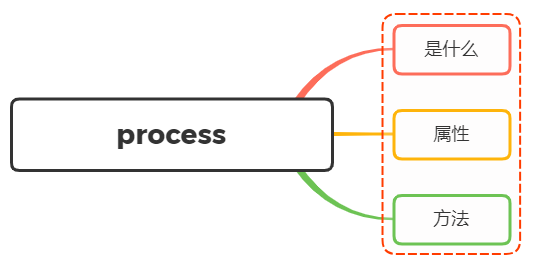
# 面试官：说说对 Node 中的 process 的理解？有哪些常用方法？



## 一、是什么

process 对象是一个全局变量，提供了有关当前 Node.js进程的信息并对其进行控制，作为一个全局变量

我们都知道，进程计算机系统进行资源分配和调度的基本单位，是操作系统结构的基础，是线程的容器

当我们启动一个js文件，实际就是开启了一个服务进程，每个进程都拥有自己的独立空间地址、数据栈，像另一个进程无法访问当前进程的变量、数据结构，只有数据通信后，进程之间才可以数据共享

由于JavaScript是一个单线程语言，所以通过node xxx启动一个文件后，只有一条主线程

## 二、属性与方法

关于process常见的属性有如下：

* process.env：环境变量，例如通过 `process.env.NODE\_ENV 获取不同环境项目配置信息
* process.nextTick：这个在谈及 EventLoop 时经常为会提到
* process.pid：获取当前进程id
* process.ppid：当前进程对应的父进程
* process.cwd()：获取当前进程工作目录，
* process.platform：获取当前进程运行的操作系统平台
* process.uptime()：当前进程已运行时间，例如：pm2 守护进程的 uptime 值
* 进程事件： process.on(‘uncaughtException’,cb) 捕获异常信息、 process.on(‘exit’,cb）进程推出监听
* 三个标准流： process.stdout 标准输出、 process.stdin 标准输入、 process.stderr 标准错误输出
* process.title 指定进程名称，有的时候需要给进程指定一个名称

下面再稍微介绍下某些方法的使用：

### process.cwd()

返回当前 Node进程执行的目录

一个Node 模块 A 通过 NPM 发布，项目 B 中使用了模块 A。在 A 中需要操作 B 项目下的文件时，就可以用 process.cwd() 来获取 B 项目的路径

### process.argv

在终端通过 Node 执行命令的时候，通过 process.argv 可以获取传入的命令行参数，返回值是一个数组：

* 0: Node 路径（一般用不到，直接忽略）
* 1: 被执行的 JS 文件路径（一般用不到，直接忽略）
* 2~n: 真实传入命令的参数

所以，我们只要从 process.argv[2] 开始获取就好了

const args = process.argv.slice(2);

### process.env

返回一个对象，存储当前环境相关的所有信息，一般很少直接用到。

一般我们会在 process.env 上挂载一些变量标识当前的环境。比如最常见的用 process.env.NODE\_ENV 区分 development 和 production

在 vue-cli 的源码中也经常会看到 process.env.VUE\_CLI\_DEBUG 标识当前是不是 DEBUG 模式

### process.nextTick()

我们知道NodeJs是基于事件轮询，在这个过程中，同一时间只会处理一件事情

在这种处理模式下，process.nextTick()就是定义出一个动作，并且让这个动作在下一个事件轮询的时间点上执行

例如下面例子将一个foo函数在下一个时间点调用

function foo() {  
 console.error('foo');  
}  
  
process.nextTick(foo);  
console.error('bar');

输出结果为bar、foo

虽然下述方式也能实现同样效果：

setTimeout(foo, 0);  
console.log('bar');

两者区别在于：

* process.nextTick()会在这一次event loop的call stack清空后（下一次event loop开始前）再调用callback
* setTimeout()是并不知道什么时候call stack清空的，所以何时调用callback函数是不确定的

### 参考文献

* http://nodejs.cn/api/process.html
* https://vue3js.cn/interview/