在之前的概念介绍中简单的介绍了一下后台脚本，本节来具体说明一下后台脚本，脚本猫系列的开发教程也是围绕着后台脚本来进行的，关于油猴脚本的可以，可以看【油猴脚本开发指南】，也可以使用脚本猫来运行。

后台脚本简而言之就是在后台中运行的脚本，相比于普通的油猴脚本，它不需要打开浏览器运行，可以由自己控制运行时间。另外开启下面两项，可以让电脑开机的时候就运行脚本。（以Edge为例）



后台脚本有两种类型：普通后台脚本与定时脚本。

**普通后台脚本**

普通后台脚本使用@background进行声明，运行时间为脚本开启时运行。在浏览器打开时，如果脚本是开启状态，也会开始运行。当然也可以手动在面板上点击运行一次进行运行。如果脚本的生命周期与浏览器的生命周期一致，可以使用本脚本类型。

例如使用CAT\_proxy实现代理的脚本；使用GM\_addValueChangeListener实现的与普通油猴脚本进行交互的脚本。

**定时脚本**

定时脚本是一种特殊的后台脚本，如果你有定时执行的需求，可以直接使用@crontab去实现定时执行的需求，与普通后台脚本不同，定时脚本只会在crontab所定义的时间上运行，并且生命周期应该是执行一次脚本，时间不宜太长。crontab表达式，可以去下面的网站测试。

[crontab在线测试](https://tool.lu/crontab/)

**Crontab 例子**

//@crontab \* \* \* \* \* \* 每秒运行一次

//@crontab \* \* \* \* \* 每分钟运行一次

//@crontab 0 \*/6 \* \* \* 每6小时的0分时执行一次

//@crontab 15 \*/6 \* \* \* 每6小时的15分时执行一次

//@crontab \* once \* \* \* 每小时运行一次

//@crontab \* \* once \* \* 每天运行一次

//@crontab \* 10 once \* \* 每天10点检测运行一次

//@crontab \* 10-23 once \* \* 每天10点后检测运行一次

//@crontab \* once 13 \* \* 每个月13号中的每小时最多运行一次

copy

**写法**

后台脚本必须使用Promise的形式，这样才能方便管理器进行管理，不遵循此规则的脚本可能会无法正常运行。除此之外，几乎与普通的油猴脚本没什么不同。因为后台脚本是在后台运行，所有针对window、dom的操作是无效的。另外为了满足后台脚本的一些交互，增强和扩展了一些API，具体可以看官方文档，教程后续也会继续讲解。

// ==UserScript==

// @name 后台脚本

// @namespace wyz

// @version 1.0.0

// @author wyz

// @background

// ==/UserScript==

return new Promise((resolve, reject) => {

if (Math.round((Math.random() \* 10) % 2)) {

resolve("ok");// 执行成功

} else {

reject("error");// 执行失败,并返回错误原因

}

});

// ==UserScript==

// @name 每天运行一次的定时脚本

// @namespace wyz

// @version 1.0.0

// @author wyz

// @crontab \* \* once \* \*

// ==/UserScript==

return new Promise((resolve, reject) => {

if (Math.round((Math.random() \* 10) % 2)) {

resolve("ok");// 执行成功

} else {

reject("error");// 执行失败,并返回错误原因

}

});