

02

OPEN ORIENTED

凹凸实验室

# vue2.0 render 函数

# 之前的做法

```

<div id="app">
  <anchored-heading :level="1">Hello world!</anchored-heading>
</div>

<script type="text/x-template" id="anchored-heading-template">
  <div>
    <h1 v-if="level === 1">
      <slot></slot>
    </h1>
    <h2 v-if="level === 2">
      <slot></slot>
    </h2>
    <h3 v-if="level === 3">
      <slot></slot>
    </h3>
    <h4 v-if="level === 4">
      <slot></slot>
    </h4>
    <h5 v-if="level === 5">
      <slot></slot>
    </h5>
    <h6 v-if="level === 6">
      <slot></slot>
    </h6>
  </div>
</script>

<script>
  Vue.component('anchored-heading', {
    template: '#anchored-heading-template',
    props: {
      level: {
        type: Number,
        required: true
      }
    }
  })

  new Vue({
    el: '#app'
  })
</script>

```

## 官网例子

<http://cn.vuejs.org/v2/guide/render-function.html>

## 渲染结果

```

▼ <div id="app">
  ▼ <div>
    <h1>Hello world!</h1>
    <!-->
    <!-->
    <!-->
    <!-->
    <!--> == $0
  </div>

```

冗余 浪费

# Render方式

```
<div id="app">
  <anchored-heading :level="2">Hello world!</anchored-heading>
</div>

<script>
  Vue.component('anchored-heading', {
    render: function (createElement) {
      return createElement(
        'h' + this.level,
        this.$slots.default
      )
    },
    props: {
      level: {
        type: Number,
        required: true
      }
    }
  })

  new Vue({
    el: '#app'
  })
</script>
```

## 渲染结果

```
▼ <div id="app">
  <h2>Hello world!</h2> == $0
</div>
```

# createElement

```
// @returns {VNode}
createElement(
  // {String | Object | Function}
  // 一个 HTML 标签字符串，组件选项对象，或者一个返回值类型为String/Object的函数，必要参数
  'div',

  // {Object}
  // 一个包含模板相关属性的数据对象
  // 这样，您可以在 template 中使用这些属性.可选参数.
  {
    // (详情见下一节)
  },

  // {String | Array}
  // 子节点 (VNodes)，可以是一个字符串或者一个数组。可选参数。
  [
    createElement('h1', 'hello world'),
    createElement(MyComponent, {
      props: {
        someProp: 'foo'
      }
    }),
    'bar'
  ]
)
```

与js中添加dom节点类似，接受三个参数，后面两个可选

第二个参数为数据对象，代表用在该节点的属性

第三个参数为该节点下面的子节点

```
var getChildrenTextContent = function (children) {
  return children.map(function (node) {
    return node.children
      ? getChildrenTextContent(node.children)
      : node.text
  }).join('')
}

Vue.component('anchored-heading', {
  render: function (createElement) {
    // create kebabCase id
    var headingId = getChildrenTextContent(this.$slots.default)
      .toLowerCase()
      .replace(/\W+/g, '-')
      .replace(/^|-|$/g, '')

    return createElement(
      'h' + this.level,
      [
        createElement('a', {
          attrs: {
            name: headingId,
            href: '#' + headingId
          }
        }, this.$slots.default)
      ]
    )
  },
  props: {
    level: {
      type: Number,
      required: true
    }
  }
})
```

## 参考资料

官方文档: <http://cn.vuejs.org/>

博文: <http://blog.csdn.net/kkae8643150/article/details/52910389>

源码浅析: <http://www.tuicool.com/articles/aMRNRf>

**T H A N K S**  
**FOR YOUR WATCHING**