Vue 3.4

Slam Dunk ————

44%
Improvement

in Vue SFC Compilation
Performance

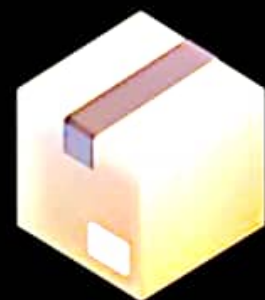## A totally Rewritten Parser that is

⚡ # 2x
Faster

⚡

100% Backwards Compatible!

— ✨ **Faster builds**

for people using Vue SFC's

defineModel() is now Stable

a new `<script setup>` macro that aims to simplify the implementation of components that support v-model.

v-bind

Same-name Shorthand

🔥 **One of the most requested features**

```
<img :src :alt />
```

🤨 **QUICK EXPLANATION**

Before

```
<script setup lang="ts">
const src = '';
const alt = '';
</script>


<template>
  <img :src="src" :alt="alt" />
</template>
```

We had to explicitly pass the prop

After

```
<script setup lang="ts">
const src = '';
const alt = '';
</script>


<template>
  <img :src :alt />
</template>
```

✅ Now the template can infer the prop from the script if they have the same name.

More Efficient Reactivity System

```vue
<script setup>
const count = ref(1);


const isEven = computed(() ⇒ count.value % 2 == 0); // false


watchEffect(() ⇒ console.log(isEven.value));
</script>



<template>
  <button @click="count += 2">
    {{ count }}
  </button>
</template>
```

Before pre 3.4 😬 | Used to run every time count changed, even if the computed result remains the same.

😎 **Now** post 3.4

Now the callback only fires if the computed result has actually changed.

— — — — — — — — — — —

🟡 **EXPLANATION**

The initial value of isEven is false, if we add 2 to count, it'll be 3, meaning it still won't be even, resulting in the same false value, and because it still is the same value, watchEffect() won't run. It'll only run if we add 1 to count making it even, resulting in a different computed value.