

HTML 5.2 新元素

1 . dialog

1 . 基础用法

```
<dialog open>

  Native dialog box!

</dialog>
```

其中, open 属性表示此时 dialog 是可见的, 如果没有 open dialog 会隐藏(我们可以使用 dialog 提供的方法来操作让它显示或者隐藏), 初始化默认位置是与页面垂直居中, 默认情况宽高由内容撑起来的;

2 . 基本操作

JavaScript 提供 dialog 的原生方法和属性, 可以很方便的处理 dialog 元素:

```
const dialog = document.querySelector('dialog');

//显示

dialog.showModal()      //带背景显示(背景可以用css自定义, 默认是灰色透明背景)

dialog.show()           //不带背景显示

//关闭

dialog.close()          //close方法可以接受参数, 关闭之后的返回值(默认支持 ESC 关闭);
```

3 . 样式

打开和关闭模态框是最基本的, 但肯定还是不够的, <dialog>的初始化样式是不怎么好看的, 因此, 我们需要为它自定义样式, 我们还可以通过 ::backdrop 伪类来修改 显示时候的背景:

```
dialog{

  padding: 10px;

  text-align: center;

  vertical-align: middle;

  border-radius: 5px;

  border: 0;

  position: absolute;

  left: 50%;

  top: 50%;

  transform: translate(-50%,-50%);

  border: 1px solid rgba(0,0,0,0.3);

  box-shadow: 0 0 20px rgba(0,0,0,0.3);

}

dialog::backdrop {

  background-color: rgba(0, 0, 0, 0.7);

}
```

接下来, 我们可以向 dialog 里面添加一些其他的 html 元素:

```
<dialog>

  <h3>title</h3>

  <p>这是一条提示信息</p>

</dialog>
```

4 . 进阶操作

通常, 我们希望能从 dialog 中获取一些用户信息(关闭时候的回调), 我们可以给 close 方法传递一个参数, 然后通过 dialog 提供的 returnValue 来获取信息; 当然, 还存在额外的监听事件, 其中最常用的可能就是 close (关闭时候的触发) 还有 cancel (取消时候的触发), 点击阴影部分会触发 dialog 的点击事件, 如果 dialog 的子元素占满了整个 dialog , 我们可以通过监听 dialog 的点击, 当target 为 dialog 元素本身的时候来关闭它

```
dialog.addEventListener('click', (event) => {

  if (event.target === dialog) {

    dialog.close('cancelled');

    console.log(dialog.returnValue);      //关闭时候的返回值

  }

});
```