

# 使用指南：index.js

---

`index.js` 是一个复杂的 JavaScript 文件，定义了一系列类和工具函数，主要用于处理浏览器环境中的 DOM 操作、事件处理、动画、加载器等。以下是对该文件的主要功能和使用方法的简要指南。

## 1. 全局设置与初始化

- 文件开头定义了一些全局变量和初始化操作，确保在浏览器环境中正常运行。
- `browserMode` 用于检测当前是否在浏览器环境中运行。

## 2. 核心类

- **RealWorld**: 事件循环类，用于管理异步任务和事件监听。
  - `onload`: 环境准备好时兑现的承诺。
  - `onceIf`: 生成条件检测承诺。
  - `cb2promise`: 将回调函数转换为 Promise。
  - `destroy`: 销毁对象。
  - `then`: 添加函数到执行队列。
- **RealNode**: 基础节点类，用于创建和管理节点。
  - `createExpression`: 创建表达式节点。
  - `check`: 检查节点是否存在。
  - `afterNow`: 在当前事件循环后执行函数。
  - `copyObj`: 复制对象。
- **RealElement**: 继承自 **RealNode**，用于处理 HTML 元素。
  - `getDomByString`: 通过字符串获取 DOM 元素。
  - `makeElement`: 创建 HTML 元素。
  - `addCSSRules`: 添加 CSS 规则。
  - `applyCSS`: 应用 CSS 样式。
- **RealCanvas**: 继承自 **RealElement**，用于处理 Canvas 元素。
  - `getImageBySrc`: 通过 URL 获取图像。
  - `animate`: 创建动画。
  - `toBlob`: 将 Canvas 内容转换为 Blob。
- **RealLoader**: 继承自 **RealElement**，用于处理文件加载。
  - `load`: 加载文件。
  - `onerror`: 加载错误时的回调。
  - `onloadend`: 加载完成时的回调。
- **RealSelect**: 继承自 **RealElement**，用于处理下拉选择框。
  - `protoSet`: 设置选项。

- **fix**: 更新下拉框内容。
- **RealDivList**: 继承自 **RealElement**, 用于处理动态列表。
  - **defineDivListClass**: 定义列表类。
  - **createByClassName**: 通过类名创建列表。
- **RealImgList**: 继承自 **RealDivList**, 用于处理图像列表。
  - **cloneImgList**: 克隆图像列表。
- **RealDivQueue**: 继承自 **RealDivList**, 用于处理队列。
  - **applyQueue**: 应用队列。

### 3. 工具函数

- **createRealDivSelect**: 创建一个自定义的下拉选择框。
- **createRealDivSearch**: 创建一个自定义的搜索框。
- **RealPromise**: 自定义的 Promise 实现, 用于处理异步任务。

### 4. 事件处理

- **RealWorld.onload**: 在页面加载完成后执行的操作。
- **RealElement.addEventListenerBySelectors**: 通过选择器添加事件监听器。

### 5. 动画与加载

- **RealCanvas.animate**: 创建 Canvas 动画。
- **RealLoader.load**: 加载文件或资源。

### 6. 示例用法

#### 6.1 创建一个 Canvas 动画

```
const canvas = new RealCanvas('myCanvas', 800, 600);
canvas.animate({
  prefix: './img/frame_',
  suffix: '.png',
  startN: 1,
  length: 60,
  timeSep: 100
});
```

#### 6.2 创建一个自定义下拉选择框

```
const select = createRealDivSelect({
  option1: 'Option 1',
  option2: 'Option 2'
```

```
}, false, (e) => {  
  console.log('Selected:', e.target.value);  
});  
document.body.appendChild(select.self);
```

## 6.3 加载文件

```
const loader = new RealLoader(true, 'file.txt', () => fetch('./file.txt').then(res  
=> res.text()));  
loader.onloadend = (n) => {  
  console.log('File loaded:', n);  
};  
loader.load();
```

## 6.4 使用 RealPromise

```
RealPromise().then(() => {  
  console.log('RealPromise resolved');  
});  
RealPromise.resolve(true);
```

## 7. 注意事项

- 该文件依赖于浏览器环境，部分功能在非浏览器环境中可能无法正常运行。
- 使用 `RealCanvas` 和 `RealLoader` 时，确保资源路径正确。
- `RealPromise` 是一个自定义的 Promise 实现，使用时需注意其与标准 Promise 的差异。

## 8. 总结

`index.js` 提供了一个强大的工具集，用于处理浏览器中的 DOM 操作、事件处理、动画和文件加载等任务。通过合理使用这些类和函数，可以大大简化复杂的前端开发任务。