TITF 出品：微信小程序实用案例代码片段大全《五》：ES6 新特性专辑 ...

一：通过增强对象字面量创建方法，省略 function 写法

1、增强对象字面量简介

- 在增强对象字面量中创建方法，可省略 function 关键字
- 可以在增强对象字面量中定义原型，即继承某个增强对象字面量
- 可以直接调用父类中的方法

2、关键代码

index.wxml

```
<button type="default" bindtap="enhanceObjectLiteral">点击显示增强对象字面量效果
</button>
```

来自 CODE 的代码片

snippet_file_0.txt

index.js

```
Page({
  data:{
    // text:"这是一个页面"
  },
  enhanceObjectLiteral:function(){
    //通过对象字面量创建对象
    var province={
      //在该对象中创建一个方法，此处可以发现这种创建方式是不需要用到 function
      provinceName(provinceName){
        console.log(provinceName);
      }
    }
    //调用 province 对象中的 provinceName 方法，此时控制台显示出的是数据是“广州”
    province.provinceName("广州");
  }
})
```

来自 CODE 的代码片

二：字符串模板：美元符号$+大括号{}变量的写法

1、字符串模板简介

- ES6 新特性中的字符串模板允许使用英文字符抑音符号`（提示：这里我们不能将其理解为单引号）来创建字符串，并且在该字符串中可以包含都【美元符号+大括号】包裹的变量
- 格式：console.log(`ES6 新特性：${name}`)
- 说明：格式中的 name 为变量名

2、关键代码

index.wxml

```
<button type="default" bindtap="stringTemplate">点击我控制台输出字符串模块数据
</button>
```

来自 CODE 的代码片

snippet_file_0.txt

index.js

```
Page({
  data:{
    // text:"这是一个页面"
    stringTemplateTxt:''
  },
  stringTemplate:function(){
    var stringTemplateTxt="字符串模板数据"//定义一个变量并初始化值
    console.log(`我是${stringTemplateTxt}`)//将 stringTemplateTxt 变量的值输出到控制台
  }
})
```

来自 CODE 的代码片

三：使用解构数组优化变量格式

1、解构数组简介

- 特点：通过解构数组，我们可以将多个值返回成为一个数组，并将数组中的值赋予到对应的变量名中。
- 格式：[province,city]=['guangong','guangzhou']

2、关键代码

index.wxml

```
<view>{{jiegouArray}}</view>
<view>{{jiegouDat}}</view>
<button type="default" bindtap="jiegouBind">点击我显示用数组解构后的数据</button>
```

来自 CODE 的代码片

snippet_file_0.txt

四：使用箭头操作符简化回调函数繁琐的编写过程

1、类操作简介

- 箭头操作符格式：inputs=>outputs；
- 箭头操作符的出现是为了解决 JavaScript 中回调函数繁琐的编程步骤。

2、关键代码

index.wxml

```
<button type="default" bindtap="consoleLog">点击实现控制台数据输出</button>
```

来自 CODE 的代码片

snippet_file_0.txt

index.js

```
Page({
  data:{
    // text:"这是一个页面"
  },
  consoleLog:function(){
    var numArray=[0,1,2,3,4,5];
    //es6 新特性：箭头表示符写法
    numArray.forEach(serialNum=>console.log('我是新特性显示出来的数据：'+serialNum));
    //传统写法
    numArray.forEach(function(serialNum){
      console.log('我是传统写法显示出来的数据：'+serialNum);
    })
  }
})
```

来自 CODE 的代码片

五：通过对类的操作来处理数据后显示在视图界面

1、类操作简介

- 关键字： class
- 类的出现可以更加直观的将对象的创建和继承的代码显示在编程员前。

2、关键代码

index.wxml

```
<button type="default" bindtap="showAreas">点击我显示所有地区</button>
<view class="areas">
  <block wx:for-items="{{areas}}">
    <view class="area">{{item}}</view>
  </block>
</view>
```

来自 CODE 的代码片

snippet_file_0.txt

index.js

```
var dat=[
  {provice:'北京',city:'朝阳市'},
  {provice:'上海',city:'虹口区'},
  {provice:'广东',city:'广州市'},
  {provice:'广东',city:'深圳市'},
];
class areaCls{
  //构造函数
  constructor(provice,city){
     this.provice=provice;
     this.city=city;
  }
  //实例方法
  showAreaInfo(){
     return '省（市）：'+this.provice+',市（区）'+this.city+'。'
```

```
  }
}
var areas=[]
for(var i=0,len=dat.length;i<len;i++){
  //创建对象并调用实例方法，最后的实例方法中的值赋予到数据 areas 里面
  areas[i]=new areaCls(dat[i].provice,dat[i].city).showAreaInfo();
}
Page({
  data:{
    // text:"这是一个页面"
    areas:[]
  },
  showAreas:function(){
    this.setData({
      areas:areas
    })
```

```
32   }
33})
```

来自 CODE 的代码片