

Vue Js notes

- Vue Js: framework of JS

Vue 2:

- Root of Vue app is vue instance(object)
- Whatever data we pass in main.js(file that has vue instance), that we can access in index.html using {{ }}
- {{ }} : called expression
- To access data in index.html, first step is to add vue in our project(using script tag) & then we need to connect index.html to main.js using an id='app'
- Whenever we change the data in main.js, the change gets reflected in index.html: so vue js is reactive
- We always display data from javascript to html using vue js

Steps to create vue 2 app:

1. Include vue in app (<script src="https://cdn.jsdelivr.net/npm/vue@2.5.13/dist/vue.js"></script>)
2. Make vue instance in main.js which also has an options object(optional properties) & assign the object to a variable app
3. Pass data in vue instance
4. Connect the main.js to index.html using id & el(element):'#app'
5. Display data in index.html using {{ }} : expression

Create app in Vue js: <https://codepen.io/aparnasoneja/pen/vYROXzX>

Attribute binding in Vue2:

- If we want to bind an attribute, i.e. display the attribute from JS to html : use v-bind
- V-bind:attribute_name="expression" – in html file
- V-bind is dynamically binding of attribute
- Eg: v-bind:src="image" in html file & in js file inside the vue instance: image="url of image"
- Shorthand of v-bind:src="image" is :src="image"
- Data binding- the host of the data is linked to the target of the data. In this case, our data is hosted by the data property of our Vue instance. And we want to target that data from our `src`.

Attribute binding Link: <https://codepen.io/aparnasoneja/pen/oNqXoyQ>

Conditional rendering in Vue2:

- V-if, v-else-if, v-else
- Pass data from js file & apply conditions in html file.

- V-show: it will only show visibility only if the condition is true, it does not insert or remove the element from the DOM. That means we need not make conditions for else if & else.

Conditional rendering: <https://codepen.io/aparnasoneja/pen/bGvdYze>

List rendering:

- We create a list/array in js file & display each element from that list using v-for
- Eg: to display an element detail from an array details: `<p v-for="detail from details">{{detail}}</p>`
- List rendering in case of multiple objects: : `<div v-for="detail from details" :key="detail.detailId"></div><p>{{detail.productName}}</p>`
- When using v-for it is recommended to give each rendered element its own unique key.

List rendering: <https://codepen.io/aparnasoneja/pen/OJvVzJN>

Event handling:

- For eg: we created a button in html file & want to increment the cart if the button is clicked, so we take the default value of cart in js file & add an onclick event in html file.
- In html: `<button v-on:click="cart+=1">Add to cart</button>` , In js: `cart:0` but this is not realistic, so let's define a method `addTocart()` in js & call that method in html on click.
- Whenever creating event handling method in js: always use `this`, this refers to the current Vue instance's data as well as other methods declared inside the instance
- Events: click, mouseover etc.
- Shorthand for v-on is `@`

Challenge of event handling: on hovering to color, the color of socks changes, create add to cart & remove from cart buttons

<https://codepen.io/aparnasoneja/pen/GRxJyNZ>

Class & style binding:

- For eg: if we now need that if the background color green is hovered, then green socks appear & if blue background color is hovered, blue socks appear, so for that we use style binding
`:style="{ backgroundColor: variant.variantColor }"`
`@mouseover="updateProduct(variant.variantImage"`
- Now we want that add to cart button should be disabled if product is not in stock & we have a class for disabled button in our css file that will show the button in white color if it's disabled. So, here we need to do class binding.
`<button v-on:click="addToCart"`

```
:disabled="!inStock"
:class="{ disabledButton: !inStock }"
>
Add to cart
</button>
```

- disabledButton class exists in css file
- You can also bind an entire class object or array of classes to an element

Challenge: When `inStock` is false, bind a class to the "Out of Stock" `p` tag that adds `text-decoration: line-through` to that element.

Solution: Give the value of `inStock` in Js file, style `text-decoration: line-through` in CSS file & bind that css class in html file.

<https://codepen.io/aparnasoneja/pen/abYOExo>

Computed properties:

These properties are only calculated & not saved. So, whenever value changes frequently, use computed properties as it's value is cached only for that time & after a new value comes, the already existing value gets out from the cache.

Eg: If u want to concatenate 2 values, use computed properties. If any of the value changes, the change is reflected back bcoz the older value is not stored.

Challenge: Add a new boolean data property `onSale` and create a computed property that takes `brand`, `product` and `onSale` and prints out a string whenever `onSale` is true.

Solution:

- JS file: initialise `onSale` property & create a computed property called `onSales()` which concatenates `brand` & `product` if the `onSale` is true. So, keep the if condition of `onSale` in html & display the value of `onSales` using `{{onSales}}`.
- <https://codepen.io/aparnasoneja/pen/KKopQzJ>

