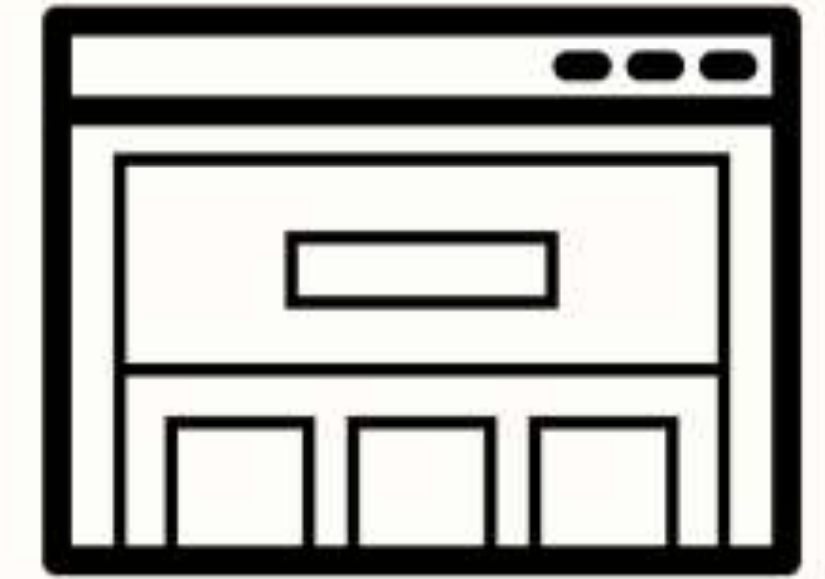
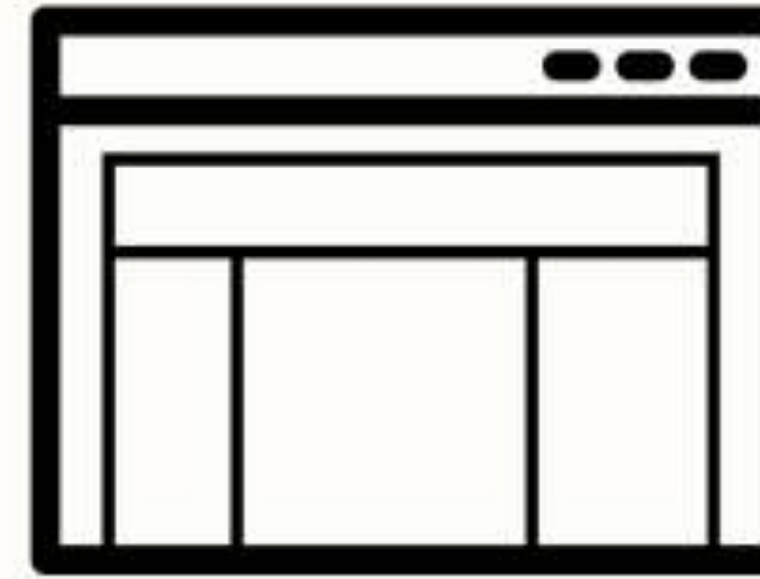
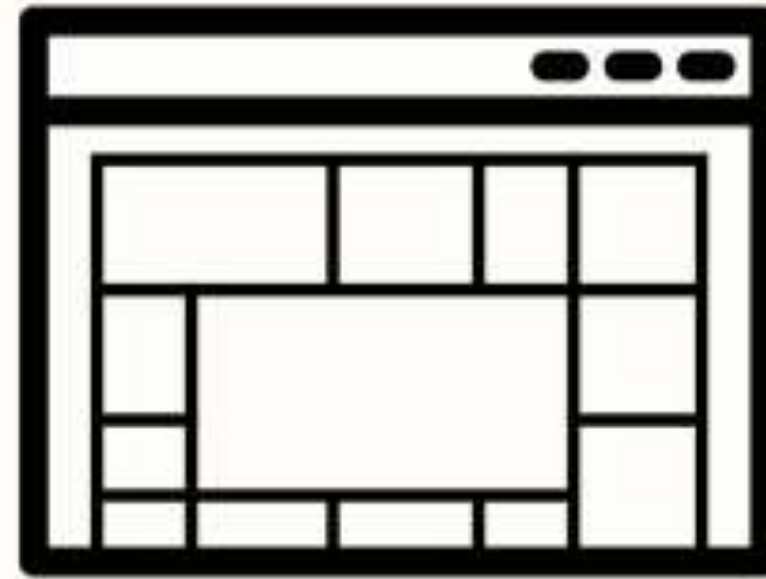
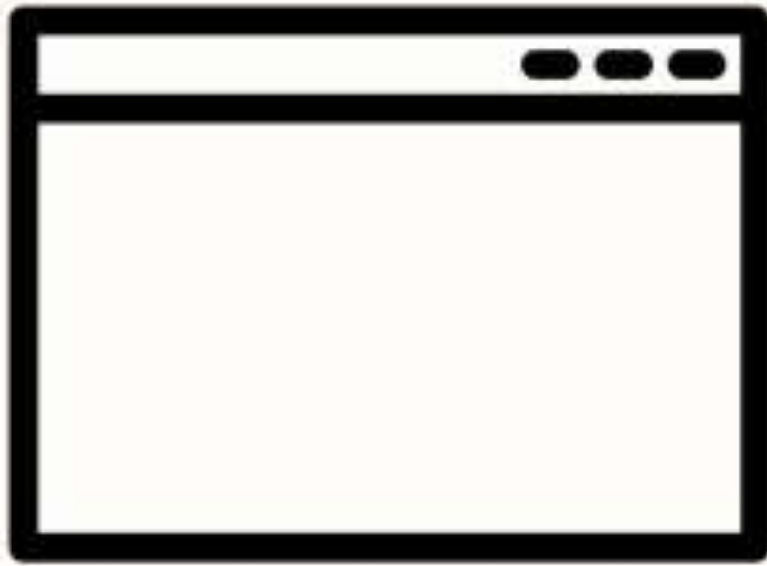
COMPOSITE & RENDER



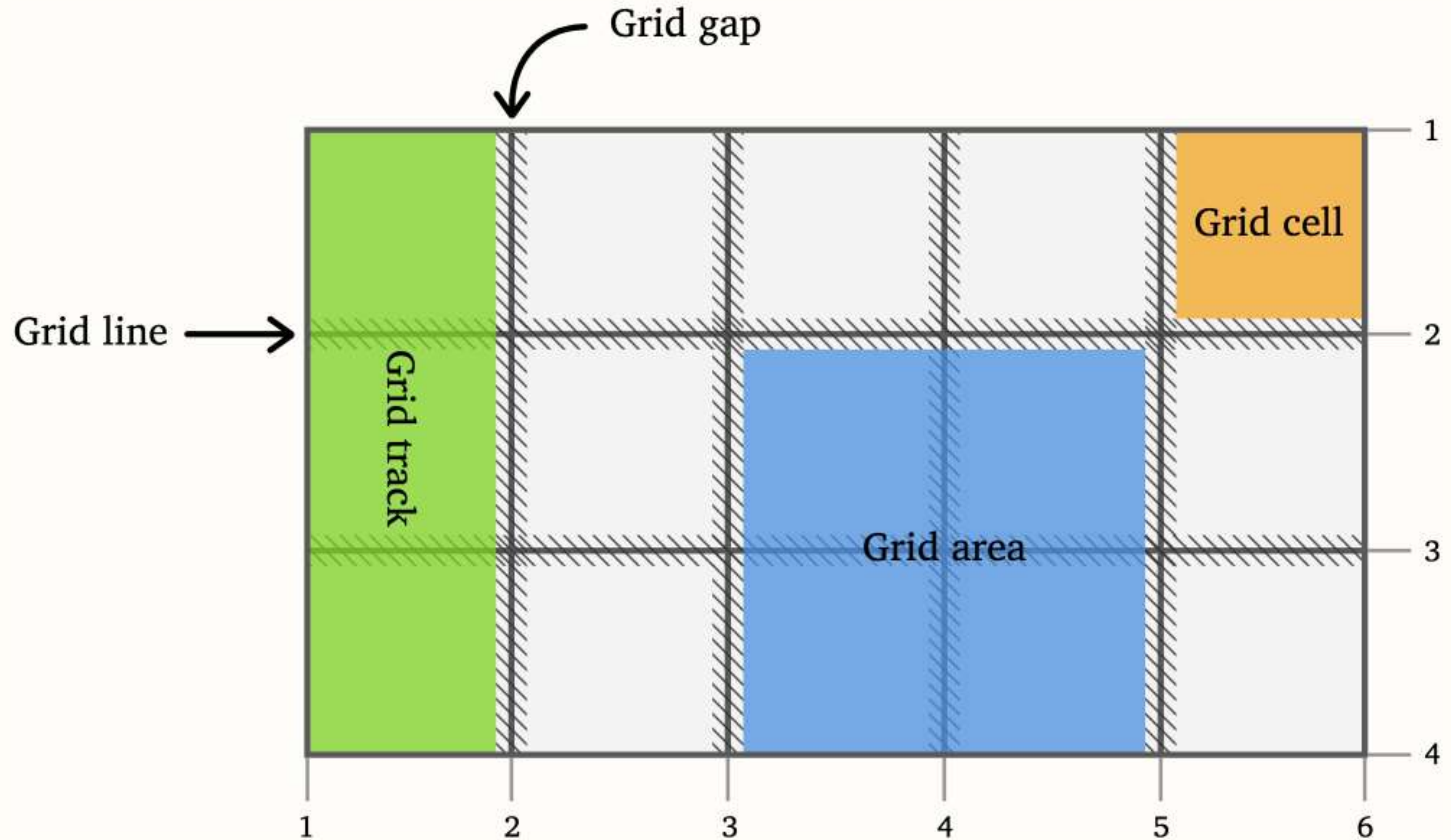
Composite the different layers into one image and render on screen



Web layouts over the years



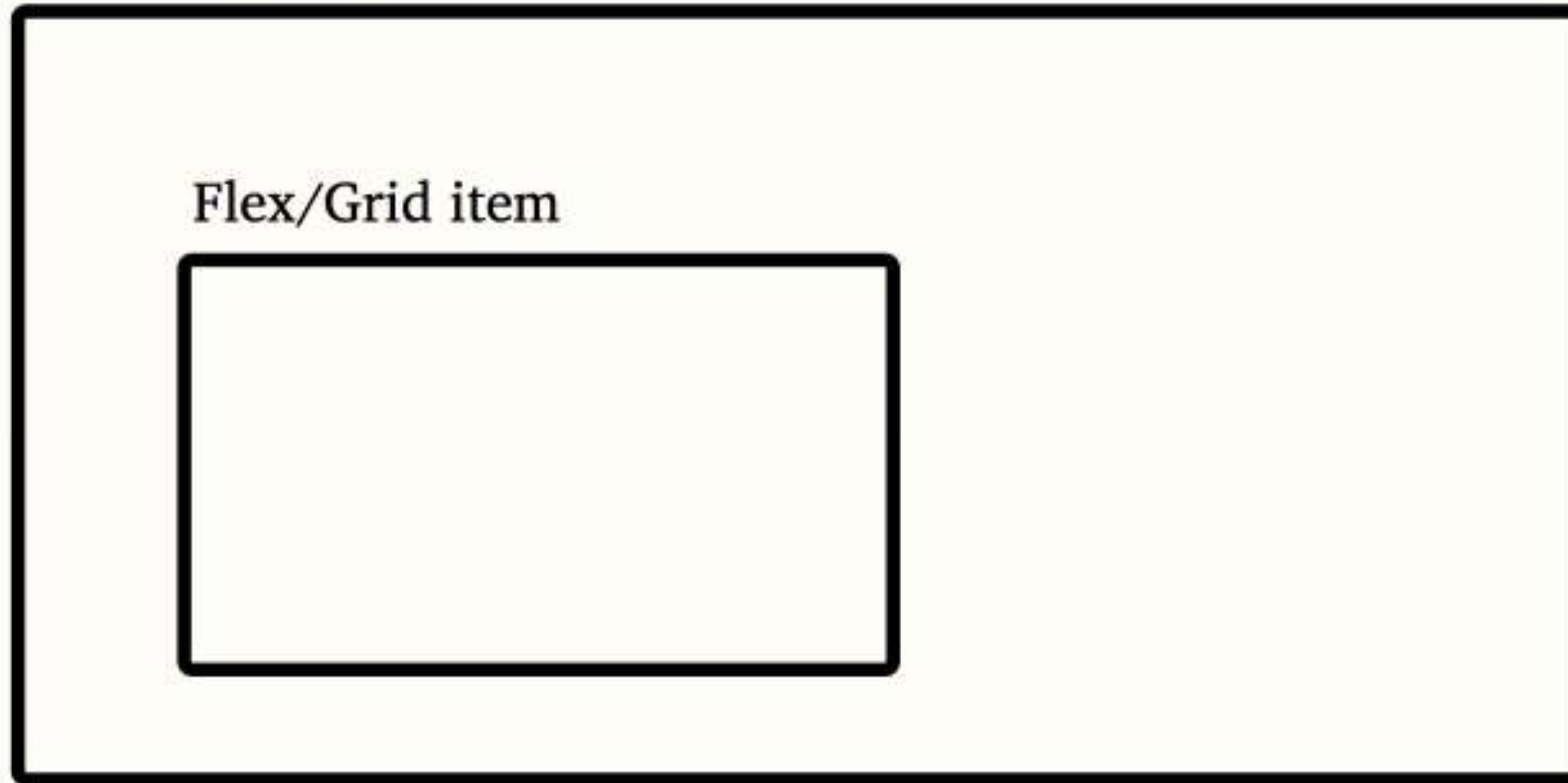
Basic terminology



Flexbox and Grid

Based on the container-child relationship

Flex/Grid container



*“Grid works from the **container** in, other layout methods start with the **item**”*

—Rachel Andrew



Layout technique: `inline-block`

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

```
.inlineblock__item {  
  display: inline-block;  
  width: calc(100% / 3);  
}
```


Layout technique: float

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

```
.float__item {  
  float: left;  
  width: calc(100% / 3);  
}
```

Layout technique: flex

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

```
.flexbox {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.flexbox_item {  
  flex: 0 0 calc(100% / 3);  
}
```

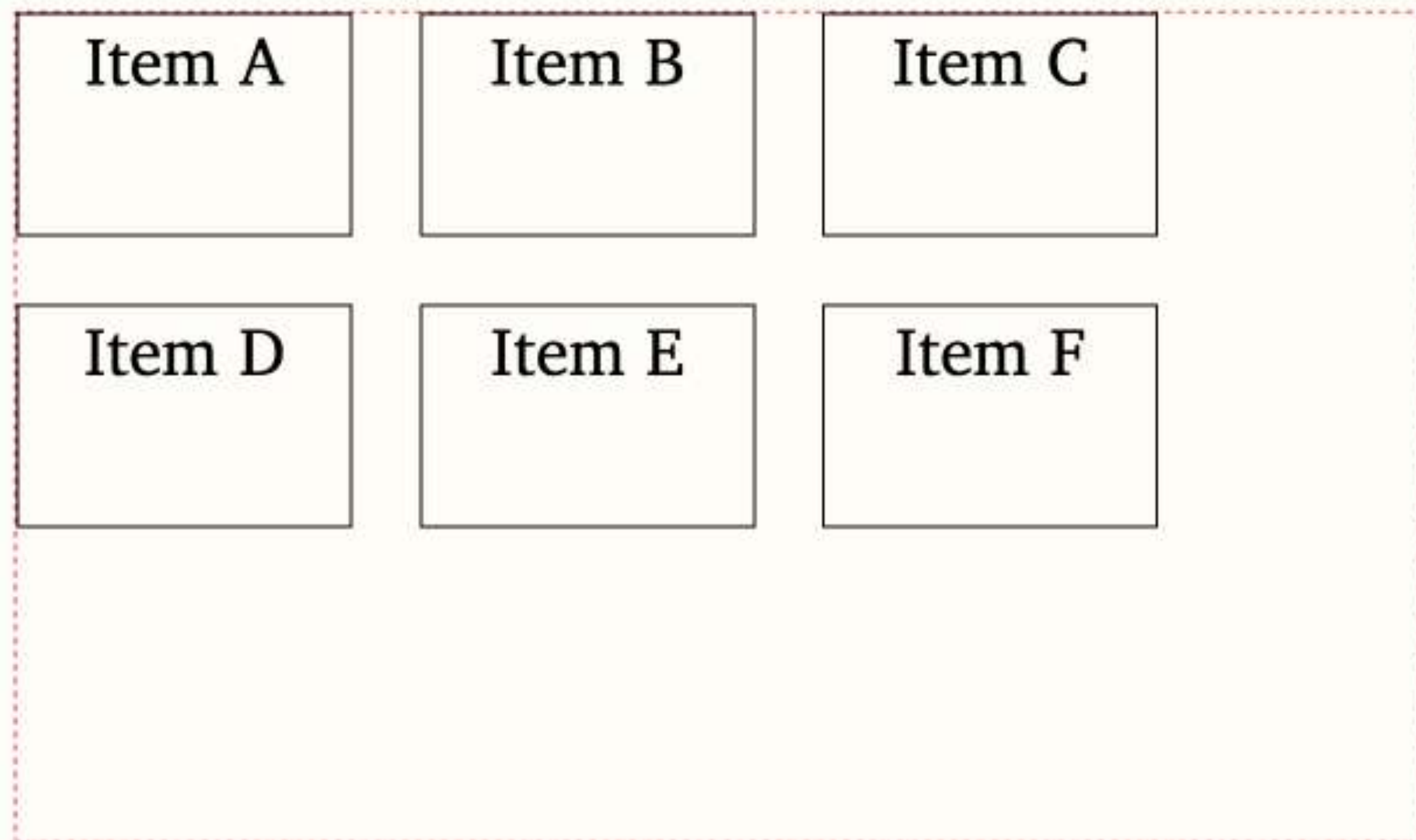

*“Grid is the only layout technique that establishes a **relationship** between rows and columns of grid items.”*



Properties on the Grid container

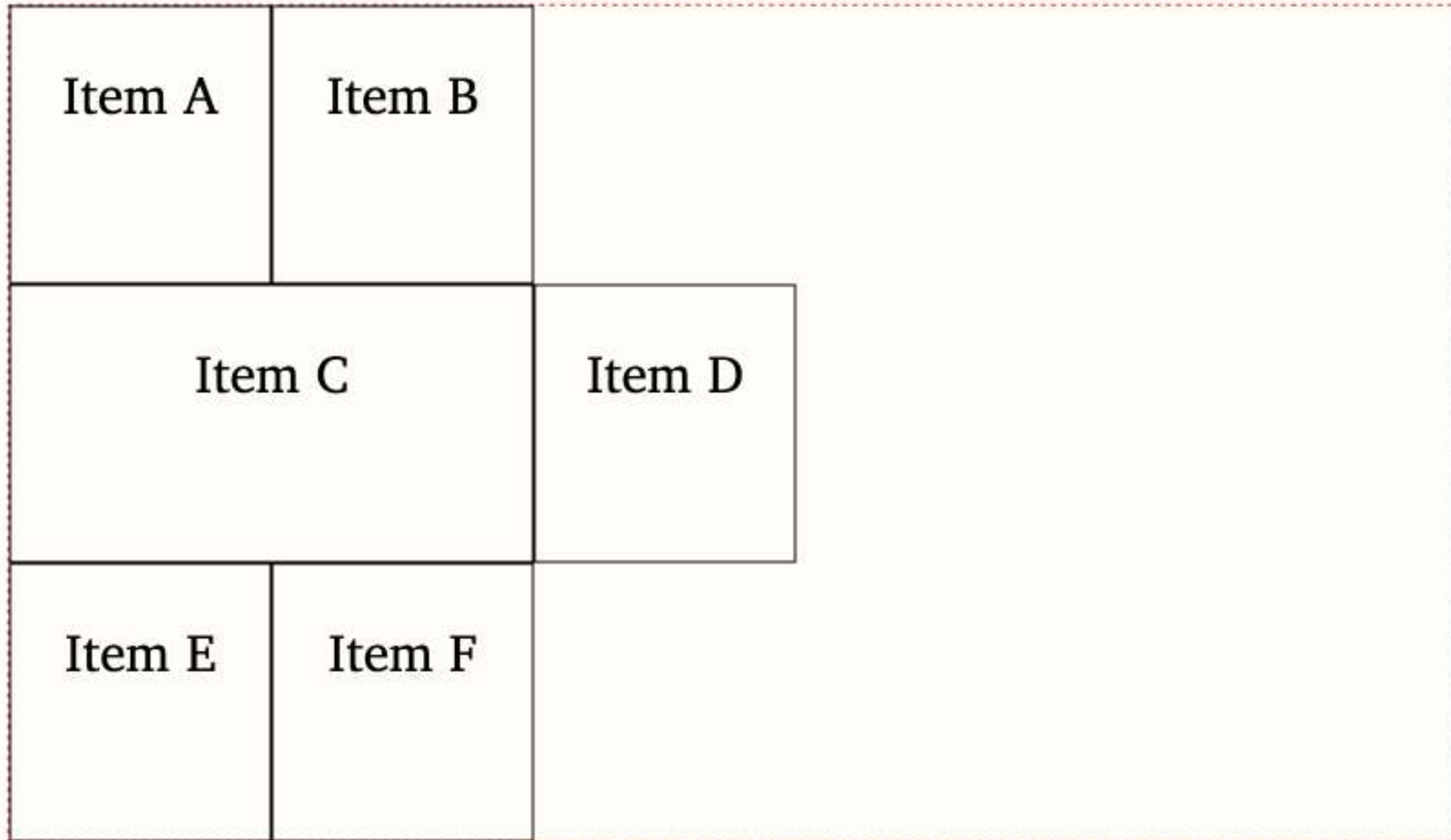
Defining a grid

```
<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>
```



```
.grid1 {
  display: grid;
  grid-template-columns: 150px 150px 150px;
  grid-template-rows: 100px 100px;
  grid-gap: 1em;
}
```

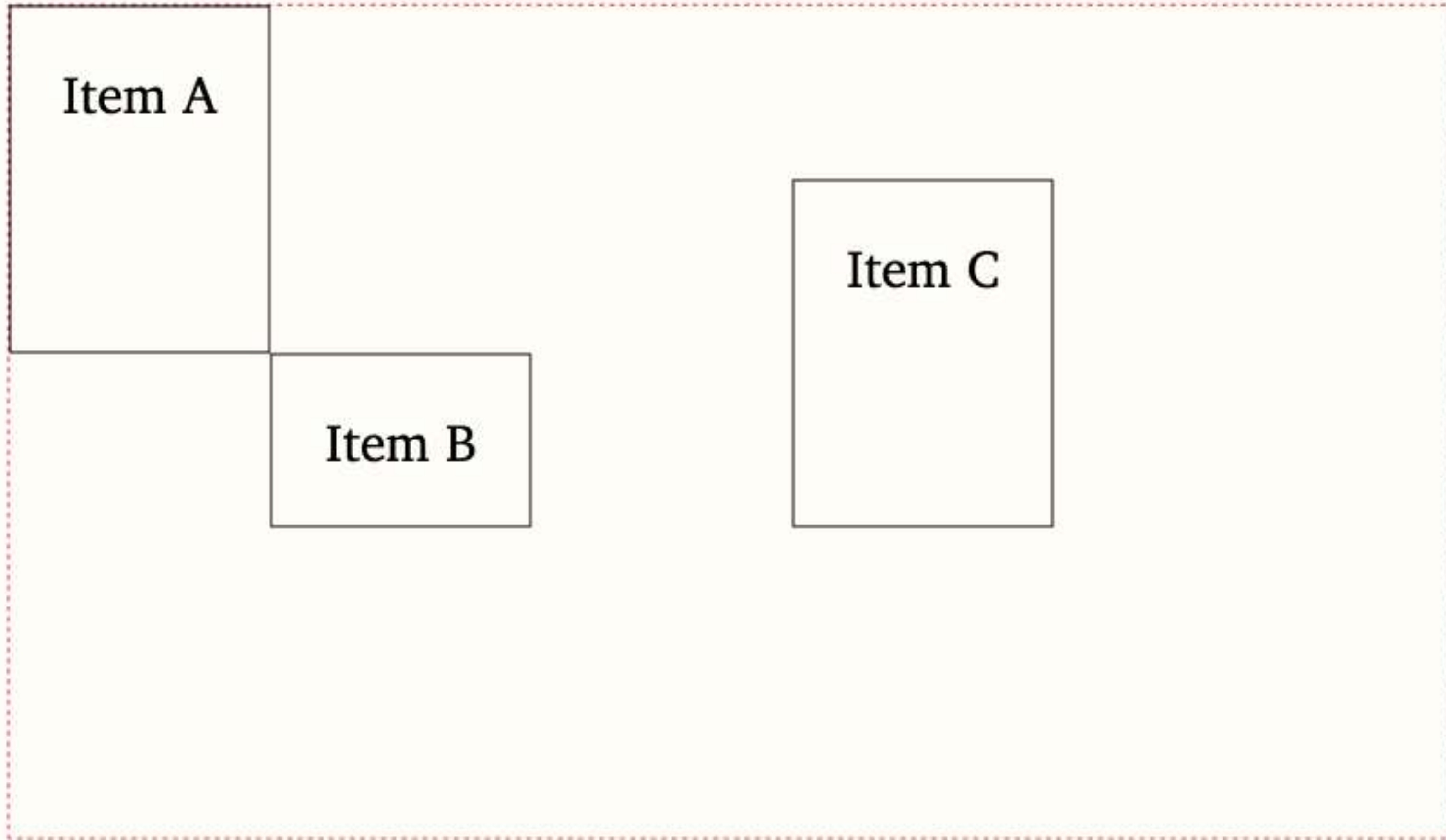

Naming grid lines



```
.grid2 {  
  display: grid;  
  grid-template-columns: [alpha-start] 150px [alpha-end beta-start] 150px [beta-end gamma-start] 150px [gamma-end];  
}  
  
.grid2 .c {  
  grid-column: alpha-start / beta-end;  
}
```



Naming grid areas



```
.grid3 {  
  display: grid;  
  grid-template-columns: 150px  
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; }
```



The fr unit

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

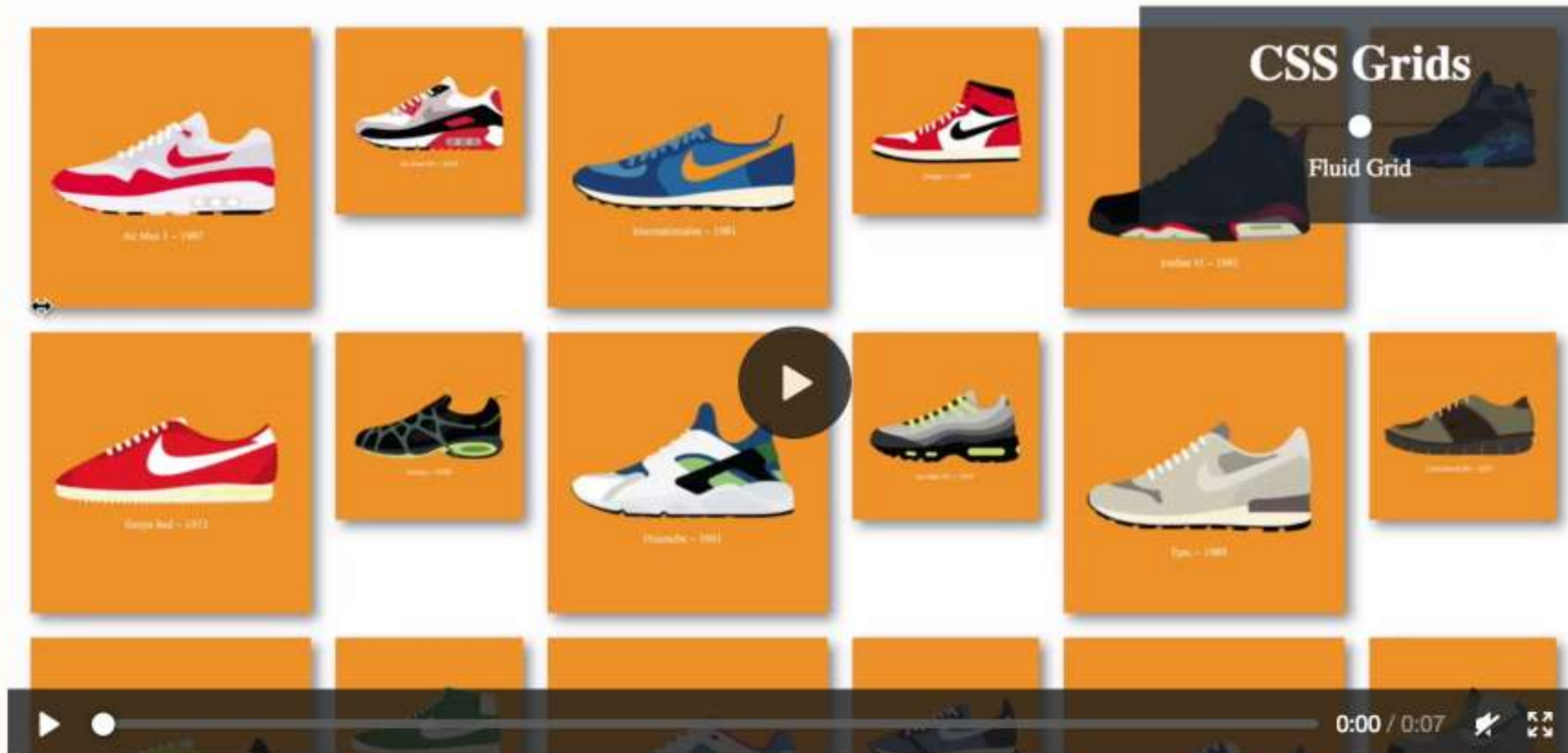
Item A	Item B	Item C
--------	--------	--------

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



Fluid CSS grid

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



The `minmax()` function

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

Item A	Item B	Item C
--------	--------	--------

```
.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
------	------	------	------	------	------	------	------

```
.grid6 {  
  display: grid;  
  grid-template-columns: repeat(4,  
75px 120px);  
}
```

auto-fill versus auto-fit

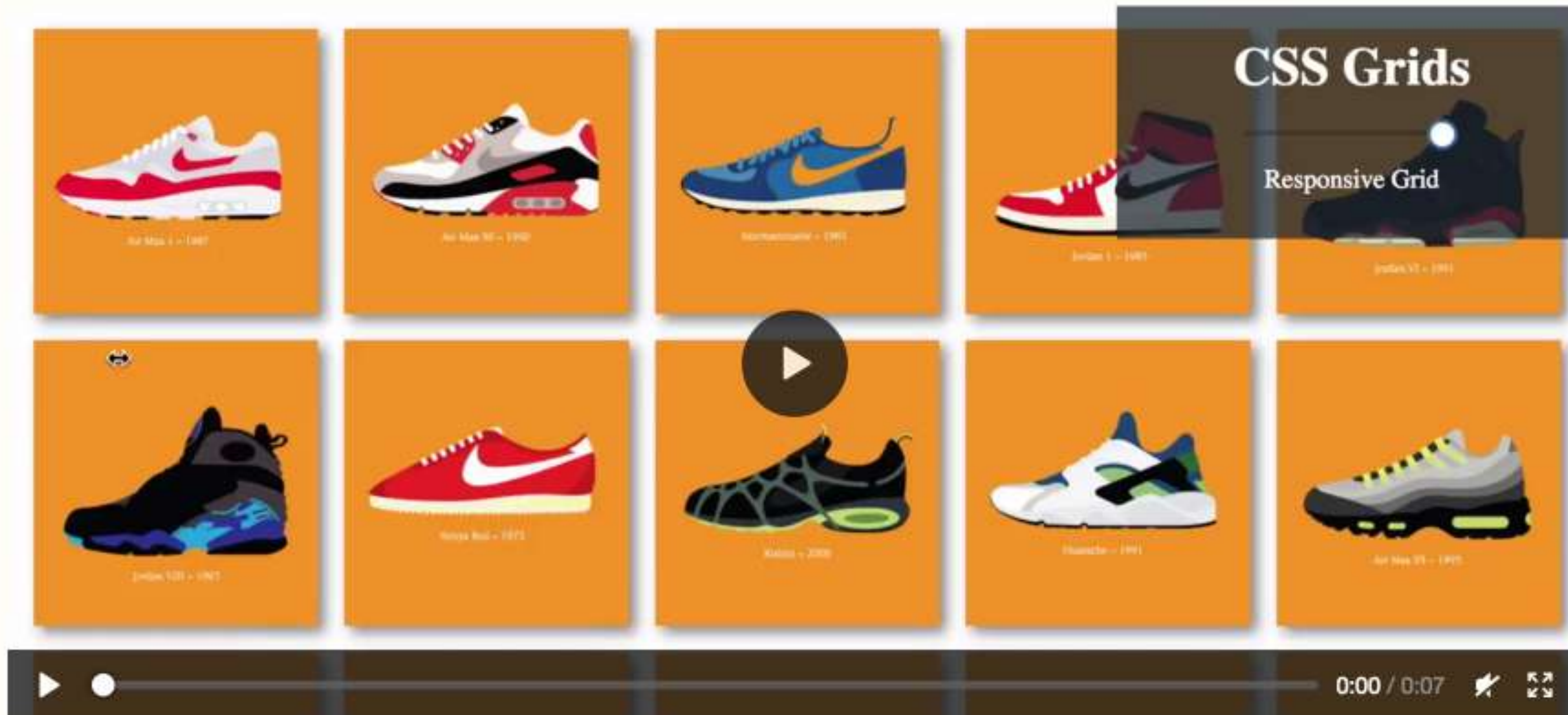
A	B	C	D	E	F

```
.keyword {  
  display: grid;  
  grid-template-columns:  
  repeat(auto-fill, minmax(100px,  
  1fr));  
}
```



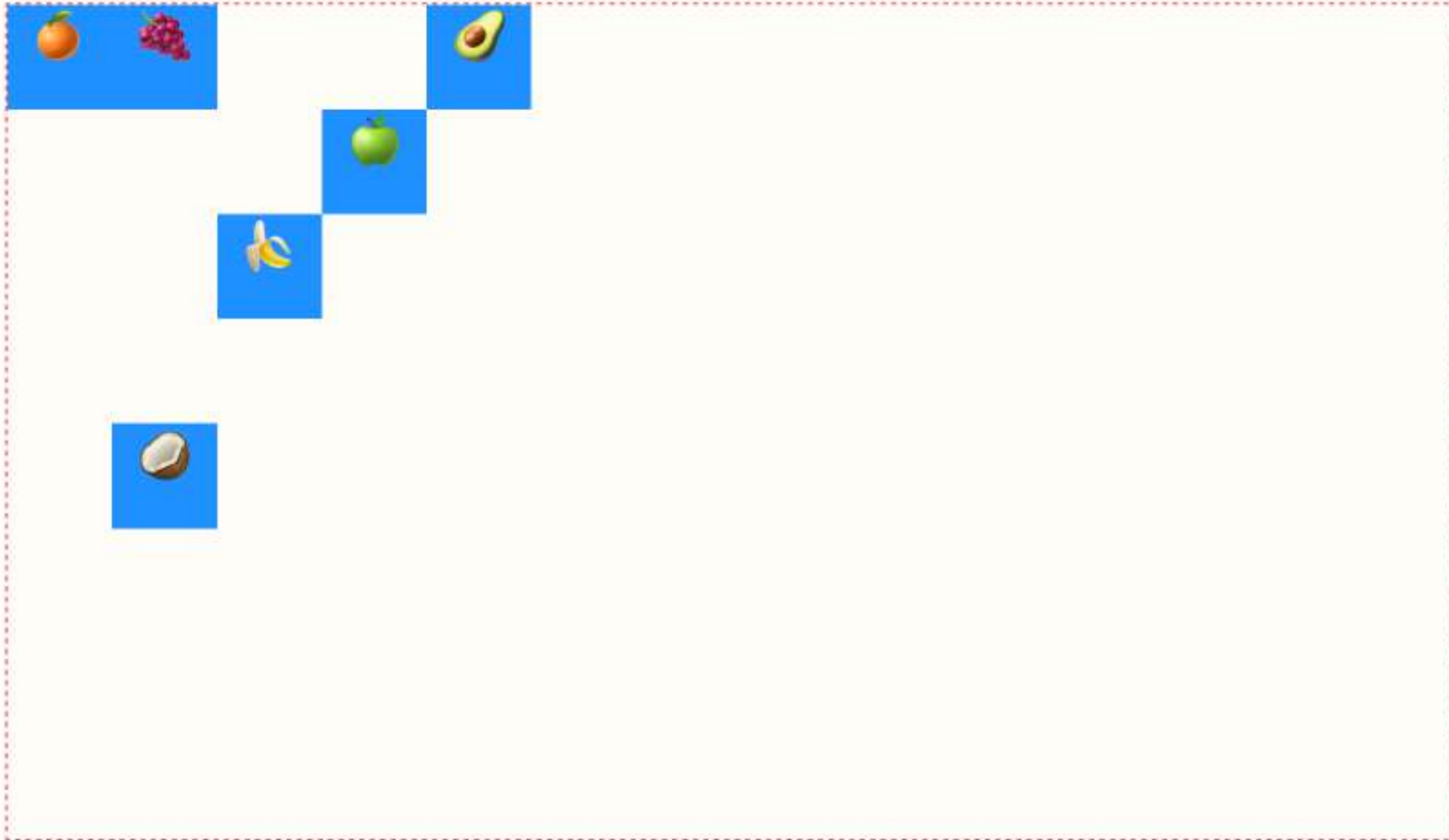
Responsive grid without media queries

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



Properties on the grid item

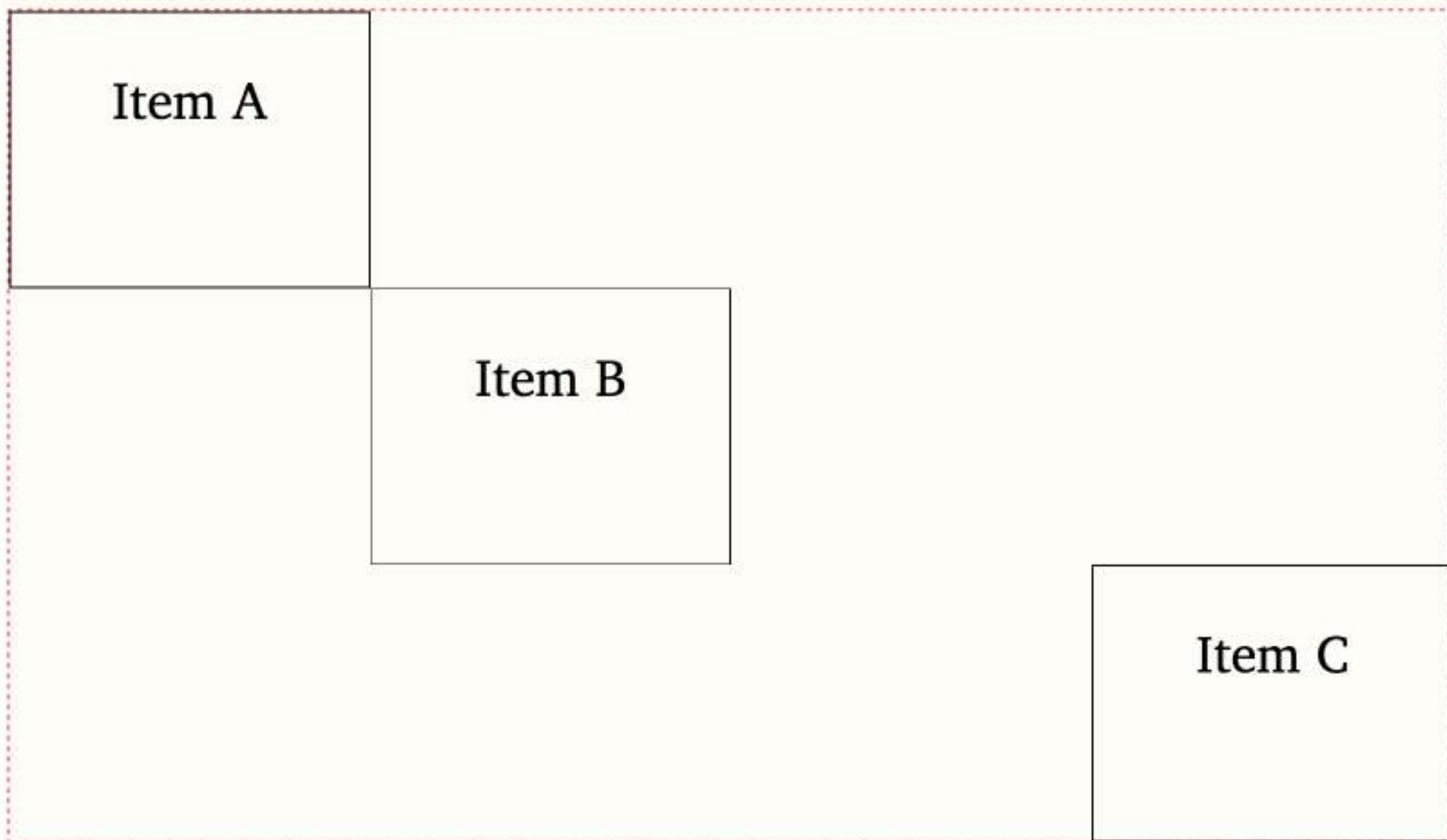
Using grid-column and grid-row



```
.grid7 {  
  display: grid;  
  grid-template-columns: repeat(6,  
2em);  
  grid-template-rows: repeat(6,  
2em);  
}  
  
.grid7__item:nth-child(1) {  
  grid-column: 4;  
  grid-row: 2;  
}  
  
.grid7__item:nth-child(3) {  
  grid-column: 2;  
  grid-row: 5;  
}
```



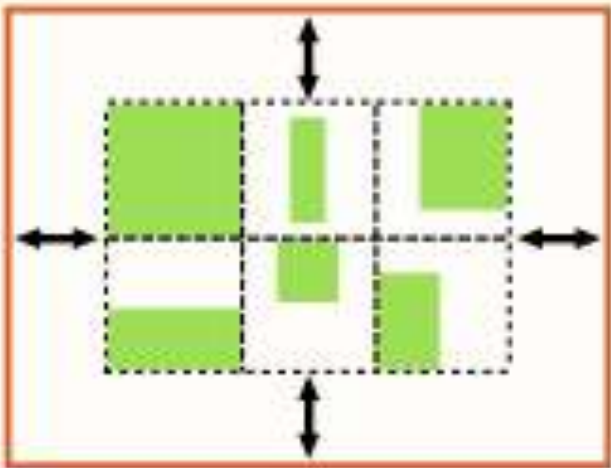
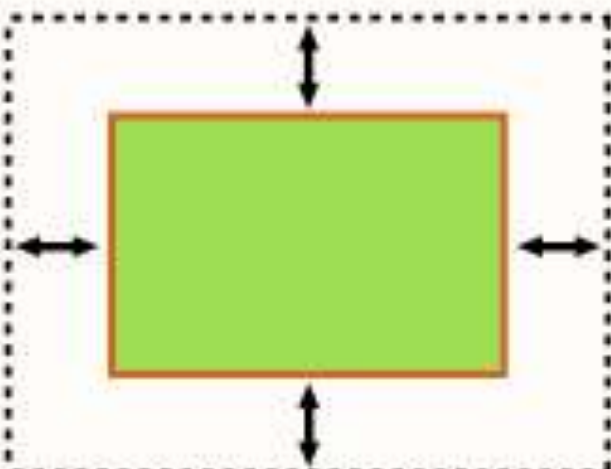
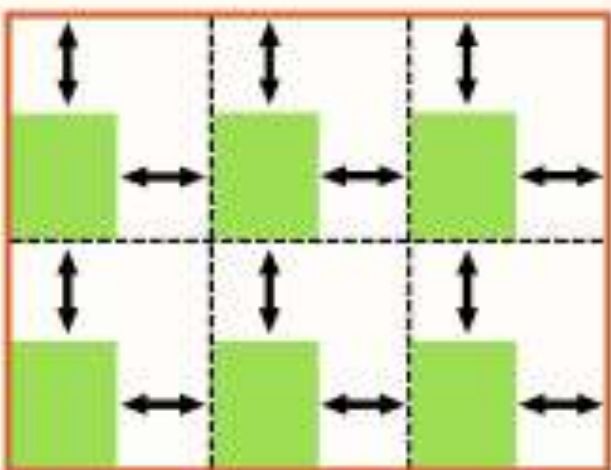
Using grid-area



```
.grid8 {  
  display: grid;  
  grid-template-columns: repeat(4,  
1fr);  
  grid-template-areas: "a . . ."  
                      ". b . ."  
                      ". . . c";  
}  
  
.grid8 .a { grid-area: a; }  
.grid8 .b { grid-area: b; }  
.grid8 .c { grid-area: c; }
```



Aligning your grid items

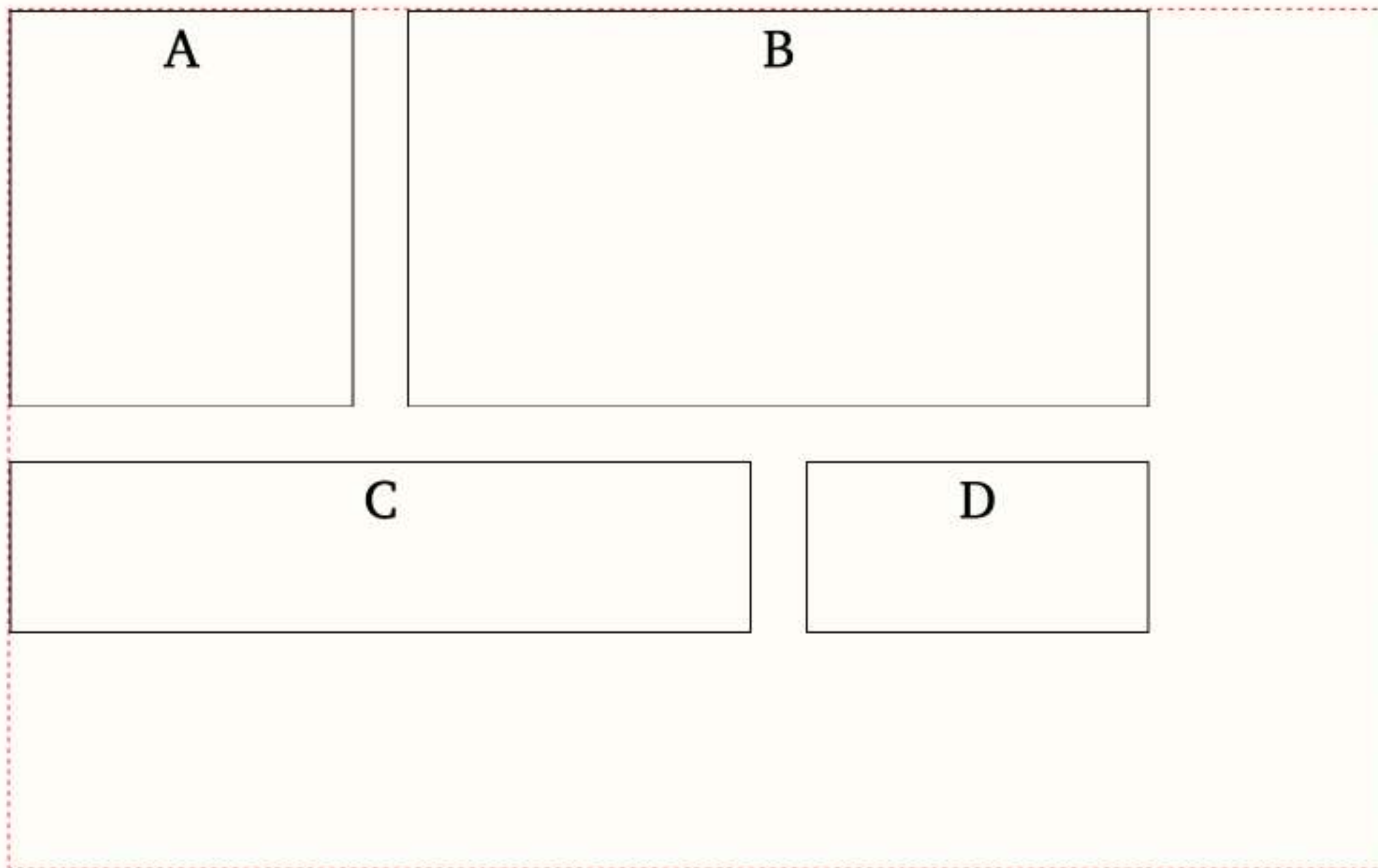
Property	Axis	Aligns		Applies to
<code>justify-content</code>	main/inline	content within element (effectively adjusts padding)		block containers, flex containers and grid containers
<code>align-content</code>	cross/block			
<code>justify-self</code>	inline	element within parent (effectively adjusts margins)		block-level boxes, absolutely-positioned boxes and grid items
<code>align-self</code>	cross/block			absolutely-positioned boxes, flex items and grid items
<code>justify-items</code>	inline	items inside box (controls child items)		block containers and grid containers
<code>align-items</code>	cross/block			flex-containers and grid-containers

Source: [CSS Box Alignment Module Level 3](#)



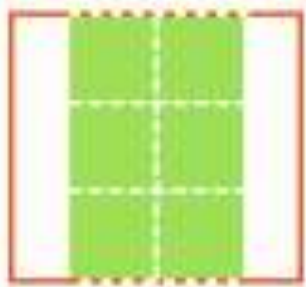

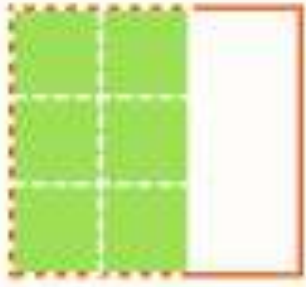

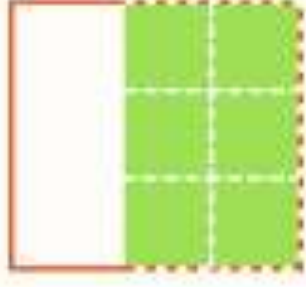



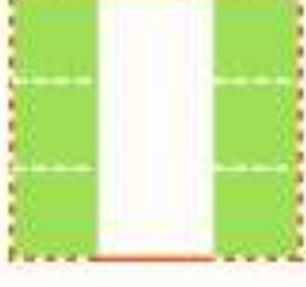
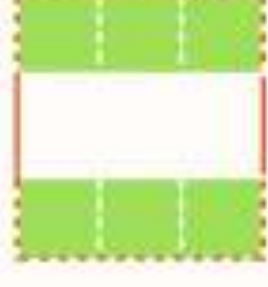

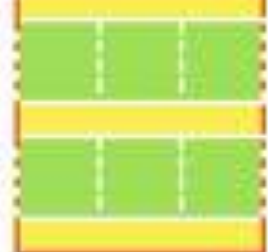
justify/align-content

content-distribution properties



```
.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;  
}
```

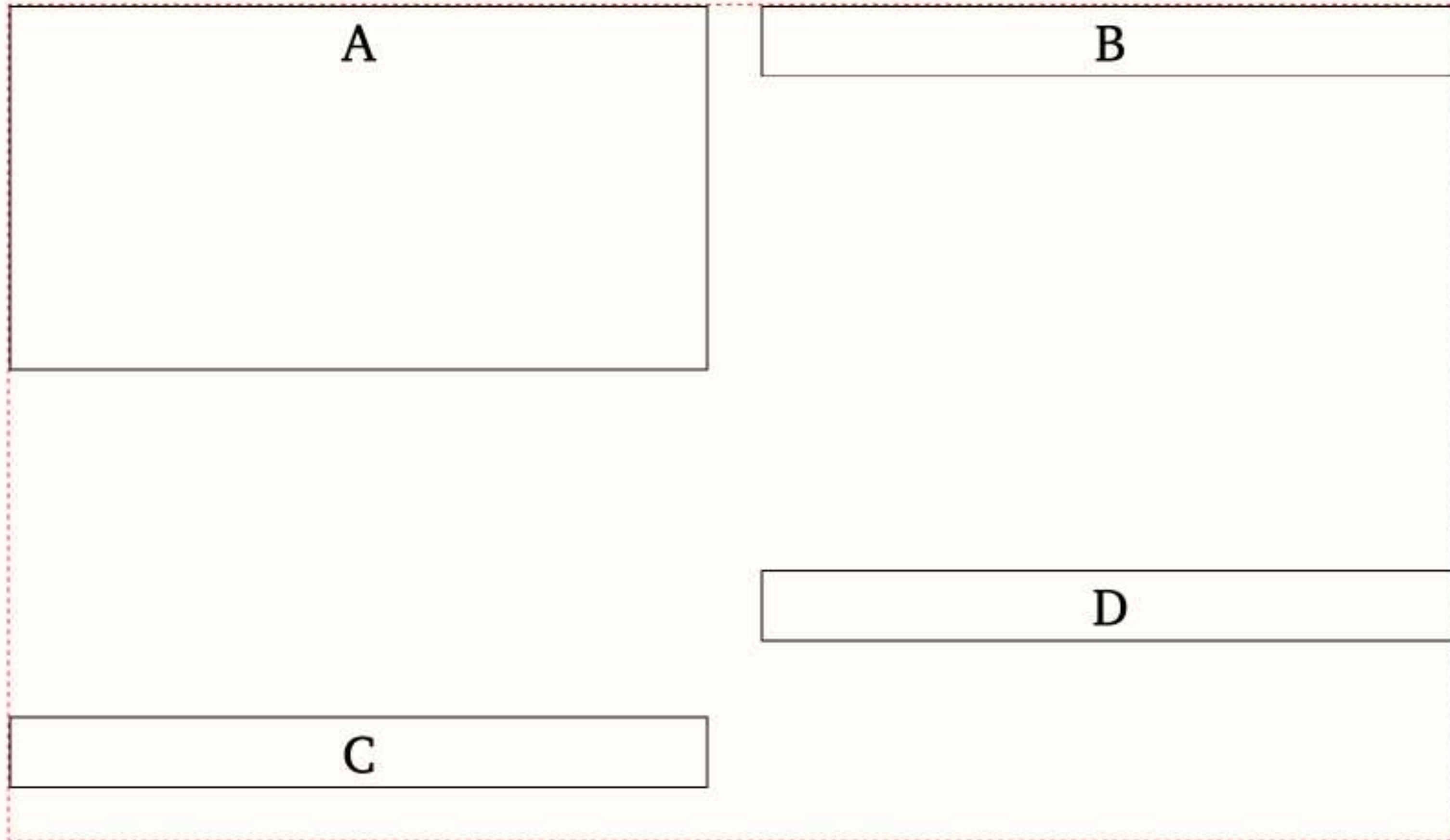


Values	<code>justify-content</code>	<code>align-content</code>
center		
start		
end		
space-around		
space-between		
space-evenly		



justify/align-self

self-alignment properties



```
.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";  
}  
  
.self_itemA {  
  grid-area: a;  
}  
  
.self_itemB {
```



justify/align-items

defaults for justify/align-self

A

B

C

D

```
.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;  
}
```



Examples / Demos

Simple responsive dashboard

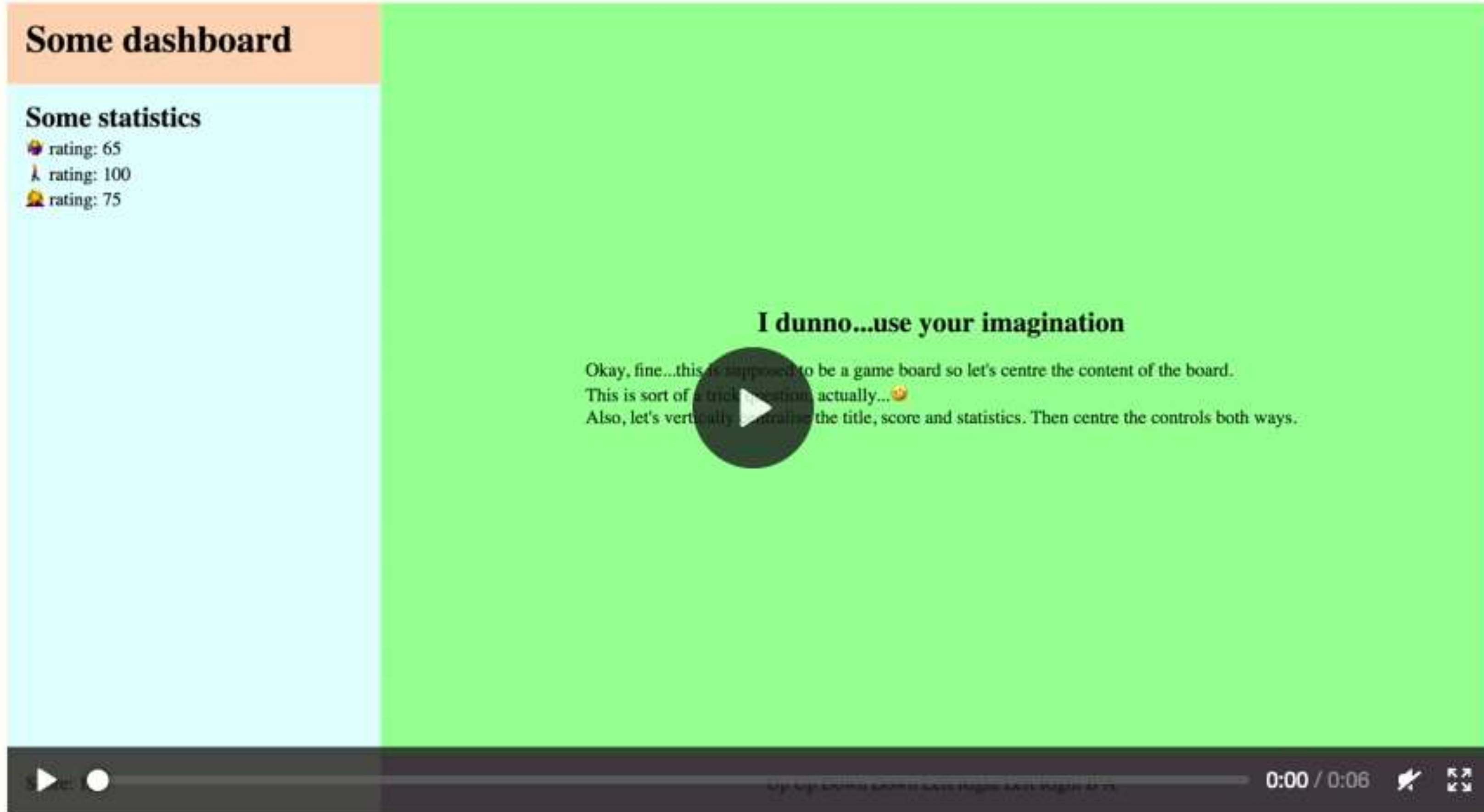
Artist profile page

Responsive header (if time permits)


JSConf.Asia logo (if time permits)



Simple responsive dashboard



Artist profile page



10.2.2010 TORONTO
THE DRAKE HOTEL

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.

[READ MORE](#)

0:00 / 0:09

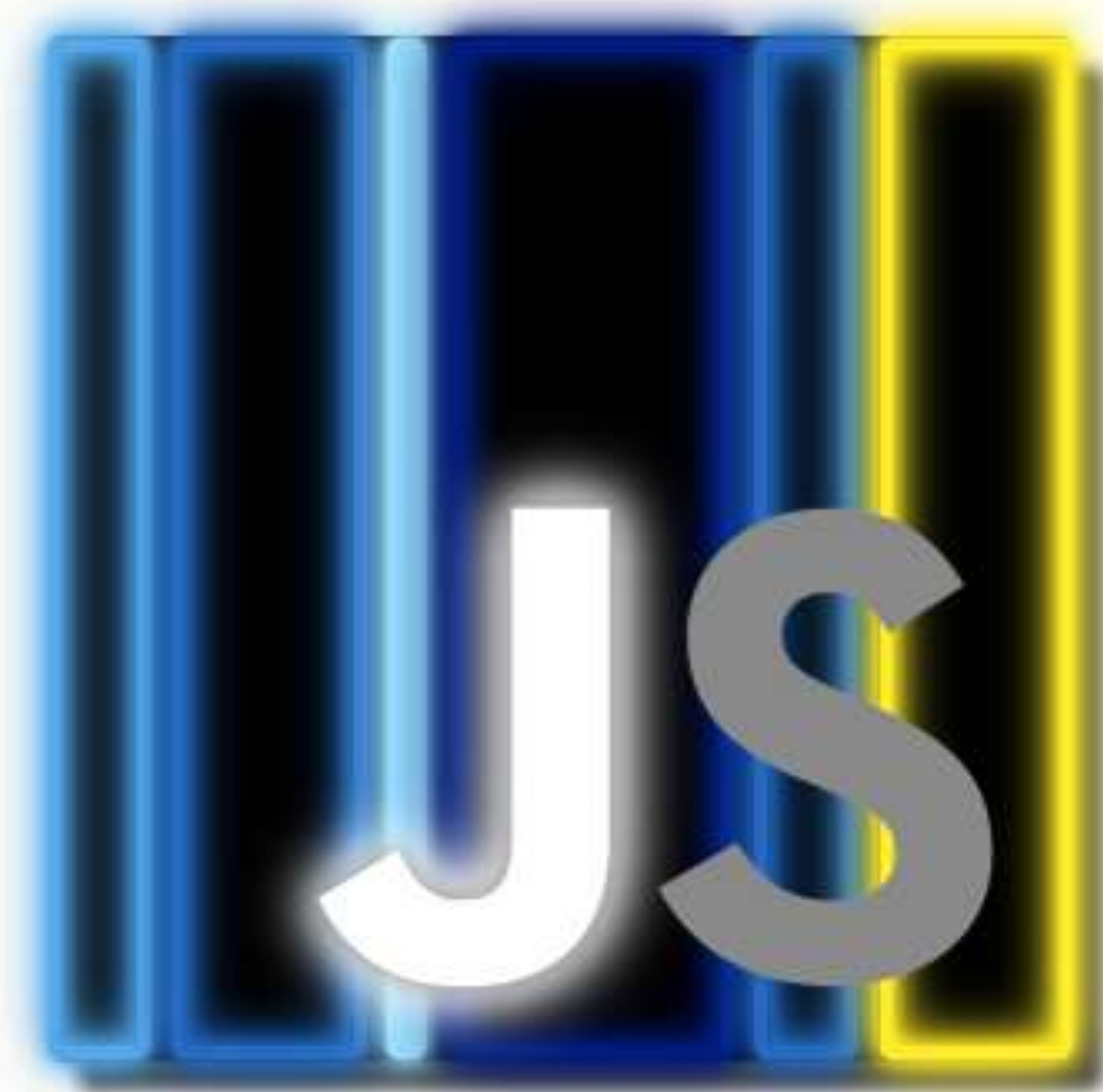
ARTIST SPOTLIGHT



Diagonal header



JSConf.Asia 2017



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](#)

