

SIT120 Practical



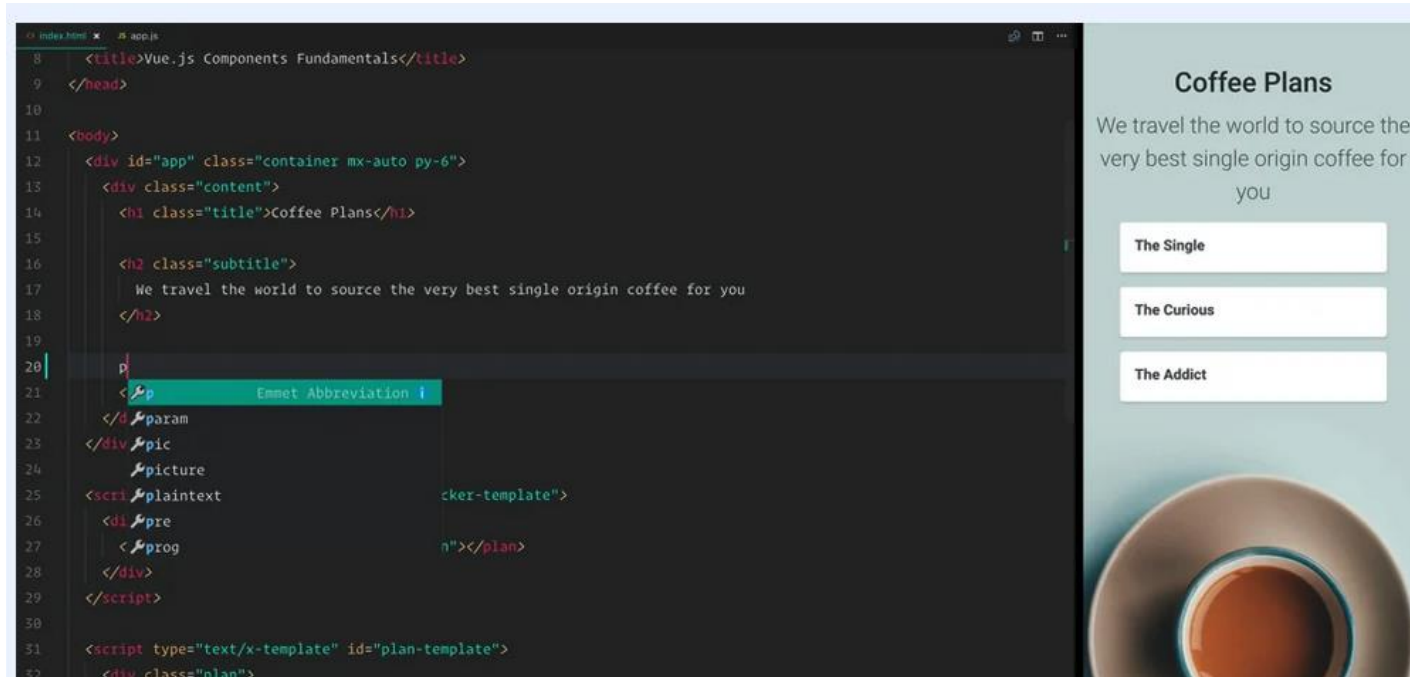
Week 7 – (Registration, Props, Custom Events, Dynamic/Async)

GOALS:

- Gain better understanding of Vue.js Framework
- How Component (Registration, Props, Custom Events, Dynamic/Async) works?
- Learn Coding of new Component concepts.

Watch Vue.js Components Video on Vue School

<https://vueschool.io/lessons/global-vs-local-components?friend=vuejs>



See boilerplate example: https://github.com/bencodezen/vue-enterprise-boilerplate/blob/main/src/components/_globals.js

Task 1: Local and Global Registration

(learn the codes with the help of your tutors and implement in your project)



<https://codepen.io/yuanyu/pen/eYOVeNr>

Vue Component Global Register

yuanyu + Follow

Hi , test

say Hi!

HTML (Pug)

```
1 #app
```

CSS

JS

```
1 Vue.component('prompt-compo', {
2   template: '<div><p>${message}</p><button @click="sayHi">say
   Hi!</button></div>',
3   delimiters: ['${', '}'],
4   data: function(){
5     return {
6       message: 'Hi , test'
7     }
8   },
9   methods: {
10    sayHi: function() {
11      alert('Hi!');
12    }
13  }
14 });
```

<https://codepen.io/yuanyu/pen/gOYvXMr>

Vue Component local Register

yuanyu + Follow

say hi test

say Hi!

HTML (Pug)

```
1 #app
```

CSS

JS

```
1 const promptComponent = {
2   template: '<div><p>${message}</p><button
   @click="sayHi">say Hi!</button></div>',
3   delimiters: ['${', '}'],
4   data: function() {
5     return {
6       message: 'say hi test'
7     }
8   },
9   methods: {
10    sayHi: function(){
11      alert('hi test');
12    }
13  }
14 };
15 let vm = new Vue({
16   el: '#app',
```

Apply and implement similar concepts in your project

Task 2: Coding Props

(learn the codes of Props with the help of your tutors and implement in your project)



<https://codepen.io/jravas/pen/GmNeXO?editors=1010>

A screenshot of a CodePen project titled "VUE component props" by a user named Josip. The project is displayed in a dark-themed editor. On the left, the preview area shows the text "hello mr. VUE". On the right, the code editor is divided into three panels: HTML, CSS, and JS. The HTML panel shows a single line of code: `<div id="app"><child :text="message"></child></div>`. The CSS panel is empty. The JS panel contains the following code:

```
1- Vue.component('child',{
2-   props: {
3-     text: {
4-       type: String,
5-       required: true
6-     }
7-   },
8-   template: `<div>{{ text }}</div>`
9- });
10
11- new Vue({
12-   el: '#app',
13-   data() {
14-     return {
15-       message: 'hello mr. VUE'
16-     }
17-   }
18- });
```

How can we pass the properties of an Object?

Apply and implement similar concepts in your project

Task 3: Custom Events

Visit and understand <https://codepen.io/team/Vue/pen/GRoPPrM>

Multiple v-models

Vue PRO + Follow

HTML

```
1 <div id="v-model-example" class="demo">
2   <p>First name: {{ firstName }}</p>
3   <p>Last name: {{ lastName }}</p>
4   <user-name
5     v-model:first-name="firstName"
6     v-model:last-name="lastName">
```

CSS

```
1 .demo {
2   font-family: sans-serif;
3   border: 1px solid #eee;
4   border-radius: 2px;
5   padding: 20px 30px;
6   margin-top: 1em;
```

JS

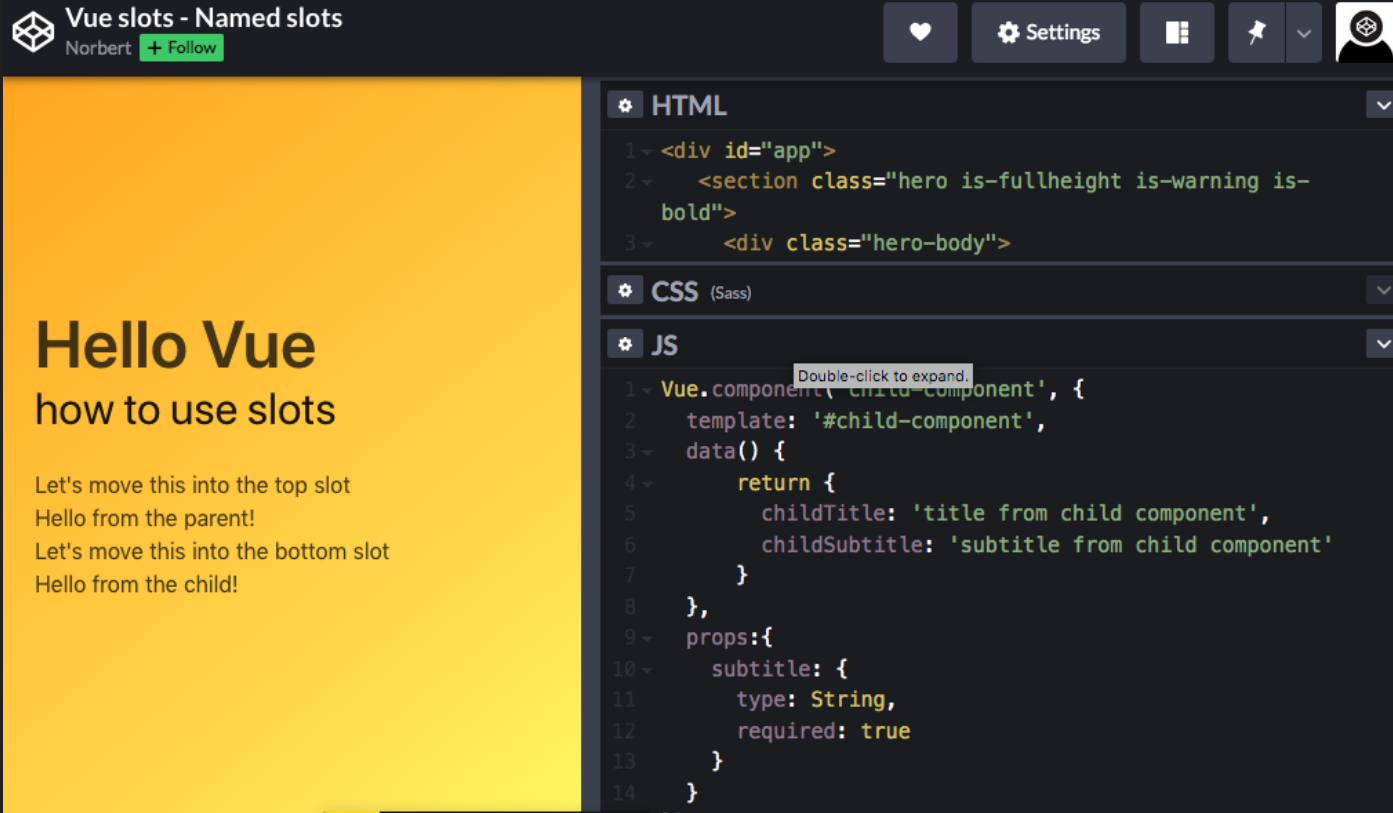
```
1 const UserName = {
2   props: {
3     firstName: String,
4     lastName: String
5   },
6   template: `
```

First name: John

Last name: Doe

Implement in your project.

Task 4: Vue Slots - coding Named Slots



Vue slots - Named slots
Norbert + Follow

HTML

```
1 <div id="app">
2   <section class="hero is-fullheight is-warning is-
  bold">
3     <div class="hero-body">
```

CSS (Sass)

JS

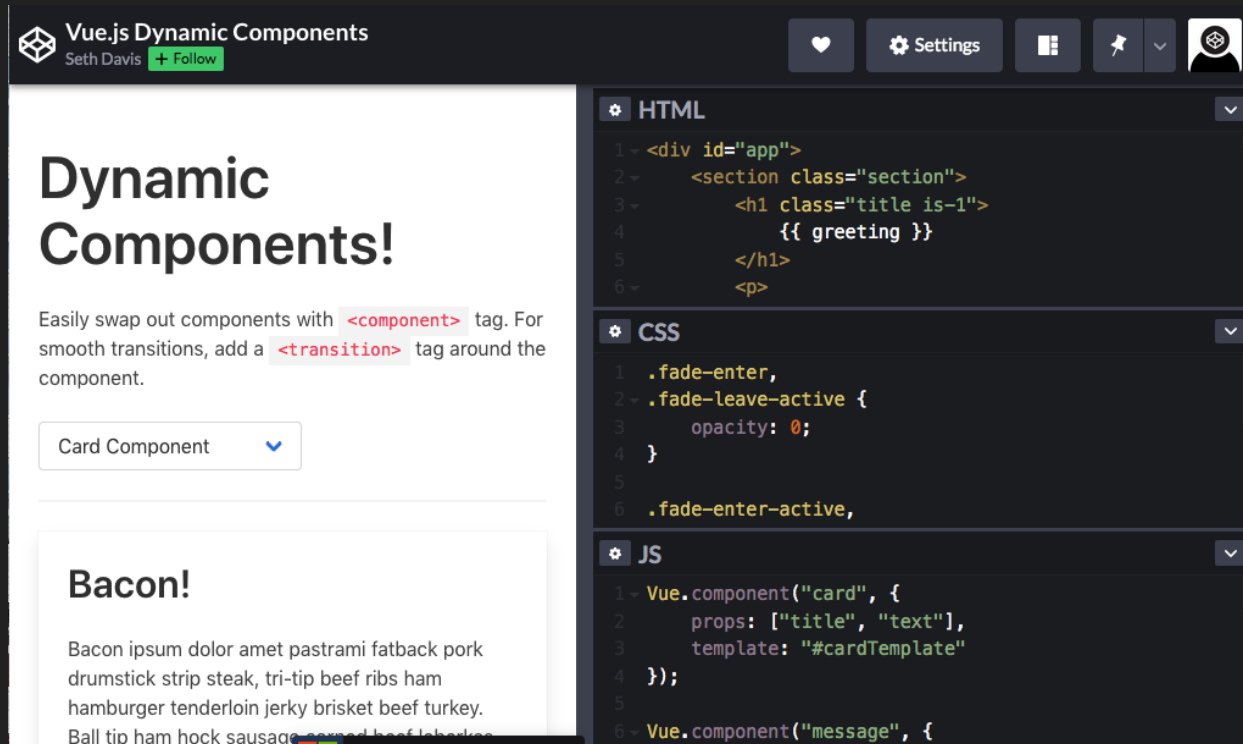
```
1 Vue.component('child-component', {
2   template: '#child-component',
3   data() {
4     return {
5       childTitle: 'title from child component',
6       childSubtitle: 'subtitle from child component'
7     }
8   },
9   props: {
10    subtitle: {
11      type: String,
12      required: true
13    }
14  }
```

<https://codepen.io/norbert989/pen/KRdopV?editors=1010>

1. Copy the code and execute in your editor
2. Implement similar concept in your project

Before Next Week: Dynamic and Edge cases

Visit code at <https://codepen.io/sethdavis512/pen/MvJapr>



1. Learn the code with your tutor and execute it in your editor

2. Implement Dynamic Components in your project

Discuss with your tutor:

How can we handle edge cases?

<https://vuejs.org/v2/guide/components-edge-cases.html>