a deeper dive into CSS

Building a Website 2018

yesterday we made **section** and **navigation** elements, **accessibly**

we did this using HTML

today we're going to use CSS for

advanced positioning

and we’re going to maintain our

simple, accessible HTML structure

the *background*

early CSS did simple stuff like what we learned last week

- colors
- backgrounds
- sizing
- fonts

as CSS evolved,

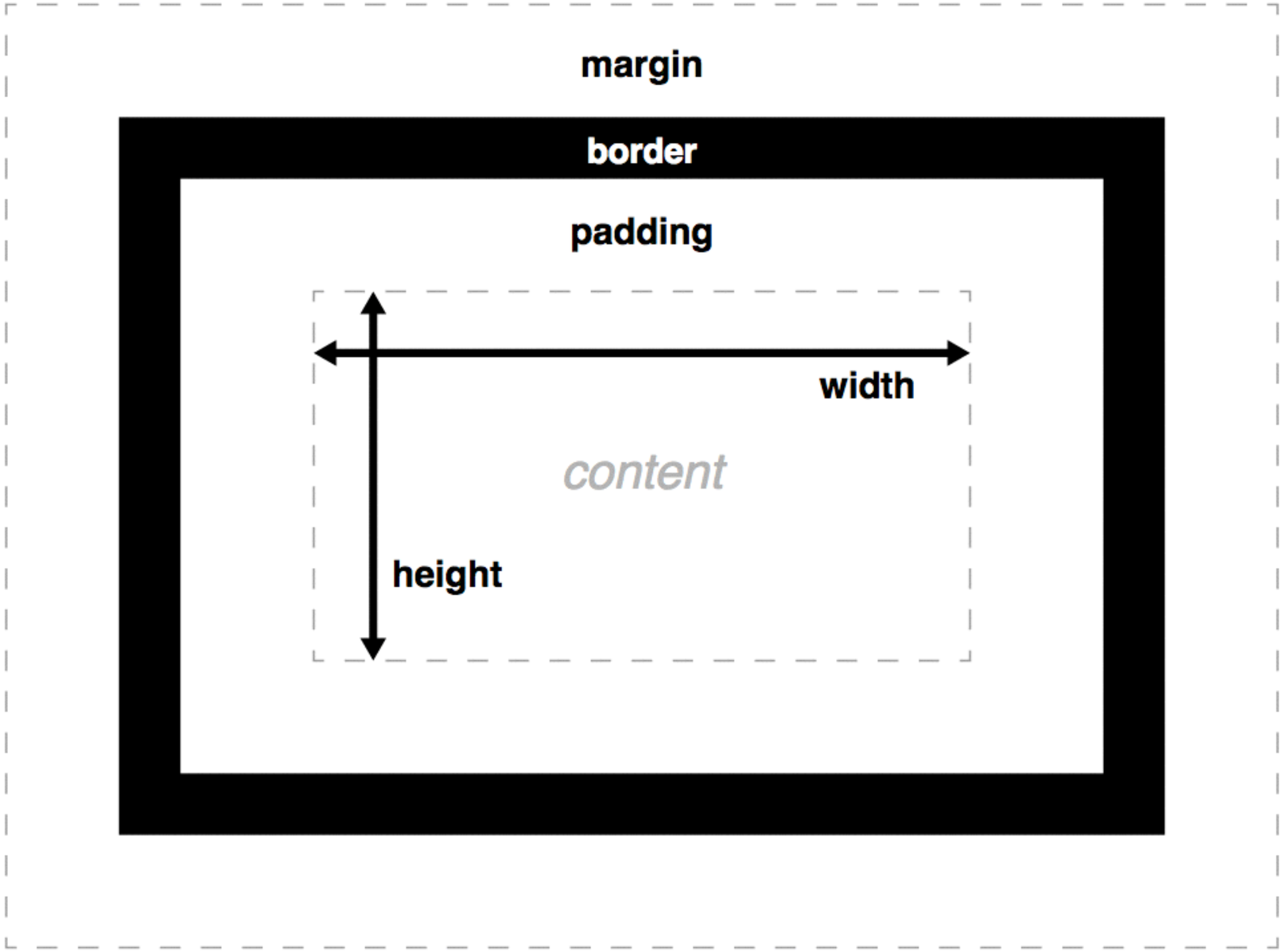
the box model emerged

Box model

Each element is represented as a rectangular box, with the box's content, padding, border, and margin built up around one another like the layers of an onion

developer.mozilla.org

the *conceptual*


margin
border
padding
width
content
height

margin

border

padding

margin

margin

border

border

padding

padding

Welcome to my puppy page

margin

border

padding

Welcome to my puppy page

joyce rice

design et al.

work about blog contact

img | 375 × 399.77

Hi, I'm Joyce!

I'm an experience designer with a focus in journalism and education. I make comics, games, and other digital products to help folks tell better stories.

I cofounded Symbolia, was a Fellow at American University, and I speak internationally about media innovation.

If you'd like to work together, talk about tabletop games, or make jam comics, holler at me on twitter or drop me a line.

if your pages look messed up,
check the box!

the *practical*

Accessible puppies

A site about accessibility and puppies

Accessibility

Keeping your work accessible

One of the primary ways we make our pages accessible is by ordering our h-tags correctly. Another is by keeping our page generally flowing top to bottom

Keeping your puppies, puppies

Check out this cute puppy, maybe he is a helper:

CSS positioning

position

flexbox

classes

position

property governing **where an element sits** on an HTML page

position

```
relative
absolute
fixed
```

position

```
position: relative;
top: 0;
bottom: 0;
left: 0;
right: 0;
```

position

- A **relatively positioned element** is an element whose computed `position` value is `relative`. The `top` and `bottom` properties specify the vertical offset from its normal position; the `left` and `right` properties specify the horizontal offset.

developer.mozilla.org

position

- An **absolutely positioned element** is an element whose computed `position` value is `absolute` or `fixed`. The `top`, `right`, `bottom`, and `left` properties specify offsets from the edges of the element's containing block. (The containing block is the

developer.mozilla.org

position

- A **stickily positioned element** is an element whose computed `position` value is `sticky`. It's treated as relatively positioned until its containing block crosses a specified threshold

developer.mozilla.org

flexbox

an efficient way to **lay out**, **align** and **distribute space** among items in a container

flexbox

flexbox is incredibly useful for building

simple, responsive pages

flexbox

flexbox is a **module** rather than a single property

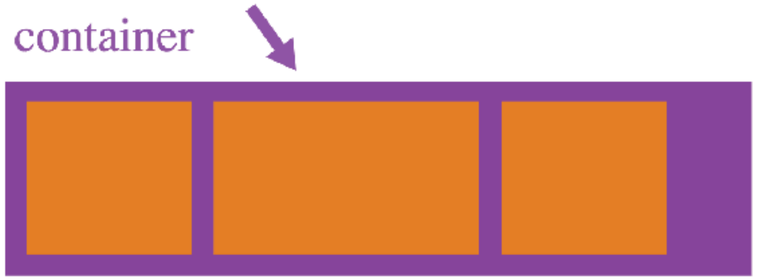
flexbox

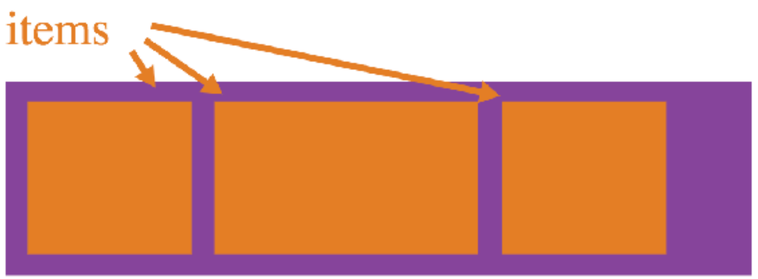
```
display: flex;
flex-wrap: wrap;
justify content: center;
```

A Complete Guide to Flexbox

BY CHRIS COYIER LAST UPDATED ON APRIL 23, 2018

FLEXBOX, LAYOUT





Properties for the Parent

(flex container)

Properties for the Children

(flex items)

classes

a way to **apply CSS rules** to several specific elements, **even if they are different types**

classes

```
.puppyStuff {
    width: 40%;
    background color: purple;
    display: block;
    margin: auto;
}
```

```
<h2 class="puppyStuff">Keeping your puppies, puppies</h2>
<p class="puppyStuff">Check out this cute puppy, maybe he is
a helper:</p>
<img src="doby.jpg"
    alt="a blue doberman puppy"
    class="puppyStuff">
<p>You can't actually keep them pupppies, i lied, i'm sorry</p
>
```

classes

class names

MUST start with a letter!

classes

otherwise, only **letters**,
numbers, **-**, and _ are allowed

Q’s?

let’s make our own!

homework

- Write the CSS to make your city website match your wireframes.

- Deploy to GitHub pages and request my review.

Due tomorrow.