

link: null
title: 珠峰架构师成长计划
description: 使用JavaScript语言编写一个可执行脚本
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=59 sentences=42, words=271

1. yargs

1. 可执行脚本

使用JavaScript语言编写一个可执行脚本

```
console.log('hello');
```

然后修改权限

```
chmod 755 hello
```

执行脚本

```
./hello  
hello
```

如果想把路径去掉可以把hello的路径加入环境变量PATH。但是，另一种更好的做法是在当前目录下创建一个package.json

```
{  
  name: 'hello',  
  "bin": {  
    "hello": "hello"  
  }  
}
```

然后npm link

```
npm link
```

在执行hello就不用输入路径了

```
hello
```

2. 命令行参数的原始写法

命令行参数可以用系统变量 process.env获取

```
console.log('hello ', process.argv[2]);
```

```
hello zfpz
```

```
process.env = ['node', 'hello', 'zfpz']
```

3. 新建进程

脚本可以通过 child_process模块新建子进程，从而执行 linux命令

```
let name = process.argv[2];  
let {exec} = require('child_process');  
let child = exec('echo hello ' + name, (err, stdout, stderr) => {  
  if (err) throw err;  
  console.info(stdout);  
});
```

```
hello zfpz
```

4. yargs

yargs模块能够解决如何处理命令行参数。

```
npm install yargs --save
```

yargs模块提供了argv对象，用来读取命令行参数

```
let argv = require('yargs').argv;  
console.log('hello ', argv.name);
```

```
hello --name=zfpz  
hello --name zfpz
```

```
process.argv
```

```
[ '/usr/local/bin/node', '/usr/local/bin/hello4', '--name=zfpz' ]
```

```
argv
```

```
{  
  name: 'zfpz',  
}
```

```
let argv = require('yargs')  
  .alias('n', 'name')  
  .argv
```

```
hello -n zfpz  
hello --name zfpz
```

argv对象有一个下划线属性，可以获取非连词线开头的参数

```
let argv = require('yargs').argv
console.log('hello ',argv.n);
console.log(argv._);
```

```
hello A -n zfpX B C
hello zfpX ['A','B','C']
```

- demand 是否必选
- default 默认值
- describe 提示

```
let argv = require('yargs')
  .demand(['n'])
  .default({n:'zfpX'})
  .describe({n:'你的名字'})
  .argv;
console.log('hello ',argv.n);
```

这个代表n不能省略，默认值为zfpX,并给出提示

option方法允许将所有的配置写入配置对象

```
let argv = require('yargs')
  .option('n',{
    alias:'name',
    demand:true,
    default:'zfpX',
    describe:'请输入你的名字',
    type:'string',
    boolean:true
  }).argv
console.log('hello',argv.n);
```

有时候，某些参数不需要，只起到开关作用。可以用boolean指定这些参数返回布尔值

```
let argv = require('yargs')
  .boolean(['private'])
  .argv
console.log('hello',argv.n);
```

参数private总是返回一个布尔值

```
hello
false
hello -private
true
hello -private zfpX
true
```

yargs模块提供以下方法，生成帮助信息

- usage 用法格式
- example 提供例子
- help 显示帮助信息
- epillog 出现在帮助信息的结尾

```
let argv = require('argv')
  .option('n',{
    alias:'name',
    demand:true,
    default:'tom',
    describe:'你的名字',
    type:'string'
  })
  .usage('Usage: hello [options]')
  .example('hello -n zfpX','say hello to zfpX')
  .help('h')
  .alias('h','help')
  .epilog('copyright 2018')
  .argv
```