

课程安排，请关注微信公众平台或者官方微博

编程语言： **Golang** 与 **html5**

编程工具： **Goland** 和 **HBuilder**

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔。

大家好，

欢迎来到 字节教育 课程的学习

字节教育官网：www.ByteEdu.Com

腾讯课堂地址：Gopher.ke.qq.Com

技术交流群： 221 273 219

微信公众号： **Golang 语言社区**

微信服务号： **Golang 技术社区**

目录：

第一季	微信小游戏基础、进阶、提高课程.....	2
第二节	Adapter.....	2
1、	课程内容.....	2
2、	课后作业—难度：★ ☆ ☆ ☆ ☆	4
3、	微信公众平台及服务号	5

第一季 微信小游戏基础、进阶、提高课程

第二节 Adapter

1、课程内容

小游戏的运行环境在 iOS 上是 [JavaScriptCore](#)，在 Android 上是 [V8](#)，都是没有 BOM 和 DOM 的运行环境，没有全局的 document 和 window 对象。因此当你希望使用 DOM API 来创建 Canvas 和 Image 等元素的时候，会引发错误。

```
var canvas = document.createElement('canvas')
```

但是我们可以使用 wx.createCanvas 和 wx.createImage 来封装一个 document。

```
var document = {
  createElement: function (tagName) {
    tagName = tagName.toLowerCase()
    if (tagName === 'canvas') {
      return wx.createCanvas()
    }
    else if (tagName === 'image') {
      return wx.createImage()
    }
  }
}
```

这时代码就可以像在浏览器中创建元素一样创建 Canvas 和 Image 了。

```
var canvas = document.createElement('canvas')
var image = document.createElement('image')
```

同样，如果想实现 new Image() 的方式创建 Image 对象，只须添加如下代码。

```
function Image () {
  return wx.createImage()
}
```

这些使用 wx API 模拟 BOM 和 DOM 的代码组成的库称之为 Adapter。顾名思义，这是对基于浏览器环境的游戏引擎在小游戏运行环境下的一层适配层，使游戏引擎在调用 DOM API 和访问 DOM 属性时不会产生错误。Adapter 是一个抽象的代码层，并不特指某一个适配小游戏的第三方库，每位开发者都可以根据自己的项目需要实现相应的 Adapter。官方实现了一个 Adapter 名为 **weapp-adapter**，并提供了完整的源码，供开发者使用和参考。

Adapter 下载地址 weapp-adapter.zip

weapp-adapter 会预先调用 wx.createCanvas() 创建一个全屏 Canvas，并暴露为一个全局变量 canvas。

```
require('./weapp-adapter')
var context = canvas.getContext('2d')
context.fillStyle = 'red'
context.fillRect(0, 0, 100, 100)
```

除此之外 weapp-adapter 还模拟了以下对象和方法：

- document.createElement
- canvas.addEventListener
- localStorage
- Audio
- Image
- WebSocket
- XMLHttpRequest
- 等等...

需要强调的是，weapp-adapter 对浏览器环境的模拟远不完整的，仅仅只针对游戏引擎可能访问的属性和调用的方法进行了模拟，也不保证所有游戏引擎都能通过 weapp-adapter 顺利无缝接入小游戏。直接将 weapp-adapter 提供给开发者，更多地是作为参考，开发者可以根据需要在 weapp-adapter 的基础上进行扩展，以适配自己项目使用的游戏引擎。

weapp-adapter 不是小游戏基础库的一部分

小游戏基础库只提供 `wx.createCanvas` 和 `wx.createImage` 等 `wx` API 以及 `setTimeout/setInterval/requestAnimationFrame` 等常用的 JS 方法。在此之上的 `weapp-adapter` 是为了让基于浏览器环境的第三方代码更快地适配小游戏运行环境的一层适配层,并不是基础库的一部分。更准确地说,我们将 `weapp-adapter` 和游戏引擎都视为第三方库,需要开发者在小游戏项目中自行引入。

目前,Cocos、Egret、Laya 已经完成了自身引擎及其工具对小游戏的适配和支持,访问对应的官方文档可以更快地接入小游戏的开发。

- Cocos : <http://docs.cocos.com/creator/manual/zh/publish/publish-wechatgame.html>
- Egret : <http://developer.egret.com/cn/github/egret-docs/Engine2D/minigame/introduction/index.html>
- Laya : <https://ldc.layabox.com/doc/?nav=zh-as-3-4-5>

再次强调,`weapp-adapter` 不是小游戏基础库的一部分,今后官方也将不再对 `weapp-adapter` 进行更新和维护。开发者应该根据自己使用的游戏引擎,实现自己的 Adapter 来使所用的游戏引擎适配小游戏的运行环境。

2、课后作业—难度：★☆☆☆☆

Weaapp-adaper 如何理解及使用:

<1> 文字表达。

社区问答系统回答: Golang 技术社区 (服务号菜单栏)

3、微信公众平台及服务号



Golang 语言社区



Golang 技术社区

字节跳动教育