

WEBTECHNOLOGIES

INTRODUCTION TO NPM

WHAT IS NPM?

- Formerly called *Node Package Manager*
- Consists of three components
 - website: www.npmjs.com
 - description of available packages
 - registry
 - contains all packages which are described at website
 - client app
 - Command Line Interface (CLI) for managing packages
 - automatically installed with *Node.js*

WHAT CAN YOU DO WITH NPM?

- Manage your project
 - author
 - version number
 - ...
- Manage your project dependencies
 - install package
 - uninstall package
 - update package
- Publish your own project to *npm* registry

PROJECT DEPENDENCIES

- Problems
 - How to find out which packages a program needs?
 - How to find out if packages should be updated?
 - How to reinstall packages on a target machine?
- Solution: `package.json`

CREATE PACKAGE.JSON

- Open a shell in empty folder `npm-demo`
- Run: `npm init`
 - Answer questions
 - Or just hit enter to accept default
- Result: Created file `package.json`

PACKAGE.JSON

```
{  
  "name": "npm-demo",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "mholzmann",  
  "license": "ISC",  
  "dependencies": {}  
}
```

INSTALL PACKAGE CHALK

- Run: `npm install chalk`
 - Installs package to `node_modules/chalk`
 - Check installed version

```
michael@DESKTOP-L8R0B7R:~/NodeProjects/npm-demo$ npm list --depth=0
npm-demo@1.0.0 /home/michael/NodeProjects/npm-demo
└─ chalk@3.0.0
```

- Check new dependency in `package.json`

```
"dependencies": {
  "chalk": "^3.0.0"
}
```

EXCURSE: SEMANTIC VERSIONING

- Version number MAJOR.MINOR.PATCH
- Increment ...
 - MAJOR when you make incompatible API changes
 - MINOR when you add functionality in a backwards compatible manner
 - PATCH when you make backwards compatible bug fixes
- More details at semver.org

VERSIONS IN PACKAGE.JSON

- 2.3.0
 - Means: Take version 2.3.0
- ~2.3.0
 - Means: Take most recent 2.3.x version
- ^2.3.0
 - Means: Take most recent 2.x.x version

IMPORT INSTALLED PACKAGE

- Create `app.js` with following content

```
const chalk = require('chalk');  
  
console.log(chalk.red('Hello world!'));
```

- Run: `node app.js`

INSTALL PACKAGE COWSAY

- **Run:** `npm install cowsay`
 - Installs package to `node_modules/cowsay`
 - Installs executables to `node_modules/.bin`
- **Run:** `npx cowsay "Hello world!"`
 - Executes `cowsay` in `node_modules/.bin`

NODE_MODULES FOLDER

- Stores all locally installed packages
 - ... and their dependencies
 - ... and the dependencies of the dependencies...
- Can contain large number of files
 - Don't copy it from machine to machine
 - Never (!!!) add it to *Git* repository
 - Therefore add file [.gitignore](#)