

实例方法：

实例方法	说明
<code>this.\$router.push(location, onComplete?, onAbort?)</code>	这个方法会向 history 栈添加按钮时，则回到之前的 URL。用 <code>router.push(...)</code> 。
<code>this.\$router.replace(location, onComplete?, onAbort?)</code>	这个方法不会向 history 添加当前的 history 记录，所以，前的 URL。
<code>this.\$router.go(n)</code>	这个方法的参数是一个整数，步，类似 <code>window.history.go</code>

注意：

- 1、在 2.2.0+，可选的在 `router.push` 或 `router.replace` 中提供 `onComplete` 和 `onAbort` 回调作为第二个和第三个参数。这些回调将会在导航成功完成 (在所有的异步钩子被解析之后) 或终止 (导航到相同的路由、或在当前导航完成之前导航到另一个不同的路由) 的时候进行相应的调用。
- 2、如果目的地和当前路由相同，只有参数发生了改变 (比如从一个用户资料到另一个 `/users/1 -> /users/2`)，你需要使用 `beforeRouteUpdate` 来响应这个变化 (比如抓取用户信息)。

参数传递方式：

vue-router提供了`params`、`query`、`meta`三种页面间传递参数的方式。

示例：

```
// 字符串，不带参数
this.$router.push('home')
```

```
// 对象，不带参数
this.$router.push({ path: 'home' })
```

```
// params (推荐)：命名的路由，params 必须和 name 搭配使用
this.$router.push({ name: 'user', params: { userId: 123 } })
```

```
// 这里的 params 不生效
this.$router.push({ path: '/user', params: { userId: 123 } })
```

```
// query：带查询参数，变成 /register?plan=private
this.$router.push({ path: 'register', query: { plan: 'private' } })
```

//meta方式：路由元信息

```
export default new Router({
  routes: [
    {
      path: '/user',
      name: 'user',
      component: user,
      meta: {
        title: '个人中心'
      }
    }
  ]
})
```

在组件中使用：

//通过 \$route 对象获取，注意是route，么有r
this.\$route.params

this.\$route.query

this.\$route.meta

问题二：\$route 和 \$router 的区别

\$route是“路由信息对象”，包括path, params, hash, query, fullPath, matched, name等路由信息参数。

\$router是“路由实例”对象包括了路由的跳转方法，钩子函数等。