

父组件传值给子组件

props

案例一

父组件代码

```
<script setup>
import Header from './components/header.vue'
</script>
```

```
<template>
  <Header title="头部" age="18"/>
</template>
```

子组件代码:header.vue

```
<script setup>
  const props = defineProps(["title", "age"]);
  console.log(props.age, props.title);
</script>
```

案例二

父组件代码

```
<!-- :params='一个变量也可' -->
<Header :params="{name: '石头人', age: 15}, {name: '诺克萨斯之手', age: 18}"/>
```

子组件代码

```
<script setup>
  const props = defineProps(["params"]);
  props.params.forEach(item => {
    console.log(item.name, item.age);
  })
</script>
```

插槽传值

props 是父组件传递一些简单的变量数据给子组件，假设我们要传递比较复杂的数据呢？比如传递一些 HTML 代码，这就需要用到插槽了。

案例一

父组件,同时传递props和模板数据

```
<Header :params="{name:'石头人',age:15},{name:'诺克萨斯之手',age:18}">
  <h1>插槽HTML内容</h1>
</Header>
```

父组件中，插槽中可以是任意HTML，也可以包含其他组件

子组件

`<slot>` 元素是一个**插槽出口** (slot outlet)，标示了父元素提供的**插槽内容** (slot content) 将在哪里被渲染。

```
<template>
  <slot></slot>
</template>
```

渲染作用域

插槽内容**无法访问**子组件的数据。Vue 模板中的表达式只能访问其定义时所处的作用域，这和 JavaScript 的词法作用域规则是一致的。换言之：

父组件模板中的表达式只能访问父组件的作用域；子组件模板中的表达式只能访问子组件的作用域。

具名插槽

父组件

#a,#b,#c就是给插槽定义名字

```
<Header :params="[ {name:'石头人',age:15}, {name:'诺克萨斯之手',age:18} ]">
  <h1>插槽内容</h1>
  <template #a>
<h1>插槽内容1</h1>
  </template>
  <template #b>
<h1>插槽内容2</h1>
  </template>
</Header>
```

子组件

```
<slot name="b"></slot>
```

子组件传值给父组件

子组件

```
<script setup>
import { ref,defineEmits } from 'vue'
// 定义要发送的emit事件
const emit = defineEmits(['updateValue'])
let selectedValue = ref("青铜");
const change = () => {
  emit("updateValue", selectedValue);
}
</script>
<template>
  <select @change="change" v-model="selectedValue">
    <option value="青铜">青铜</option>
    <option value="白银">白银</option>
    <option value="黄金">黄金</option>
    <option value="铂金">铂金</option>
  </select>
```

```
</template>
```

父组件

```
<script setup>
  import { ref } from 'vue'
  import HelloWorld from './components/HelloWorld.vue'
  const valueFromChild = ref(0);

  const handleValueFromchild = (value) => {
    valueFromChild.value = value;
  }
</script>

<template>
  <HelloWorld @updateValue="handleValueFromchild"/>
  <h1>Received Value from Child: {{ valueFromChild }}</h1>
</template>
```

父组件中@updateValue 的名字必须和子组件中defineEmits定义的名字一致。