**Introduction to JavaScript**

**What is JS?**

JavaScript was initially created to “make web pages alive”.

The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads.

Scripts are provided and executed as plain text. They don’t need special preparation or compilation to run.

Today, JavaScript can execute not only in the browser, but also on the server, or actually on any device that has a special program called the JavaScript engine.  
The browser has an embedded engine sometimes called a “JavaScript virtual machine”.

**More here:**[**https://javascript.info/intro**](https://javascript.info/intro)

**Setup working environment**

**Choose a code editor**

* Visual Studio Code: <https://code.visualstudio.com/> (preferable option)
* Atom: <https://atom.io/>
* Sublime Text: <https://www.sublimetext.com/>

**Optional (Setup useful plugins for VS Code):**

* All Autocomplete
* Auto Close Tag
* Auto Complete Tag
* Auto Rename Tag
* Beautify
* Code Runner
* GitLens — Git supercharged
* Import Cost
* JavaScript (ES6) code snippets
* jshint
* Live Server
* vscode-icons
* ESLint

**Download and Install Git:**

**Git:** <https://git-scm.com/downloads>

Git Bash enables to execute Linux commands on Windows machines

**Download and install** [NodeJS](https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/) **and** [NPM](https://docs.npmjs.com/about-npm/)**:**

**What is NodeJS?**

The Node.js run-time environment includes everything you need to execute a program written in JavaScript.

**NPM:** is the world’s largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.

**NPM can be used to:**

* Adapt packages of code for your apps or incorporate packages as they are.
* Download standalone tools you can use right away.
* Run packages without downloading using npx.
* Share code with any npm user, anywhere.
* Restrict code to specific developers.
* Create Orgs (organizations) to coordinate package maintenance, coding, and developers.
* Form virtual teams by using Orgs.
* Manage multiple versions of code and code dependencies.
* Update applications easily when underlying code is updated.
* Discover multiple ways to solve the same puzzle.
* Find other developers who are working on similar problems and projects.

**References:**

* [NodeJS](https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/)
* [NPM](https://docs.npmjs.com/about-npm/)
* [package.json](https://jsbin.com/?js,console,output)

**Download**: <https://nodejs.org/en/download/>

**After the install:**

* Open CMD or Git Bash or use Terminal in VS code
* Execute the command *'node - v'* to review the installed version

**Create a working directory:**

1. Create a folder
2. Open CMD or Git Bash or use Terminal in VS code
3. Navigate to the created folder
4. Execute

npm init

1. Follow the instructions in the console
2. Open and review **[package.json](https://jsbin.com/?js,console,output" \o "https://jsbin.com/?js,console,output" \t "_blank)** file

**Git. Fundamentals**

**References:**

* [Video](https://videoportal.epam.com/video/VJwm8Yov)
* [Documentation](https://git-scm.com/book/en/v2)
* [Git. How to](https://githowto.com/ru)
* [Learn Git](https://learngitbranching.js.org/)
* [GitHub](https://github.com/)

**How to create a working repository:**

1. Navigate to any folder
2. Open CMD or Git Bash or use Terminal in VS code
3. Execute *'git init'* command // .git hidden folder appear in your directory
4. Execute *git config --local user.name "YourName"*
5. Execute *git config --local user.email example@gmail.com*
6. If you set up Git for the first time you can set the setting globally. To do that just replace *--local* by *--global*

**Basic Commands**

[*git status*](https://www.atlassian.com/git/tutorials/inspecting-a-repository)*:*displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven’t, and which files aren’t being tracked by Git.

[*git Add*](https://git-scm.com/docs/git-add)*-A*: Add file contents to the index. This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit.

[*git commit*](https://git-scm.com/docs/git-commit)*-a -m"any message"*: Record changes to the repository. Create a new commit containing the current contents of the index and the given log message describing the changes.

[*git log:*](https://git-scm.com/docs/git-log)Show commit logs

[*git push:*](https://git-scm.com/docs/git-push)Update remote refs along with associated objects