FIRST HTML FILE

- Create a work directory
  - Helpful habit to learn early: Organize your work
- Create a `index.html` file:

```
Hello World
```

- In Chrome: File->Open File->Select your index.html

BROWSERS ARE TOLERANT

Inspect the rendered page:

- Right-Click -> Inspect
- See the elements in the Elements sub-tab

See all the elements the browser "assumed" for you

YOU DO NOT WANT TO RELY ON THIS!

It will fail you later

YOUR SECOND HTML FILE

Edit `index.html`:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Second HTML File</title>
</head>
<body>
  <p>Hello Again, World</p>
  Here
  Is
  More     Text
</body>
</html>
```

HTML BASICS

Remember, tags may nest but may not overlap.

Notice how the whitespace visually collapses to one space.

- newlines between tags will be a space!

Notice the uppercase/lowercase difference in "doctype"

- We will use lowercase for tags

REAL HTML CASE

Imagine a chat application

- A list of users
- A list of messages (text, sender, timestamp, avatar)
- Somewhere to type
- A button to send

DO NOT THINK IN TERMS OF HOW IT WILL LOOK

"Semantic" is about what it is and what it means, not what it looks like

CHAT - HIGH LEVEL

HTML is a series of nested and/or sibling containers

Page (Document)

- List of Users
- List of Messages
- Typing Area

CHAT - SOME DETAILS

Page (Document)

- List of Users
- List of Messages
    - Each Message
        - Avatar
        - Username
        - Timestamp
        - Text
- Typing Area
    - Input area for message to send
    - Send Button

CHAT - STRUCTURAL BONES

(contents of `<body>`)

```
<div id="chat-app">
  <ul id="users">
  </ul>
  <ol id="messages">
  </ol>
  <div id="outgoing">
  </div>
</div>
```

Why a base `<div>` for the app at all, why not just put contents in `<body>`?

Why are some `<ol>` and `<ul>` (ordered/unordered lists) and some `<div>`?

WHY DEM BONES?

Why base `<div>` and not just contents in `<body>`?

- Allows contents to be managed as a unit
    - Formatting
    - Add to page (controls, non-app details, ads, etc)

Why are some `<ol>`, `<ul>` (lists) and some `<div>`?

- Semantics
    - `<ul>` contents are related to each other
    - `<div>` are unrelated containers

HOW TO DECIDE ON TAGS

Why `<ol>` vs `<ul>`?

- Does order matter?

Why `<div>` and not `<p>` or `<span>`?

- `<p>` is a paragraph
- `<span>` is a portion of text
- `<div>` is very generic - be specific when you can, but you often can't

MDN is your friend. Google: `MDN ul`

Semantics are arguable

ADDING FLESH TO THE BONES

Still need more details

```
<div id="chat-app">
  <ul id="users">
  </ul>
  <ol id="messages">
  </ol>
  <div id="outgoing">
  </div>
</div>
```

FLESHING OUT USER LIST

```
<ul id="users">
  <li>
    <div class="user">
      <span class="username">Amit</span>
    </div>
  </li>
  <li>
    <div class="user">
      <span class="username">Bao</span>
    </div>
  </li>
</ul>
```

BUT WHY

```
<li>
  <div class="user">
    <span class="username">Amit</span>
  </div>
</li>
```

We could make `<li class="user">`, but what if you want to list a user elsewhere that isn't in a list?

We could skip the `<span>`, but what if we want to add more to user (avatar? last active? status message?)

In Reality you might not worry about those until they come up, but see how the semantics make those an option.

FLESHING OUT MESSAGE LIST

```
<ol id="messages">
  <li>
    <div class="message">
      <div class="sender">
        <img class="avatar" alt="user-avatar" src="/images/avatar-amit.jpg"/>
        <span class="username">Amit</span>
        <span class="timestamp">18:45:32</span>
      </div>
      <p class="message-text">You up?</p>
    </div>
  </li>
</ol>
```

ARGUABLE, BUT WHAT ARGUMENTS?

- `<div class="message">` not just `<li>`?
- `<div class="sender">`?
- `<img class="avatar" .../>` not in a `<div>`?
- `<span class="username">` not a `<p>`
- `<span class="timestamp">` a sibling of username and avatar but not of message-text?
- `<p class="message-text">` a `<p>` and not a `<div>`?
- `message-text` and not `text`?

FLESHING OUT THE OUTGOING

```
<div id="outgoing">
  <form action="/chat">
    <input class="to-send" value="" placeholder="Enter message to send"/>
    <button type="submit">Send</button>
  </form>
</div>
```

BUT WHY - OUTGOING

```
<form action="/chat">
```

- We'll cover HTML Forms separately

```
<input class="to-send" .../>
```

- Classes for interact data can be hard to name

```
<button type="submit">Send</button>
```

- Might need a class - let's wait to minimize complexity
- `foo-button` is NOT a good class name
    - but sometimes happens because of hard naming problems

SEEING IT IN ACTION

Now that we have Semantic HTML, we can look at an example.

THAT LOOKS TERRIBLE

Semantics ALLOW for flexible styling that is (mostly) from the CSS.

Writing Semantic HTML will make it easier to create a certain look and to adjust to new needs.

Writing HTML to look a certain way will look better at first, but will be hard to tweak and may break on different devices/platforms.

- Amit
- Bao

1. 

   Amit 18:45:32

   You up?

2. 

   Bao 18:46:50

   Yeah, still working on this INFO6250 work, but I keep getting distracted by cat videos

Enter message to send Send

DIFFERENCES

Notice the changes in what we didn't specify

- Font size, font weight
- Text color, background color
- Image size
- Overall padding and margins

SUMMARY

- Organize even your experimental files
- Browsers are tolerant
    - Don't rely on the tolerance
- HTML whitespace will collapse to single space
- HTML whitespace is for humans (99% of the time)
- Design with Semantics without considering appearance
- Think about the data when considering structure
- Be as specific as you can
    - Often you can't be very specific

SUMMARY - PART 2

Requirements for this class:

- tag names, attributes in lowercase
- HTML attribute values with no space around =
- HTML attribute values quoted with double quotes (")
- Class names will be in all lowercase with hypens (kebab-case, not camelCase, MixedCase, or snake_case)
- Class names identify what the element represents,
    - NOT what it will look like