

Introduction to Composition API

A new and different way to write your Vue apps.

A little about me

- Senior DX Engineer @ Netlify
- Vue.js Core Team
- Vue Mastery Instructor
- Google Developer Expert

What is the Composition API?

At its core, the Composition API is another way to build Vue.js apps and organize your code.

**At its core, the Composition API is
another way to build Vue.js apps
and organize your code.**

**The Composition API
does not replace the
Options API.**

Let's ReVue: The Options API



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

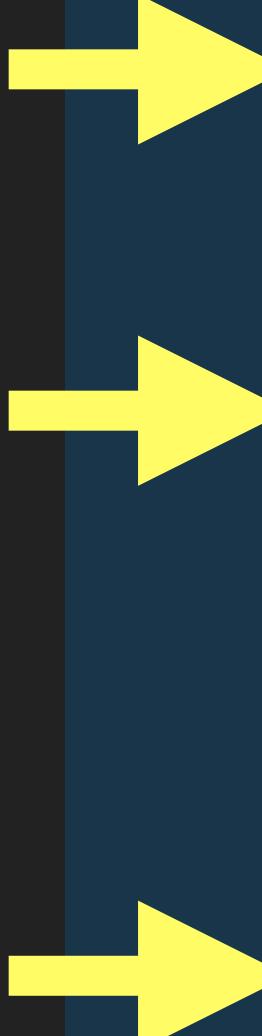
<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```





simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
import { computed, ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const doubleCount = computed(() => {
      return count.value * 2
    })

    const incrementCount = () => {
      count.value += 1
    }

    return {
      count,
      doubleCount,
      incrementCount
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
import { computed, ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const doubleCount = computed(() => {
      return count.value * 2
    })

    const incrementCount = () => {
      count.value += 1
    }

    return {
      count,
      doubleCount,
      incrementCount
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
import { computed, ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const doubleCount = computed(() => {
      return count.value * 2
    })

    const incrementCount = () => {
      count.value += 1
    }

    return {
      count,
      doubleCount,
      incrementCount
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
import { computed, ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const doubleCount = computed(() => {
      return count.value * 2
    })

    const incrementCount = () => {
      count.value += 1
    }

    return {
      count,
      doubleCount,
      incrementCount
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
import { computed, ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const doubleCount = computed(() => {
      return count.value * 2
    })

    const incrementCount = () => {
      count.value += 1
    }

    return {
      count,
      doubleCount,
      incrementCount
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



simple-option-api.vue

```
<script>
export default {
  data: () => ({
    count: 0
  }),
  computed: {
    doubleCount() {
      return this.count * 2
    }
  },
  methods: {
    incrementCount() {
      this.count++
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
from 'vue'

computed(() => {
  * 2
  = () => {
```

```
<template>
  <p>{{ count }}</p>
</template>
```



simple-composition-api.vue

```
<script>
export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, context) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, ctx) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, { attrs, slots, emit }) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, ctx) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, ctx) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, ctx) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```



simple-composition-api.vue

```
<script>
export default {
  setup(props, ctx) {
    const count = 0

    return {
      count
    }
  }
}
</script>
```

Reactive Data in Composition API



reactive-in-composition-api.vue

```
<script>
export default {
  data() {
    return {
      count: 0
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```

ref



reactive-in-composition-api.vue

```
<script>
export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    ★ const count = ref(0)

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      count.value++
    }

    return {
      count,
      incrementCount
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      this.count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      this count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
      count += 1
    }

    return {
      count
    }
  }
}
</script>

<template>
<p>{{ count }}</p>
```



reactive-in-composition-api.vue

```
<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)

    const incrementCount = () => {
       count.value += 1
    }

    return {
      count
    }
  }
}

</script>

<template>
  <p>{{ count }}</p>
</template>
```

reactive



reactive-in-composition-api.vue

```
<script>
export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const count = 0

    return {
      count
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const count = 0

    const state = reactive({})

    return {
      count
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const count = 0

    const state = reactive({})

    return {
      count,
      state
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const count = 0

    const state = reactive({})

    return {
      count,
      state
    }
  }
}
</script>

<template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      count,
      state
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      state
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      state
    }
  }
}
</script>

<template>
  <p>{{ state.count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const incrementCount = () => {
      state.value += 1
    }

    return {
      state
    }
  }
}
</script>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const incrementCount = () => {
      count += 1
    }

    return {
      state
    }
  }
}
</script>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const incrementCount = () => {
      state.count += 1
    }

    return {
      state
    }
  }
}
</script>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      state
    }
  }
}
</script>
```

```
<template>
  <p>{{ state.count }}</p>
</template>
```

toRefs



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      state
    }
  }
}
</script>
```

```
<template>
  <p>{{ state.count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...state
    }
  }
}
</script>

<template>
  <p>{{ state.count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...state
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...state
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...state
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...state
    }
  }
}
</script>

<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...toRefs(state)
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```



reactive-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...toRefs(state) ★
    }
  }
}
</script>
```

```
<template>
  <p>{{ count }}</p>
</template>
```

Computed Properties



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      ...toRefs(state)
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state)
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```





computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2
     const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = state.count * 2

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = computed(() => {
      return state.count * 2
    })

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

     const doubleCount = computed(() => {
      return state.count * 2
    })

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const doubleCount = computed(() => {
      return state.count * 2
    })

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0,
      doubleCount: computed(() => {
        return state.count * 2
      })
    })

    return {
      ...toRefs(state),
      doubleCount
    }
  }
}
</script>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0,
      doubleCount: computed(() => {
        return state.count * 2
      })
    })

    return {
      ...toRefs(state)
    }
  }
}
</script>

<template>
```



computed-in-composition-api.vue

```
<script>
import { computed, reactive, toRefs } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0,
      doubleCount: computed(() => {
        ★ return state.count * 2
      })
    })

    return {
      ...toRefs(state)
    }
  }
}
</script>

<template>
```

Methods



methods-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    return {
      state
    }
  }
}
</script>
```



methods-in-composition-api.vue

```
<script>
import { reactive } from 'vue'

export default {
  setup() {
    const state = reactive({
      count: 0
    })

    const incrementCount = () => {
      state.count += 1
    }

    return {
      state
    }
  }
}
</script>
```

```
setup() {
  const state = reactive({
    count: 0
  })

  const incrementCount = () => {
    state.count += 1
  }

  return {
    state
  }
}

</script>

<template>
  <p>{{ state.count }}</p>
</template>
```

```
setup() {
  const state = reactive({
    count: 0
  })

  const incrementCount = () => {
    state.count += 1
  }

  return {
    state,
    incrementCount
  }
}

</script>

<template>
  <p>{{ state.count }}</p>
</template>
```

```
setup() {
  const state = reactive({
    count: 0
  })

  const incrementCount = () => {
    state.count += 1
  }

  return {
    state,
    incrementCount
  }
}

</script>

<template>
  <button @click="incrementCount">
    Increment Count: {{ count }}
  </button>
</template>
```

```
setup() {
  const state = reactive({
    count: 0
  })

  const incrementCount = () => {
    state.count += 1
  }

  return {
    state,
    incrementCount
  }
}

</script>

<template>
  <button @click="incrementCount">
    Increment Count: {{ count }}
  </button>
</template>
```

That's the basics!



**YOU OTTER BE PROUD
OF YOURSELF!**

Is there more?

- Lifecycle Hooks
- Watch
- Modules
- And many more APIs...

Next Steps

Trade-offs

- Can be more verbose and have less structure compared to Options API
- More JavaScript knowledge is needed
- Deeper understanding of Vue.js is required to use it
- Architectural decisions are up to you

**You may be tempted to rewrite
your entire app in Composition API.**

You may be tempted to rewrite
your entire app in Composition API.

Please don't.

**The Composition API is a powerful technique
to understand and have in your skillset, but
remember that at the end of the day.**

**The Composition API is a powerful technique
to understand and have in your skillset, but
remember that at the end of the day.**

**All code is
compromise.**

**“But I want more
Composition API!”**

Jamstack Explorers

Learning Platform

Greetings, fellow explorer! Welcome to your mission control – a place to learn about building for the web with modern tools and techniques.

As a Jamstack Explorer, you can chart a course through new frameworks, through bright new tools and APIs, through [Netlify's](#) features and workflows, and through the very fabric of HTML, CSS, and JavaScript itself.

Choose your mission. Track your progress. Earn rewards. And join us as we explore the Jamstack together!



This Mission: Launching with Composition API



Instructor: **Ben Hong**



Learn how to use the new Vue.js 3 Composition API works!

WHAT YOU'LL LEARN

- ◆ Introduction
- ◆ Project Setup
- ◆ What is Composition API?
- ◆ Reactive Data
- ◆ Computed Properties
- ◆ Methods
- ◆ Lifecycle Hooks
- ◆ Watch
- ◆ Next Steps

Prepare to embark on a journey into the new Vue.js 3 Composition API!

(∩ ` -`)—☆° .*•°

In this mission, we'll be:

The ultimate learning resource for Vue developers

Weekly Vue.js tutorials to guide your journey to Mastery.

[Explore courses](#)

Thank you!

Find me at [@bencodezen](#)

- [GitHub.com/bencodezen](#)
- [Twitter.com/bencodezen](#)
- [YouTube.com/bencodezen](#)

