

Message 消息提示

常用于主动操作后的反馈提示。与 Notification 的区别是后者更多用于系统级通知的被动提醒。

基础用法

从顶部出现，3 秒后自动消失。

:::demo Message 在配置上与 Notification 非常类似，所以部分 options 在此不做详尽解释，文末有 options 列表，可以结合 Notification 的文档理解它们。Element 注册了一个 `$message` 方法用于调用，Message 可以接收一个字符串或一个 VNode 作为参数，它会被显示为正文内容。

```
<template>
  <el-button :plain="true" @click="open">打开消息提示</el-button>
  <el-button :plain="true" @click="openVn">VNode</el-button>
</template>

<script>
  export default {
    methods: {
      open() {
        this.$message('这是一条消息提示');
      },

      openVn() {
        const h = this.$createElement;
        this.$message({
          message: h('p', null, [
            h('span', null, '内容可以是 '),
            h('i', { style: 'color: teal' }, 'VNode')
          ])
        });
      }
    }
  }
</script>
```

...

不同状态

用来显示「成功、警告、消息、错误」类的操作反馈。

:::demo 当需要自定义更多属性时，Message 也可以接收一个对象为参数。比如，设置 `type` 字段可以定义不同的状态，默认为 `info`。此时正文内容以 `message` 的值传入。同时，我们也为 Message 的各种 `type` 注册了方法，可以在不传入 `type` 字段的情况下像 `open4` 那样直接调用。

```
<template>
  <el-button :plain="true" @click="open2">成功</el-button>
  <el-button :plain="true" @click="open3">警告</el-button>
  <el-button :plain="true" @click="open1">消息</el-button>
  <el-button :plain="true" @click="open4">错误</el-button>
</template>
```

```

</template>

<script>
  export default {
    methods: {
      open1() {
        this.$message('这是一条消息提示');
      },
      open2() {
        this.$message({
          message: '恭喜你，这是一条成功消息',
          type: 'success'
        });
      },
      open3() {
        this.$message({
          message: '警告哦，这是一条警告消息',
          type: 'warning'
        });
      },
      open4() {
        this.$message.error('错了哦，这是一条错误消息');
      }
    }
  }
</script>

```

...

可关闭

可以添加关闭按钮。

demo 默认的 Message 是不可以被人工关闭的，如果需要可手动关闭的 Message，可以使用 `showClose` 字段。此外，和 Notification 一样，Message 拥有可控的 `duration`，设置 0 为不会被自动关闭，默认为 3000 毫秒。

```

<template>
  <el-button :plain="true" @click="open1">消息</el-button>
  <el-button :plain="true" @click="open2">成功</el-button>
  <el-button :plain="true" @click="open3">警告</el-button>
  <el-button :plain="true" @click="open4">错误</el-button>
</template>

<script>
  export default {
    methods: {
      open1() {
        this.$message({
          showClose: true,
          message: '这是一条消息提示'
        });
      }
    }
  }

```

```

    });
  },

  open2() {
    this.$message({
      showClose: true,
      message: '恭喜你, 这是一条成功消息',
      type: 'success'
    });
  },

  open3() {
    this.$message({
      showClose: true,
      message: '警告哦, 这是一条警告消息',
      type: 'warning'
    });
  },

  open4() {
    this.$message({
      showClose: true,
      message: '错了哦, 这是一条错误消息',
      type: 'error'
    });
  }
}
}
}
</script>

```

...

文字居中

使用 `center` 属性让文字水平居中。

:::demo

```

<template>
  <el-button :plain="true" @click="openCenter">文字居中</el-button>
</template>

<script>
export default {
  methods: {
    openCenter() {
      this.$message({
        message: '居中的文字',
        center: true
      });
    }
  }
}

```

```
}  
</script>
```

...

使用 HTML 片段

`message` 属性支持传入 HTML 片段

::demo 将 `dangerouslyUseHTMLString` 属性设置为 `true`, `message` 就会被当作 HTML 片段处理。

```
<template>  
  <el-button :plain="true" @click="openHTML">使用 HTML 片段</el-button>  
</template>  
  
<script>  
  export default {  
    methods: {  
      openHTML() {  
        this.$message({  
          dangerouslyUseHTMLString: true,  
          message: '<strong>这是 <i>HTML</i> 片段</strong>'  
        });  
      }  
    }  
  }  
</script>
```

...

::warning `message` 属性虽然支持传入 HTML 片段,但是在网站上动态渲染任意 HTML 是非常危险的,因为容易导致 [XSS 攻击](#)。因此在 `dangerouslyUseHTMLString` 打开的情况下,请确保 `message` 的内容是可信的,永远不要将用户提交的内容赋值给 `message` 属性。...

全局方法

Element 为 `Vue.prototype` 添加了全局方法 `$message`。因此在 `vue instance` 中可以采用本页面中的方式调用 `Message`。

单独引用

单独引入 `Message`：

```
import { Message } from 'element-ui';
```

此时调用方法为 `Message(options)`。我们也为每个 type 定义了各自的方法,如 `Message.success(options)`。并且可以调用 `Message.closeAll()` 手动关闭所有实例。

Options

参数	说明	类型	可选值	默认值
----	----	----	-----	-----

message	消息文字	string / VNode	—	—
type	主题	string	success/warning/info/error	info
iconClass	自定义图标的类名，会覆盖 type	string	—	—
dangerouslyUseHTMLString	是否将 message 属性作为 HTML 片段处理	boolean	—	false
customClass	自定义类名	string	—	—
duration	显示时间, 毫秒。设为 0 则不会自动关闭	number	—	3000
showClose	是否显示关闭按钮	boolean	—	false
center	文字是否居中	boolean	—	false
onClose	关闭时的回调函数, 参数为被关闭的 message 实例	function	—	—
offset	Message 距离窗口顶部的偏移量	number	—	20

方法

调用 `Message` 或 `this.$message` 会返回当前 `Message` 的实例。如果需要手动关闭实例，可以调用它的 `close` 方法。

方法名	说明
close	关闭当前的 Message