

Lunch Break!

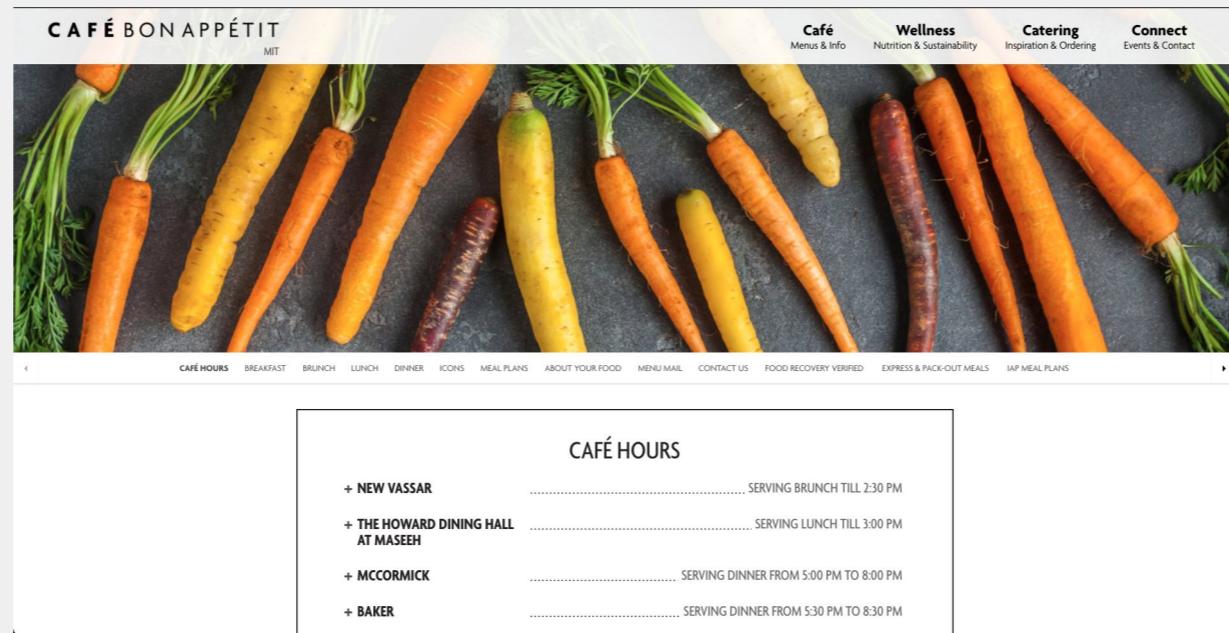


HTML & CSS

Thanh Nguyen

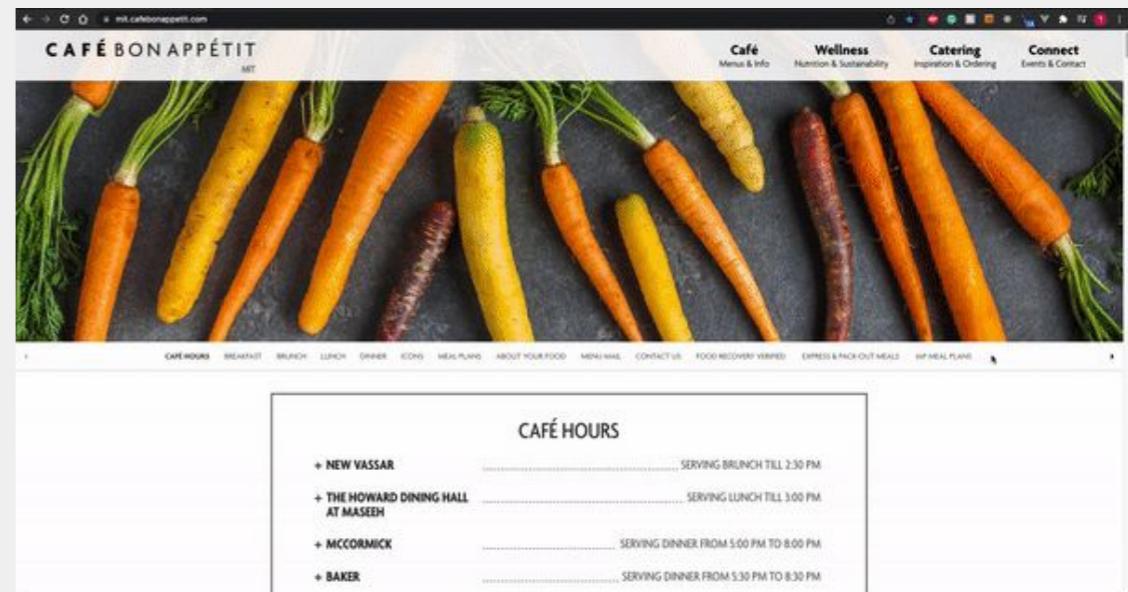
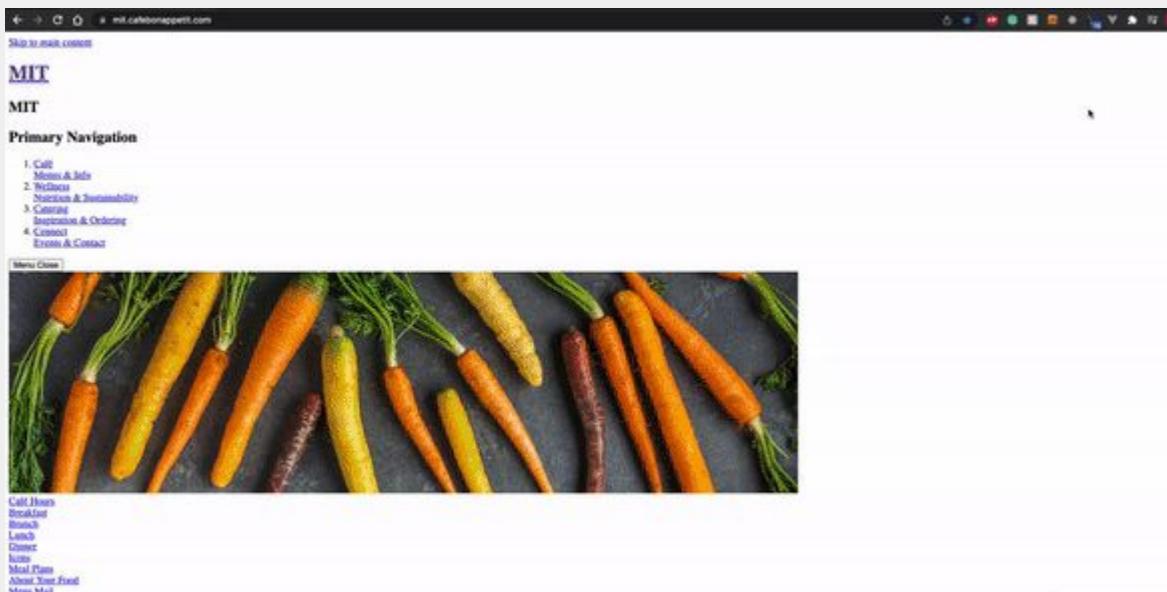


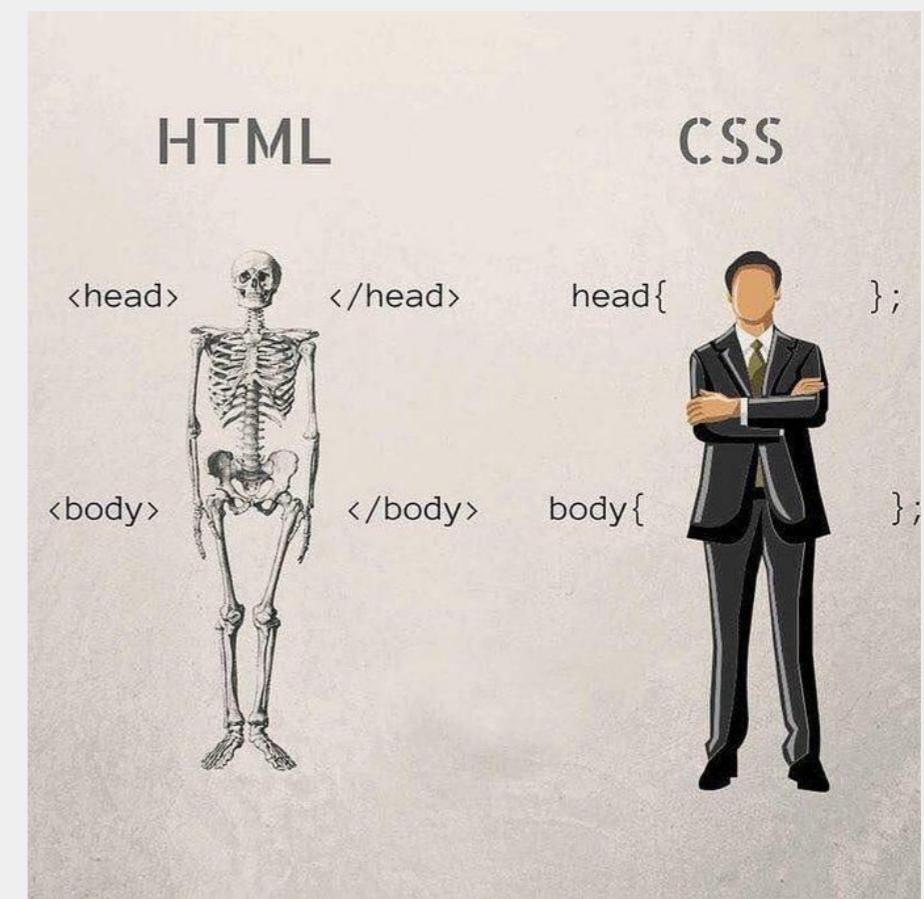
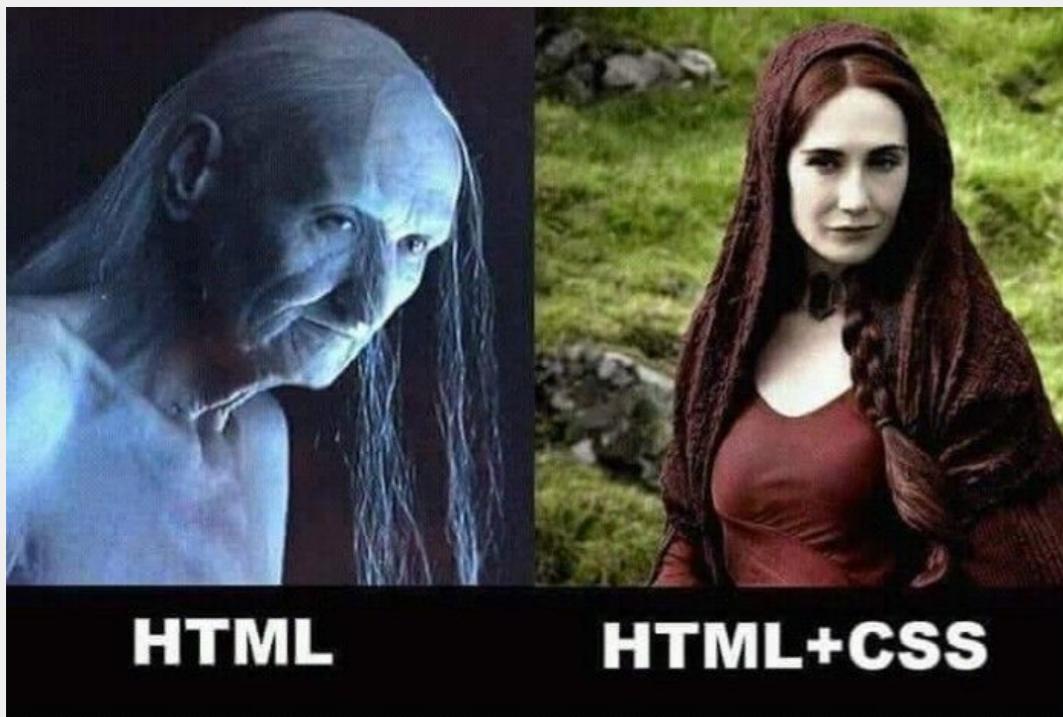
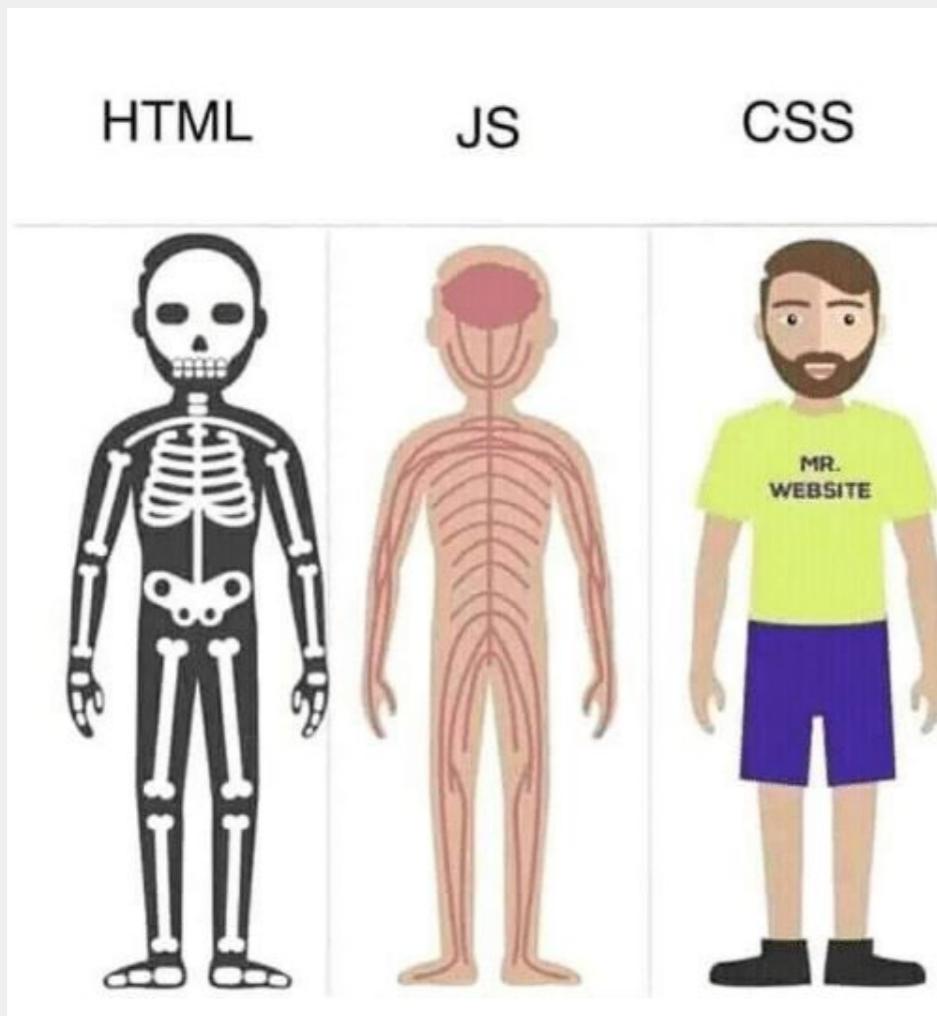
Why does this matter??



html

css





HTML

Hypertext Markup Language

The language your web browser uses to describe the content and structure of web pages

HTML

Hypertext Markup Language

“What does it mean”

HTML = Nested Boxes

ARCHIVE
WHAT IF?
BLAG
STORE
ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

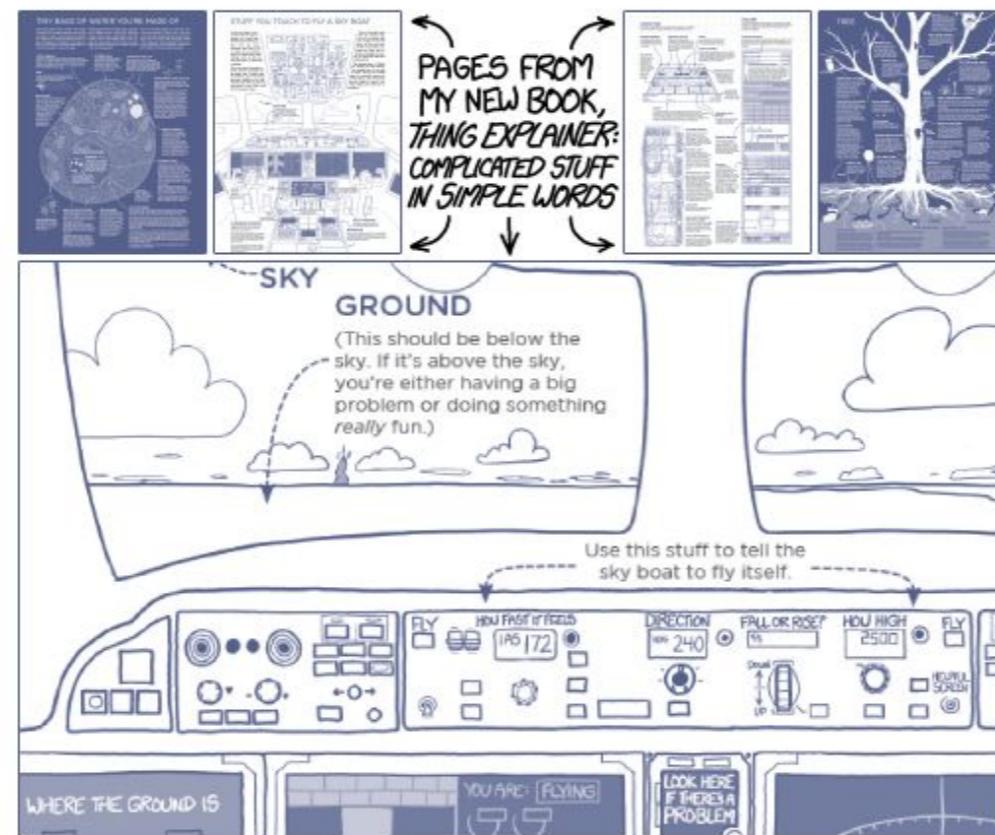
TAGS

<| < PREV RANDOM NEXT > |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

<| < PREV RANDOM NEXT > |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



ARCHIVE
WHAT IF?
BLAG
STORE
ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

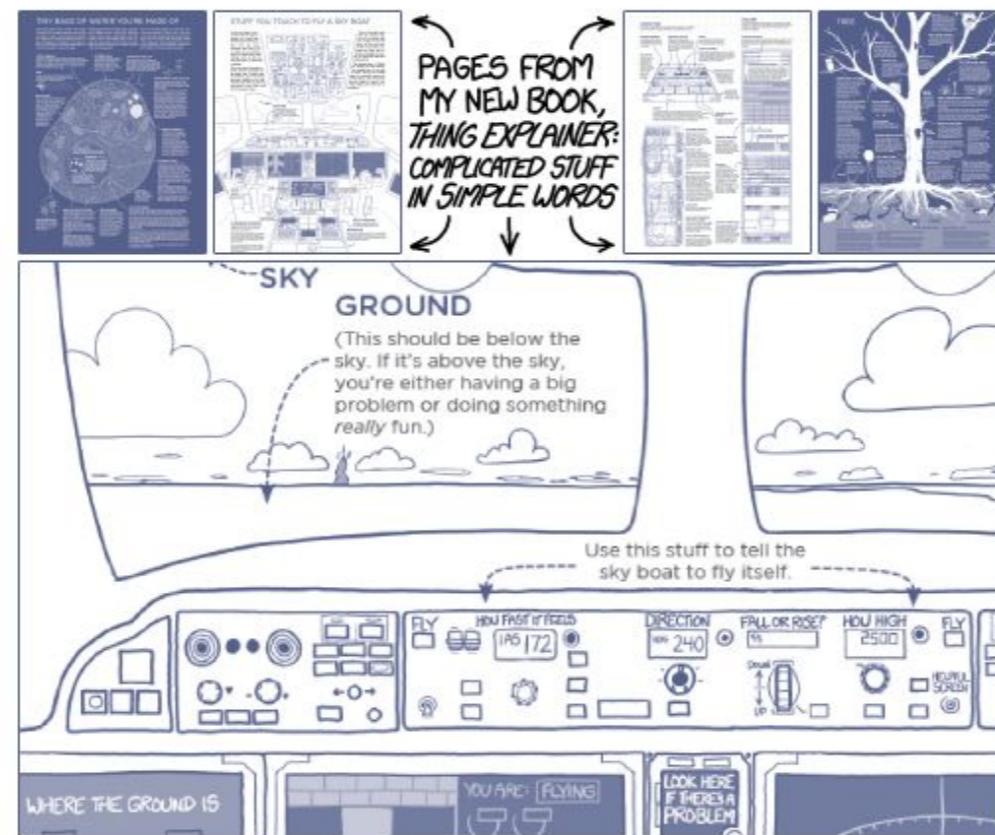
TAGS

<| < PREV RANDOM NEXT > |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

<| < PREV RANDOM NEXT > |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



ARCHIVE
WHAT IF?
BLAG
STORE
ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

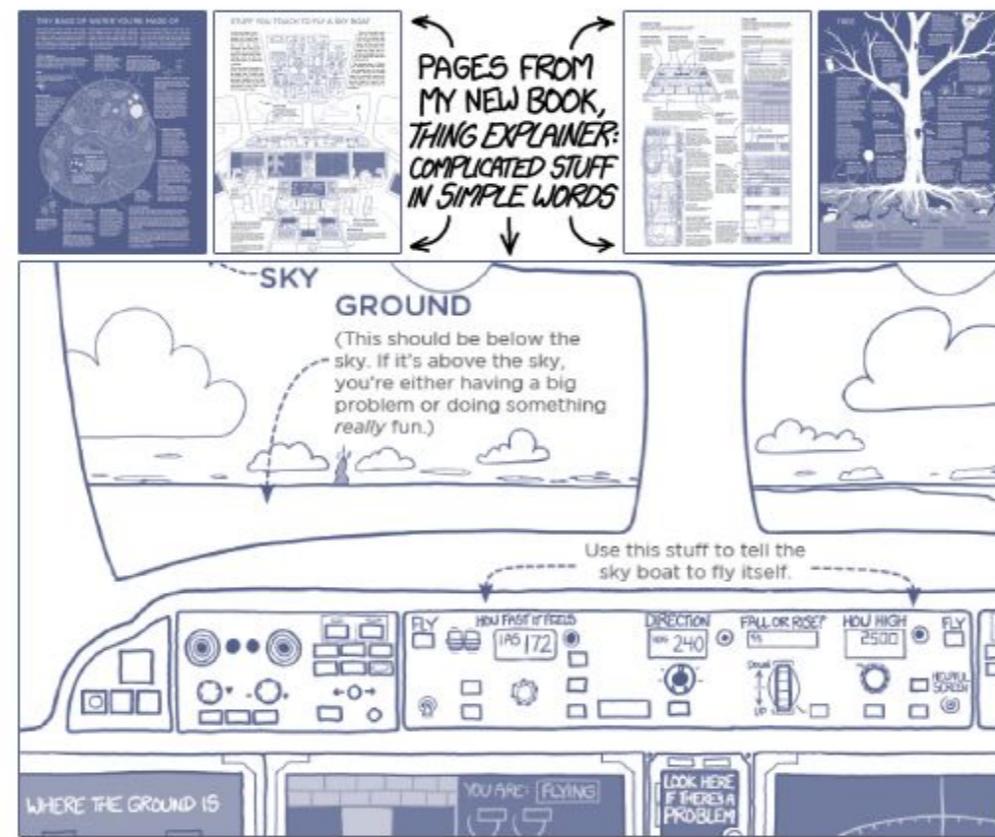
TAGS

<| < PREV RANDOM NEXT > |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

<| < PREV RANDOM NEXT > |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://IMGS.XKCD.COM/COMICS/TAGS.PNG)



ARCHIVE
WHAT IF?
BLAG
STORE
ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

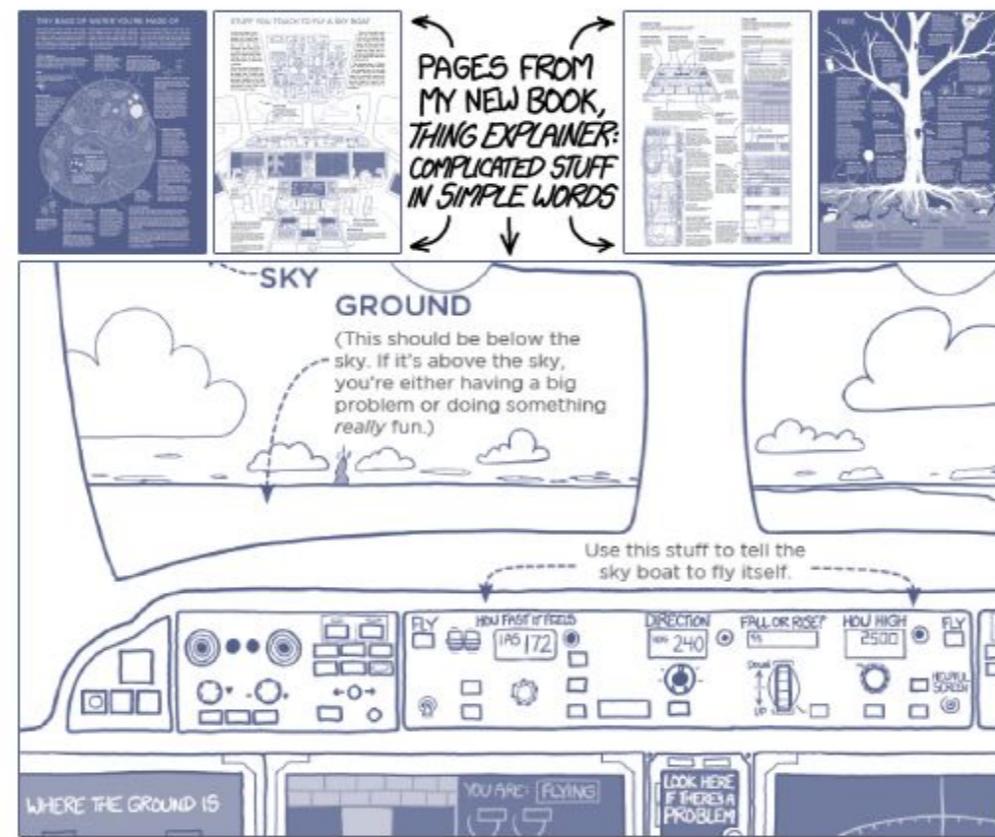
TAGS

<| [< PREV](#) [RANDOM](#) [NEXT >](#) |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

<| [< PREV](#) [RANDOM](#) [NEXT >](#) |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://IMGS.XKCD.COM/COMICS/TAGS.PNG)



ARCHIVE

WHAT IF?

BLAG

STORE

ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

TAGS

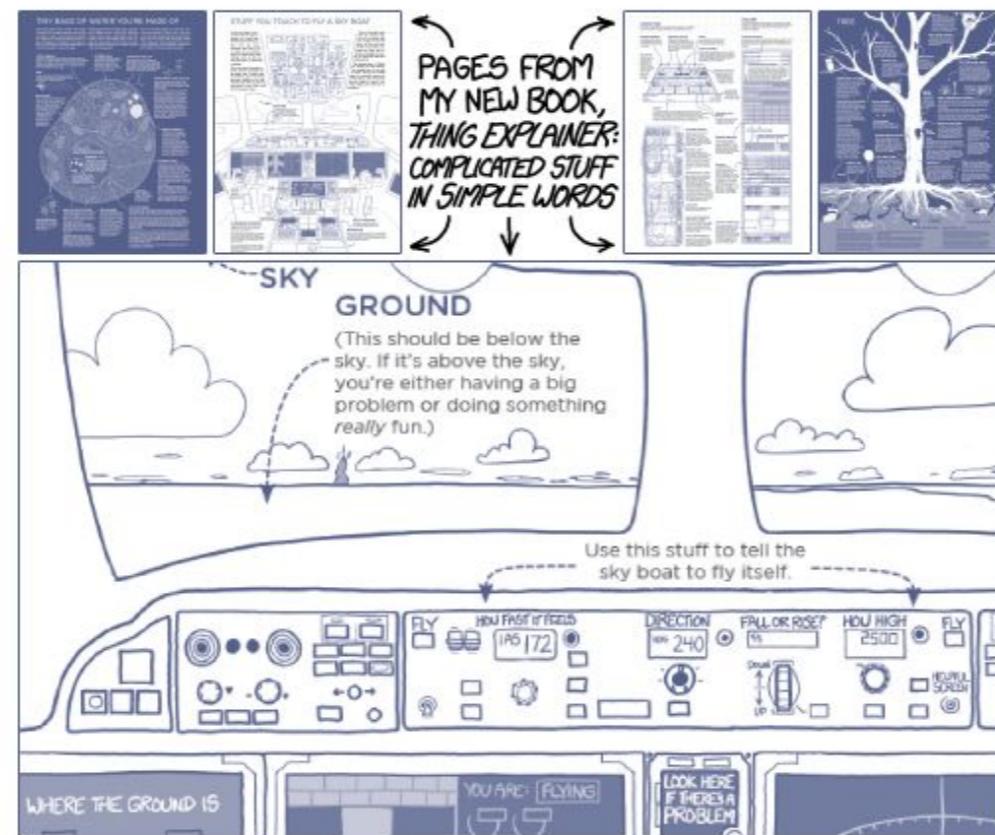
<| [< PREV](#) [RANDOM](#) [NEXT >](#) |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

<| [< PREV](#) [RANDOM](#) [NEXT >](#) |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)

IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



ARCHIVE

WHAT IF?

BLAG

STORE

ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

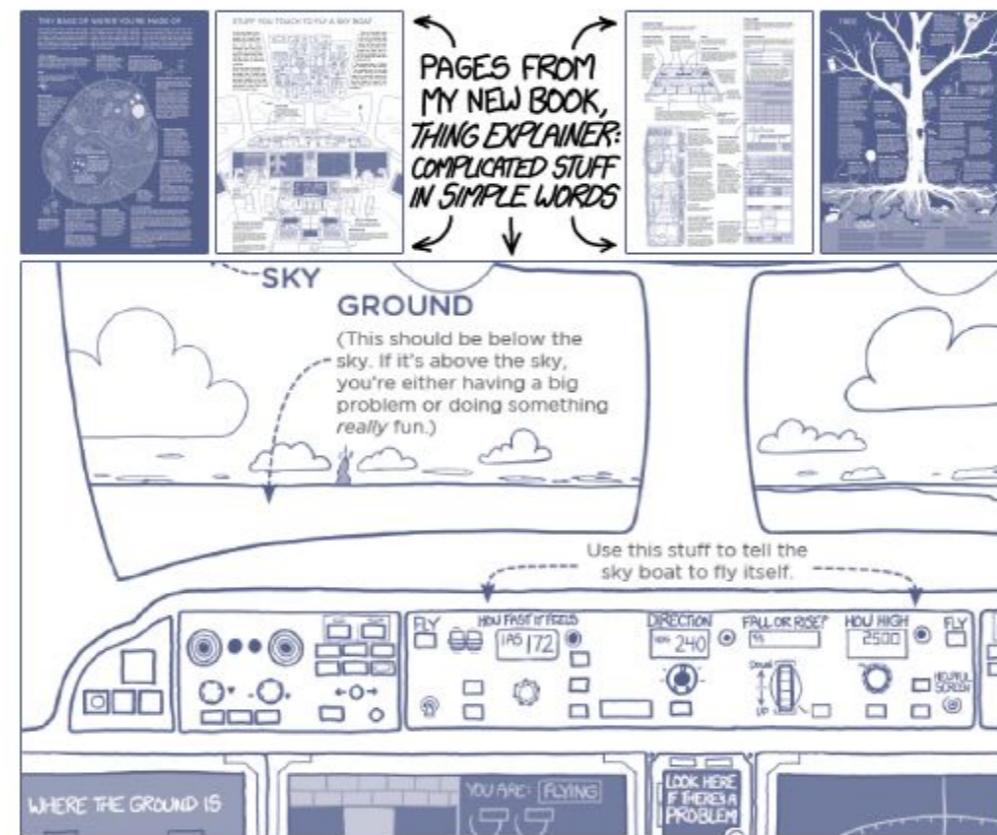
TAGS

|< < PREV RANDOM NEXT > >|

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

|< < PREV RANDOM NEXT > >|

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



ARCHIVE

WHAT IF?

BLAG

STORE

ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

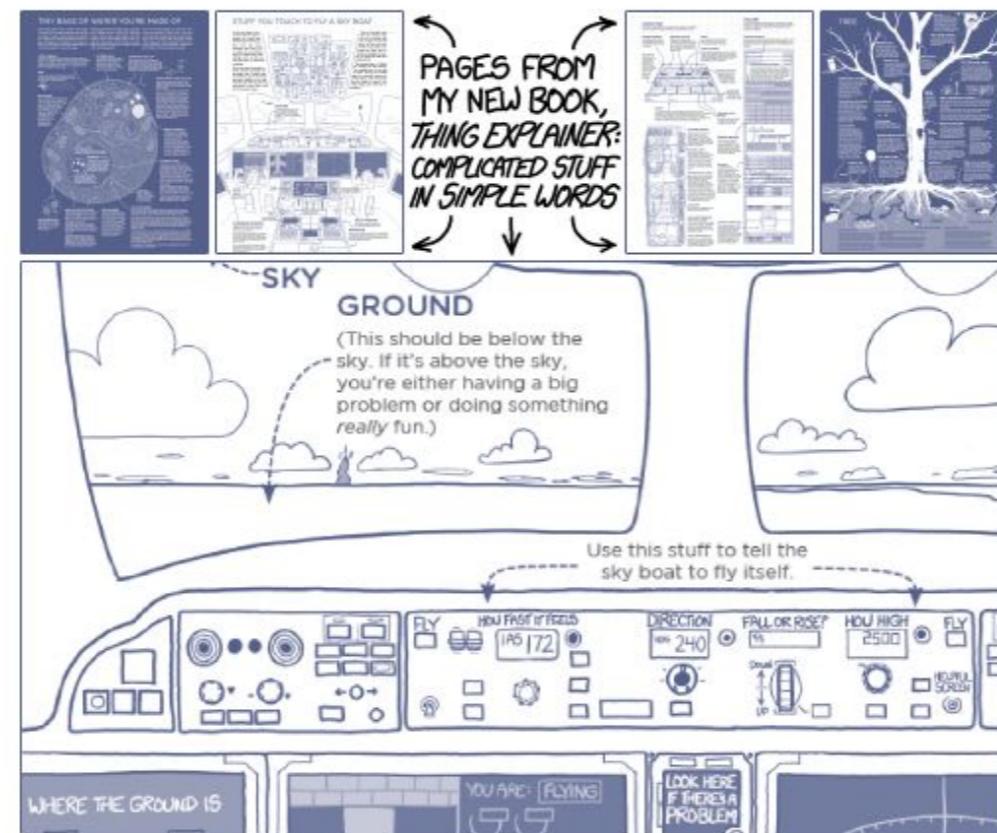
TAGS

|<| < PREV | RANDOM | NEXT > |>|

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

|<| < PREV | RANDOM | NEXT > |>|

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



ARCHIVE

WHAT IF?

BLAG

STORE

ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

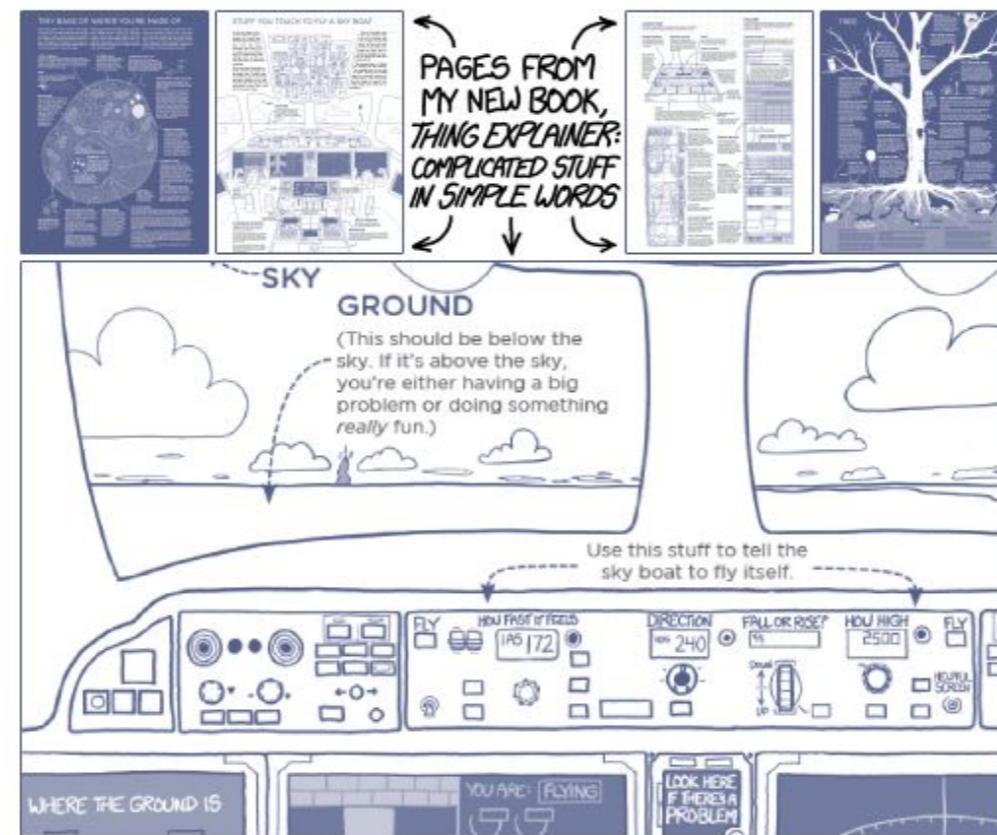
THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

TAGS

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

| < PREV | RANDOM | NEXT > |

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)

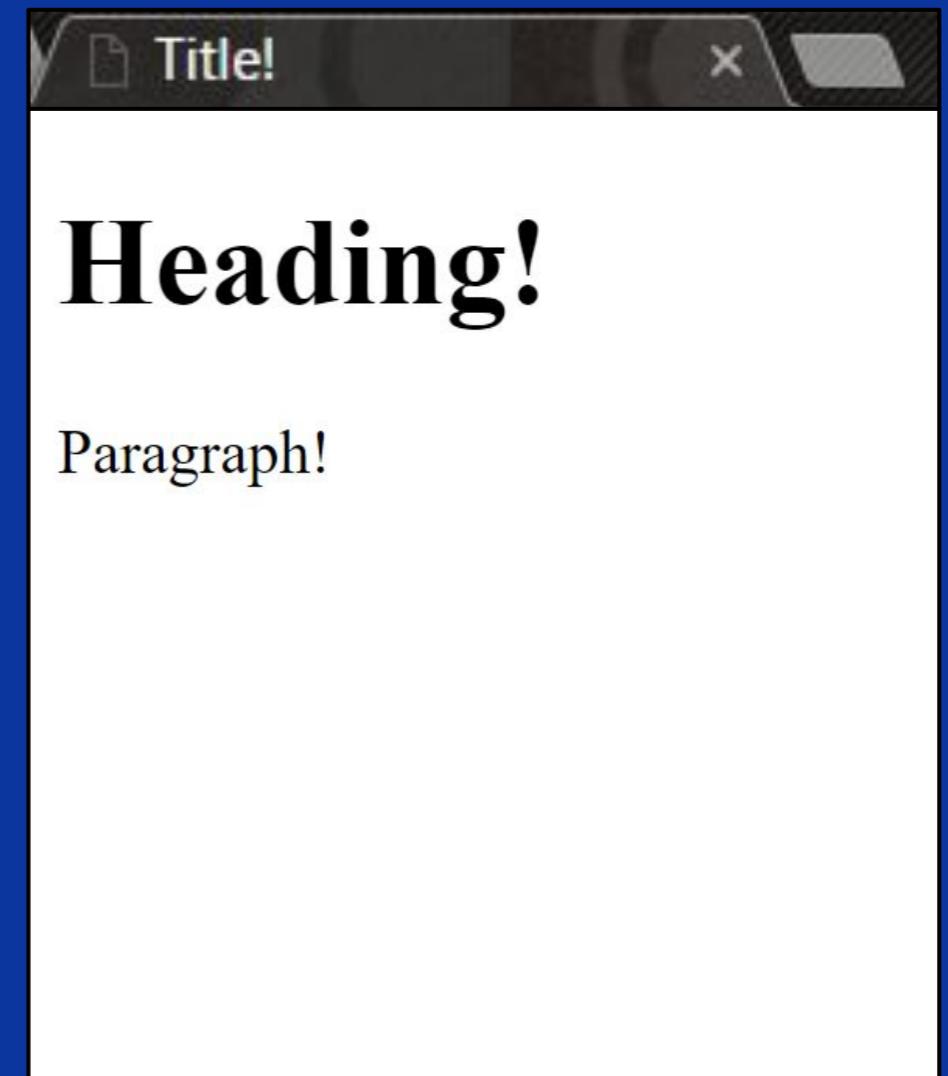


hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```

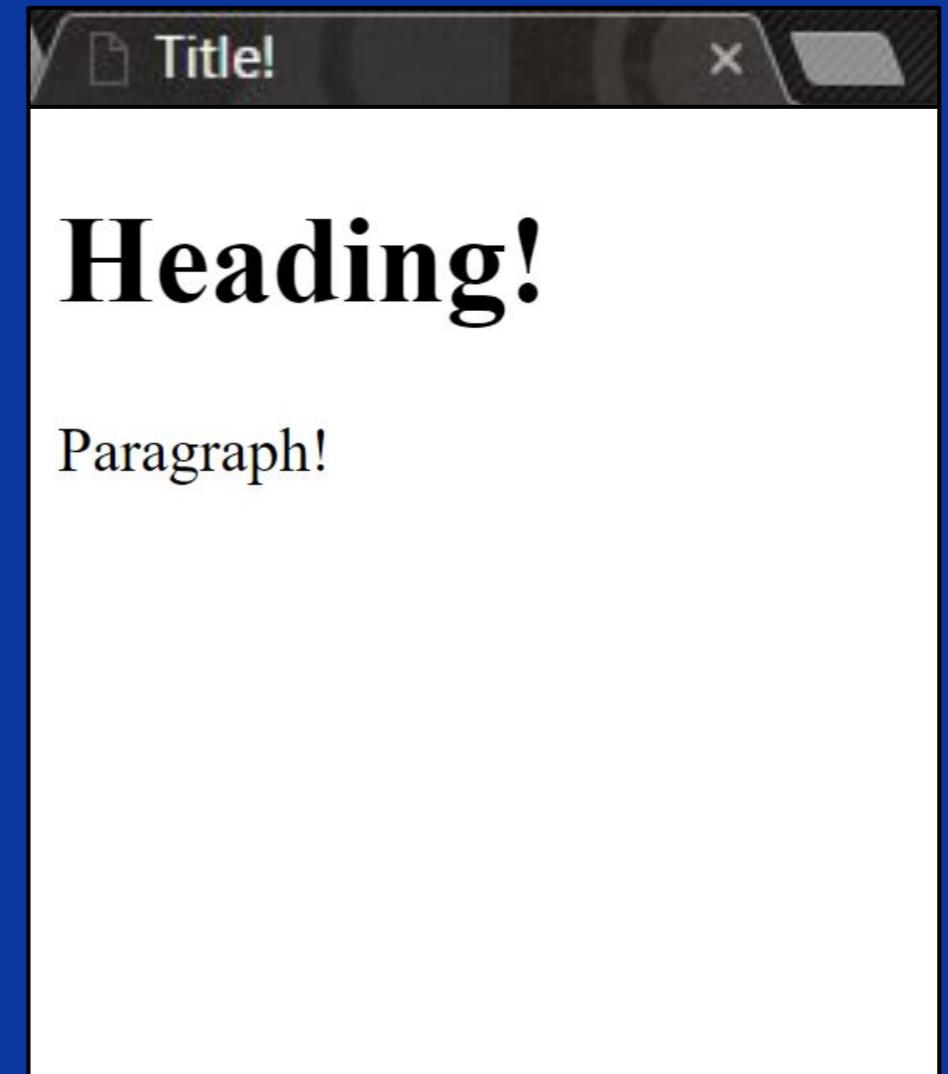
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



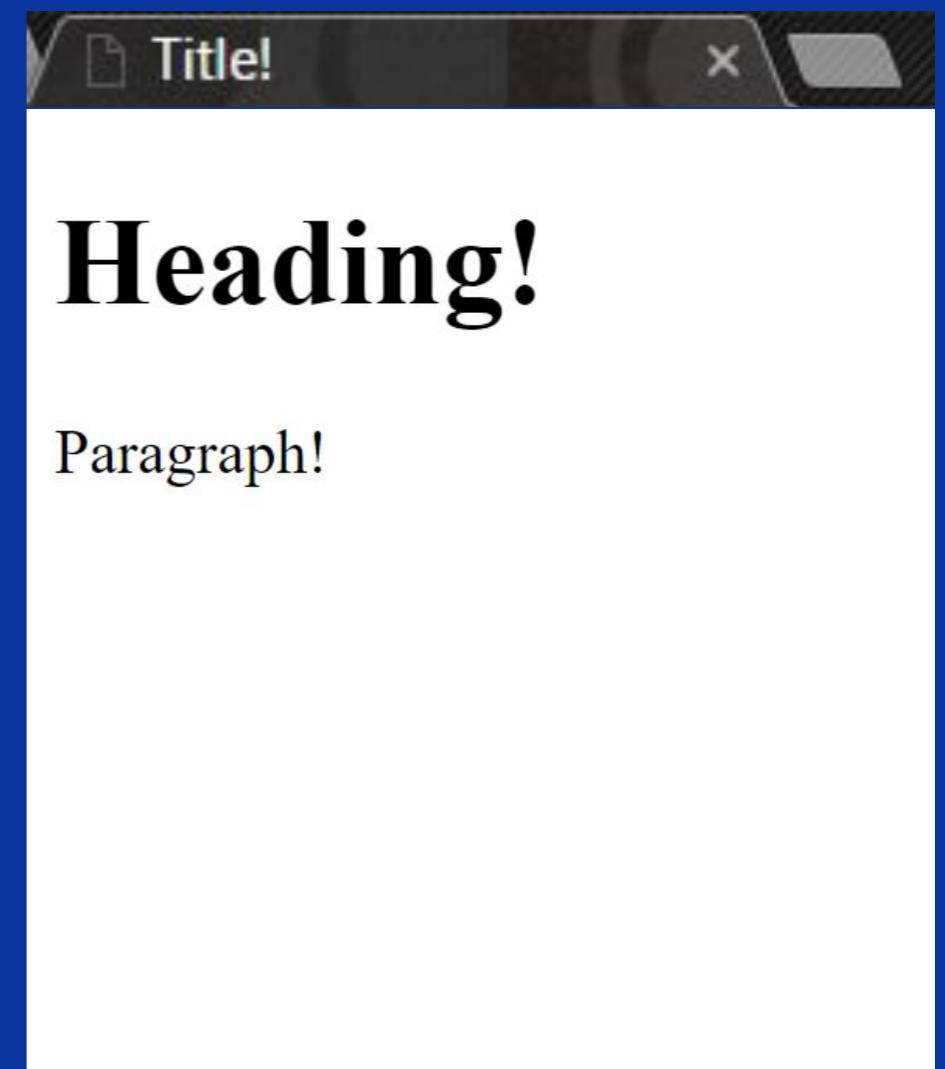
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



HTML Element

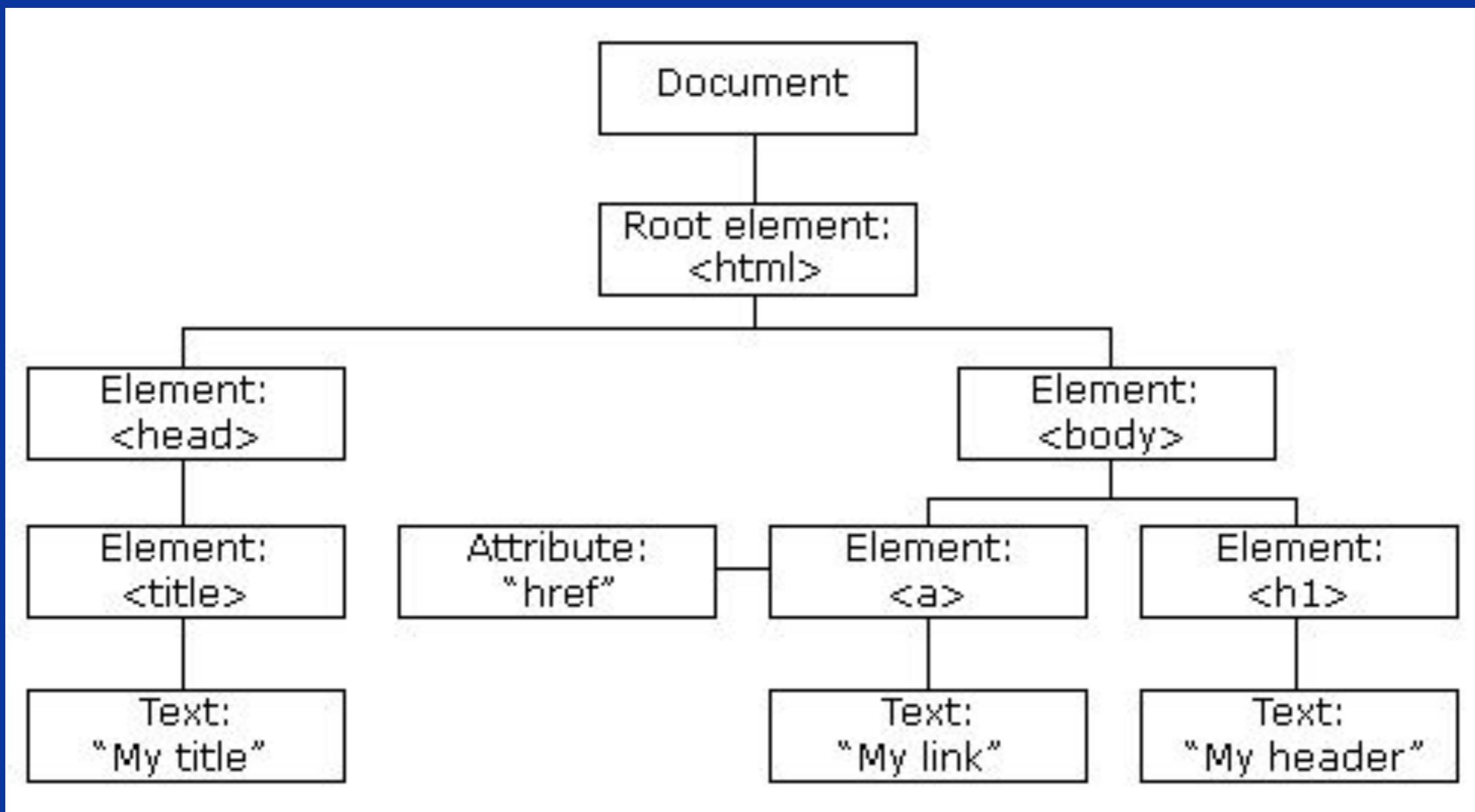
<html>  Opening tag

content goes here

</html>  Closing tag

hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



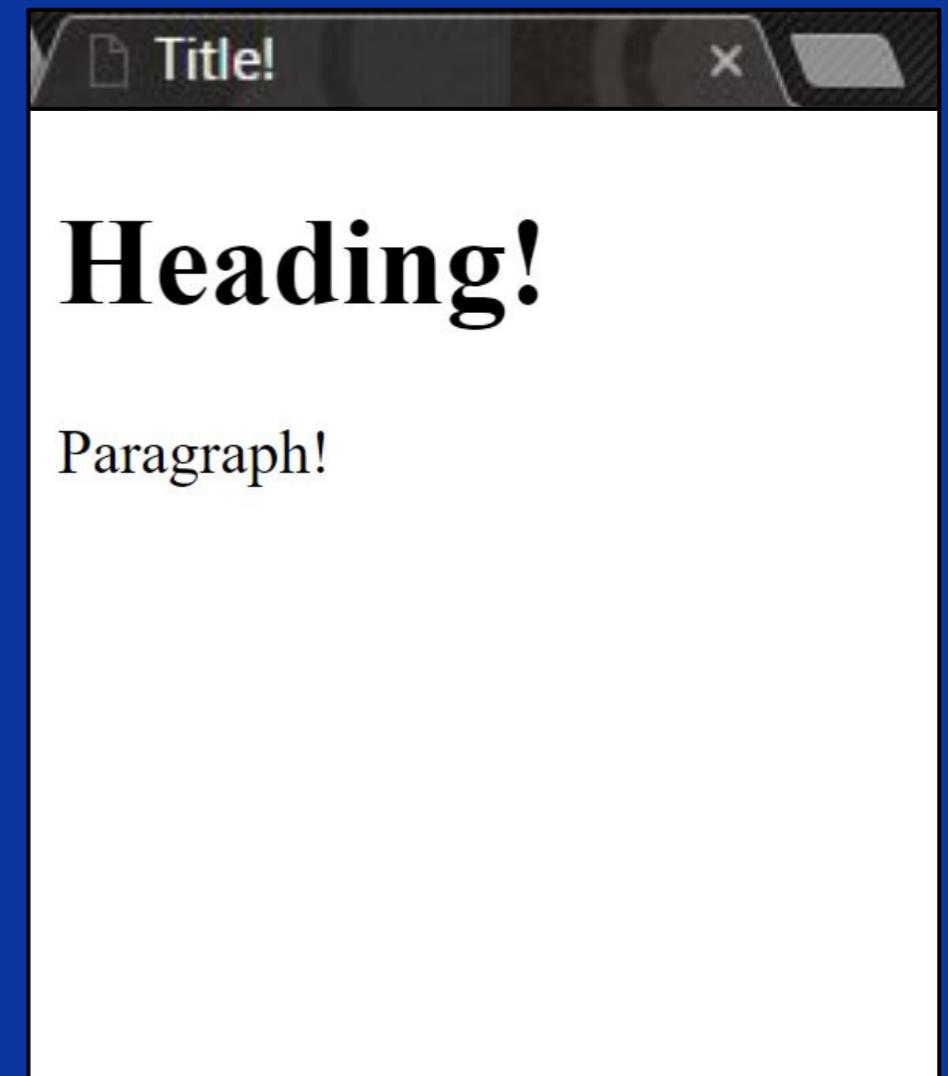
Nesting

~~<a>~~
~~~~
~~~~
~~~~



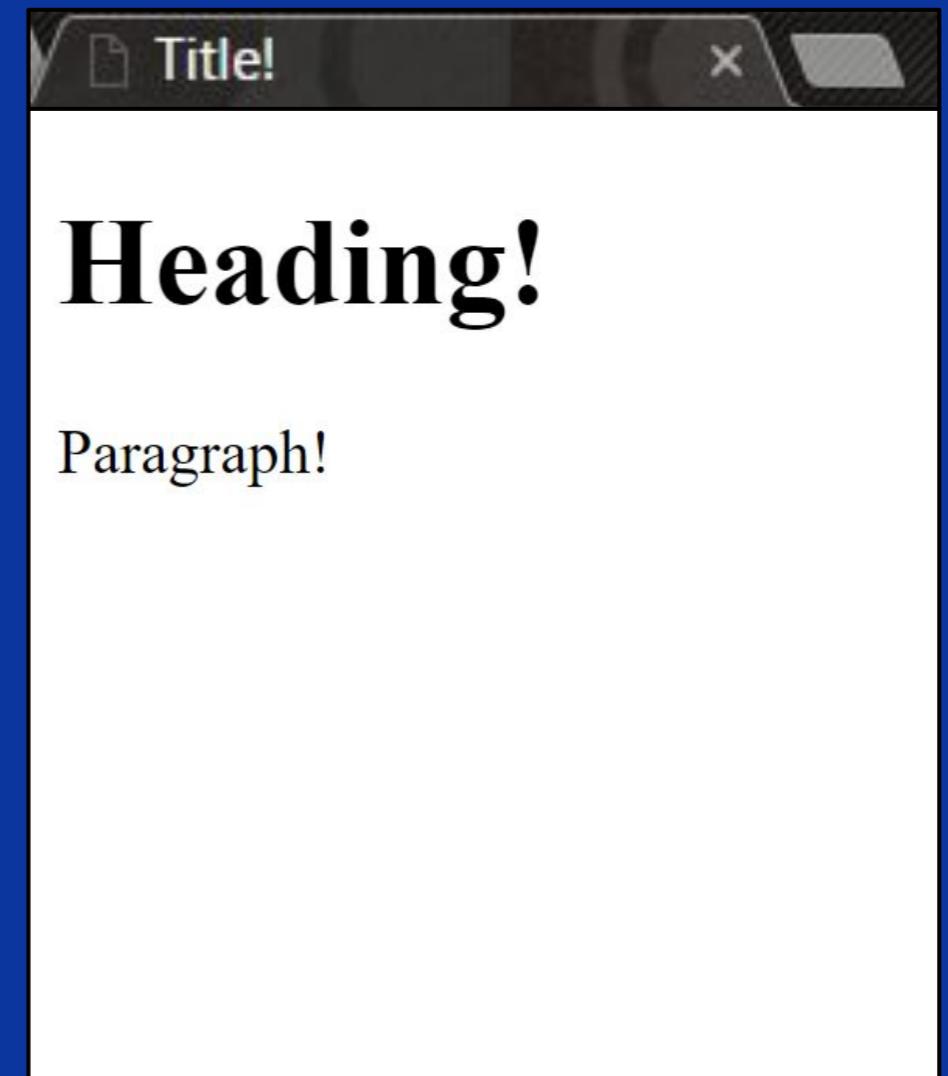
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



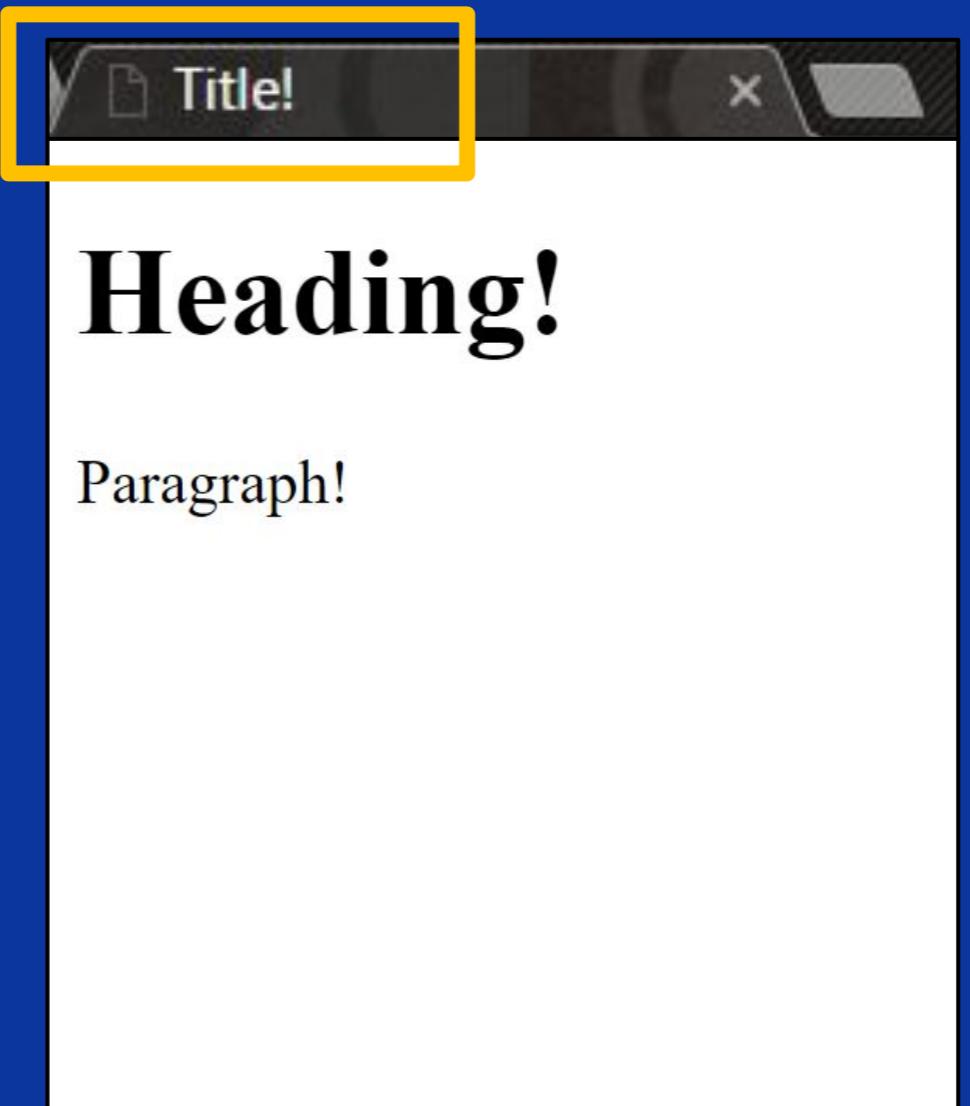
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



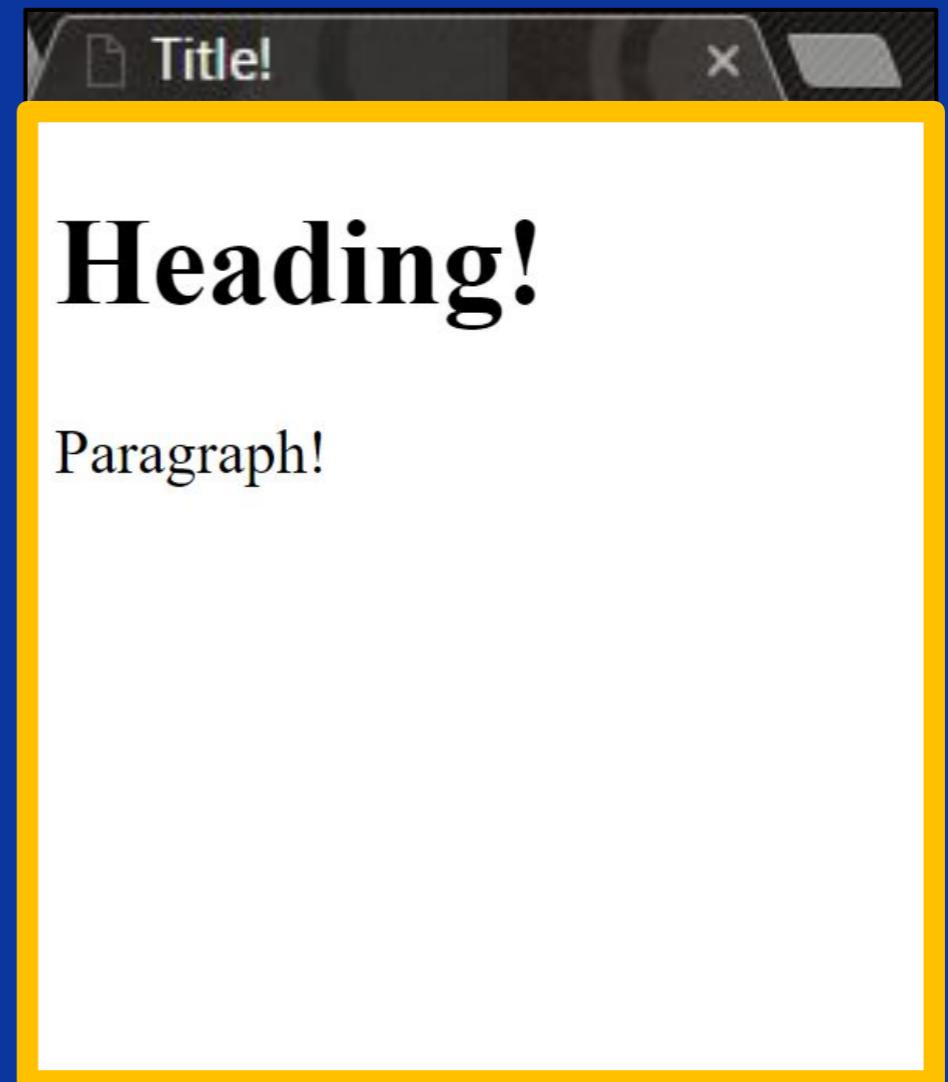
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



ARCHIVE
WHAT IF?
BLAG
STORE
ABOUT



A WEBCOMIC OF ROMANCE,
SARCASM, MATH, AND LANGUAGE.

THING EXPLAINER IS AVAILABLE AT: [AMAZON](#), [BARNES & NOBLE](#), [INDIE BOUND](#),
[HUDSON](#)

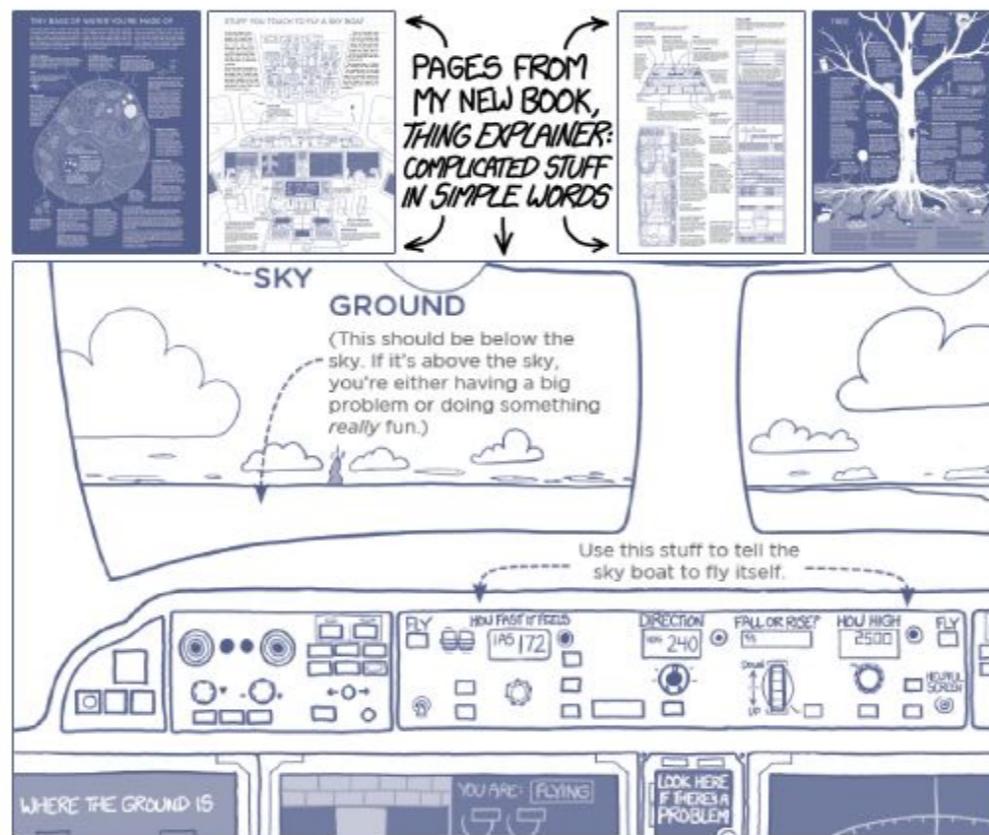
TAGS

<| < PREV RANDOM NEXT > |>

<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?

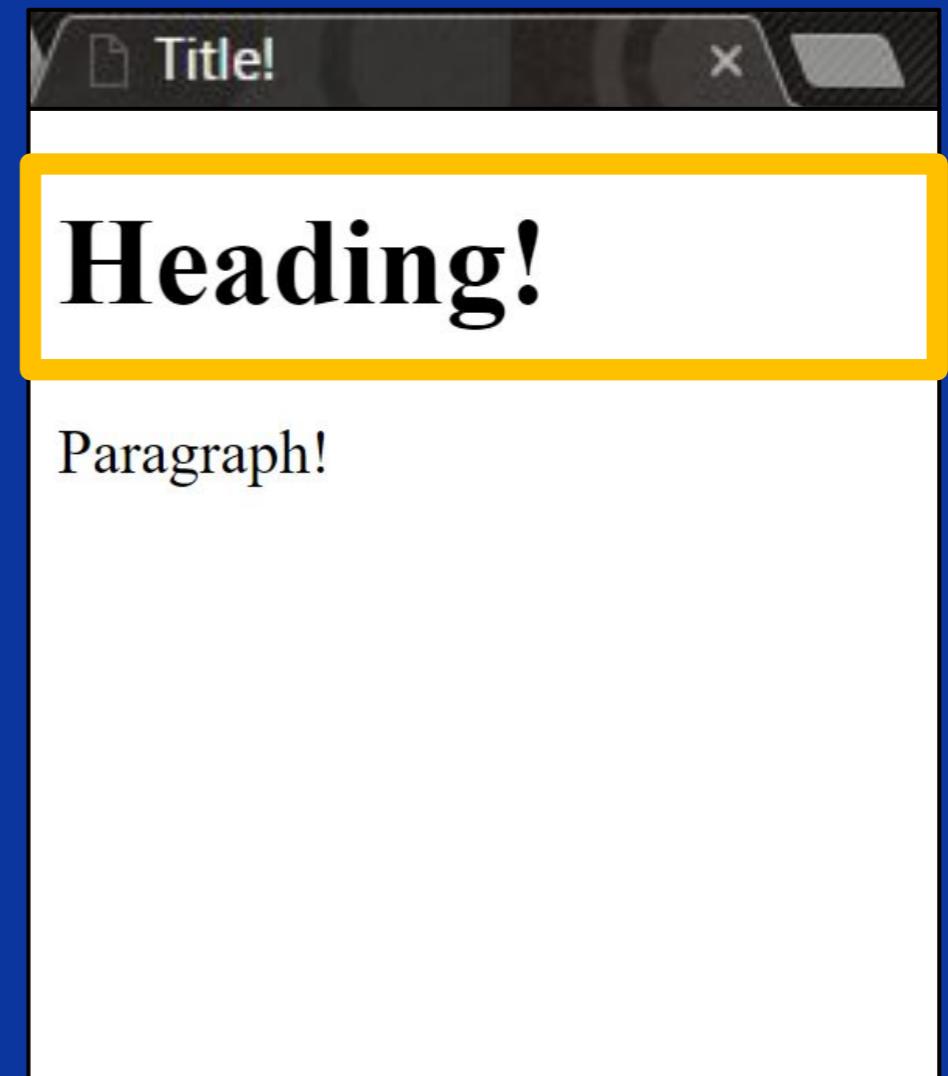
<| < PREV RANDOM NEXT > |>

PERMANENT LINK TO THIS COMIC: [HTTP://XKCD.COM/1144/](http://xkcd.com/1144/)
IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTP://IMGS.XKCD.COM/COMICS/TAGS.PNG](http://imgs.xkcd.com/comics/tags.png)



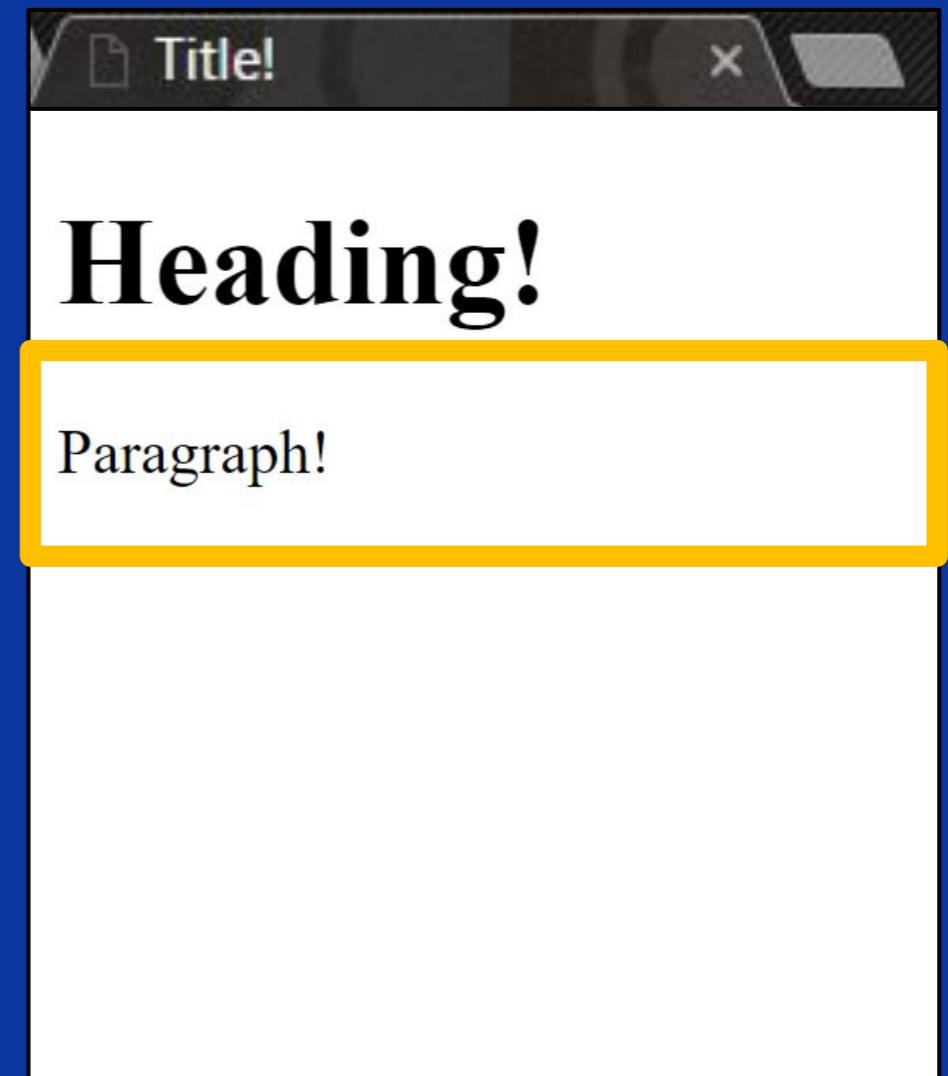
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



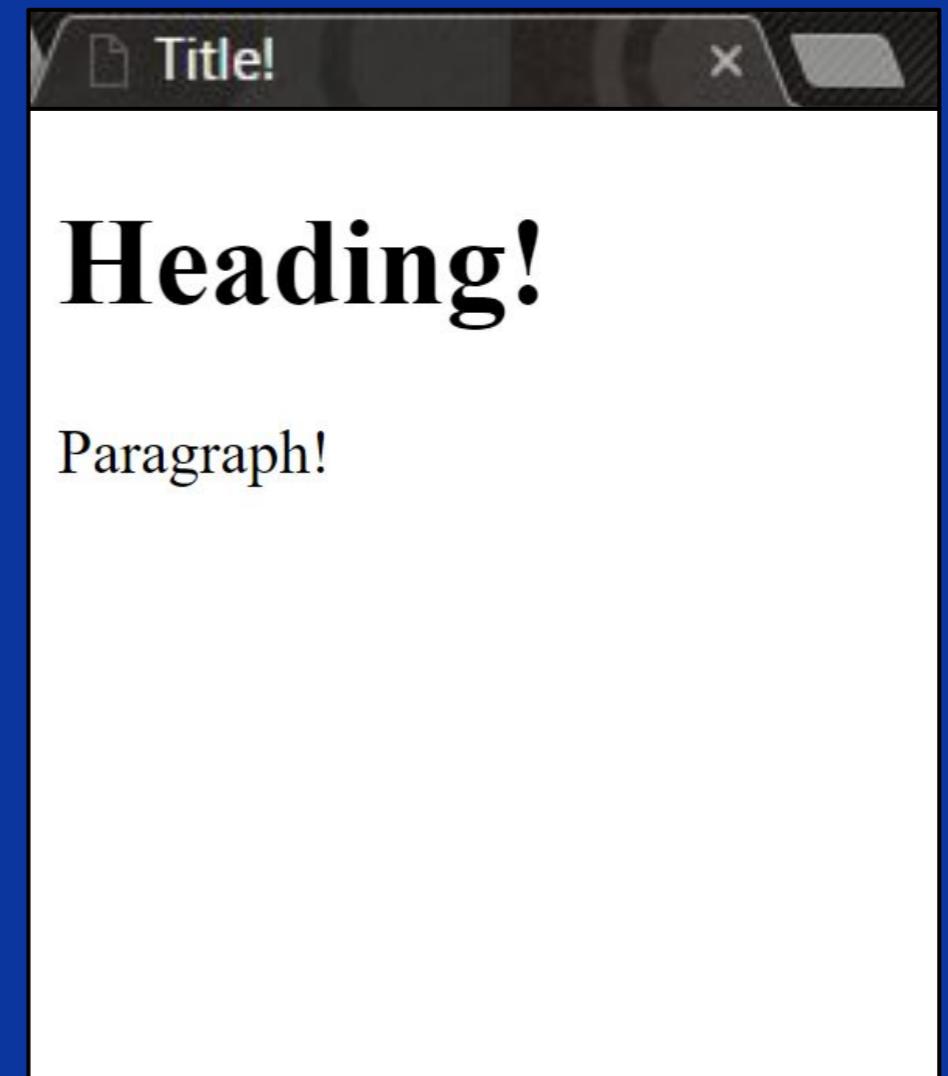
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



Basic HTML Tags

<html>

Root of HTML Document

<head>

Info about Document

<body>

Document Body

<h1>, <h2>, <h3>, ...

Header tags

<p>

Paragraph tag

<div>

Generic block section tag

Generic inline section tag

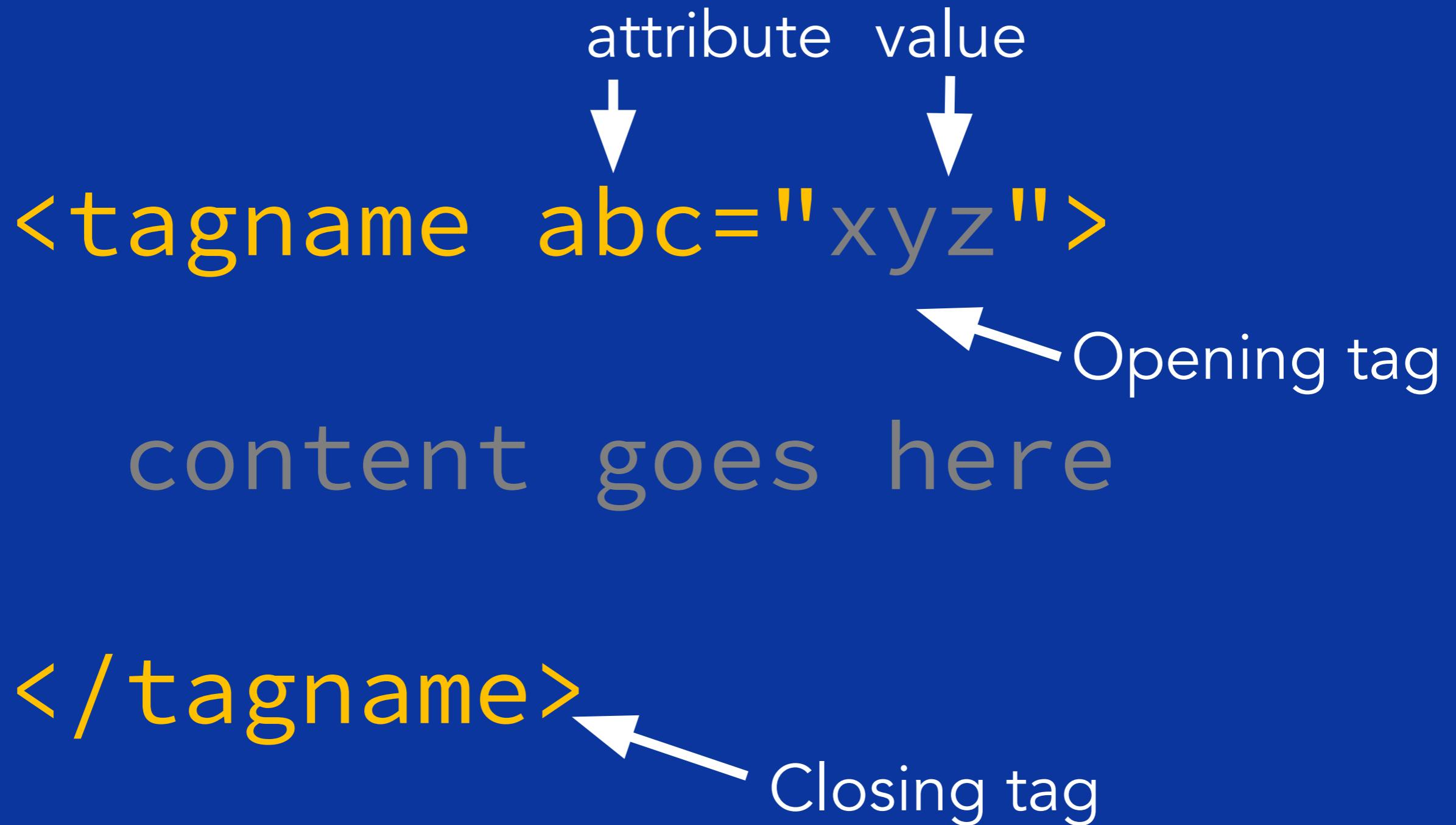
HTML Attributes

<tagname>
 ↑
 Opening tag

content goes here

</tagname>
 ↑
 Closing tag

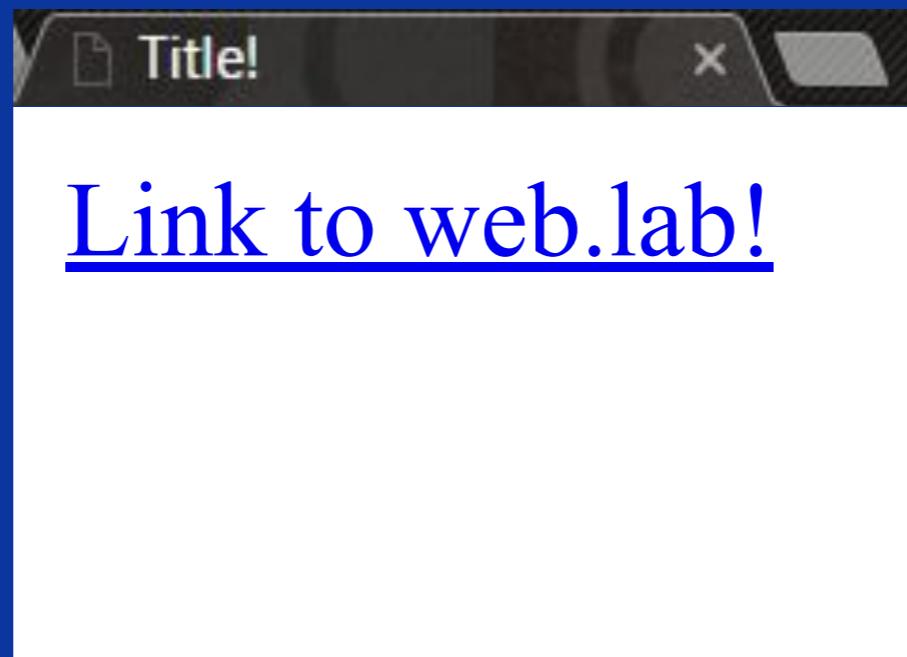
HTML Attributes



Inserting Links

<a>

Link



HTML Attributes

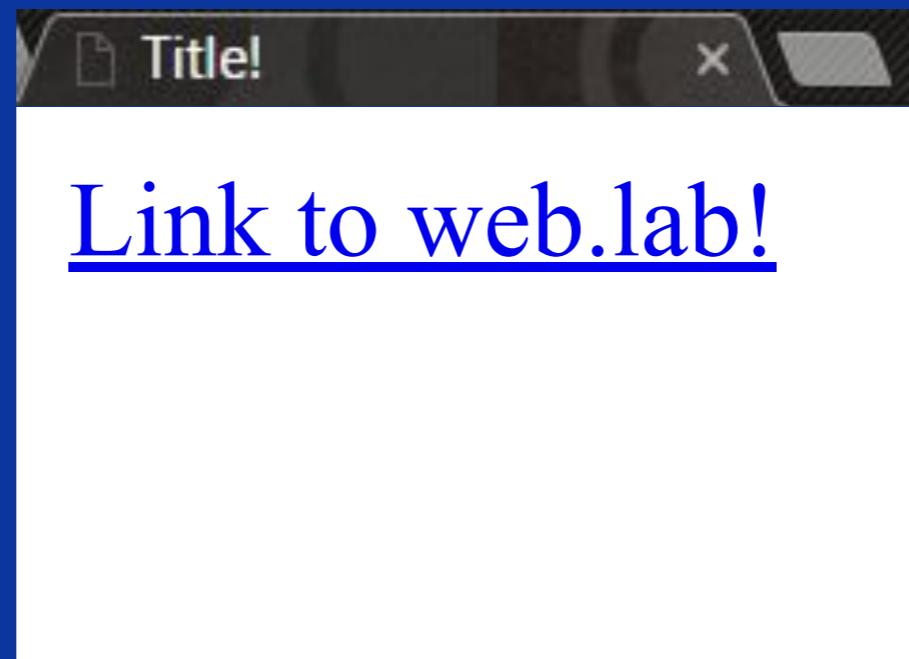
The diagram illustrates the structure of an HTML anchor tag (`<a>`) with the following annotations:

- attribute**: Points to the `href` attribute of the opening tag.
- value**: Points to the value of the `href` attribute, which is `"link here"`.
- Opening tag**: Points to the closing bracket of the opening tag (`>`).
- content goes here**: A placeholder text indicating where the link content would be placed.
- Closing tag**: Points to the opening bracket of the closing tag (``).

```
<a href="link here">  
    content goes here  
</a>
```

Inserting Links

```
<a href="http://weblab.mit.edu">Link to web.lab!</a>
```



Inserting Images

Image

Inserting Images

```
</img>
```

Inserting Images

```
</img>
```

Inserting Images

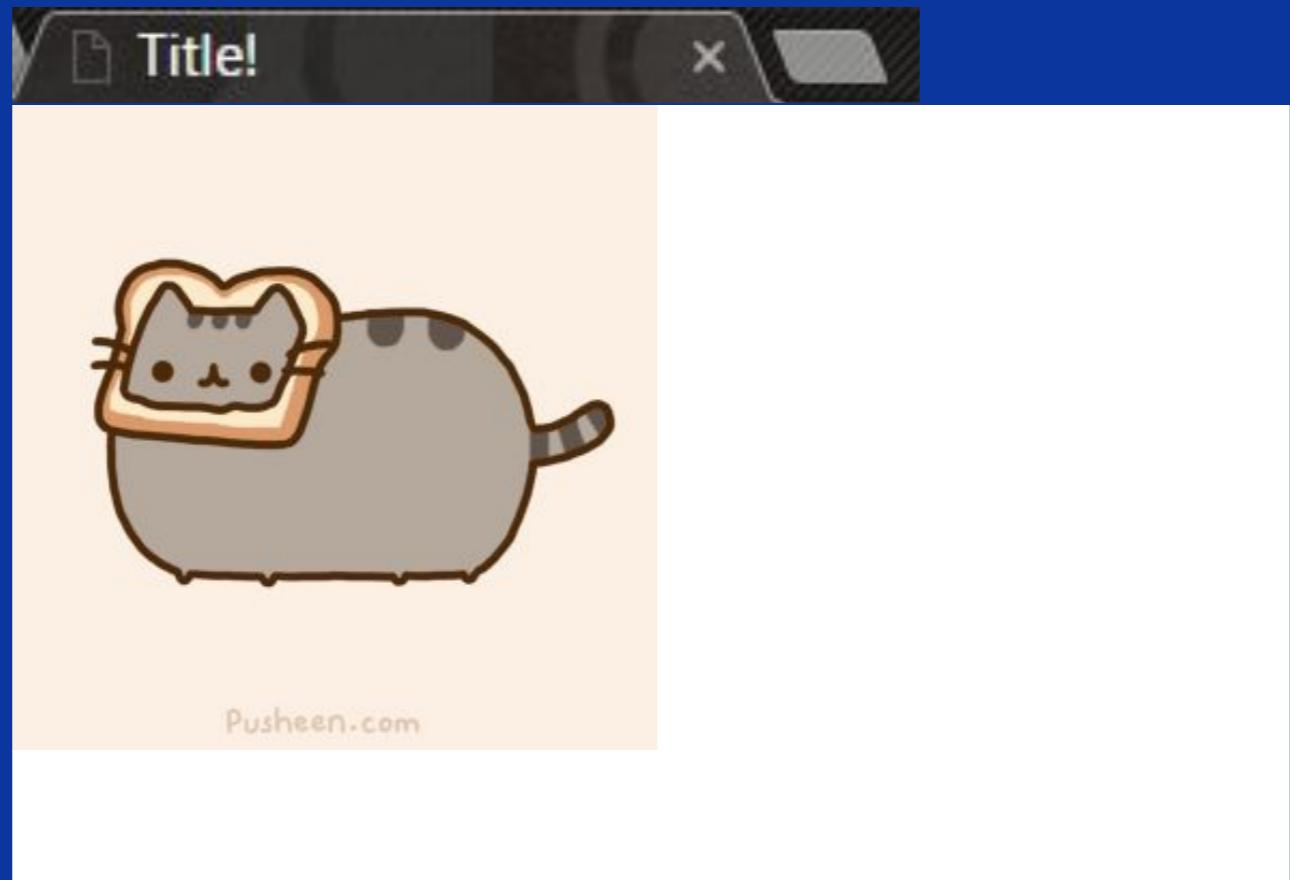
```

```

Inserting Images

```

```



Inserting Images

```

```

app/
hello.html
pusheen.gif

Inserting Images

~~~~



app/  
hello.html  
images/  
pusheen.gif

# Lists

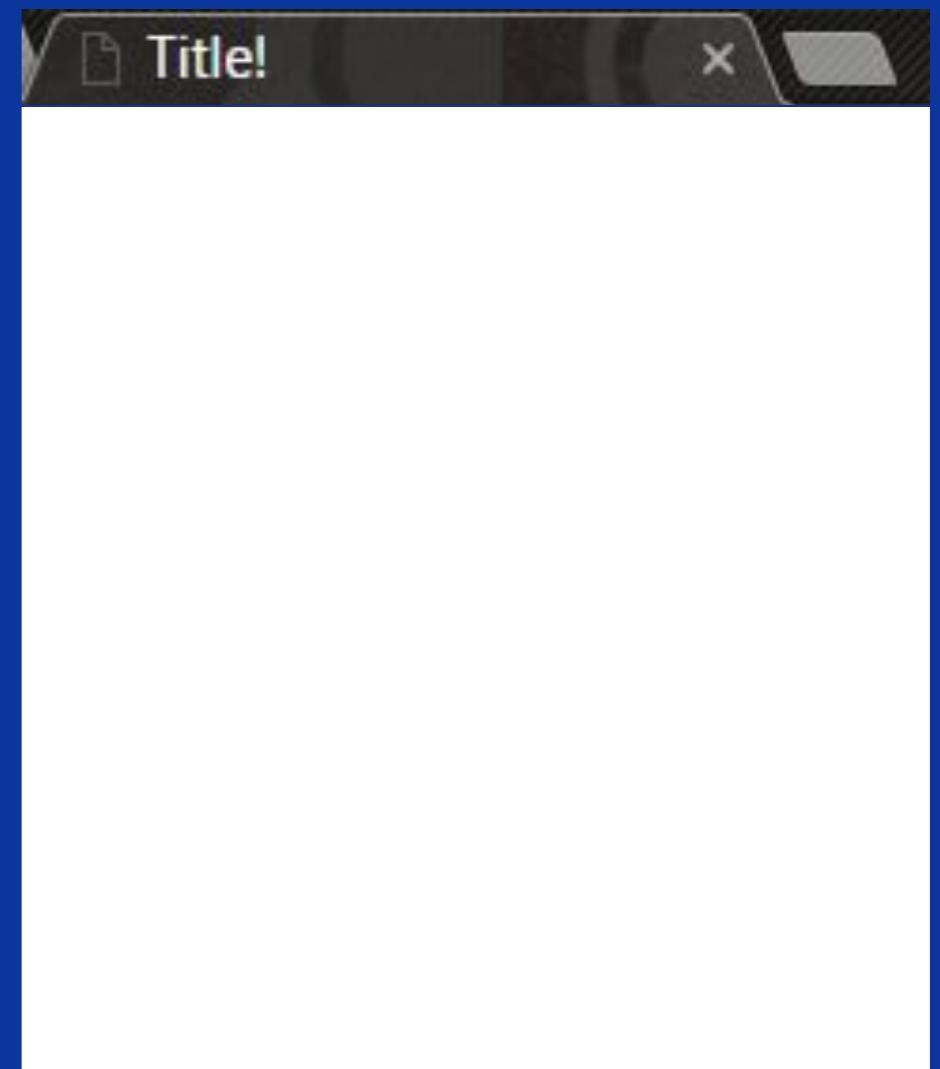
<ol> Ordered List (1, 2, 3...)

<ul> Unordered List (bullets)

<li> List Item

## hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <ul>
      </ul>
    </body>
  </html>
```



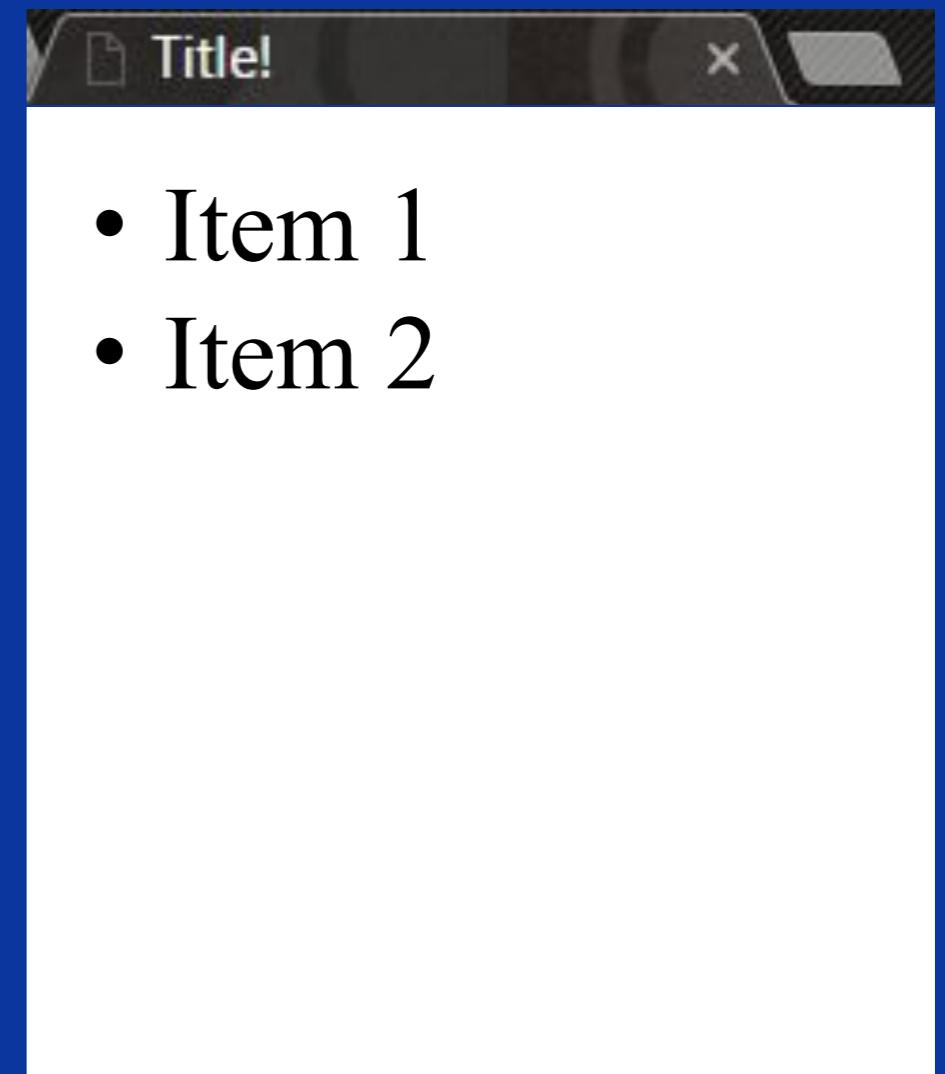
## hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <ul>
      <li>Item 1</li>
    </ul>
  </body>
</html>
```



## hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
    </ul>
  </body>
</html>
```



# div & span

<div>

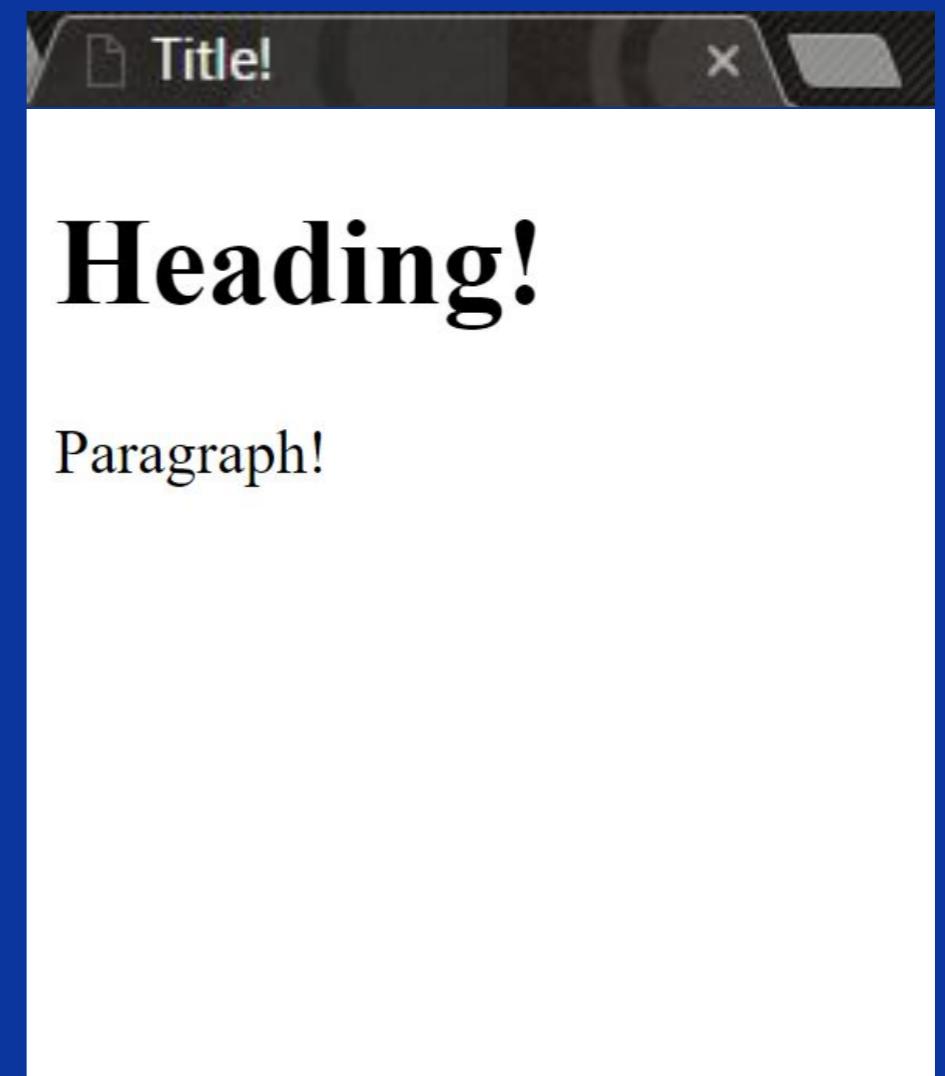
Block Section in Document

<span>

Inline Section in Document

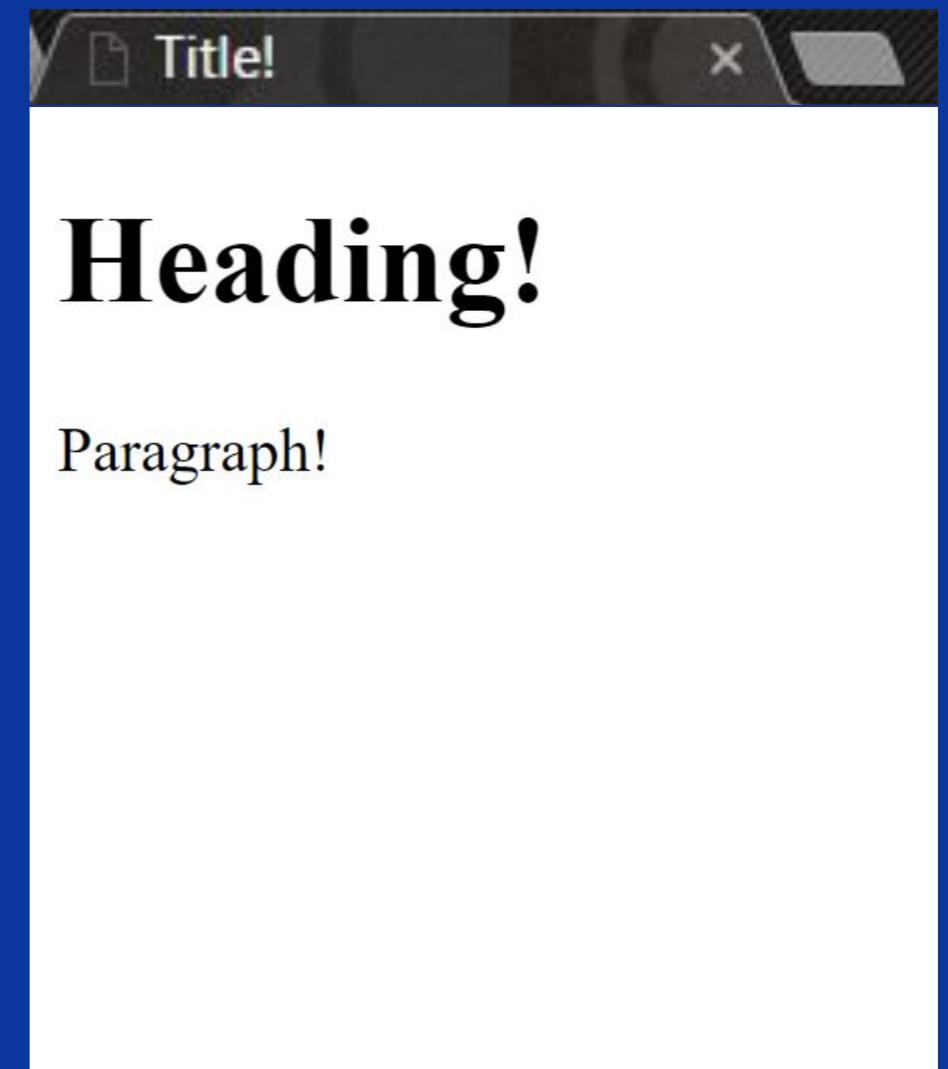
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```



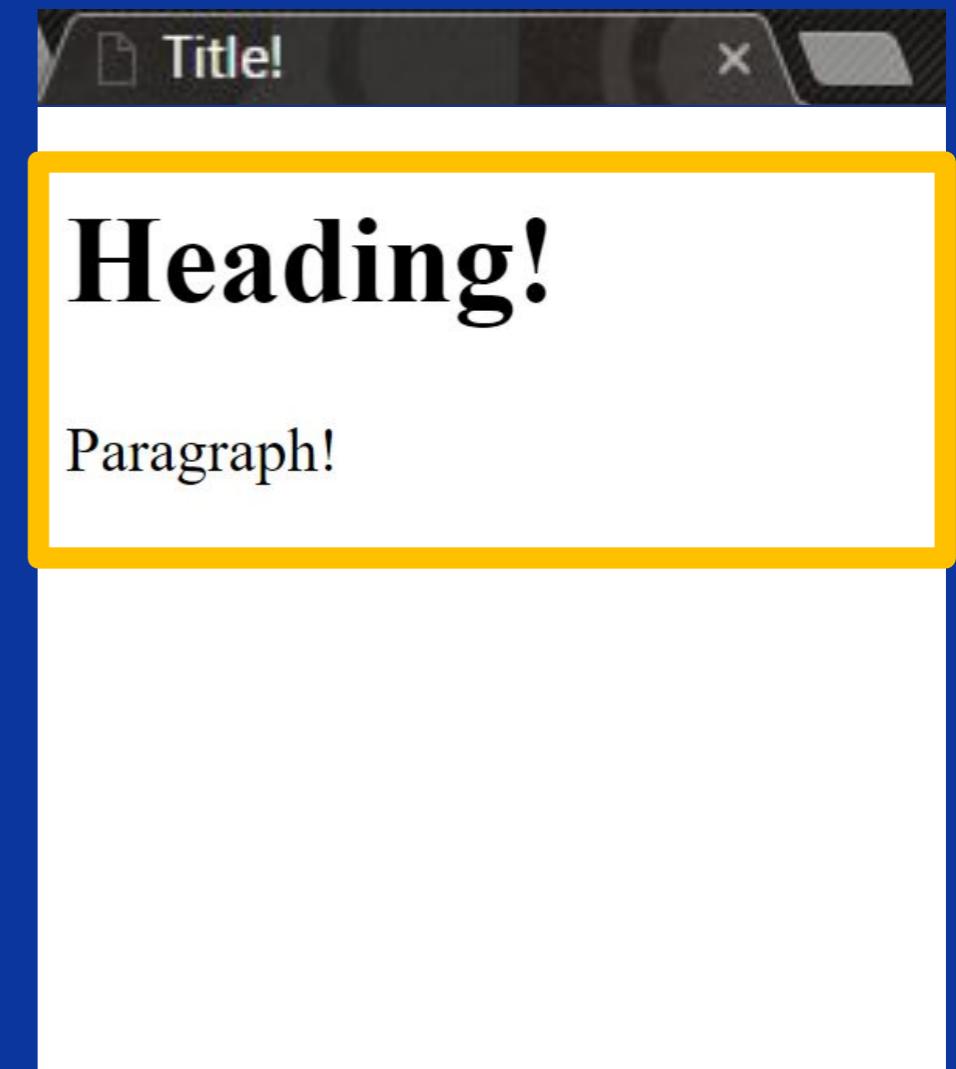
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <div>
      <h1>Heading!</h1>
      <p>Paragraph!</p>
    </div>
  </body>
</html>
```



hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <div>
      <h1>Heading!</h1>
      <p>Paragraph!</p>
    </div>
  </body>
</html>
```



so many tags...

<ul></ul>

<span></span>

so many tags...

<nav></nav>

# the real way to learn Web Dev

Step 1:  
Google!

html span

All Videos Images Books Shopping More Settings Tools

About 209,000,000 results (0.66 seconds)

The **span** tag is like the **div** tag. It has no meaning at all and is mostly used for styling by using an id or class. The difference between the two is that **div** is a block element, it's on a separate line. **span** however is an inline element, meaning that it can be on a line with other elements. Jan 4, 2013

[What does the span tag do? | Codecademy](#)  
[https://www.codecademy.com/en/forum\\_questions/502ad0ea558dfe0002026d69](https://www.codecademy.com/en/forum_questions/502ad0ea558dfe0002026d69)

[About this result](#) [Feedback](#)

People also ask

Why is span used in HTML?  
Why is SPAN tag used in HTML?  
Is a span a block element?  
What is a div tag in HTML?

[Feedback](#)

**HTML span tag - W3Schools**  
[https://www.w3schools.com/tags/tag\\_span.asp](https://www.w3schools.com/tags/tag_span.asp) ▾  
Definition and Usage. The <span> tag is used to group inline-elements in a document. The <span> tag provides no visual change by itself. The <span> tag provides a way to add a hook to a part of a text or a part of a document.  
[HTML DOM Span Object](#) · Try it Yourself · Strike

**HTML span Attribute - W3Schools**  
[https://www.w3schools.com/tags/att\\_span.asp](https://www.w3schools.com/tags/att_span.asp) ▾  
<table> <colgroup> <col span="2" style="background-color:red"> <col style="background-color:yellow"> </colgroup> <tr> <th>ISBN</th> <th>Title</th> <th>Price</th> </tr> <tr> <td>3476896</td> <td>My first HTML</td> <td>\$53</td> </tr> </table>. Try it Yourself » ...

**<span> - HTML | MDN**  
[https://developer.mozilla.org/.../HTML/HTML\\_elements\\_reference](https://developer.mozilla.org/.../HTML/HTML_elements_reference) ▾  
Sep 19, 2017 - The **HTML <span>** element is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the `class` or `id` attributes), or because they share attribute values, such as `lang`.

# the real way to learn Web Dev

## Step 2: Learn.

The **span** tag is like the div tag. It has no meaning at all and is mostly used for styling by using an id or class. The difference between the two is that div is a block element, it's on a separate line. **span** however is an inline element, meaning that it can be on a line with other elements. Jan 4, 2013

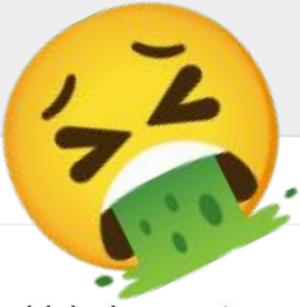
[What does the span tag do? | Codecademy](#)

[https://www.codecademy.com/en/forum\\_questions/502ad0ea558dfe0002026d69](https://www.codecademy.com/en/forum_questions/502ad0ea558dfe0002026d69)

# the real way to learn Web Dev

MDN web docs  Technologies ▾ References & Guides ▾ Feedback ▾  Search MDN Sign in

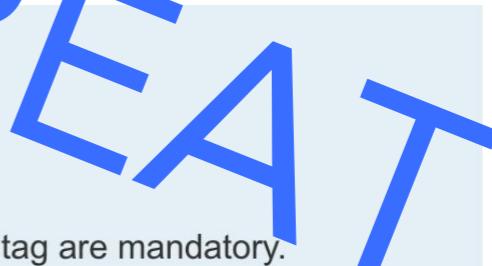
## <span>

Web technology for developers > HTML: Hypertext Markup Language > HTML elements reference > <span> English ▾

On this Page

- Attributes
- Example 1
- Example 2
- Specifications
- Browser compatibility
- See also

The **HTML <span> element** is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the `class` or `id` attributes), or because they share attribute values, such as `lang`. It should be used only when no other semantic element is appropriate. <span> is very much like a <div> element, but <div> is a **block-level element** whereas a <span> is an **inline element**.

**W3SCHOOLS** 

| Content categories   | Flow content, phrasing content.                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------|
| Permitted content    | Phrasing content.                                                                                    |
| Tag omission         | None, both the starting and ending tag are mandatory.                                                |
| Permitted parents    | Any element that accepts <b>phrasing content</b> , or any element that accepts <b>flow content</b> . |
| Permitted ARIA roles | Any                                                                                                  |
| DOM interface        | <code>HTMTSpanElement</code> (before HTML5, the interface was                                        |

Related Topics

- <a>
- <abbr>
- <b>

<https://developer.mozilla.org/en-US/docs/Web>

Why not...

“just use <div>”

Why not...

“just use <div>”



why could this be bad?

# <div>

## <div>: The Content Division element

Web technology for developers > HTML: HyperText Markup Language > HTML elements reference > <div>: The Content Division element

English (US) Change language

### Jump to section

[Attributes](#)  
[Usage notes](#)  
[Examples](#)  
[Specifications](#)  
[Browser compatibility](#)  
[See also](#)

The **HTML Content Division element** (`<div>`) is the generic container for flow content. It has no effect on the content or layout until styled in some way using CSS (e.g. styling is directly applied to it, or some kind of layout model like Flexbox is applied to its parent element).

 **HTML Demo: <div>** Reset

**HTML** **CSS**

```
1 <div class="warning">
2   Beware of the leopard</p>
5 </div>
6
```

**Output**

![An intimidating leopard.](/media/examples/leopard.jpg)

### Related Topics

[<blockquote>](#)  
[<dd>](#)  
[<div>](#)  
[<dl>](#)  
[<dt>](#)  
[<figcaption>](#)  
[<figure>](#)  
[<hr>](#)  
[<li>](#)

As a "pure" container, the `<div>` element does not inherently represent anything. Instead, it's used to group content so it can be easily styled using the `class` or `id` attributes, marking a section of a document as being written in a different language (using the `lang` attribute), and

# <div>

As a "pure" container, the <div> element does not inherently represent anything. I

## Usage notes

- The <div> element should be used only when no other semantic element (such as <article> or <nav>) is appropriate.

## See also

- Semantic sectioning elements: <section>, <article>, <nav>, <header>, <footer>

there might be  
better options...

# Why Semantics?

a brief aside...

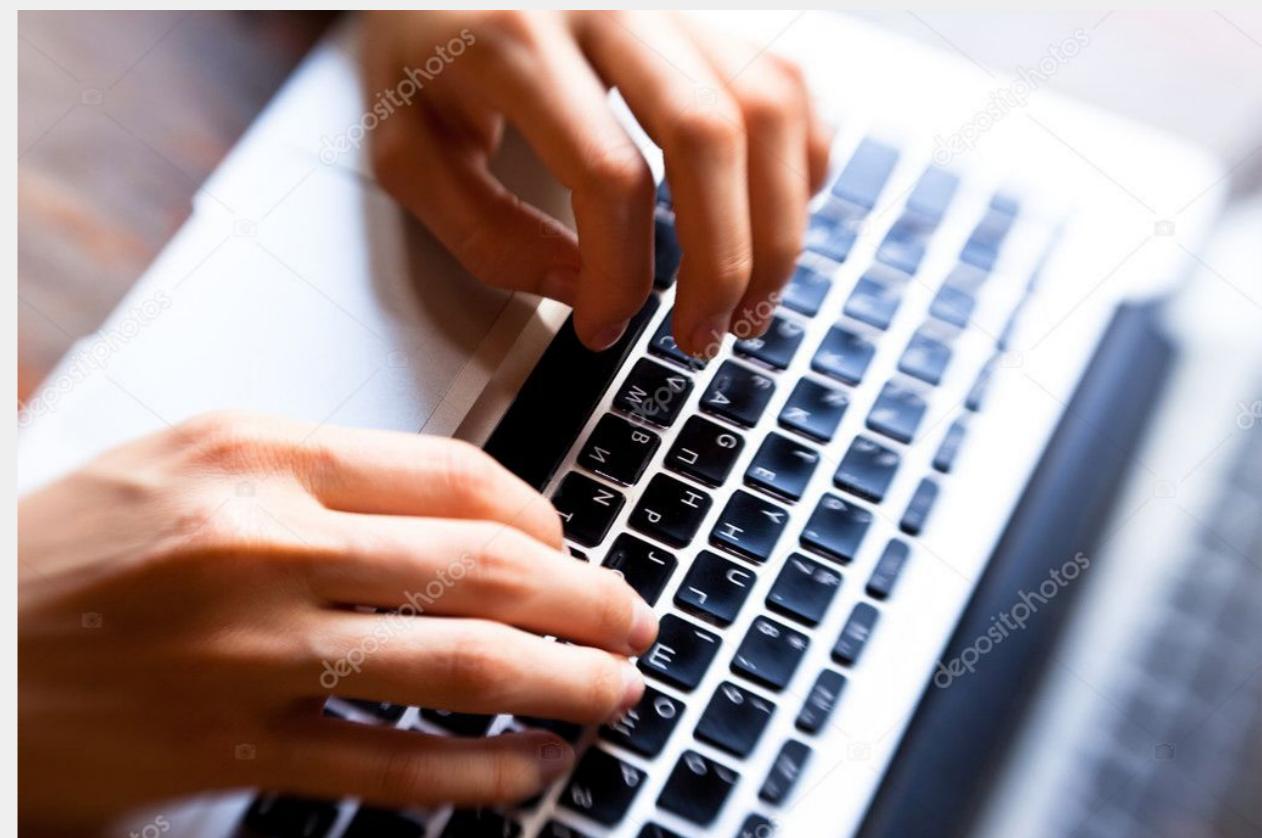
# Why Semantics?

## Screen readers



## **KB Navigation**

(why can you tab through an order or sign up screen?)

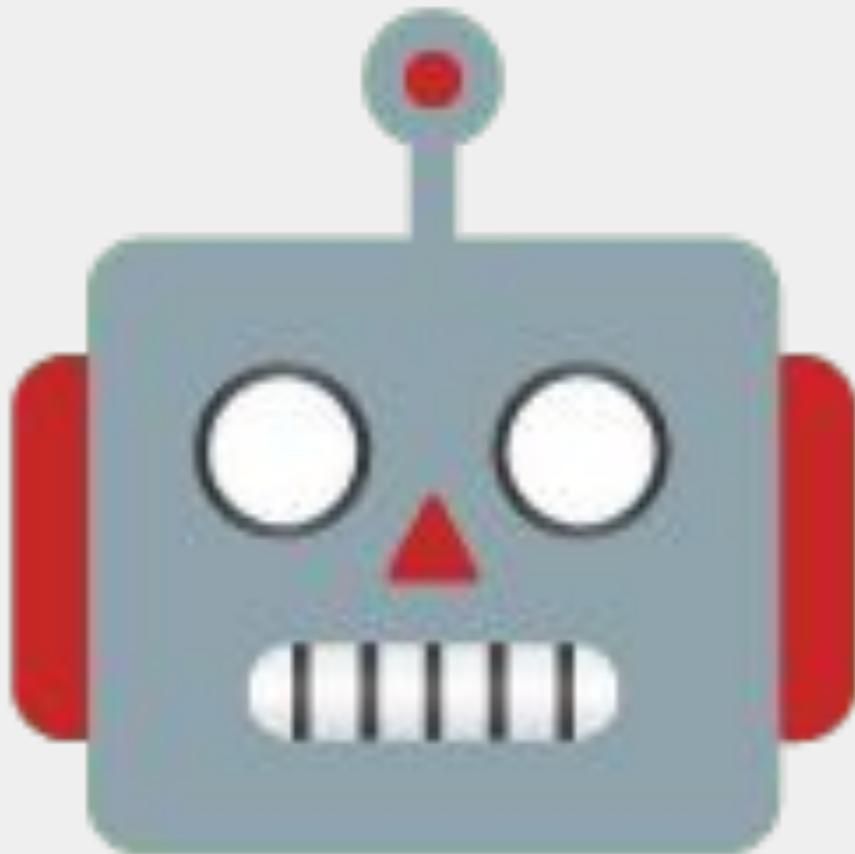


a brief aside...

# Why Semantics?

**Machine readable/friendly to  
web crawlers**  
(make your site more  
discoverable)

**Human readable**  
(easier to maintain in the  
future)



Don't just use <div>.  
MDN is your friend!

# HTML



## Hypertext Markup Language

The language your web browser uses to describe the content and structure of web pages

CSS

# CSS

## Cascading Style Sheets

The rules that tell your web browser how stuff looks

CSS = A list of descriptions

# Holidays Leaving You Short on ZZZ's?

What you eat plays a role in the quality of your sleep.

[7 eating tips to support healthy sleep →](#)



[CAFÉ HOURS](#)   [BREAKFAST](#)   [BRUNCH](#)   [LUNCH](#)   [DINNER](#)   [ICONS](#)   [MEAL PLANS](#)   [ABOUT YOUR FOOD](#)   [MENU MAIL](#)   [CONTACT US](#)   [FOOD RECOVERY VERIFIED](#)   [EXPRESS & PACK-OUT MEALS](#)   [IAP MEAL PLANS](#)

## CAFÉ HOURS

+ NEW VASSAR

SERVING DINNER FROM 5:00 PM TO 8:00 PM

+ THE HOWARD DINING HALL  
AT MASELLI

CLOSED

[Skip to main content](#)

**MIT**

**MIT**

## Primary Navigation

1. [Café](#)  
[Menus & Info](#)
2. [Wellness](#)  
[Nutrition & Sustainability](#)
3. [Catering](#)  
[Inspiration & Ordering](#)
4. [Connect](#)  
[Events & Contact](#)

[Menu Close](#)





**HTML**



**HTML + CSS**

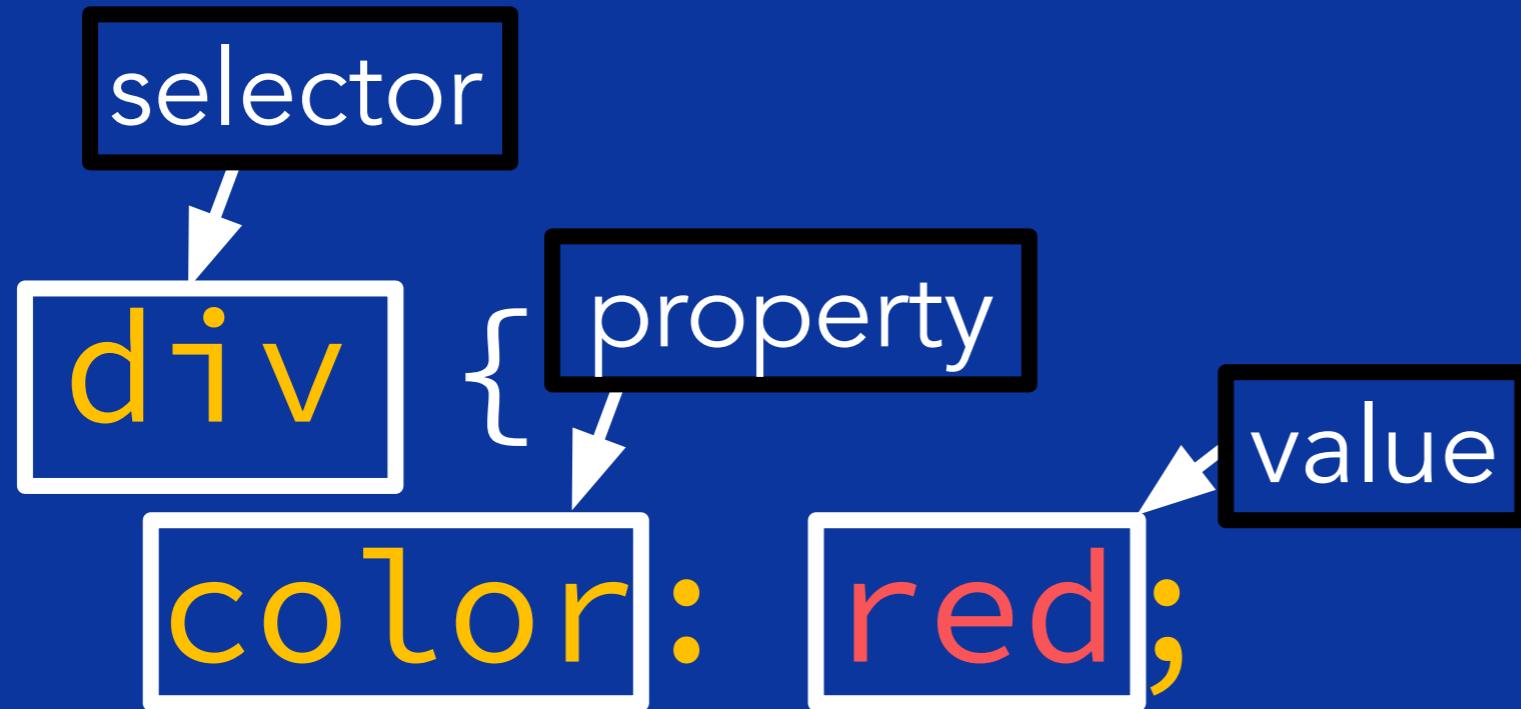


**HTML + a lot of CSS**

**CSS = A list of descriptions**

aka making things look pretty

# CSS ruleset

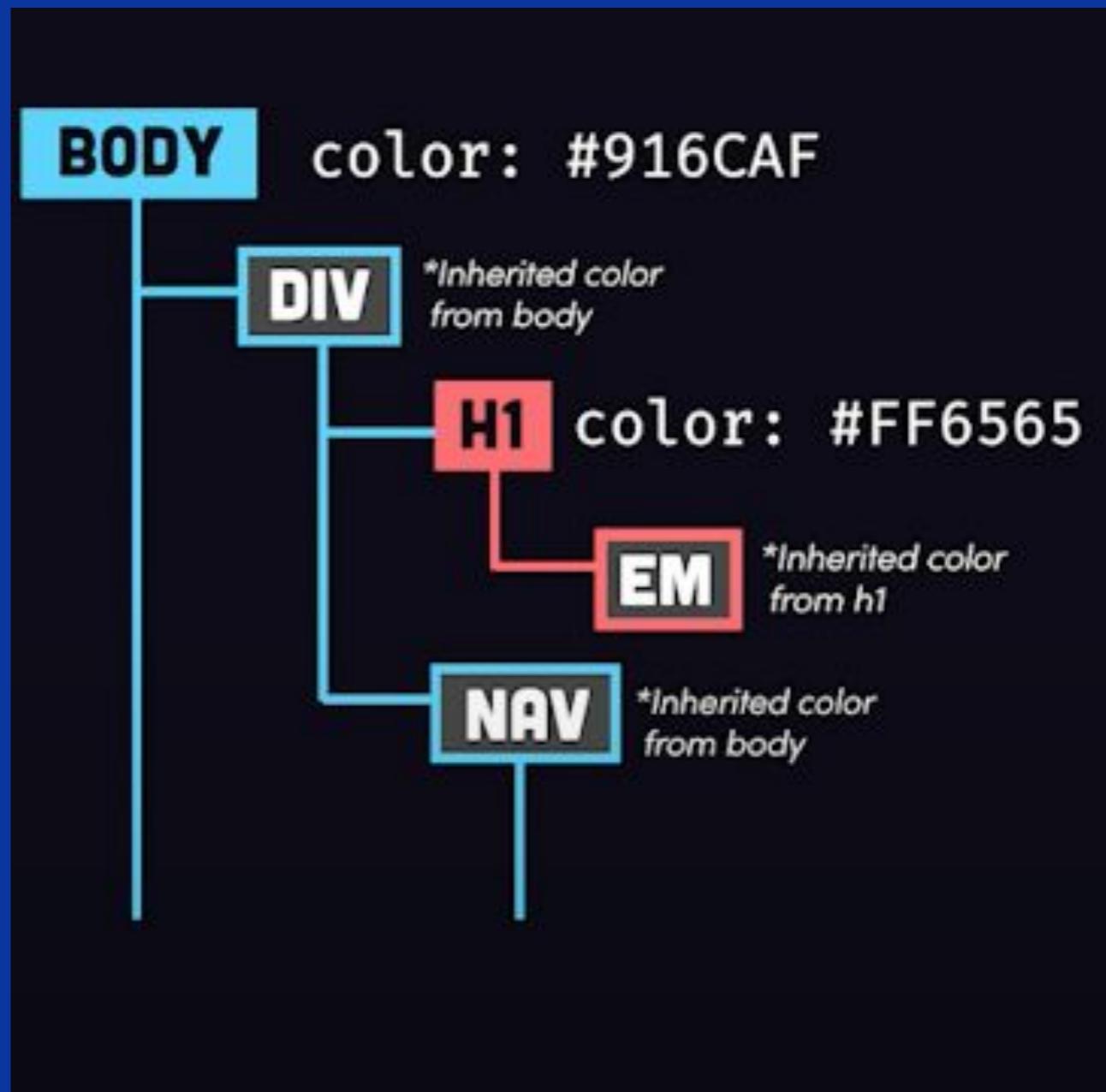
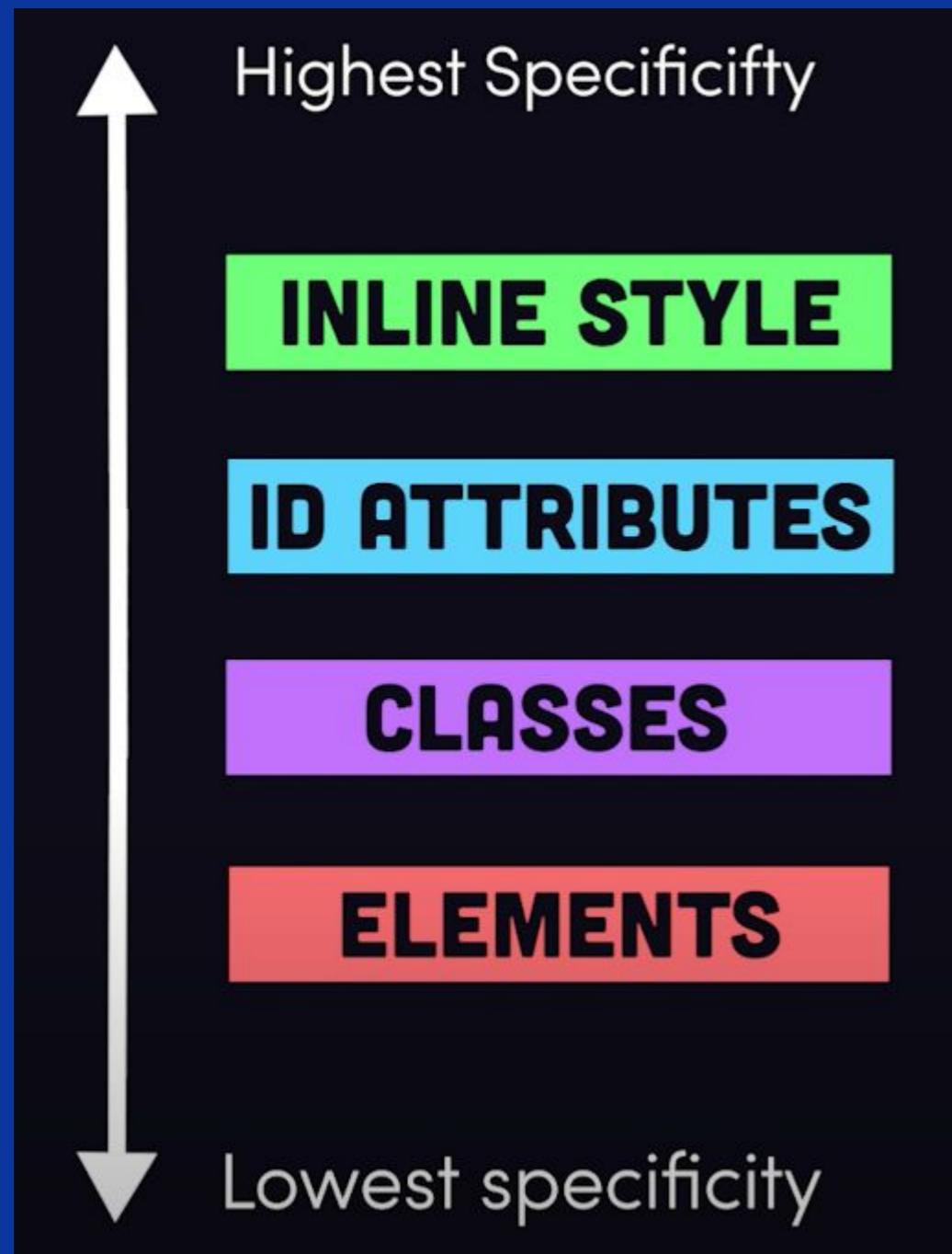


font-family: Arial;

font-size: 24pt;

}

# CSS Hierarchy



# em vs px vs pt vs %

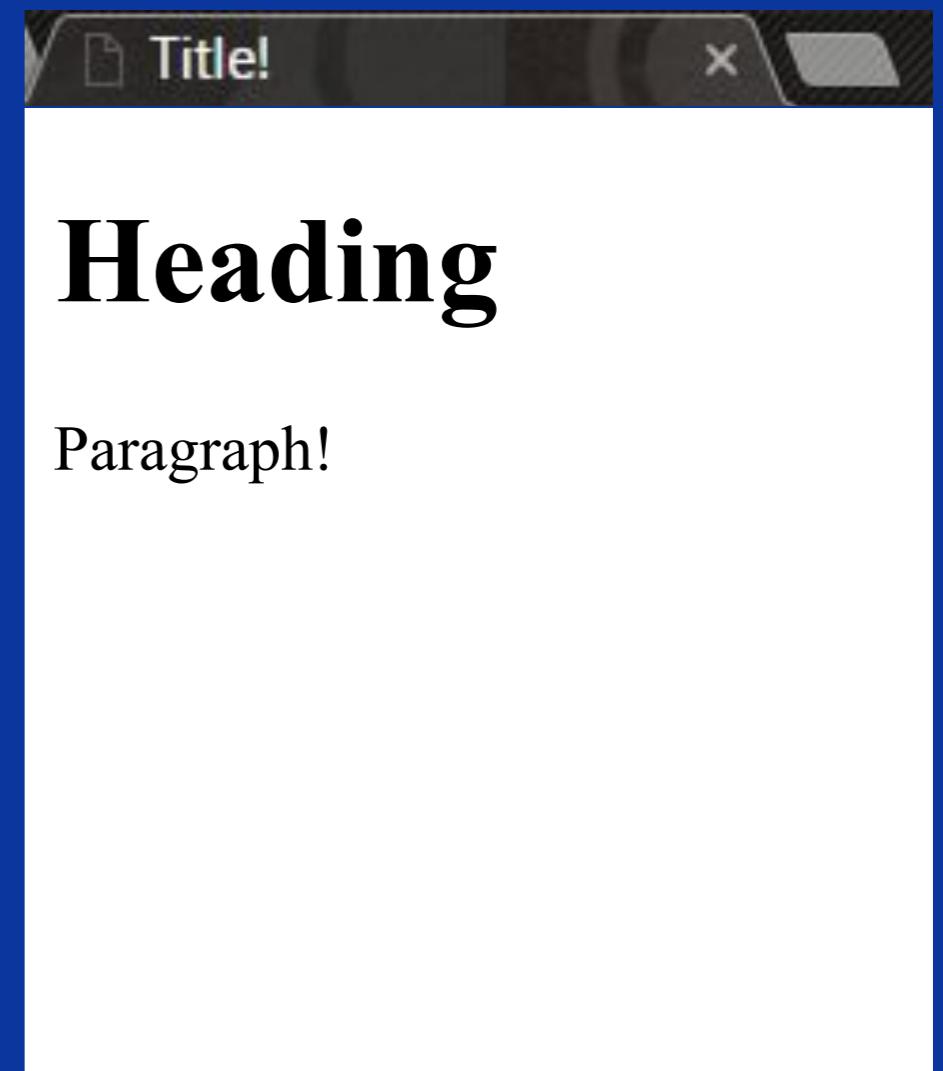
```
{
```

```
  font-size: 24pt;
```

```
}
```

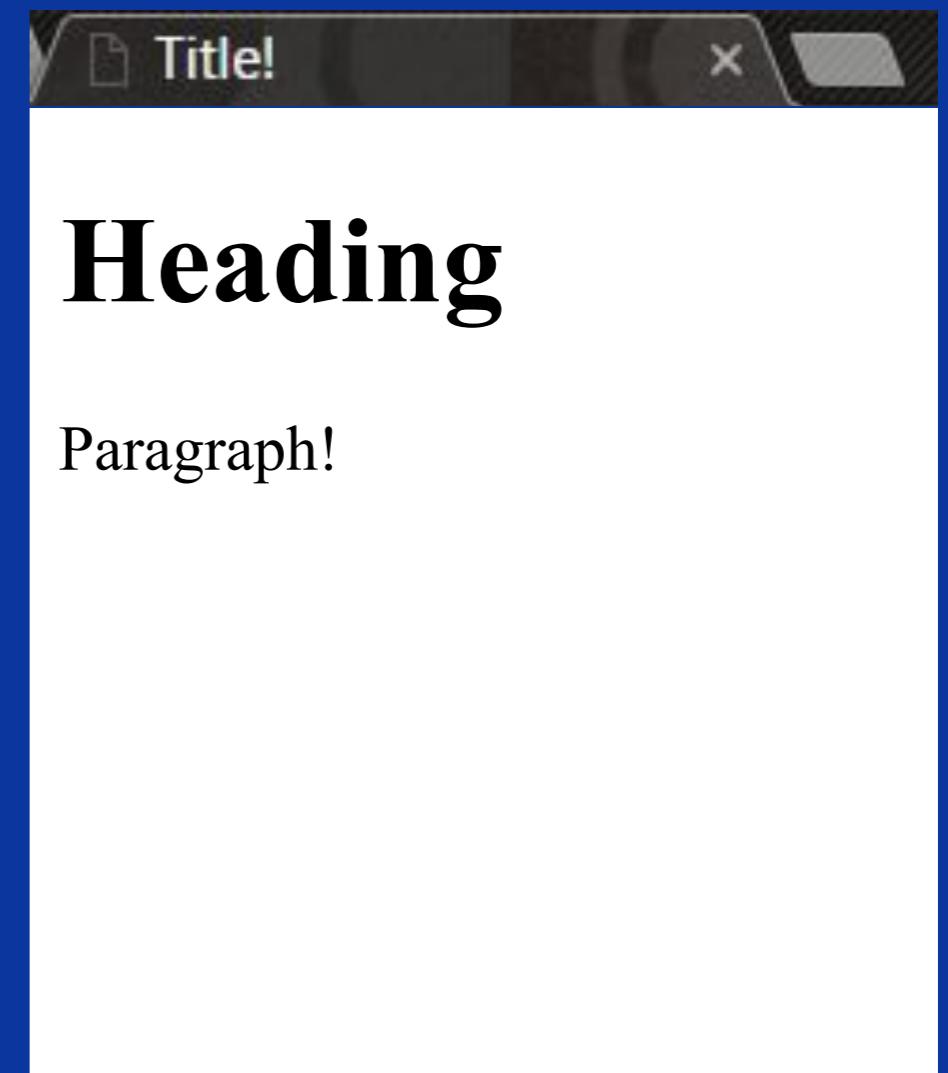
hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <div>Paragraph!</div>
  </body>
</html>
```



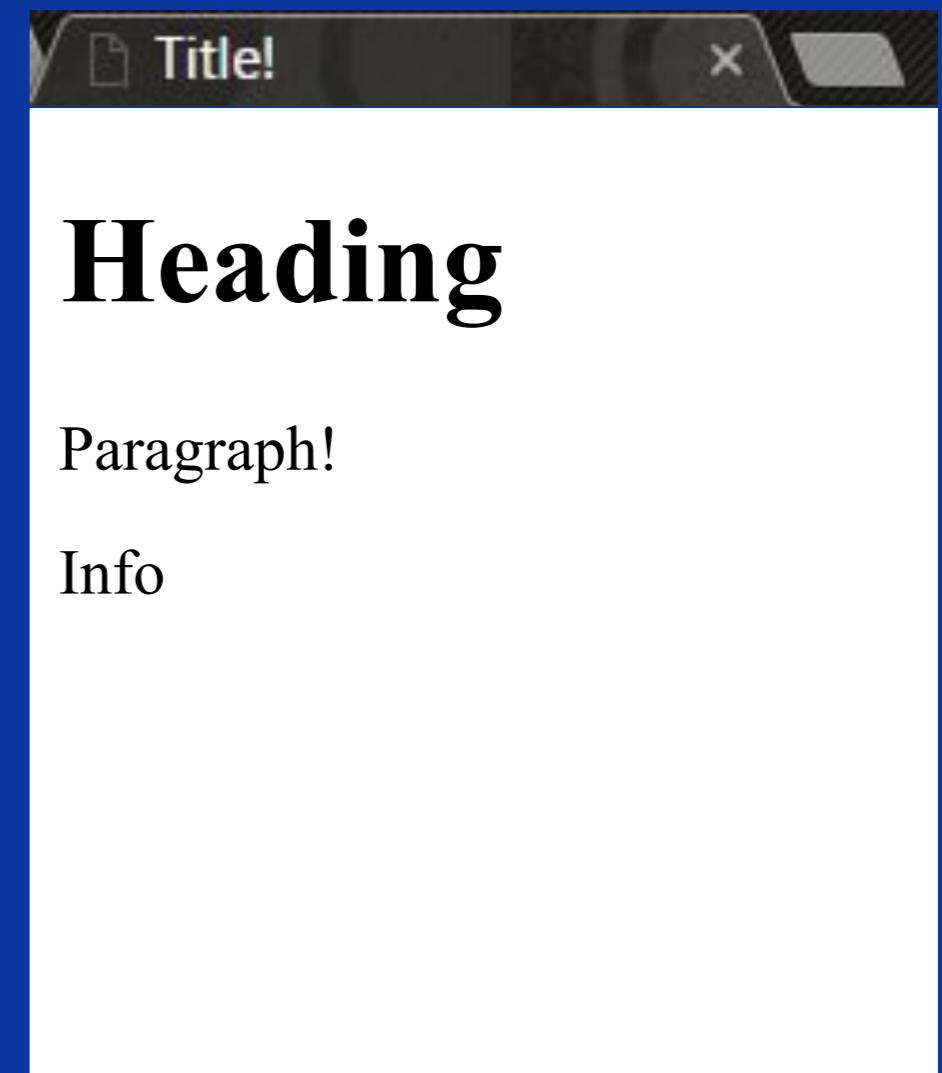
hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
```



hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
<div>Info</div>
```

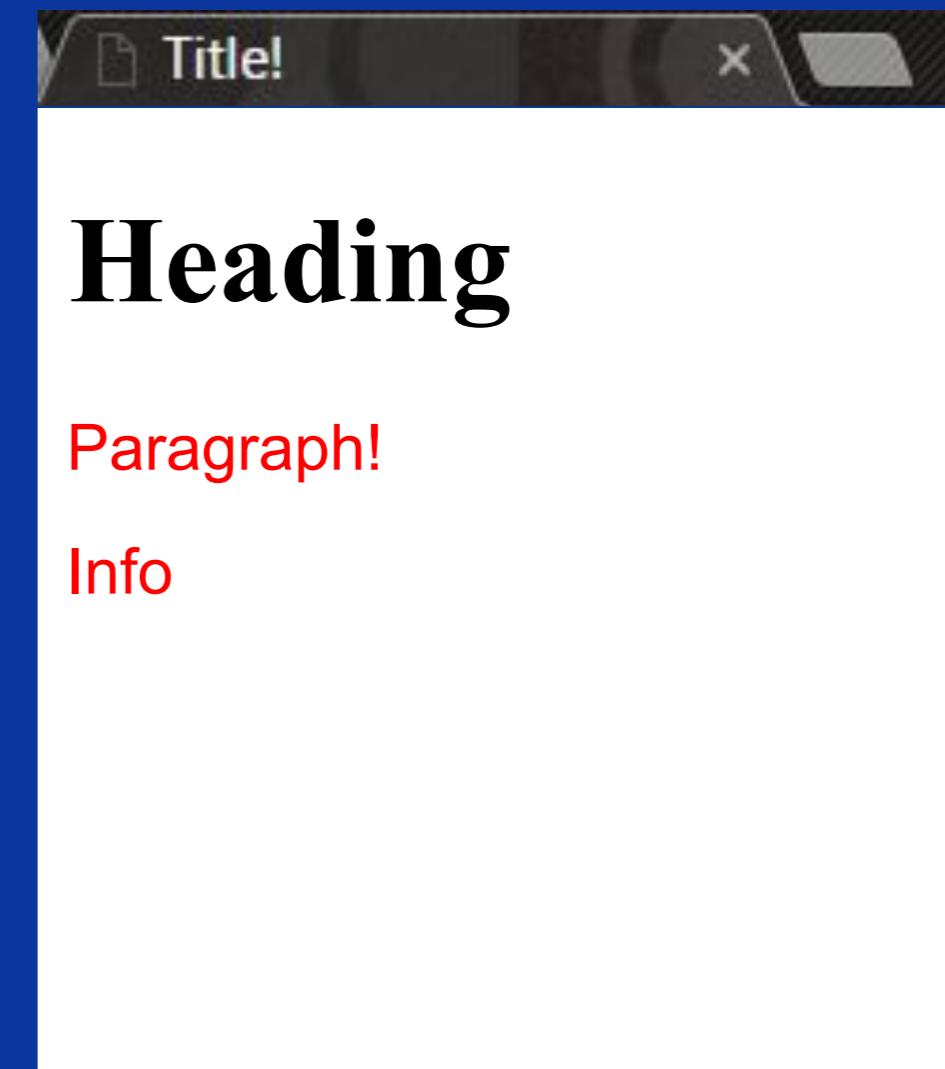


hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
<div>Info</div>
```

style.css

```
div {
    color: red;
    font-family: Arial;
    font-size: 24pt;
}
```

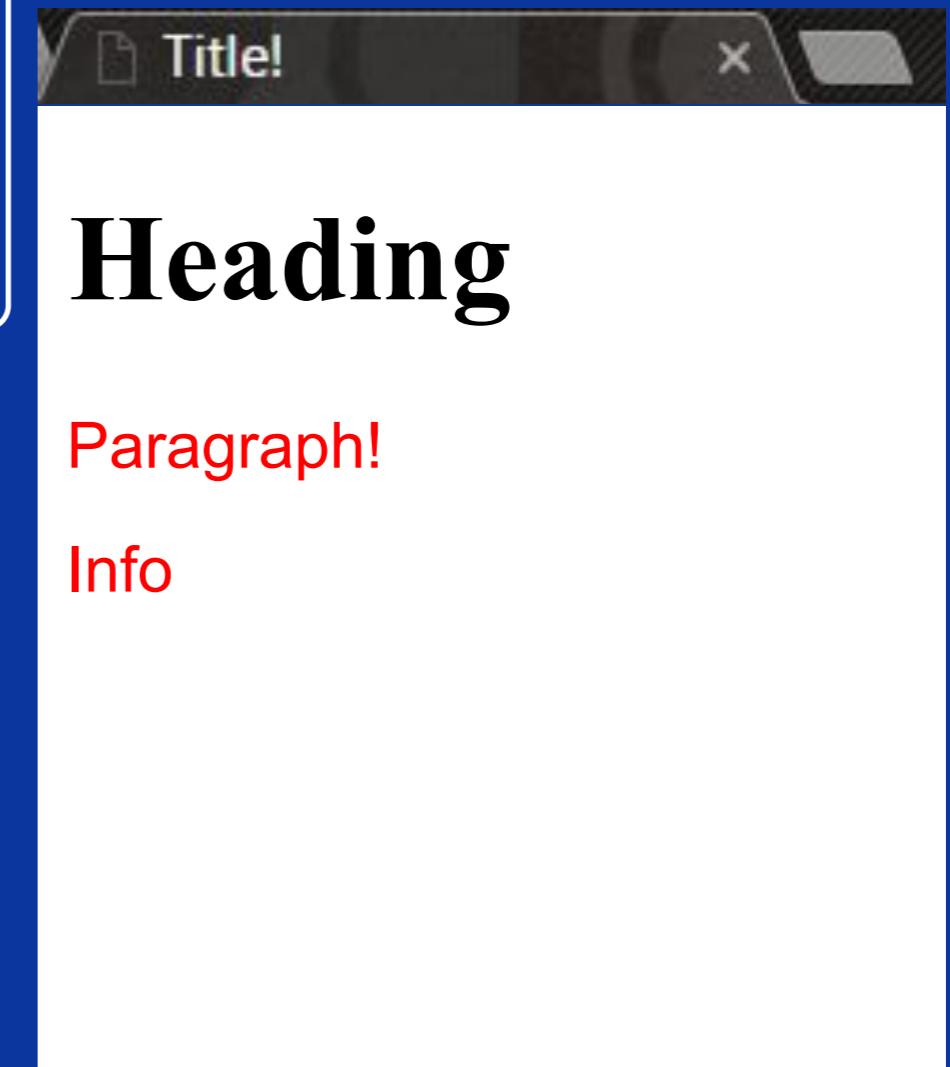


hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
<div class="info">Info</div>
```

style.css

```
div {
  color: red;
  font-family: Arial;
  font-size: 24pt;
}
```

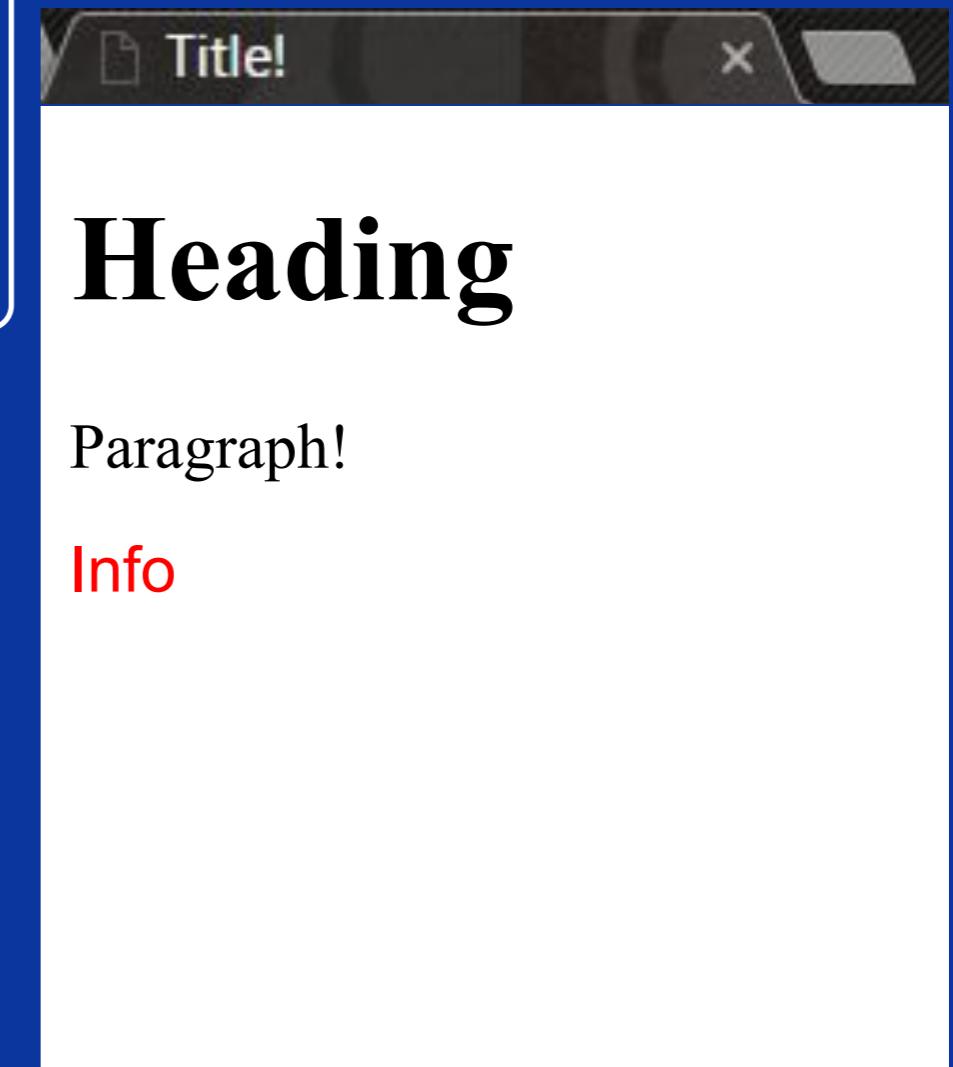


hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
<div class="info">Info</div>
```

style.css

```
.info {
    color: red;
    font-family: Arial;
    font-size: 24pt;
}
```

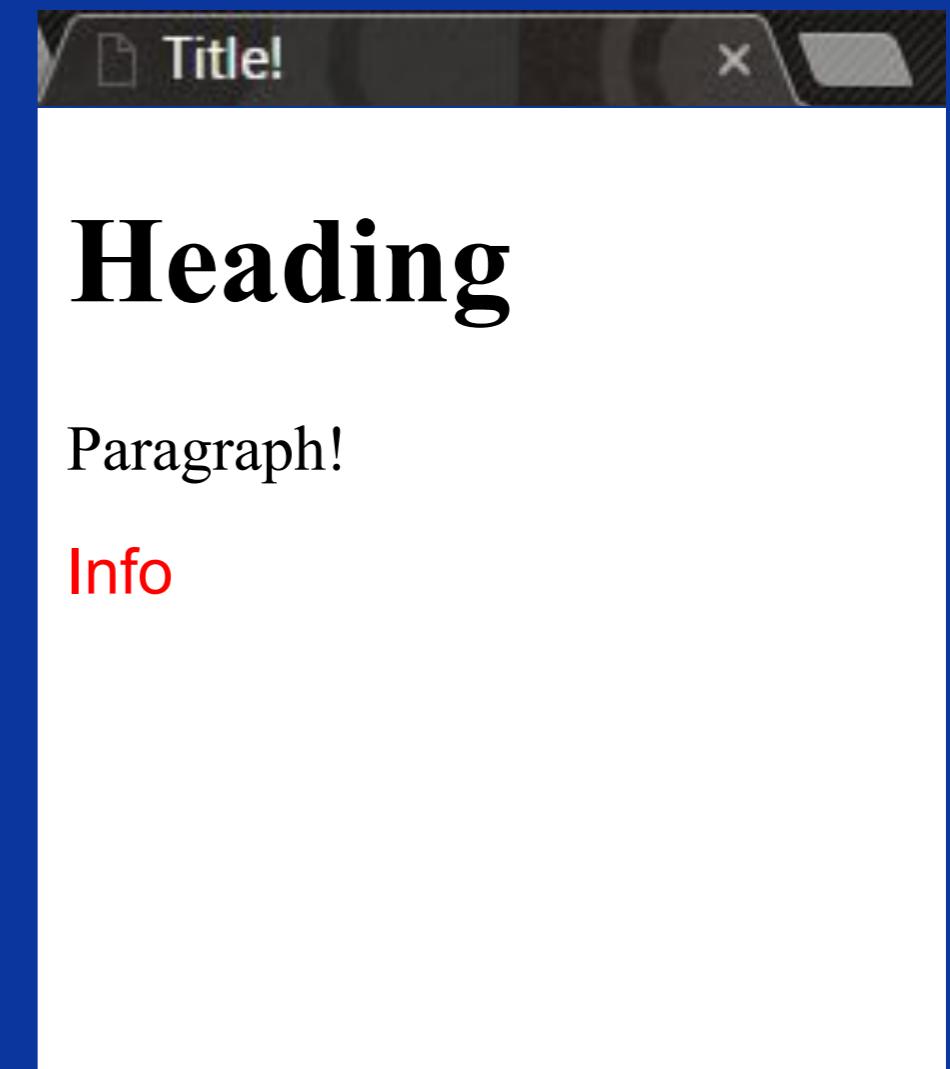


hello.html

```
<h1>Heading!</h1>
<div>Paragraph!</div>
<div id="unique">Info</div>
```

style.css

```
#unique {
    color: red;
    font-family: Arial;
    font-size: 24pt;
}
```



# ID vs Class

**ID**

**Class**

# ID vs Class

## ID

An element can have only one ID

## Class

An element can have multiple classes

# ID vs Class

## ID

An element can have only one ID

## Class

An element can have multiple classes

```
<div id="thisisanid">
```

```
<div class="class1  
class2 class3">
```

# ID vs Class

## ID

An element can have only one ID

## Class

An element can have multiple classes

IDs must be unique in any given HTML document

Can use the same class on multiple elements

# ID vs Class

## ID

An element can have only one ID

IDs must be unique in any given HTML document

## Class

An element can have multiple classes

Can use the same class on multiple elements

```
#id {  
...  
}
```

```
.classname {  
...  
}
```

Only use classes for  
CSS styling!

# Important Concept!

Tags such as <h1> have some default styling while <div> and <span> do not.

```
h1 {  
    display: block;  
    font-size: 2em;  
    margin-block-start: 0.67em;  
    margin-block-end: 0.67em;  
    margin-inline-start: 0px;  
    margin-inline-end: 0px;  
    font-weight: bold;  
}
```

```
div {  
    display: block;  
}
```

# Important Concept!

Tags such as `<h1>` have some default styling while `<div>` and `<span>` do not.

Any custom applied CSS rules will overwrite it!

# Important Concept!

Tags such as `<h1>` and `<p>` are basically the same as `<div>` except they have some default styling.

**NO THIS IS BAD**

So, you can build a webpage using `<p>` and `<h1>`, and then override the styling for those elements by creating CSS rules for `p` and `h1`...

... or you could **build your webpage (almost) entirely out of divs**, **create CSS rules for various classes**, and **apply classes to divs** on the webpage!

(often a combination of both)

# Combining HTML and CSS

hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```

hello.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title!</title>
    <link rel="stylesheet"
          href="style.css" />
  </head>
  <body>
    <h1>Heading!</h1>
    <p>Paragraph!</p>
  </body>
</html>
```

# In Conclusion:

HTML = nested boxes!  
CSS = a list of descriptions

A description in the CSS  
affects a box in the HTML

# 5 min break

\*Setup VSCode (or your desired editor) and clone Catbook if you haven't!!!

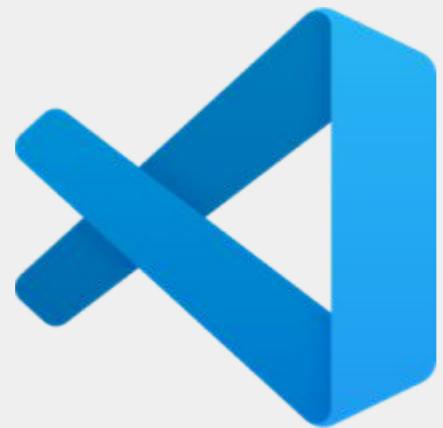
# HTML/CSS Workshop

Making your Catbook profile!

# Catbook

The purrfect social media website

# Your Tools

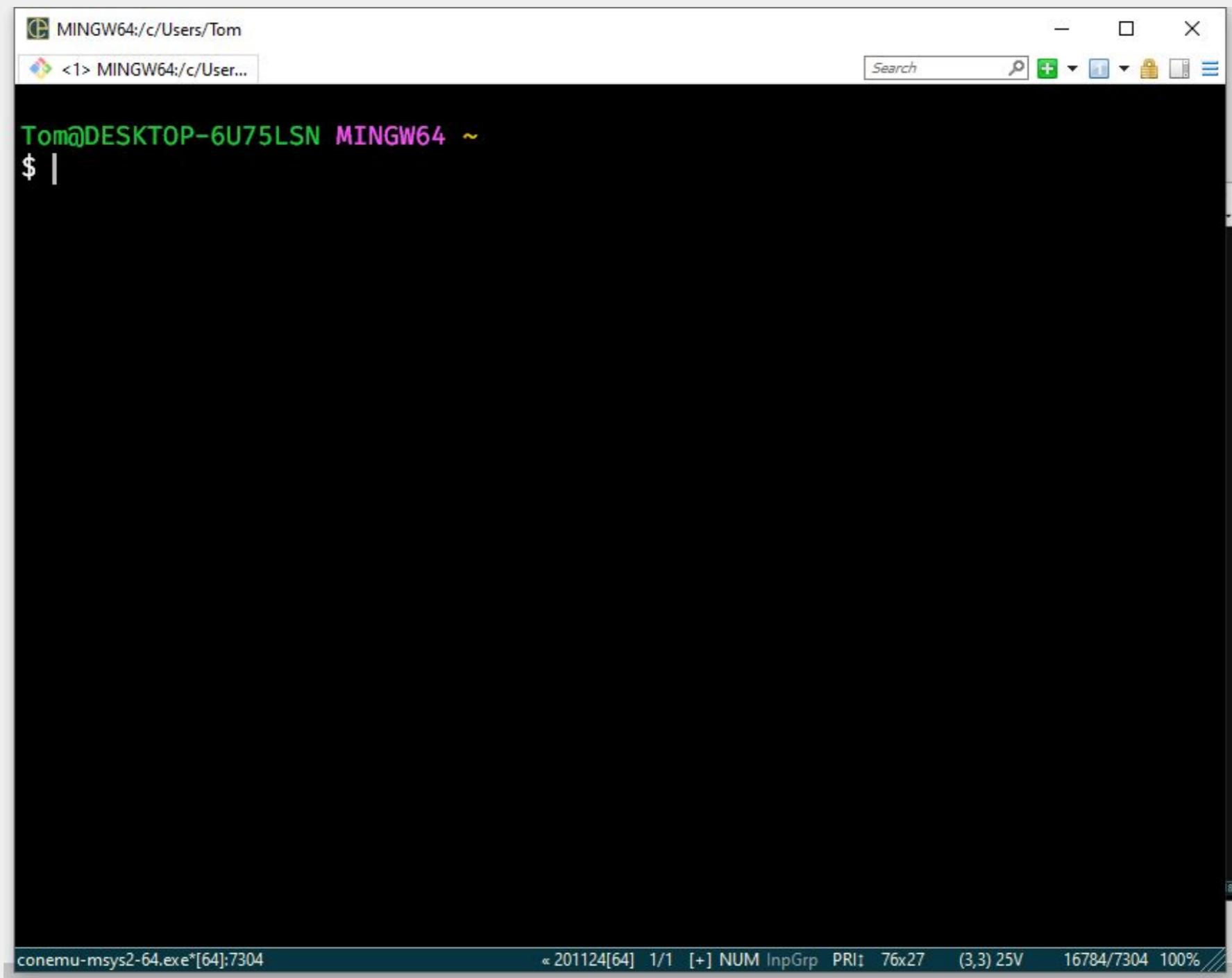


Code Editor

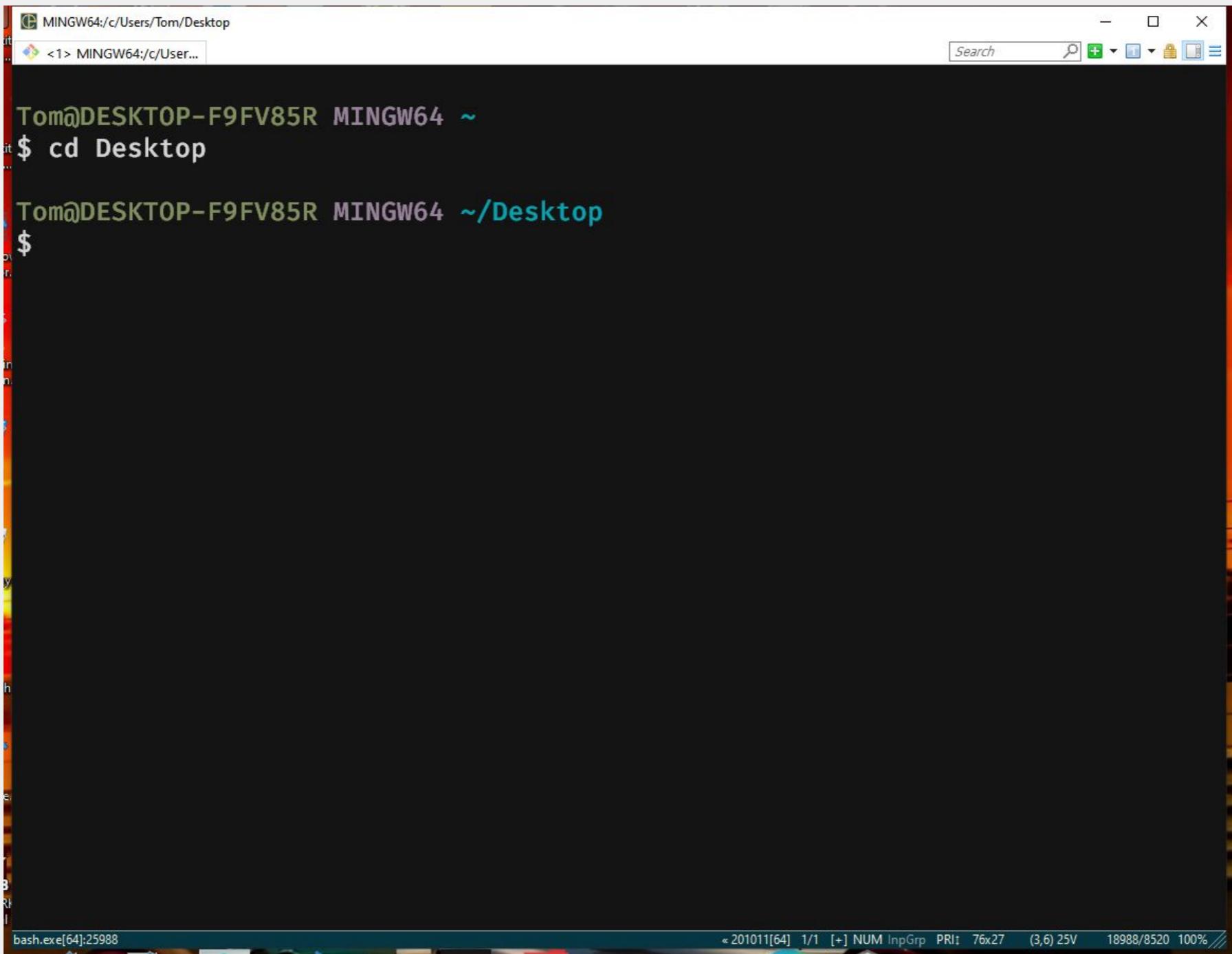


Terminal/Git Bash

# Open Console



# Open Console



A screenshot of a terminal window titled "MINGW64:/c/Users/Tom/Desktop". The window contains the following text:

```
Tom@DESKTOP-F9FV85R MINGW64 ~
$ cd Desktop

Tom@DESKTOP-F9FV85R MINGW64 ~/Desktop
$
```

The terminal has a dark background with light-colored text. The title bar shows the path "MINGW64:/c/Users/Tom/Desktop". The window includes standard operating system controls (minimize, maximize, close) and a toolbar with icons for search, new tab, and file operations. At the bottom, there is a status bar with system information.

# Git the Workshop!

[weblab.to/catbook](https://weblab.to/catbook)

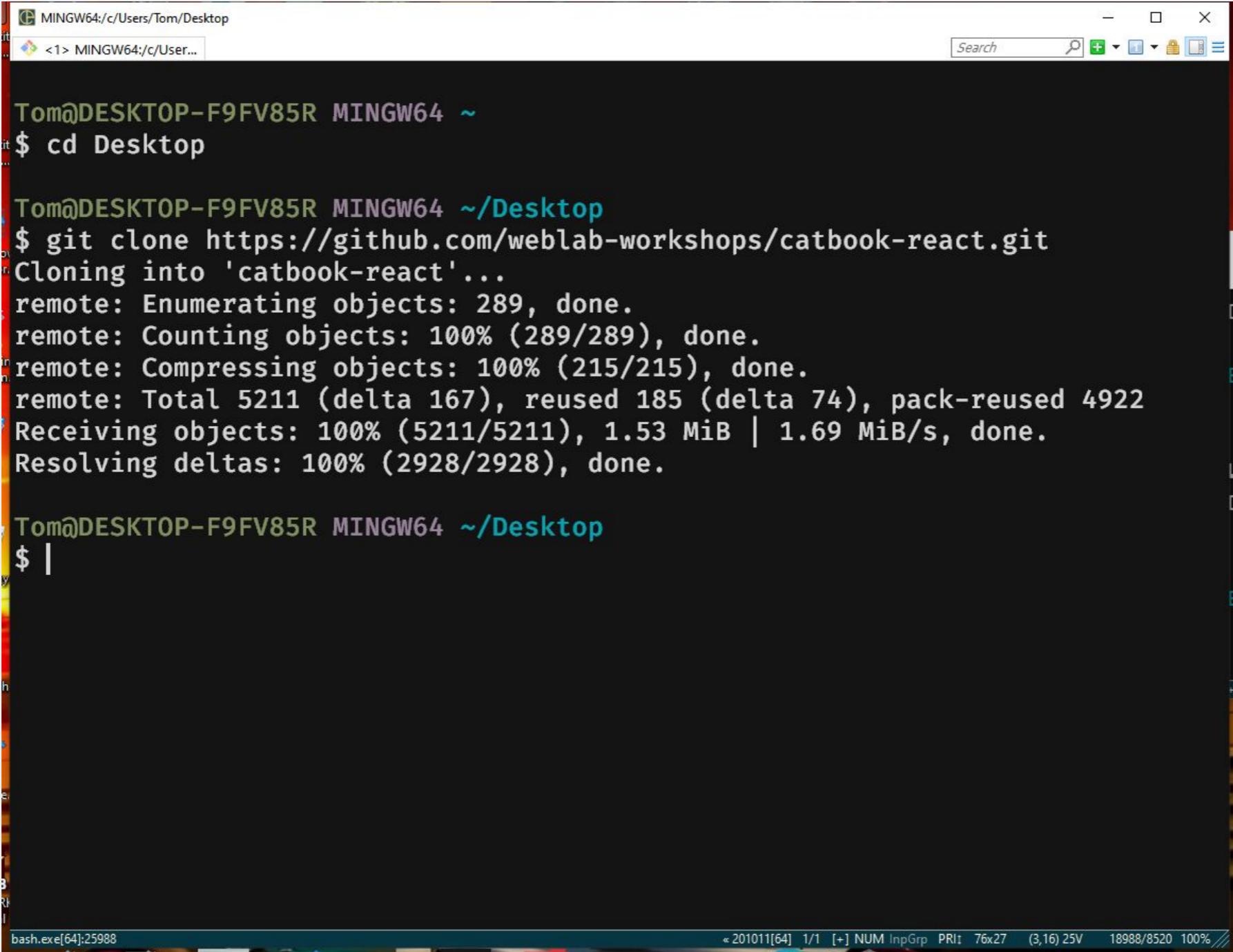
Cloning the repository! (we did this earlier!)

The screenshot shows a GitHub repository page for 'weblab-workshops / catbook-react'. The 'Code' tab is selected. A yellow warning box at the top right says: '⚠ We found potential security vulnerabilities in your dependencies. You can see this message because you have been granted access to Dependabot alerts for this repository.' It includes a 'See Dependabot alerts' button. Below the code tabs, there are buttons for 'master', '170 branches', '0 tags', 'Go to file', 'Add file', and a green 'Code' dropdown menu. A tooltip for the 'Code' dropdown shows options: 'Clone' (selected), 'HTTPS' (disabled), 'SSH' (disabled), 'GitHub CLI', a URL 'https://github.com/weblab-workshops/catbook-react', 'Open with GitHub Desktop', and 'Download ZIP'. Two blue arrows point to these options: arrow 1 points to the 'Code' dropdown, and arrow 2 points to the 'Clone' option in the tooltip.

1

2

# Git the Workshop!



The screenshot shows a terminal window titled "MINGW64:/c/Users/Tom/Desktop". The user has navigated to their desktop directory and cloned a repository from GitHub. The command used was \$ git clone https://github.com/weblab-workshops/catbook-react.git. The output shows the progress of the cloning process, including object enumeration, counting, compressing, and receiving objects, as well as resolving deltas. The terminal window also shows the user's name (Tom) and the host (DESKTOP-F9FV85R) at the top.

```
Tom@DESKTOP-F9FV85R MINGW64 ~
$ cd Desktop

Tom@DESKTOP-F9FV85R MINGW64 ~/Desktop
$ git clone https://github.com/weblab-workshops/catbook-react.git
Cloning into 'catbook-react'...
remote: Enumerating objects: 289, done.
remote: Counting objects: 100% (289/289), done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 5211 (delta 167), reused 185 (delta 74), pack-reused 4922
Receiving objects: 100% (5211/5211), 1.53 MiB | 1.69 MiB/s, done.
Resolving deltas: 100% (2928/2928), done.

Tom@DESKTOP-F9FV85R MINGW64 ~/Desktop
$ |
```

bash.exe[64]:25988 \* 201011[64] 1/1 [+] NUM InpGrp PRlt 76x27 (3,16) 25V 18988/8520 100%

# Git the Workshop!

The screenshot shows a terminal window titled 'MINGW64:/c/Users/Tom/Desktop/catbook-react'. The terminal output is as follows:

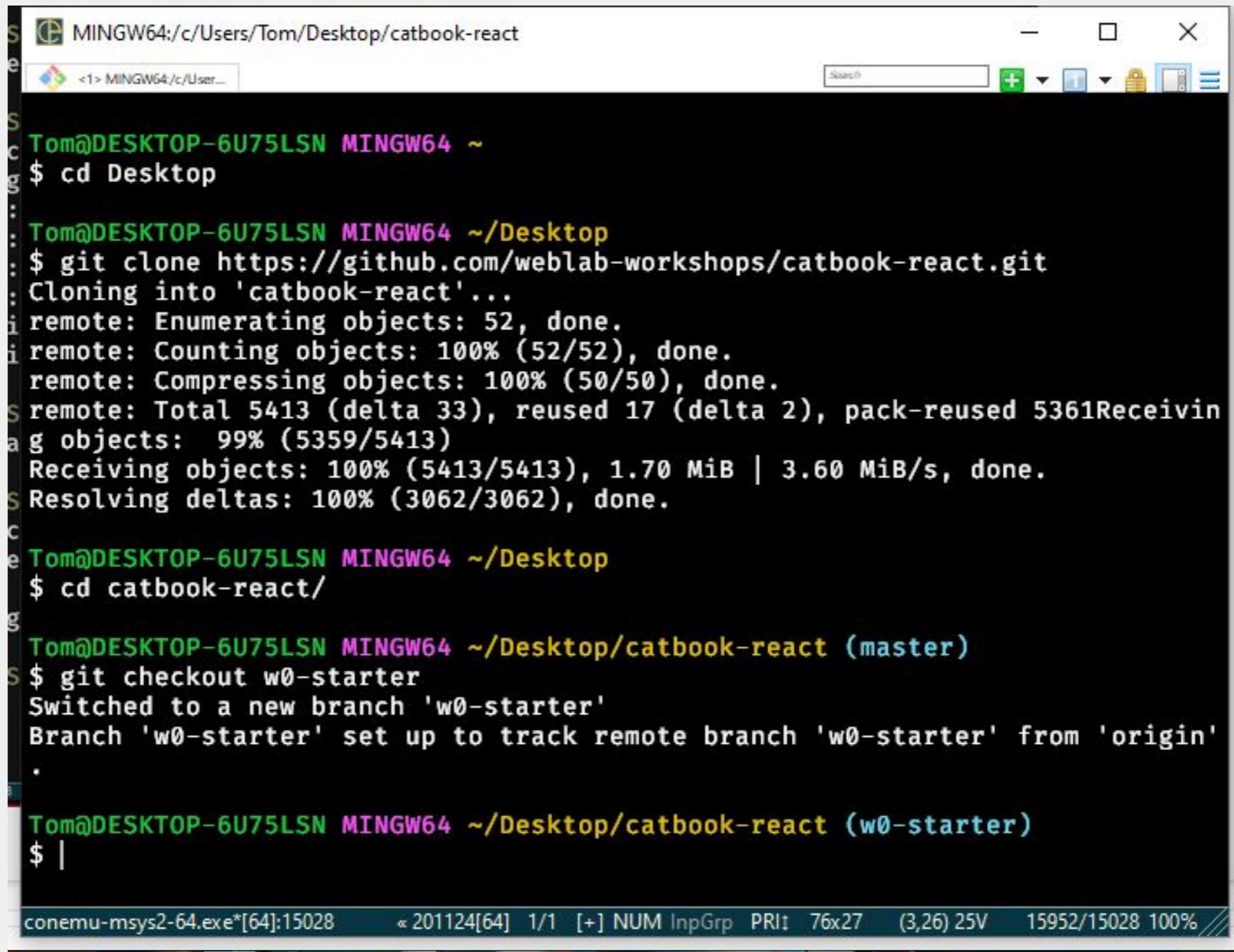
```
Tom@DESKTOP-F9FV85R MINGW64 ~
$ cd Desktop

Tom@DESKTOP-F9FV85R MINGW64 ~/Desktop
$ git clone https://github.com/weblab-workshops/catbook-react.git
Cloning into 'catbook-react'...
remote: Enumerating objects: 289, done.
remote: Counting objects: 100% (289/289), done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 5211 (delta 167), reused 185 (delta 74), pack-reused 4922
Receiving objects: 100% (5211/5211), 1.53 MiB | 1.69 MiB/s, done.
Resolving deltas: 100% (2928/2928), done.

Tom@DESKTOP-F9FV85R MINGW64 ~/Desktop
$ cd catbook-react/
$
```

At the bottom of the terminal window, there is a status bar with the following information: bash.exe[64]:25988, \* 201011[64] 1/1 [+], NUM, InpGrp, PRI+, 76x27, (3,19) 25V, 18988/8520, 100%, //.

# Git the Workshop!



The screenshot shows a terminal window titled "MINGW64:/c/Users/Tom/Desktop/catbook-react". The session starts with the user navigating to their desktop and cloning a GitHub repository named "catbook-react". The terminal then switches to the cloned directory and checks out a branch named "w0-starter". The status bar at the bottom indicates the process is running in "conemu-msys2-64.exe" with PID 15028.

```
MINGW64:/c/Users/Tom/Desktop/catbook-react
<1> MINGW64:/c/User...
Tom@DESKTOP-6U75LSN MINGW64 ~
$ cd Desktop
Tom@DESKTOP-6U75LSN MINGW64 ~/Desktop
$ git clone https://github.com/weblab-workshops/catbook-react.git
Cloning into 'catbook-react'...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 5413 (delta 33), reused 17 (delta 2), pack-reused 5361
Receiving objects: 100% (5413/5413), 1.70 MiB | 3.60 MiB/s, done.
Resolving deltas: 100% (3062/3062), done.
Tom@DESKTOP-6U75LSN MINGW64 ~/Desktop
$ cd catbook-react/
Tom@DESKTOP-6U75LSN MINGW64 ~/Desktop/catbook-react (master)
$ git checkout w0-starter
Switched to a new branch 'w0-starter'
Branch 'w0-starter' set up to track remote branch 'w0-starter' from 'origin'
.
Tom@DESKTOP-6U75LSN MINGW64 ~/Desktop/catbook-react (w0-starter)
$ |
```

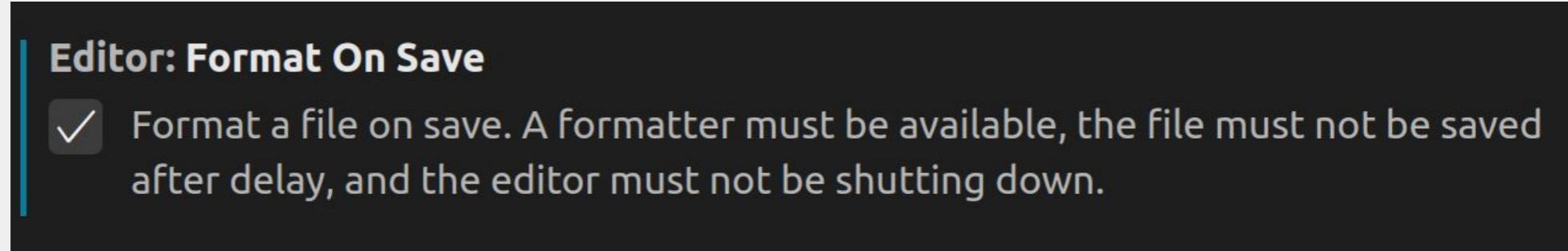
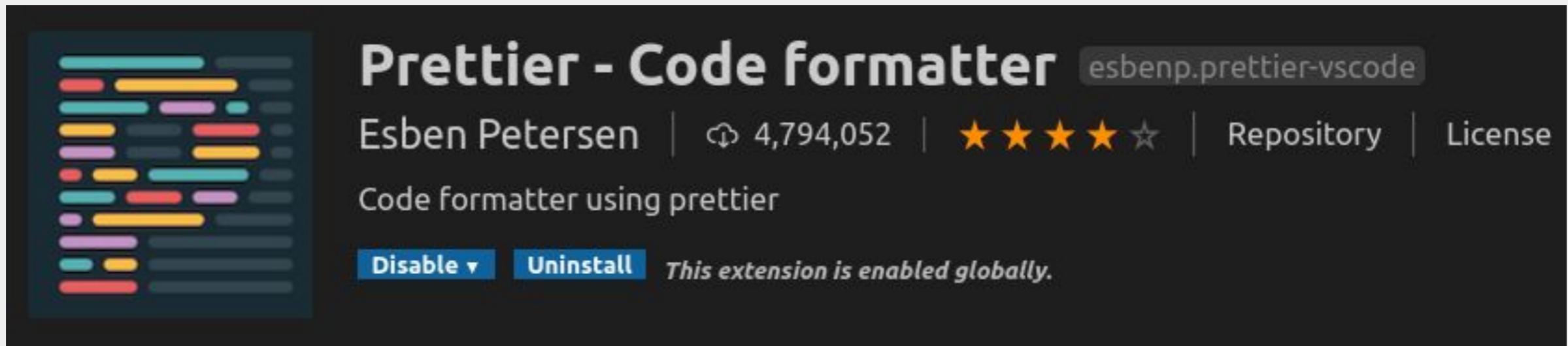
conemu-msys2-64.exe\*[64]:15028 201124[64] 1/1 [+] NUM InpGrp PRI: 76x27 (3,26) 25V 15952/15028 100%

# Pretty indentation/formatting is hard

```
return (<>
  <div className="u-flex u-relative Chatbook-container">
    <div className="Chatbook-userList">
      <ChatList setActiveUser={this.setActiveUser}
        userId={this.props.userId}
        users={this.state.activeUsers}
        active={this.state.activeChat.recipient} />
    </div>
    <div className="Chatbook-chatContainer u-relative">
      <Chat data={this.state.activeChat} />
    </div>
    </div>
  );
});
```

# Introducing prettier

- VSCode extension



- Makes your code pretty every time you hit “save”
  - Careful...make sure that everyone is using Prettier when checking this!

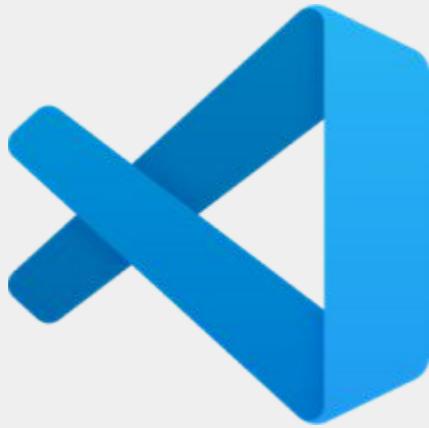
# Old code

```
return (<>
  <div className="u-flex u-relative Chatbook-container">
    <div className="Chatbook-userList">
      <ChatList setActiveUser={this.setActiveUser}
        userId={this.props.userId}
        users={this.state.activeUsers}
        active={this.state.activeChat.recipient} />
    </div>
    <div className="Chatbook-chatContainer u-relative">
      <Chat data={this.state.activeChat} />
    </div>
    </div>
  );
});
```

# After Ctrl+S

```
return (
  <>
    <div className="u-flex u-relative Chatbook-container">
      <div className="Chatbook-userList">
        <ChatList
          setActiveUser={this.setActiveUser}
          userId={this.props.userId}
          users={this.state.activeUsers}
          active={this.state.activeChat.recipient}
        />
      </div>
      <div className="Chatbook-chatContainer u-relative">
        <Chat data={this.state.activeChat} />
      </div>
    </div>
  ) ;
```

# Let's look at the code!

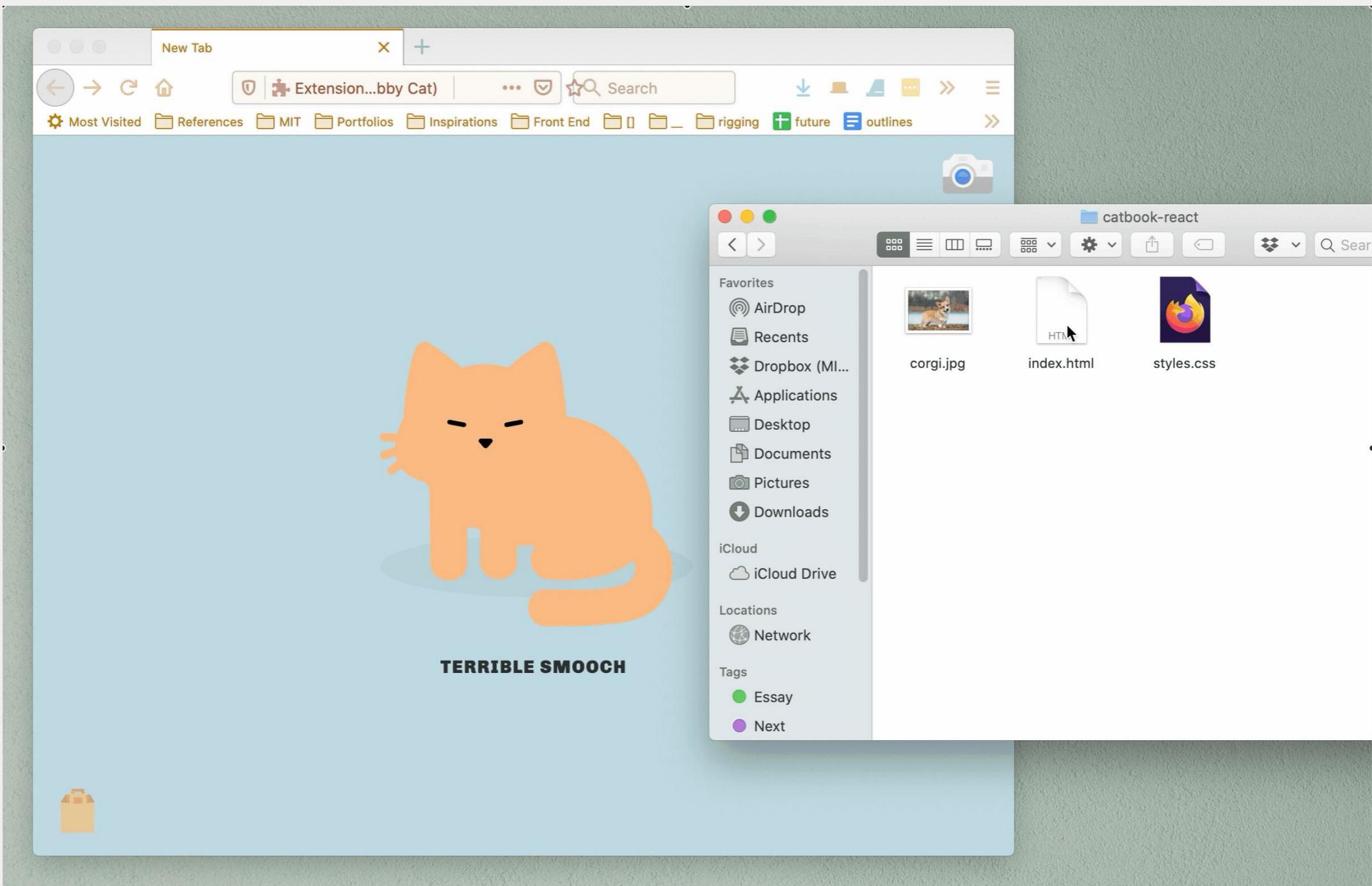


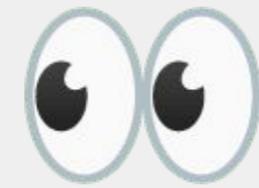
```
<> index.html <  
<> index.html > ...  
1  <!DOCTYPE html>  
2  <html>  
3  | <head>  
4  | | <title>Catbook</title>  
5  | </head>  
6  
7  <body>  
8  | Hello World  
9  </body>  
10 </html>
```

But first...

A coloring scheme!

# Let's look at the site!





# our goal

Catbook



Kevin Sun Chen

---

## About Me

I am more of a dog person but I'm just trying to fit in and get  
catbook

## My Favorite Type of Cat

I actually like dogs

# Let's do the HTML!

## Step 1: Add some text!

Hint: Recall the `<html>`, `<h1>`, `<p>` and `<div>` tags.  
Perhaps some others such as `<section>` might be useful... 🤔

## Step 2: Add your profile pic!

Hint: Recall the `<img>` tag and its attribute?

### Kevin Sun Chen

#### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

#### My Favorite Type of Cat

I actually like dogs



Kevin Sun Chen

# Exercise 1: Kitten Started

Let's try adding some text about us in `index.html`!

Recall the `<html>`, `<h1>`, `<p>` and `<div>` tags.

Perhaps some others such as `<section>` might be useful... 

## Kevin Sun Chen

### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

### My Favorite Type of Cat

I actually like dogs

# Need Help?

**In person:** Raise your hand!

**Virtual:** [weblab.to/q](https://weblab.to/q)

# Exercise 1: Kitten Started

**git reset --hard** (*do this before any checkout command!*)

git checkout w0-step1

## Kevin Sun Chen

---

### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

### My Favorite Type of Cat

I actually like dogs

# Exercise 2: Pretty Paws

Let's add a profile picture `index.html`!

Hint! Recall the `<img>` tag and its attribute?



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Exercise 2: Pretty Paws

git checkout w0-step2



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Making it Stylish

It's time to link to our CSS stylesheet.

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

# Making it Stylish

Let's center the text together!

```
.u-textCenter {  
    text-align: center;  
}
```



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Making it Stylish

Let's center the text together!

```
.u-textCenter {  
    text-align: center;  
}
```



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Making it Stylish

We call these kinds of classes that just apply one css attribute, **utility classes**.

# Making it Stylish

We call these kinds of classes that just apply one css attribute, **utility classes**.

We'll format our utility classes so that they're prefaced by a “u-”

```
.u-textCenter {  
    text-align: center;  
}
```

# Making it Stylish

git checkout w0-step3



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Exercise 3: Centerpiece

Use **u-textCenter** to center the other text!



**Kevin Sun Chen**

---

#### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

#### My Favorite Type of Cat

I actually like dogs

# Exercise 3: Centerpiece

git checkout w0-step4



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Exercise 3: Centerpiece

```
<section class="u-textCenter">
  <h4>About Me</h4>
  <p>
    | I am more of a dog person but I'm just trying to fit in and get catbook
  </p>
</section>
<section class="u-textCenter">
  <h4>My Favorite Type of Cat</h4>
  <p>
    | I actually like dogs
  </p>
</section>
```

# Exercise 3: Centerpiece

```
<section class="u-textCenter">
  <h4>About Me</h4>
  <p>
    | I am more of a dog person but I'm just trying to fit in and get catbook
  </p>
</section>
<section class="u-textCenter">
  <h4>My Favorite Type of Cat</h4>
  <p>
    | I actually like dogs
  </p>
</section>
```

# Importing Fonts

We're slowly getting there! Let's change fonts now.

# Importing Fonts

[fonts.google.com](https://fonts.google.com)

The screenshot shows the Google Fonts interface. At the top, there's a search bar with "open sans" typed in, and a dropdown menu set to "Sentence". Below the search bar are filters for "Categories", "Language", "Font properties", and a checkbox for "Show only variable fonts". The main area displays two font families: "Open Sans" by Steve Matteson with 10 styles, and "Open Sans Condensed" by Steve Matteson with 3 styles. Both fonts are demonstrated with a sentence: "Almost before we knew it, we had left the ground." The "Open Sans" card has a larger arrow pointing to its preview text, while the "Open Sans Condensed" card has a smaller arrow pointing to its preview text.

Google Fonts

open sans Sentence Type something

Categories Language Font properties Show only variable fonts

2 of 1023 families

Open Sans Steve Matteson 10 styles

Open Sans Condensed Steve Matteson 3 styles

Almost before we knew it, we had left the ground.

# Importing Fonts

[fonts.google.com](https://fonts.google.com)

The screenshot shows the Google Fonts interface for the 'Open Sans' font family. The main content area displays preview text for various styles: Light 300, Light 300 italic, Regular 400, Regular 400 italic, and Semi-bold 600. Each preview includes a text input field for previewing and a font size selector (40px). To the right, a sidebar titled 'Selected family' lists the chosen styles: Light 300 and Semi-bold 600. Below this is the 'Review' section, which contains the font's name and a code editor showing the @import URL. The 'Use on the web' section provides instructions for embedding the font and shows the CSS rule `font-family: 'Open Sans', sans-serif;`. The bottom right corner features a large number '4'.

1a

1b

2

3

4

Google Fonts

Browse fonts   Featured   Articles   About

Selected family

Open Sans

Designed by Steve Matteson

Download family

Select styles   Glyphs   About   License   Pairings

Styles

Type here to preview text

Almost before we knew it, we had left the ground.

Light 300

Almost before we knew it, we had left the ground.

Light 300 italic

Almost before we knew it, we had left the ground.

Regular 400

Almost before we knew it, we had left the ground.

Regular 400 italic

Almost before we knew it, we had left the ground.

Semi-bold 600

Almost before we knew it, we had left the ground.

Review

Open Sans

Light 300

Semi-bold 600

Add more styles   Remove all

Use on the web

To embed a font, copy the code into the <head> of your html

<link>  @import

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;600&display=swap');
</style>
```

CSS rules to specify families

```
font-family: 'Open Sans', sans-serif;
```

API docs   Download all

# Importing Fonts

git checkout w0-step5

```
@import url('https://fonts.googleapis.com/css2?  
family=Open+Sans:wght@300;600&display=swap');
```

```
html,  
body {  
  font-family: "Open Sans", sans-serif;  
}
```



**Kevin Sun Chen**

---

#### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

#### My Favorite Type of Cat

I actually like dogs

# More CSS!

Step 4: Center more text  
with our new ruleset!

# Adding a Navbar

Let's add the tag `<nav>` at the top for our navbar content.

```
<nav class="navContainer">
| <h1 class="navTitle">Catbook</h1>
</nav>
```

# Adding a Navbar

Next, we're gonna wanna style it colorful. But first, let's define some colors!

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout.

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout. We'll define our colors as variables.

```
:root {  
    --primary: #396dff;  
    --grey: #f7f7f7;  
    --white: #fff;  
}
```

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout. We'll define our colors as variables.

```
:root {  
    --primary: #396dff;  
    --grey: #f7f7f7;  
    --white: #fff;  
}
```

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout. We'll define our colors as variables.

```
:root {  
    --primary: #396dff;  
    --grey: #f7f7f7;  
    --white: #fff;  
}
```

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout. We'll define our colors as variables.

```
.navTitle {  
  color: var(--primary);  
  font-size: 20px;
```

# CSS Variables

CSS has **CSS variables** that can be defined and then used throughout. We'll define our colors as variables.

```
.navTitle {  
  color: var(--primary);  
  font-size: 20px;
```

```
/* Below cancels out some of the default styling of h1 */  
margin: 0;  
font-weight: normal;
```

add this under .navTitle too!

# Adding a Navbar

Try changing the color in `:root` and see the color change when you refresh the page!

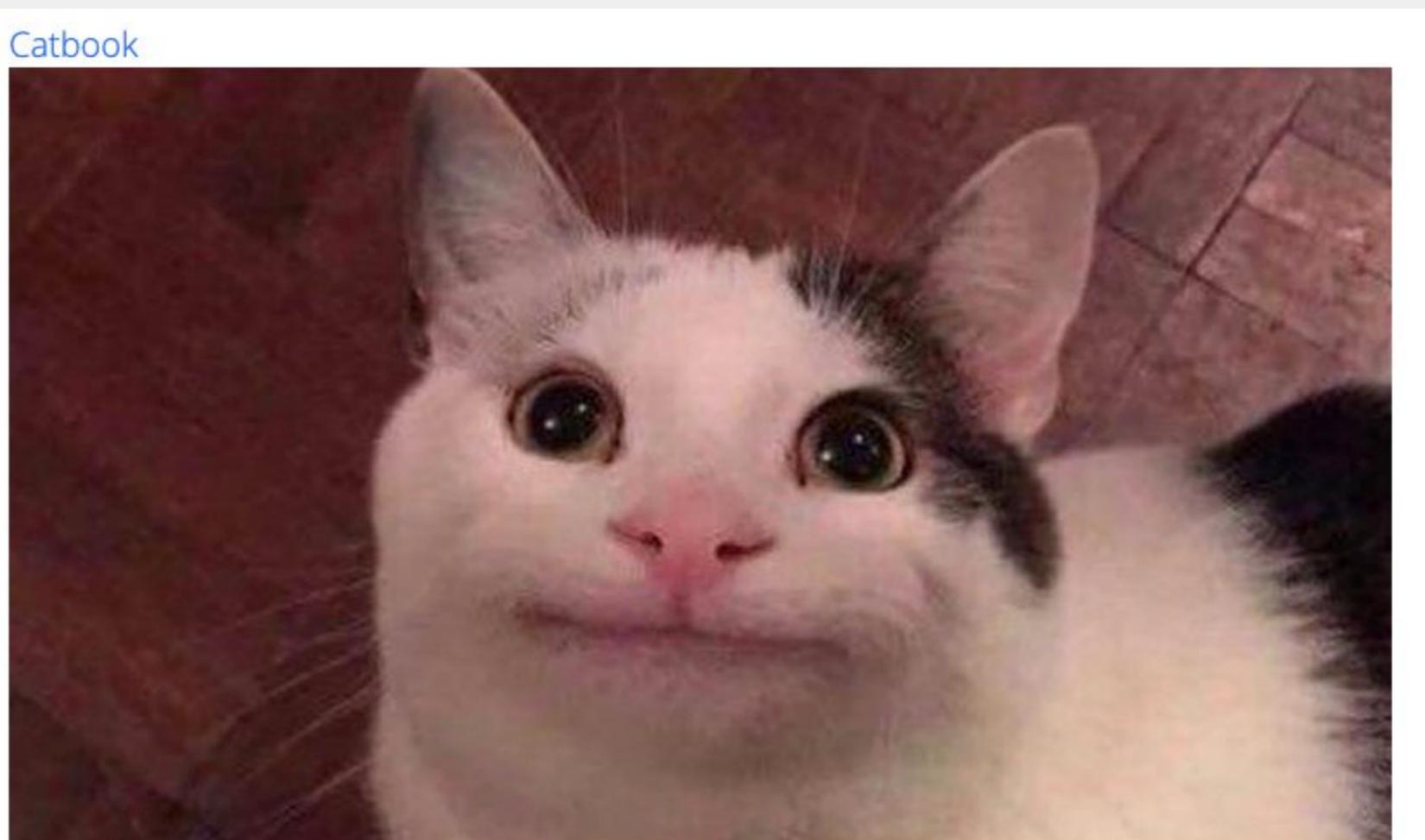
Catbook



**Kevin Sun Chen**

# Adding a Navbar

git checkout w0-step6

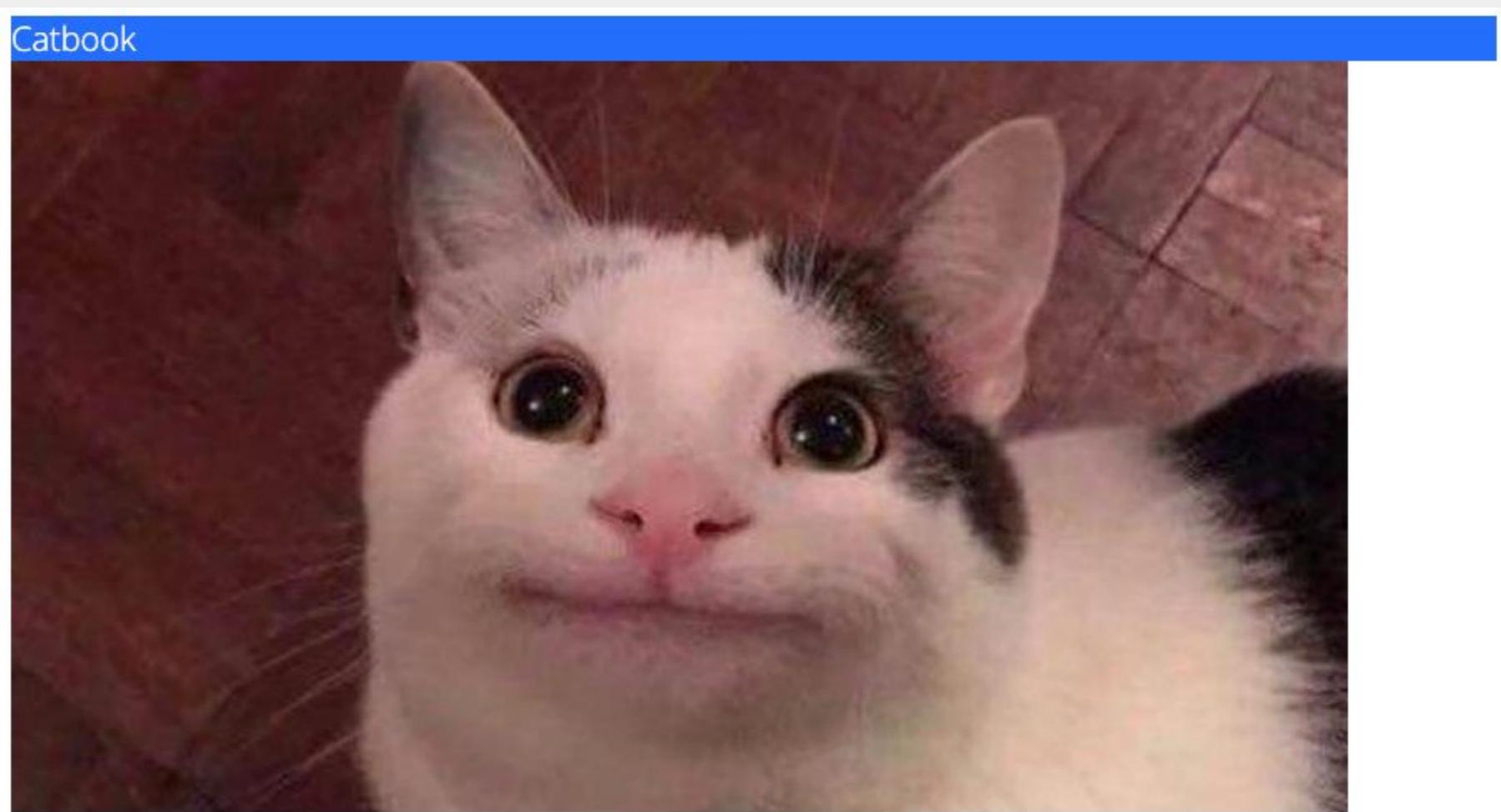


**Kevin Sun Chen**

# Exercise 4: Cat's Head

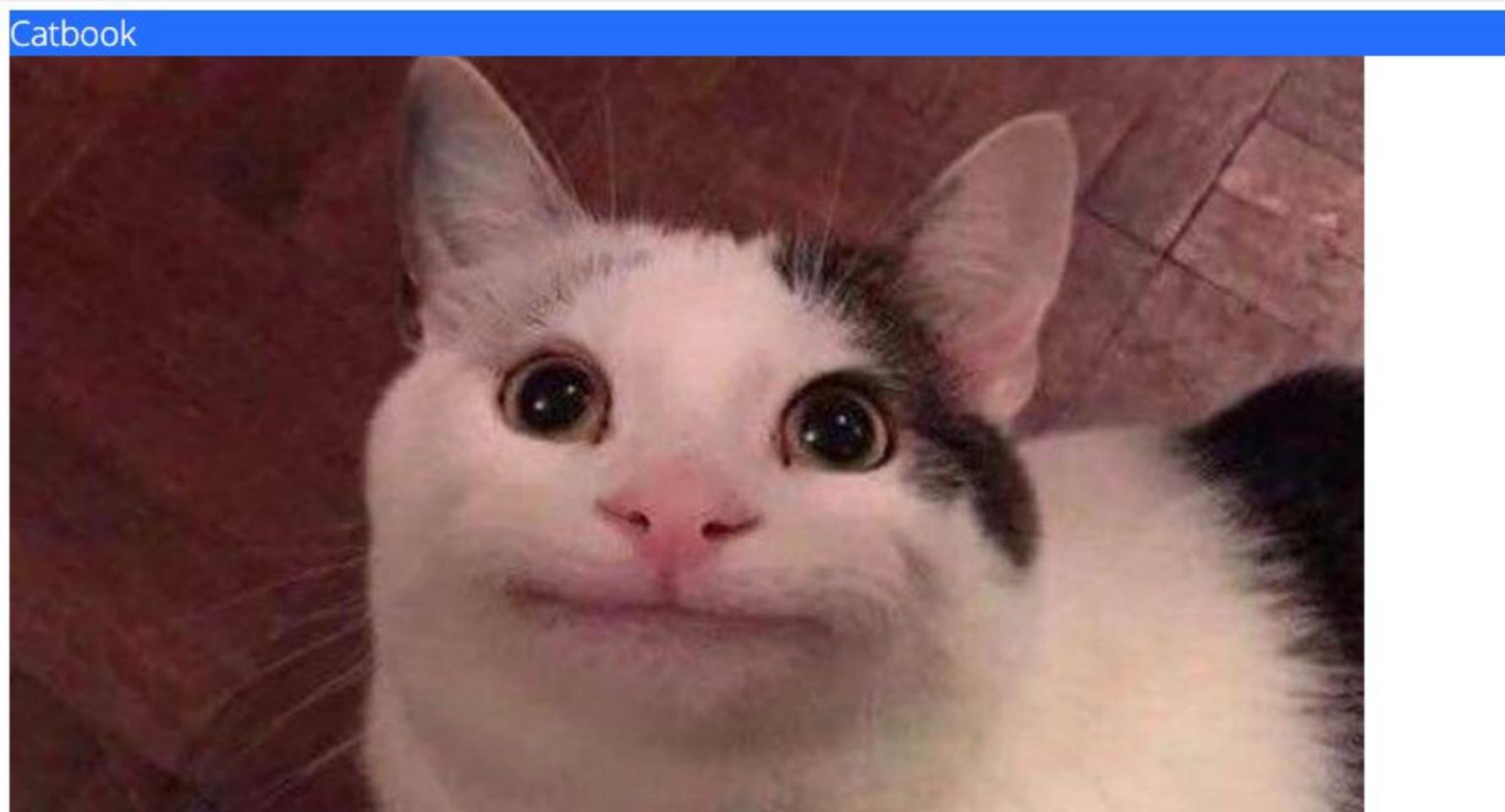
Now try using those variables to style the nav bar!

Hint: you may need a new container class!



# Exercise 4: Cat's Head

git checkout w0-step7



**Kevin Sun Chen**

# Blank Space

(Taylor's Version)



The Void.

# The Void.

Catbook



**Kevin Sun Chen**

---

## About Me

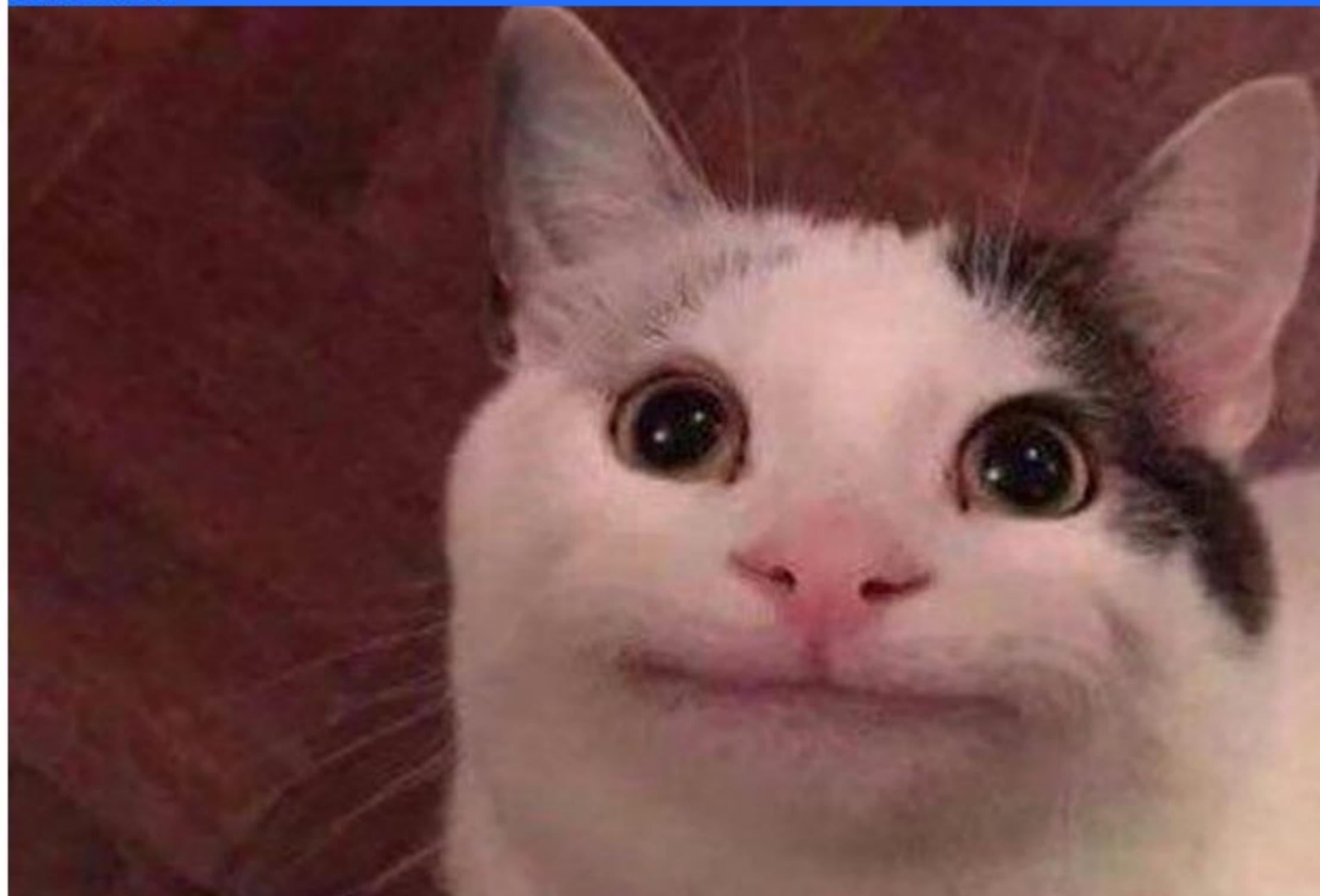
I am more of a dog person but I'm just trying to fit in and get catbook

## My Favorite Type of Cat

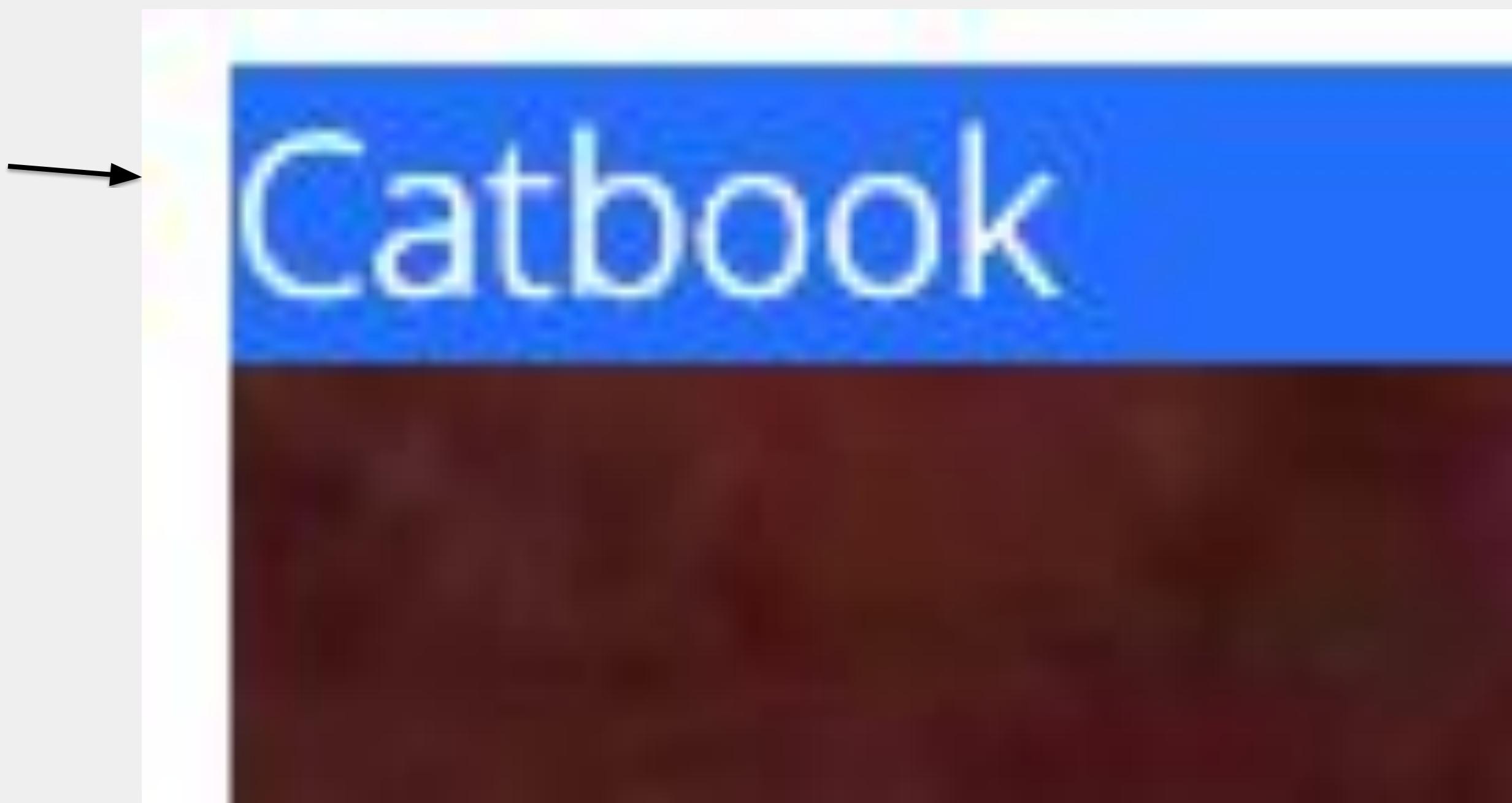
I actually like dogs

# The Void.

Catbook



# The Void.

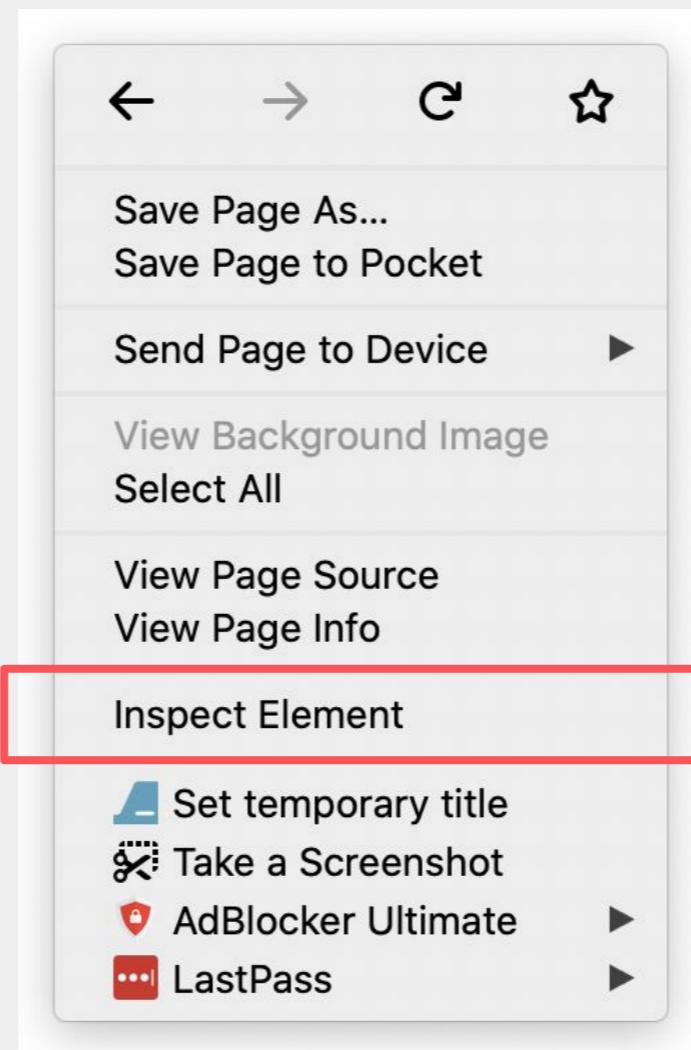


# We Must Fill It.

Let's inspect where this is coming from! 

# The Void.

Right click + click Inspect Element



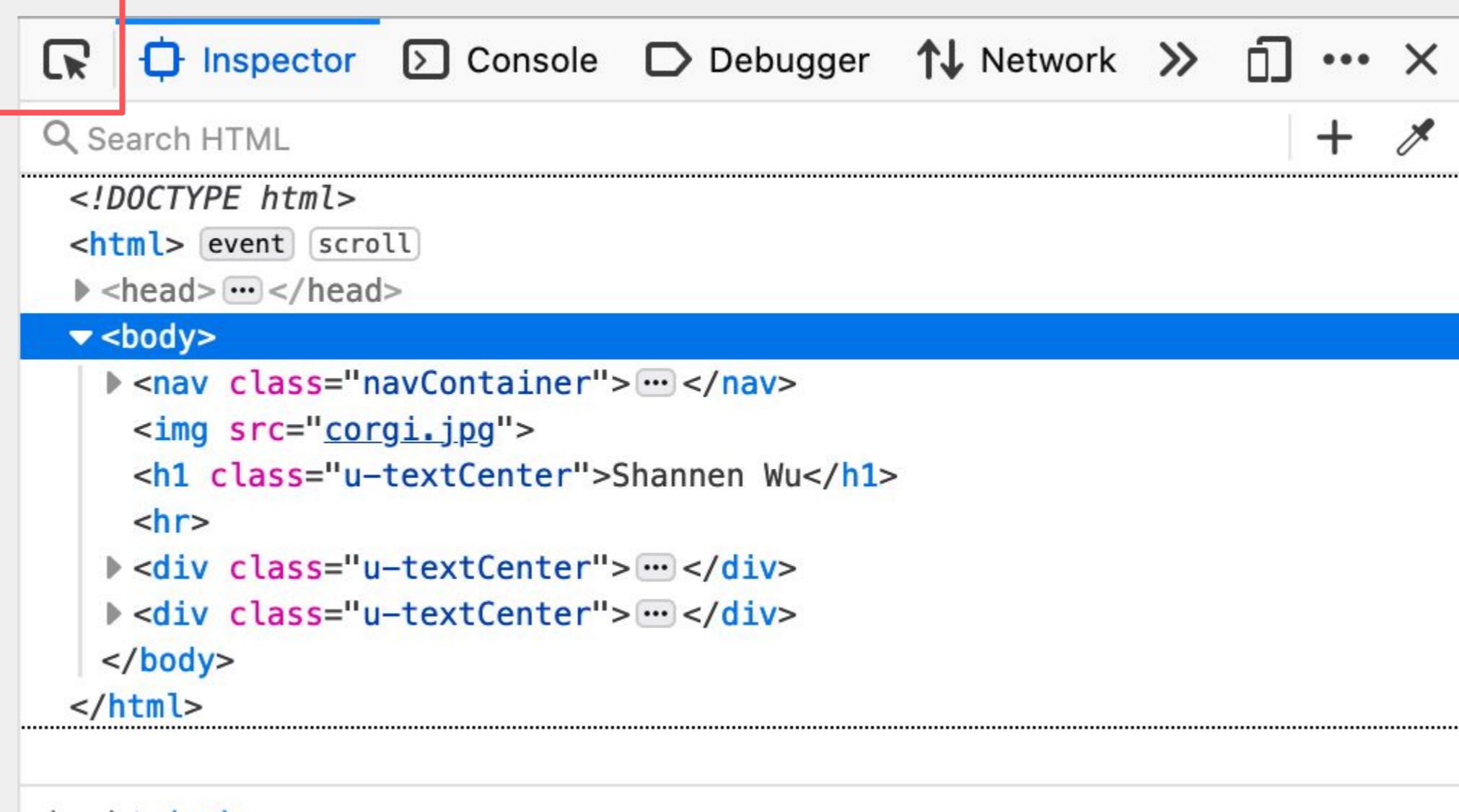
# The Void.

A screenshot of a browser's developer tools, specifically the Inspector tab, displaying the HTML structure of a web page. The page content includes a navigation bar, a corgi image, and a large heading. The browser interface shows various tabs like Inspector, Console, Debugger, and Network, along with search and edit functions.

```
<!DOCTYPE html>
<html> event scroll
  > <head> ...
  > <body>
    > <nav class="navContainer">...
    > 
    > <h1 class="u-textCenter">Shannen Wu</h1>
    > <hr>
    > <div class="u-textCenter">...
    > <div class="u-textCenter">...
  </body>
</html>
```

html > body

# The Void.

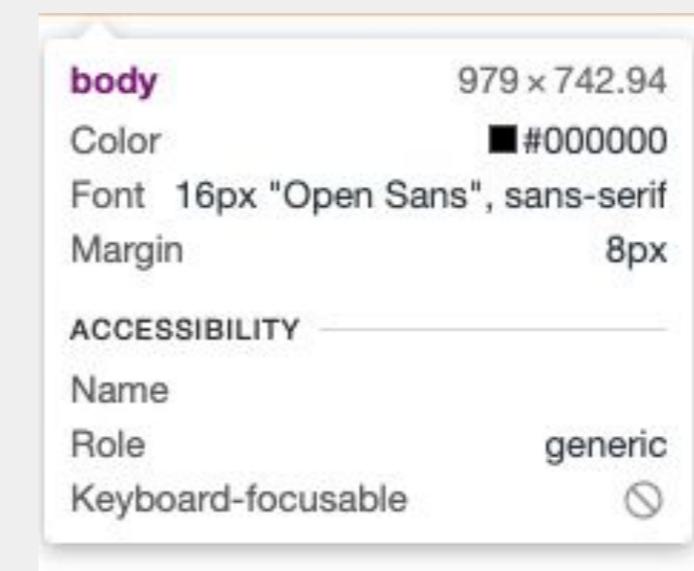
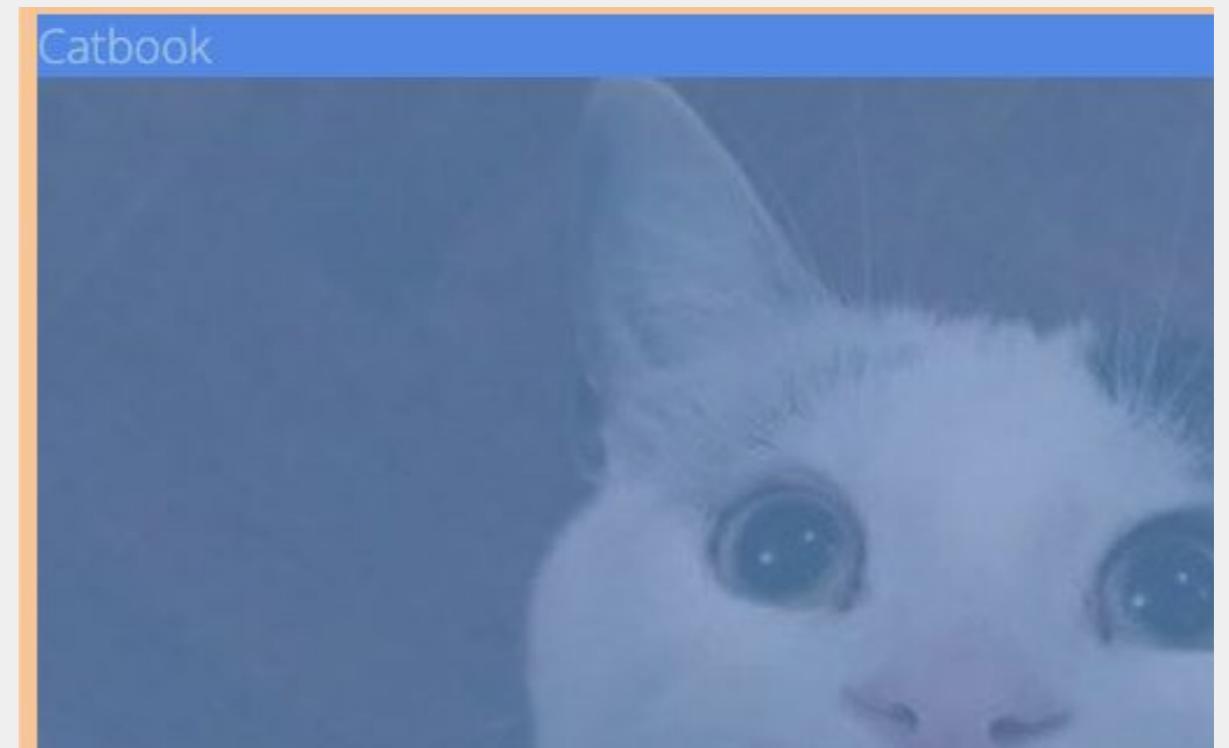
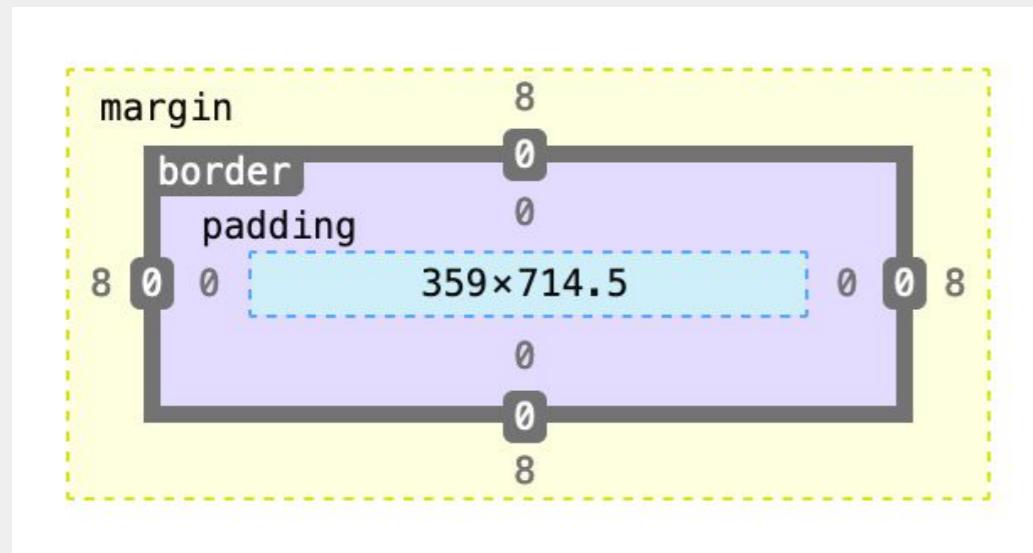


A screenshot of a browser's developer tools, specifically the Inspector tab, showing the DOM structure of a page. The DOM tree is displayed with nodes expanded or collapsed. A red box highlights the 'Inspector' icon in the top navigation bar. The code editor at the bottom shows the HTML file structure.

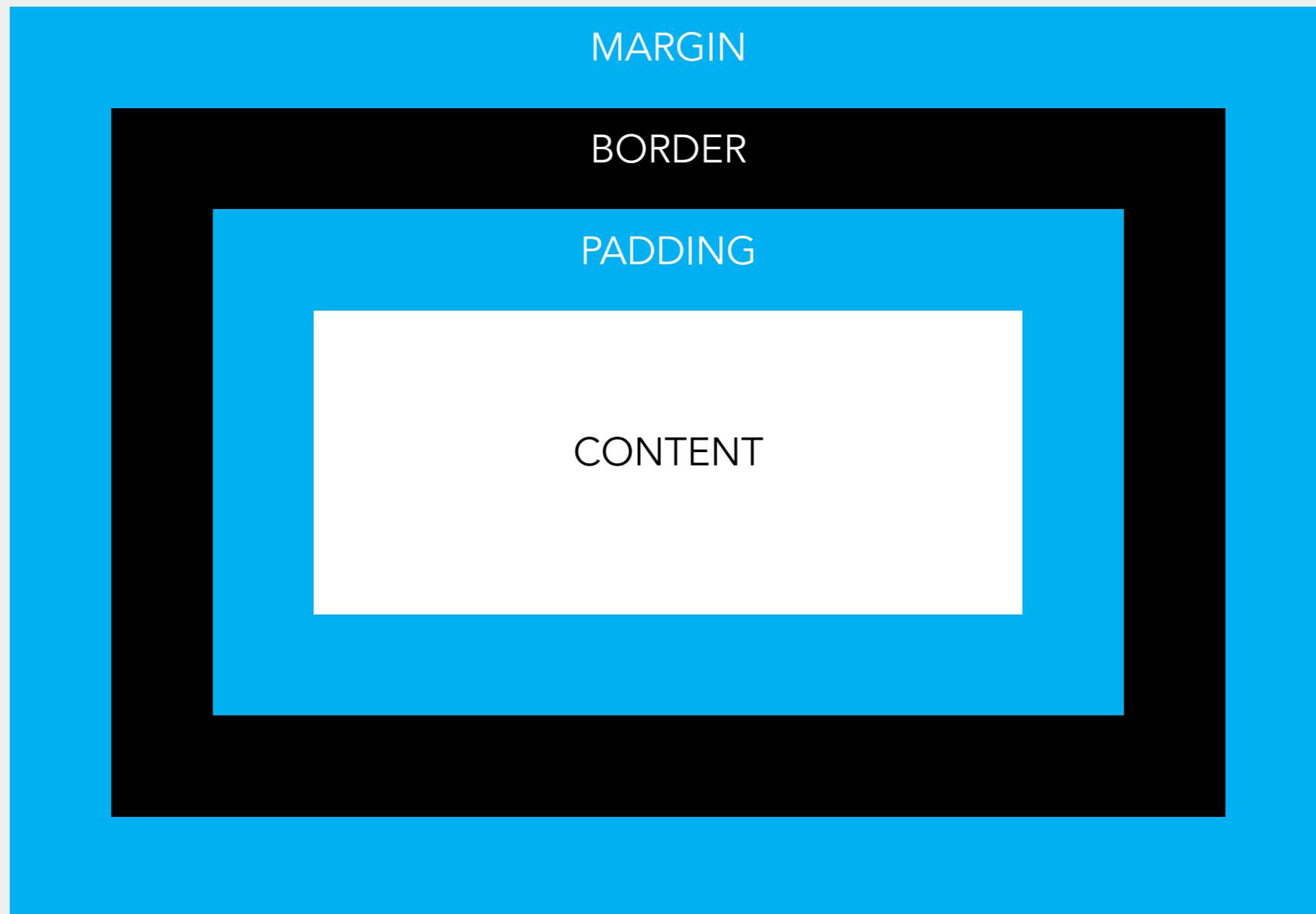
```
<!DOCTYPE html>
<html> event scroll
  > <head> ...
  > <body>
    > <nav class="navContainer">...
    
    <h1 class="u-textCenter">Shannen Wu</h1>
    <hr>
    > <div class="u-textCenter">...
    > <div class="u-textCenter">...
  </body>
</html>
```

html > body

# Margin



# Box Model



# Box Model

The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

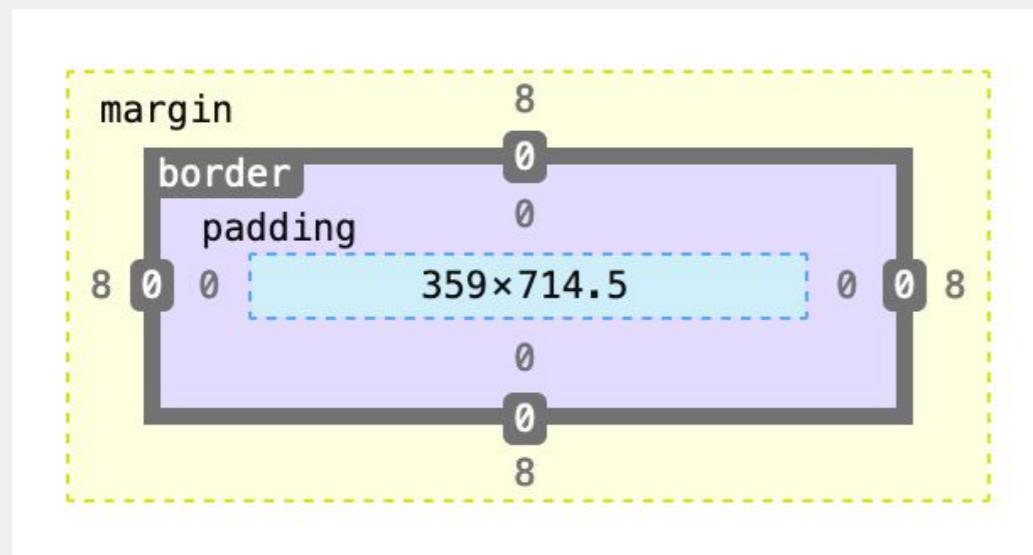
```
1 <div class="box">
2   content
3 </div>
```

The 'CSS' tab contains the following code:

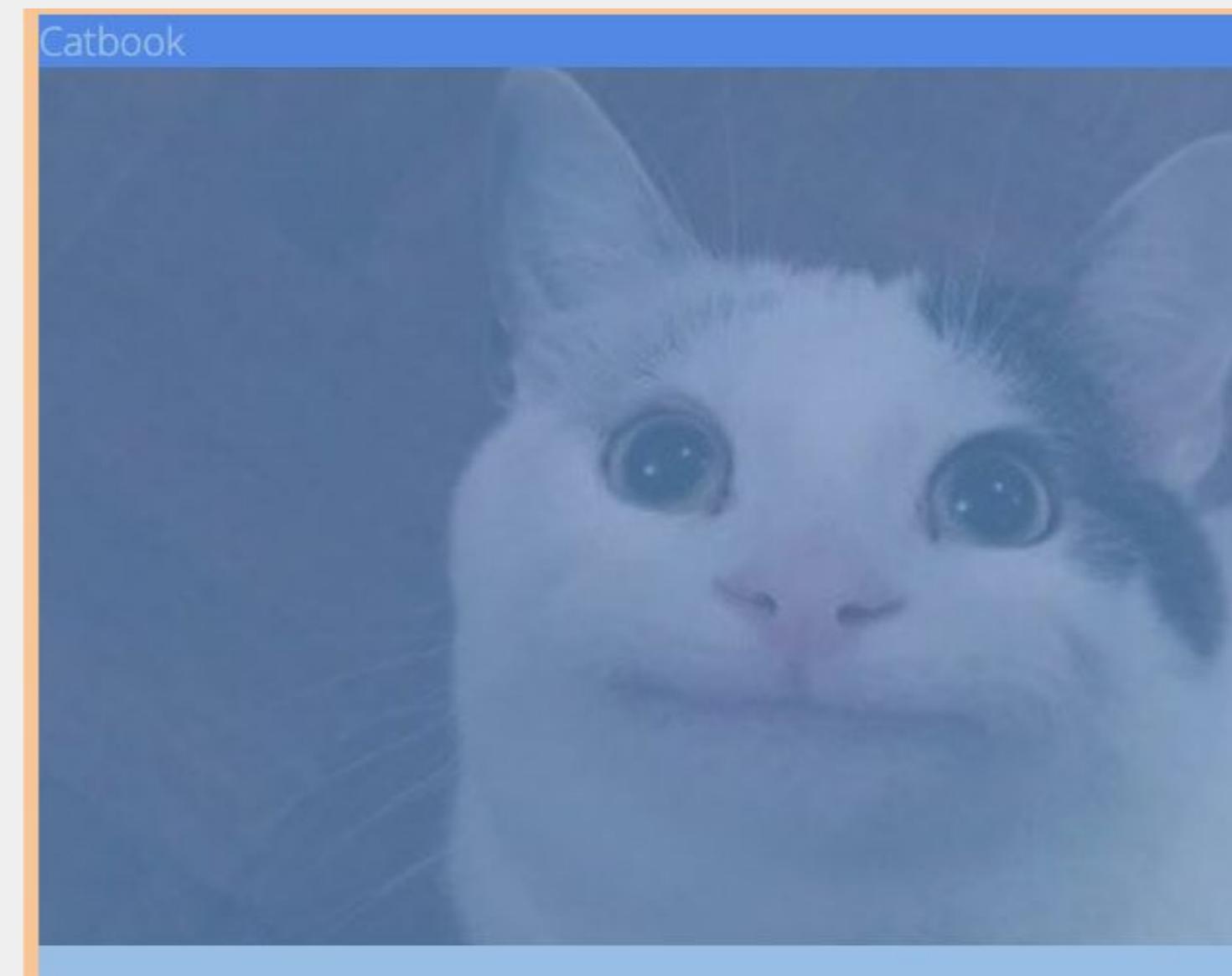
```
1 .box {
2   margin: 0px;
3   border: 1px solid black;
4   padding: 10px;
5 }
```

Below the code editor, there is a preview window showing a white rectangular box with the word 'content' centered inside it. The box has a thin black border and a 10px padding around the content text.

# Margin



```
html,  
body {  
  margin: 0;
```



# Ordering

Some CSS attributes can take multiple values, like padding and margin.

# Ordering

Some CSS attributes can take multiple values, like padding and margin.

`margin: 10px;`

**10px**

10px

10px

10px

# Ordering

Some CSS attributes can take multiple values, like padding and margin.

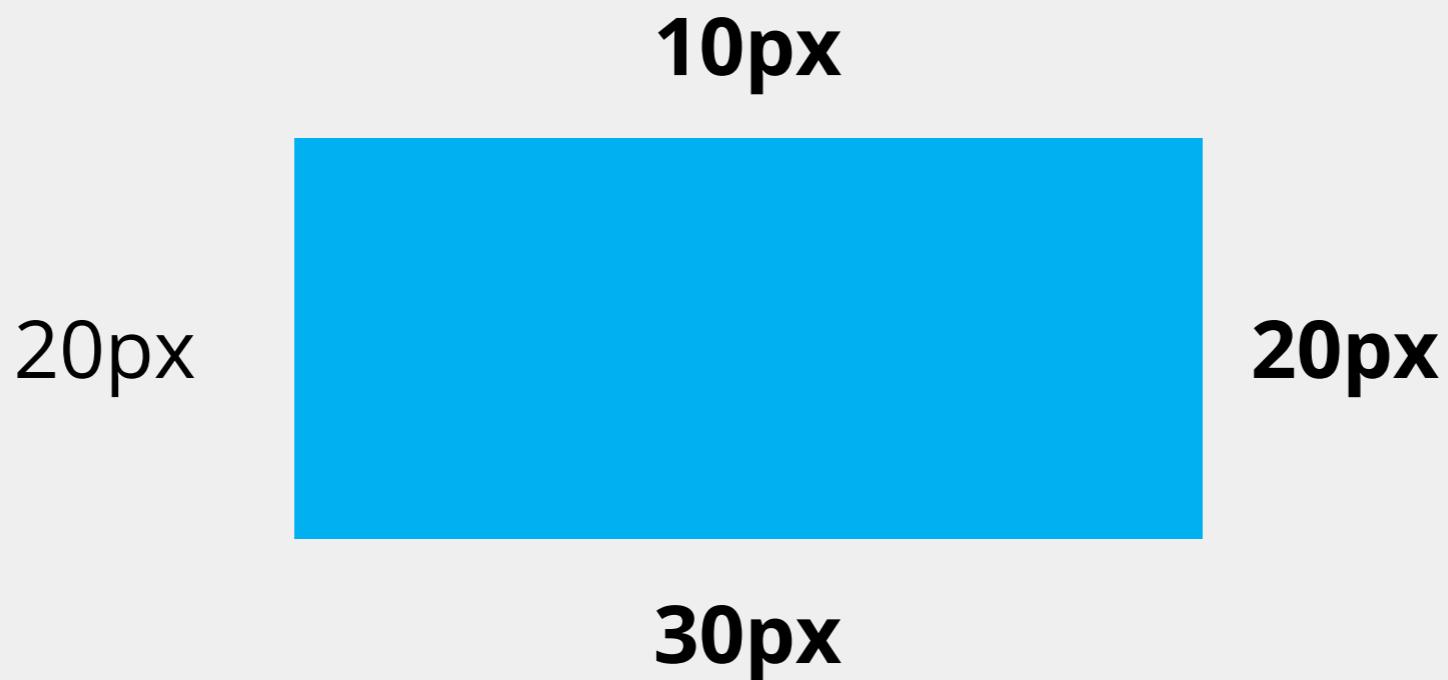
```
margin: 10px 20px;
```



# Ordering

Some CSS attributes can take multiple values, like padding and margin.

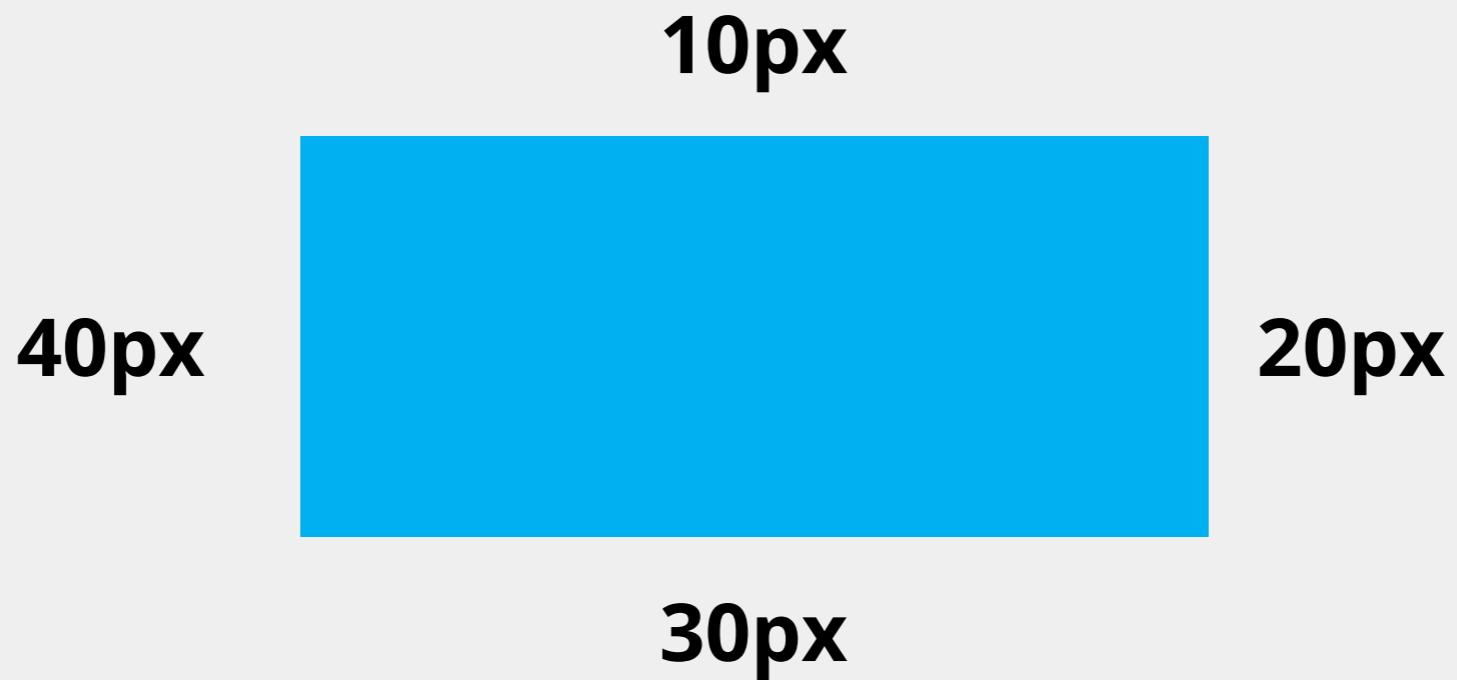
```
margin: 10px 20px 30px;
```



# Ordering

Some CSS attributes can take multiple values, like padding and margin.

```
margin: 10px 20px 30px 40px;
```

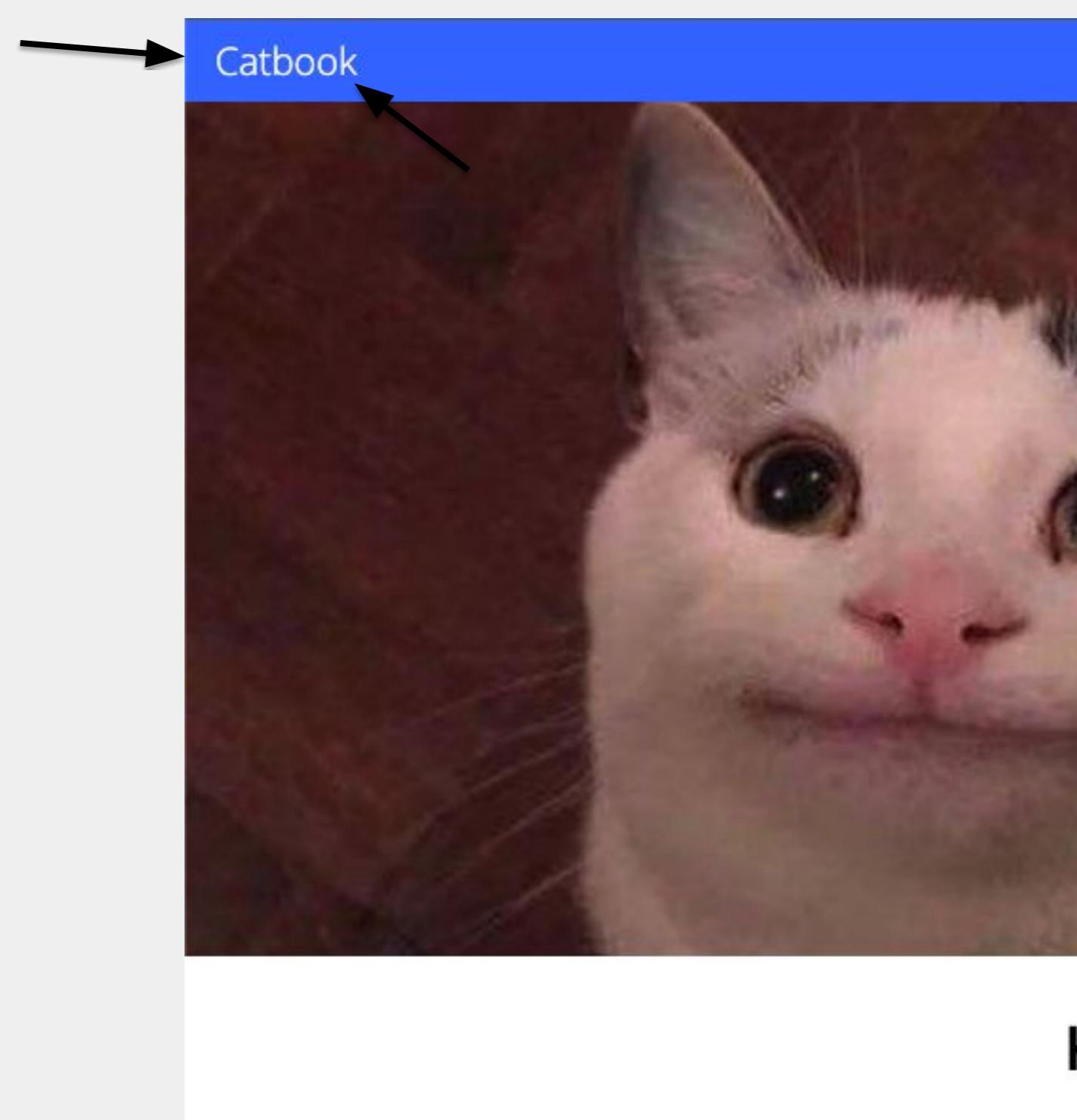
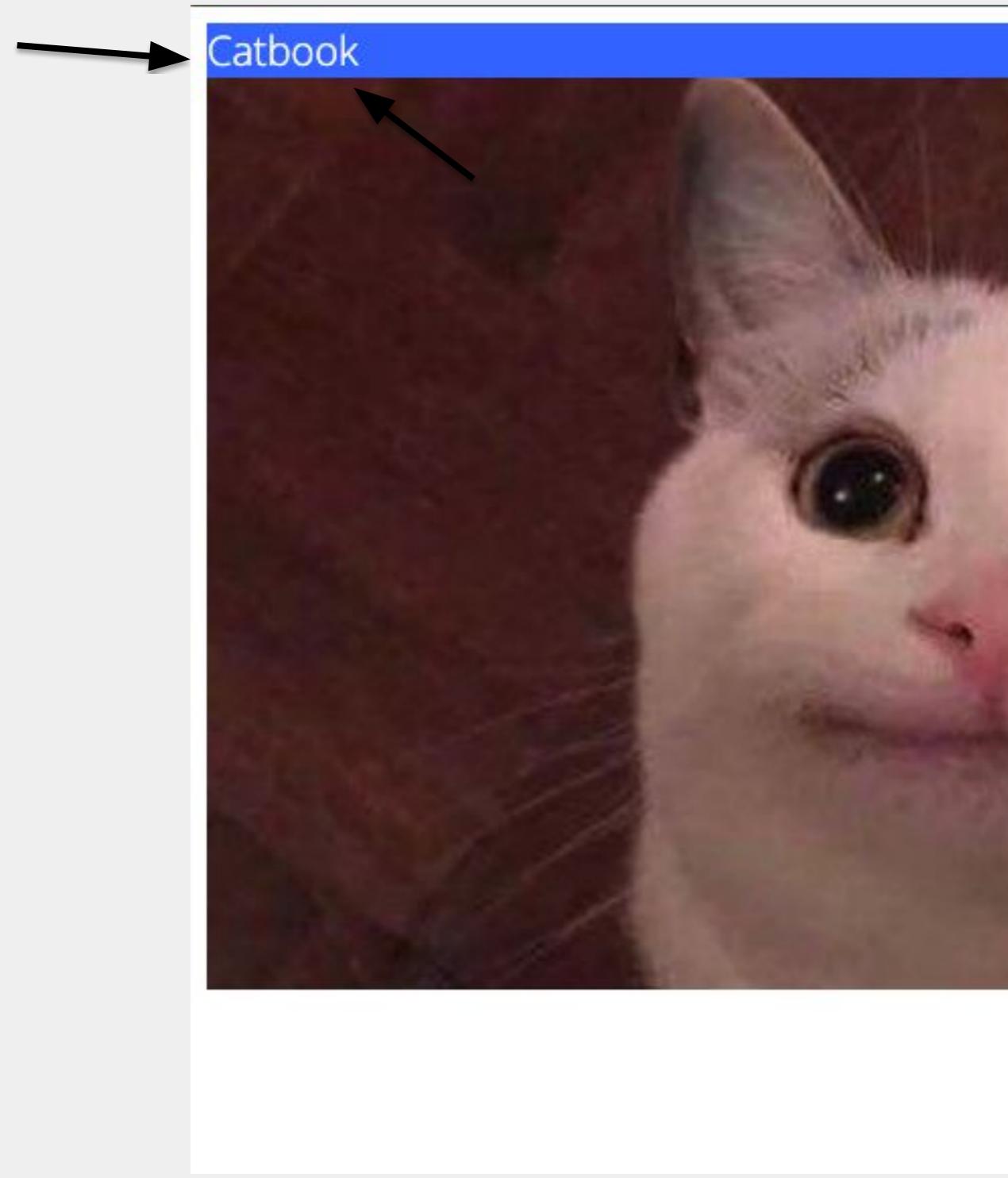


# Ordering

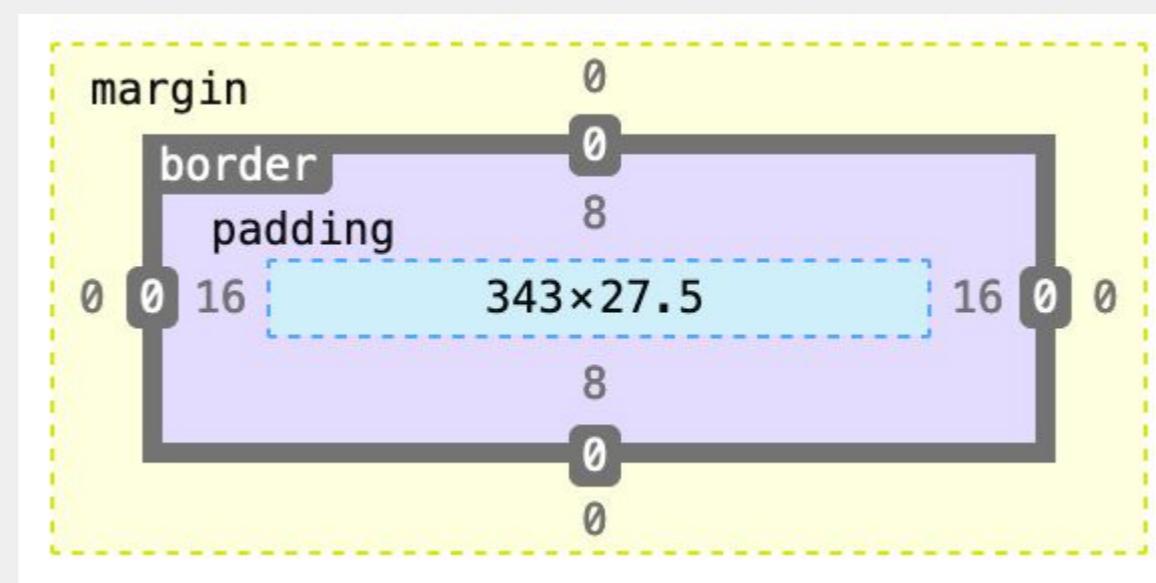
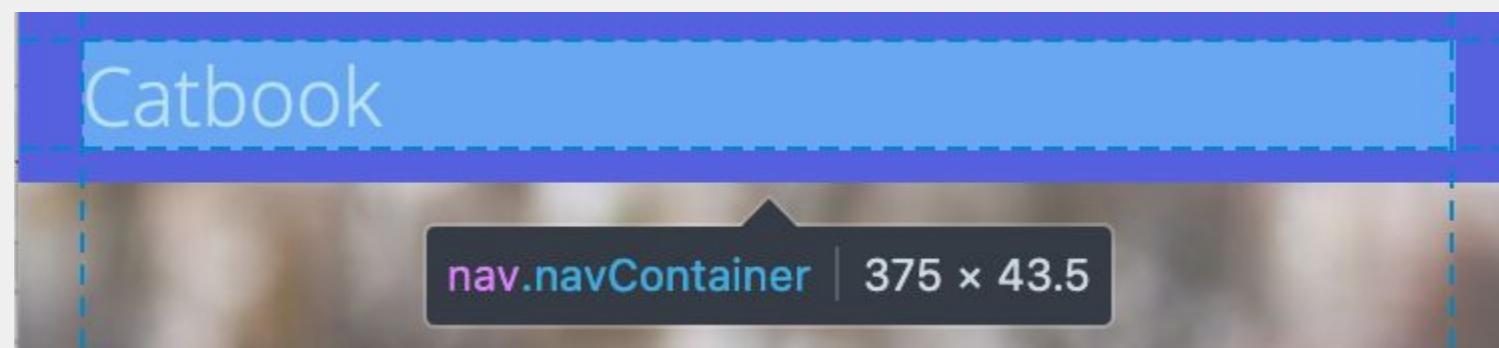
Some CSS attributes can take multiple values, like padding and margin.



# Blank Space



# Padding

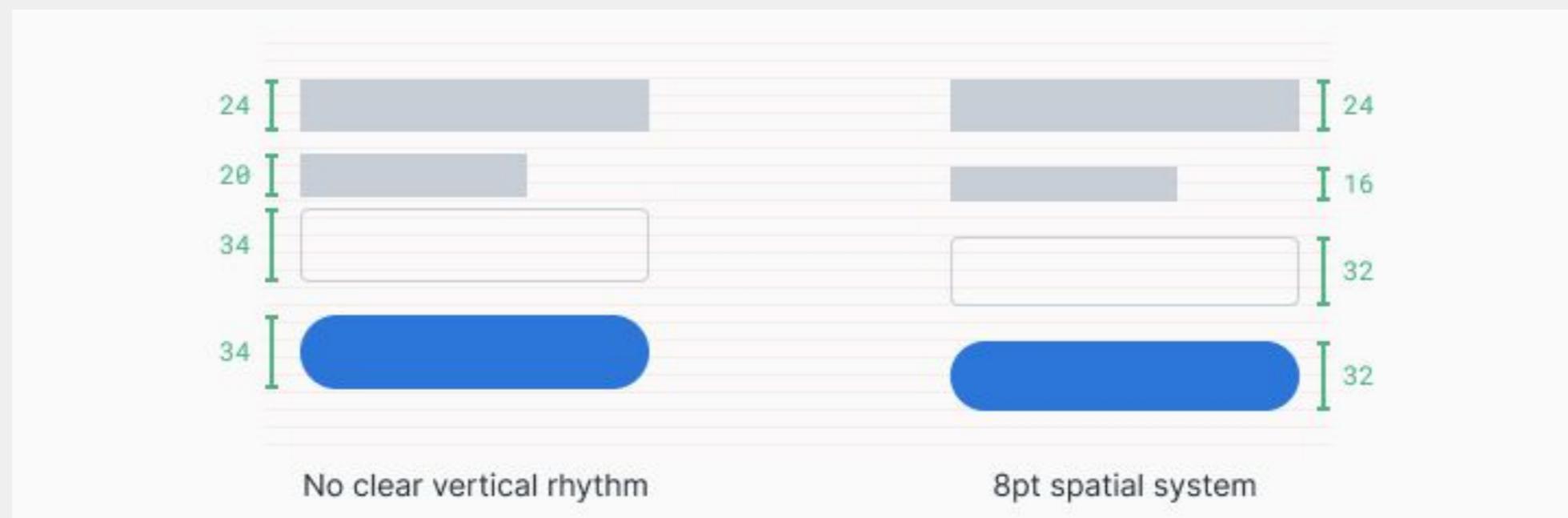


# Padding

We could arbitrarily add some px padding, but...

# 8pt Grid System

# 8pt Grid System



# 8pt Grid System

```
:root {  
  --primary: #396dff;  
  --grey: #f7f7f7;  
  --white: #fff;  
  
  --xs: 4px;  
  --s: 8px;  
  --m: 16px;  
  --l: 24px;  
}
```

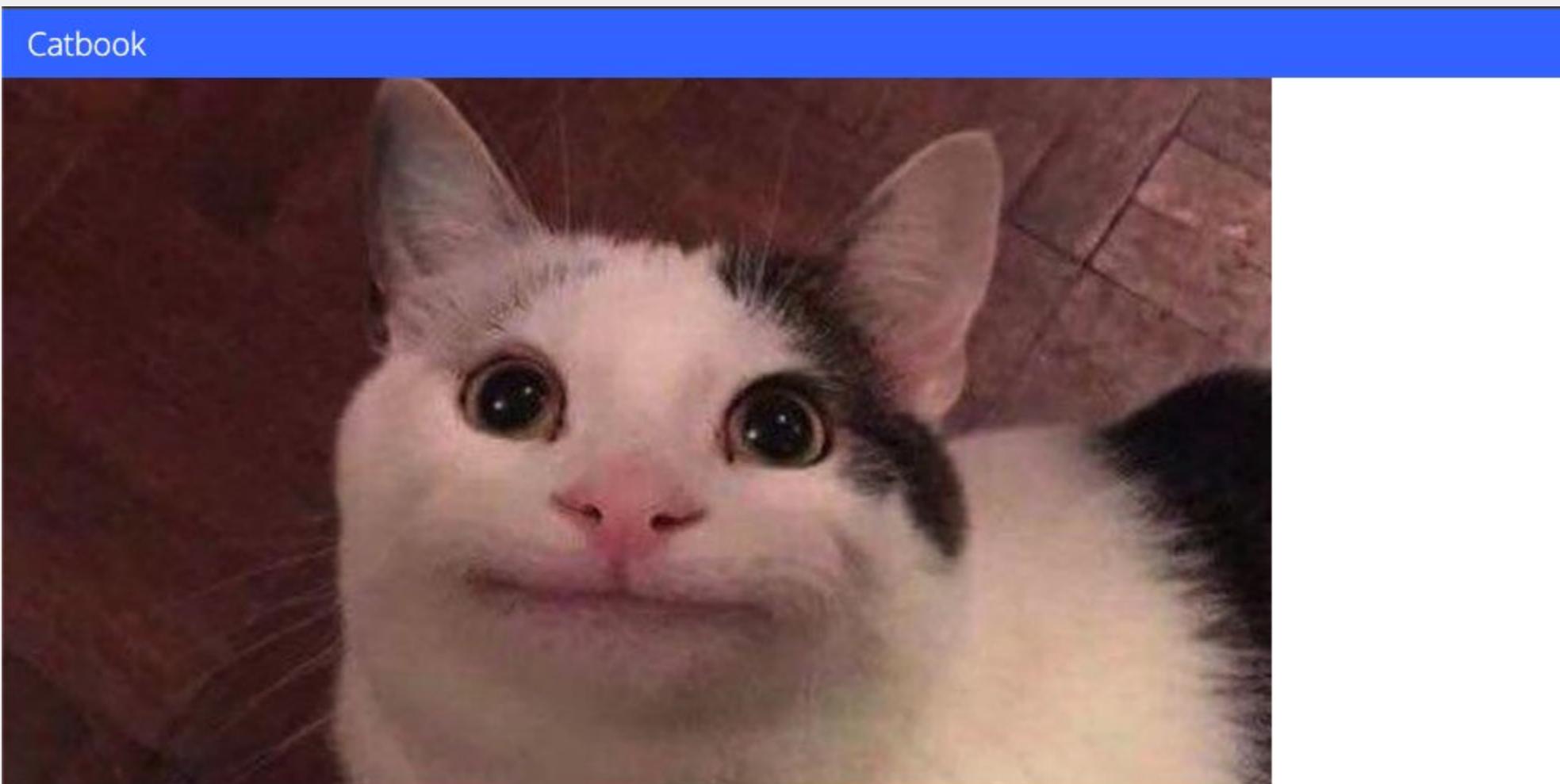
# 8pt Grid System

```
:root {  
  --primary: #396dff;  
  --grey: #f7f7f7;  
  --white: #fff;  
  
  --xs: 4px;  
  --s: 8px;  
  --m: 16px;  
  --l: 24px;  
}  
  
.navContainer {  
  padding: var(--s) var(--m);  
  background-color: var(--primary);  
}
```

# Blank Space

(Thanh's Version)

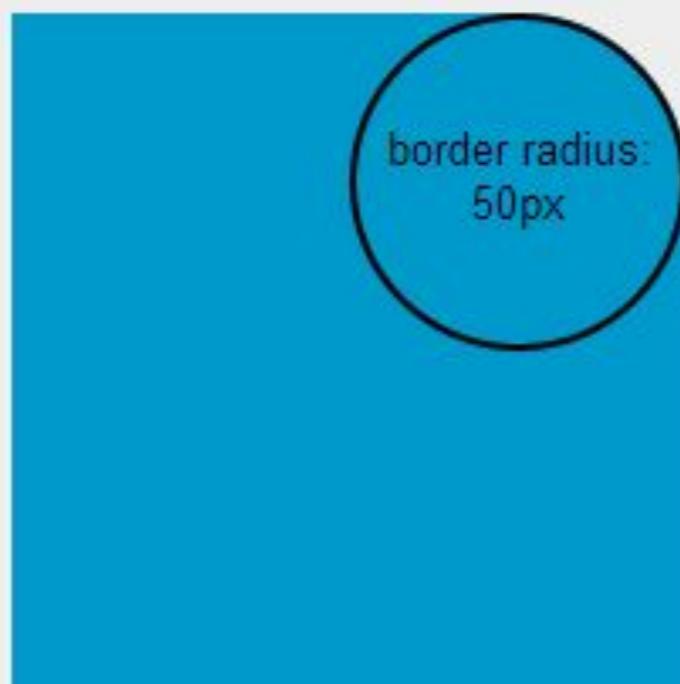
git checkout w0-step8



**Kevin Sun Chen**

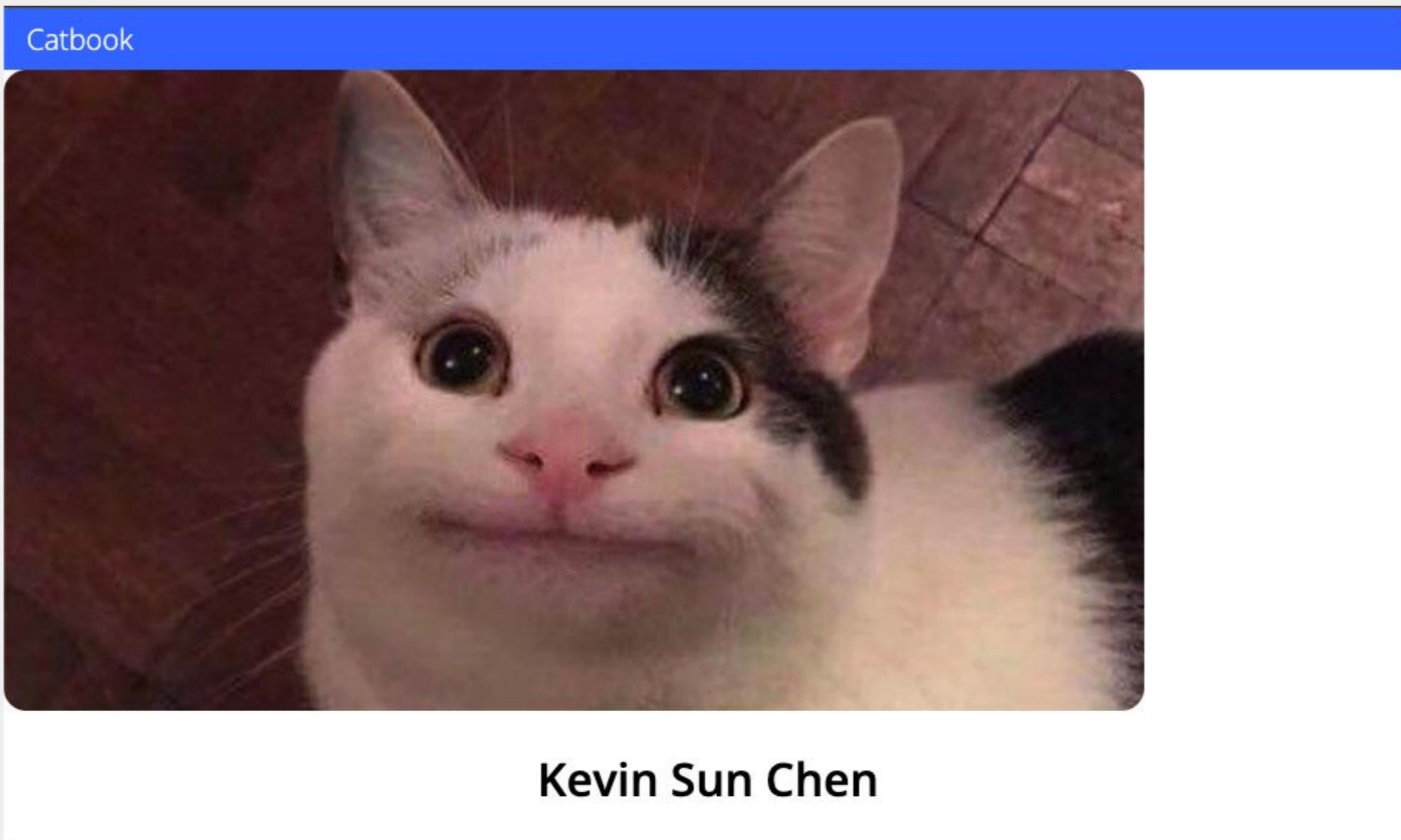
# Exercise 5: Feline Good

Let's talk about border-radius



# Exercise 5: Feline Good

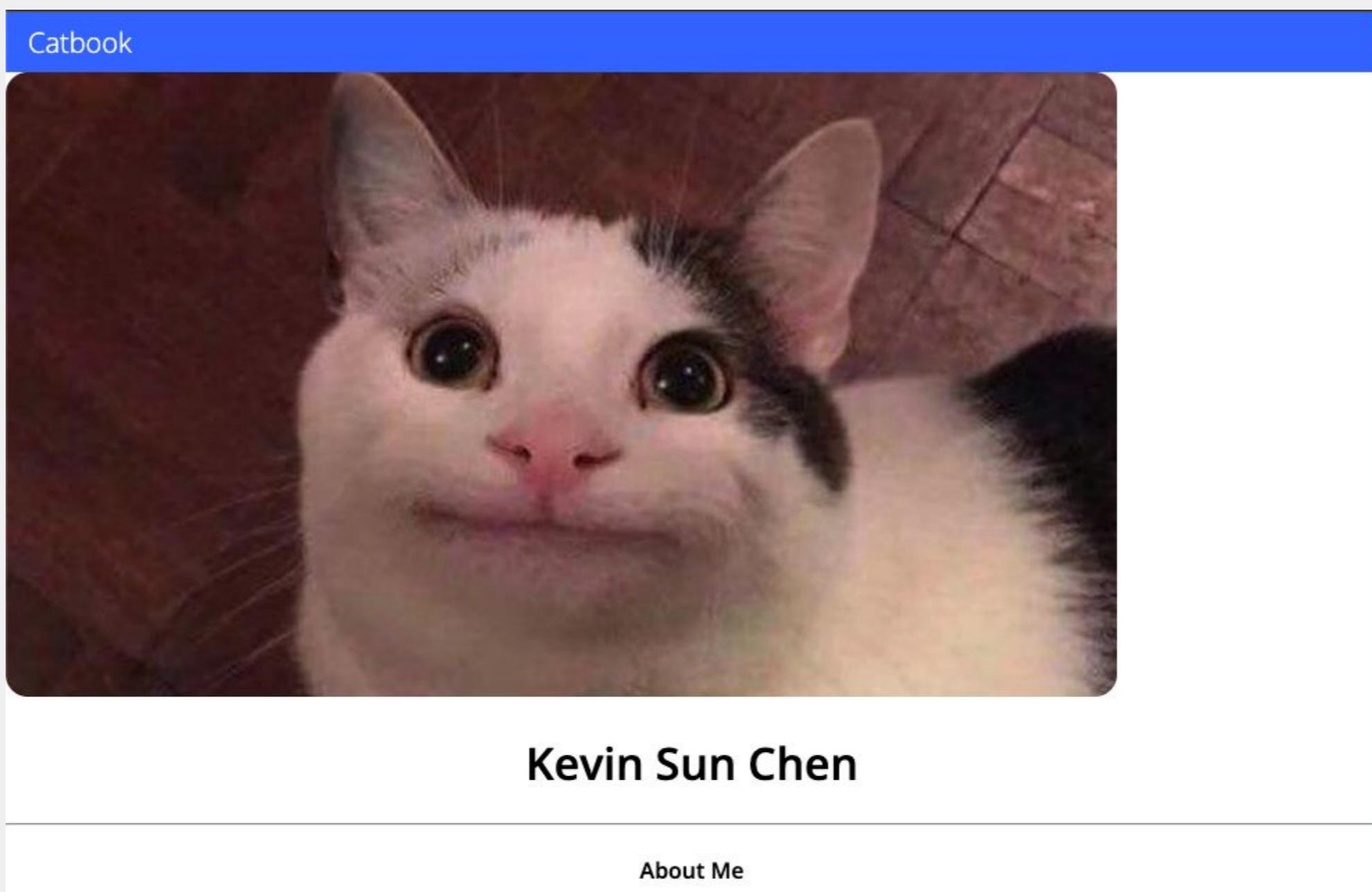
Using the variables we defined for the 8pt system,  
try rounding the corners of your profile picture.



# Exercise 5: Feline Good

git checkout w0-step9

Catbook



Kevin Sun Chen

---

About Me

# Exercise 6: Stay Pawsitive

But can you make it perfectly round?

Catbook



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook

**My Favorite Type of Cat**

I actually like dogs

# Exercise 6: Stay Pawsitive

But can you make it perfectly round?

✨ The Internet is Your Friend! ✨

or you can just peep at our  
code :P

git checkout w0-step10

# So Close!

Catbook



**Kevin Sun Chen**

## About Me

I am more of a dog person but I'm just trying to fit in and get catbook

## My Favorite Type of Cat

I actually like dogs

Catbook



**Kevin Sun Chen**

## My Favorite Type of Cat

I am more of a dog person but I'm just trying to fit in and get  
catbook

## My Favorite Type of Cat

I actually like dogs

# So Close!

Catbook



**Kevin Sun Chen**

## About Me

I am more of a dog person but I'm just trying to fit in and get catbook

## My Favorite Type of Cat

I actually like dogs

Catbook



**Kevin Sun Chen**

## My Favorite Type of Cat

I am more of a dog person but I'm just trying to fit in and get catbook

## My Favorite Type of Cat

I actually like dogs

✨ The Internet is Your Friend! ✨

## Float Example

```
.column {  
    float: left;  
    width: 50%;  
}  
  
/* Clear floats after the columns */  
.row:after {  
    content: "";  
    display: table;  
    clear: both;  
}
```

[Try it Yourself »](#)

A modern way of creating two columns, is to use [CSS Flexbox](#). However, it is not supported in Internet Explorer 10 and earlier versions.

## Flex Example

```
.row {  
    display: flex;  
}  
  
.column {  
    flex: 50%;  
}
```

[Try it Yourself »](#)

## Float Example

```
.column {  
    float: left;  
    width: 50%;  
}  
  
/* Clear floats after the columns */  
.row:after {  
    content: "";  
    display: table;  
    clear: both;  
}
```



[Try it Yourself »](#)

A modern way of creating two columns, is to use [CSS Flexbox](#). However, it is not supported in Internet Explorer 10 and earlier versions.

## Flex Example

```
.row {  
    display: flex;  
}  
  
.column {  
    flex: 50%;  
}
```

[Try it Yourself »](#)

## Float Example

```
.column {  
    float: left;  
    width: 50%;  
}  
  
/* Clear floats after the columns */  
.row:after {  
    content: "";  
    display: table;  
    clear: both;  
}
```



[Try it Yourself »](#)

A modern way of creating two columns, is to use [CSS Flexbox](#). However, it is not supported in Internet Explorer 10 and earlier versions.

## Flex Example

```
.row {  
    display: flex;  
}  
  
.column {  
    flex: 50%;  
}
```

[Try it Yourself »](#)

✨ The Internet **can** be Your Friend! ✨

## Float Example

```
.column {  
    float: left;  
    width: 50%;  
}  
  
/* Clear floats after the columns */  
.row:after {  
    content: "";  
    display: table;  
    clear: both;  
}
```

[Try it Yourself »](#)

A modern way of creating two columns, is to use [CSS Flexbox](#). However, it is not supported in Internet Explorer 10 and earlier versions.

## Flex Example

```
.row {  
    display: flex;  
}  
  
.column {  
    flex: 50%;  
}
```



[Try it Yourself »](#)



This is still bad tho cuz  
w3 schools does it  
wrong





# Flex



Flexbox is a flexible box that lets you control the direction, sizing, distribution, and more of items.



ALIGNMENT



DIRECTION



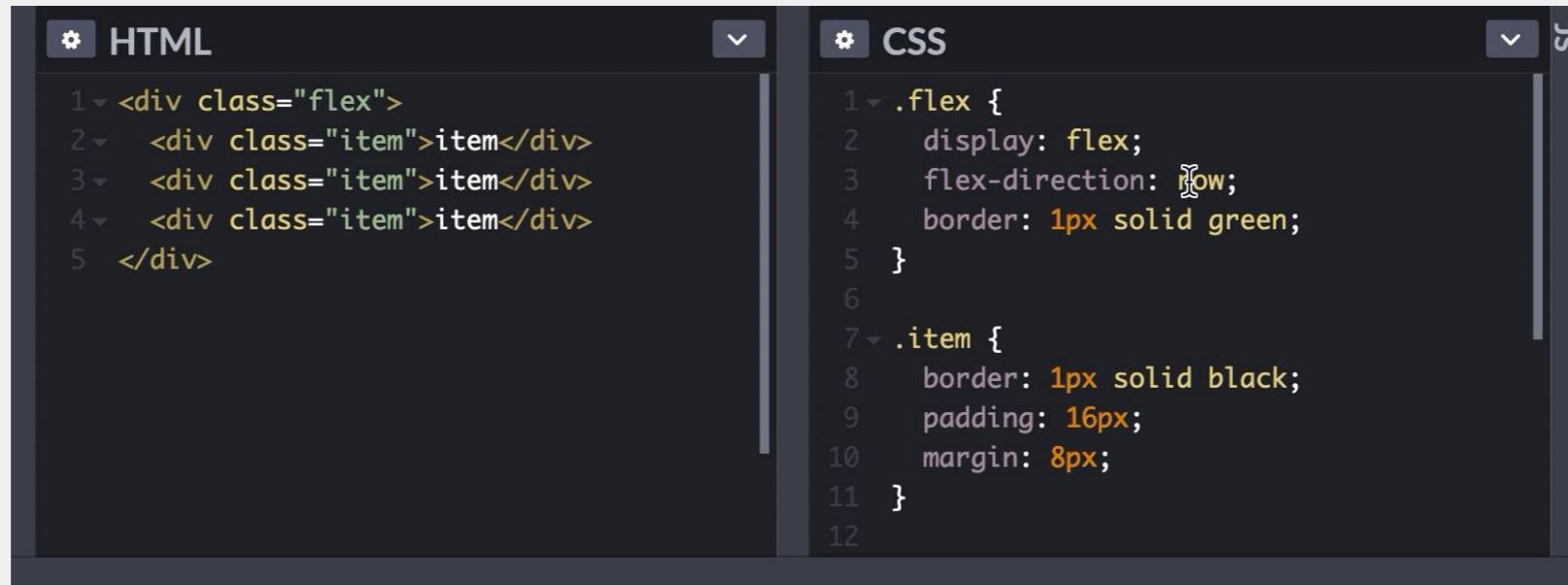
ORDER



SIZE

# Flex

Flexbox is a flexible box that lets you control the direction, sizing, distribution, and more of items.



The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="flex">
2   <div class="item">item</div>
3   <div class="item">item</div>
4   <div class="item">item</div>
5 </div>
```

The 'CSS' tab contains the following code:

```
1 .flex {
2   display: flex;
3   flex-direction: row;
4   border: 1px solid green;
5 }
6
7 .item {
8   border: 1px solid black;
9   padding: 16px;
10  margin: 8px;
11 }
```



# Flex



Catbook



**Kevin Sun Chen**

#### About Me

I am more of a dog person but I'm just trying to fit in and get catbook

#### My Favorite Type of Cat

I actually like dogs

Catbook



**Kevin Sun Chen**

#### My Favorite Type of Cat

I am more of a dog person but I'm just trying to fit in and get catbook

#### My Favorite Type of Cat

I actually like dogs

# Flex

Let's just try adding a flex box around those sections!

# Flex



## git checkout w0-step11

Catbook



**Kevin Sun Chen**

---

**About Me**

I am more of a dog person but I'm just trying to fit in and get catbook I actually like dogs

**My Favorite Type of Cat**

# Flex



Hmmm

Catbook



**Kevin Sun Chen**

---

**About Me**

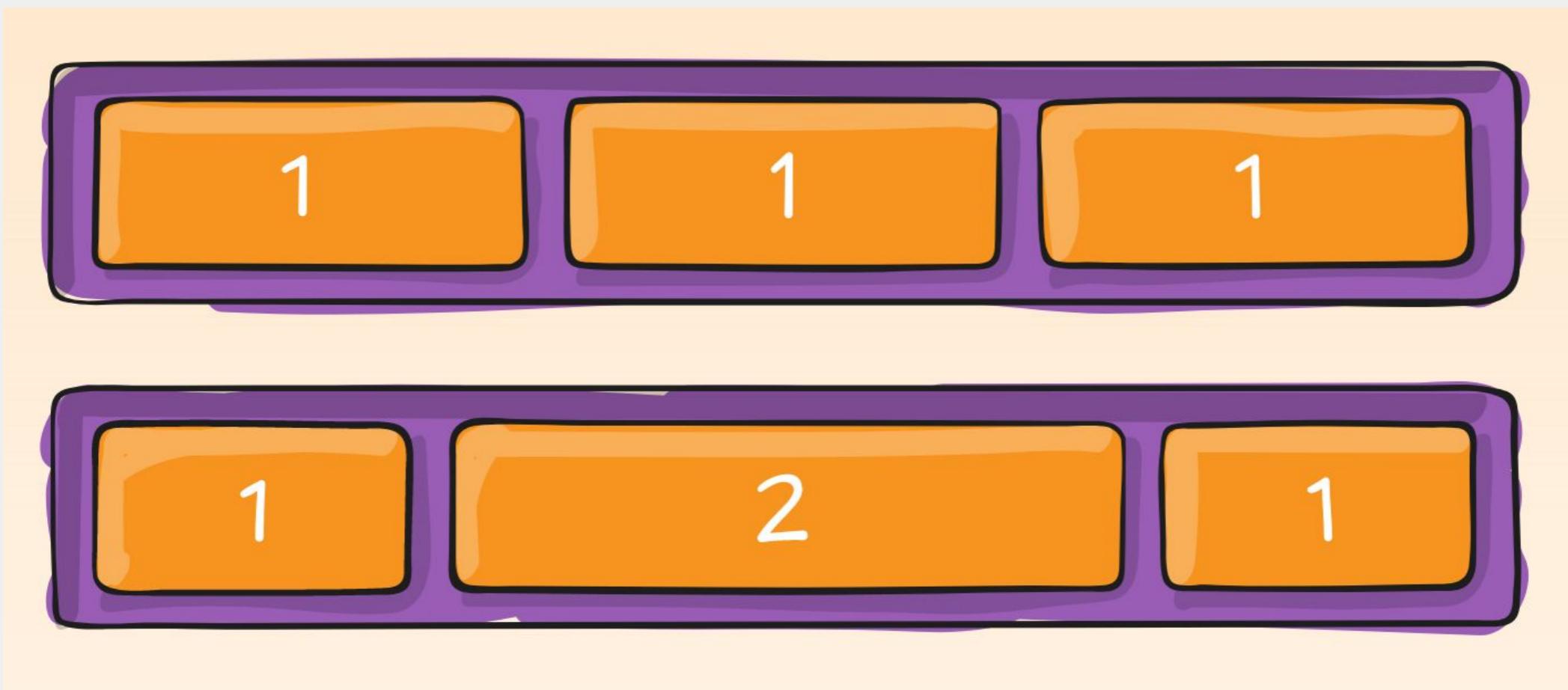
I am more of a dog person but I'm just trying to fit in and get catbook. I actually like dogs

**My Favorite Type of Cat**

# Flex



Flexbox is a flexible box that lets you control the direction, **sizing**, distribution, and more of items.



# flex-basis



We can define the “default” size of items.

The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="flex">
2   <div class="item small">item</div>
3   <div class="item large">item</div>
4   <div class="item small">item</div>
5 </div>
```

The 'CSS' tab contains the following code:

```
13
14 .small {
15   flex-basis: 0;
16 }
17
18 .large {
19   flex-basis: 0;
20 }
```

Below the code editor, there is a visual representation of three items: 'item', 'item', and 'item'. The first item is larger than the second and third, which are of equal size. This visualizes how the 'flex-basis' property affects the size of flex items.

# flex-grow



We can define what items gets to take up more space.

The screenshot shows a code editor with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="flex">
2   <div class="item grow">item</div>
3   <div class="item groow">item</div>
4   <div class="item grow">item</div>
5 </div>
```

The 'CSS' tab contains the following code:

```
14 .grow {
15   flex-grow: 1;
16 }
17
18 .groow {
19   flex-grow: 0;
20 }
```



# Exercise 7: A Little Fur-ther

Using this overview and [weblab.to/flex](https://weblab.to/flex) 🐈, try and make the columns the same size

Catbook



Kevin Sun Chen

---

|                                                                                            |                                                        |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>About Me</b><br>I am more of a dog person but I'm just trying to fit in and get catbook | <b>My Favorite Type of Cat</b><br>I actually like dogs |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------|

# Exercise 7: A Little Fur-ther

git checkout w0-step12

Catbook



**Kevin Sun Chen**

---

|                                                                         |                                |
|-------------------------------------------------------------------------|--------------------------------|
| <b>About Me</b>                                                         | <b>My Favorite Type of Cat</b> |
| I am more of a dog person but I'm just trying to fit in and get catbook | I actually like dogs           |

Congrats! You now know a tiny bit about  
the vast sea that is CSS!

You're a CSS Meowster!

# Flexing is Hard

We recommend the guide linked in: [weblab.to/flex](https://weblab.to/flex)

And these games:

<https://flexboxfroggy.com/> - good for learning 

<http://www.flexboxdefense.com/> - good for 

# Touch it Up!

We have some homework lined up for you to practice some simpler styling:

`git checkout w0-step13`

If you want to peek the answers later:

`git checkout w0-complete`

There's so much more CSS I could teach you,  
but that's a lesson for another time...

Specifically, the *Intermediate CSS* lecture later this  
week!!

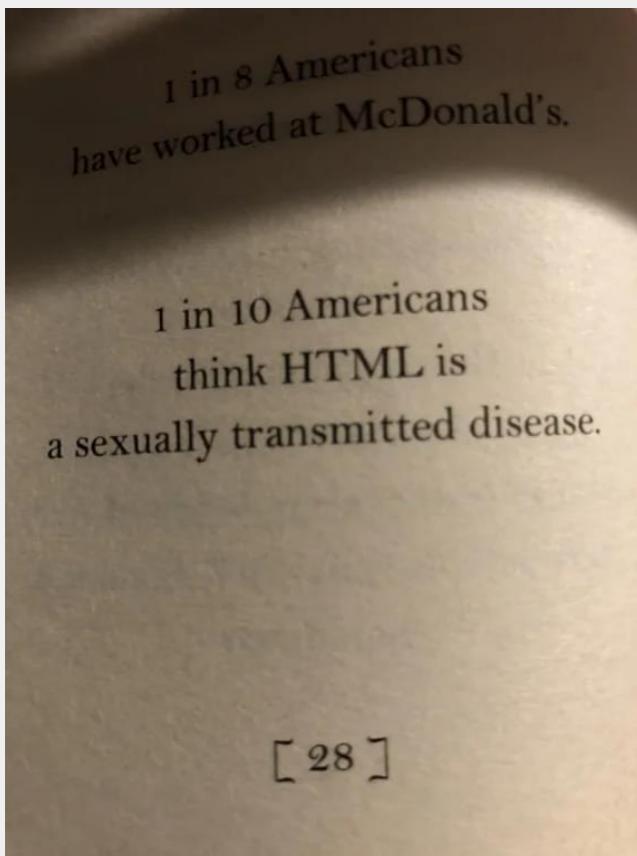
**THIS IS HOW I FELT**

**WHEN I REALIZED THE  
POWER OF FLEX**

# Setup Homework

[weblab.to/homework1](http://weblab.to/homework1)

# Memes



me: let's rewrite the CSS  
my website:

