Transition Classes

Step 1: jei element ta ke transition korte hobe shetake **</transition>** block e wrap up kore ekta **name** dite hobe,

```
<transition  name="fade">
  <div v-if="showP">
    Hello, ninja
  </div>
</transition>
```

Step 2: then name wise style add korte hobe,

```
<style>
  .fade-enter-from {}
  .fade-enter-to {}
  .fade-enter-active {}
  .fade-leave-from {}
  .fade-leave-to {}
  .fade-leave-active {}
</style>
```

For an example, a **</transition>** block with styles,

```
<style>
  .fade-enter-from {
    opacity: 0;
  }
  .fade-enter-to {
    opacity: 1;
  }
  .fade-enter-active {
    transition: all 2s ease;
  }
  .fade-leave-from {
    opacity: 1;
  }
  .fade-leave-to {
    opacity: 0;
  }
  .fade-leave-active {
    transition: all 2s ease;
  }
</style>
```

#Code Example

From **Home.vue**,

```
<transition name="toast">
  <Toast v-if="showToast"/>
</transition>
```

From **<style>**,

```
<style>
.toast-enter-from {
  opacity: 0;
  transform: translateY(-60px);
}

.toast-enter-to {
  opacity: 1;
  transform: translateY(0);
}

.toast-enter-active {
  transition: all 0.3s ease;
}

.toast-leave-from {
  opacity: 1;
  transform: translateY(0);
}

.toast-leave-to {
  opacity: 0;
  transform: translateY(-60px);
}

.toast-leave-active {
  transition: all 0.3s ease;
}

</style>
```

#Keyframes

Keyframes hocche ekta css property jeta animation er kon phase e ki hobe eta ke decide kore.

```
@keyframes wobble {
  0% {
    transform: translateY(-60px);
    opacity: 0;
  }

  50% {
    transform: translateY(0);
    opacity: 1;
  }

  60% {
    transform: translateX(8px);
  }

  70% {
    transform: translateX(-8px);
  }

  80% {
    transform: translateX(4px);
  }

  90% {
    transform: translateX(-4px);
  }

  100% {
    transform: translateX(0);
  }
}
```

Now using this animation,

```
.toast-enter-active {
  animation: wobble 0.5s ease;
}
```

#Transition Group

Transition group hocche ekadhik element er upor based kore kaj kore like list of items. Transition group er ekta **tag** thake. For an example,

```
<transition-group tag="ul" name="list">

  <li v-for="todo in todos" :key="todo.id" @click="deleteTodo(todo.id)">
    {{ todo.text }}
  </li>

</transition-group>
```

Style tag er style lekhar style same e,

```
.list-enter-from {
  opacity: 0;
  transform: scale(0.6);
}

.list-enter-to {
  opacity: 1;
  transform: scale(1);
}


.list-enter-active {
  transition: all 0.4s ease;
}
```

#Transition Initial Items

Add **appear** property with the transition or transition group, here is an example.

```
<transition-group tag="ul" name="list" appear>

  <li v-for="todo in todos" :key="todo.id" @click="deleteTodo(todo.id)">
    {{ todo.text }}
  </li>

</transition-group>
```

Another example with the **move** property, it triggers when transition is moved,

```
.list-move{
  transition: all 0.3s ease;
}
```

#Transitioning Between Elements

Jodi duita element thake like with conditions, tahole **transition** er shathe mode add kora jay like for an example, “out-in”

```
<transition name="switch" mode="out-in">

  <div v-if="todos.length">
    <transition-group tag="ul" name="list" appear>

      <li v-for="todo in todos" :key="todo.id" @click="deleteTodo(todo.id)">
        {{ todo.text }}
      </li>

    </transition-group>
  </div>
  <div v-else>Woohoo, nothing left todo!</div>

</transition>
```

Now from <style> code,

```
/* switch transition */

.switch-enter-from,
.switch-enter-to {
  opacity: 0;
  transform: translateY(20px);
}

.switch-enter-active, .switch-leave-active {
  transition: all 0.3s ease;
}
```

#Transitioning JS Hooks

```
<transition name="fade" appear
            @before-enter="beforeEnter"
            @enter="enter"
            @after-enter="afterEnter"

>
  <h1 v-if="showAbout">About</h1>
</transition>
```

From Script tag,

```
<script>

import {ref} from "vue";

export default {


  setup() {

    const showAbout = ref(true)

    const beforeEnter = () => {
      console.log('beforeEnter')
    }

    const enter = () => {
      console.log('enter')
    }

    const afterEnter = () => {
      console.log('afterEnter')
    }

    return {
      beforeEnter, enter, afterEnter, showAbout
    }
  }
}
</script>
```