

global——全局

process——控制当前 Node.js 进程

`process.argv` 属性返回一个数组，这个数组包含了启动Node.js进程时的命令行参数。第一个元素为`process.execPath`。如果需要获取`argv[0]`的值请参见 `process.argv0`。第二个元素为当前执行的JavaScript文件路径。剩余的元素为其他命令行参数。

命令行 进入当前路径 node 5.js a=1 b=2 带参数运行（实现带参数运行当前程序）

process.execPath

`process.execPath` 属性，返回启动Node.js进程的可执行文件所在的绝对路径。

```
'/usr/local/bin/node'
```

process.env

process.env属性返回一个包含用户环境信息的对象 See `environ(7)`.

例如这样的对象：

```
{
  TERM: 'xterm-256color',
  SHELL: '/usr/local/bin/bash',
  USER: 'maciej',
  PATH: '~/.bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin',
  PWD: '/Users/maciej',
  EDITOR: 'vim',
  SHLVL: '1',
  HOME: '/Users/maciej',
  LOGNAME: 'maciej',
  _: '/usr/local/bin/node'
}
```

可以修改这个对象，但是下面例子的做法是不会生效的：

```
$ node -e 'process.env.foo = "bar"' && echo $foo
```

下面的做法会生效：

```
process.env.foo = 'bar';  
console.log(process.env.foo);
```

在`process.env`中新增一个属性，会将属性值转换成字符串

例如：

```
process.env.test = null;  
console.log(process.env.test);  
// => 'null'  
  
process.env.test = undefined;  
console.log(process.env.test);  
// => 'undefined'
```

用 `delete`从`process.env`中删除一个属性

例如：

```
process.env.TEST = 1;  
delete process.env.TEST;  
console.log(process.env.TEST);  
// => undefined
```

在Windows系统下，环境变量是不区分大小写的

例如：

```
process.env.TEST = 1;  
console.log(process.env.test);  
// => 1
```

process.pid (node进程的pid)

`process.pid`属性返回进程的PID。

```
console.log(`This process is pid ${process.pid}`);
```

process.pid(当前进程的名称)

process.arch

属性返回一个标识Node.js在其上运行的处理器架构的字符串。例如 'arm', 'ia32', or 'x64'

NodeJS

- global
 - process 对象
 - platform
 - 返回当前操作系统平台
 - cwd()
 - 返回当前进程的工作目录
 - chdir(directory)
 - 改变当前进程的工作目录
 - memoryUsage()
 - 返回node进程的内存使用情况，单位是byte
 - exit(code)
 - 退出
 - kill(pid)
 - 向进程发送信息

妙味课