

keep-alive 缓存组件

缓存包裹在其中的动态切换组件。

keep-alive 是 Vue 提供的一个抽象组件，主要用于保留组件状态或避免重新渲染。**<keep-alive>** 包裹动态组件时，会缓存不活动的组件实例，而不是销毁他们。它自身不会渲染一个DOM元素，也不会出现在父组件链中。

```
<!-- index.vue -->
<template>
  <div>
    <button @click="onChangeTab('a')">A</button>
    <button @click="onChangeTab('b')">B</button>
    <button @click="onChangeTab('c')">C</button>

    <keep-alive-a v-if="tab === 'a'"></keep-alive-a>
    <keep-alive-b v-if="tab === 'b'"></keep-alive-b>
    <keep-alive-c v-if="tab === 'c'"></keep-alive-c>
  </div>
</template>

<script>
export default {
  name: 'KeepAliveComponent',
  data() {
    return {
      tab: 'a',
    }
  },
  components: {
    // 用到了异步组件
    'keep-alive-a': () => import('./a.vue'),
    'keep-alive-b': () => import('./b.vue'),
    'keep-alive-c': () => import('./c.vue'),
  },
  methods: {
    onChangeTab(value) {
      this.tab = value;
    }
  },
}
</script>
```

```
<!-- a.vue -->
<template>
  <div>keep-alive A</div>
</template>
```

```
<script>
export default {
  name: 'KeepAliveA',
  mounted() {
    console.log("A mounted");
  },
  destroyed() {
    console.log("A destroyed");
  }
}
</script>
```

```
<!-- b.vue -->
<template>
  <div>keep-alive B</div>
</template>

<script>
export default {
  name: 'KeepAliveB',
  mounted() {
    console.log("B mounted");
  },
  destroyed() {
    console.log("B destroyed");
  }
}
</script>
```

```
<!-- c.vue -->
<template>
  <div>keep-alive C</div>
</template>

<script>
export default {
  name: 'KeepAliveC',
  mounted() {
    console.log("C mounted");
  },
  destroyed() {
    console.log("C destroyed");
  }
}
</script>
```

在上面的组织文件结构中，`index.vue` 中包含三个子组件 `a.vue`、`b.vue` 和 `c.vue`。三个子组件中分别在生命周期钩子函数（`mounted` 和 `destroyed`）中打印标识。

依次点击 'B' -> 'C' -> 'B' 时，输出如下：

1. 首先是 `KeepAliveA` 默认显示，会输出 `A mounted`；
2. 当点击 B 时，会先输出 `A destroyed`，再输出 `B mounted`；
3. 当点击 C 时，会先输出 `B destroyed`，再输出 `C mounted`；
4. 再点击 B 时，会先输出 `C destroyed`，再输出 `B mounted`。每一次都会先卸载当前组件，再渲染下一个组件。

如果组件较大，会造成性能开销较大。又如果想保留组件的状态的话，就无法做到。

这时可以通过给需要缓存状态的组件包裹一个 `<keep-alive></keep-alive>`，做到让组件缓存，不会再卸载 DOM 和重新渲染 DOM。

在 `index.vue` 中通过给三个子组件包裹一个 `<keep-alive></keep-alive>`，实现子组件缓存：

```
...  
<keep-alive>  
  <keep-alive-a v-if="tab === 'a'"></keep-alive-a>  
  <keep-alive-b v-if="tab === 'b'"></keep-alive-b>  
  <keep-alive-c v-if="tab === 'c'"></keep-alive-c>  
</keep-alive>  
...
```

当切换到其它组件时，当前组件将不会再卸载，并且后面再次点击该组件时，不会再重新渲染。

再重复上面点击步骤，输出如下：

1. 首先是 `KeepAliveA` 默认显示，会输出 `A mounted`；
2. 当点击 B 时，输出 `B mounted`；
3. 当点击 C 时，输出 `C mounted`；
4. 再点击 B 时，不会有任何输出了。

Vue 2 中是 `<keep-alive></keep-alive>` 小写，在 Vue3 中是 `<KeepAlive></KeepAlive>` 大写。