

校企合作项目记录表（跟岗学习专用）

记录时段	2019 年 11 月 18 日 —— 2019 年 11 月 22 日	
企业名称	广东元心科技有限公司	
记录类型	学习记录	
学生姓名	苏俊贤	指导老师签名：
记录内容（可另页附）	<p>由于工作的需要，学习了一种新的变量类型：Map</p> <p>Map 对象是 es6 新引入的一种类型，它保存键值对。任何值(对象或者原始值)都可以作为一个键或一个值。</p> <h3>语法</h3> <pre>new Map([iterable])</pre> <h3>参数</h3> <p>iterable</p> <p>Iterable 可以是一个 数组 或者其他 iterable 对象，其元素为键值对(两个元素的数组，例如: [[1, 'one'],[2, 'two']])。每个键值对都会添加到新的 Map。null 会被当做 undefined。</p> <p>一个 Map 对象在迭代时会根据对象中元素的插入顺序来进行一个 for...of 循环在每次迭代会返回一个形式为[key, value]的数组。</p> <p>它跟 Object 相比较：</p> <p>Objects 和 Maps 类似的是，它们都允许你按键存取一个值、删除键、检测一个键是否绑定了值。因此（并且也没有其他内建的替代方式了）过去我们一直都把对象当成 Maps 使用。不过他们还是有一些重要的区别，在以下的情况下用 Map 会更好：</p> <ol style="list-style-type: none">(1) 一个 Object 的键只能是字符串或者 Symbols，但一个 Map 的键可以是任意值，包括函数、对象、基本类型。(2) Map 中的键值是有序的，而添加到对象中的键则不是。因此，当对它进行遍历时，Map 对象是按插入的顺序返回键值。(3) 你可以通过 size 属性直接获取一个 Map 的键值对个数，而 Object 的键值对个数只能手动计算。(4) Map 可直接进行迭代，而 Object 的迭代需要先获取它的键数组，然后再进行迭代。(5) Object 都有自己的原型，原型链上的键名有可能和你自己在对象上的设置的键名产生冲突。虽然 ES5 开始可以用 map = Object.create(null) 来创建一个没有原型的对象，但是这种用法不太常见。	

(6) Map 在涉及频繁增删键值对的场景下会有些性能优势。

Map 可以通过 `Map.length` 来获取 Map 对象的键/值对的数量。

一些常用的方法：

Map.prototype.set(key, value)

设置 Map 对象中键的值。返回该 Map 对象。

Map.prototype.get(key)

返回键对应的值，如果不存在，则返回 `undefined`。

Map.prototype.has(key)

返回一个布尔值，表示 Map 实例是否包含键对应的值。

Map.prototype.keys()

返回一个新的 Iterator 对象，它按插入顺序包含了 Map 对象中每个元素的键。

Map.prototype.values()

返回一个新的 Iterator 对象，它按插入顺序包含了 Map 对象中每个元素的值。

Map.prototype.clear()

移除 Map 对象的所有键/值对。

Map.prototype.delete(key)

如果 Map 对象中存在该元素，则移除它并返回 `true`；否则如果该元素不存在则返回 `false`

本周任务：把 titan 项目的订餐页面重构，每天的餐时变量类型用 Map 存储。