后盾网 人人做后盾

www.houdunwang.com

组件开发

后盾网 2011-2017 v2.0

组件（Component）是 Vue.js 最强大的功能之一。组件可以扩展 HTML 元素，封装可重用的代码。

要确保在初始化根实例之前注册了组件：

Vue.js

组件必须在根实例前创建

```
<div id="app">
    <hd-user></hd-user>
</div>
<script>
    //注册组件
    Vue.component('hdUser', {
        template: '<h1>你好后盾</h1>'
    })
    //创建根实例
    new Vue({
        el: '#app',
    })
</script
```

template属性指组件使用的模板内容即视图。

定义组件

不必在全局注册每个组件。通过使用组件实例选项注册，可以使组件仅在另一个实例/组件的作用域中可用

```
<div id="app">
    <hd-user></hd-user>
</div>
<script>
    //创建根实例
    new Vue({
        el: '#app',
        components: {
            hdUser: {
                template: "<h2>你好 houdunren.com</h2>"
            }
        }
    })
</script>
```

局部注册组件

设置is属性的值为组件名称同样也可调用执行这个组件。

```
<div id="app">
    <h2 is="hd-user"></h2>
</div>
<script>
    //创建根实例
    new Vue({
        el: '#app',
        components: {
            hdUser: {
                template: "<h2>你好 houdunren.com</h2>"
            }
        }
    })
</script>
```

使用is属性执行组件

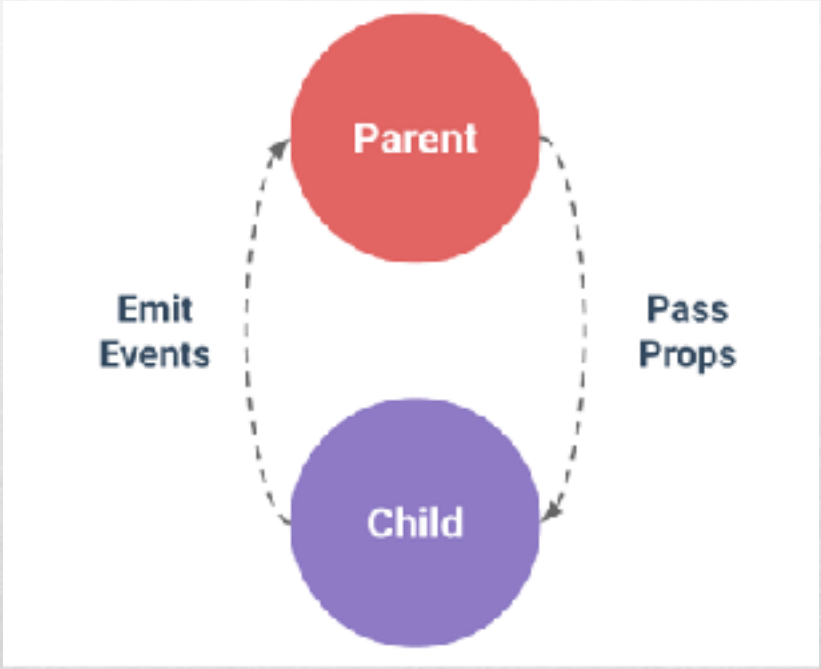
组件的data属性是为组件配置显示的数据使用的。
组件的data属性值必须是函数，这是因为组件是复用的，为了让每个组件使用独立的数据，我们要在函数体中返回对象。

```
<div id="app">
    <hd-php></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        template: "<h2>你好 {{title}}</h2>",
        data: function () {
            return {
                title: '向军老师'
            }
        }
    })
    //创建根实例
    new Vue({
        el: '#app'
    })
</script>
```

组件data属性

组件不是孤立的组件间需要数据传递，这样的组件才更好用。

在 Vue.js 中，父子组件的关系可以总结为 props down, events up 。父组件通过 props 向下传递数据给子组件，子组件通过 events 给父组件发送消息。看看它们是怎么工作的



父子组件

组件实例的作用域是孤立的。这意味着不能(也不应该)在子组件的模板内直接引用父组件的数据。要让子组件使用父组件的数据，我们需要通过子组件的props选项。

子组件要显式地用 props 选项声明它期待获得的数据,就像 data 一样，prop 可以用在模板内。同样也可以在 vm 实例中像 “this.message” 这样使用。经过以下两步我们就可以向组件传递字符数据了。

1. 首先定义props
2. 为组件的标签设置属性值

```
<div id="app">
    <hd-php message="你好后盾网"></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        props: ['message'],
        template: "<h2>{{message}}</h2>"
    })
    //创建根实例
    new Vue({
        el: '#app'
    })
</script>
```

向组件中传递字符串数据

在模板中，要动态地绑定父组件的数据到子模板的props，与绑定到任何普通的HTML特性相类似，就是用 v-bind。每当父组件的数据变化时，该变化也会传导给子组件。

下面的示例hd-php组件的title变量就可以使用父级的content的值了。

```
<div id="app">
    <input type="text" v-model="content">
    <hd-php :title="content"></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        props: ['title'],
        template: "<h2>{{title}}</h2>"
    })
    new Vue({
        el: '#app',
        data:{
            content:'你好后盾'
        }
    })
</script>
```

向组件中动态传递数据

prop 是单向绑定的：当父组件的属性变化时，将传导给子组件，但是不会反过来。这是为了防止子组件无意修改了父组件的状态。

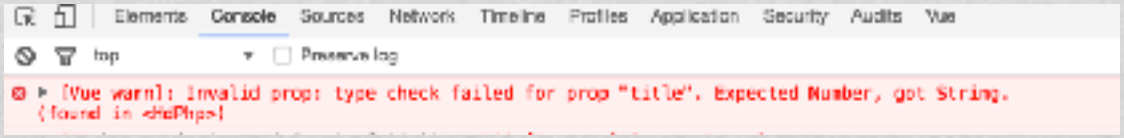
```
<div id="app">
    <input type="text" v-model="content">
    <hd-php :title="content"></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        props: ['title'],
        template: "<h2>{{mess}}</h2>",
        data: function () {
            return {mess: this.title};
        }
    })
    new Vue({
        el: '#app',
        data: {
            content: '你好后盾'
        }
    })
</script>
```

组件获取父级数据并断开联系

我们可以对传递给组件的数据进行验证，如果传入的数据不符合规格，Vue 会发出警告。当组件给其他人使用时这很有用。

```
<div id="app">
    <hd-php :title="1"></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        props: {
            title: {
                type: Number
            }
        },
        template: "<h2>{{title}}</h2>"
    })
    new Vue({
        el: '#app'
    })
</script>
```

如果设置属性为 <hd-php :title="'hdphp'"></hd-php>时在控制台会显示错误



prop验证

如果没有为prop传递数据时，我们可以通过设置默认值让他有数据展示。

```
<div id="app">
    <hd-php></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        props: {
            title: {
                type: String,
                default: '这是默认值'
            }
        },
        template: "<h2>{{title}}</h2>"
    })
    new Vue({
        el: '#app',
        data: {
            content: ''
        }
    })
</script>
```

prop设置默认值

```
Vue.component('example', {
    props: {
        // 基础类型检测 （`null` 意思是任何类型都可以）
        propA: Number,
        // 多种类型
        propB: [String, Number],
        // 必传且是字符串
        propC: {
            type: String,
            required: true
        },
        // 数字，有默认值
        propD: {
            type: Number,
            default: 100
        },
        // 数组 / 对象的默认值应当由一个工厂函数返回
        propE: {
            type: Object,
            default: function () {
                return { message: 'hello' }
            }
        },
        // 自定义验证函数
        propF: {
            validator: function (value) {
                return value > 10
            }
        }
    }
})
```

常用的prop验证规则

组件是不可以改变父级的数据数据的，他只能接收数据。但我们可以使用$emit通过自定义事件向父级发送通知。

```
<div id="app">
    {{pid}}
    <hd-php @emit="parent"></hd-php>    <hd-php @emit="parent"></hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        data: function () {
            return {num: 1}
        },
        template: "<button @click='add'>{{num}}</button>",
        methods: {
            add: function () {
                this.num++;
                this.$emit('emit', 2)
            }
        }
    })
    new Vue({
        el: '#app',
        data: {pid: 1},
        methods: {
            parent: function (n) {
                this.pid += n;
            }
        }
    })
</script>
```

通过自定义事件向父级推送通知

内容分发指我们在父级可以向组件内容分发内容。父级内容会分发到子组件的<slot></slot>容器中。

```
<div id="app">
    <hd-php>我是父级内容</hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        data: function () {
            return {num: 1}
        },
        template: "<h1>我是子组件内容:<slot></slot></h1>",
    })
    new Vue({
        el: '#app'
    })
</script>
```

内容分发slot

子组件中可以有多个slot，可以为slot设置name名称。

```
<div id="app">
    <hd-php>
        <span slot="s1">放在name为s1的slot中</span>
        <span>放在没有name的slot中那是默认的</span>
    </hd-php>
</div>
<script>
    Vue.component('hdPhp', {
        data: function () {
            return {num: 1}
        },
        template: "<h1><slot name='s1'></slot><slot></slot></h1>",
    })
    new Vue({
        el: '#app'
    })
</script>
```

具名分发slot