

# First Serve

Chrome File -> Open File

- IS rendering
- IS NOT a "served" html file

# Installing via NPM

NPM is

- N - Node
- P - Package
- M - Manager

Manages the installation and update of tools and libraries that involve NodeJS

# Global vs Local

"Global" install

- can run from any directory
- for system tools not specific to a project

"Local" install

- into a specific project
- intended only for that project
- when the version of the tool/library is specific to project

# When Global

Can I use a different version of this, or even have a different program for that functionality?

- Global is fine
- Examples: Editors, browsers, spellcheckers

# When Local

Do all users/developers need the same version of this to have the same experience?

- Local is appropriate

Do users need this to run/use the application, but it's not something I can assume they have installed?

- Local is probably appropriate

# This time we are global

We will be installing a simple, general use webserver

It is not specific to the test code we'll be using

We will use it with many test codes

We don't expect others to use this at all

- Global install
- use `npx` so we don't need to alter PATH and permissions

# Installing Serve

(Don't do this, we'll use `npx`)

```
npm install -g serve
```

- `npm` is running npm
- `install` is passed to npm program to use
- `-g` is a "switch", alters the command
- `serve` tells `npm install` what to install
- Because this is a global install, doesn't matter where you are when you run it
- You DON'T want to run this as `sudo`

# NPX

`npx` node package executor

- 2 main uses
- Runs an installed local node-based binary
  - from `./node_modules/bin/`
    - if not be in your OS `PATH` when you run the command it won't find the binary
  - e.g. after `npm install cat-me`, `cat-me` gives `command not found`, but `npx cat-me` will work
- Installs and runs binaries from a global cache
  - e.g. `npx serve` will run `serve`
  - installs in global cache if needed



# Using NPX

`npx serve`

- Installs `serve` AND runs it
- npx will install the first time and ask for confirmation
- No installation or confirmation on later runs
- npx doesn't care what directory you are
- BUT `serve` does care where you are at command line

# Running serve

```
npx serve
```

This will run a webserver that serves static files where the document root is the current directory

# Wut?

```
run a webserver
```

- A program that returns ("serves") web pages to incoming web (HTTP) requests
- Your machine can be reached as `127.0.0.1` using DNS name of `localhost`
  - These mean "me"
  - Globally true, not just with this webserver
- It will default to port 5000 if available
  - This is specific to this webserver
- "port" is a "subaddress" on the machine. Servers "listen" and "send" traffic on a port.

# Wut? - Static Files

```
that serves static files
```

A "Static" file is a file that exists as a file.

- The webserver will send the contents of the file from the filesystem

The opposite of a "static" file is a "dynamic" file.

A dynamic file doesn't exist as a file

- generated in response to the request

Like someone telling you the contents of an existing note (static) vs telling you something new on the spot (dynamic).

# Wut? - Document Root

```
the document root is the current directory
```

When the webserver needs access to static files, it needs to find them.

You don't want the webserver giving access to every file on the computer.

The "document root" is the root of the directory tree of files it can offer access to.

# That is wut

Running the command `npx serve` will run a webserver that serves static files where the document root is the current directory

(Using a program that relies on NodeJS, installed using npm)

# Defaults

Run `npx serve`; Browse `http://localhost:5000/`

- We each get only our machine as localhost
- Other IP address is based on current network
- Many places (like NEU) block web traffic INTO personal machines

If no `index.html` file, `serve` shows a file listing

- Default `index.html` is common, but not universal.
  - Up to *webserver* to decide this
- Most production servers won't show a directory

# Summary - NPM

- NPM installs and manages tools and libraries for NodeJS
- You can install locally or globally
  - Global installs for tools not part of an app
  - Local installs for anything part of an app
- You should **not** install as sudo/root/administrator
- `npx` can install packages into a global cache
- `npx` can run package binaries regardless of OS `PATH`



# Summary - Webservers

- Webservers listen to a `port`
- Webservers serve `static` and `dynamic` "assets" (files)
- `127.0.0.1/localhost` means "me"/"yourself"
- `index.html` is a common default if no file in URL for a static asset
- Development servers may show a directory listing if no `index.html`
- Static assets are served from a "document root" that act as the root for url paths for static assets