

Vue script setup

Vue 版本 3.2.0 以后引入了 `<script setup>`。

- 顶级变量可以直接用于 `template` 模版使用。
- 在 `script` 中要加上 `setup` 属性。
- 没有输出、返回。
- 可以定义原生的 JS, 和 `<script setup>` 区分开

```
<script>
  function add(a, b) {
    return a + b;
  }
</script>

<script setup>
  import { ref } from "vue";
  const count = ref(100);

  const a = 10;
  const b = 20;
  add(a, b);
</script>

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

属性和事件

子组件中定义属性通过 `defineProps`, 定义事件通过 `defineEmits`。

```
<!-- child.vue -->
<script setup>
  import { defineProps, defineEmits } from "vue";
  const props = defineProps({ name: String, age: Number });

  const emits = defineEmits(['confirm', 'cancel']);

  function onConfirm() {
    emits('confirm');
  }
</script>

<template>
  <div>{{ props.name }}</div>
  <button @click="onConfirm">确认</button>
</template>
```

```
<!-- parent.vue -->
<script setup>
  import { ref } from "vue";
  import child from "./child.vue";

  const name = ref('Zhang San');

  function onConfirm() {
    console.log("子组件传来的事件");
  }
</script>

<template>
  <child :name="name" @confirm="onConfirm"></child>
</template>
```

defineExpose 暴露子组件信息

```
<!-- child.vue -->
<script setup>
  import { ref } from "vue";
  const count = ref(100); // 响应式变量
  const age = 'Zhang San'; // 普通变量

  defineExpose({
    a,
    b
  });
</script>
```

```
<!-- parent.vue -->
<script setup>
  import { ref } from "vue";
  import child from "./child.vue";

  const expose = ref(null);

  onMounted(() => {
    console.log("expose.a", expose.value.a);
    console.log("expose.b", expose.value.b);
  });
</script>

<template>
  <div ref="expose"></div>
</template>
```

