# I Bootcamps overview

* Bootstrap: 5 Weeks
* React: 5 Weeks
* React Native: 4 Weeks

Requirements

**On sat 20, 2023**

---------------------

* Git, github, Node/NPM
* To be downloaded: from npm, read documentation and download bootstrap 5
* Review and read documentation on VS code. Vscode.com
  + Create a directory -> NUCAMPSITE
  + The active package should b pointing to the directory containing package.json file
  + Bash
  + Zsh
  + Gitbash
  + Powershell
  + Integrated development
  + Npm -v
* Bash commands: cd, ls, mkdir, pwd, cd ../, cd ../../, mkdir js, rmdir js, mkdir css
* Frontend/backend
  + Frontend
  + Backend -> server side
  + Front end -> combination of both front end & back end
* Mobile first design vs responsive web design
  + Responsive web design is a design that can understand and adjust its self to the size and type of the client device/browser (web/mobile, chrome, firefox, …)
* HTML Page setup for Bootstrap
  + Inside head element
  + Inside body element -> order of commands for bootstrap configuration
* Bootstrap.css
* Bootstrap.js
* Breakpoints
  + Media query vs breakpoint
  + Sm, md, lg, xl
  + Documentation: getbootstrap.com
* Bootstrap Grid structure
  + Questions: col-mid6, col-12, practical exercise on bootstrap grid.
* Bootstrap order - \* classes
  + Get this from getbootstrap.com/docs/4.5/layout/ …
  + Order first
  + Order last
  + Order -1, order -12, …
* Bootstrap offset classes
  + Offset - \*
  + Offset -sm-4
  + Col-md-4
  + Col-md-3 .mlmd-auto
  + …
  + Column breaks
* Adding custom styles
* Bootstrap alignment classes
  + Align-item-center
  + Text-lg-center
  + ….

## Asking questions

#### **Asking for Help Effectively**

* Let's go over some best practices when asking for help. These best practices will help you get help faster, and others will be able to help you better if you follow them.
* Learning these will be useful to you beyond Nucamp in your career as a developer, as the need to ask questions will not end after your bootcamp.

#### **The best way to ask for help, which will ensure the quickest response, is to provide all of the necessary details all in one single message or post, like this:**

1. **I need help with X** (state the exercise or assignment youʻre working on, and include a link to the instructions if applicable)
2. **This is the issue/error I am encountering**(be as detailed and specific as possible, and include screenshots of the issue/errors youʻre seeing)
3. **This is what I have already tried**(again, be specific)
4. **Upload the relevant files**(attach the complete file or files, rather than just code snippets or screenshots, as the true problem may be elsewhere in your code)

By laying your question out in this way, it makes it much easier for those who can help you to help when they have a spare moment. Knowing what to ask AND how to ask is the first step to success in your coding journey, and youʻll be able to phrase better questions with practice.  Soon, youʻll be helping others!

#### **Best Practices**

* **DO be descriptive**
  + If the question is about an exercise, challenge, or assignment task, always first state what you're working on:
  + Examples:

I am working on Task 1 of the Week 2 Assignment, and ...

I need help with the Week 1 Anchors exercise ...

* + Include information about what you have already tried while troubleshooting.
  + Be specific about the problem.

Bad example: "It's not working."  
Good example: "When I click on the Submit button, nothing happens."

* **DO mention your instructor so they are aware of your question**
  + Always "mention" your instructor in your message. This will send your instructor a notification. If they are available, they will answer you. If not, then others who are available will still see your question and be able to help you.
  + To mention your instructor, use the @ symbol. In your message, type **@** followed immediately by your instructor's username. There should be no space between the **@** and the name (though there can be spaces in the username).
* **DO attach the code you are working on**
  + Attach the file(s) that you are working on to your message before you send it. You can drag and drop the file(s) into your message.
  + It may be uncomfortable at first to show your work in public, but your potential helpers need to see it. Otherwise, they can only make guesses, and there are *many*different ways code can go wrong. Be kind to your helpers, help them help you!
  + **DON'T** cut and paste your code into the chat, and don't take photos of your code. Attach the actual file that has the code in it.
* **DO attach relevant screenshots**
  + If you're seeing an error message anywhere, and/or your webpage doesn't look right in the browser, take a screenshot and attach it to your message.
  + **DON'T** take screenshots of your code in VS Code. Attach the actual code file instead (the file with the .html, .css, or .js extension).
  + **DON'T** take screenshots using your phone camera. Use your operating system's built-in screenshot tool.
  + If you don't know how to take a screenshot, visit [this page for Windows 10 instructions](https://www.windowscentral.com/how-use-snip-sketch-take-screenshots-windows-10-october-2018-update), and [this page for macOS instructions](https://support.apple.com/en-us/HT201361).

* **DON'T**cross-post the same question in multiple channels
* **DON'T** post course-specific questions in the #general channel

* **DO contribute back by reading and answering questions from other students!**
  + Besides a nice gesture, helping your classmates by reading their code and figuring out their issues helps you level up your own skills by applying critical thinking, practicing code-reading and debugging, and writing your new knowledge into your brain's long-term memory.

## GIT

Check the shell type

>echo $0

* The output should include the word **bash** or **zsh**.
* If it contains the word **zsh**, switch your default terminal to **bash** by entering the following command:

chsh -s /bin/bash

* Close Terminal then open it again. You will see a message that says "**The default interactive shell is now zsh.**" Don't panic! This is normal. Type **echo $0** again to verify that your shell is now **bash**.

#### **Global Git configuration**

* To configure your user name to be used by Git, type the following at the prompt**(include the quotation marks)**:

git config --global user.name "Your Name"

EXAMPLE:

git config --global user.name "Bob Bobbington"

* To configure your email to be used by Git, type the following at the prompt**(leave out the angle brackets):**

git config --global user.email <your email address>

EXAMPLE:

git config --global user.email bob@nucamp.co

* You can check that your configuration is correct by typing the following at the prompt:

git config --global --list

You should be able to see your user.name and user.email in the output.

To rename your Git master branch to main, you must do the following steps:

1. Navigate to your repository in the command line and issue the following commands: - git branch -m master main - git push -u origin main
2. Change your default branch on GitHub by going to your GitHub repository in your browser, and navigate to Settings > Branches and click on the dropdown and switch from master to main and click Update (this will only show if you have two or more branches). The main branch is now your default branch.
3. Update the tracking of the branch from your command line with the following command: - git branch -u origin/main main
4. Delete the old master branch by going to your GitHub repository in your browser, navigate to your branches page, and click the Delete this branch button (the red trash bin icon). Your master branch is now gone.

If someone has a local clone of your repository, they can update their locals by following these steps:

1. Got to the master branch: git checkout master
2. Rename master to main locally: git branch -m master main
3. Get the latest commits from the server: git fetch
4. Remove the link to origin/master: git branch --unset-upstream
5. Add a link to origin/main: git branch -u origin/main
6. Update the default branch to be origin/main: git symbolic-ref refs/remotes/origin/HEAD refs/remotes/origin/main
7. Delete origin/master completely: git branch --remotes -d origin/master

Thanks Scott Hanselman for the instructions, you can read the full article here: [Easily rename your Git default branch from master to main](https://www.hanselman.com/blog/EasilyRenameYourGitDefaultBranchFromMasterToMain.aspx)

## Node and NPM

Initialize NPM

>npm init