

1. The Dev Workflow

[Github Repository](#). | [Vimeo Video Recording](#)

Topics to cover

- [X] Curriculum overview
- [X] Problem solving tips
- [X] Writing code incrementally
- [X] Basic git workflow (init -> add -> commit -> push)

Problem solving tips

Before we jump into coding, it is always better to **understand the problem at hand** and plan out how we are going to tackle the problem.

Dividing the problem into smaller problems, and then try and understand each one of them is much better than just attacking the overall problem as it is.

Try to use comments in javascript as much as you need to document your thought process and enlist the steps you will be taking:

```
/**
  PROBLEM:

  Write a Node program that takes in an unlimited number of commandline
  arguments,
  goes through each and prints out the sum of them.

  If any argument is NOT a whole number, skip it.
  Do not support negative numbers.
  If any argument is not a number, output an error message.
  We need at least two arguments.
*/

// 1) how to take in an unlimited number of command line arguments?

// 2) how can we go through each one of command line arguments?

// 3) how can we print the sum of them?...
```

Try to use google as much as you can to read and find examples and explanations that might be helpful to achieve your goals.

It is not easy to filter out what you need amongst all the information that exists out there, so this is another important ability that you will be developing over time.

Writing code incrementally

Don't try to solve everything at once. Try adding some code, test it out, use `console.log()` to check everything that you are curious about or don't understand what is doing.

As you take each step and start doing progress, don't forget to always use git and **commit your progress**, as this will generate a history that you will be able to revisit at any time in the future.

```

const arguments = process.argv
let sum = 0;

if (arguments.length >= 4){
  let countPositive = 0;
  for (let i = 2; i < arguments.length; i++){
    if (! isNaN(Number(arguments[i]))){ //hello
      if (Number(arguments[i]) > 0){
        countPositive ++;
        sum += Number(arguments[i]);
      }
    }
  }

  if (countPositive === 0){
    console.log("Error, no numbers found!");
  }
  else if (countPositive === 1){
    console.log("Error, you need to provide at least 2 numerical
arguments!")
  }
  else{
    console.log("The sum is " + sum);
  }
}
else{
  console.log("Error, you need to provide at least 2 arguments!")
}

```

Also, if something breaks or you need to do a rollback, you can always go back to previous git commits.

Basic git workflow

In order to use Git, [the Version Control System](#) we can check if Git is installed by running the following command on terminal:

```

$ git --version
# should output git's installed version

```

We can transform any directory we are working with into a **Git Repository**.

To do so, we have to position ourselves in the working directory, using the terminal. Then execute the following command:

```
$ git init
```

Once we have successfully initialized a git repository, we can check the status of it by running:

```
$ git status
```

Note: You can run this command as many times as you need to understand what is happening in your local repository

Now we are ready to start with the git workflow! On a regular basis, you will be modifying, adding or deleting files in your directory/project. Each time you want to save these changes into a *checkpoint* (A.K.A. *commit*) we must run the following command:

```
$ git add .  
# by using the ".", we are adding every change  
# to be considered into the commit
```

Once we have every change added into the *staging area*, we can run `git status` once again just to check what's going on in our local repository.

Then, to do the *commit* and save these changes into a checkpoint, we must run the command:

```
$ git commit -m "description of what i'm committing goes here"
```

Finally, we must push our now existing local git repository into **Github**, so we can access it online and share it with anyone. To do so, we must create a new empty repository in [Github](#) and follow the instructions under *"...push an existing repository from the command line"*.

The commands that we must execute in order to push our local repo into github should look like this:

```
$ git remote add origin git@github.com:Masavi/my-repo  
$ git branch -M main  
$ git push -u origin main
```

And that's it! You should now be able to visit your repository directly in Github's website.

Extra content: How to run VS Code from the terminal?

To run vs code from the terminal you can follow this guide:

MacOs/Linux (under the section "launch from the command line")<https://code.visualstudio.com/docs/setup/mac#:~:text=Launching%20from%20the%20command%20line,code'%20command%20in%20PATH%20command.>

Windows:<https://stackoverflow.com/questions/29963617/how-to-call-vs-code-editor-from-terminal-command-line>

[View on Compass](#) | [Leave Feedback](#)

