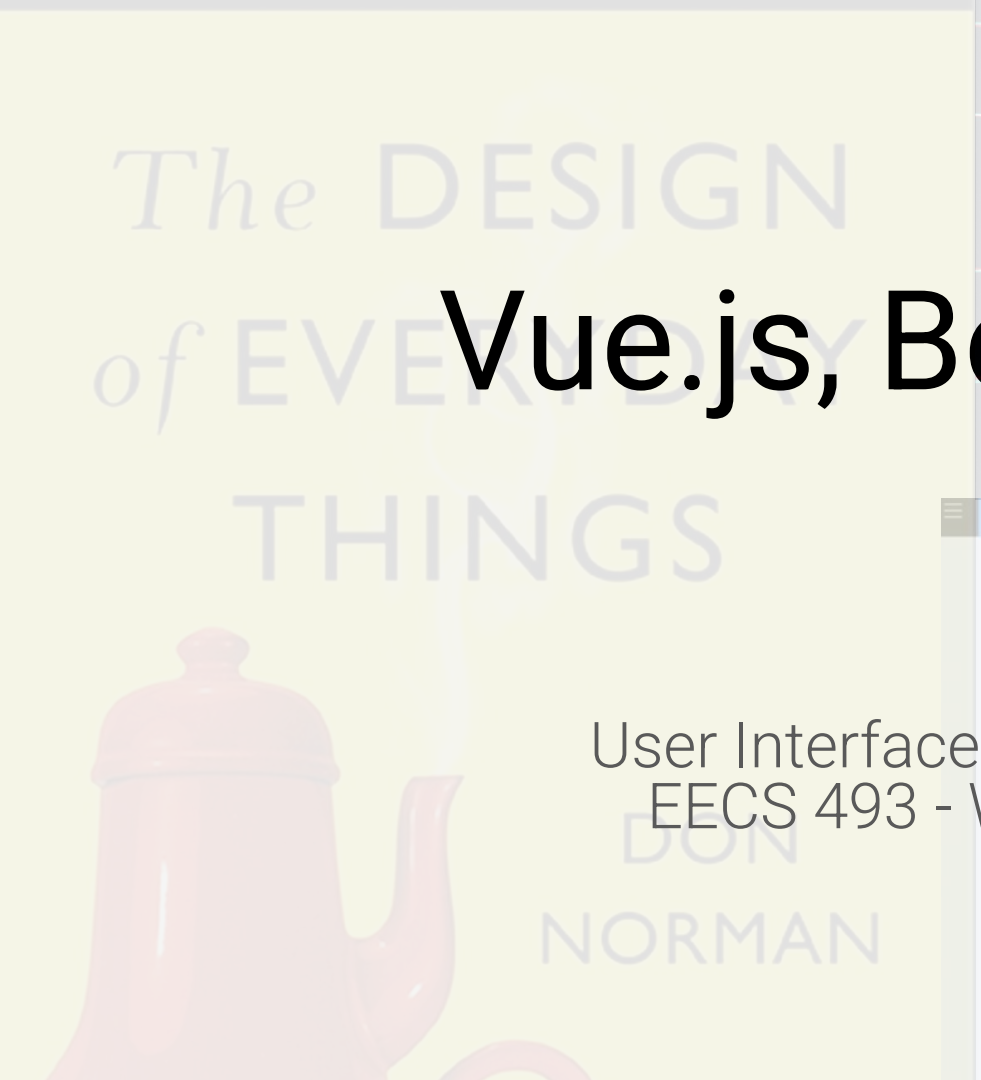
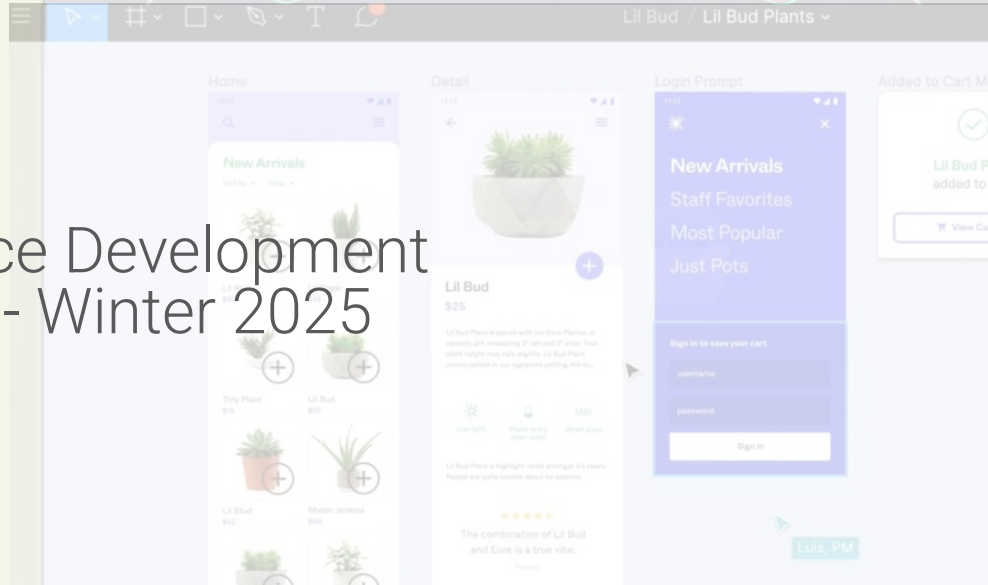
Vue.js, Bootstrap



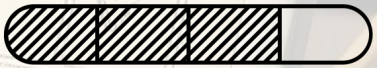


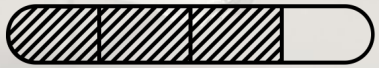
User Interface Development
EECS 493 - Winter 2025

Class progress

Building a toolbox for human-centered software design and development

1. User research
   a. Assignment 1, Final Project all milestones
2. Web programming
   a. Assignment 2, 3, 5
3. Design and prototyping
   a. Assignment 4, Final Project milestones 3-4



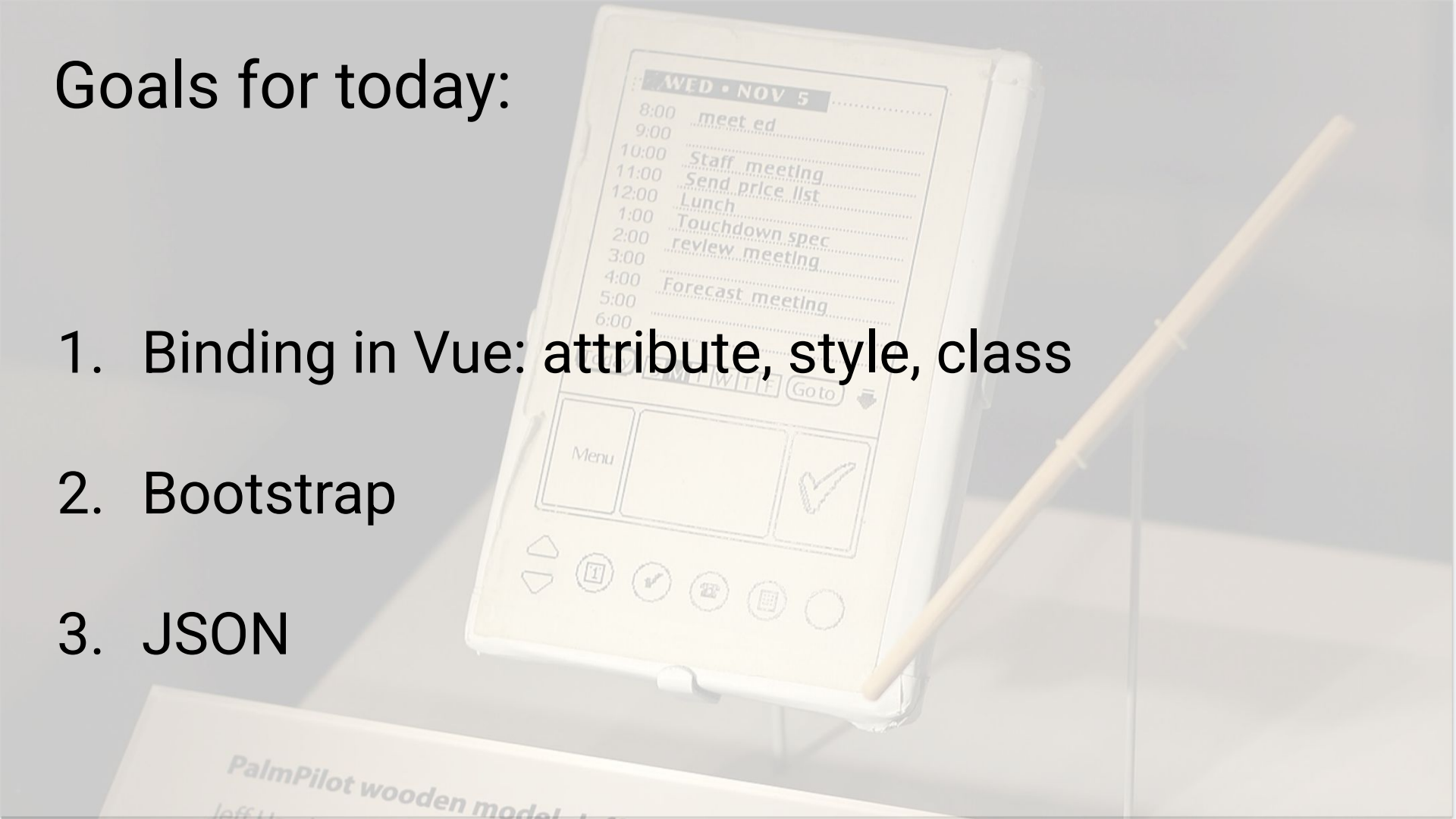


Class progress

Building a toolbox for human-centered software design and development

1. User research
   a. Assignment 1, Final Project all milestones
2. **Web programming**
   a. Assignment 2, 3 (JavaScript, jQuery)
   b. Assignment 5 (Vue.js framework)
3. Design and prototyping
   a. Assignment 4, Final Project milestones 3-4

Goals for today:

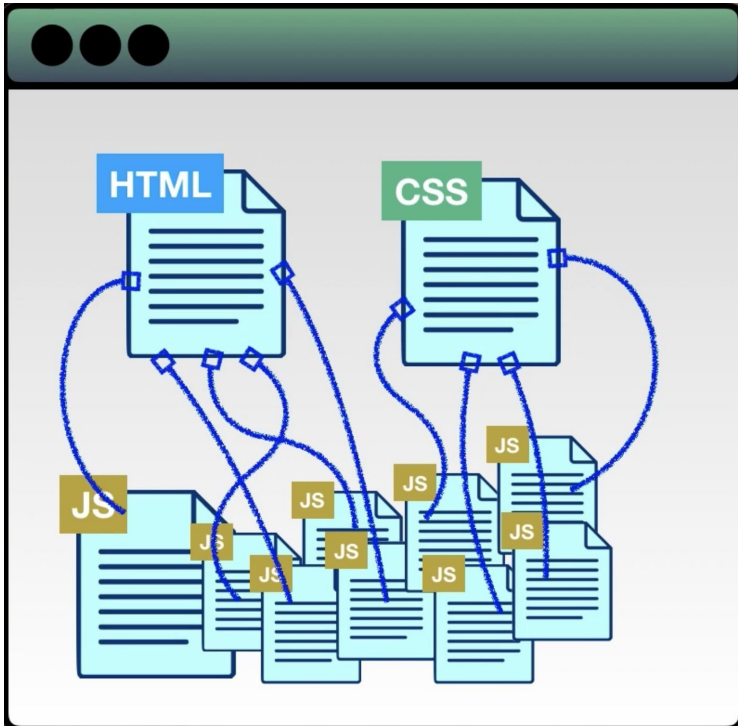


1. Binding in Vue: attribute, style, class
2. Bootstrap
3. JSON

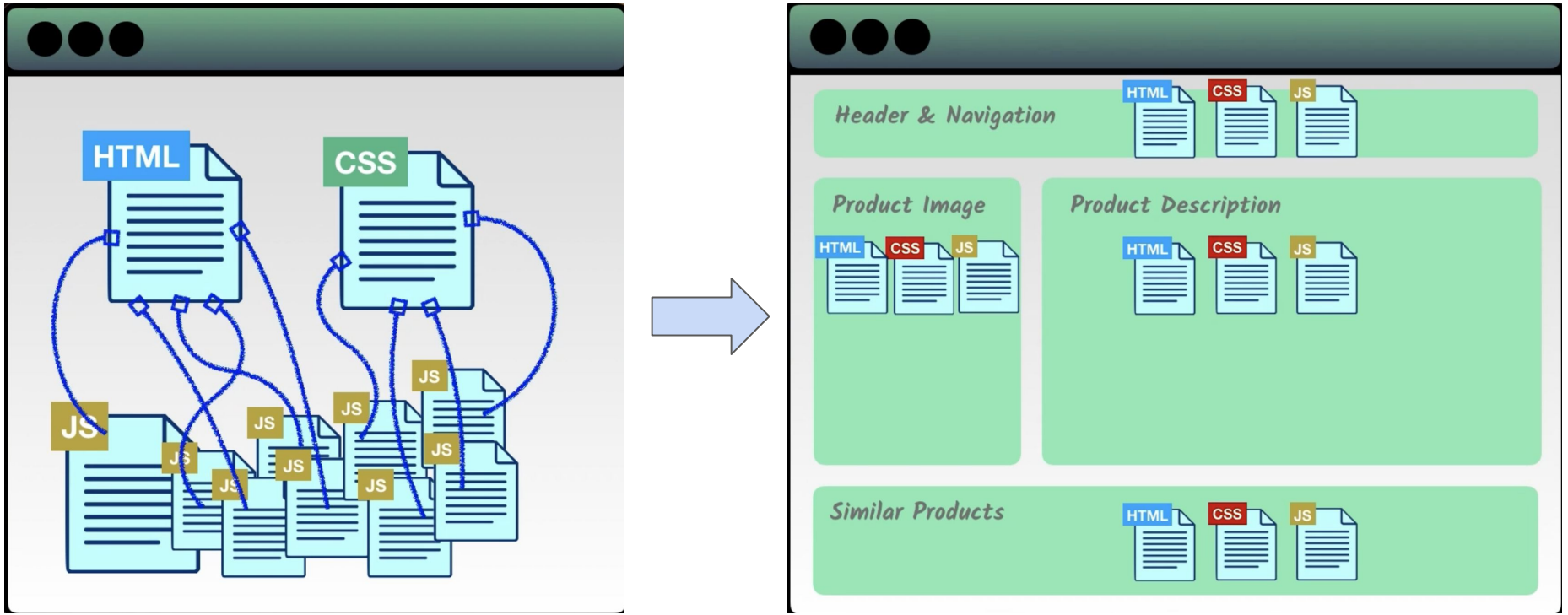
What is Vue

A JavaScript Framework for building more approachable, versatile, performant web pages

Makes things modular



Vue splits a webpage into reusable components.



Same code in plain JavaScript vs Vue.js

plain JavaScript

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Product App</title>
5  </head>
6
7  <body>
8      <div id = "app">
9          <h1>My product:</h1>
10         <h1 id = "product-info"></h1>
11     </div>
12 </body>
13
14 <script>
15 document.getElementById("product-info").
       innerHTML = "Boots";
16 </script>
17 </html>
```

Use element selector to update component

Same code in plain JavaScript vs Vue.js

Vue.js

Bind the view with the data, it'll display the value of the 'product' data property in the Vue instance —>

Create a Vue application —>

Defines the data object for the Vue application. It returns an object that contains the "product" property —>

Mount the Vue application to DOM —>

```
<!DOCTYPE html>
<html>
<head>
<title>Product App</title>
<script src="https://unpkg.com/vue@3"></script>
</head>

<body>
    <div id = "app">
        <h1>My Product:</h1>
        <h1>{{ product }}</h1>
    </div>
</body>

<script>
    const { createApp } = Vue;

    createApp({
        data(){
            return {
                product: 'Boots',
            }
        },
    }).mount('#app');
</script>
</html>
```

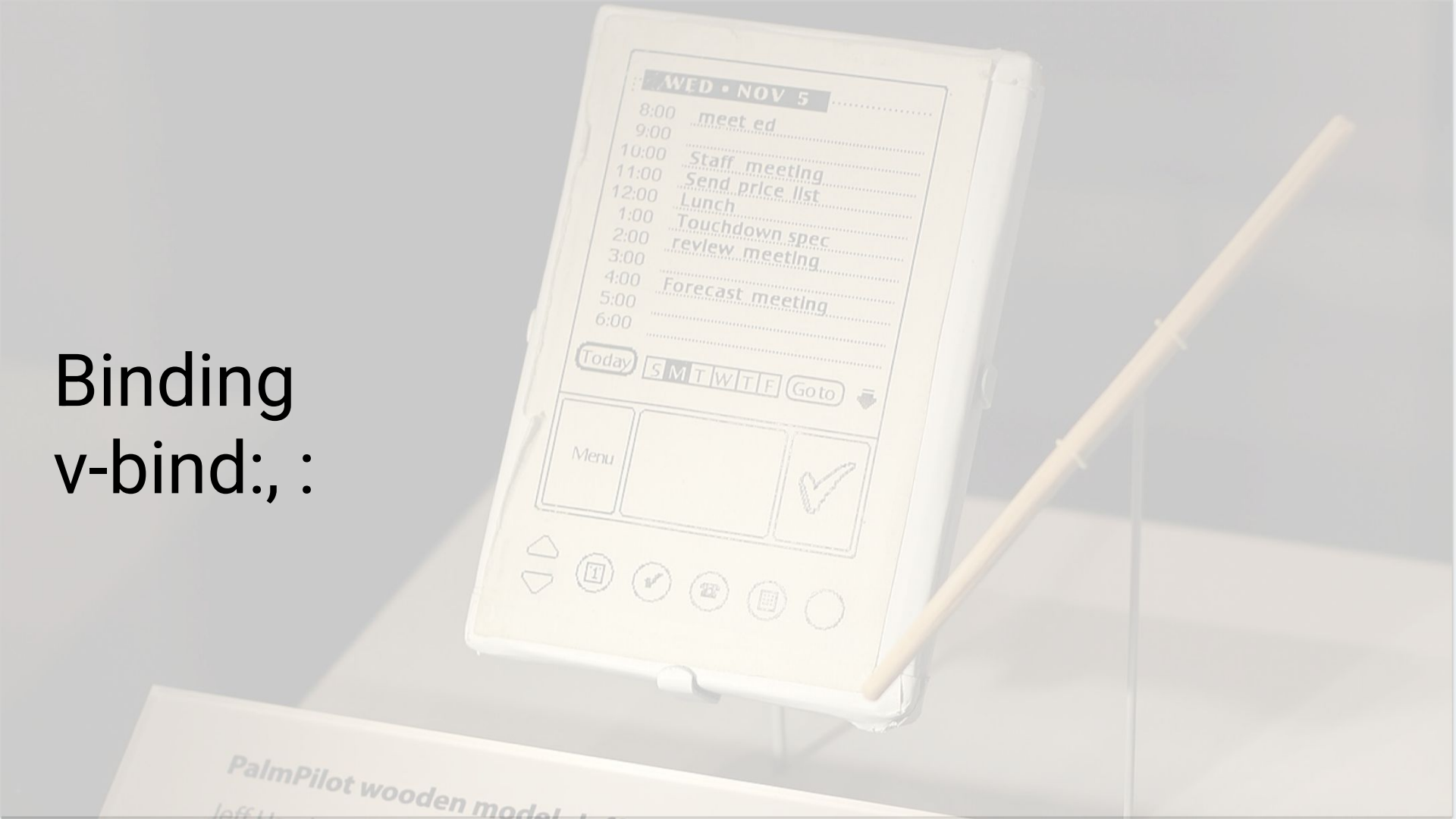
Vue.js - Directives

We discussed in the previous lecture

- v-bind
- v-if
- v-for
- v-on

Directives can be added to any DOM elements

Binding
v-bind:, :



Bind attribute to an HTML element

My product: Tshirt

Bind attribute to an HTML element

My product: Tshirt

```
10  <body>
11      <div id = "app">
12          <div class = "product">
13              <div class = "product-image">
14                  <img v-bind:src="image">
15              </div>
16              <div class = "product-info">
17                  <h1>My product: {{product}}</h1>
18              </div>
19          </div>
20      </div>
21  </body>
22
23  <script>
24      const { createApp } = Vue;
25
26      createApp({
27          data(){
28              return {
29                  product: 'Tshirt',
30                  image:"./assets/shirt-yellow.jpeg"
31              }
32          },
33      }).mount('#app');
34  </script>
```

A variety of attributes

Syntax:

v-bind: or :

Bind style, url, etc.

```
<body>
    <div id = "app">
        <div class = "product">
            <div class = "product-image">
                <img v-bind:src="image">
            </div>
            <div class = "product-info" :style="{color: textColor,
            fontSize: textSize}">
                <h1>My product: {{product}}</h1>
                <a :href="url">Visit UM</a>
            </div>
        </div>
    </div>
</body>
```

Review Syntax

- v-bind:href = “url” -> no curly braces, when the data is only a string
- v-bind:style = “{color:textColor}” -> {} defines an object, it’s more than a string
- {{product}} -> {{ }} display the value of the data property

Class Binding

Tshirt

Out of Stock

- 80% cotton
- 20% polyester
- Gender-neutral

Add to Cart

Cart:0

Class Binding

I define a class in my CSS code

```
.disabledButton {
  background-color: #d8d8d8;
}
```

When the condition (“inventory ==0”) is true, Vue will add the “disabledButton” class to this button.

Tshirt

Out of Stock

- 80% cotton
- 20% polyester
- Gender-neutral

Add to Cart

Cart:0

```
<button v-on:click="addToCart"
:class = "{disabledButton: inventory==0}">
Add to Cart</button>
```

Class Binding

Tshirt

Out of Stock

- 80% cotton
- 20% polyester
- Gender-neutral

Add to Cart

Cart:0

Tshirt

Almost sold out

- 80% cotton
- 20% polyester
- Gender-neutral

Add to Cart

Cart:0

Livecoding example

classbinding.html

Review Syntax

:class = "{disabledButton: inventory==0}"

- The :class directive dynamically binds CSS classes to the element.
- The value inside the curly braces {} is an object, where:
  - The key (disabledButton) is the CSS class name.
  - The value (inventory == 0) is a condition that determines whether the class is applied.

:class = "{disabledButton: inventory==0, lowstock: inventory<=5&inventory>0}"

- You can add multiple classes dynamically

Lecture 17-Survey 1

```
10  <body>
11      <div id = "app">
12          <div class = "product">
13              <div class = "product-image">
14                  <img v-bind:src="image">
15              </div>
16              <div class = "product-info">
17                  <h1>{{product}}</h1>
18                  <p v-if="inventory>10">In Stock</p>
19                  <p v-else-if="inventory<=10 && inventory>0">Almost sold out</p>
20                  <p v-else>Out of Stock</p>
21                  <ul>
22                      <li v-for ="detail in details">{{detail}}</li>
23                  </ul>
24
25                  <div class = "color-box"
26                  v-for="variant in variants"
27                  :key="variant.varitID"
28                  :style = "{ backgroundColor: variant.variantColor }"
29                  @mouseover="updateProduct(variant.variantImage)"
30                  >
31                  </div>
32
33                  <button v-on:click="addToCart"
34                  :class = "{disabledButton: inventory==0}">
35                  Add to Cart</button>
36
37                  <div class ="cart">
38                      <p>Cart:{{cart}}</p>
39                  </div>
40
41              </div>
42          </div>
43      </div>
44
45  </body>
```

In this piece of code using Vue, what are some examples of using binding? *

- [ ] L14: <img v-bind:src="image">
- [ ] L27: :key="variant.varitID"
- [ ] L28: :style = "{ backgroundColor: variant.variantColor }"
- [ ] L33: <button v-on:click="addToCart"
- [ ] L34: :class = "{disabledButton: inventory==0}"
- [ ] L38: Cart:{{cart}}

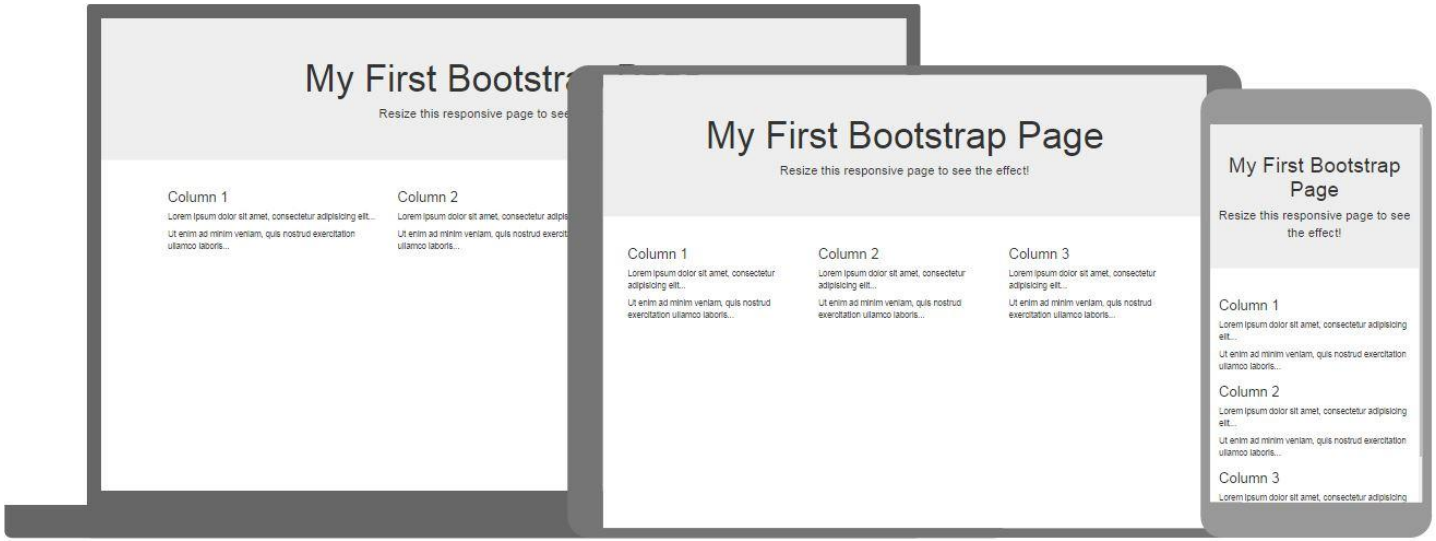
Bootstrap


WED • NOV 5
8:00 meet ed
9:00
10:00 Staff meeting
11:00 Send price list
12:00 Lunch
1:00 Touchdown spec
2:00 review meeting
3:00
4:00 Forecast meeting
5:00
6:00
Today
S M T W T F
Goto
Menu
PalmPilot wooden model

What is Bootstrap

Bootstrap is the most popular **CSS Framework** for developing **responsive** and mobile-first websites.
It includes a predefined CSS stylesheet for styling and layout.

Bootstrap for **responsiveness**

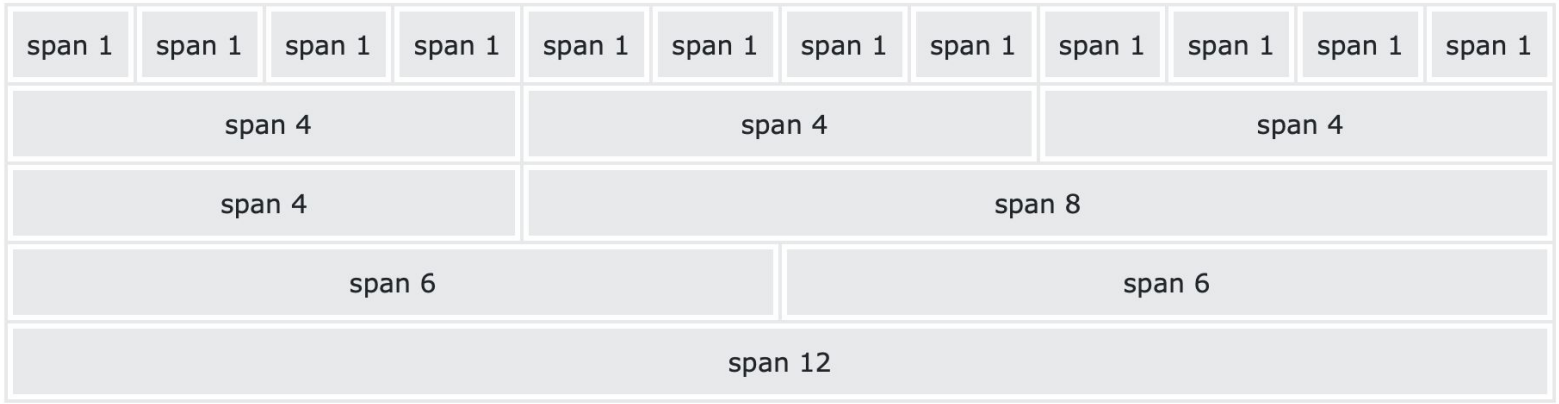


See Examples: https://www.w3schools.com/bootstrap5/index.php

The Core Thing: Bootstrap Grid System

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:



The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

Column examples: https://www.w3schools.com/bootstrap5/bootstrap_grid_basic.php

Livecoding example

bootstrap-basic.html

Review Syntax

Built-in utility classes (used in this example):

- container (with margin)
- container-fluid
- bg-primary, bg-success, bg-danger (set background color)
- mt-4 (margin top spacing size: larger)
- text-white
- p-3 (padding size)

More examples: Bootstrap Nav

Tabs

Active | Link | Link | Disabled

Turn the nav menu into navigation tabs with the `.nav-tabs` class. Add the `.active` class to the active/current link. If you want the tabs to be togglable, see the last example on this page.

Example

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

https://www.w3schools.com/bootstrap5/bootstrap_navs.php

Bootstrap Buttons

Bootstrap 5 Buttons

‹ Previous Next ›

Button Styles

Bootstrap 5 provides different styles of buttons:

Basic Primary Secondary Success Info Warning Danger Dark Light

Link

Example

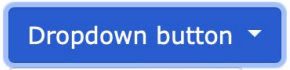
```
<button type="button" class="btn">Basic</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-link">Link</button>
```

https://www.w3schools.com/bootstrap5/bootstrap_buttons.php

Bootstrap Dropdown

Basic Dropdown

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list:



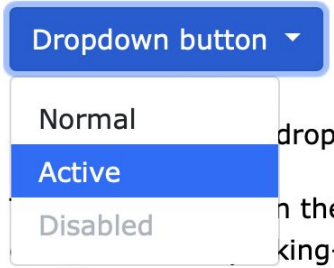
Link 1
Link 2
Link 3

```
<div class="dropdown">
  <button type="button" class="btn btn-primary dropdown-toggle" data-bs-
toggle="dropdown">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Link 1</a></li>
    <li><a class="dropdown-item" href="#">Link 2</a></li>
    <li><a class="dropdown-item" href="#">Link 3</a></li>
  </ul>
</div>
```

https://www.w3schools.com/bootstrap5/bootstrap_dropdowns.php

Bootstrap Disable and Active items

Disable and Active items



dropdown item with the `.active` class (adds a blue background color).

n the dropdown menu, use the `.disabled` class (gets a light-grey text
king-sign" icon on hover):

Example

```
<li><a class="dropdown-item" href="#">Normal</a></li>
<li><a class="dropdown-item active" href="#">Active</a></li>
<li><a class="dropdown-item disabled" href="#">Disabled</a></li>
```

https://www.w3schools.com/bootstrap5/bootstrap_dropdowns.php

Lecture 17-Survey 2

The Bootstrap grid system is based on how many columns? *

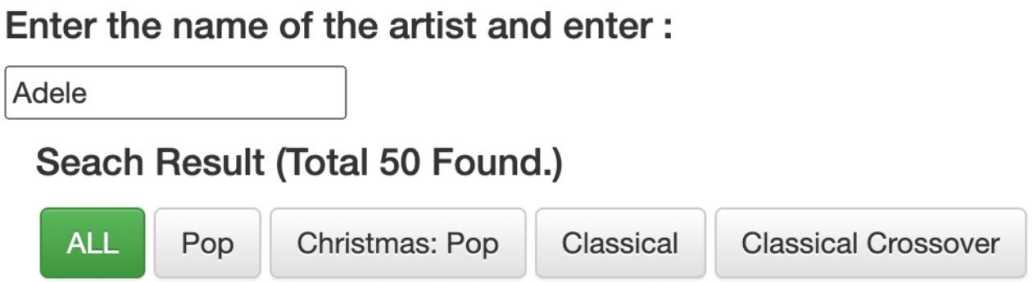
- 9
- 3
- 12
- 6

What is a primary advantage of using Bootstrap framework in web development? *

- It requires no CSS or JavaScript to function.
- It provides a responsive, mobile-first design out of the box.
- It is only compatible with static HTML pages.
- It eliminates the need for any custom styling.

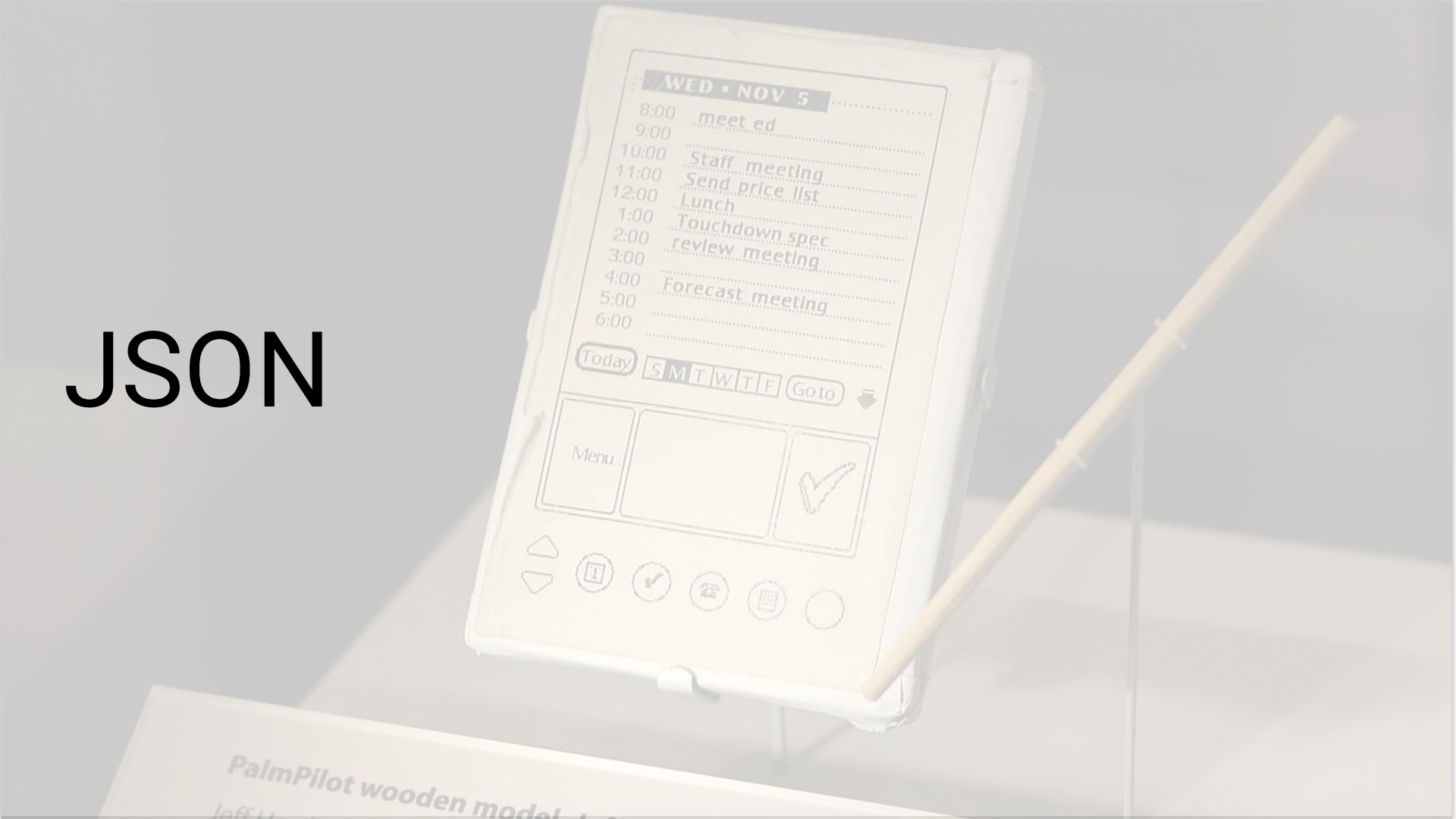
Lecture 17-Survey 2

If I want to create buttons with different colors using Bootstrap, what should I do? *



- ( ) Specify color using names, e.g., green
- ( ) Specify color using hex code, e.g., #00FF00
- ( ) Specify color using button class, e.g., btn-success
- ( ) Bootstrap is not the best for this

JSON


WED • NOV 5
8:00 meet ed
9:00
10:00 Staff meeting
11:00 Send price list
12:00 Lunch
1:00 Touchdown spec
2:00 review meeting
3:00
4:00 Forecast meeting
5:00
6:00
Today
S M T W T F
Goto
Menu
PalmPilot wooden model

JSON: Javascript Object Notation

- It is a string representation of an object that adheres to strict syntax rules.
- It is commonly used for data exchange between systems (e.g. between a server and a web application)

```
// JSON string
const jsonString = '{"name": "Alice", "age": 30, "isStudent": false,"hobbies":
  ["reading", "cycling", "hiking"],"address": {"street": "123 Main St","city":
  "Ann Arbor","state": "MI"}}';
```

JSON string

```
13  // JSON string
14  const jsonString = '{"name": "Alice", "age": 30,
      "isStudent": false,"hobbies": ["reading",
      "cycling", "hiking"],"address": {"street":
      "123 Main St","city": "Ann Arbor","state":
      "MI"}}';
15
```

Accessing JSON data, parsing data into JavaScript objects

```
28 // Convert JSON string to JavaScript object
29 const jsObject = JSON.parse(jsonString);
30
31 // Accessing properties of the JavaScript object
32 console.log(jsObject.name); // Output: Alice
33 console.log(jsObject.age); // Output: 30
34 console.log(jsObject.isStudent); // Output: false
35 console.log(jsObject.hobbies); // Output: [ 'reading',
   'cycling', 'hiking' ]
36 console.log(jsObject.address.city); // Output: Ann Arbor
37
```

Javascript to JSON

```
39  const jsObject = {
40    name: "Alice",
41    age: 30,
42    isStudent: false,
43    hobbies: ["reading", "cycling", "hiking"],
44    address: {
45      street: "123 Main St",
46      city: "Ann Arbor",
47      state: "MI"
48    }
49  };
```

Javascript to JSON

```
51  // Convert JavaScript object to JSON string
52  const jsonString = JSON.stringify(jsObject);
53
54  // Output the JSON string
55  console.log(jsonString);
56
```

Comparison

JSON	JavaScript Object
The key in key/value pair should be in double quotes.	The key in key/value pair can be without double quotes.
JSON cannot contain functions.	JavaScript objects can contain functions.
JSON can be created and used by other programming languages.	JavaScript objects can only be used in JavaScript.

JSON doesn’t know functions

- Your functions don’t get output in a JSON format.
  - It’s data only.

```
> let hourlyEmployee = {
    name: "Unassigned",
    hoursPerWeek : 40,
    computePay : function()
    {return this.hoursPerWeek *
this.hourlyRate;},
    reportPay : function()
    {console.log(this.name+" is paid
"+this.computePay());}
    }
<· undefined
> JSON.stringify(hourlyEmployee)
<· "{"name":"Unassigned","hoursPerWeek":4
0}"
```

JSON object references

- If your object references another object, JSON includes the referenced object

```
45  let place = {
46    name: "Beyster Building",
47    address: "2260 Hayward Street",
48    city: "Ann Arbor",
49    state: "Michigan"
50  };
51
52  let XuOffice ={
53    name:"Xu Wang",
54    building:place,
55    roomNumber:3737
56  };
```

```
> JSON.stringify(XuOffice)
< '{"name":"Xu Wang","building":{"name":"Beyster Buildi
  ng","address":"2260 Hayward Street","city":"Ann Arbo
  r","state":"Michigan"},"roomNumber":3737}'
```

Offices in Beyster

```
let place = {
  name: "Beyster Building",
  address: "2260 Hayward Street",
  city: "Ann Arbor",
  state: "Michigan"
};

let XuOffice ={
  name:"Xu Wang",
  building:place,
  roomNumber:3737
};

let MarkOffice={
  name:"Mark Ackerman",
  building;place,
  roomNumber:3912
};
```

JavaScript knows that:

```
> XuOffice.building
<· ▶ {name: 'Beyster Building', address: '2260 Hayward S
     treet', city: 'Ann Arbor', state: 'Michigan'}
> MarkOffice.building
<· ▶ {name: 'Beyster Building', address: '2260 Hayward S
     treet', city: 'Ann Arbor', state: 'Michigan'}
> XuOffice.building == MarkOffice.building
<· true
```

JSON object references

- JSON doesn't know they are the same object

```
> JSON.stringify(XuOffice)
<· '{"name":"Xu Wang","building":{"name":"Beyster Buildi
ng","address":"2260 Hayward Street","city":"Ann Arbo
r","state":"Michigan"},"roomNumber":3737}'
> JSON.stringify(MarkOffice)
<· '{"name":"Mark Ackerman","building":{"name":"Beyster
Building","address":"2260 Hayward Street","city":"Ann
Arbor","state":"Michigan"},"roomNumber":3912}'
> JSON.parse(JSON.stringify(XuOffice)).building
<· ▶ {name: 'Beyster Building', address: '2260 Hayward S
treet', city: 'Ann Arbor', state: 'Michigan'}
> JSON.parse(JSON.stringify(MarkOffice)).building
<· ▶ {name: 'Beyster Building', address: '2260 Hayward S
treet', city: 'Ann Arbor', state: 'Michigan'}
> JSON.parse(JSON.stringify(XuOffice)).building ==
JSON.parse(JSON.stringify(MarkOffice)).building
<· false
```

Ways around it

Give places unique IDs.

When you read in the JSON, match the object reference to the place IDs.

```
let place = {
  name: "Beyster Building",
  address: "2260 Hayward Street",
  city: "Ann Arbor",
  state: "Michigan",
  ID: 1234
};
```

```
> JSON.parse(JSON.stringify(XuOffice)).building
< {name: 'Beyster Building', address: '2260 Hayward S
  treet', city: 'Ann Arbor', state: 'Michigan', ID: 1
  234}
> JSON.parse(JSON.stringify(XuOffice)).building ==
  JSON.parse(JSON.stringify(MarkOffice)).building
< false
> JSON.parse(JSON.stringify(XuOffice)).building.ID ==
  JSON.parse(JSON.stringify(MarkOffice)).building.ID
< true
```

How can you get JavaScript object references right when generating JSON for your objects? *

- ◯ It's not a problem -- it just works.
- ◯ Don't use object references - reference unique IDs to look up.
- ◯ It's not a problem because JavaScript objects can't contain other object references.

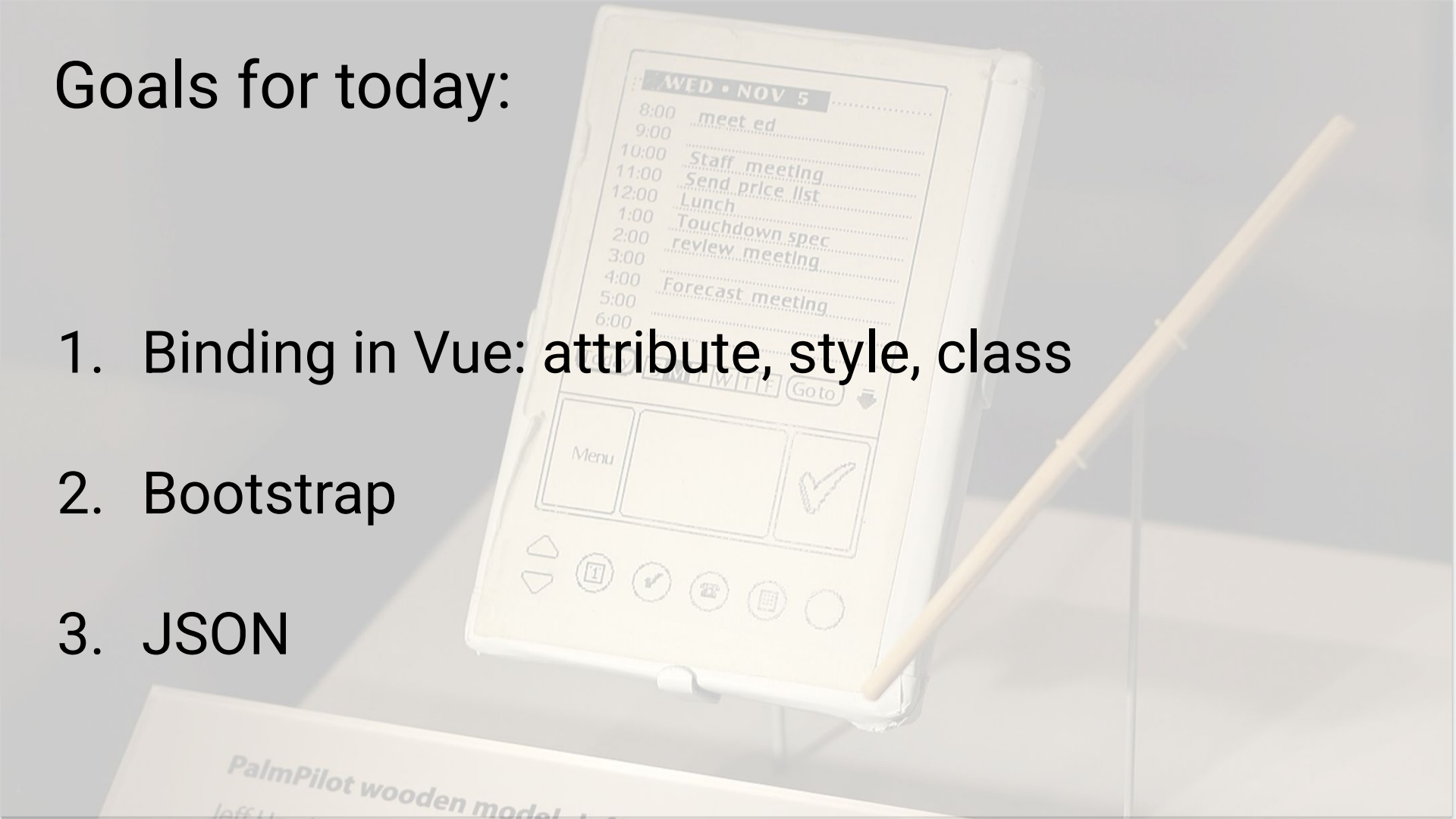
Which of the following is true? *

- JSON is no big deal -- most object-oriented languages have a way of typing in objects directly.
- JSON could be easily used to output C++ objects.
- JSON has an asynchronous version called JASON.
- JSON is unable to represent functions, dates, or shared object references.

Assume myCount is an object. What will be the result of evaluating: myCount === JSON.parse(JSON.stringify(myCount)) *

- ○ It's true, obviously.
- ○ It's false, because JSON.parse will capitalize the property names
- ○ It's false, because they become two different objects, even if values are the same
- ○ It will generate an error

Goals for today:



1. Binding in Vue: attribute, style, class
2. Bootstrap
3. JSON