包管理工具

# 1.基本概念

|  |
| --- |
|  |

# 2.npm的使用

|  |
| --- |
|  |

## npm init初始化注意事项

|  |
| --- |
|  |

## npm搜索包

|  |
| --- |
|  |

## require导入npm模块的流程

|  |
| --- |
|  |

## 下面这三种写法都是可以的，不过建议使用最上面的写法，因为它很灵活，会一级一级往上找，直到找到根目录

|  |
| --- |
|  |

# 3.生产依赖和开发依赖

|  |
| --- |
|  |

## 小案例

|  |
| --- |
|  |

# 4.npm全局安装

|  |
| --- |
|  |

# 5.设置powershell运行执行脚本

|  |
| --- |
|  |

# 6.npm安装所有依赖

|  |
| --- |
|  |

# 7.npm安装指定版本的包和npm删除包

|  |
| --- |
|  |

# 8.npm配置命令别名

|  |
| --- |
|  |

## 小案例

|  |
| --- |
|  |

## 也就是说在一个目录里面安装的node\_modules文件夹里面的包可以个这个文件夹的所有子文件夹使用

# 9.cnpm

|  |
| --- |
|  |

# 10.npm配置淘宝镜像

|  |
| --- |
|  |

# 11.yarn

## 1.特点

|  |
| --- |
|  |

## 2.yarn常用命令

|  |
| --- |
|  |

## 需要把这个安装目录添加到环境变量

## 补充内容

|  |
| --- |
|  |

## 其实nodejs17版本以上自带yarn，需要执行命令corepack enable来启用yarn就可以使用

# 12.用npm发布一个包

|  |
| --- |
|  |

## 首先需要到[npm | Home (npmjs.com)](https://www.npmjs.com/)注册一个账号，我的是

|  |
| --- |
| username：kennycaipw: name+year+month |

### 端口终端窗口定位到项目路径，输入：npm adduser，注意add和user中没有空格，然后按回车就可以端口一个登录页面，输入用户名密码就可以登录了

## 我们来实际操作一下

## 1.新建一个文件夹：mypkg，然后用npm init初始化，注意包名称不能含有test否则会被认为是垃圾包

|  |
| --- |
|  |

## 2.在这个包里面新建一个index.js文件，注意需要和package.json的main属性指定的文件夹一致

|  |
| --- |
| function anySum(...params) {      let sum = 0      for(let i of params){          sum +=i      }      return sum  }  // console.log(anySum(1,2,3,4,5));  module.exports = anySum |

## 3.在终端输入：npm adduser，然后回车，在打开的页面中输入用户名密码登录，登录后终端也有确认信息

|  |
| --- |
|  |

## 4.在终端输入nrm use npm，确保你使用的npm源是npmjs.com的，否则发布不上，如果提示没有安装nrm使用命令：npm install nrm -g安装

|  |
| --- |
|  |

## 5.发布包：在终端输入npm publish，出现错误，可能这个名称已经存在，我们修改一下名字，继续发布

|  |
| --- |
|  |
|  |

## 发布成功

|  |
| --- |
|  |
|  |

## 6.然后我们再新建一个文件夹初始化package.json,然后下载我们的包

|  |
| --- |
|  |
|  |
|  |

### 效果

|  |
| --- |
|  |

# 13. npm更新与删除包

|  |
| --- |
|  |

### 如果用上面的命令无法删除包，可以使用npm unpublish --force命令

## 如：我们可以给myutil下面的index.js添加一个方法，然后测试然后修改版本号，自己导出的内容，然后再发布：npm publish

|  |
| --- |
| function anySum(...params) {      let sum = 0      for(let i of params){          sum +=i      }      return sum  }  function anyMax(...params) {      let max = params[0]      for(let i of params){         if(i>max){           max = i         }      }      return max  }  // console.log(anySum(1,2,3,4,5));  module.exports = {      anySum,      anyMax  } |
|  |
|  |

## 发布成功

|  |
| --- |
|  |
|  |

# 14.包管理工具扩展，其他编程语言的包管理工具

|  |
| --- |
|  |

# 15.nvm：nodejs版本管理工具

## 安装nvm

### 用这个网址

|  |
| --- |
|  |

### 或者：

### 打开GitHub，搜索nvm,找到这个windows版本，mac或者Linux使用另外的版本

|  |
| --- |
|  |

### 往下滚动内容，可以看到一个下载链接

|  |
| --- |
|  |

### 点击这个链接，可以进入一个下载界面，往下滚动可以看到文件列表

|  |
| --- |
|  |

## 我选择nvm-setup.exe,安装路径：C:\Program Files\nvm

|  |
| --- |
|  |

## 用nvm安装nodejs **18.18.0**

|  |
| --- |
| nvm install 18.18.0 |

### 安装完后，测试，注意需要先运行nvm use 18.18.0否则说找不到

### 然后就可以检查nodejs的版本了

|  |
| --- |
|  |

### 需要将nodejs的安装路径设置到Path环境变量，可以先新建一个NODE\_HOME环境变量，然后把它设置到Path中

|  |
| --- |
|  |
|  |

## nvm常用命令

|  |
| --- |
| 安装 node # 安装最新版本 Node  nvm install node # "node" is an alias for the **latest version**  # 安装指定版本 Node  nvm install 6.14.4 # or 10.10.0, 8.9.1, etc  # 安装最新LTS(Long-term Support)版本Node  **nvm install --lts** 查看安装node # 查看远程 Node 所有版本  nvm ls-remote  # 查看远程 Node 所有 LTS 版本  nvm ls-remote --lts  # 查看本地 Node 版本  **nvm ls**  # 查看当前使用 Node 版本  nvm current  # 查看 node 安装位置  nvm which 14 使用 node 版本 # 当前 shell 下使用某个 Node 版本  nvm use <node的版本>  # 指定全局默认版本  nvm alias default <node版本号> 删除 node nvm uninstall <node 版本号/别名> |
|  |

##### 注意需要先安装nvm，然后利用nvm安装node然后node就会在nvm里面