

Vue Beyond Vue Loader

蒋豪群

关于我

- 蒋豪群
- Vue CLI / Vue Loader
- Weibo @哦胖茶
- Twitter @haoqunjiang

书写一个 Vue 组件

Vue.component

```
Vue.component('todo-item', {  
  props: ['todo'],  
  template: '<li>{{ todo.text }}</li>'  
})
```

In-DOM Template

```
<script
  type="text/x-template"
  id="hello-world-template"
>
  <p>Hello hello hello</p>
</script>

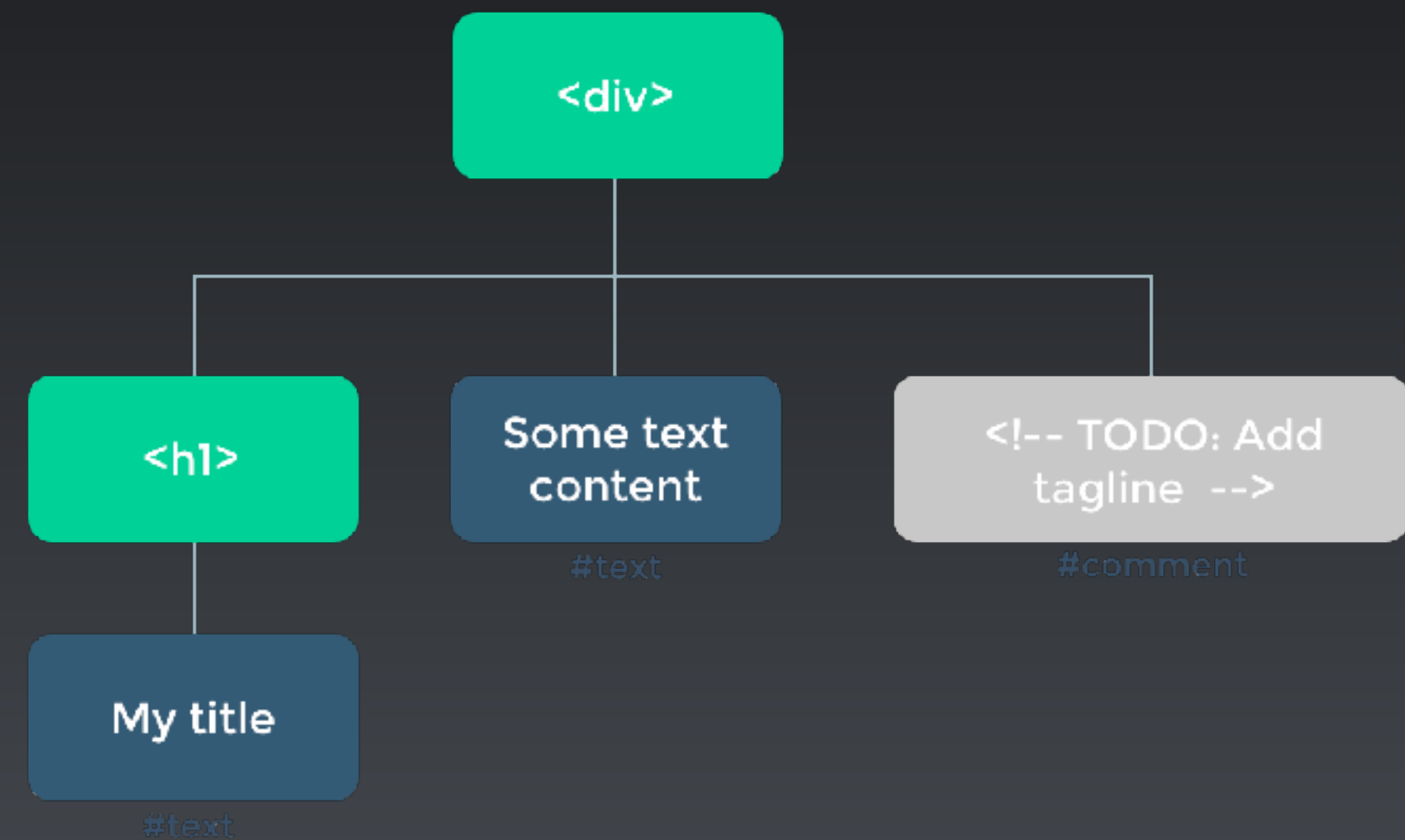
Vue.component('hello-world', {
  template: '#hello-world-template'
})
```

Behind The Scene

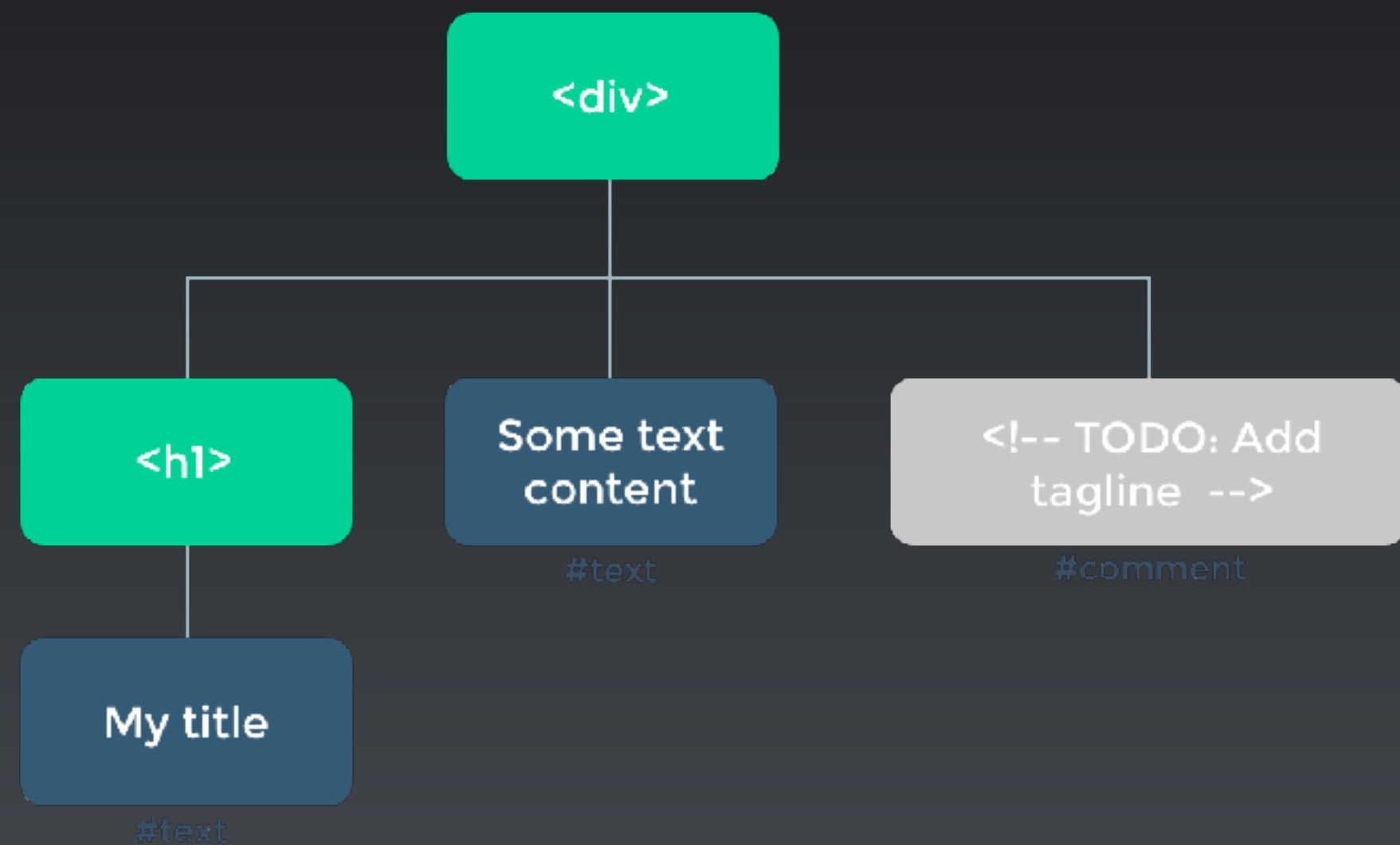
Render Functions

Virtual DOM & VNode

```
<div>  
  <h1>My title</h1>  
  Some text content  
  <!-- TODO: Add tagline -->  
</div>
```



Virtual DOM & VNode



```
createElement(  
  'div', [  
    createElement(  
      'h1',  
      [createTextNode("My title")]  
    ),  
    createTextNode("Some text content"),  
    createEmptyVNode("TODO: Add tagline")  
  ])
```

Hyper Script

```
createElement(  
  'div', [  
    createElement(  
      'h1',  
      [createTextNode("My title")]  
    ),  
    createTextNode("Some text content"),  
    createEmptyVNode("TODO: Add tagline")  
  ])  
)
```

```
render(h) {  
  return h('div', [  
    h(  
      'h1',  
      [h("My title")]  
    ),  
    h("Some text content")  
  ])  
}
```

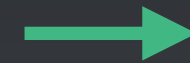
所以.....

预编译 (使用Vue Loader)

```
<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'app',
  components: {
    HelloWorld
  }
}
</script>
```



```
var render = function () {var _vm=this;var _h=_vm.$createElement;var _c=_vm._self._c||_h;return _c('div',{attrs:{"id":"app"}},[_c('img',{attrs:{"alt":"Vue logo","src":__webpack_require__("cf05")}}),_c('HelloWorld',{attrs:{"msg":"Welcome to Your Vue.js App"}})],1)}
var staticRenderFns = []

var App_component = normalizeComponent(
  src_Appvue_type_script_lang_js_,
  render,
  staticRenderFns,
  false,
  null,
  null,
  null
)
```

Vue Loader 做了什么

主要工作:

- 模板解析和编译
- 表达式的新语法支持 + 作用域分析
- 合并组件的所有区块
并支持 scoped css, 资源 url 变换, 等等

vue-template-compiler

vue-template-es2015-compiler

@vue/component-compiler-utils

vue-loader

- 但是.....
- 假如我不想/不能用 webpack?
- 假如我需要更进一步自定义我的工具链?

SFC without Vue Loader

rollup-plugin-vue



“Rollup is ideal for library bundling”

```
import vue from 'rollup-plugin-vue'

export default {
  input: 'src/MyComponent.vue',
  output: {
    format: 'esm',
    file: 'dist/MyComponent.js'
  },
  plugins: [
    vue()
  ]
}
```


@vue/dev-server

- Demo
- 一个用于静态文件服务器的中间件
- 直接在浏览器中使用 `import App from 'App.vue'` 语句
- 原理：
 - 遇到 `.vue` 文件时自动处理成 ES modules
 - 浏览器只需要 MIME type, 扩展名对执行无影响

自己动手！

- `@vue/component-compiler-utils`
- 用于 Parcel, Gridsome 等开源项目



Any Alternatives to SFC?

JSX

- 手动拼装 VNode 数据结构
- @vue/babel-preset-jsx
- @babel/plugin-syntax-jsx

JSX - Directives & Modifiers

```
<input vOn:click="newTodoText" />
```

```
<input vOn:click_stop_prevent="newTodoText" />
```

Alternative Syntaxes

Template Strings

- 换一种方式手动拼装 VNode
- 考虑利用 JavaScript 内置语言特性解析模板，进一步减小体积
- vue-html

vue-html

```
import HTML from 'vue-html'

Vue.use(HTML)
const Todos = {
  props: ['todos'],
  render(html) {
    return html`
      <ul>
        ${this.todos.map((todo, index) => {
          return html`
            <li key=${index}>${todo}</li>
          `
        })}
      </ul>
    `
  }
}
```


类型安全？

Vue ♥ TypeScript

Vue 2

- vue-class-component
- vue-property-decorator

Vue 2

```
import { Component, Prop, Vue } from 'vue-property-decorator';

@Component
export default class HelloWorld extends Vue {
  // props
  @Prop(String) private msg = 'VueConf Shanghai';

  // computed
  get computedMsg () {
    return 'computed ' + this.msg
  }

  // method
  greet () {
    alert('greeting: ' + this.msg)
  }

  // lifecycle hook
  mounted () {
    this.greet()
  }
}
```

Decorators 的问题

- Stage 2 标准，未来仍可能变动
- TypeScript 的实现已经与最新标准脱节很久
- Babel 的实现与 TypeScript 的实现也有差别

Decorators 的问题

```
@Component
class MyComp extends Vue {
  foo = 'foo + ' + this.bar
  @Prop(String) bar
  baz = 'baz + ' + this.bar
}

const c = new MyComp({
  propsData: { bar: 'bar' }
})
```

Vue 3

- Class API 不再在 **Vue Core** 中支持
- Composition function 能提供更好的类型推断
- vue-class-component & vue-property-decorator 仍然会在 Vue 3 中继续提供支持

Thanks