HTML & CSS: FUNDAMENTALS OF DEVELOPMENT

Instructor: Beck Johnson

Week 3

SESSION OVERVIEW

- Review Intro to CSS
- The CSS box model
- Block vs inline elements
- Content Management Systems
- Coding from a design “comp”

REVIEW!

{ } REVIEW: EXAMPLE CSS RULE

```
p { color: blue; }
```

- `selector` is `p` (all `<p>` tags in the HTML)
- `property` is `color`
- `value` is `blue` (many color names are supported, or use the hex code `#0000ff`)

{ } REVIEW: BACKGROUND IMAGES

Can set background of an element as an **image** (instead of a color) with the property `background-image`

The `value` is `url("path")`, where `path` is the **relative** or **absolute** path to where the image lives, like this:

```
p {
    background-image: url("images/kitten.jpg");
    color: white;
}
```

{ } REVIEW: BACKGROUND IMAGES

`background-position:` allows you to move a background image around within its container

`background-attachment:` images usually scroll with the main view, but setting to `fixed` means the image stays in place when the user scrolls the page

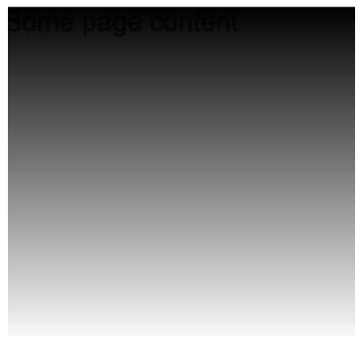
`background-repeat:` defines if (and how) the background image will repeat

`background-size:` specifies how much of the container that the image covers

{ } REVIEW: BACKGROUND IMAGES

You can set `background-image` to `linear-gradient`, which is a gradient that the browser draws for you:

```
section { background: linear-gradient(black, white); }
```



By default `linear-gradient` draws from top to bottom, but you can set the gradient to draw at an angle instead by starting with `to`

```
section {
	background: linear-gradient(to right, red, #f06d06, yellow, green);
}
```

{ } REVIEW: HEIGHT AND WIDTH

`height` and `width` can be set on (most) elements to change how much room they take up on the page.

- We'll discuss later why elements like `<a>` and `<em>` don't change when you set their `height` or `width`

The `value` of this property must be a positive number.

- Units are either px or em
- Or you can specify a percentage

```
header { height: 6em; }
```

REVIEW: LINKING TO EXTERNAL STYLESHEET

```
<link href="css/styles.css" rel="stylesheet">
```

- Tells the browser to find and load the styles.css file from the css directory
- The `rel` attribute stands for "relation" - in this case, this link's relationship to the document is "stylesheet"
- This tag goes inside the `<head>` element
- Should be on every page that needs the styles

REVIEW: THE "CASCADING" PART

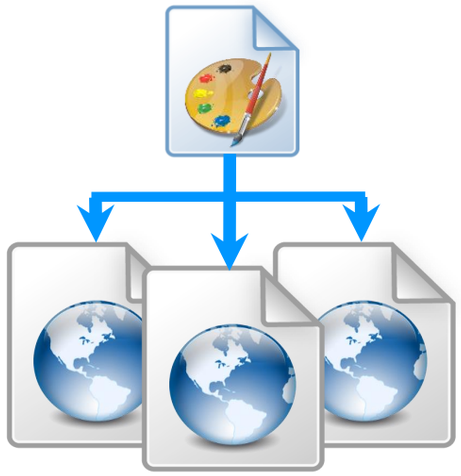
The 3 rules for determining how styles get applied:

- Styles are applied from **far** to **near**
- Styles are applied from **top** to **bottom**
- **Children** elements are more specific than **parents**

REVIEW: NEAR TO FAR

Styles that are “closer” to the elements they style take precedence

Browser default



External styles
(in a **.css** file)

Internal styles
(in the <head>)

Inline styles
(directly on an element)

Closer to element

REVIEW: TOP TO BOTTOM

If the same `property` is styled multiple times for the same `selector`, the last one sticks.

```
p { color: #2f4251; }
p { color:  #daa645; } /*this wins*/
```

REVIEW: CHILDREN ARE SPECIFIC

If one style is **more specific** than another, it takes precedence

```
p { color: #daa645; } /* all paragraphs */

a { color: #e7c0c8; } /* links in general */

p a { color: #c4fe46; } /* links in paragraphs */
```

QUESTIONS?

THE CSS BOX MODEL

CSS BOX MODEL

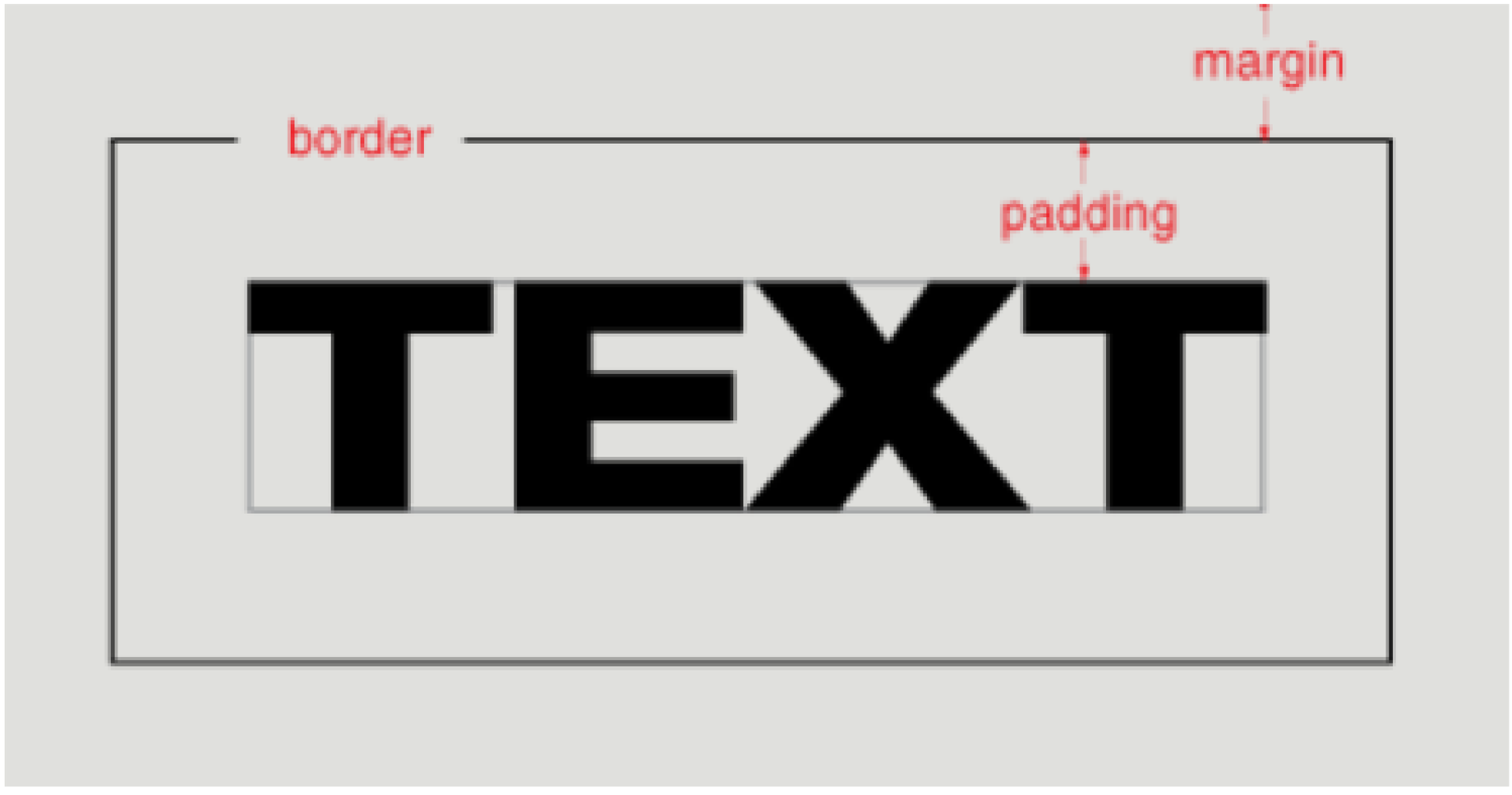
CONTENT: stuff in the box

PADDING: bubble wrap and packing peanuts

BORDER: sides of the box

MARGIN: space between multiple boxes

CSS BOX MODEL



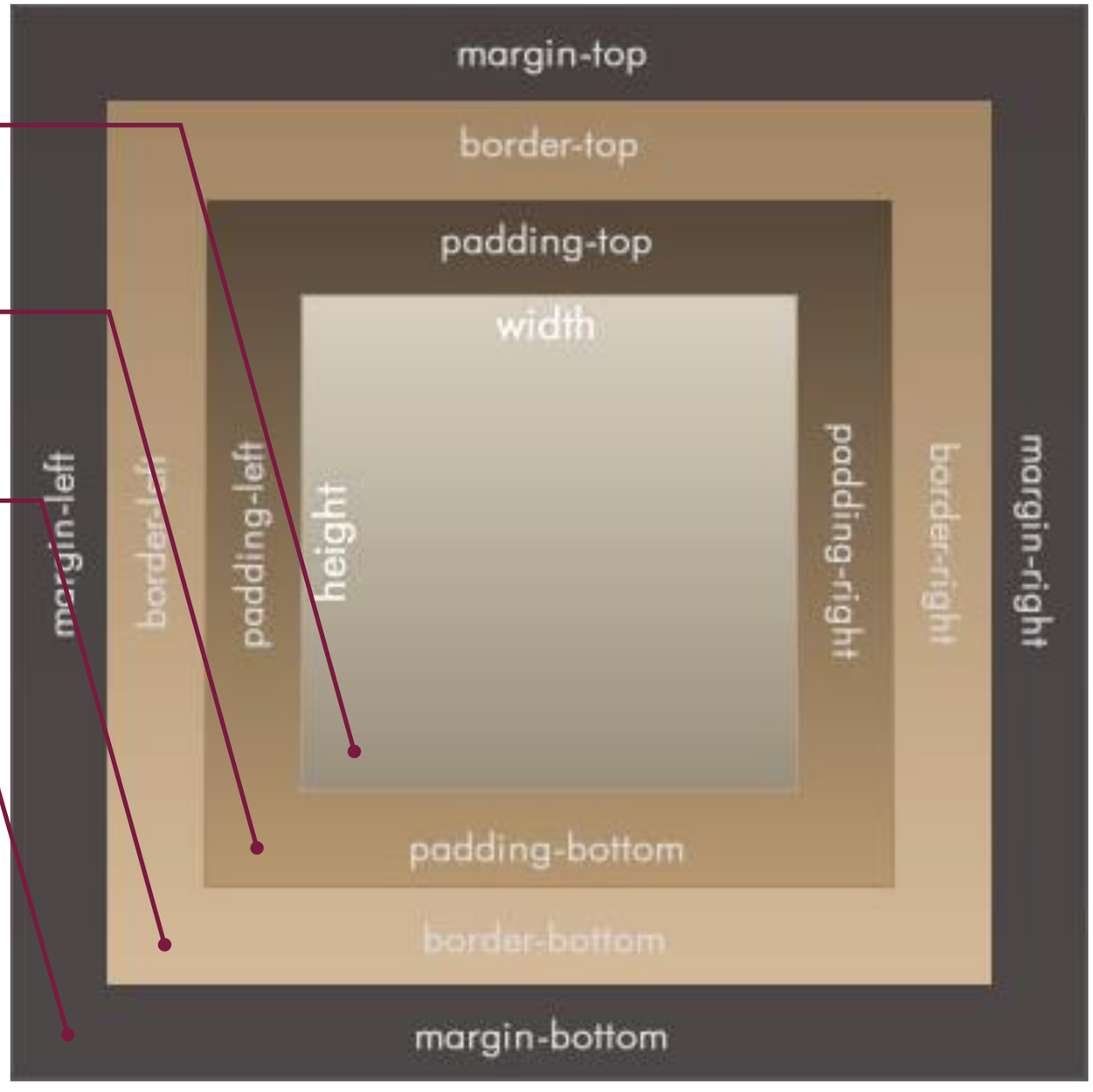
CSS BOX MODEL

Content

Padding

Border

Margin

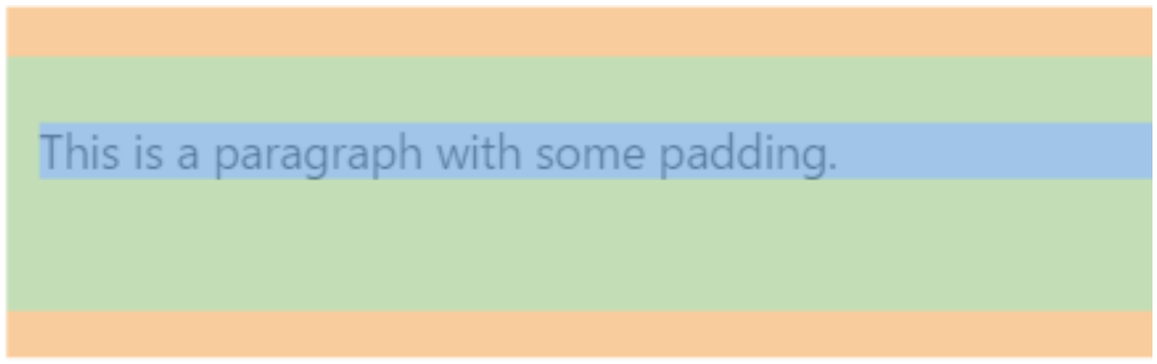


PADDING

Padding creates space **inside** an element.

Padding affects how far content is from the border.

```
p {
    padding-top: 20px;
    padding-right: 5px;
    padding-bottom: 40px;
    padding-left: 10px;
}
```



Shorter way:

```
p {
    padding: 20px 5px 40px 10px;
}
```

PADDING

Padding is useful for moving content away from the edges of its container.

This is a paragraph with zero padding (default)

This is a paragraph with 1em padding

PADDING

If **top/bottom** and **left/right** padding match...

```
p {
    padding-top: 20px;
    padding-right: 10px;
    padding-bottom: 20px;
    padding-left: 10px;
}
```

Combine them!

```
p { padding: 20px 10px; }
```

PADDING

If **all** padding matches...

```
p {
    padding-top: 20px;
    padding-right: 20px;
    padding-bottom: 20px;
    padding-left: 20px;
}
```

Combine EVEN MORE!

```
p { padding: 20px; }
```

PADDING

Padding can be applied only to the top, only to the bottom, and so on – or any combination of those:

```
p {
    padding-left: 40px;
}

p {
    padding-top: 20px;
    padding-right: 10px;
}
```

MARGIN

Margin creates space **outside** an element.

- Same abbreviation style and rules as padding

```
p {
    margin-top: 20px;
    margin-right: 5px;
    margin-bottom: 40px;
    margin-left: 10px;
}
```

Is the same as

```
p { margin: 20px 5px 40px 10px; }
```

MARGIN

You can give `margin` a **negative** value to shift elements in the opposite direction.

```
p {
    margin-top: -20px;
}
```

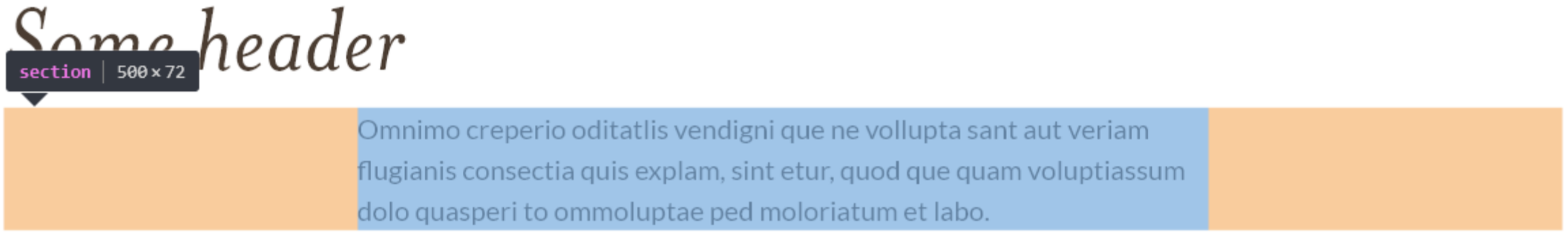
This may result in overlapping text!

MARGIN AUTO

To automatically center elements, you can use the property `auto`, which evenly applies a margin on both sides

- When using `auto`, a `width` must be applied to the element, so that the browser knows how much margin to automatically apply

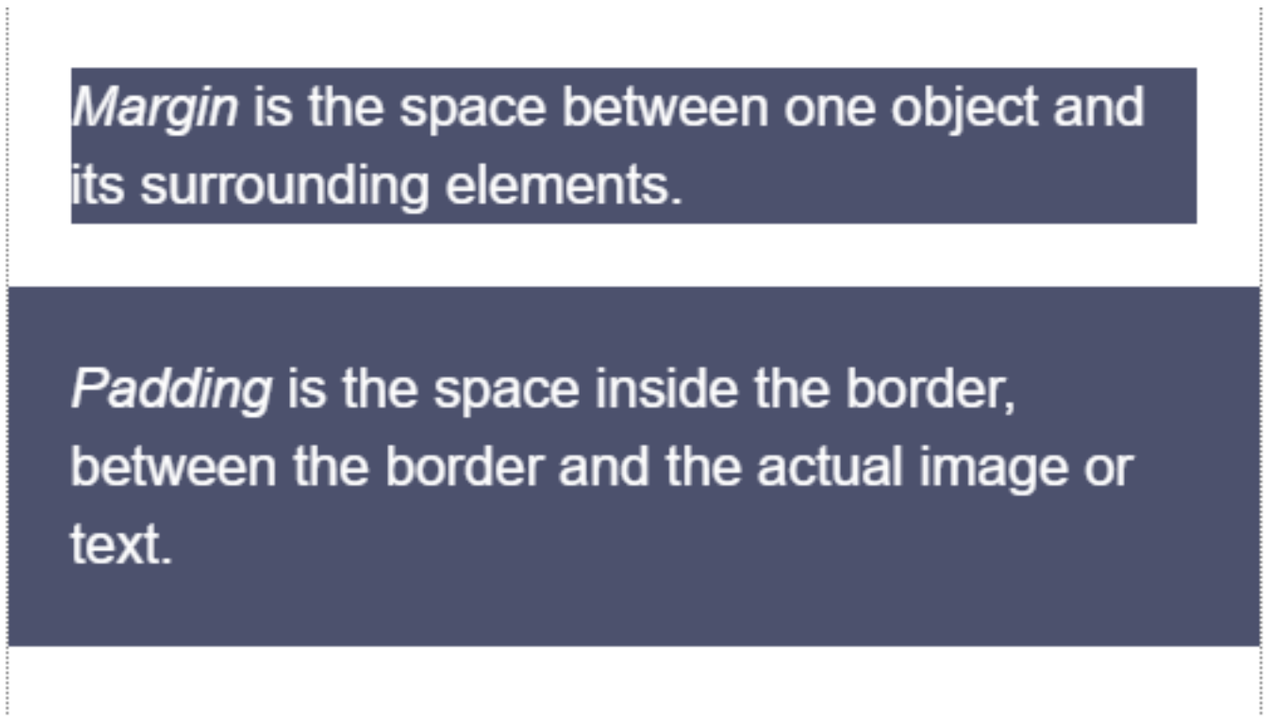
```
section {
    margin: 0 auto;
    width: 500px;
}
```



MARGIN VS. PADDING

Use `margin` to separate the element from the things that are around it.

Use `padding` to move the element away from the edges of the block.



BORDER STYLES

Between margin and padding, you can set a `border`

Values are separated with spaces, in this order:

- Width (usually in pixels, but can be em)
- Border style (solid, dotted, dashed, etc)
- Color

```
p {
    border: 2px dotted #ff0000;
}
```

BORDER STYLES

Border styles:

`solid` — Solid line.

`dotted` — Series of dots.

`dashed` — Series of dashes.

`double` — Two solid lines.

`groove` — Representation of a carved groove. Opposite of `ridge`.

`ridge` — Representation of an embossed ridge. Opposite of `groove`.

`inset` — Representation of an inset depression. Opposite of `outset`.

`outset` — Representation of an outset extrusion. Opposite of `inset`.

BORDER STYLES

You can set a border on only one side of an element:

```
h1 { border-bottom: 3px solid black; }
```

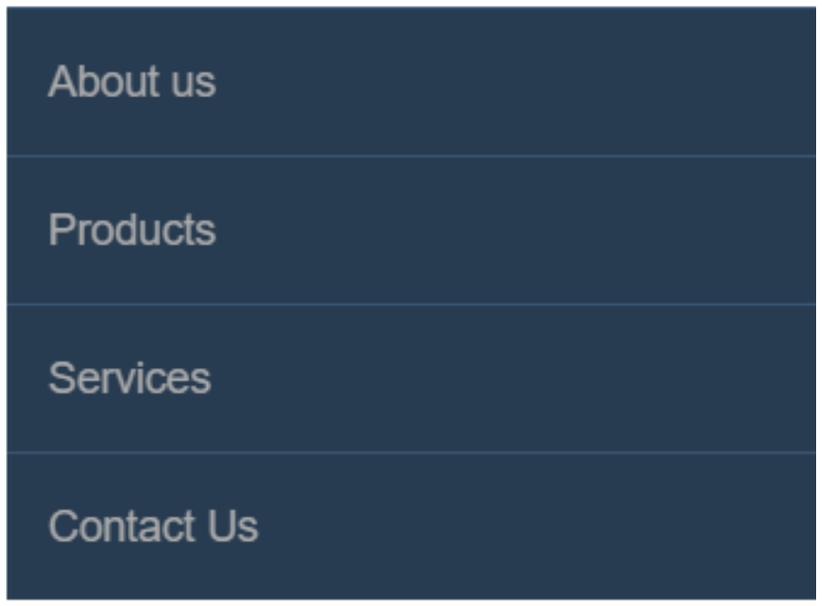
HEADER WITH BORDER BOTTOM

BORDER STYLES

A common use of `border` is to visually separate list items in a navigation menu.

```
ul {
      list-style : none;
}

li {
      padding: 1em;
      background-color: #283c51;
      border-top: 1px solid #395673;
      color: #adadad;
}
```



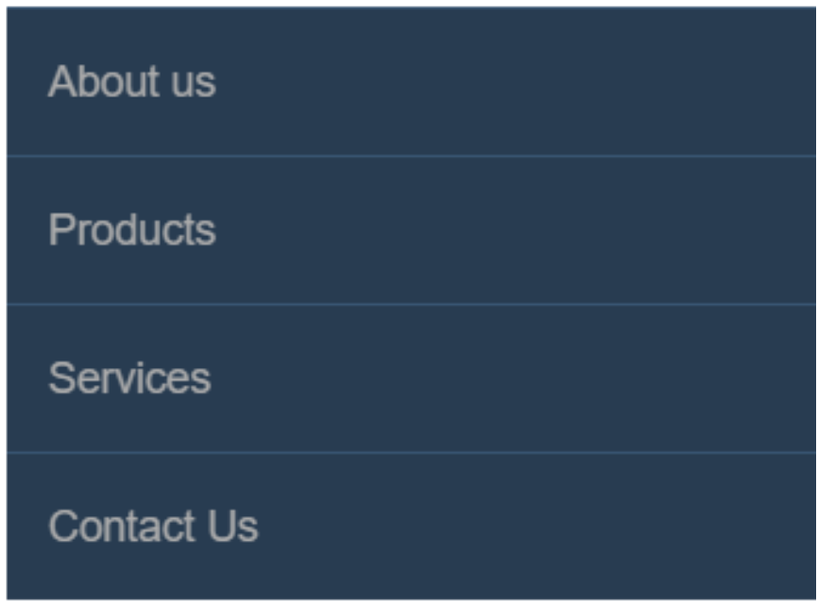

```
<ul>
      <li>About us</li>
      <li>Products</li>
      <li>Services</li>
      <li>Contact Us</li>
</ul>
```

LIST STYLE

Note that we set

```
ul {
     list-style: none;
}
```

to remove the bullets that appear by default on an unordered list

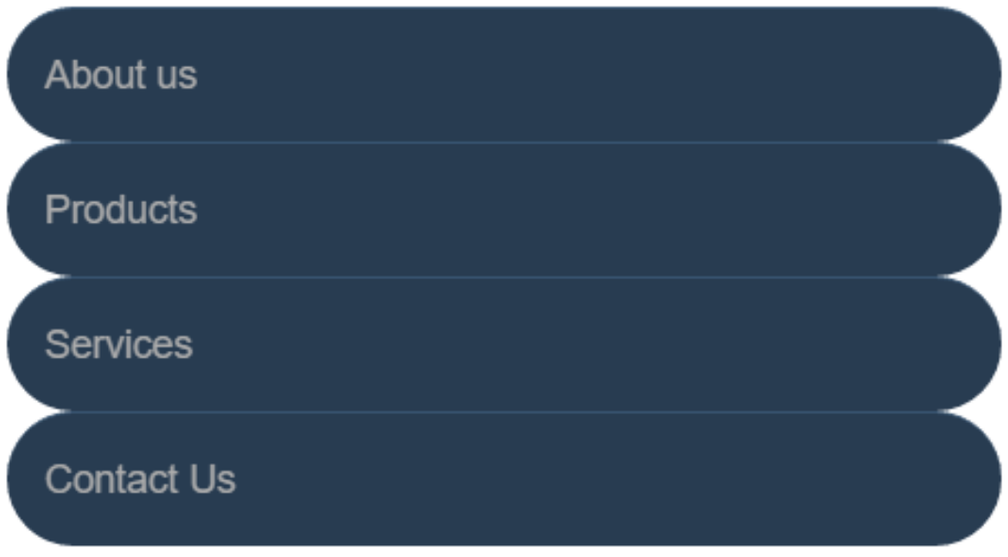


BORDER RADIUS

To make an element appear curved, use the property `border-radius`

- The value is a number (in px or em) or percentage
- You can use `border-radius` even if you don't explicitly set a `border`

```
li {
      /* same styles... */
      border-radius: 2em;
}
```



BORDER RADIUS

`border-radius` can be used to create a circle.

- Set `border-radius` to `50%`
- Set `height` and `width` to the same value

```
li {
      border-radius: 50%;
      background-color: black;
      color: white;
      text-align: center;
      height: 3em;
      width: 3em;
      line-height: 3em;
      margin: 5px;
}
```

BORDER RADIUS

This technique can be used on images to crop them into a circle

- Note: if the image itself doesn't have a square ratio, it will look distorted

```
<img src="kitten.jpg" />

img {
      border-radius: 50%;
      height: 200px;
      width: 200px;
}
```

PRACTICE TIME!

ASSIGNMENT

- Add a **list** of links in your navigation menu

- Make the navigation menu pretty by using padding, margin, border, background color, and other tricks we've learned.
  - ONLY style lists that are in the nav menu – not any lists that may appear on the rest of the page

- Give your design some "breathing room" with padding and/or margin.

BLOCK VS. INLINE ELEMENTS

BLOCK ELEMENTS

- Expand naturally to fill their parent container
  - Takes up a “full line”
- Can have margin and/or padding
- Can have height and/or width
- By default, will be placed **below** previous elements in the markup

<> BLOCK ELEMENTS

BLOCK ELEMENTS EXPAND NATURALLY →

AND NATURALLY DROP BELOW OTHER ELEMENTS ↙

<> BLOCK ELEMENTS

Examples of block elements:

- Headings `<h1>`...`<h6>`
- Paragraphs `<p>`
- Lists `<ul>`, `<ol>`

INLINE ELEMENTS

- Flow along with text content
- Only take up as much space as necessary
- Ignore width and height properties
- Margin and padding only pushes other elements away horizontally, not vertically
- Top and bottom margin/padding is ignored

<> INLINE ELEMENTS

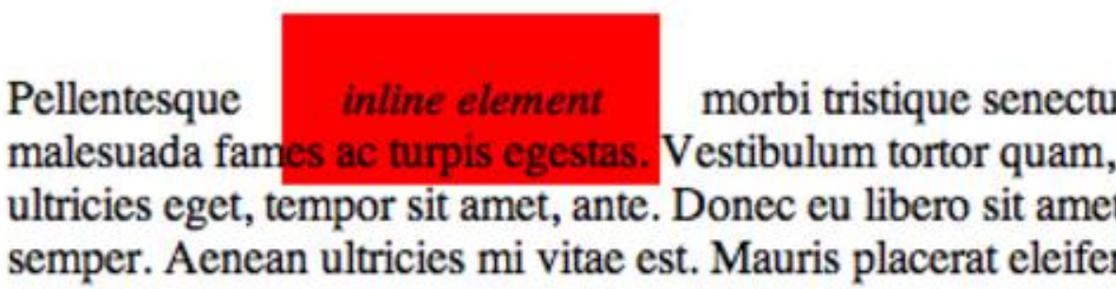
INLINE ELEMENTS FLOW WITH TEXT

PELLENTESQUE HABITANT MORBI TRISTIQUE SENECTUS ET NETUS ET MALESUADA FAMES AC TURPIS EGESTAS. VESTIBULUM `INLINE ELEMENT` VITAE, ULTRICIES EGET, TEMPOR SIT AMET, ANTE. DONEC EU LIBERO SIT AMET QUAM EGESTAS SEMPER. AENEAN ULTRICIES MI VITAE EST. MAURIS PLACERAT ELEIFEND LEO.

<> BLOCK & INLINE ELEMENTS

Examples of inline elements:

- Links `<a>`
- Font emphasis `<em>`
- Font bold `<strong>`



<> INLINE BLOCK

INLINE-BLOCK ELEMENT

- Is a hybrid of inline and block
- Takes up width and height like block-level elements
- Flows with content
- Can have margin and padding
- Examples of inline-block elements:
  - Image `<img />`

<> INLINE BLOCK

senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

<> DISPLAY

You can change whether or not any element is block, inline, or inline-block by using the CSS `display` property.

- This means we can do some neat things!

```
li {
    display: inline-block;
}
```

PRACTICE TIME!

ASSIGNMENT

Update your navigation to a horizontal menu using CSS.

- Give the `li` elements a `display` property of either `inline` or `inline-block`. Which looks better? Why do you think that is?
- Update your styles so that they look nice in the new orientation

Create an `<a>` link that looks like a button.

- Style the button differently on hover and click

<html>

(MORE) HTML ELEMENTS

<SPAN> ELEMENTS

```
<span></span>
```

A `<span>` is a **generic inline element**

- No default style

- Used to style inline content

<DIV> ELEMENTS

```
<div></div>
```

A `<div>` is a **generic block element**

- No default style

- Heavily used as a wrapper for other elements, to create complex layouts

WHY USE DIV OR SPAN?

Both `div` and `span` really need something extra to be useful, since they have no presentation style by default.

- Used mostly to create **layout**
- Have no semantic meaning
- You don't need to "reset" them before making them fit your design (like `ul` or `p`)

CONTENT MANAGEMENT SYSTEMS (CMS)

CONTENT, DESIGN, & CODE

CONTENT

most important part of any website

DESIGN

critical to the best user experience

CODE

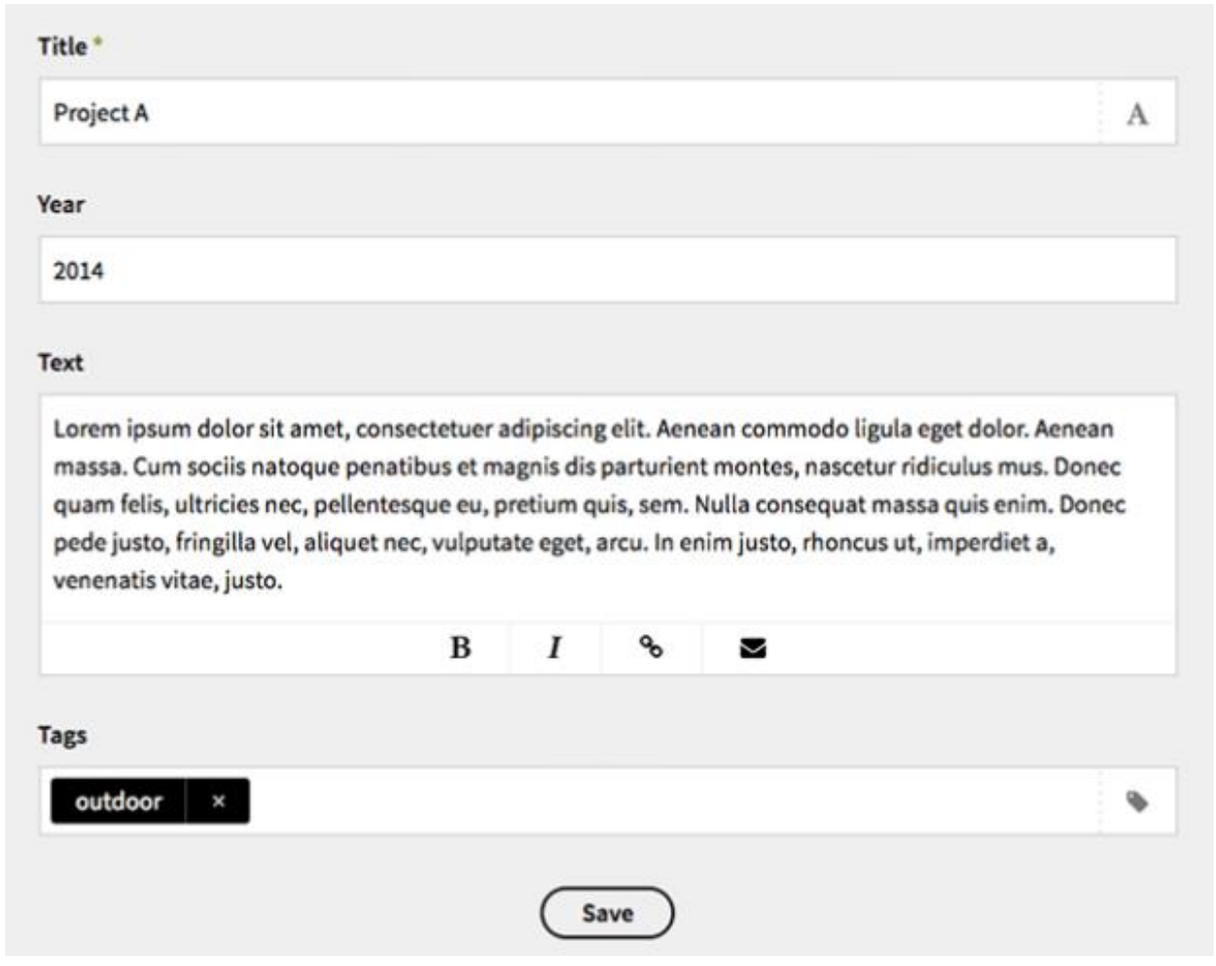
brings content and design to life

CMS

Most modern websites use a Content Management System to store content

A CMS is a 3rd party system that holds images and text.

That content is dynamically injected into a markup template.



A developer builds markup as normal, but uses placeholders instead of actual content:

```
<div class="row">
    <h1>@locationName</h1>
    <p>@Model.Location.Address @Model.Location.City @Model.Location.State_Province @Model.Location.Phone_Number
</div>

<div class="row">

    <h2>Associates</h2>

    <img src="@employee.Employee_Image" alt="@employee.First_Name @employee.Last_Name" />
    <h3>@employee.First_Name @employee.Middle_Name @employee.Last_Name</h3>
    <h4>@employee.Job_Title</h4>
    <p>@employee.Content</p>

</div>
```

CMS PROS

- Non-developers can easily edit text and change images, and even create new pages
- Allows translation services to load different languages without changing the design or code of the website
- Since images are loaded from a third party, this distributes the load across multiple servers, which usually speeds pageload

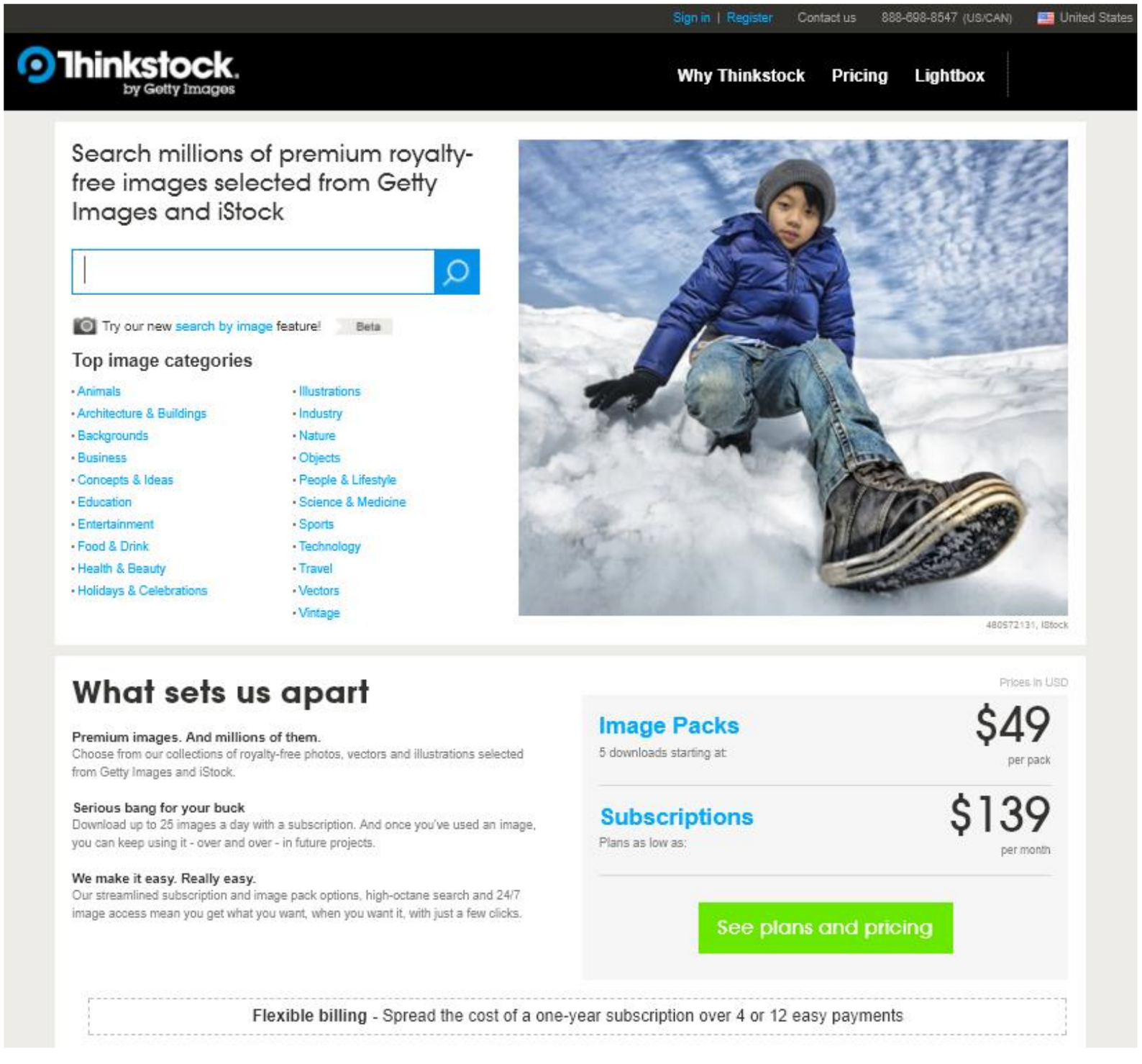
CMS CONS

- Most CMS are expensive and challenging to set up, and require ongoing training and administration

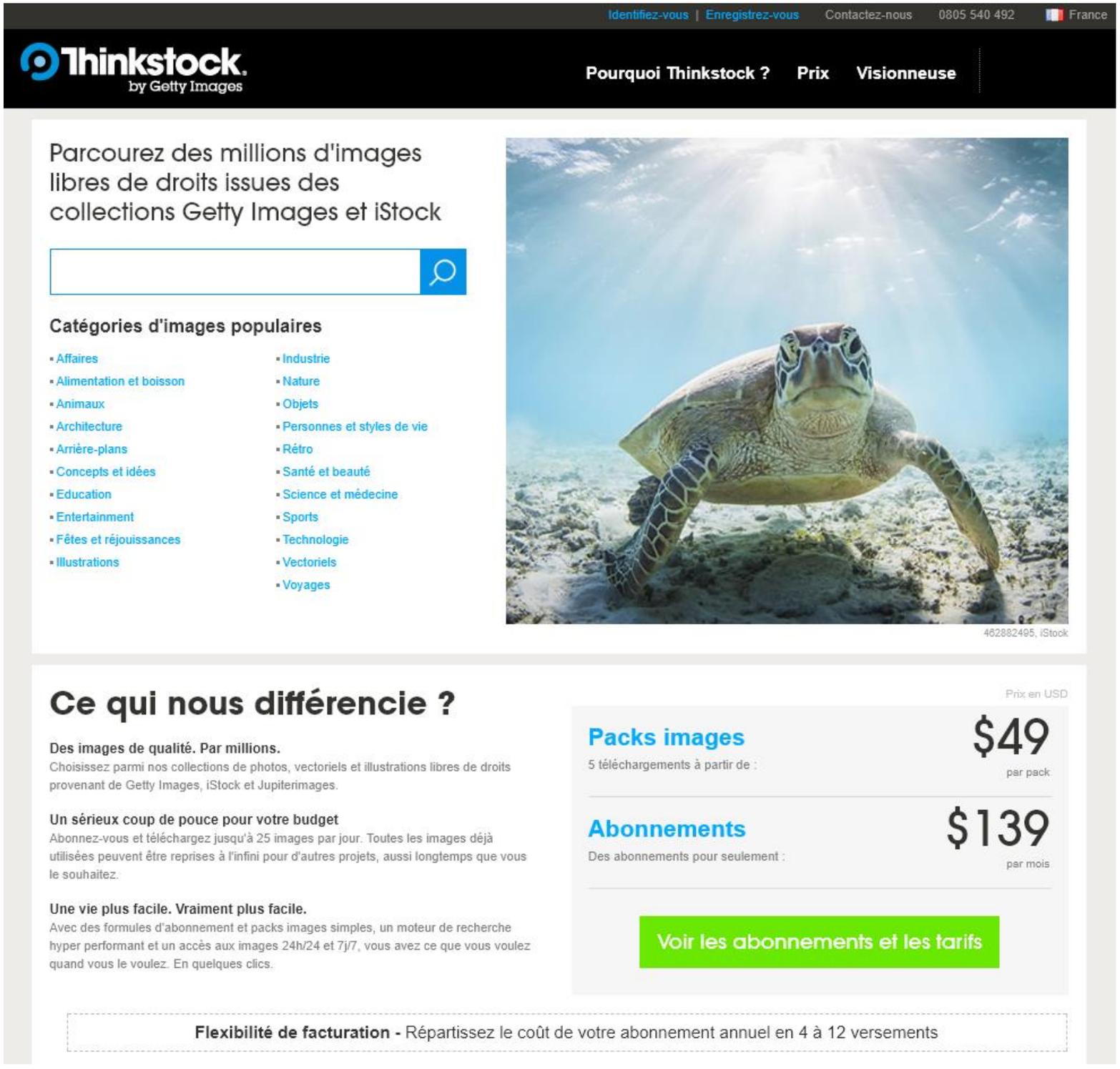
- Because you can no longer know how much content is going on a page, you have to design for more flexibility

- Most European languages are more than 2x longer than English (especially German!)

CMS FOR TRANSLATIONS



CMS FOR TRANSLATIONS



POPULAR CMS

There are hundreds of Content Management Systems, all using different ways to store, edit, and access data

PUTTING IT TOGETHER:
WORKING FROM A DESIGN COMP

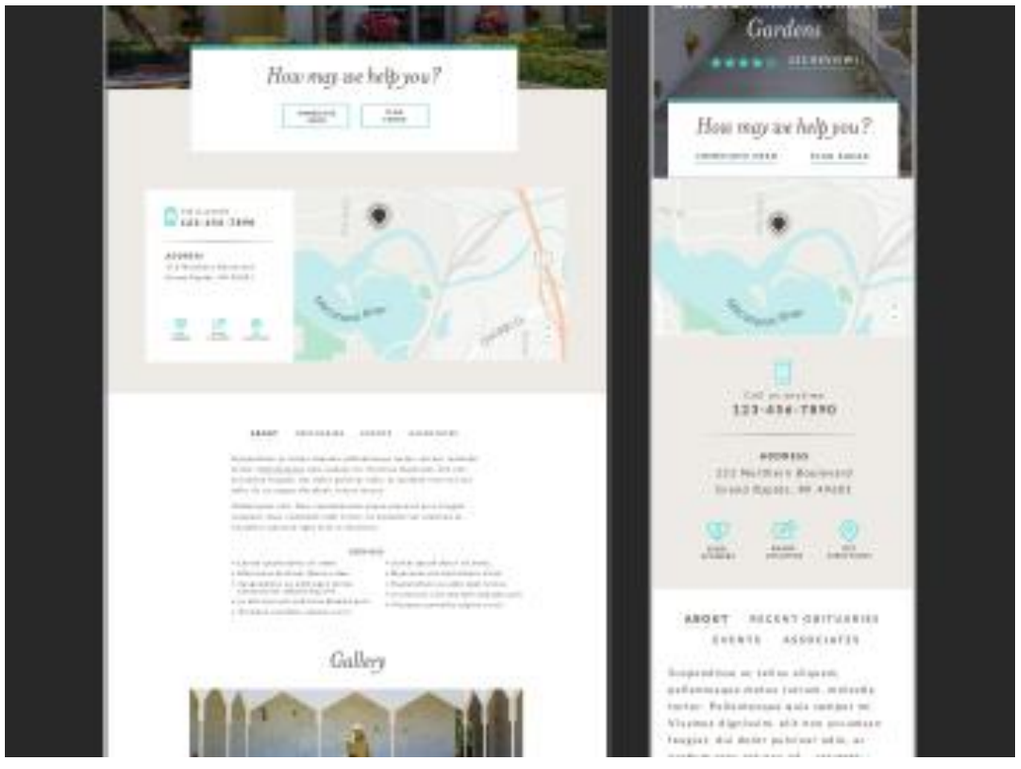
COMPS

A standard way for designers to tell developers what they need to code is via a design “comp” (comprehensive layout)

- A comp is a picture of the website the way it should look
- Can be delivered as a Photoshop psd file, or an annotated png or pdf file

COMPS

- Different states should be shown (for example, what happens when you open a dropdown menu?)
- Different screen resolutions should be shown (mobile, tablet, desktop)
- The developer inspects the text, colors, and font types in Photoshop to determine what to code



STYLE GUIDE

Alternatively, designers can provide developers with a style guide instead of a comp

- Used in conjunction with wireframes, which show what content belongs on a page and how that content should be laid out

- A style guide shows an example of common elements on the site and the color palette to use

STYLE GUIDE

COLOR PALETTE

- **CANDY APPLE** #e82535
- **CRIMSON** #c92830
- **COOL BLUE** #074167
- **DEEP LAKE** #143548
- **GRAVEL** #565656
- **WET CEMENT** #71848e

TYPOGRAPHY

Header 1

Font: Montserrat Bold | Color #183548

HEADER 2

Font: Montserrat Bold | Color #183548

HEADER 3

Font: Montserrat Bold | Color #074167

HEADER 4

Font: Montserrat Bold | Color #074167

Header 5

Font: Montserrat Bold | Color #565656

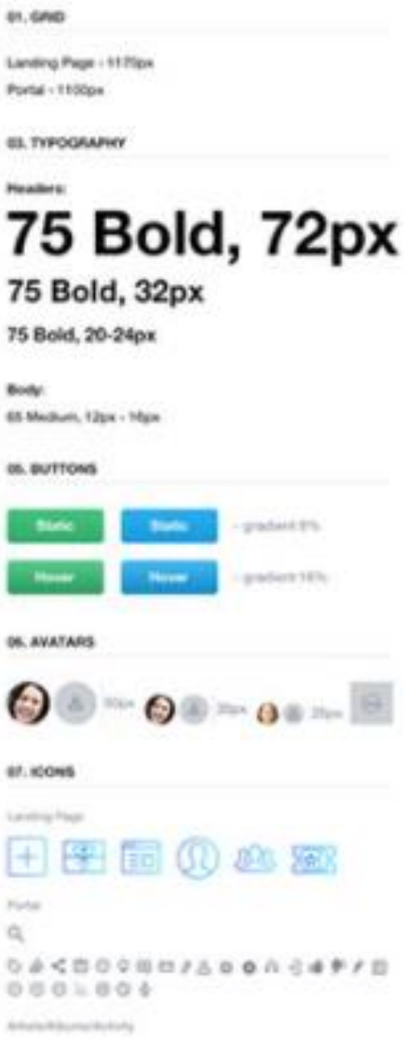
This is Body copy. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent eu semper ligula, nec fermentum odio. Aenean non blandit neque, ac accumsan nibh. Morbi elementum neque id sodales blandit. Morbi eget turpis in urna sodales pharetra. Aenean quis pulvinar lacus, sed lacinia sem.

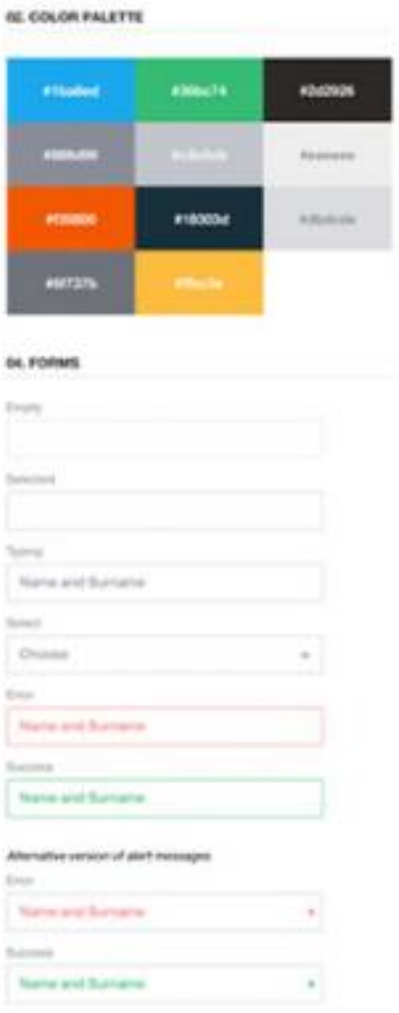
Font: Libre Baskerville | Color #565656

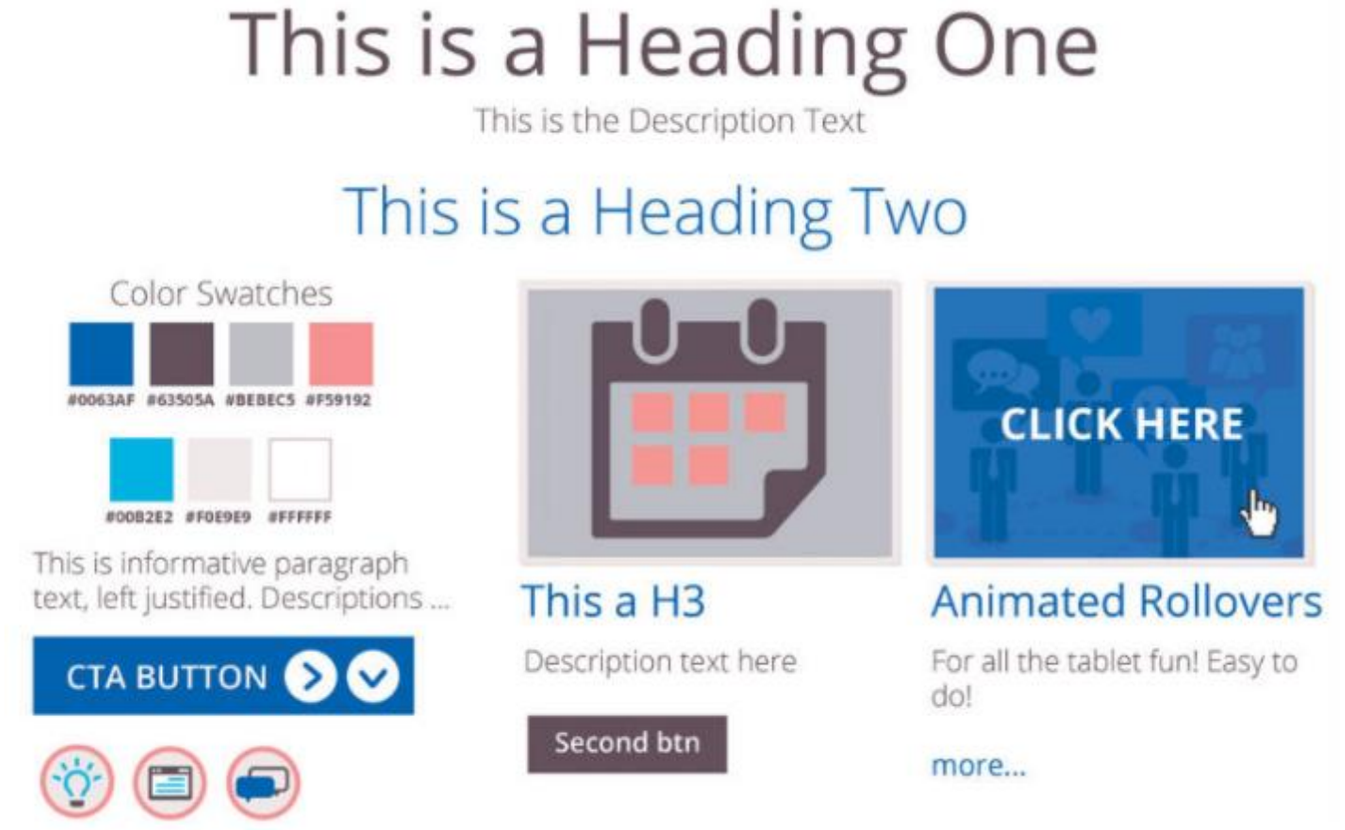
This is an inline link

Font: Libre Baskerville | Color #e82535 | Underline: dotted

STYLE GUIDE


01. GRID
Landing Page - 1170px
Portal - 1100px
03. TYPOGRAPHY
Headers:
75 Bold, 72px
75 Bold, 32px
75 Bold, 20-24px
Body:
65 Medium, 12px - 16px
05. BUTTONS
Static
Static
Hover
Hover
06. AVATARS
07. ICONS

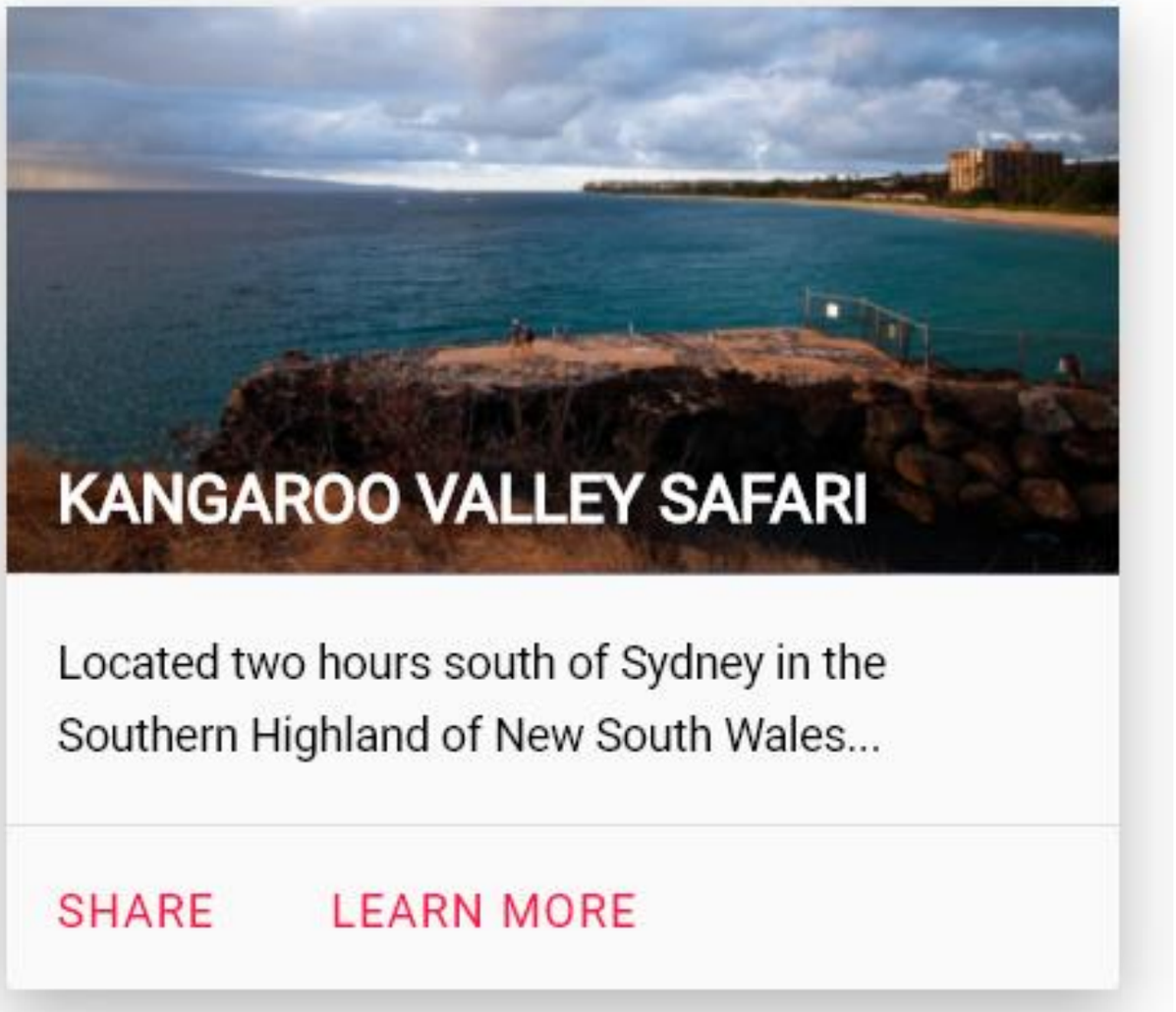

02. COLOR PALETTE
04. FORMS
Name and Surname
Name and Surname
Name and Surname
Name and Surname


This is a Heading One
This is the Description Text
This is a Heading Two
Color Swatches
#0063AF #63505A #BEBEC5 #F59192
#00B2E2 #F0E9E9 #FFFFFF
This is informative paragraph text, left justified. Descriptions ...
CTA BUTTON
This a H3
Description text here
Second btn
CLICK HERE
Animated Rollovers
For all the tablet fun! Easy to do!
more...

PRACTICE TIME!

{ } MATCH THE COMP

Using all the techniques you've learned, try to match this comp as closely as you can:



{ } MATCH THE COMP: TIPS

- Use a `<section>` for the entire tile, and specify a `width` using CSS

- Use a `<header>` to hold the heading, and apply a background image to that element

- Use this URL to generate a random nature photo: https://placeimg.com/344/204/nature

- Use a `<nav>` for the row that contains links, and specify a `border-top` using CSS

- To get the drop-shadow effect, apply this CSS to the `section`:

```
box-shadow: 0 7px 35px 0 rgba(0, 0, 0, 0.3);
```

“HOMEWORK”

- Practice!

- Optional: read chapter 8 of *HTML and CSS: Design and Build Websites*

- Try playing with this interactive demo of the CSS box model