

## 2.1.9

## Learn the Git Workflow for a New Feature

Next, we'll create a feature branch for the first issue that we created, complete the work, and then merge the feature into the remote `develop` branch.

The following overview lists the steps we'll follow:

1. Create a local feature branch called `hero-text` from the `develop` branch.
2. Add the HTML for the contact form to that branch.
3. Use Git to push the local feature branch to the remote repo.
4. Checkout to `develop` and merge the local feature branch into the `develop` branch.
5. Pull the remote `develop` branch into the local `develop` branch to check for any differences between them.
6. Merge the local `develop` branch into the remote `develop` branch.

Don't worry if this seems like a lot. We'll review each step in detail. Let's begin!

## Create the Feature Branch

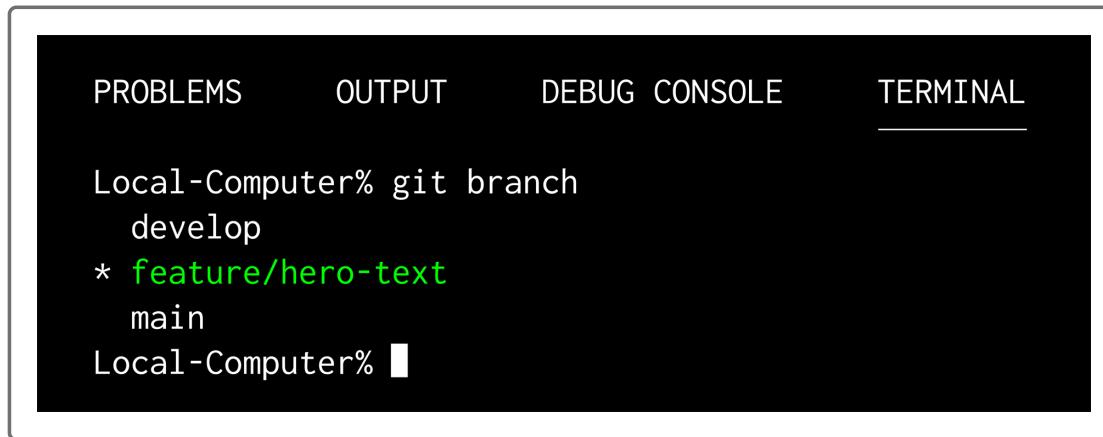
The first step is to create a new feature branch. Remember from the Git workflow introduction that the first step when creating a feature branch is to `checkout` into the `develop` branch. Then you'll create a feature branch and set it as the active branch.

To do this, run the following commands at the command line:

```
git checkout develop  
git checkout -b feature/hero-text
```

This made an exact copy of the `develop` branch, including the newly created `README.md` file. Note that as soon as we start to work and commit in the feature branch, these branches will begin to diverge.

Now confirm that your branch was created. Type `git branch`, and you'll see something like the following image:



A screenshot of a terminal window with a dark background. At the top, there are tabs labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active. The terminal output shows the following text:

```
Local-Computer% git branch  
  develop  
* feature/hero-text  
  main  
Local-Computer%
```

Notice that `git branch` was also used to verify the active branch.

## Update the Hero Section's HTML

Let's take a quick look at the GitHub issue and the mock-up from the design team to see what we need to add to the hero text block.

The following image shows the corresponding GitHub issue and requirements:

**Updates for Hero Section #2**

**lernantino** opened this issue 3 days ago · 0 comments

**lernantino** commented 3 days ago · edited

**Updates for Hero Section**

**Content Requirements**

- Heading: "Start Building Habits."
- Text Body: Lorem ipsum text placeholder
- Download new hero image for hero section

**Questions/suggestions for Design Team**

- What will be the marketing text?

**lernantino** self-assigned this 2 days ago

**lernantino** added the **enhancement** label 2 days ago

**lernantino** changed the title **Hero Section Text Block Updates for Hero Section** 36 seconds ago

**Assignees**: lernantino

**Labels**: enhancement

**Projects**: None yet

**Milestone**: No milestone

**Notifications**: Customize

**Unsubscribe**

You're receiving notifications because you're watching this repository.

1 participant

The following image shows the mock-up:

**RUN BUDDY**

**Start Building Habits.**

seitan heirloom post-ironic pop-up iPhone mlkshk hella selfies fashion axe occupy ready-made put a bird on it messenger bag Wes Anderson Schlitzi plaid Bushwick church-key lo-fi skateboard slow-carb hashtag trust fund Williamsburg biodiesel fixie farm-to-table 8-bit banjo XOXO Banksy chillwave bicycle rights retro chiche tattooed bespoke irony mumblecore Shoreditch deep v polaroid McSweeney's bitters cray gentrify tofu Moma you probably haven't heard of them yr banh mi asymmetrical art party selvedge letterpress High Life.

What We Do   What You Do   Your Trainers   Reach Out

**Get Started Today**

Fill out this form and one of our trainers will schedule a consult.

Your Name  
Your Name

Your Email  
Email Address

Phone Number  
Phone Number

Have you worked out with a trainer before?  Yes  No

I acknowledge that I am at least 18 years of age.

**Get running!**

On your own, practice your HTML skills by fulfilling the GitHub issue requirements, following the mock-up for guidance.

## DEEP DIVE

### DEEP DIVE

We use a mock-up here, but another common tool in the development process is a wireframe. For more information, refer to the [Wikipedia page on website wireframes](https://en.wikipedia.org/wiki/Website_wireframe) ([https://en.wikipedia.org/wiki/Website\\_wireframe](https://en.wikipedia.org/wiki/Website_wireframe)).

When you're done, the HTML should resemble the following code, placed directly in the `<section class="hero">` element:

```
<div class="hero-cta">
  <h2>Start Building Habits.</h2>
  <p>
    seitan heirloom post-ironic pop-up iPhone mlkshk hella selfies fashion a
  </p>
</div>
```

Save your file and then refresh the browser. The updated page should resemble the following image:

The screenshot shows a website for 'RUN BUDDY'. At the top, there's a navigation bar with links: 'What We Do', 'What You Do', 'Your Trainers', and 'Reach Out'. Below the navigation, there's a large image of a person's legs and feet as they lace up their running shoes. To the right of the image is a yellow sign-up form titled 'Get Started Today!'. The form includes fields for 'Your Name', 'Email Address', and 'Phone Number', each with a placeholder text. Below these fields are two radio buttons for 'Yes' and 'No' with the question 'Have you worked out with a trainer before?'. There's also a checkbox for 'I acknowledge that I am at least 18 years of age.' At the bottom of the form is a blue button labeled 'Get running!'.

## PAUSE

Why does the text run into the sign-up form?

The sign-up form was declared with absolute position, so it is removed from the document flow. This allows the text block to overlap with the sign-up form. We'll fix this in the next section when we add CSS styling.

[Hide Answer](#)

## Replace the Hero Image

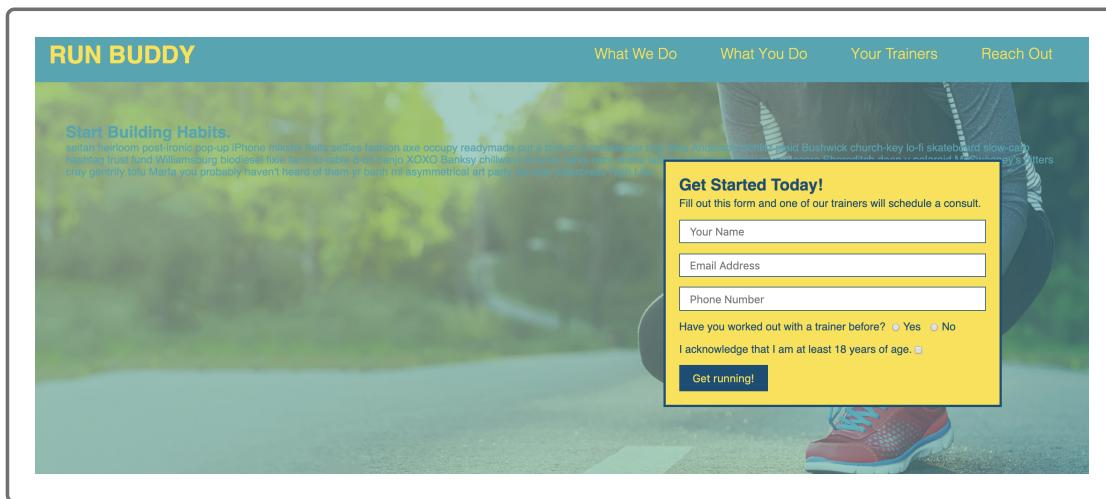
There's one more thing: the design team wants to flip the image of the person lacing up their shoes to the right side of the page. We'll do this by replacing the current image with a flipped version.

To replace the image, follow these steps:

1. Open the following image in Chrome: [flipped-hero-image.jpg](#)  
[\(https://static.fullstack-bootcamp.com/module-2/flipped-hero-image%2Ejpg\)](https://static.fullstack-bootcamp.com/module-2/flipped-hero-image%2Ejpg).

2. With the image open, press **Ctrl+S** (Windows) or **Command+S** (macOS) to save the image to your computer.
3. Change the name of the downloaded image to **hero-bg.jpg**.
4. Move the image into the **images** folder in the **assets** directory of the project.
5. Click the Replace button.

Refresh the **index.html** page in the browser. You should see the following:



## Save Your Progress Using Git

It doesn't look great now, but we'll fix that later. For now, the issue is complete, so let's add and commit all of our work in this branch.

You'll follow a slightly different Git workflow in this instance by pushing the local feature branch up to a remote feature branch. This creates a backup of your work just in case something like a lightning bolt or spilled cup of coffee accidentally destroys your local computer.

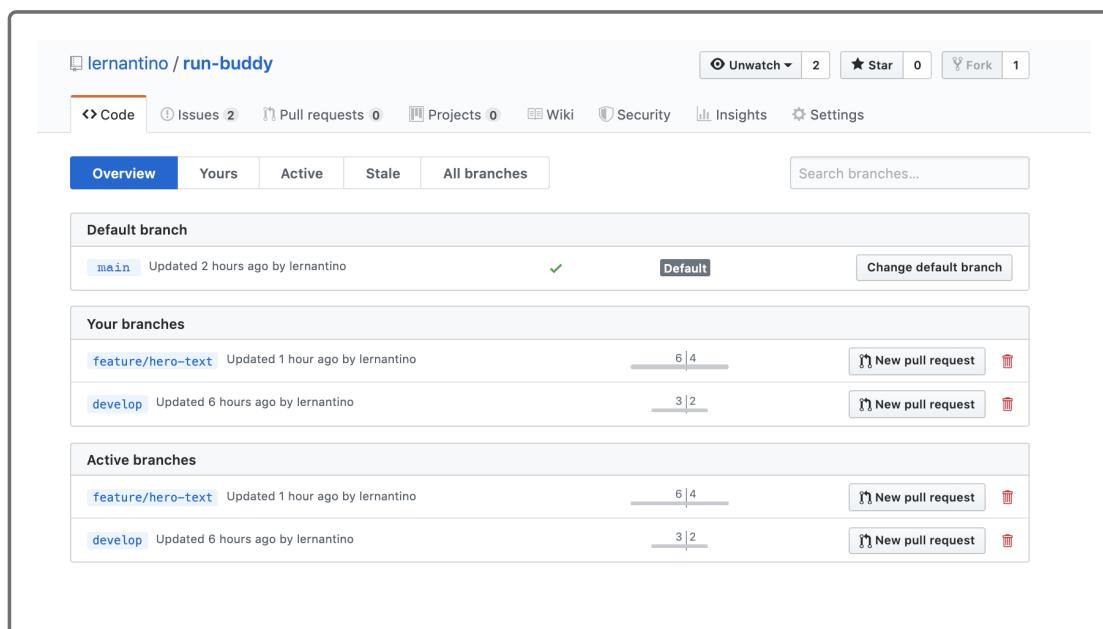
To do this, run the following commands:

```
git add .
git commit -m "add content to hero, replace image"
git push origin feature/hero-text
```

The slight variation in the `push` command illustrates that we're no longer pushing to the remote `main` branch; instead, we're pushing to a new remote feature branch named `feature/hero-text`. Hypothetically, we could've named this anything, but consistent naming conventions means one less branch name to commit to memory!

From now on, we'll add this step to the Git workflow after every add and commit of the feature branch, to secure the hard work we've done. After a successful push, we can check the GitHub repo and see that the backup branch is now safe and secure.

The following image shows the Code tab on GitHub, where we can see the `develop` and `feature/hero-text` branches under "Your branches":



## Merge the Feature into the Development Environment

In this final step, we'll integrate the new `hero-text` feature into the `develop` branch.

First, let's make `develop` the active branch by typing the following command:

```
git checkout develop
```

In VS Code, let's look at the hero section in the `index.html` file. Note that the new HTML we just added isn't visible here. It only exists in the feature branch, `feature/hero-text`. To add the new code to the `develop` branch, we need to perform a `merge` operation.

Let's do that now by typing the following command:

```
git merge feature/hero-text
```

### HIDE PRO TIP

Using the integrated terminal in VS Code is highly recommended because it's easy to check the status bar to see the active branch. To open the terminal in VS Code, go to the Terminal menu and choose New Terminal.

On successful merging, the command line should resemble the following image:

```
Updating 5c19f6c..bddfbdf
Fast-forward
 index.html | 180 ++++++
+-----+
 1 file changed, 180 insertions(+)
```

## REWIND

---

This should resemble the `git pull` operation that we executed earlier, because we also performed a `git merge` operation then as well.

Excellent work! Now `develop` is updated with the new HTML and has remained a stable codebase throughout.

© 2020 - 2021 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.