前置知识：

vue创建组件简易写法

const testCompoment = {templeat : ' test '}

正文：

很多时候，我们需要将给定匹配模式的路由映射到同一个组件。例如，我们可能有一个 `User` 组件，它应该对所有用户进行渲染，但用户 ID 不同。在 Vue Router 中，我们可以在路径中使用一个动态字段来实现，我们称之为 路径参数：

```js
const User = {
  template: '<div>User</div>',
}

// 这些都会传递给 `createRouter`
const routes = [
  // 动态字段以冒号开始
  { path: '/users/:id', component: User },
]
```

现在像 `/users/johnny` 和 `/users/jolyne` 这样的 URL 都会映射到同一个路由。

路径参数 用冒号 `:` 表示。当一个路由被匹配时，它的 *params* 的值将在每个组件中以 `this.$route.params` 的形式暴露出来。因此，我们可以通过更新 `User` 的模板来呈现当前的用户 ID：

```js
const User = {
  template: '<div>User {{ $route.params.id }}</div>',
}
```

你可以在同一个路由中设置有多个 路径参数，它们会映射到 `$route.params` 上的相应字段。例如：

匹配模式	匹配路径	$route.params
ısers/:username	/users/eduardo	`{ username: 'eduardo' }`
ısers/:username/posts/:postId	/users/eduardo/posts/123	`{ username: 'eduardo', postId: '12`

除了 $route.params 之外，$route 对象还公开了其他有用的信息，如 $route.query（如果 URL 中存在参数）、

$route.hash 等。你可以在 API 参考中查看完整的细节

响应路由参数的变化

使用带有参数的路由时需要注意的是，当用户从 `/users/johnny` 导航到 `/users/jolyne` 时，**相同的组件实例将被重复使用**。因为两个路由都渲染同个组件，比起销毁再创建，复用则显得更加高效。**不过，这也意味着组件的生命周期钩子不会被调用。**

要对同一个组件中参数的变化做出响应的话，你可以简单地 watch `$route` 对象上的任意属性，在这个场景中，就是 `$route.params` ：

```js
const User = {
  template: '...',
  created() {
    this.$watch(
      () => this.$route.params,
      (toParams, previousParams) => {
        // 对路由变化做出响应...
      }
    )
  },
}
```

或者，使用 `beforeRouteUpdate` 导航守卫，它也可以取消导航：

```js
const User = {
  template: '...',
  async beforeRouteUpdate(to, from) {
    // 对路由变化做出响应...
    this.userData = await fetchUser(to.params.id)
  },
}
```