Composition API

- The **Composition API** offers a new way to build Vue components in Vue. It *complements* the existing **Options API**.

- We define all our logic in a **setup** method or **<script setup>** section in the component file. The latter way is recommended by the Vue team.

Replacing Data

- The **setup** method/section does *not* have access to the **this** keyword.

- Use the **ref** function to define a piece of reactive state (data that will change over time). This is analogous to **data** in the **Options API**.

Replacing Methods

- Use JavaScript functions to operate on and change the reactive data. Remember to read/write the **value** property. These functions are analogous to **methods** in the **Options API**.

Replacing Computed Properties

- Use the **computed** function to re-run logic whenever a reactive piece of state changes. Pass a function as an argument.

- Vue will re-invoke the function whenever a piece of state referenced inside it updates. This is analogous to **computed** properties in the **Options API**.

Receiving Props in the Setup Method

- The **setup** method receives a reactive object of **props** as its first argument. You can utilize those props (for example, to compute CSS classes).

- The **props** object is reactive but its individual properties are not. Pass the **props** object to the **toRefs** function to return an object with all reactive properties.

The toRefs Function

- You can destructure properties from the object returned by **toRefs**. Remember that the properties are now individual reactive objects. Thus, you must access **.value** when using them in your JavaScript.

- When you use reactive objects in your HTML, Vue knows to extract their underlying value. Thus, no need to add **.value**.

More on <script setup>

- Use the **defineProps** function to define props validation when using **<script setup>**.
- The function will return the **props** object if you want to use it elsewhere in the section.
- We do not have to return an object with **<script setup>**. Our top-level names are available for use within the template.