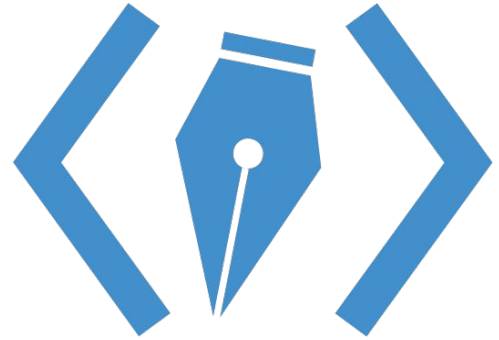


Week 03

Box Model and Spacing



Announcements

Homework 3

Will be released after lecture (finstas!)

Figma Lab on Thursday

Highly recommended to come to section! We'll show you how to use Figma!

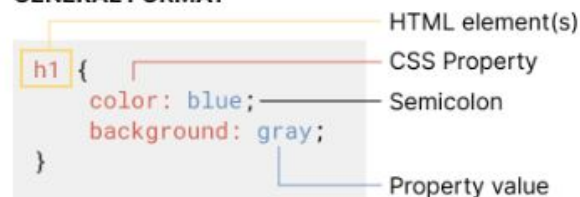
Important for project (released next week)!

Give us anonymous feedback at wdd.io/go/feedback

Add songs to our class playlist: wdd.io/go/spotify

CSS CHEAT SHEET

GENERAL FORMAT



IMPORTANT + USEFUL SELECTORS

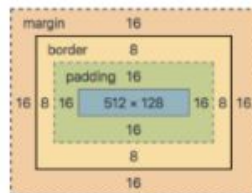
.classname	Class selector (many elements)
#idname	ID selector (unique element)
:first-child	First object within the selected
:hover	Format when cursor is hovering over the selected
:active	Format when selected is clicked/activated
:clicked	Format when selected is clicked/activated
sel1 sel2	Selects all of type sel2 within sel1
:visited	Format after selected has been clicked on
:before	Slightly more advanced selector, generate content around the selected element
:after	Slightly more advanced selector, generate content around the selected element

LINKING YOUR CSS IN HTML

Put this inside your <head> tags in HTML:

```
<link rel="stylesheet" type="text/css" href="file path + file name" />
```

FORMATTING BASICS



Color-coded example:



display: block;
everything in its own line



display: inline;
everything on the same line, ignores margin/padding



display: inline-block;
everything on the same line with margin/padding



float: left;



float: right;



REMEMBER: Separate each element with a comma to select multiple elements!

CSS Selectors

Box Model

Intro to Layout

CSS Selectors

Box Model

Intro to Layout

HTML Reminders

- `<head>` is a special part of the HTML document
 - Think of it like the title/copyright page at the front of a book: not actual content, but important for the production of the book!
- `<body>` is where all your actual content goes! (like the chapters of the book)
- `<header>` is a tag for your content, similar to `<div>` but with a more descriptive and meaningful name.

CSS Reminders

- **color** refers to text color, whereas **background** or **background-color** changes background color

- for changing fonts, you can either do it separately:

```
p {  
    font-size: 18px;  
    font-family: Arial;  
}
```

- or together:

```
p {  
    font: 18px Arial;  
}
```

- **semicolon at the end of every line!!!**

- for hover effects:

```
a:hover {  
    color: blue;  
}
```

Selectors Examples - <https://jsbin.com/suzogoyido/edit?html,css,output>

index.html

```
<h1 class="highlight">Food Places</h1>

<ul id="food-list">
  <li>Mezzo</li>
  <li>Pho K&K</li>
  <li class="highlight">Taco Bell</li>
</ul>

<ul id="cafe-list">
  <li>Yali's</li>
  <li class="highlight">Asha Tea House</li>
  <li>Brewed Awakening</li>
</ul>
```

style.css

```
h1, li {
  font-family: Arial;
}

#food-list li {
  color: blue;
}

ul .highlight {
  background-color: yellow;
}

#cafe-list li {
  color: green;
}

Web Design DeCal Fall 2020
```


Select **multiple things** by separating with **commas**

index.html

```
<h1 class="highlight">Food Places</h1>

<ul id="food-list">
  <li>Mezzo</li>
  <li>Pho K&K</li>
  <li class="highlight">Taco Bell</li>
</ul>

<ul id="cafe-list">
  <li>Yali's</li>
  <li class="highlight">Asha Tea House</li>
  <li>Brewed Awakening</li>
</ul>
```

style.css

```
h1, li {
  font-family: Arial;
}

#food-list li {
  color: blue;
}

ul .highlight {
  background-color: yellow;
}

#cafe-list li {
  color: green;
}
```

Web Design DeCal Fall 2020

Select only the `` inside `#food-list`

index.html

```
<h1 class="highlight">Food Places</h1>

<ul id="food-list">
  <li>Mezzo</li>
  <li>Pho K&K</li>
  <li class="highlight">Taco Bell</li>
</ul>

<ul id="cafe-list">
  <li>Yali's</li>
  <li class="highlight">Asha Tea House</li>
  <li>Brewed Awakening</li>
</ul>
```

style.css

```
h1, li {
  font-family: Arial;
}

#food-list li {
  color: blue;
}

ul .highlight {
  background-color: yellow;
}

#cafe-list li {
  color: green;
}
```

Select only `class="highlight"` inside ``

index.html

```
<h1 class="highlight">Food Places</h1>

<ul id="food-list">
  <li>Mezzo</li>
  <li>Pho K&K</li>
  <li class="highlight">Taco Bell</li>
</ul>

<ul id="cafe-list">
  <li>Yali's</li>
  <li class="highlight">Asha Tea House</li>
  <li>Brewed Awakening</li>
</ul>
```

style.css

```
h1, li {
  font-family: Arial;
}

#food-list li {
  color: blue;
}

ul .highlight {
  background-color: yellow;
}

#cafe-list li {
  color: green;
}
```

Web Design DeCal Fall 2020

Select only `class="highlight"` inside ``

index.html

```
<h1 class="highlight">Food Places</h1>
```

```
<ul id="food-list">
```

```
<li>Mezzo
```

```
<li>Ph
```

```
<li class="highlight">Bell</li>
```

```
</ul>
```

```
<ul id="cafe-list">
```

```
<li>Yali's</li>
```

```
<li class="highlight">Asha Tea House</li>
```

```
<li>Brewed Awakening</li>
```

```
</ul>
```

Does not
highlight `<h1>`
because it's not
inside ``

style.css

```
h1, li {  
    font-family: Arial;  
}
```

```
#food-list li {  
    color: blue;  
}
```

```
ul .highlight {  
    background-color: yellow;  
}
```

```
#cafe-list li {  
    color: green;  
}
```

Web Design DeCal Fall 2020

CSS Selectors

Box Model

Intro to Layout



Picture frame analogy

content

The picture itself



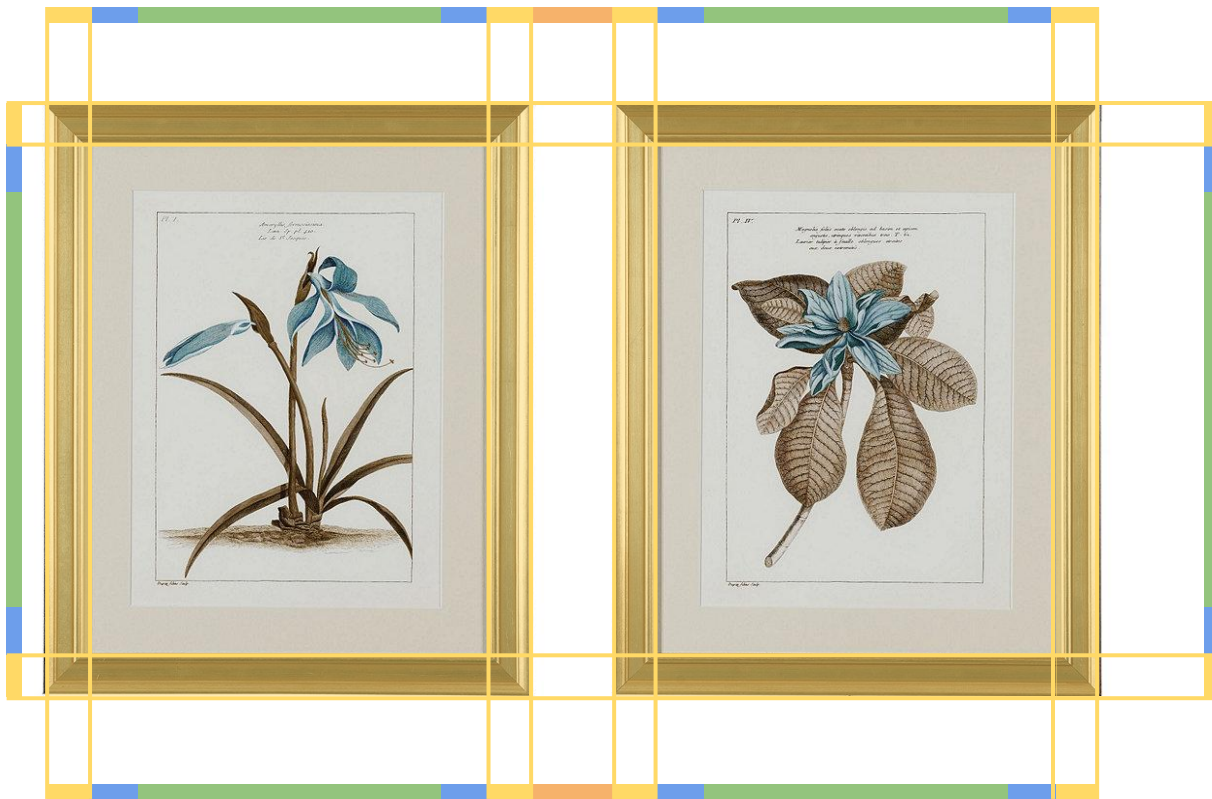
Picture frame analogy

Web Design DeCal Fall 2020



padding

Space between the picture and the picture frame



border

The picture frame

Picture frame analogy

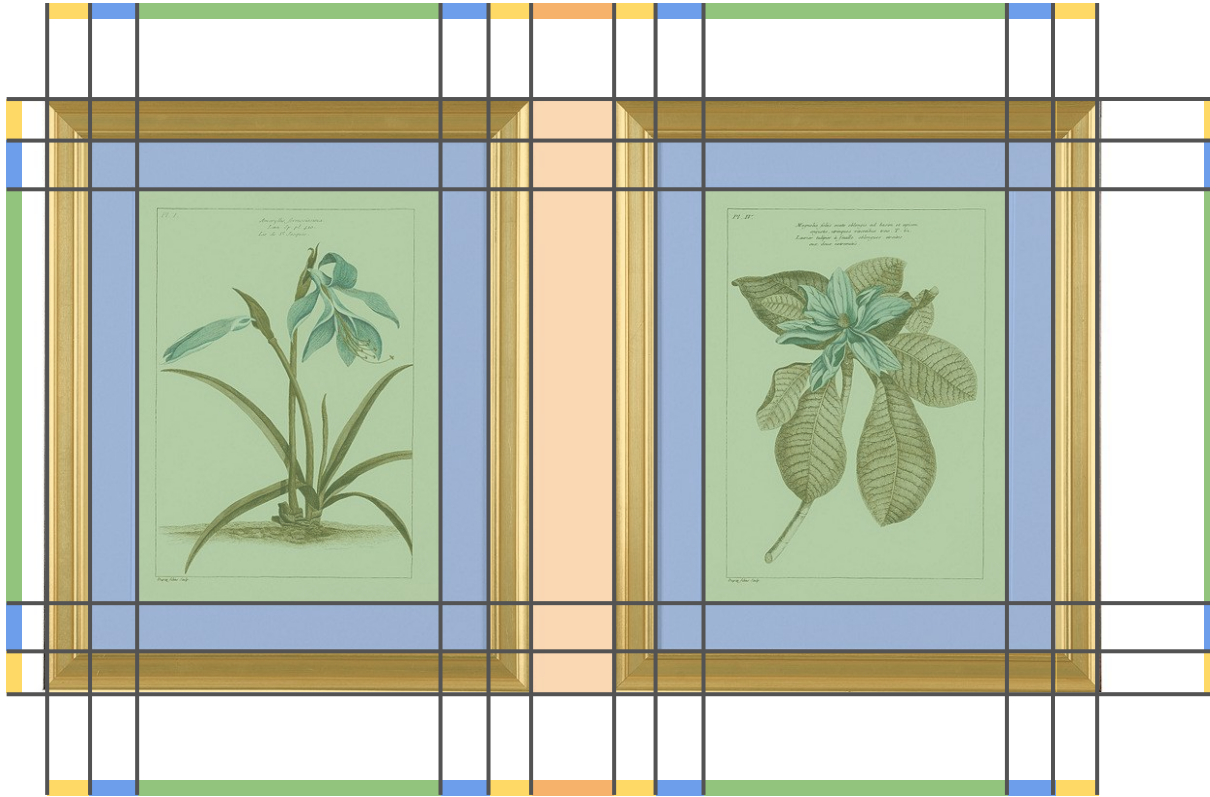
Web Design DeCal Fall 2020



margin

Spacing in between
pictures

margin



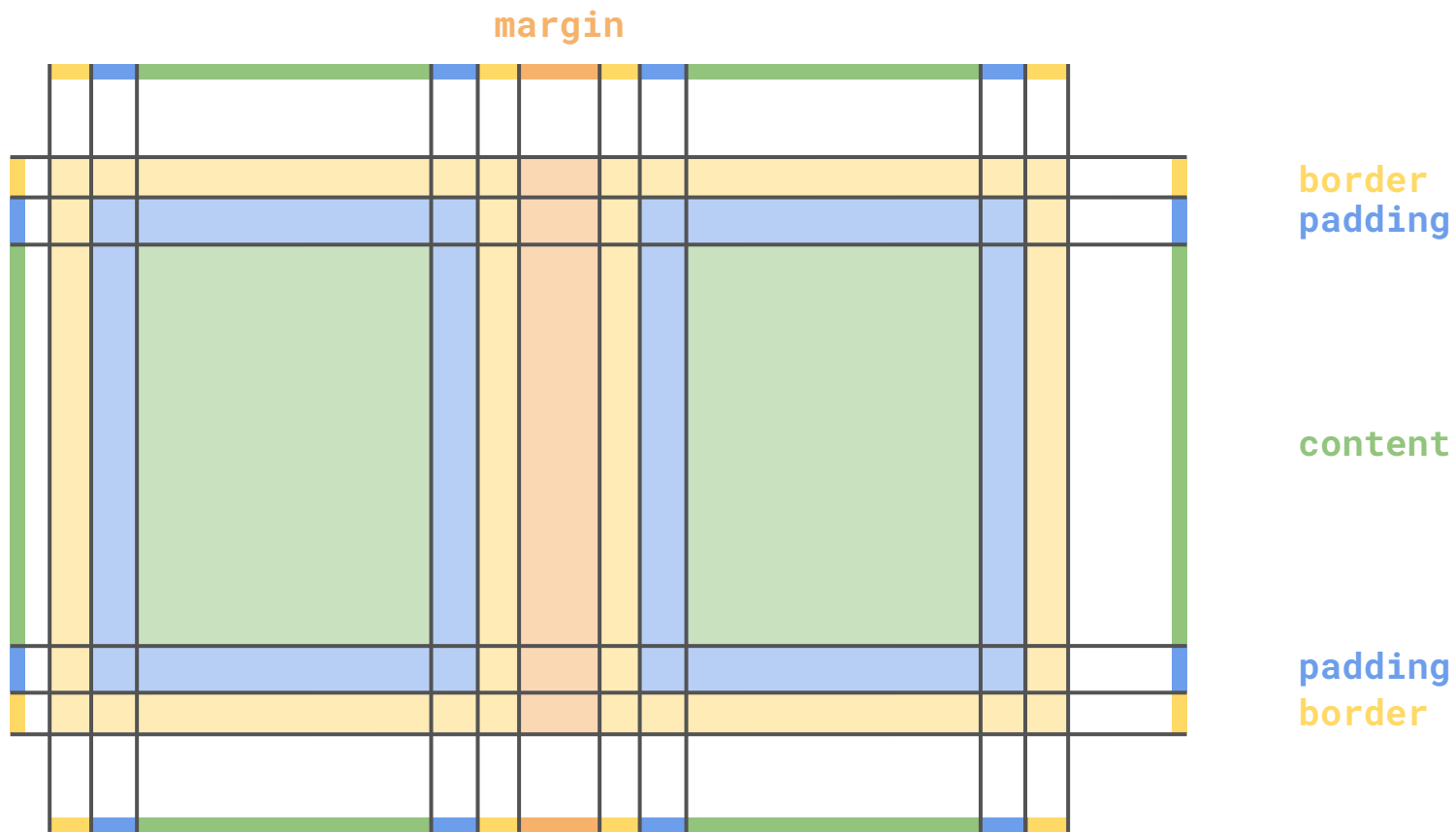
border
padding

content

padding
border

Picture frame analogy, all highlighted

Web Design DeCal Fall 2020



Picture frame analogy, now without the picture frame :o

Box model

content

The picture itself

padding

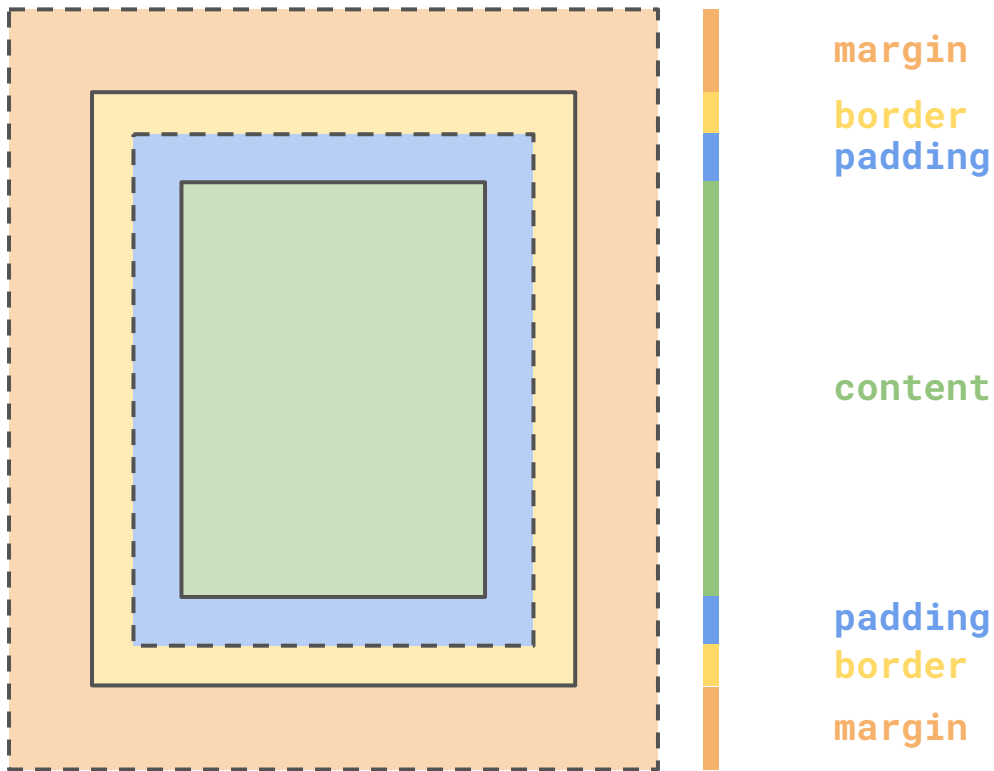
Space between the picture and the picture frame

border

The picture frame

margin

Spacing in between pictures



<https://jsbin.com/yekuhalawu/edit?html,css,output>

Welcome to WDD!
Homework 3 will be released tonight.

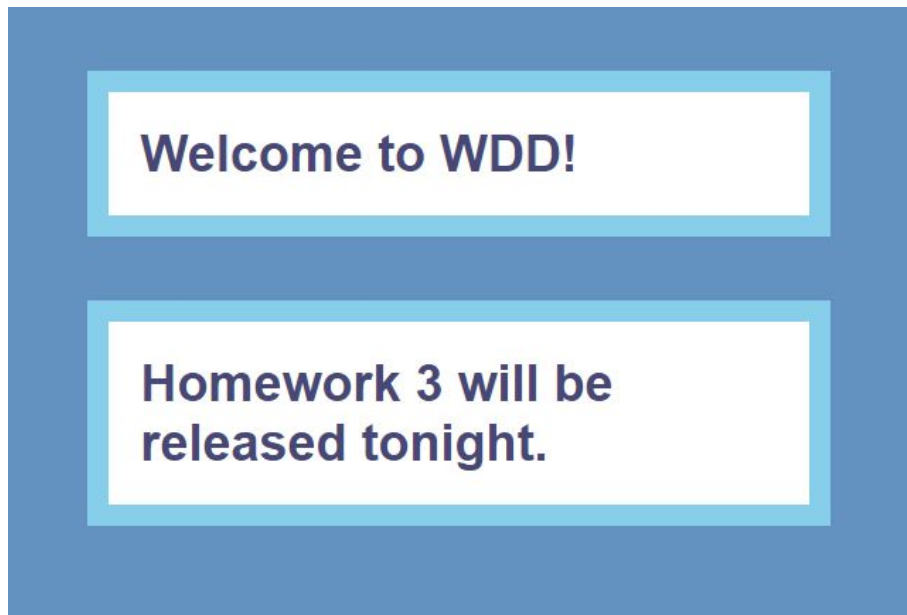
- No spacing
- No visual hierarchy
- Low-contrast colors → hard to read

Design Issues

```
<div class="box">  
  Welcome to WDD!  
</div>  
  
<div class="box">  
  Homework 3 will be released  
  tonight.  
</div>
```

HTML

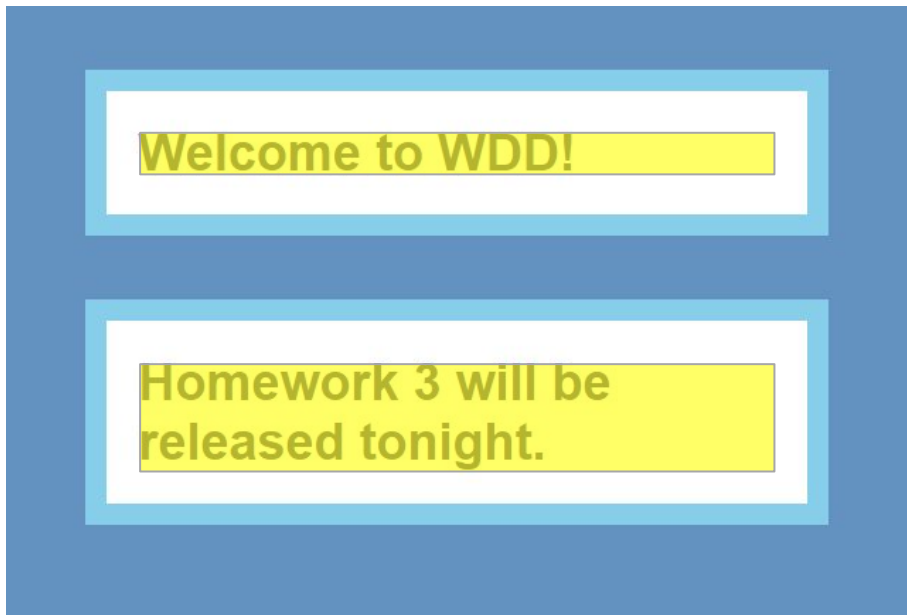
Identify the content box, padding, border, and margin



```
.box {  
    background-color: white;  
    width: 300px;  
    padding: 15px;  
    border: 10px solid skyblue;  
    padding: 30px;  
}
```

CSS

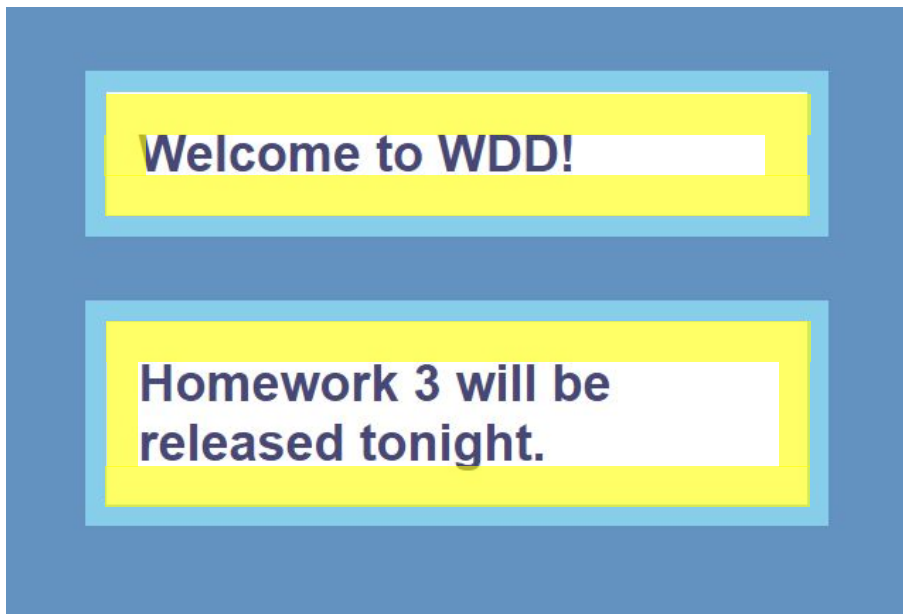
Identify the **content box**, padding, border, and margin



```
.box {  
    background-color: white;  
    width: 300px;  
    padding: 15px;  
    border: 10px solid skyblue;  
    margin: 30px;  
}
```

CSS

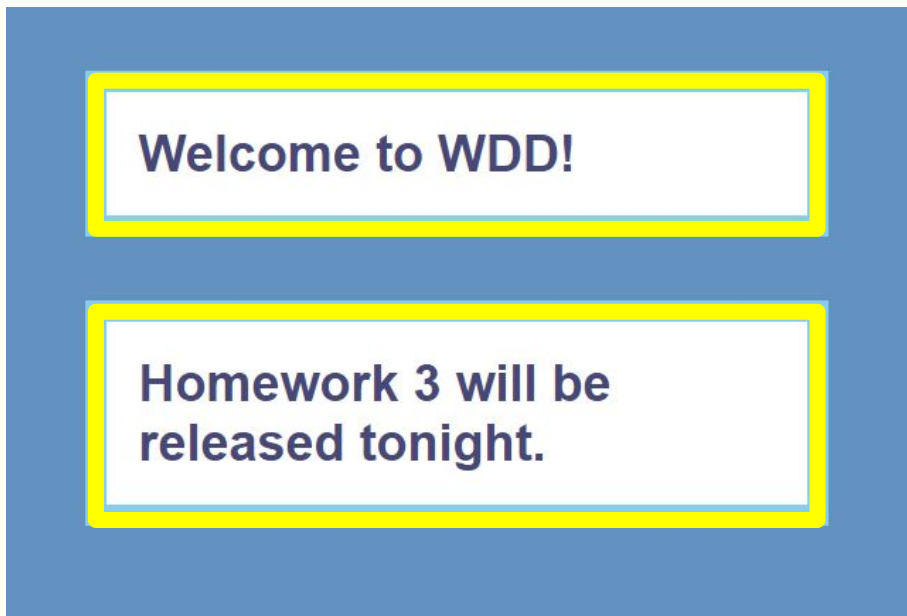
Identify the content box, padding, border, and margin



```
.box {  
    background-color: white;  
    width: 300px;  
    padding: 15px;  
    border: 10px solid skyblue;  
    margin: 30px;  
}
```

CSS

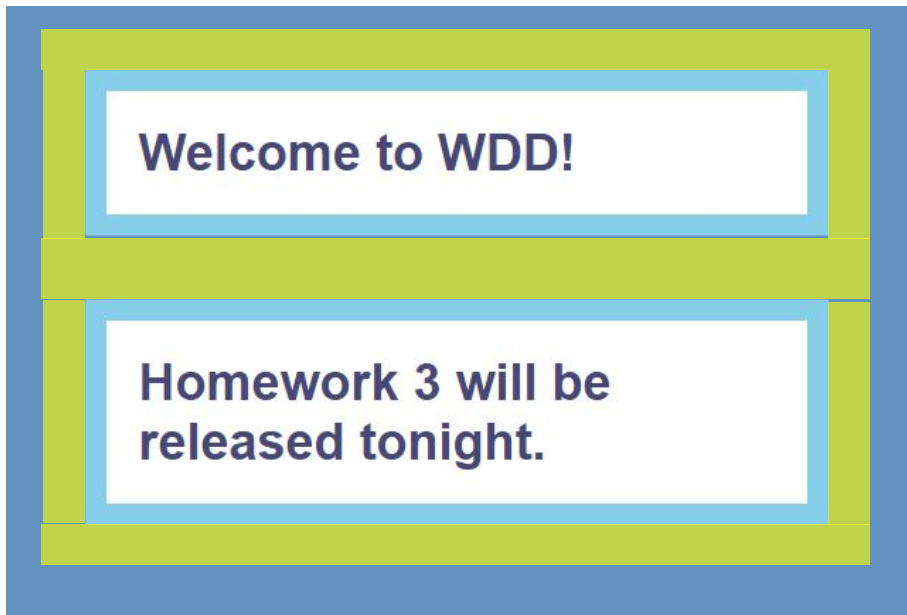
Identify the content box, padding, **border**, and margin



```
.box {  
    background-color: white;  
    width: 300px;  
    padding: 15px;  
    border: 10px solid skyblue;  
    margin: 30px;  
}
```

CSS

Identify the content box, padding, border, and margin



```
.box {  
  background-color: white;  
  width: 300px;  
  padding: 15px;  
  border: 10px solid skyblue;  
  margin: 30px;  
}
```

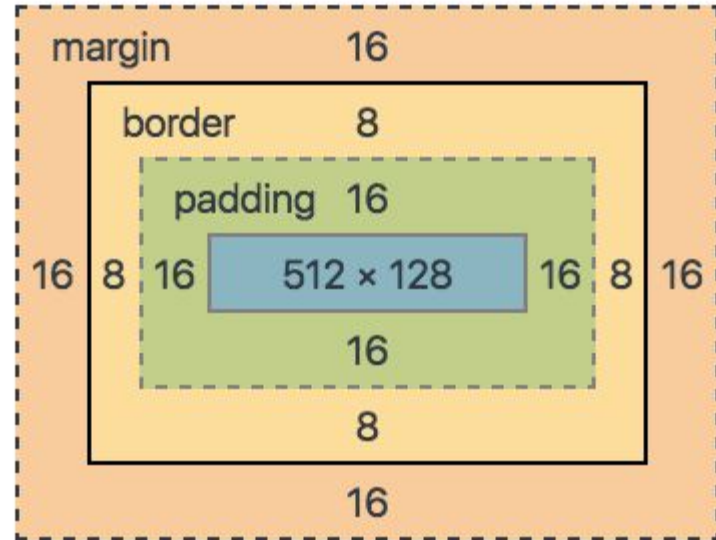
CSS

We can have different amounts of spacing on different sides of the box! The four sides are: **top**, **right**, **bottom**, **left**

```
width: 512px;  
height: 128px;
```

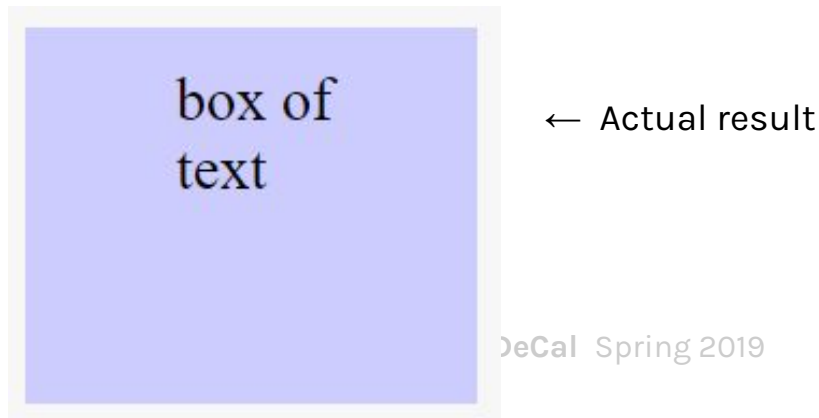
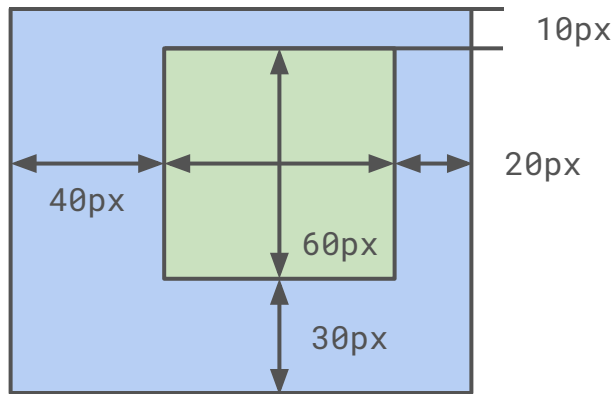
```
padding-top: 16px;  
padding-right: 16px;  
padding-bottom: 16px;  
padding-left: 16px;
```

```
margin-top: 16px;  
margin-right: 16px;  
margin-bottom: 16px;  
margin-left: 16px;
```



Example of different padding on different sides

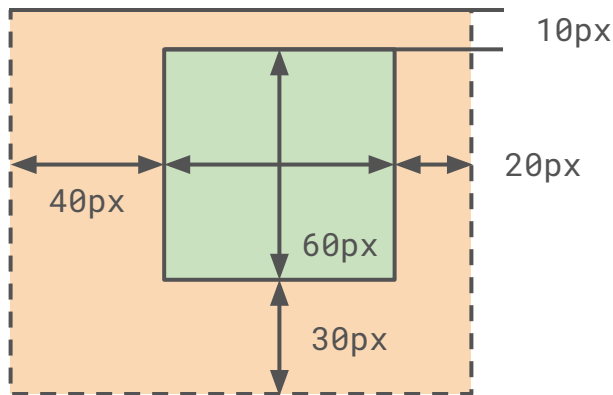
```
#box {  
  width: 60px;  
  height: 60px;  
  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 40px;  
  /* or more neatly... */  
  padding: 10px 20px 30px 40px;  
}
```



You can do the same thing for **margin**!

Specifying margins

```
#box {  
  width: 60px;  
  height: 60px;  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
  /* or more neatly... */  
  margin: 10px 20px 30px 40px;  
}
```



Units

- **Pixels:** the tiny squares that make up your screen!
 - 100px is *approx* an inch, or a little less than an inch
 - Good for cases when you know the exact size of something, and you know that the size does not need to scale

Units

- **Percents:** Depends on size of the parent HTML element
 - Helpful to an extent for responsive design

```
<div id="outer">  
  <div id="inner"></div>  
</div>
```

```
#outer {  
  width: 50%;  
  height: 200px;  
}
```

```
#inner {  
  width: 50%;  
  height: 50%;  
}
```

Result: width is 50% of the page width (**<body>** has 100% width unless you specify otherwise)

Result: width is 25% of the page width, and height is 100px.

Units for typography

- **Points:** (like, 12pt font) physically, it's 1/72 of an inch.
 - But can be inconsistent, so stick with pixels or em.

CSS Selectors

Box Model

Intro to Layout

Example: Design

Header

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec non eros ex. Sed varius velit sit amet volutpat pharetra. Cras tincidunt justo diam, at dictum est pretium in. Sed in facilisis urna, vitae blandit ligula. Curabitur tempor faucibus sapien ut vulputate. Suspendisse sed sollicitudin tellus. Ut nunc risus, aliquam ut congue non, venenatis nec leo. Integer porttitor sagittis vestibulum. Proin tincidunt viverra orci in interdum. Vivamus eget tristique ligula. Mauris vitae vestibulum ante.

Example: Thought Process 🤔



We want the main, white content container to always be centered on the page and 100px from the top.

Should the container be a set width, like 600px? Or a percent, like 50%?

→ A pixel width. Something like 50% can be pretty narrow at a smaller screen size.

Example: Code ✨

What we need to make:

white content container

- centered
- 100px from the top
- 600px width
- has heading and text



Thing to remember:

If we set a **width**, then we can use **margin: auto** to center the item!

HTML

```
<div id="container">
  <h1>Header</h1>
  <p>Lorem ipsum filler text</p>
</div>
```

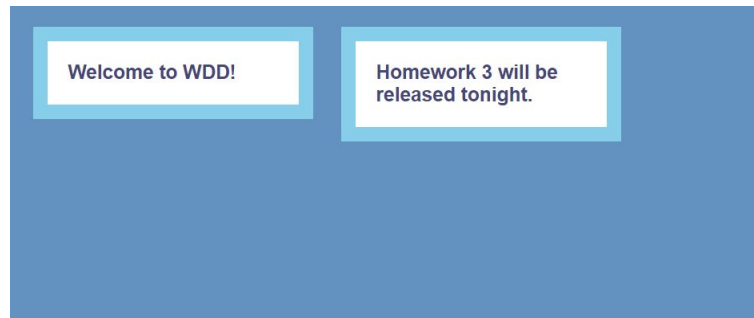
style.css

```
#container {
  width: 600px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
}
```

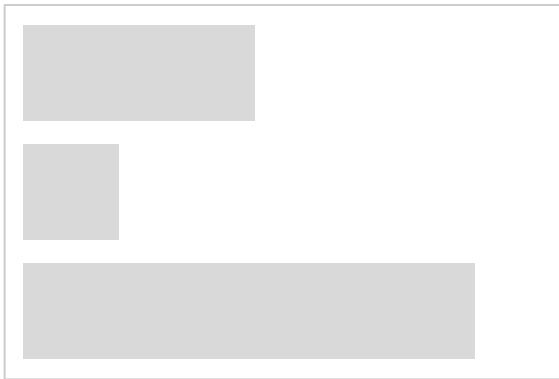
How do we put stuff next to each other?

We've noticed how **by default**, a lot of html elements like `<div>` will span **as wide as it can** across the page, and each new div goes **on its own line**.

Sometimes we want stuff full-width.
But sometimes we want to create layouts where boxes can be **side-by-side**, just like on this slide!



CSS display

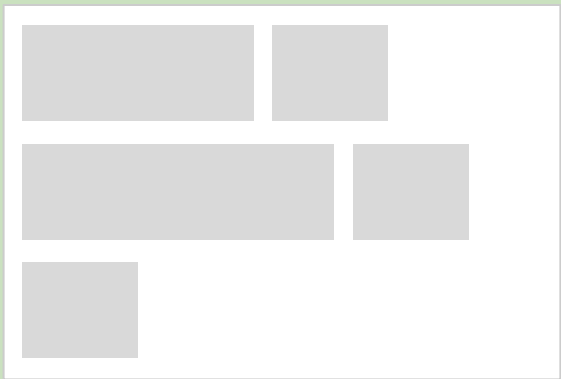


display: block;

Everything goes on its own line, no matter the width!

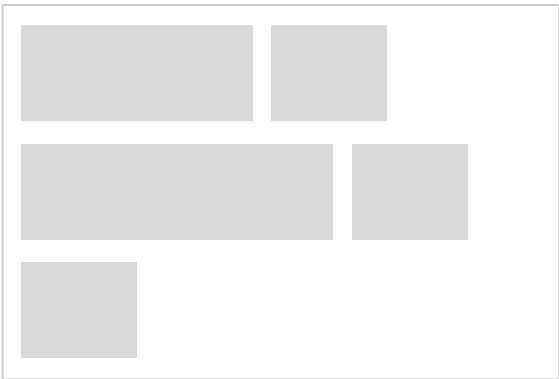
Can use **margin** and **padding**

Examples: <div>, <h1>, <p>



display: inline-block;

A happy medium. Stuff goes on the same line **and** uses **margin/padding** !!



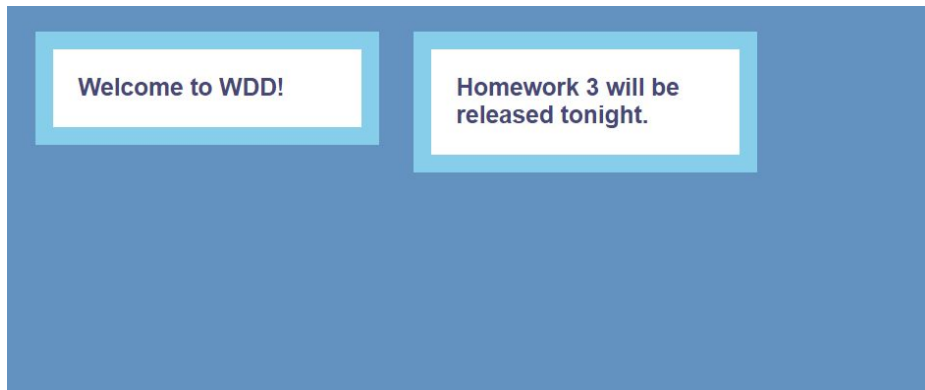
display: inline;

Everything on the same line, only as long as the text inside it is.

Does not listen to margin/padding

Examples: , <a>, text

Use **display** and **float**



```
.box {  
  display: inline-block;  
  float: left;  
  
  background-color: white;  
  width: 300px;  
  padding: 15px;  
  border: 10px solid skyblue;  
  padding: 30px;  
}
```

CSS

Berkeley Food Guide

[Home](#)

[Restaurants](#)

[Coffee Shops](#)

This site features food places in Berkeley that WDD students recommend!

Restaurants

UC Berkeley has a large variety of restaurants nearby. Check out favorites on southside, northside, and downtown.

Coffee Shops

There are also several coffee shops and cafés that have outlets and free wifi, ideal for studying!

Berkeley Food Guide

[Home](#)

[Restaurants](#)

[Coffee Shops](#)

This site features food places in Berkeley that WDD students recommend!

Restaurants

UC Berkeley has a large variety of restaurants nearby. Check out favorites on southside, northside, and downtown.

Coffee Shops

There are also several coffee shops and cafés that have outlets and free wifi, ideal for studying!

HTML skeleton with content - **notice how the layout is grouped by the HTML tags**

Berkeley Food Guide

[Home](#)[Restaurants](#)[Coffee Shops](#)

This site features food places in Berkeley that WDD students recommend!

Restaurants

UC Berkeley has a large variety of restaurants nearby. Check out favorites on southside, northside, and downtown.

Coffee Shops

There are also several coffee shops and cafés that have outlets and free wifi, ideal for studying!

Now we add the CSS from this lecture to create the spacing and layout that we want.

body

div

header

h1

nav

ul

li

[a](#)

li

[a](#)

li

[a](#)

div

div

h2

p

div

h2

p

Here's the bare HTML skeleton

Berkeley Food Guide

[Home](#)

[Restaurants](#)

[Coffee Shops](#)

This site features food places in Berkeley that WDD students recommend!

Restaurants

UC Berkeley has a large variety of restaurants nearby. Check out favorites on southside, northside, and downtown.

Coffee Shops

There are also several coffee shops and cafés that have outlets and free wifi, ideal for studying!

Chrome DevTools

Chrome DevTools

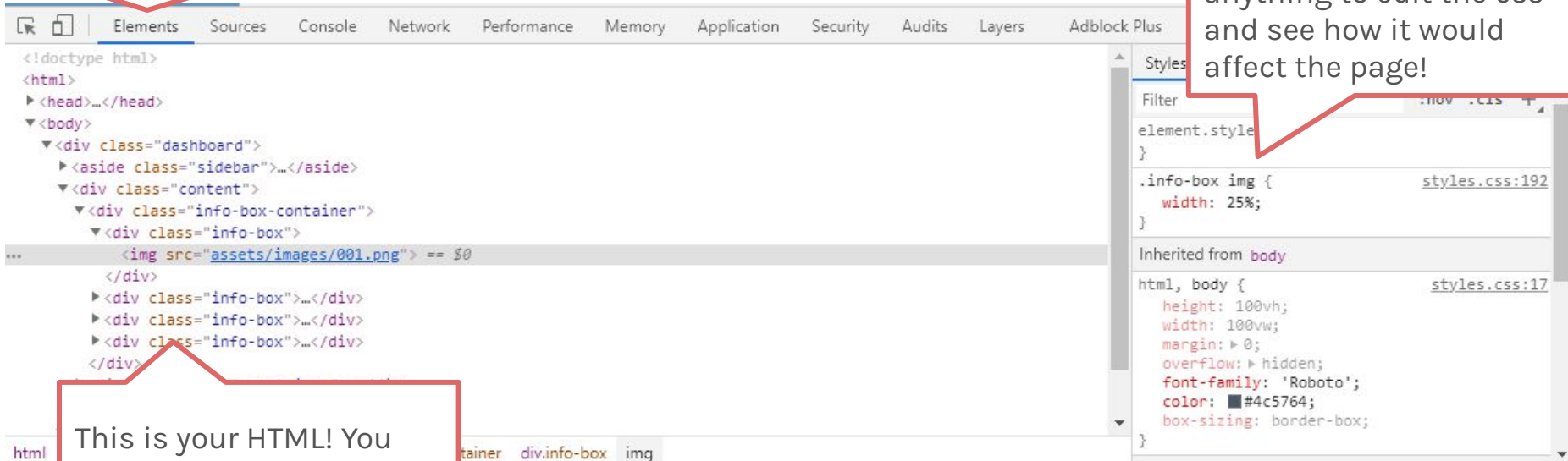
- **Best way** to debug & figure out what's going on with your code!
- *Best in Chrome*, decent in Firefox, iffy in other browsers
- Ways to get to it:
 - Right-click → Inspect Element
 - ctrl/cmd + shift + i

Chrome DevTools Demo

For HTML and CSS, make sure you're on the Elements tab.

This is the CSS for the HTML element you clicked on from the left panel.

You can double click on anything to edit the css and see how it would affect the page!



This is your HTML! You can click on the arrows on the left to hide or expand the elements.

Chrome DevTools - some things you can do!

- Hover over an HTML tag → it will show you the box model diagram of the element (its width, height, padding, margin)
- Click on an HTML tag → see what CSS rules are affecting it
- Click on a CSS property or value → edit the CSS and see the changes happen!
- Click in the CSS rules area → add CSS to the specific HTML tag you're on and see the changes happen!

(if time) How to Think About Layouts

Thinking of a page as building blocks put together

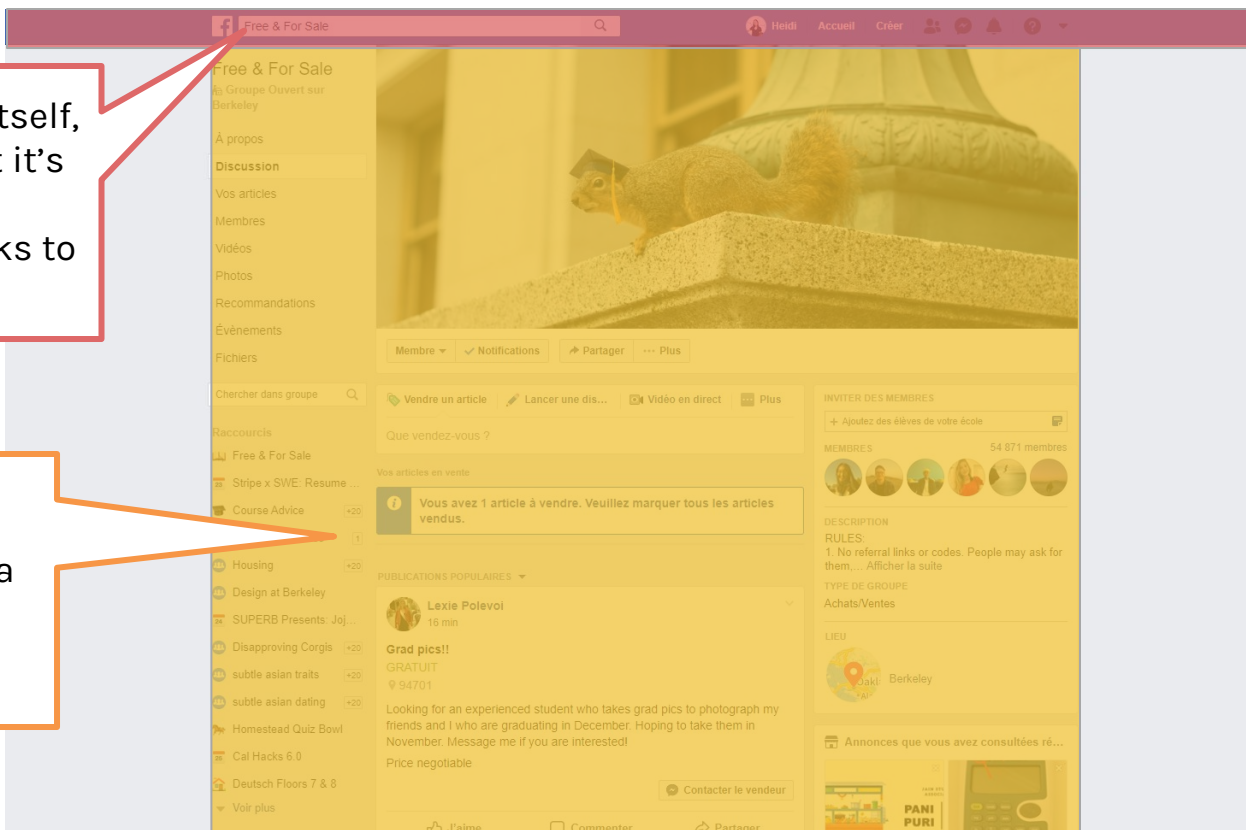
The screenshot shows a Facebook group page for "Free & For Sale" in Berkeley. The page is structured into several key sections:

- Top Navigation Bar:** Includes the Facebook logo, the group name "Free & For Sale", a search bar, and user profile links for Heidi, Accueil, and Créer.
- Left Sidebar:** Contains navigation links for "À propos", "Discussion" (highlighted), "Vos articles", "Membres", "Vidéos", "Photos", "Recommandations", "Événements", and "Fichiers". Below these are search filters and a list of "Raccourcis" (shortcuts) for various categories like "Free & For Sale", "Stripe x SWE: Resume...", "Course Advice", "Computer Science", "Housing", "Design at Berkeley", "SUPERB Presents: Jo...", "Disapproving Corgis", "subtle asian traits", "subtle asian dating", "Homestead Quiz Bowl", "Cal Hacks 6.0", and "Deutsch Floors 7 & 8".
- Main Content Area:** Features a large photo of a squirrel wearing a graduation cap. Below the photo are buttons for "Membre", "Notifications", "Partager", and "Plus". A section titled "Vendre un article" (Sell an item) includes a search bar and a list of "Vos articles en vente" (Your items for sale). A post by Lexie Polevoi, titled "Grad pics!!", is visible, offering a "GRATUIT" (free) service for photographing graduates.
- Right Sidebar:** Contains group statistics, including "54 871 membres" (54,871 members). It also lists "RÈGLES" (rules) and "TYPE DE GROUPE" (group type) as "Achats/Ventes" (Buy/Sell). A "LIEU" (location) section shows a map of Berkeley.

Thinking of a page as building blocks put together

Header bar by itself,
we can tell that it's
fixed position
because it sticks to
the top

Main content
container, has a
set width,
centered



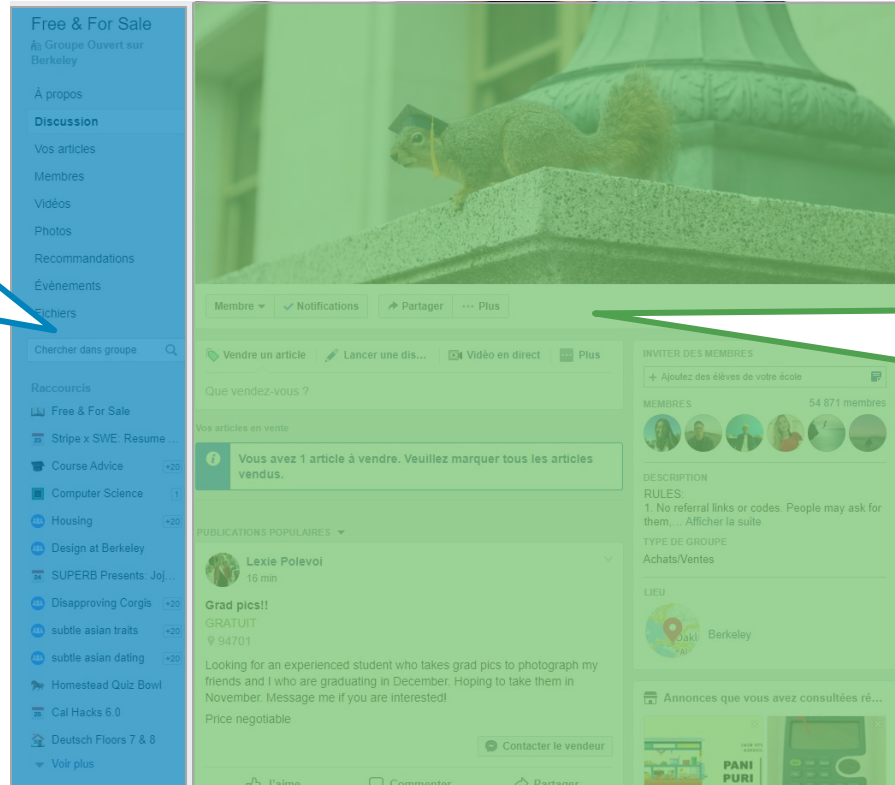
Thinking of a page as building blocks put together

Now let's look
at what's in
the main
content
container

The screenshot shows a Facebook group page for 'Free & For Sale' (Berkeley). The left sidebar contains navigation links: 'À propos', 'Discussion' (highlighted), 'Vos articles', 'Membres', 'Vidéos', 'Photos', 'Recommandations', 'Événements', and 'Fichiers'. Below these is a search bar and a list of shortcuts including 'Free & For Sale', 'Stripe x SWE: Resume...', 'Course Advice', 'Computer Science', 'Housing', 'Design at Berkeley', 'SUPERB Presents: Jo...', 'Disapproving Corgis', 'subtle asian traits', 'subtle asian dating', 'Homestead Quiz Bowl', 'Cal Hacks 6.0', and 'Deutsch Floors 7 & 8'. The main content area features a large photo of a squirrel wearing a graduation cap. Below the photo are buttons for 'Membre', 'Notifications', 'Partager', and 'Plus'. A section titled 'Vendre un article' (Sell an item) includes a text input field 'Que vendez-vous?' and a button 'Vendre un article'. Below this is a post by 'Lexie Polevoi' titled 'Grad pics!! GRATUIT' with a description about taking graduation photos for friends. The right sidebar shows 'INVITER DES MEMBRES' (Add members), 'MEMBRES' (54 871 members), 'DESCRIPTION' (Rules: No referral links or codes), 'TYPE DE GROUPE' (Achat/Ventes), 'LIEU' (Berkeley), and 'Annonces que vous avez consultées...' (Ads you've viewed).

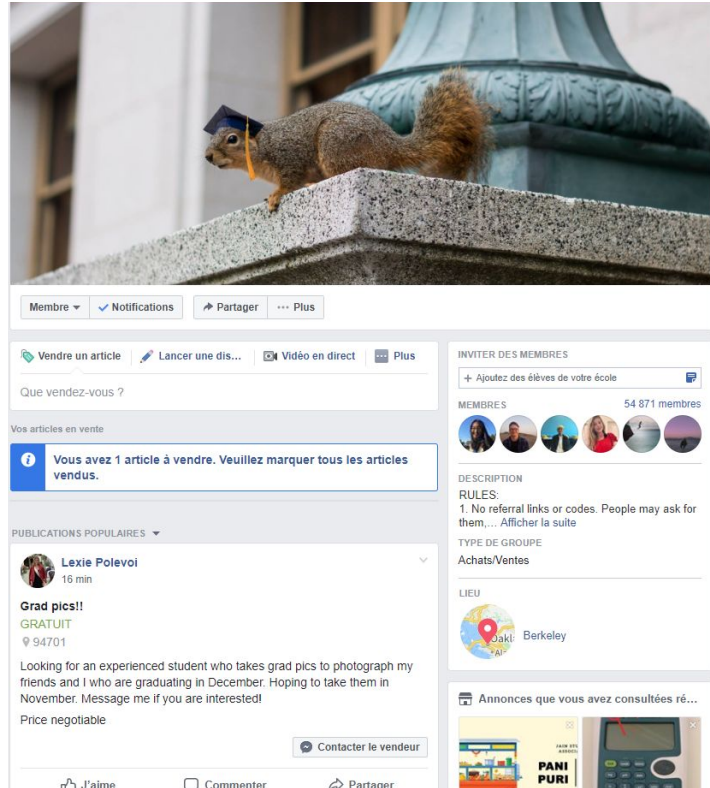
Thinking of a page as building blocks put together

Column on the left. Each section (like the title, first set of links, search bar, second set of links) is its own thing



Right column has a lot of stuff, let's keep breaking this down!

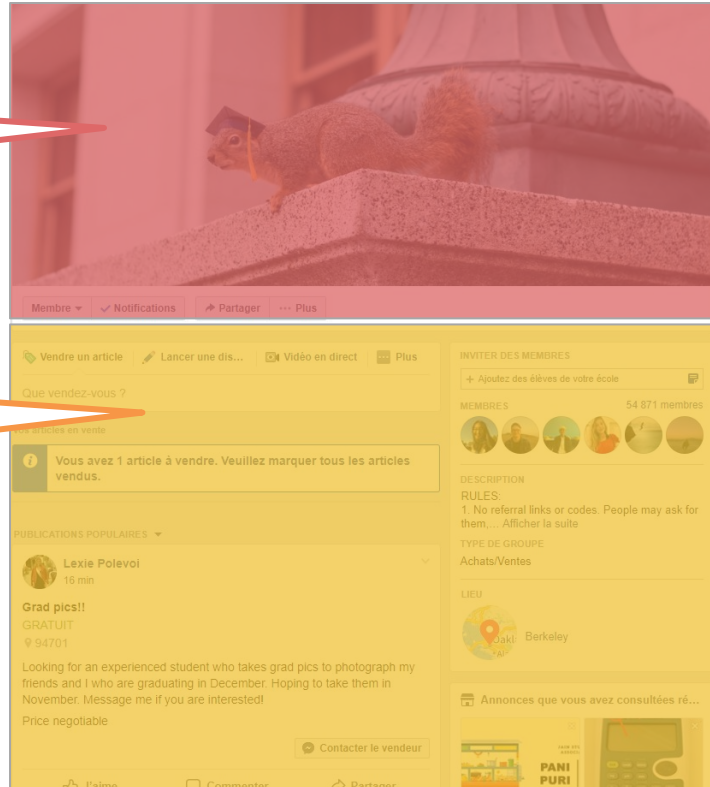
Thinking of a page as building blocks put together



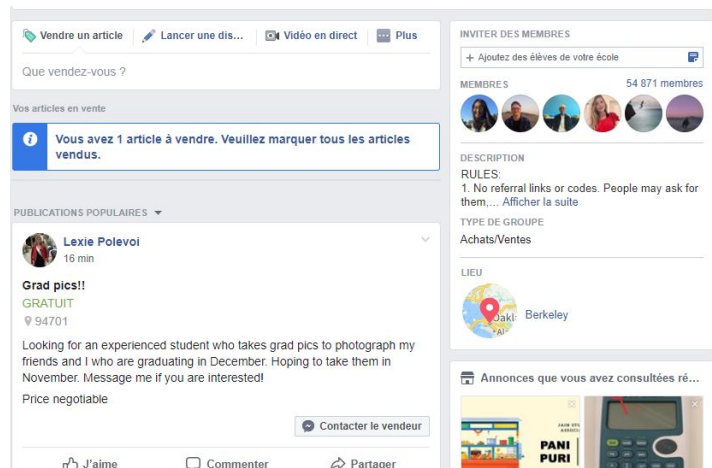
Thinking of a page as building blocks put together

This is one
“row”

This is another
“row” (with more
columns and
rows inside)

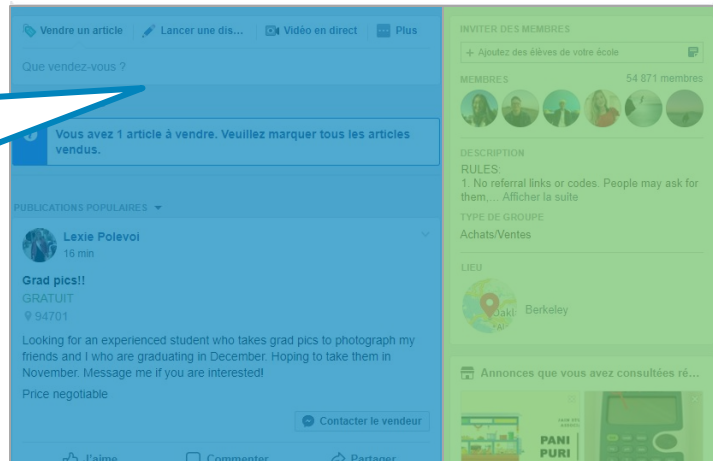


Thinking of a page as building blocks put together



Thinking of a page as building blocks put together

Again, we have columns, each of which have distinct blocks of content (text field to make a new post, alerts, posts, etc)

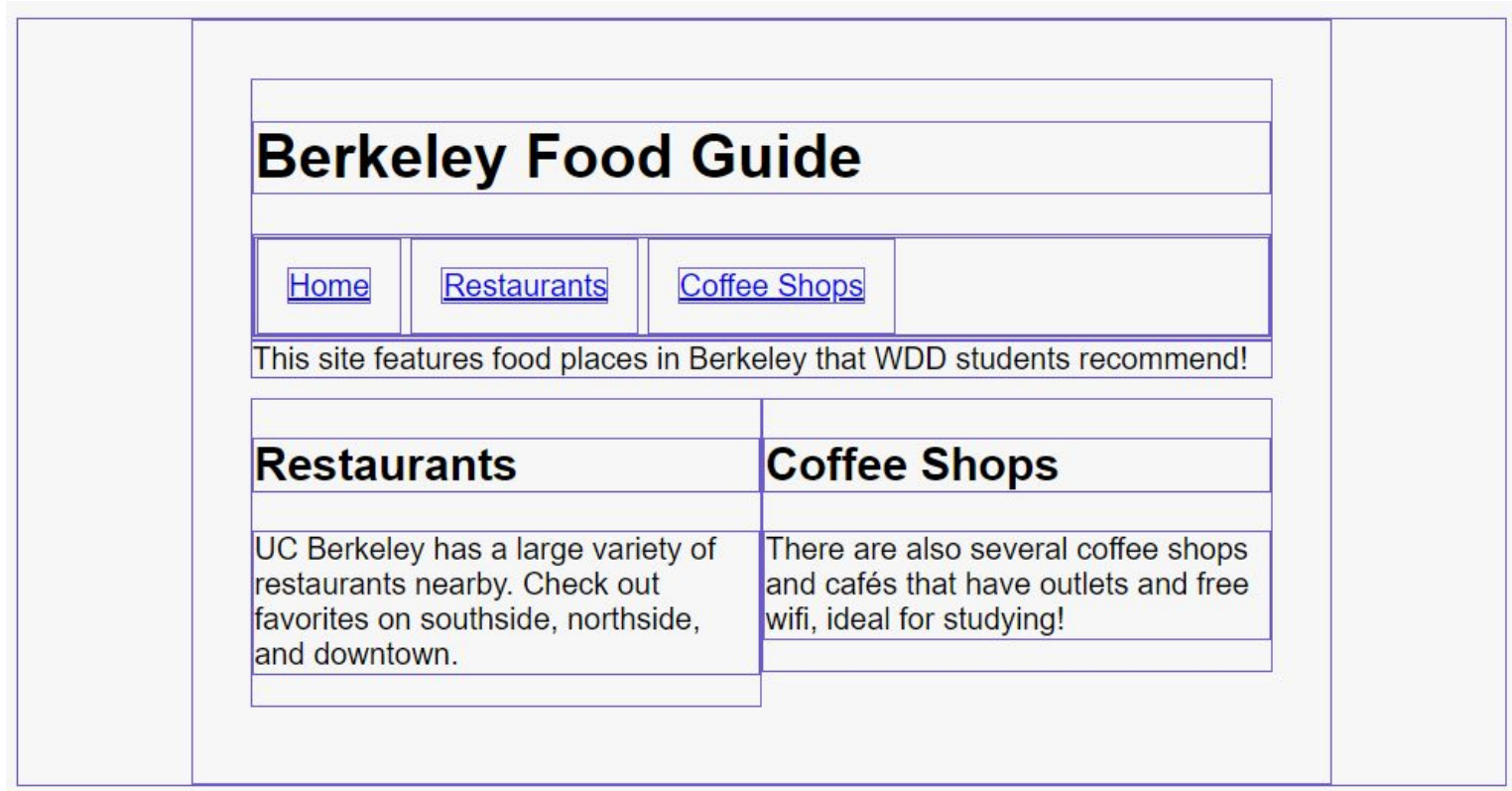


Thinking of a page as building blocks put together

The image shows a Facebook page for a group named "Free & For Sale". The page is composed of several distinct sections, each highlighted with a different colored border to illustrate the concept of building blocks.

- Top Navigation Bar (Orange):** Contains the Facebook logo, the group name "Free & For Sale", a search bar, and user avatars for Heidi, Accueil, and Créer.
- Left Sidebar (Yellow):** Contains navigation links such as "À propos", "Discussion", "Vos articles", "Membres", "Vidéos", "Photos", "Recommandations", "Événements", "Fichiers", and a search bar "Chercher dans groupe".
- Main Content Area (Cyan):** Features a large photo of a squirrel wearing a graduation cap. Below the photo are buttons for "Membre", "Notifications", "Partager", and "Plus".
- Right Sidebar (Light Blue):** Contains information about the group, including "Ajoutez des élèves de votre école", "MEMBRES" (54 871 members), "DESCRIPTION" (RULES: 1. No referral links or codes. People may ask them... Afficher la suite), "TYPE DE GROUPE" (Achats/Ventes), and "LIEU" (Oakland, Berkeley).
- Bottom Section (Red):** Contains a post by Lexie Polevoi, 16 min ago, with the text "Vous avez 1 article à vendre. Veuillez marquer tous les articles vendus." and a button "Contactez le vendeur".

It's like our diagram from week 3!



Some common patterns in layout structure

- There is a **main container** for the page that holds everything
 - The **header** might be separate since it's a block of its own
- A **column** of content is a div
 - These columns have **components** like a list of links, some text, a post
- Columns of content next to each other are put next to each other in a **row**

Start from simple components

→ piece together in rows and columns

→ complex layouts are just put together from simple layouts!

Some common patterns in layout structure !!!

- There is a **main container** for the page that holds everything
 - The **header** might be separate since it's not part of the main content
- A **column** of content is a **div**
 - These columns have **components** like buttons, text, images, etc.
- Columns of content next to each other are put next to each other in a **row**

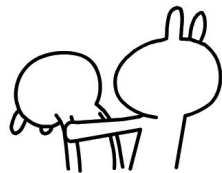
This is how web developers in the industry structure stuff - content is divided into small building blocks by themselves, which then creates larger building blocks!

Start from simple components

→ piece together in rows and columns

→ complex layouts are just put together from simple layouts!

You can probably create like 90% of all websites ever like this!!!!



Questions?