

- Review from yesterday
- Simple Vue app
- JavaScript for Vue in more detail
- Static data (magic 8 ball)
- Whack a mole
- Address book challenge
- Start project 1
- MAKE SURE YOU HAVE A RIDE FOR WED LUNCH (TMR)

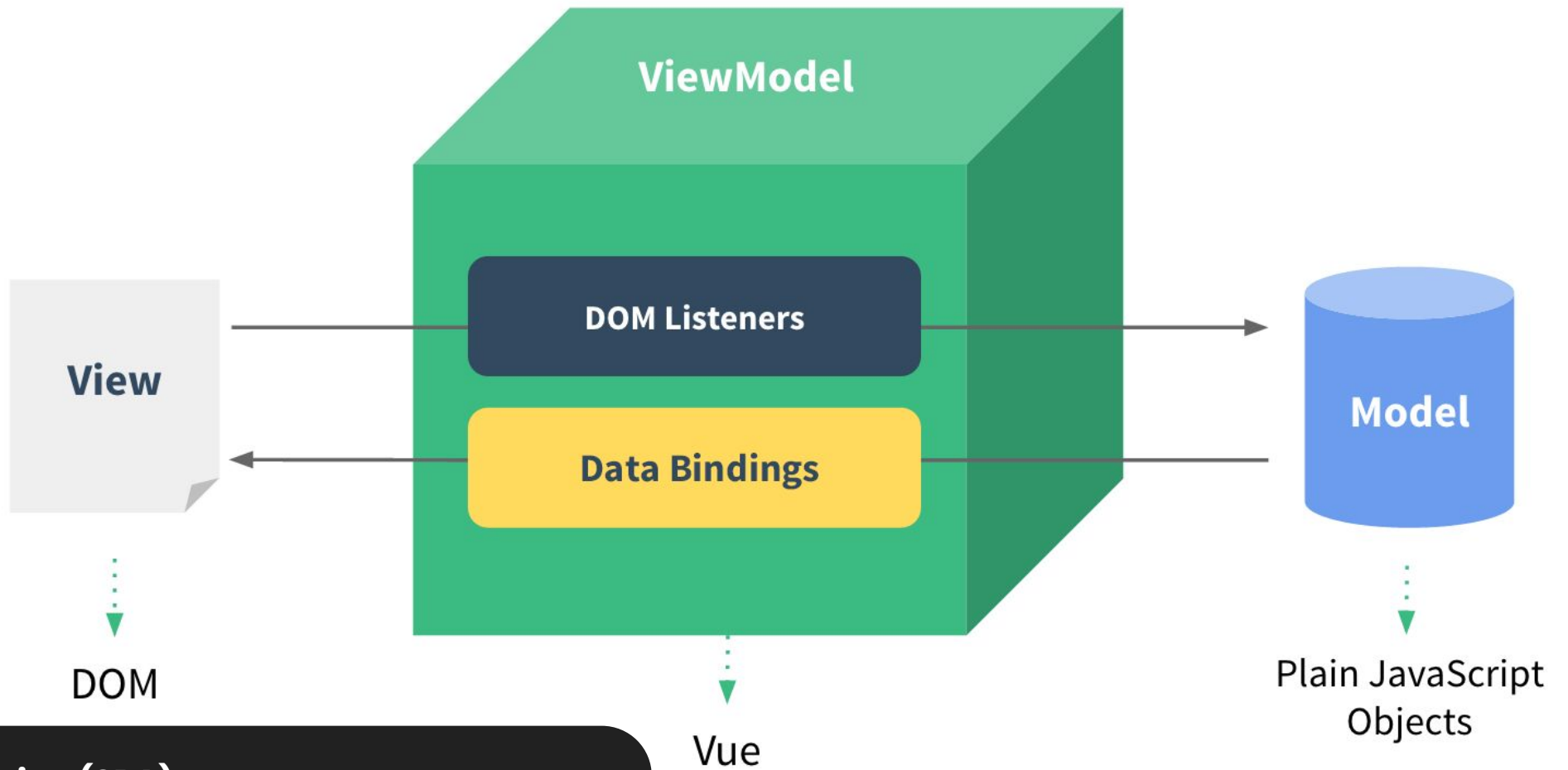
To-Do: 6/13



Vue: A JavaScript framework

- Reactive
 - Don't need to refresh page to see data changes
- Declarative
 - Can build page templates

No longer need to query the DOM to access objects



Single Page Application (SPA)

- Handle data updates and navigation without reloading the page
- Based in single index.html file

[Vue Reference](#)

Vue

Directives

Reference



v-if = "condition"

Render an element if condition is true

v-else

Render an element if *directly previous* condition is *not* true

v-show = "condition"

Display an element if condition is true

v-on:click = "foo()"

When you click this element, call the function 'foo.' Other options include: :mouseover, :keydown, etc.

v-cloak

Special directive that hides element until Vue has fully loaded. Add to css as well: [v-cloak] { display: none; }

v-for = "item in items"

Create an instance of an element for every item in items array

v-bind:style = "{ color: '#fff' }"

Change all text in this element white

v-model = "modelname"

Bind 'modelname' data value and input value together (2-way data binding)

Templating / Declarative Rendering

If we have a Vue data property:

```
hotel = {  
  rooms: 40,  
  booked: 25,  
  name: 'Red Cliffs'  
}
```

To display a value on the page:

```
<p>{{ hotel.name }}</p>
```



Template syntax

JavaScript Quick Review

Global: variables can be accessed anywhere in our script

Local: variables are specific to a section of code and can only be accessed in that section (block)

In Vue:

this.dataProperty = kinda like self.data_member in Python

- Need to use the 'this' keyword to access data properties in the Vue app

01

var

- Global scope when OUTSIDE a function
- Function/locally scoped when INSIDE function
- Can be redeclared and updated

```
var tester = "hey hi";

function newFunction() {
  var hello = "hello";
}

console.log(hello); // error: hello is not defined
```

02

let

- Scope is whatever block it's in (if statement, function, etc.)
- Can be updated but not redeclared

```
if (times > 3) {
  let hello = "say Hello instead";
  console.log(hello); // "say Hello instead"
}

console.log(hello) // hello is not defined
```

03

const

- Scope is whatever block it's in (like let)
- Can't be updated or redeclared

```
const greeting = "say Hi";
greeting = "say Hello instead"; // error: Assignment to constant variable.
```

JavaScript Objects

```
var hotel = {  
  rooms: 40,  
  booked: 25,  
  name: 'Red Cliffs'  
}
```

To access/update a **property**:

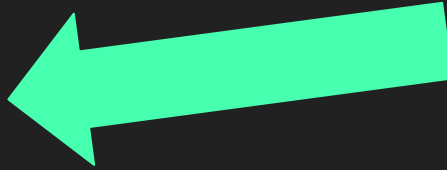
```
hotel.rooms = 50
```

Data properties can be any type

```
hotel.checkAvailability = function () { ...do something }
```


JavaScript Functions

```
function print() {  
  // can access 'this'  
  console.log("Hello world");  
}
```



OR

```
const print = () => {  
  // 'this' always refers to the outer global scope, so can't access Vue objects  
  console.log("Hello world");  
}
```

JavaScript Debugging

CONSOLE.LOG() PEOPLE!!!!

If something isn't doing what you expect, check the console before you do anything else

To make everything will be 50% easier:

- Learn to read the common error messages you get
- Log variables and calculations as you go along – piece by piece programming
 - Don't expect to code the entire thing or copy what I do and it just works

Resources

What it is	Where it's at
Vue Boilerplate	<u>GitHub</u>
JavaScript Courses	<u>freeCodeCamp</u>