**Git a Web Developer Job: Mastering the Modern Workflow | Udemy**

**1. - Git**

1. What is Git?

* World’s most popular version control system
* Git helps us manage our project files
* *History* - Git tracks changes
* *Collaboration* – With this collaboration you can continue working on a file while a partner is working on the same one. Git merges the changes from everyone
* *Feature branches* – it allows you to work on multiple features at once and to update the project just with the finished tasks. After you complete other branch task you can merge it all.

1. Why should we learn Git?

* Industry Standard – almost a most

Vocabulary:

* Project 🡪 Reposotory (repo): where Git saves all of the data, history, changes that it tracks.
* Working directory: Folding location on computer where the project lives (repository)
* Staging 🡪 before commiting we need to stage our changes. Preparing or getting ready. Control what gets committed
* Commit 🡪 Git’s way of “saving”

Commands:

* git status
* git add –A (staged)
* git reset file (unstage)
* git commit –m ‘msg’
* git checkout -- . 🡪 restores back to the files of the last commit (local)
* git push nameProject 🡪 to commit to GitHub’s server
* git pull nameProject 🡪 to pull the latest changes from GitHub’s server to your local to work with the most updated files
* git remote –v 🡪 dirección donde se guardarían los archivos con un push
* git remote set-url origin https://….. 🡪 para cambiar la dirección en donde se haría el push
* **git push origin master** 
  + origin: la dirección a la que apunta el push
  + master es un branch (luego veremos eso)

**2. – Node.js and NPM**

* While we build our website we want automation and organization.
  + Automation – would save us time
    - Take a task and find a way to have your computer to do it for you.
    - Automatic autorefresh browser
    - CSS Autoprefixers
  + Organization – would save our sanity
    - CSS Organization
    - Package Management
* **Node.js**
  + It is a JavaScript runtime that let us do just about anything with javascript
  + Camp #1: Install Node.js on a server
    - Taking requests, retrieving data from databases and sending it back to the users.
  + **Camp #2: Install Node.js on your personal computer**
    - Using it to create our own robot assistant
    - We are using it to automate all sorts of web development tasks that can save us a ton of time
  + In the terminal:
    - node –v 🡪 to check if it is installed
    - node test.js 🡪 to execute the code
* **NPM (Node Package Manager)**
  + Centralized place where developers share their code with the world for free. It is a place where bunch of code lives
    - To create better work
    - Avoid reinventing the wheel
  + Package Management (used as a one stop shop that we need)
    - Used to obtain all the listed libraries or APIs that you need automatically. It also checks for the new versions so you don’t use an outdated or with bugs code.
  + Types of files/packages we hope to find on NPM
    - Node Packages
      * Automation
      * Build tools
      * Server tasks
    - Other things
      * jQuery
      * Bootstrap
      * Lodash
      * Normalize.css
  + Commands:
    - npm install jquery --save
      * with --save 🡪 saves our package to our recipe file
    - npm init 🡪 to create the package.json (ingredients list)
      * package.json 🡪 can be used to keep track of which packages we are using
    - npm install
      * will download all the dependencies that are listed on our package.json
      * So if you erase by accident your node-modules folder you can recover them with this command as long as you have your package.json
  + You can search for packages at npmjs.com or at google that do what you need in your application
  + The folder node-modules is ignored by git as it is by default in the .gitignore file 🡪 this is not a problem because as the dependencies are already stated in the package.json npm would download them when working with the project

**3. - Gulp**