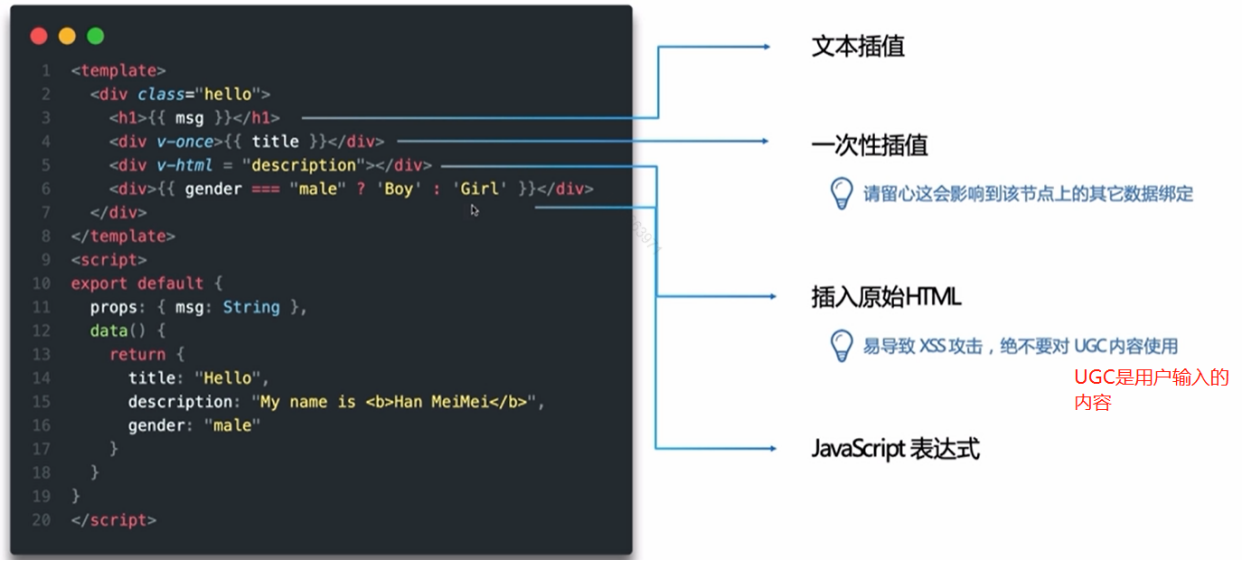
课程目标

模板语法

模板语法

vue使用了基于HTML的模板语法，所有的vue.js的模板语法都是合法的html，所以能被遵循规范的浏览器和HTML解析器解析。

模板插值



指令

常用指令

指令 (Directives) 是带有 v- 前缀的特殊特性

```vue
<template>
  <div class="hello">
    <img v-bind:alt="title" src="../assets/logo.png">
    <button v-on:click="showTitle">showTitle</button>
  </div>
</template>
<script>
export default {
  props: { msg: String },
  data() {
    return {
      title: "Hello Li Lei",
    }
  },
  methods: {
    showTitle() { alert(this.title); }
  }
}
</script>
```

```vue
<img :alt="title" src="../assets/logo.png">
<button @click="showTitle">showTitle</button>
```

条件渲染指令

v-if

```vue
<div v-if="male">i'm super man!</div>
<div v-else-if="female">i'm super woman!</div>
<div v-else>i'm no one!</div>
```

- “真正”的条件渲染，在切换过程中条件块内的事件监听器和子组件会被适当地被销毁和重建。
- 是 惰性 的：直到条件第一次变为真时，才会开始渲染条件块。

v-show

```vue
<div v-show="handle">
  <v-rich-editor></v-rich-editor>
</div>
```

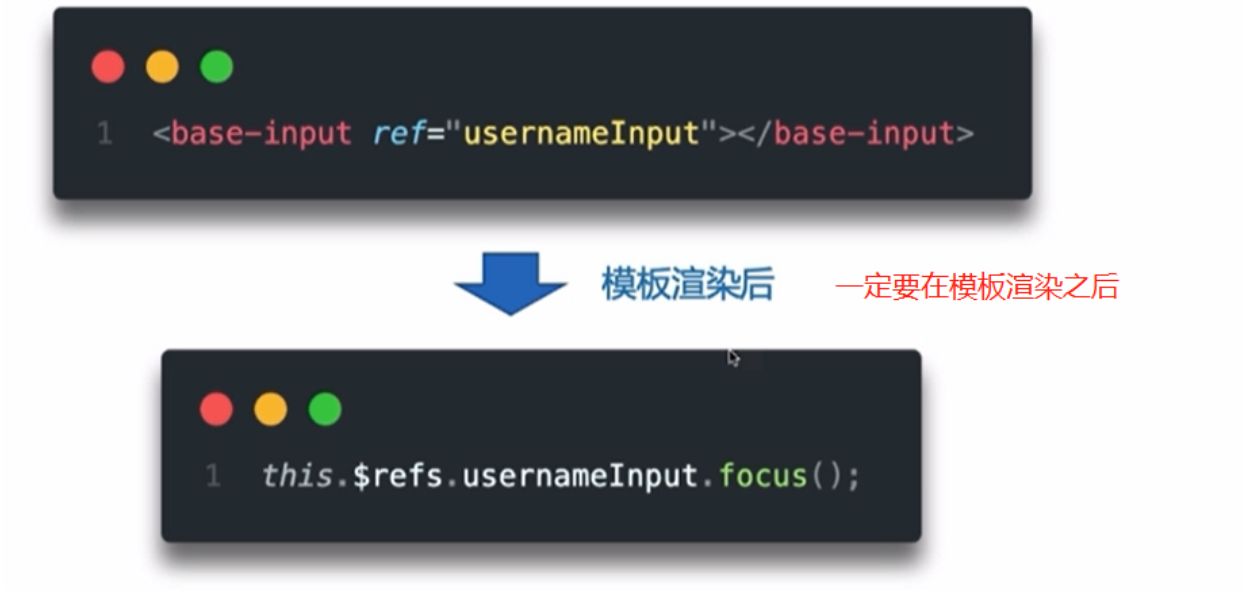
- 带有 v-show 的元素始终会被渲染并保留在 DOM 中。v-show 只是简单地切换元素的 CSS 属性 display

列表循环指令

```vue
<template>
  <div class="hello">
    <div v-for="(p, index) in classmates" :key="p.id">
        {{`${index}.${p.name}`}}
    </div>
  </div>
</template>
<script>
export default {
  props: { msg: String },
  data() {
    return {
      classmates: [{id:1, name:"Han Meimei"},
        {id: 2, name: "Li Lei"},
        {id: 2, name: "Lily"}]
    }
  }
}
</script>
```

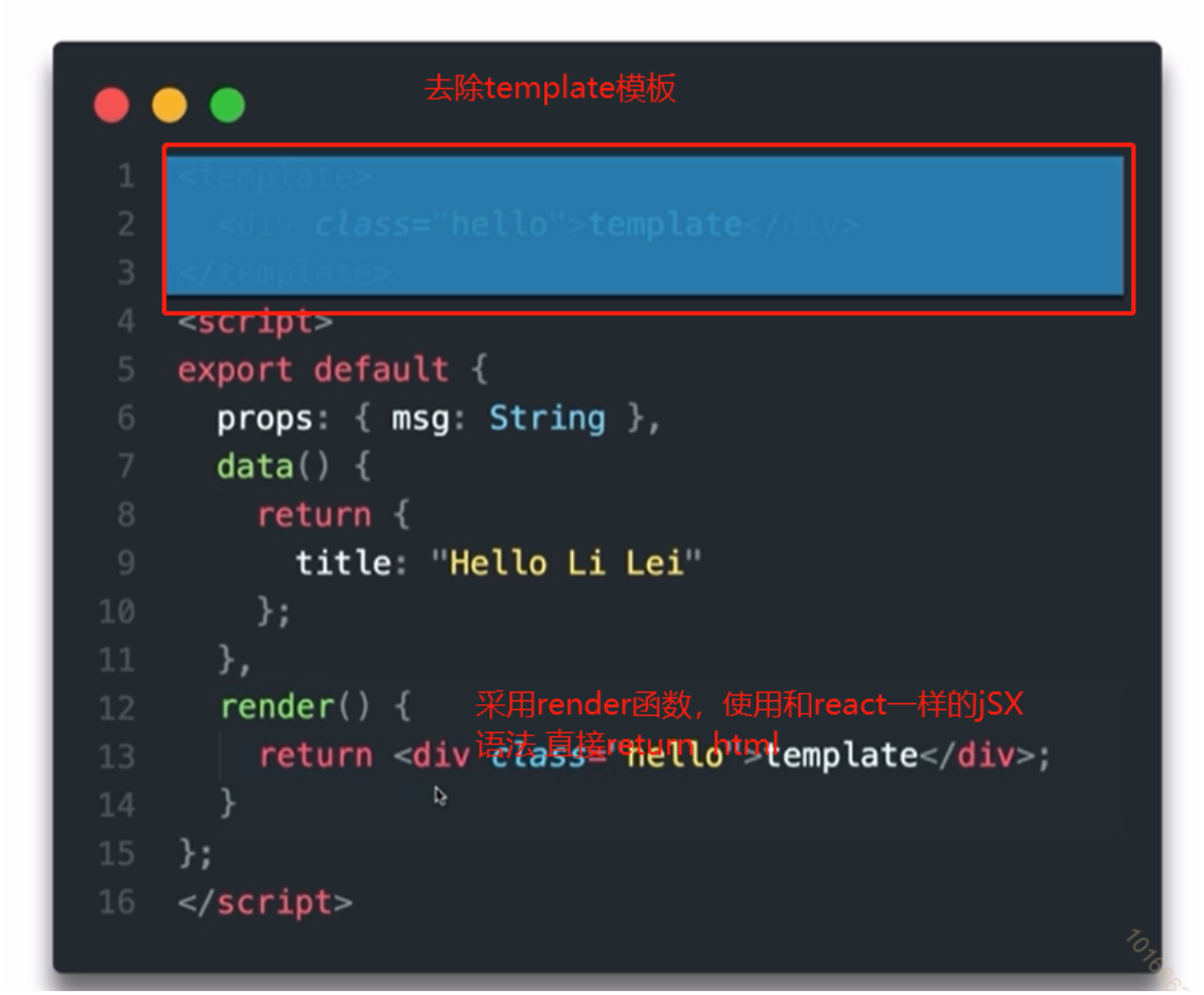
ref

需要在JavaScript里直接访问一个子组件或者模板上的dom节点，这时候就可以使用到ref，例如：


<base-input ref="usernameInput"></base-input>
模板渲染后
一定要在模板渲染之后
this.$refs.usernameInput.focus();

JSX

渲染函数


去除template模板
<template>
template
</template>
<script>
export default {
props: { msg: String },
data() {
return {
title: "Hello Li Lei"
};
},
render() {
采用render函数，使用和react一样的jSX
语法 直接return html
return template ;
}
};
</script>

jsx中的条件渲染

```
<script>
export default {
  props: { msg: String },
  data() {
    return {
      user: {
        age: 19,
        name: "Li Lei"
      }
    };
  },
  render() {
    if (this.user.age > 18) {
      return <div> Welcome, {this.user.name}</div>;
    }
    return <div>No Log</div>;
  }
};
</script>
```

Jsx中的列表渲染

```vue
<script>
export default {
  data() {
    return {
      classmates: [
        { id: 1, name: "Han Meimei" },
        { id: 2, name: "Li Lei" },
        { id: 3, name: "Lily" }
      ]
    };
  },
  render() {
    return (
      <div class="hello">
        {this.classmates.map((p, index) => (
          <div key={p.id}>{`${index}.${p.name}`}</div>
        ))}
      </div>
    );
  }
};
</script>
```