

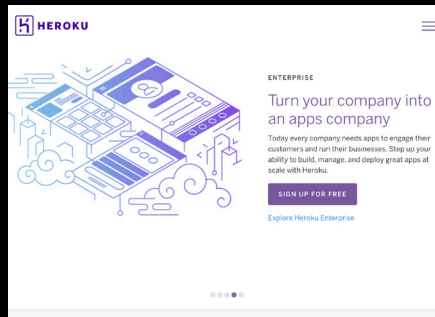


FlexBox and CSS Grid

MODERN STANDARDS for LAYOUT + POSITIONING

FLEXBOX

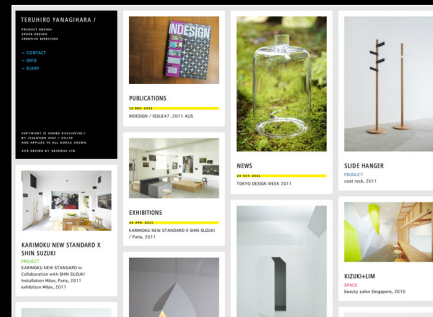
`display: flex;`



- 1-dimensional control
- Enables horizontal **or** vertical arrangement
- Parent element is the flex container; child elements are flex items.

CSS GRID

`display: grid;`



- 2-dimensional control
- Enables horizontal **and** vertical arrangement across a true grid.
- Enormous layout potential with small amount of code
- Parent element is the grid container; child elements are grid items.

Z-INDEX

`z-index: 50;`

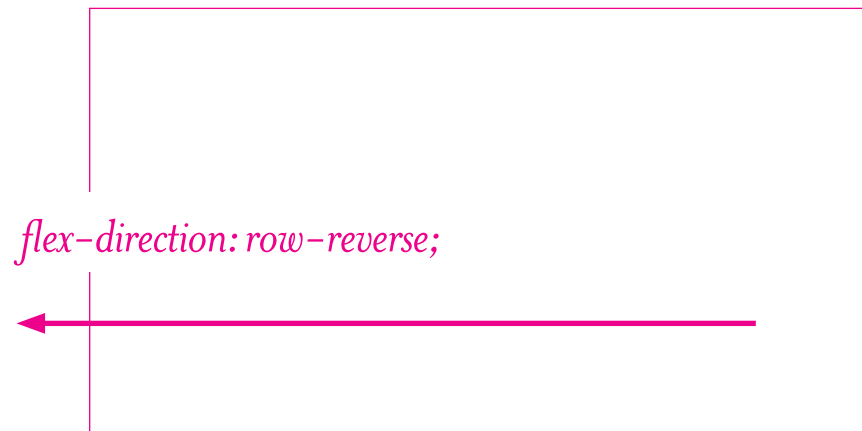
Set parent to position: relative;



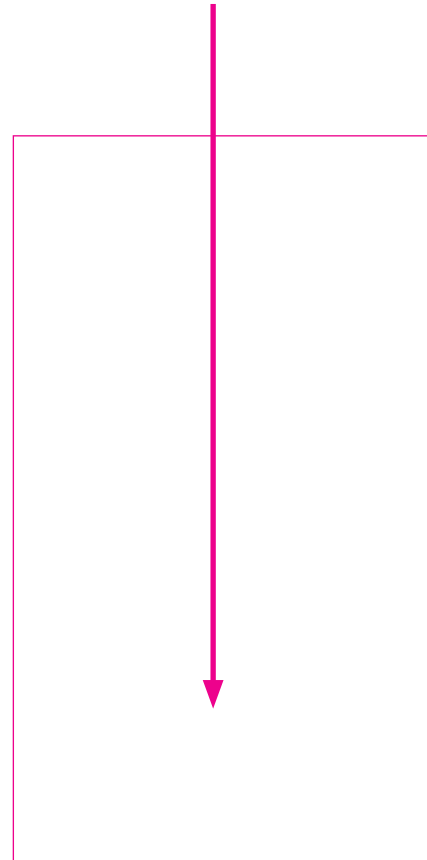
- Controls stacking order of elements when they overlap.
- All elements naturally occur in the depth order they appear in the html, unless otherwise specified by a z-index value. The higher the number, the closer to the front of the screen.
- Parent must be set to *position: relative;*

FLEX: ONE-DIMENSIONAL

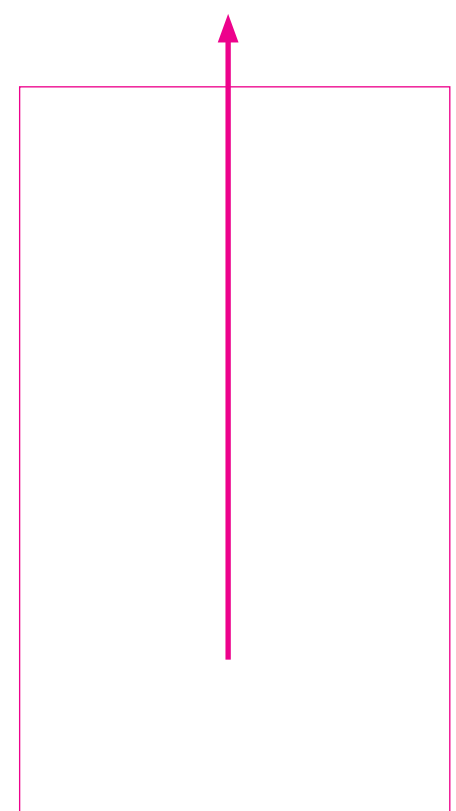
```
#header {  
  display: flex;  
  flex-direction: row; }  
}
```



flex-direction: column;



flex-direction: column-reverse;



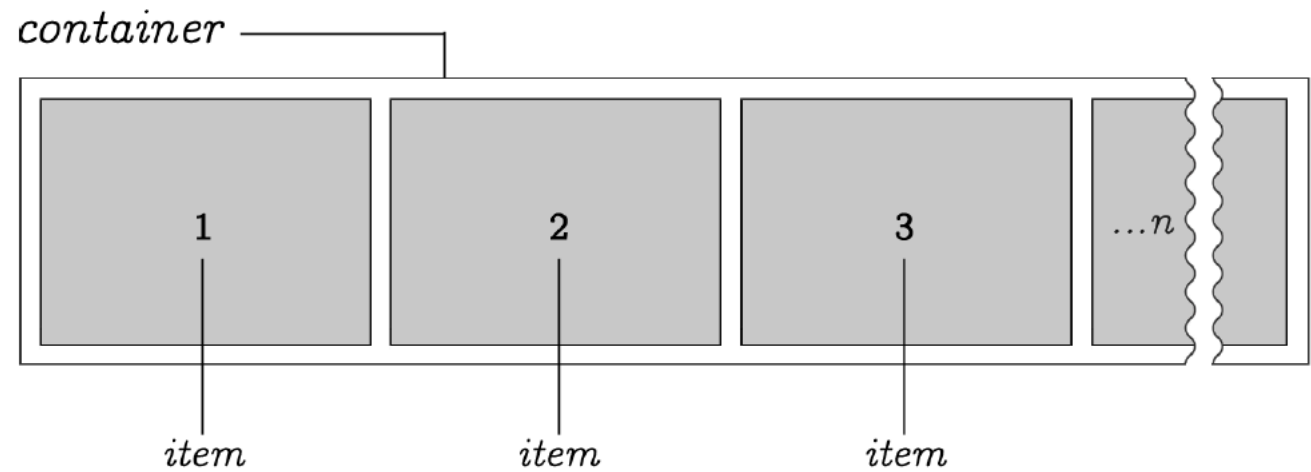
BASIC FLEX SYNTAX

HTML:

```
<div id="container">
  <img _____>
  <h2> Hello world </h2>
  <p> onrvowrhgo </p>
</div>
```

CSS:

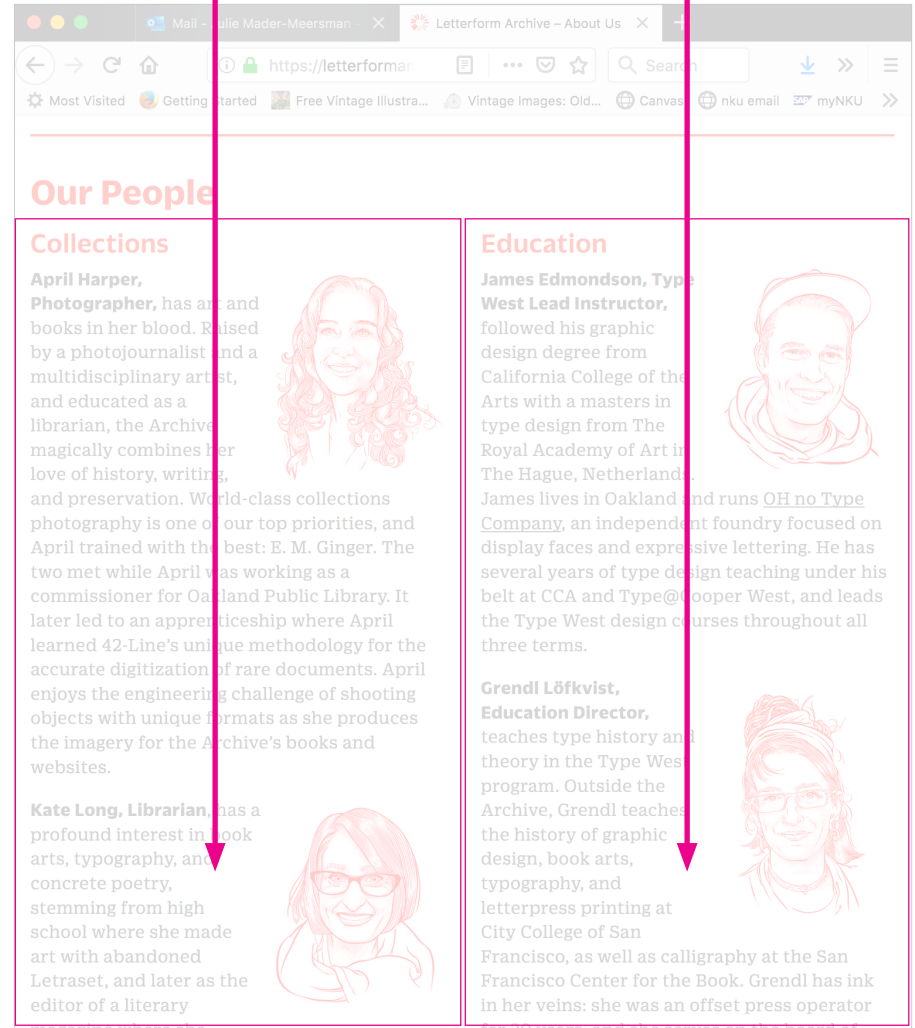
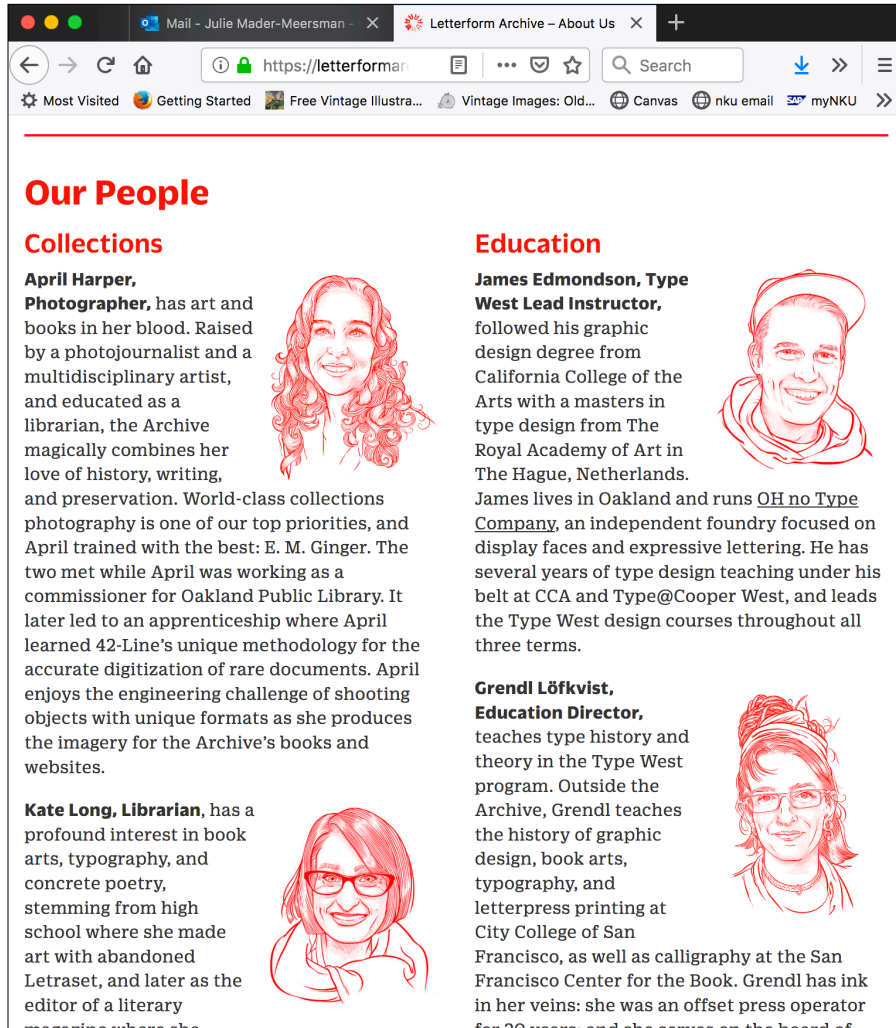
```
#container {
  display: flex;
  flex-direction: row;
}
```



FLEX EXAMPLE: LETTERFORM ARCHIVE

flex-direction: column;

flex-direction: column;



EXCELLENT DIGITAL
RESOURCE FOR FLEXBOX



The image is a screenshot of a webpage from CSS-Tricks. At the top left, there is a logo consisting of an orange asterisk followed by the text 'CSS-TRICKS'. To the right of the logo are a search icon (magnifying glass) and a hamburger menu icon (three horizontal lines). Below the navigation bar, the word 'Guide' is written in a small, light gray font. The main title of the article is 'A Complete Guide to Flexbox', displayed in a large, bold, white font. Underneath the title, the text 'Last Updated Feb 3, 2020' is shown in a smaller, light gray font. The main body of the article begins with a paragraph: 'Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.' The background of the article content is dark, with a subtle, faint image of a pair of glasses. The bottom of the screenshot shows a decorative gradient bar transitioning from orange on the left to pink on the right.

*** CSS-TRICKS**

Guide

A Complete Guide to Flexbox

Last Updated
Feb 3, 2020

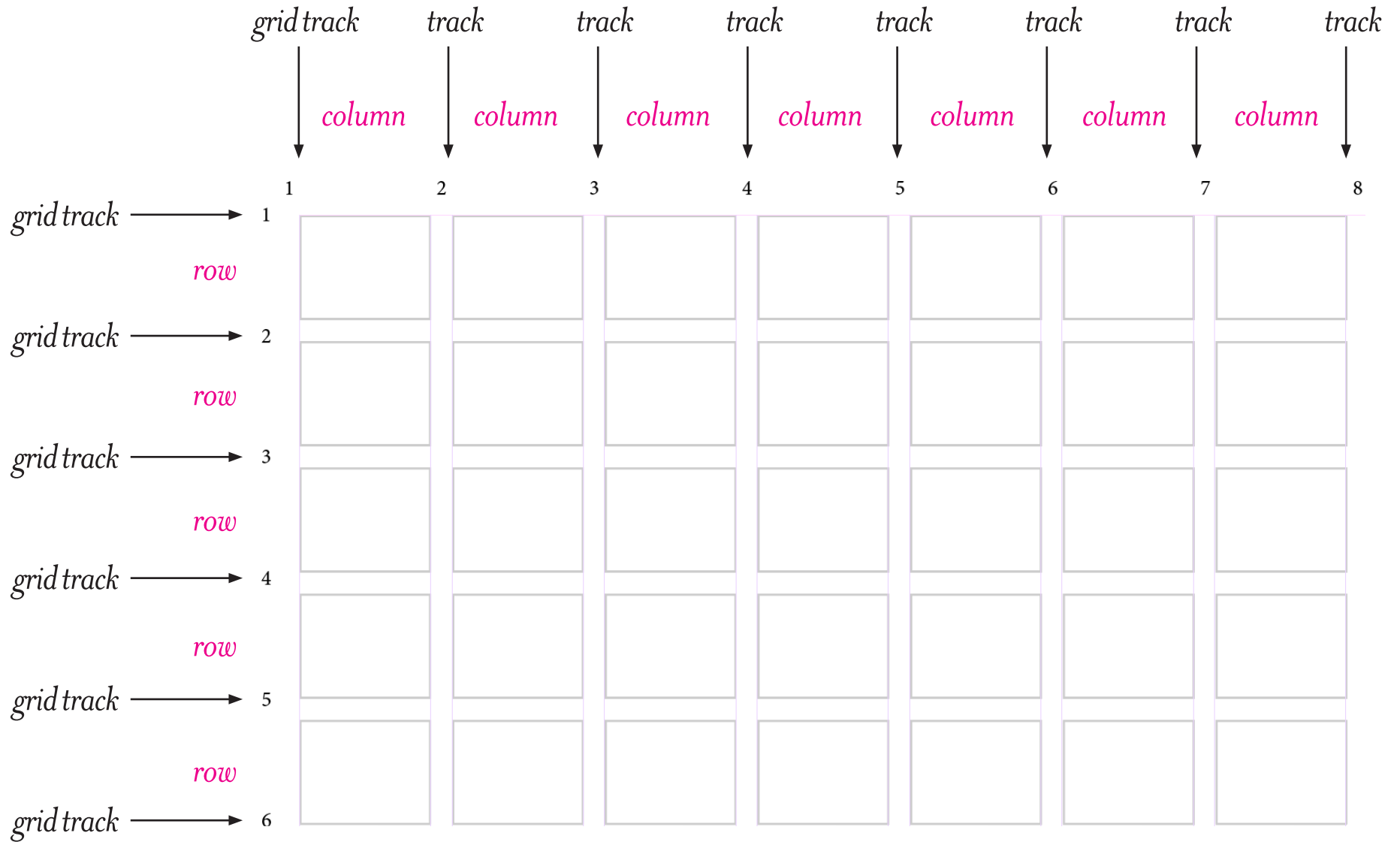
Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.

Chris Coyier: CSS-Tricks

“The language of CSS Grid is incredibly straightforward...There are two main steps to set up a page using CSS Grid: define the grid tracks and then place elements in those areas.”

— JOHNA MANDEL

CSS GRID ANATOMY: TWO-DIMENSIONAL

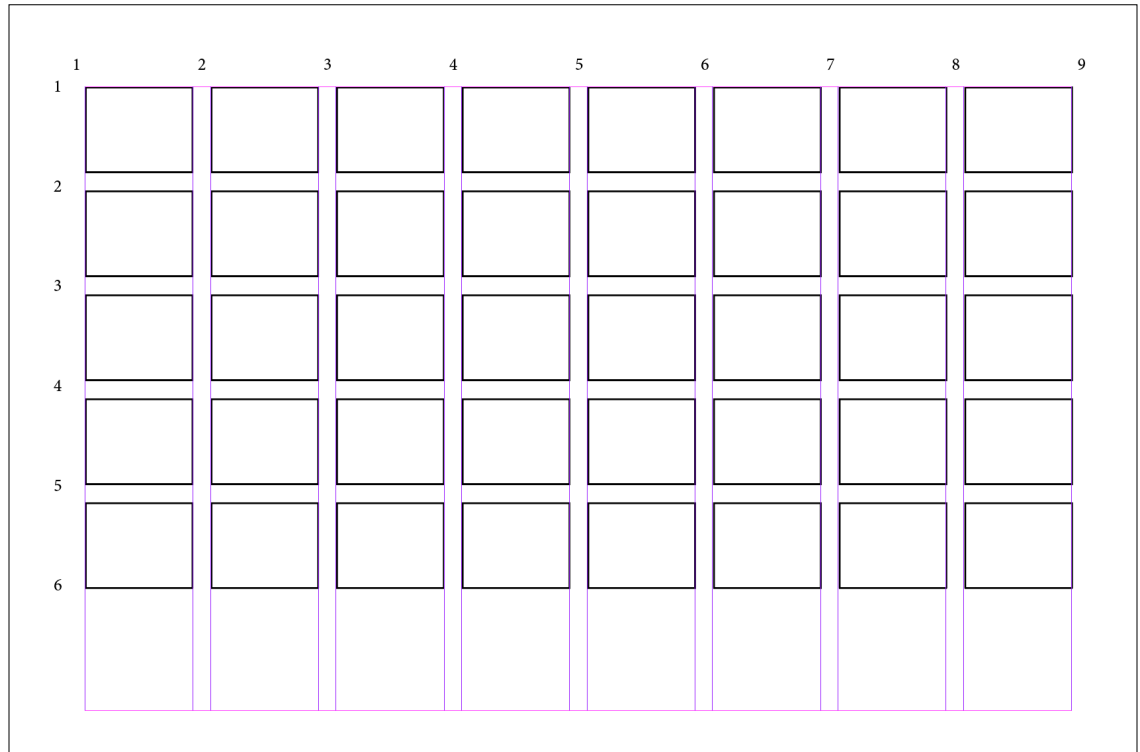


DEFINING A GRID & SHORTHAND

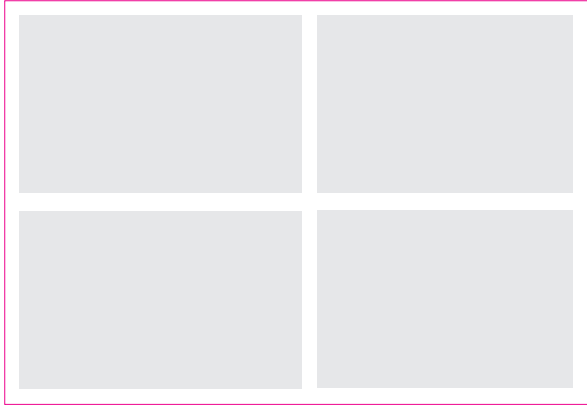
```
div {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr 1fr  
  1fr 1fr 1fr 1fr;  
  grid-template-rows: auto;  
  grid-gap: 10px;  
}
```

IS THE SAME AS:

```
div {  
  display: grid;  
  grid-template-columns: repeat(8, 1fr);  
  grid-template-rows: auto;  
  grid-gap: 10px;  
}
```



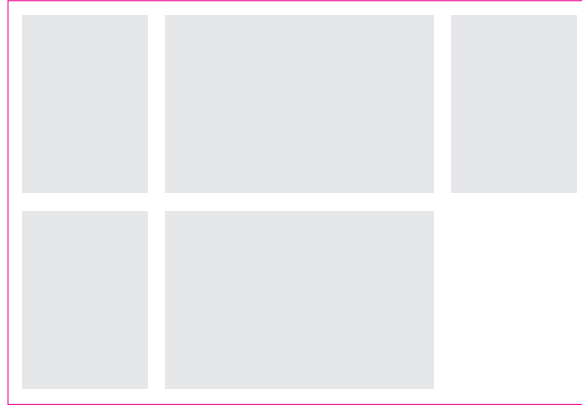
DEFINING DIFFERENT GRIDS



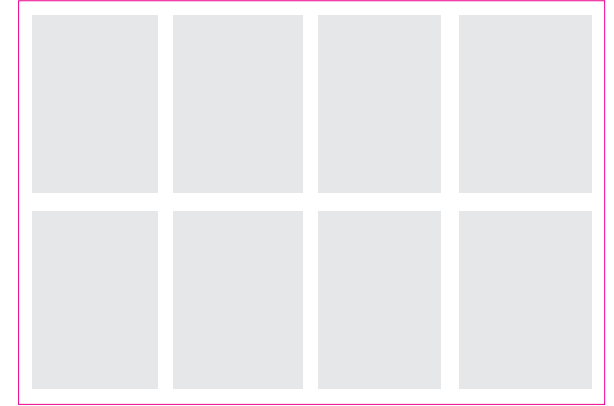
```
div {  
  display: grid;  
  grid-template-columns: 50% 50%;  
  grid-template-rows: auto;  
  grid-gap: 10px;  
}
```

IS THE SAME AS:

```
div {  
  display: grid;  
  grid-template-columns: repeat(2, 1fr);  
  grid-template-rows: auto;  
  grid-gap: 10px;  
}
```



```
article {  
  display: grid;  
  grid-template-columns: 1fr 2fr 1fr;  
  grid-template-rows: auto;  
  grid-gap: 10px;  
}
```



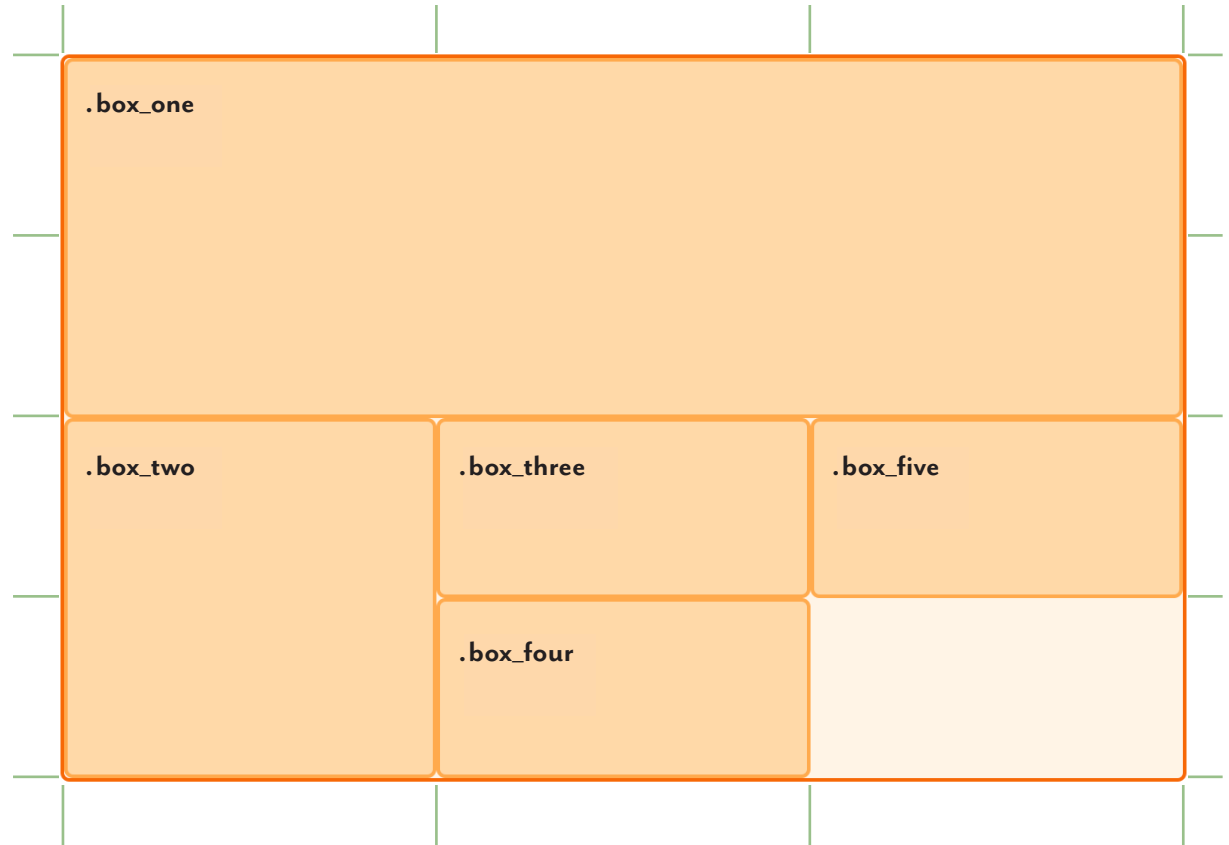
CAN EASILY RE-FLOW IN BROWSERS WITH:

```
div {  
  display: grid;  
  grid-template-columns:  
    repeat(auto-fit, 200px);  
  grid-template-rows: auto;  
  grid-gap: 15px;  
}
```

Excellent example of intrinsic web design that's possible with grid.

CSS GRID: SPECIFIC PLACEMENT

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-auto-rows: 100px;  
}  
  
.box_one {  
  grid-column-start: 1;  
  grid-column-end: 4;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}  
  
.box_two {  
  grid-column-start: 1;  
  grid-column-end: 2;  
  grid-row-start: 3;  
  grid-row-end: 5;  
}
```



CSS GRID: PLACEMENT PLANNING & SHORTHAND

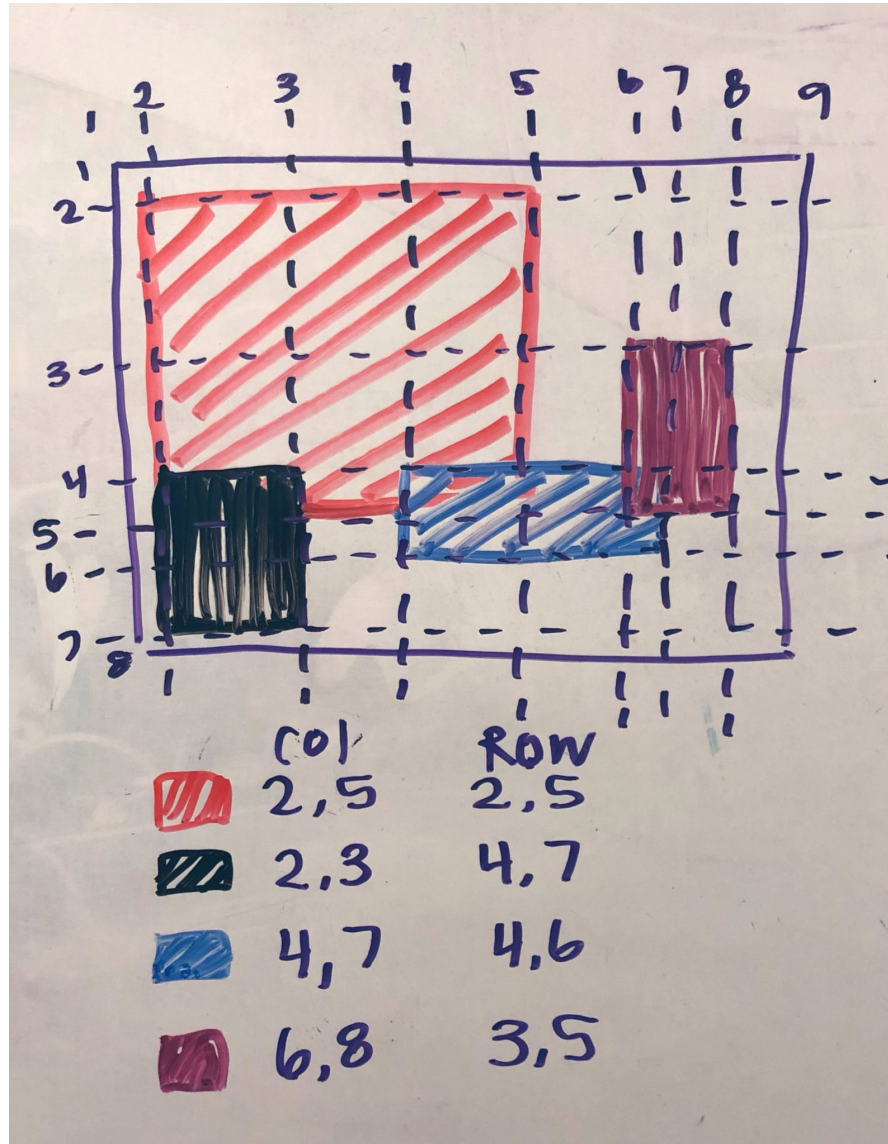
```
.item_1 {  
  grid-column-start: 2;  
  grid-column-end: 5;  
  grid-row-start: 2;  
  grid-row-end: 5;  
}
```

```
.item_2 {  
  grid-column-start: 2;  
  grid-column-end: 3;  
  grid-row-start: 4;  
  grid-row-end: 7;  
}
```

IS THE SAME AS:

```
.item_1 {  
  grid-column: 2 / 5;  
  grid-row: 2 / 5;  
}
```

```
.item_2 {  
  grid-column: 2 / 3;  
  grid-row: 4 / 7;  
}
```



GRID EXAMPLE:
TERUHIRO YANAGIHARA

TERUHIRO YANAGIHARA /

PRODUCT DESIGN
SPACE DESIGN
CREATIVE DIRECTION

→ CONTACT
→ INFO
→ DIARY

COPYRIGHT IS OWNED EXCLUSIVELY
BY ISOLATION UNIT / CO.LTD
AND APPLIES TO ALL WORKS SHOWN.
SITE DESIGN BY DEZIGNO LTD.



KARIMOKU NEW STANDARD X
SHIN SUZUKI

PROJECT
KARIMOKU NEW STANDARD in
Collaboration with SHIN SUZUKI
Installation Milan, Paris, 2011
exhibition Milan, 2011



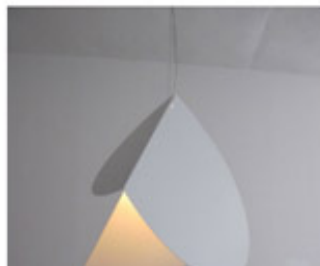
PUBLICATIONS

13 DEC. 2011
INDESIGN / ISSUE47 .2011 AUS



EXHIBITIONS

28 APR. 2011
KARIMOKU NEW STANDARD X SHIN SUZUKI
/ Paris, 2011



NEWS

28 OCT. 2011
TOKYO DESIGN WEEK 2011



SLIDE HANGER

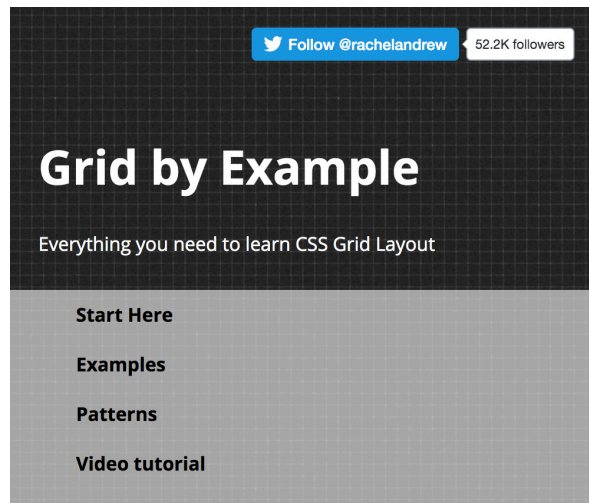
PRODUCT
coat rack, 2011



KIZUKI+LIM

SPACE
beauty salon Singapore, 2010

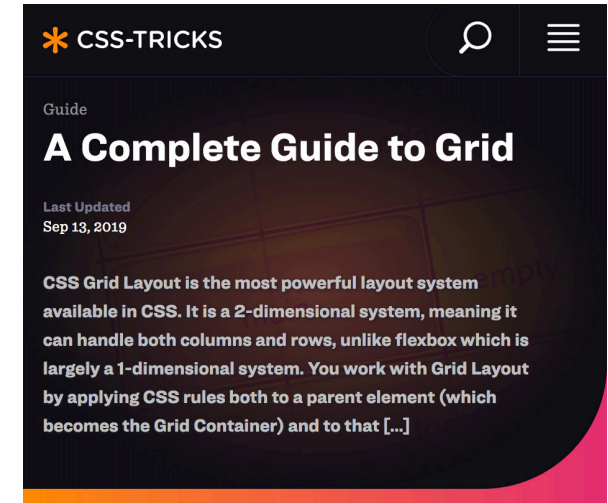
CSS GRID: TOP RESOURCES



Rachel Andrew: Grid by Example



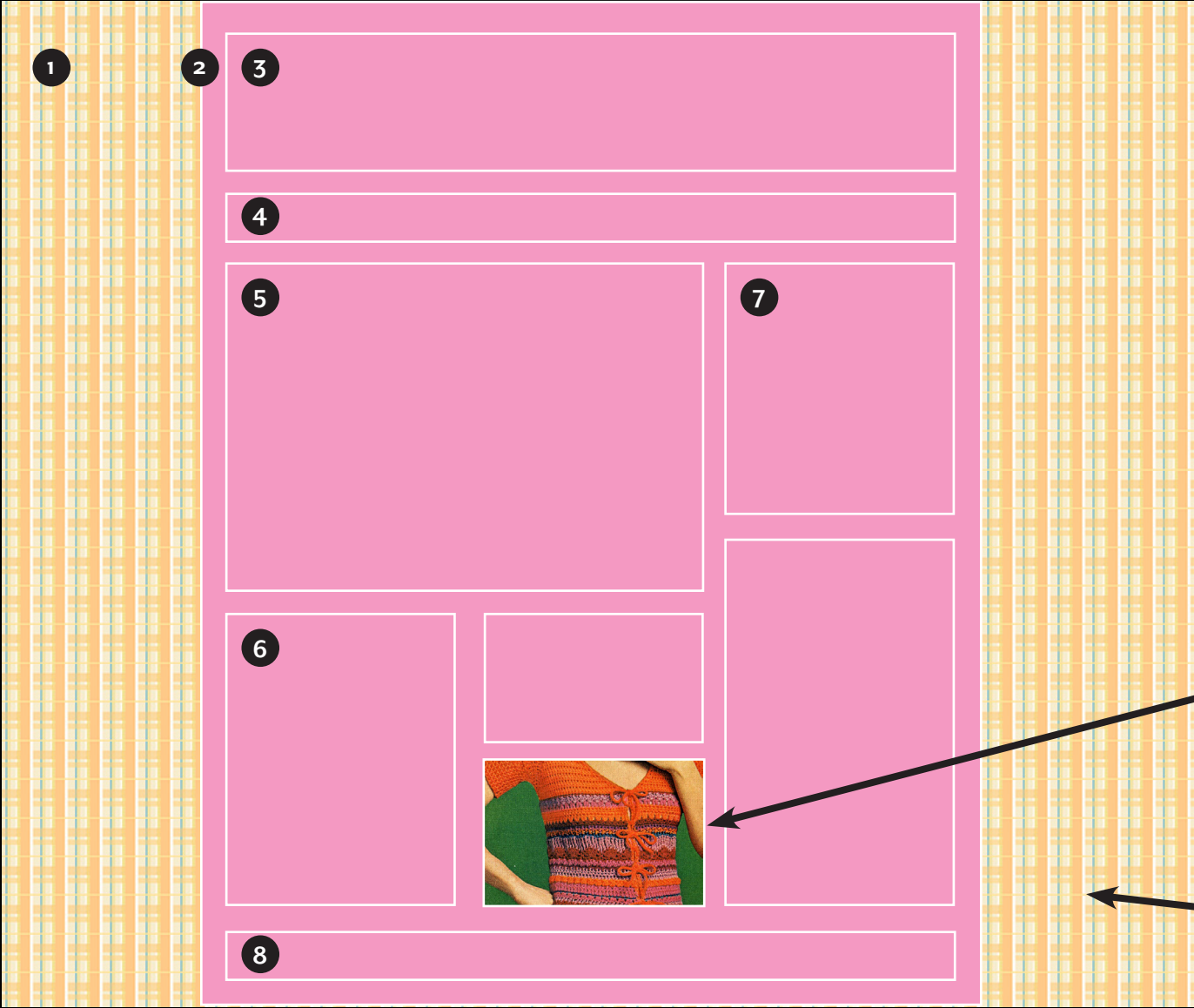
Jen Simmons: The Experimental Layout Lab



Chris Coyier: css-Tricks

Images

ANATOMY OF A WEB PAGE: IMAGE DISTINCTIONS



1: body background

2: wrapper

3: header

4: navigation

5: feature

6: content

7: sidebar

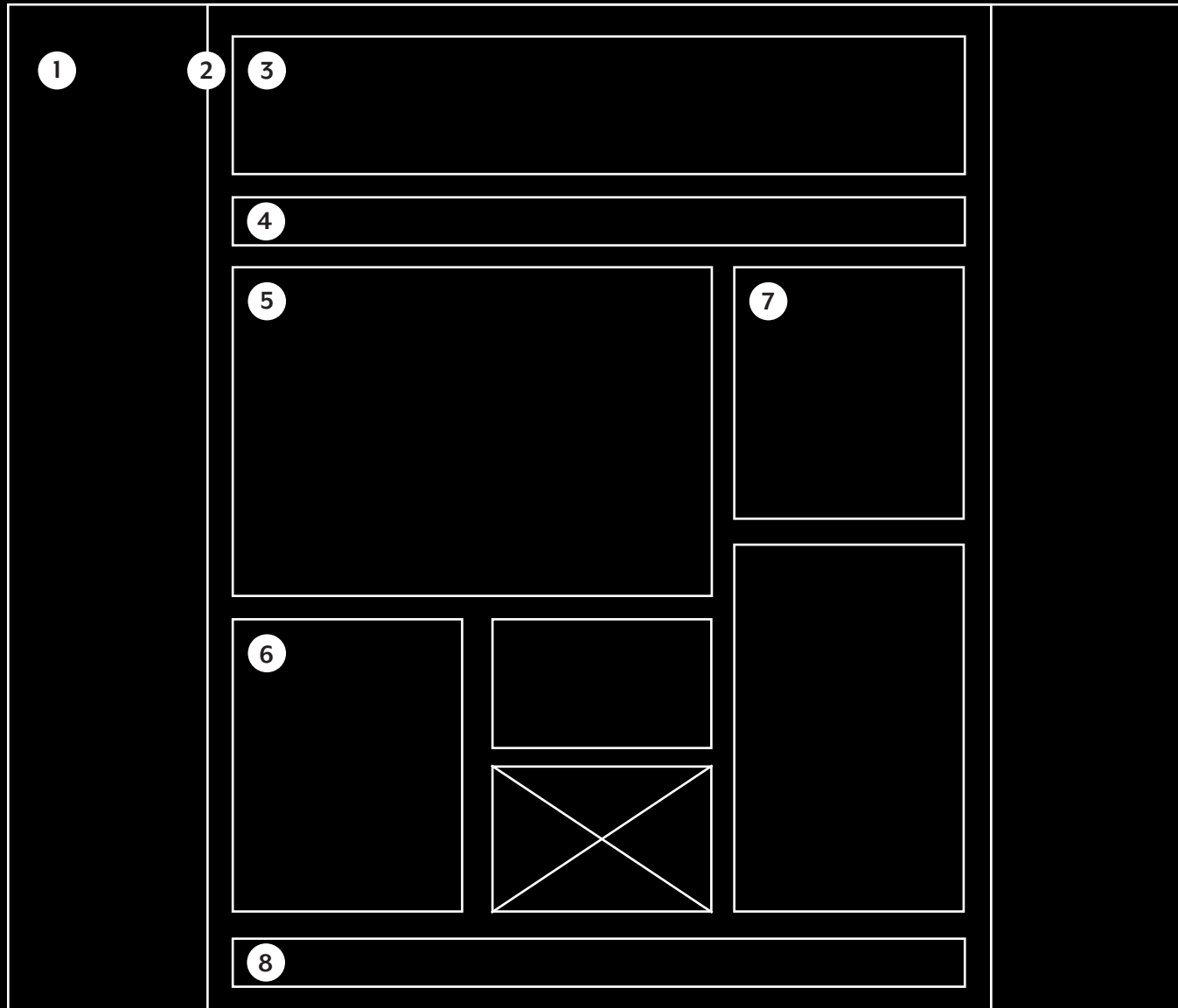
8: footer

• Content images
go into HTML.

• Presentational images
go into CSS.



ANATOMY OF A WEB PAGE: IMAGE DISTINCTIONS



1: body (browser space)

2: container/wrapper

3: header

4: navigation

5: feature

6: content

7: sidebar

8: footer

** All of these (and all other html elements) can have background images!*

CONTENT IMAGES:
HTML SYNTAX

image tag includes element, source, w+h and alt text

*image
element*

``

tag just ends with

`/>`



NON-SEARCHABLE
IMAGES: USE CSS

HTML:

```
<div class="sidebar">  
  <p> Lorem ipsum dolor sit amet,  
  sectetur adipiscing elit. Donec  
  elementum arcu ac egestas rhoncus.  
  Vivamus a quam. Pellentesque  
  erat eu faucibus hendrerit...</p>  
</div>
```

CSS:

```
.sidebar {  
  background-image: bkgrd_stripe.jpg;  
  background-repeat: repeat;  
  padding: 7em;  
}
```



**Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Donec elementum arcu ac
egestas rhoncus. Vivamus a
tincidunt quam. Pellentesque
venenatis erat eu velit faucibus
hendrerit. Morbi non laoreet
mauris, quis pharetra velit. Nam
in volutpat nisi. Maecenas odio
mi, molestie vitae fringilla at,
suscipit ac est. Donec cursus
diam in porta sagittis.**

HTML IMAGES: USE CSS TO SIZE & PLACE

HTML:

```
<body>
<section>
  <p>  Lorem
    ipsum dolor sit amet, adipi
    scing...</p> </section> </body>
```

CSS:

```
body {
  background-image: bkgrd_stripe.jpg;
  background-repeat: repeat;
  padding: 7em; }
```

```
section img {
  float: left;
  width: 50%;
  height: auto;
  margin-right: 20px; }
```

