



Professional Certificate in Coding: Full Stack Development with MERN: Week 7

---

**Introduction To GitHub, Testing, And The DOM**

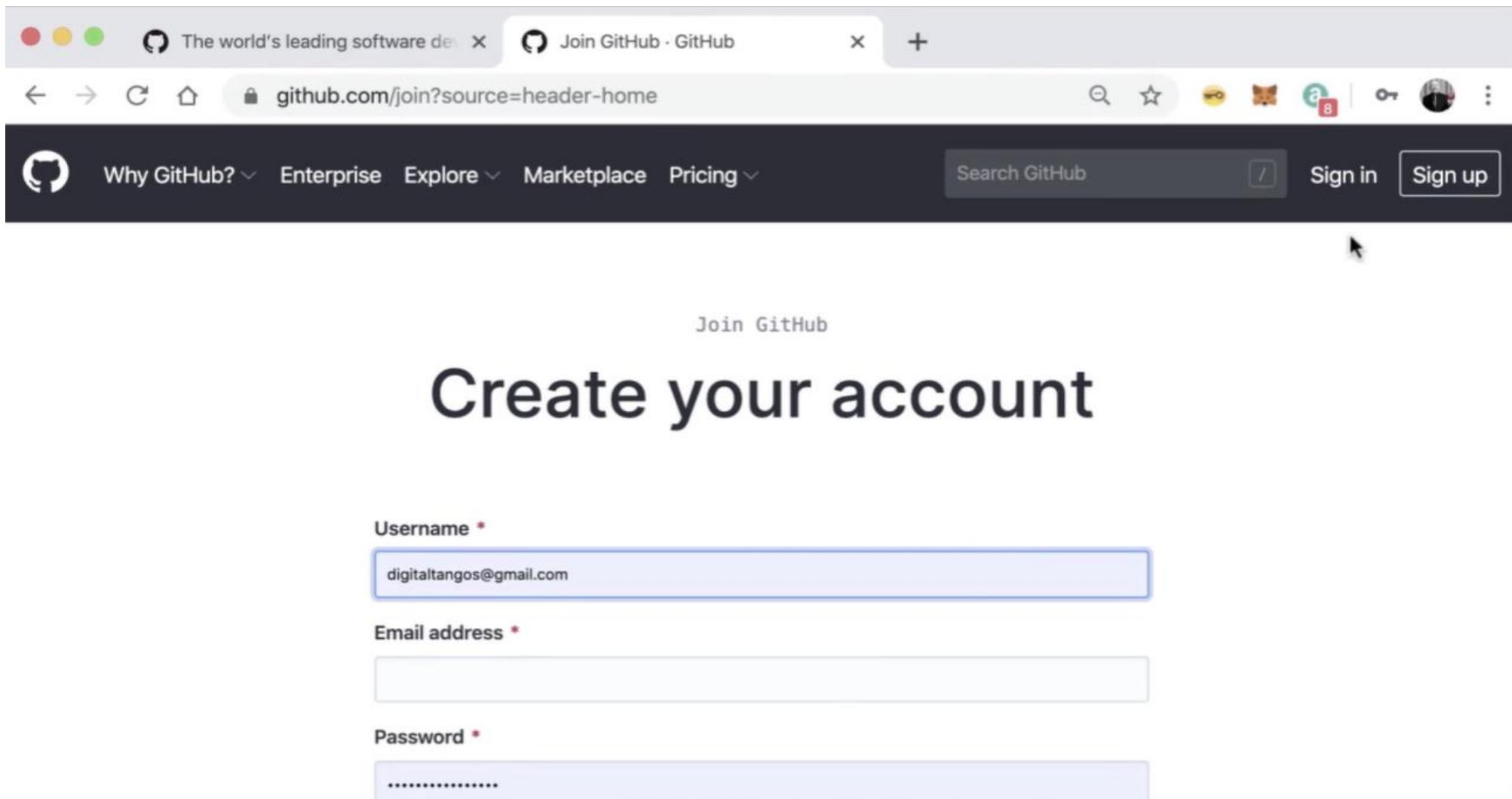
## Setup GitHub And Git: Login To GitHub (1/6)

- Get your own space on GitHub. Login + Password <https://github.com>
- Get Git on your machine. <https://git-scm.com>
- Generate SSH Keys on your machine and register them with Git Agent
  - <http://bit.ly/2Qr6MzO>
- Give GitHub your “public key” so it can recognize your Git Client

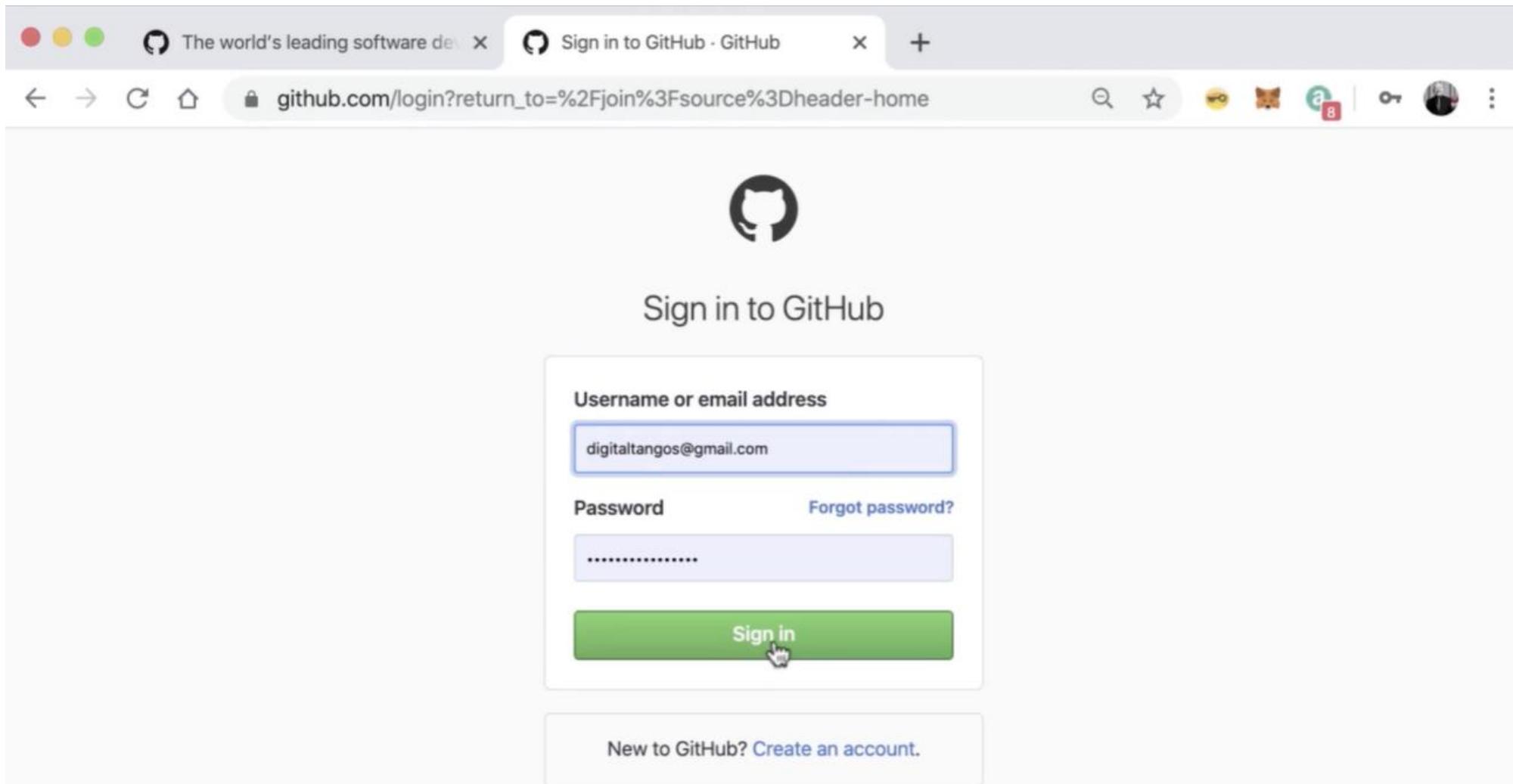
## Setup GitHub And Git: Login To GitHub (2/6)

A screenshot of the GitHub homepage. The top navigation bar includes links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar and 'Sign in' button are also present. The main content features a large heading 'Built for developers' followed by a paragraph about GitHub's mission. To the right, there are two sections: 'Individuals' (described as everything a developer needs) and 'Organizations' (described as essential management and security for teams). Each section has a blue 'View individual plans →' or 'View organization plans →' button.

## Setup GitHub And Git: Login To GitHub (3/6)



## Setup GitHub And Git: Login To GitHub(4/6)



# Setup GitHub And Git: Login To GitHub (5/6)

The world's leading software dev X GitHub X +

← → C H 🔒 github.com

Search or jump to... / Pull requests Issues Marketplace Explore

johntango

Repositories New

Find a repository...

johntango/jupyterCourse100  
johntango/johntango.github.io  
johntango/temp  
johntango/websocket-app  
johntango/s3colordemo  
johntango/test  
johntango/pacmanaws

Show more

Akhan304 forked Akhan304/circleciexpress from johntango/circleciexpress 2 days ago

johntango/circleciexpress ★ Star

JavaScript Updated Jan 7

zealdalal forked zealdalal/circleciexpress from johntango/circleciexpress 2 days ago

johntango/circleciexpress ★ Star

JavaScript Updated Jan 7

usmanaslam75 forked usmanaslam75/circleciexpress from johntango/circleciexpress 2 days ago

Signed in as johntango

Set status

Your profile

Your repositories

Your projects

Your stars

Your gists

Feature preview

Help

Settings

Sign out

# Setup GitHub And Git: Login To GitHub (6/6)

The screenshot shows a web browser window with the GitHub interface. The address bar displays `github.com/johntango?tab=repositories`. The main content area shows the user's profile picture (a smiling man in a suit), name (**John Williams**), and handle (`johntango`). Below this is an [Edit profile](#) button. The top navigation bar includes links for **Pull requests**, **Issues**, **Marketplace**, and **Explore**. The user statistics shown are: Overview, **Repositories 66**, Projects 0, Packages 0, Stars 2, Followers 6, Following 0. A search bar says "Find a repository..." with filters for Type: All and Language: All, and a green "New" button. Two repositories are listed: **circleciexpress** (JavaScript, 81 stars, updated 2 days ago) and **pacMan** (HTML, 16 stars, updated 18 days ago). Each repository has a "Star" button.

## Setup GitHub And Git: Install Git (1/6)

- Get your own space on GitHub. Login + Password <https://github.com>
- Get Git on your machine. <https://git.scm.com>
- Generate SSH Keys on your machine and register them with Git Agent
  - <http://bit.ly/2Qr6MzO>
- Give GitHub your “public key” so it can recognize your Git Client

# Setup GitHub And Git: Install Git (2/6)

The screenshot shows a web browser window with the URL `git-scm.com`. The page title is `git --local-branching-on-the-cheap`. The main content area contains two paragraphs about Git's features and performance. To the right of the text is a 3D-style diagram illustrating a distributed version control system with multiple repositories connected by bidirectional arrows.

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint](#) with [lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), [convenient staging areas](#), and [multiple workflows](#).

# Setup GitHub And Git: Install Git (3/6)

The screenshot shows the official Git website ([git-scm.com](https://git-scm.com)) with a clean, modern design. On the left, there are four main navigation links: "About" (with a gear icon), "Documentation" (with a book icon), "Downloads" (with a download arrow icon), and "Community" (with a speech bubble icon). Below these, a section for the "Pro Git" book is shown with a thumbnail image and a link to Amazon.com. On the right, a large callout box highlights the "Latest source Release" which is version 2.25.1, with a "Release Notes (2020-02-17)" link and a prominent "Download 2.23.0 for Mac" button. At the bottom, links for "Mac GUIs", "Tarballs", "Windows Build", and "Source Code" are provided.

The website URL in the browser bar is [git-scm.com](https://git-scm.com).

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

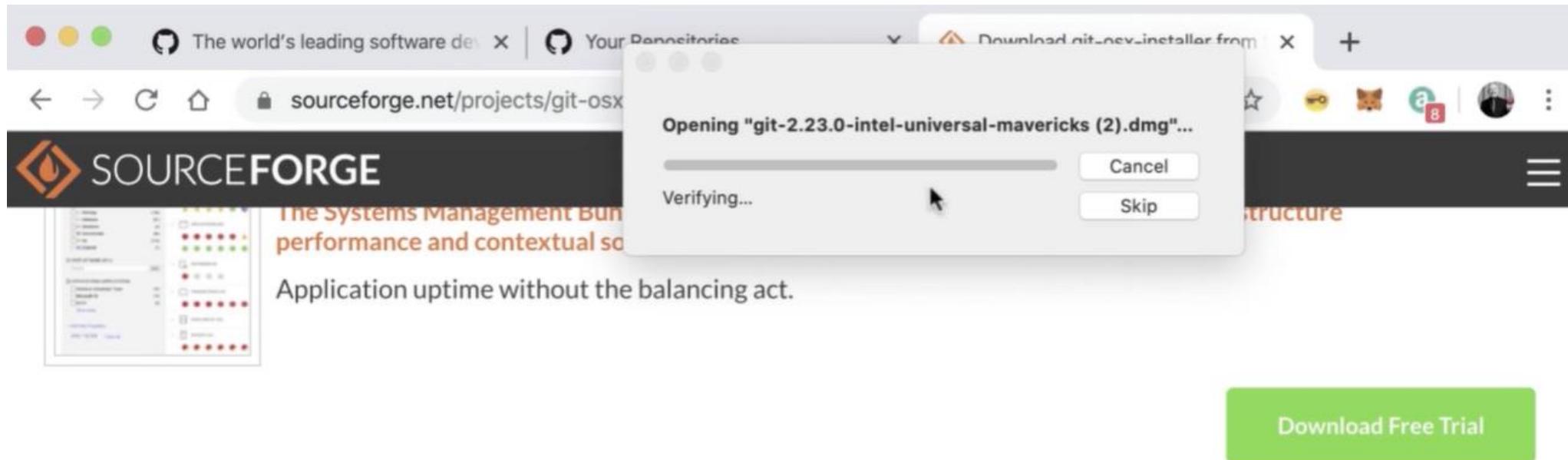
**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

**Pro Git** by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

Latest source Release  
**2.25.1**  
Release Notes (2020-02-17)  
Download 2.23.0 for Mac

Mac GUIs Tarballs  
Windows Build Source Code

# Setup GitHub And Git: Install Git (4/6)





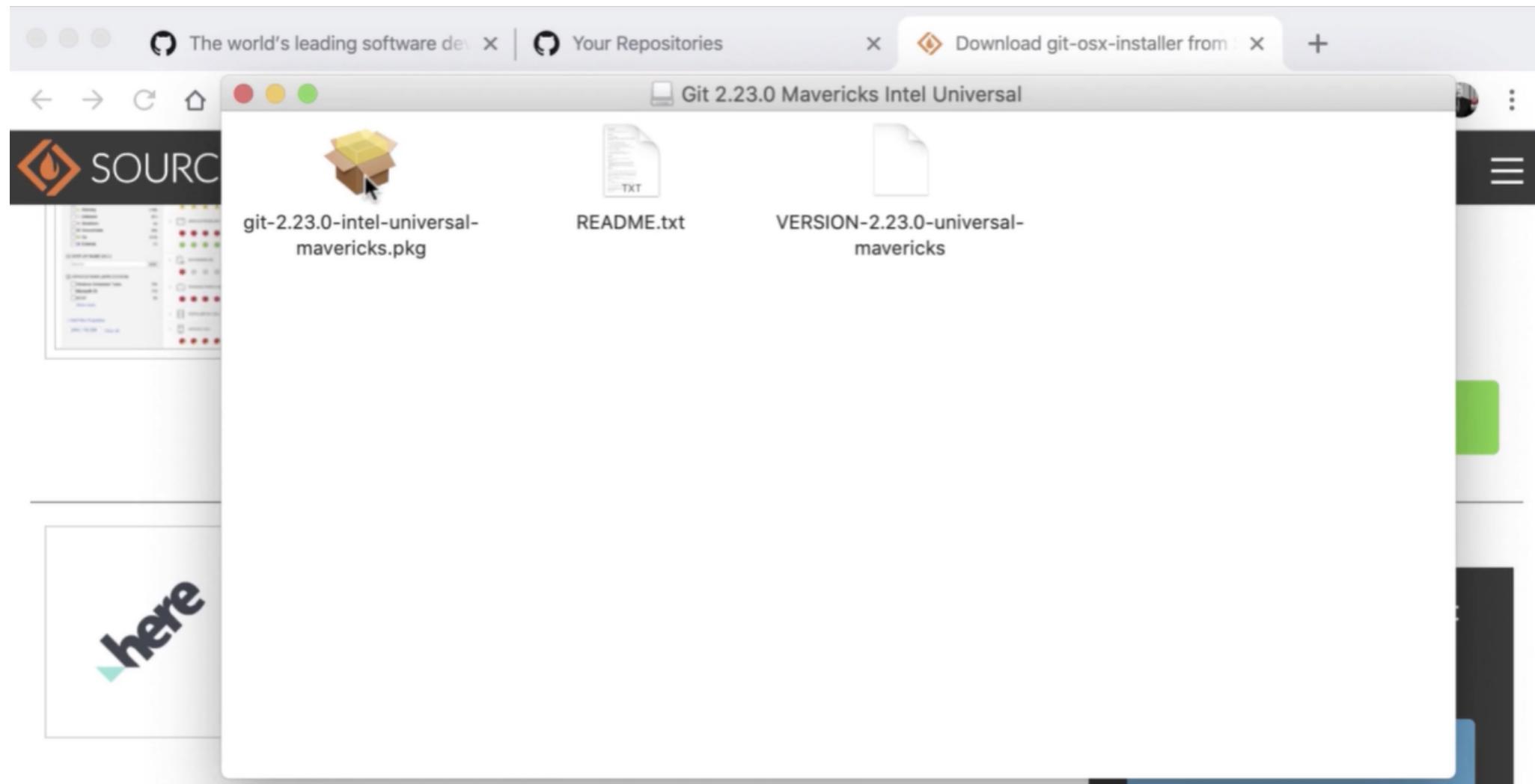
**Build location-aware applications**  
20+ APIs to help AWS developers solve complex problems by harnessing the power of location data.

Location data gives applications an entirely new layer of context, and in turn, opens up a world of possibilities. HERE Location Services is the world's #1 location platform performing billions of requests every day. It provides location-based mapping, routing, translation between geocoordinates and addresses, and much more.

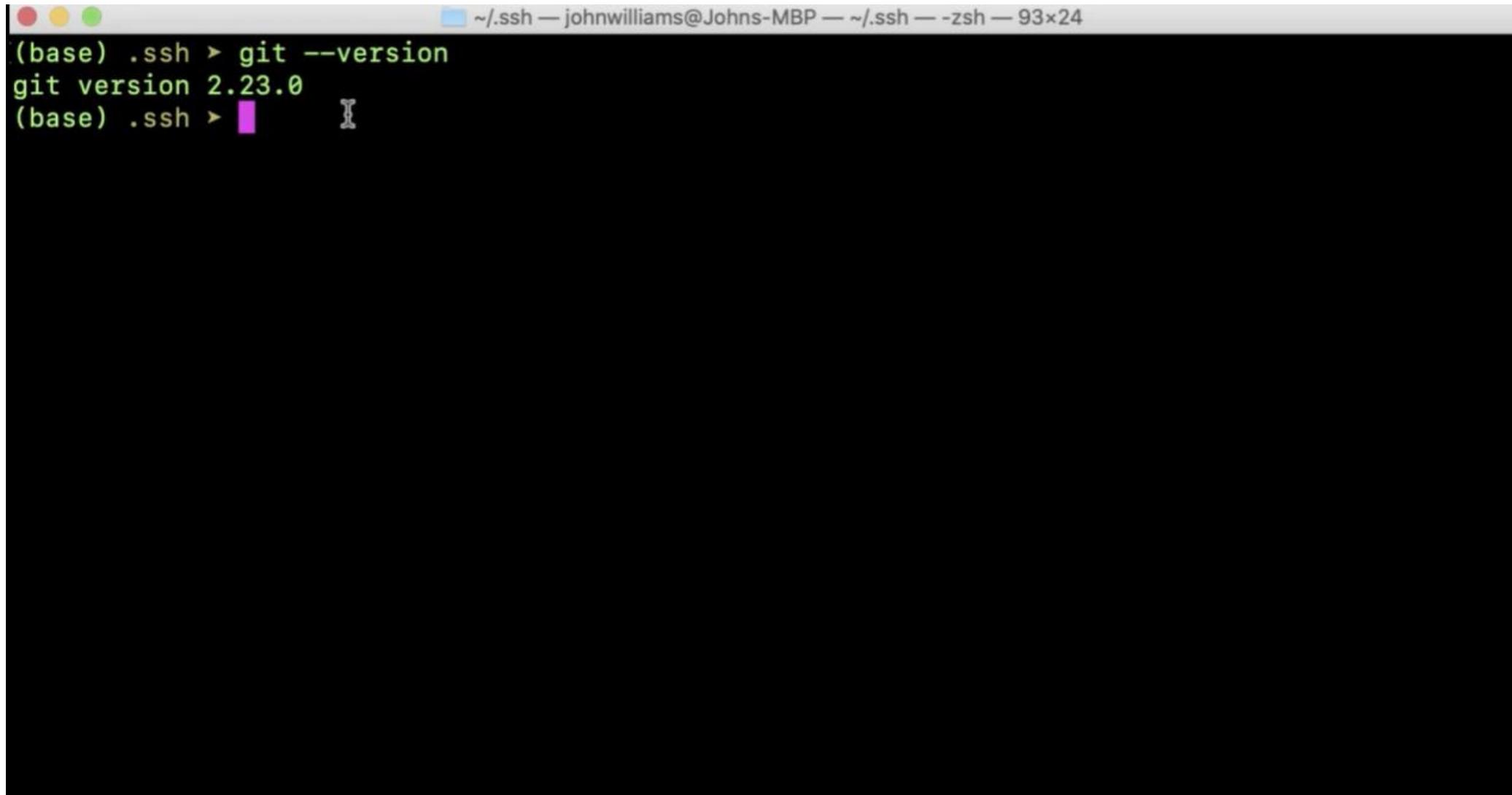
**Get latest updates about Open Source Projects, Conferences and News.**

[Sign Up](#)

# Setup GitHub And Git: Install Git (5/6)



# Setup GitHub And Git: Install Git (6/6)



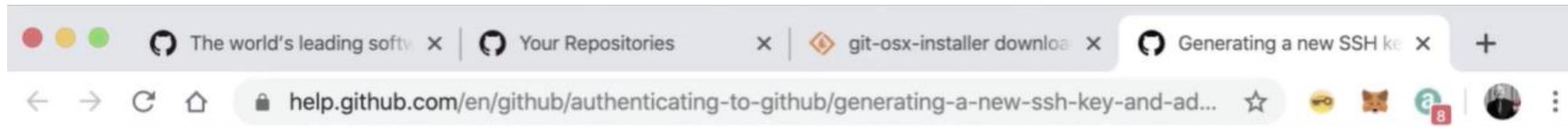
A screenshot of a terminal window titled 'zsh' with the path '~/.ssh'. The window shows the command 'git --version' being run, which outputs 'git version 2.23.0'. The terminal has a dark background with light-colored text and a light blue header bar.

```
(base) .ssh > git --version
git version 2.23.0
(base) .ssh >
```

## Setup GitHub And Git: Generate SSH Keys (1/13)

- Get your own space on GitHub. Login + Password <https://github.com>
- Get Git on your machine. <https://git-scm.com>
- Generate SSH Keys on your machine and register them with Git Agent
  - <http://bit.ly/2Qr6MzO>
- Give GitHub your “public key” so it can recognize your Git Client

# Setup GitHub And Git: Generate SSH Keys (2/13)



The screenshot shows a web browser window with several tabs open. The active tab is titled "Generating a new SSH key" and has the URL [help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent](https://help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent). The page content discusses the SSH agent and provides instructions for generating a new SSH key.

## Generating a new SSH key

- 1 Open Terminal.
- 2 Paste the text below, substituting in your GitHub email address.

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

This creates a new ssh key, using the provided email as a label.

```
> Generating public/private rsa key pair.
```

- 3 When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

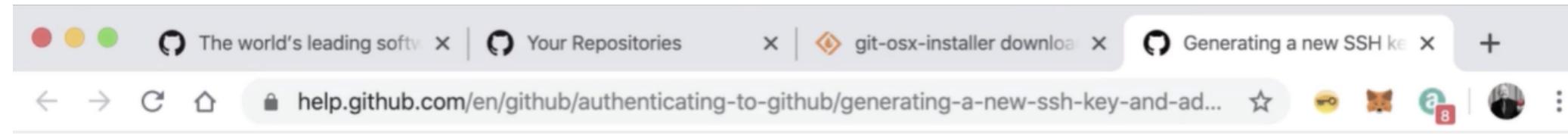
# Setup GitHub And Git: Generate SSH Keys (3/13)

```
● ● ● ~/.ssh — ssh-keygen -t rsa -b 4096 -C "jrw@mit.edu" — ssh-keygen — ssh-keygen -t rsa -b 4096 -C jrw@mit.edu — 93x24
(base) .ssh > git --version
git version 2.23.0
(base) .ssh > ssh-keygen -t rsa -b 4096 -C "jrw@mit.edu"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/johnwilliams/.ssh/id_rsa): /Users/johnwilliams/.s
sh/test1
Enter passphrase (empty for no passphrase):
Enter same passphrase again: ?
```

## Setup GitHub And Git: Generate SSH Keys (4/13)

```
sh/test1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/johnwilliams/.ssh/test1.
Your public key has been saved in /Users/johnwilliams/.ssh/test1.pub.
The key fingerprint is:
SHA256:I9fHVPRnX9J/vtQiz0PJ6BjNvGI/E8V9z0XsEHycOZw jrw@mit.edu
The key's randomart image is:
+---[RSA 4096]---+
|          o=++|
|          .oE=|
|          ...*B|
|          . * oo+X|
|          . S o B.+.B|
|          o . =.o .=|
|          + +.=.o|
|          . oo+.o.|
|          .o . |
+---[SHA256]---+
(base) .ssh > ls test1
test1
(base) .ssh > ls test1.pub
test1.pub
(base) .ssh >
```

# Setup GitHub And Git: Generate SSH Keys (5/13)



## Adding your SSH key to the ssh-agent

Before adding a new SSH key to the ssh-agent to manage your keys, you should have [checked for existing SSH keys](#) and [generated a new SSH key](#). When adding your SSH key to the agent, use the default macOS `ssh-add` command, and not an application installed by [macports](#), [homebrew](#), or some other external source.

- 1 Start the ssh-agent in the background.

```
$ eval "$(ssh-agent -s)"  
> Agent pid 59566
```

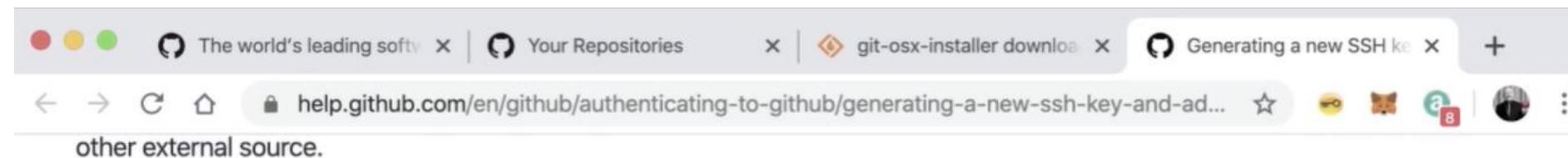
- 2 If you're using macOS Sierra 10.12.2 or later, you will need to modify your `~/.ssh/config` file to automatically load keys into the ssh-agent and store passphrases in your keychain.

```
Host *
```

# Setup GitHub And Git: Generate SSH Keys (6/13)

```
~/.ssh — johnwilliams@Johns-MBP — ~/.ssh — -zsh — 93x24
Enter same passphrase again:
Your identification has been saved in /Users/johnwilliams/.ssh/test1.
Your public key has been saved in /Users/johnwilliams/.ssh/test1.pub.
The key fingerprint is:
SHA256:I9fHVPRnX9J/vtQiz0PJ6BjNvGI/E8V9z0XsEHyc0Zw jrw@mit.edu
The key's randomart image is:
+---[RSA 4096]---+
|   o=++|
|   .oE=|
|   ...*B|
|   . * oo+X|
|   . S o B.+.B|
|   o . =.o .=|
|   + +.=.o|
|   . oo+.o.|
|   .o . |
+---[SHA256]---+
(base) .ssh > ls test1
test1
(base) .ssh > ls test1.pub
test1.pub
(base) .ssh > eval "$(ssh-agent -s)"
Agent pid 5164
(base) .ssh >
```

# Setup GitHub And Git: Generate SSH Keys (7/13)



The screenshot shows a Mac OS X desktop with a browser window open. The title bar has four tabs: "The world's leading softy", "Your Repositories", "git-osx-installer download", and "Generating a new SSH key". The main content area of the browser shows a guide from help.github.com. The visible text reads: "other external source.

- 1 Start the ssh-agent in the background.

```
$ eval "$(ssh-agent -s)"  
> Agent pid 59566
```

- 2 If you're using macOS Sierra 10.12.2 or later, you will need to modify your `~/.ssh/config` file to automatically load keys into the ssh-agent and store passphrases in your keychain.

```
Host *  
  AddKeysToAgent yes  
  UseKeychain yes  
  IdentityFile ~/.ssh/id_rsa
```

- 3 Add your SSH private key to the ssh-agent and store your passphrase in the keychain. If you created your key with a different name, or if you are adding an existing key that has a different name, replace `id_rsa` in the command with the name of your private key file.

## Setup GitHub And Git: Generate SSH Keys (8/13)

```
The key's randomart image is:  
+---[RSA 4096]---+  
|          o=++|  
|          .oE=|  
|          ...*B|  
|          . * oo+X|  
| . S o B.+.B|  
| o . =.o .=|  
| + +.=.o|  
| . oo+.o.|  
|          .o . |  
+---[SHA256]---+  
(base) .ssh > ls test1  
test1  
(base) .ssh > ls test1.pub  
test1.pub  
(base) .ssh > eval "$(ssh-agent -s)"  
Agent pid 5164  
(base) .ssh > cat config  
Host *  
  AddKeysToAgent yes  
  UseKeychain yes  
  IdentityFile ~/.ssh/jrwcats%  
(base) .ssh > █
```

# Setup GitHub And Git: Generate SSH Keys (9/13)

```
~/.ssh — johnwilliams@Johns-MBP — ~/.ssh — -zsh — 93x24
+----[SHA256]----+
(base) .ssh > ls test1
test1
(base) .ssh > ls test1.pub
test1.pub
(base) .ssh > eval "$(ssh-agent -s)"
Agent pid 5164
(base) .ssh > cat config
Host *
  AddKeysToAgent yes
  UseKeychain yes
  IdentityFile ~/.ssh/jrwcats%
(base) .ssh > ssh-add -K ~/.ssh/test1
Identity added: /Users/johnwilliams/.ssh/test1 (jrw@mit.edu)
(base) .ssh > cat test1.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACQC1few9i5/DolRKdr7c1BED11i8lmsky01dK7vGUcqvP4YPXQZSYeWhp
2fy52vekoj01P5Uoq0dB1NDZMDrNunBjMGjqSPeKmtq3zC89DTdSgBfqqjZF5nEaIMTmeZbycf1/GknbDpCPORI3azVDG
dDqgsqH5i4JGhxpzf+1E+1iW1Zx+3BaO1SgvSy1EgAkdeIhADEiUOE15AtXrviqR1w7oaMAXvodfnPc2gV3FiVHLXj4a
/DcJPIR/ukDIBI/6Y3XJd3v2dmBMyG3nJyGwIatAug+1Hdf+i2DgjLHUNyN2YtGHmtzmaUyg137LTm0VBRrsIkyDINhQR
5nSyDOQdhNjqpVA1/hVjI5bSyaFstVrE/KN92FVTMR4dJJQsVaR7XWa5UDHerepuIig53MvR9fvZqcZNstNcUM64kzy8X
YHw5ZTGMjHqqpKHTHP9EMq8vmiFG7mxrVrJw0nNZKsWMssP7HSpolqxdhzsee+gM/2WwH8WfTTq107DK7mX5escrB73Iu
X5dh0NfSa/ObJS1L90cPSKmhQL+Yst80PW2rkJ9C2aJg0jIa9P1YdGBp+S5uIIPG4tz9x6cP2qq3zdUjz34ICGYmHWKN
1CZj27qTjw+l2u+v54XPtMYrq/sa5upnFVuNFhVjBBfCIIVs/iWW58CvVgR+FzCjgSJxQ== jrw@mit.edu
(base) .ssh >
```

# Setup GitHub And Git: Generate SSH Keys (10/13)

The screenshot shows a Mac OS X desktop with a browser window open to [github.com/settings/keys](https://github.com/settings/keys). The browser has four tabs: "The world's leading soft...", "SSH and GPG keys", "git-osx-installer download...", and "Generating a new SSH key". The main content area displays the "SSH keys" section under "Personal settings". A sidebar on the left lists "Profile", "Account", "Security", "Security log", "Emails", "Notifications", "Billing", "SSH and GPG keys" (which is selected and highlighted in orange), and "Blocked users". On the right, the "SSH keys" section shows three entries:

- macbook**  
35:3f:b1:b8:d0:61:24:53:4a:84:ed:5b:72:0f:ce:96  
Added on Jan 7, 2019  
Last used within the last 11 months — Read/write  
Delete
- digitaltangos**  
41:c0:e9:89:11:cc:7f:20:10:ff:c7:6a:40:d4:0f:30  
Added on Jul 16, 2019  
Last used within the last 6 months — Read/write  
Delete
- jrwcats**

A green "New SSH key" button is visible in the top right corner of the main content area.

# Setup GitHub And Git: Generate SSH Keys (11/13)

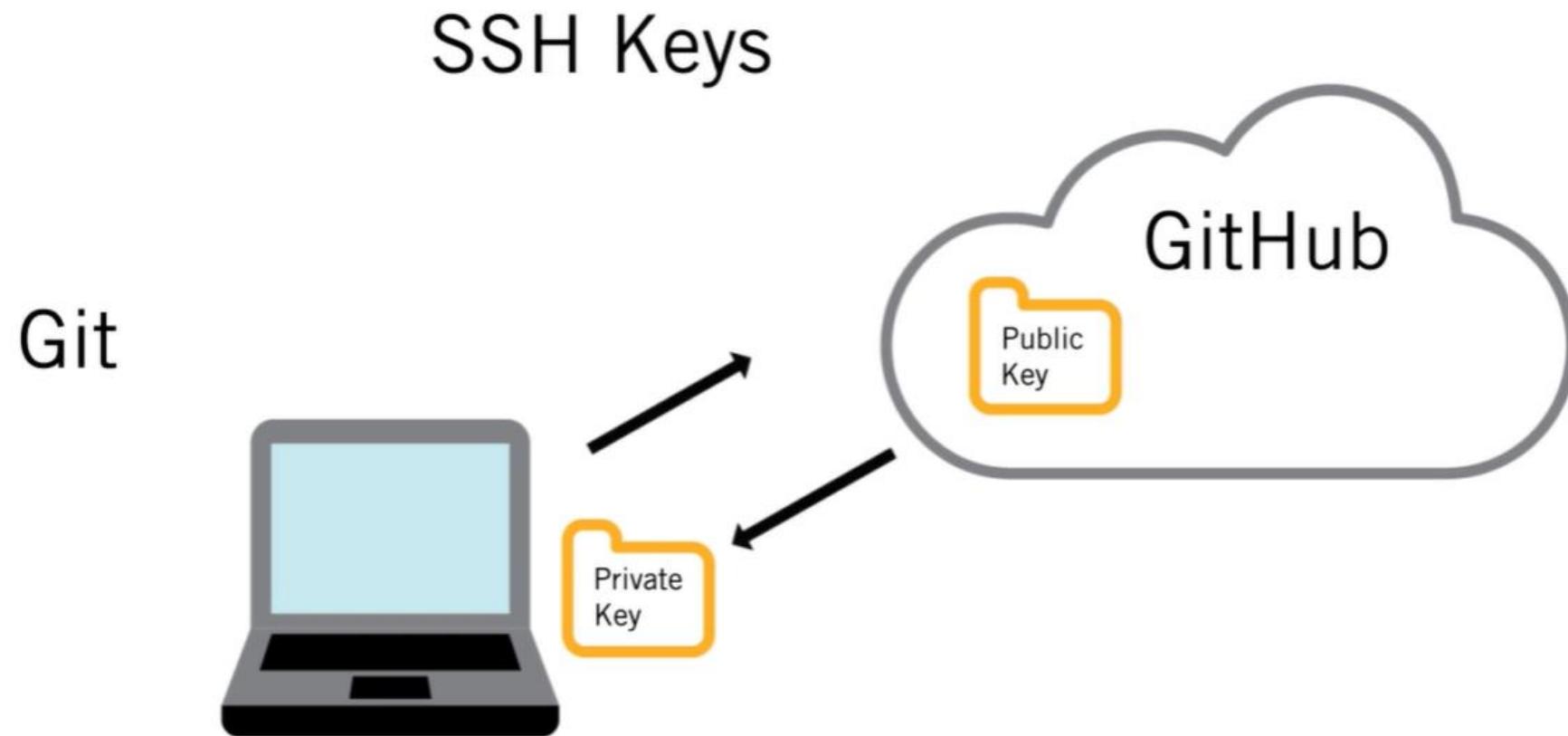
The screenshot shows a Mac OS X desktop with several browser tabs open in a window. The tabs include "The world's leading soft...", "Add new SSH keys", "git-osx-installer download", "Generating a new SSH key", and a new tab indicator. The address bar shows the URL [github.com/settings/ssh/new](https://github.com/settings/ssh/new). Below the tabs is a dark navigation bar with icons for search, star, emoji, notifications, and user profile, along with a count of 8 notifications.

The main content area is titled "SSH keys / Add new". On the left, a sidebar under "Personal settings" has a "Profile" section selected, followed by "Account", "Security", "Security log", "Emails", "Notifications", "Billing", "SSH and GPG keys" (which is highlighted in blue), "Blocked users", and "Repositories".

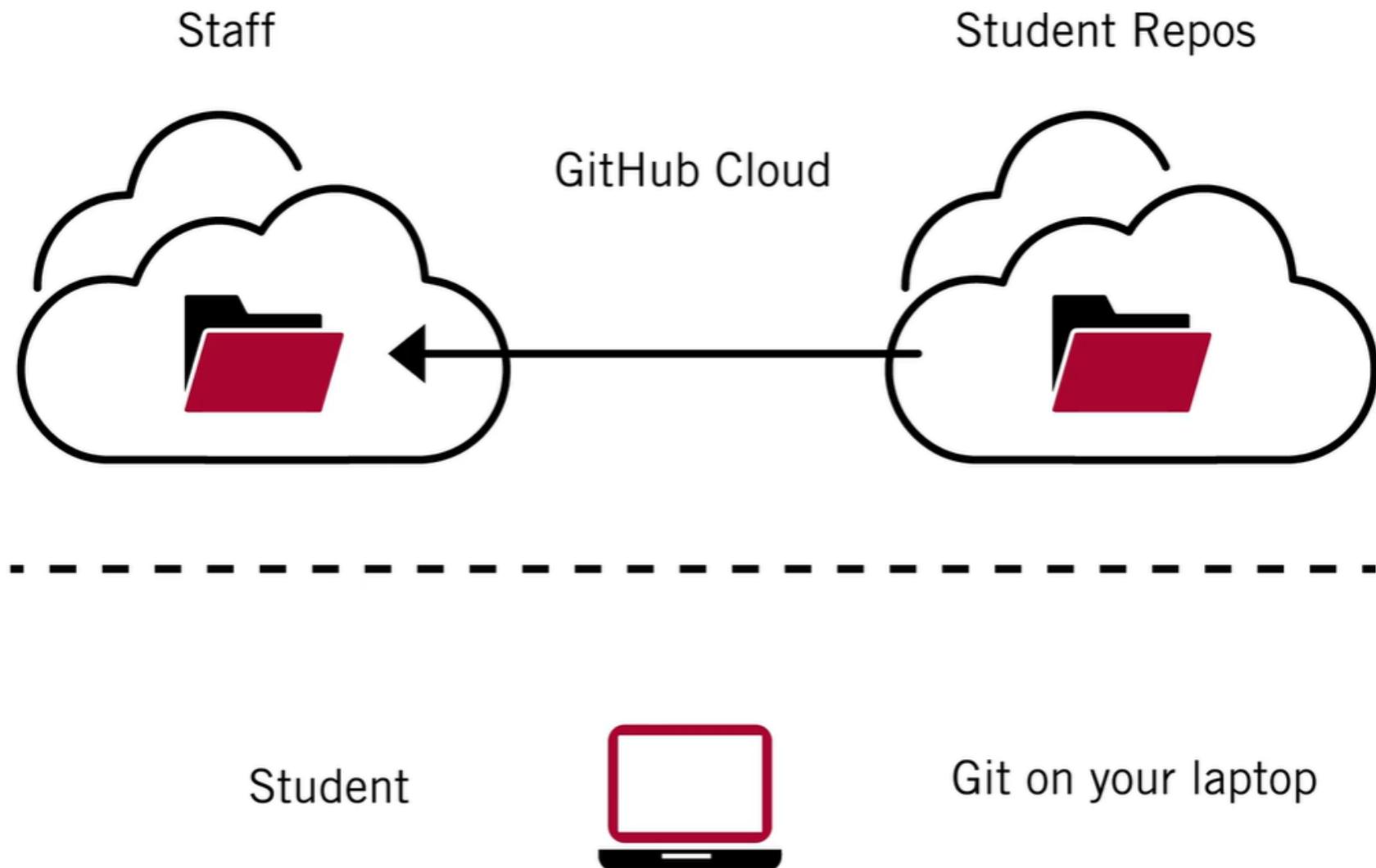
The "Key" section contains an "ssh-rsa" key block with a long string of characters, ending with "jrw@mit.edu". At the bottom of this section is a green "Add SSH key" button.

# Setup GitHub And Git: Generate SSH Keys (12/13)

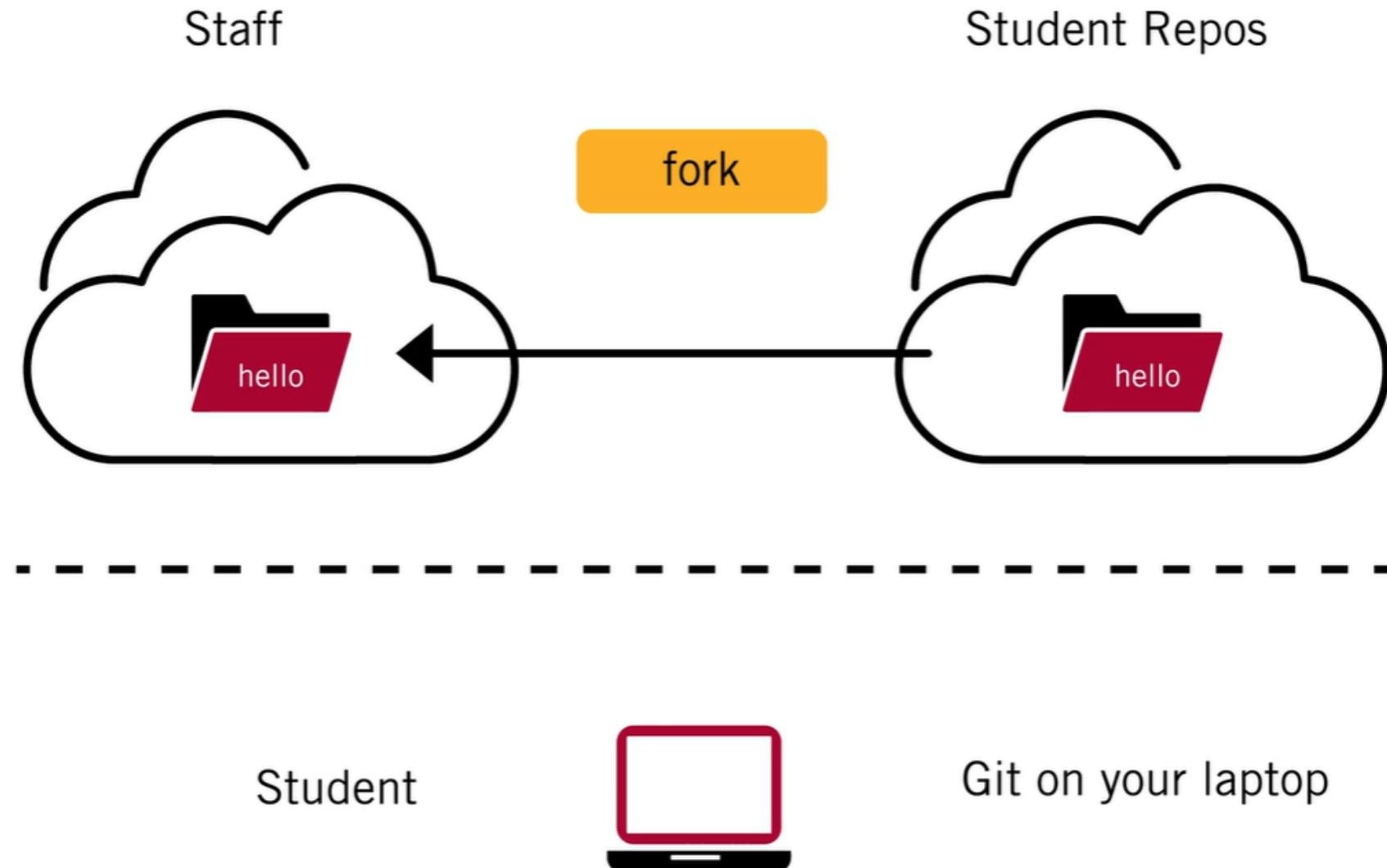
# Setup GitHub And Git: Generate SSH Keys (13/13)



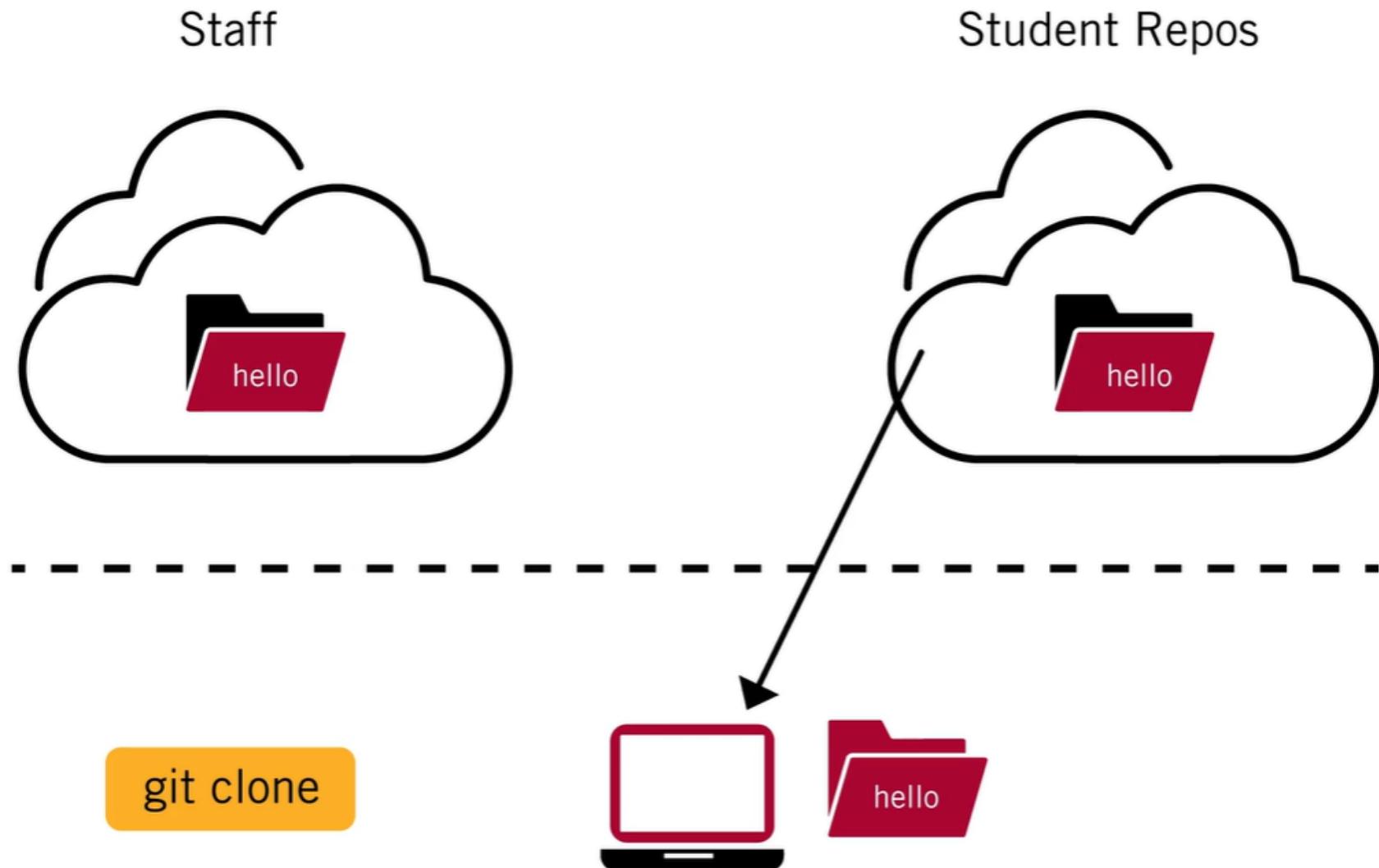
# Use Of GitHub (1/7)



## Use Of GitHub (2/7)



## Use Of GitHub (3/7)



## Use Of GitHub (4/7)

Staff



Student Repos

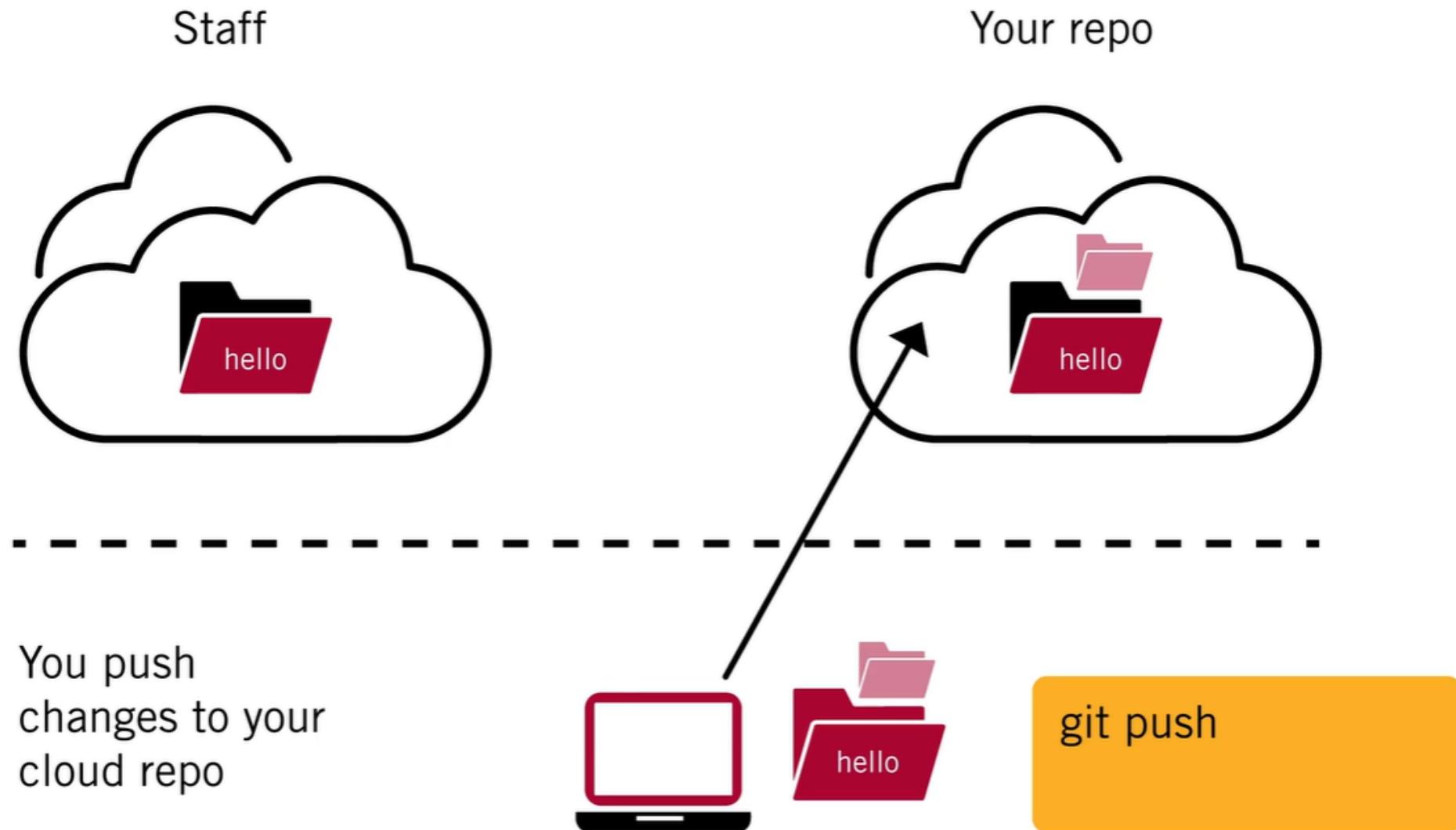


You make  
changes to hello  
save using git

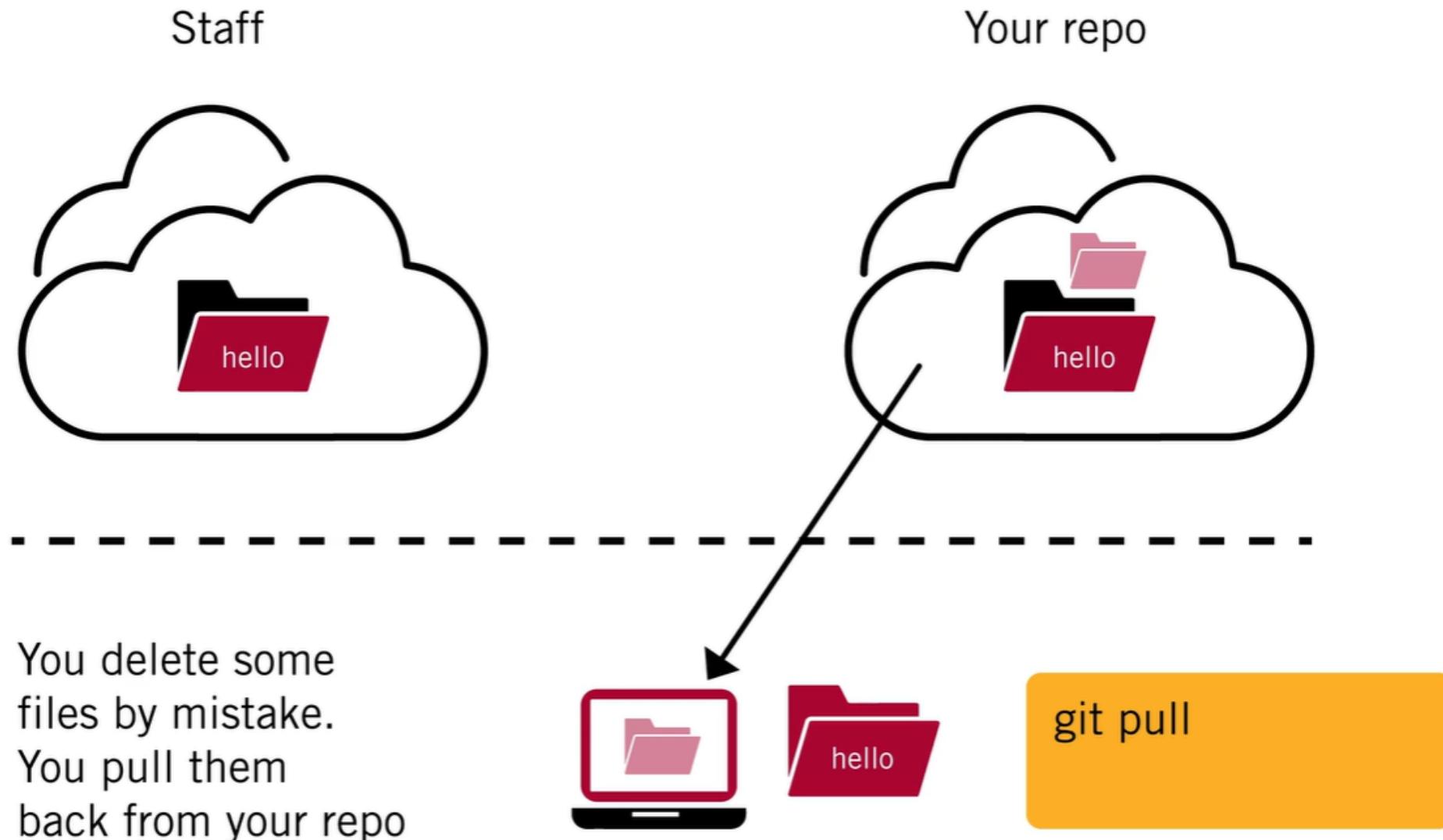


```
git add .  
git commit -m 'first'
```

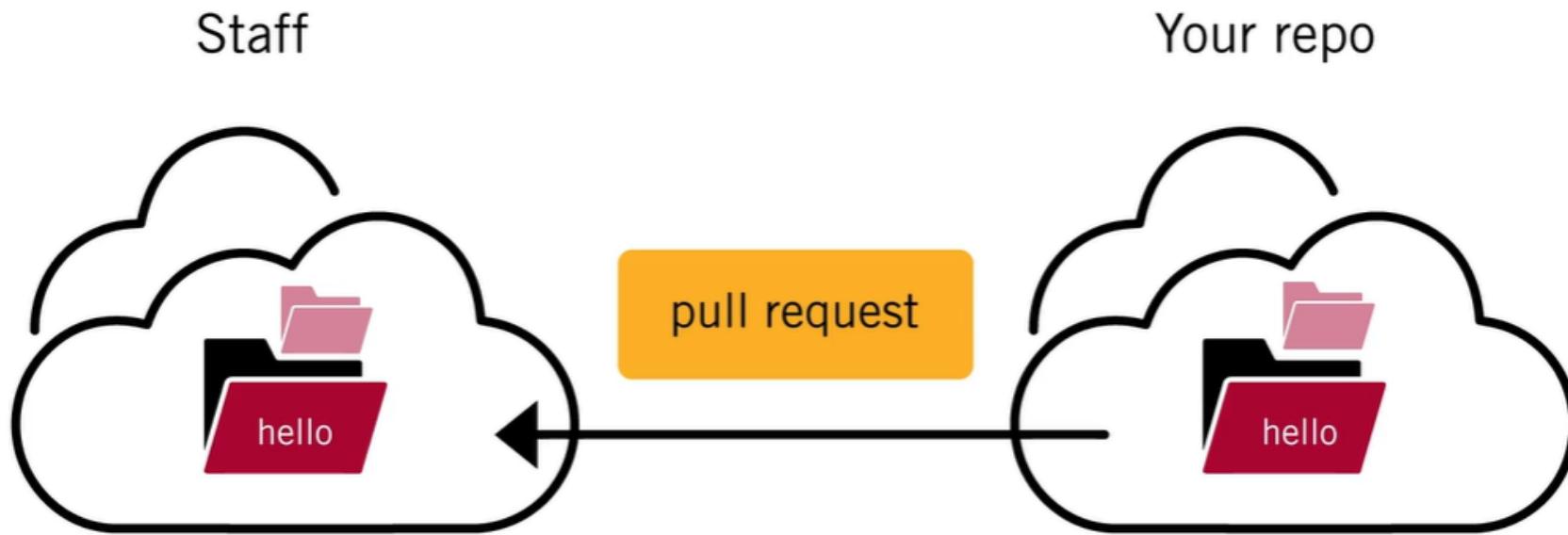
## Use Of GitHub (5/7)



## Use Of GitHub (6/7)



# GitHub Cycle (7/7)



You request staff pull your changes to their repo

-----



# GitHub Cycle: Demonstration (1/12)

A screenshot of a GitHub user profile page for 'johntango'. The profile shows a message: 'Your repository "johntango/hello-1" was successfully deleted.' Below this, there's a list of recent forks:

- jjcs227 forked jjcs227/circleciexpress from johntango/circleciexpress 16 days ago
- hwajdi forked hwajdi/circleciexpress from johntango/circleciexpress 16 days ago
- monica-ramirez-MIT forked monica-ramirez-MIT/circleciexpress from johntango/circleciexpress 17 days ago

The right sidebar contains a dropdown menu for the user 'johntango'.

- Signed in as **johntango**
- [Set status](#)
- [Your profile](#)
- [Your repositories](#) (highlighted in blue)
- [Your organizations](#)
- [Your projects](#)
- [Your stars](#)
- [Your gists](#)
- [Feature preview](#)
- [Help](#)
- [Settings](#)
- [Sign out](#)

At the bottom right, there are links to other repositories: [tomwhite/covid-19-uk-data](#) and [HTML ★ 147](#).

# GitHub Cycle: Demonstration (2/12)

The screenshot shows a GitHub search results page for the query "hello". The search bar at the top contains "github.com/johntango?tab=repositories" and the search term "hello". The results list includes:

- johntango/hello
- johntango/hellojs
- johntango/helloTest
- kogspace/hello-johntango
- simonqian/react-helloworld
- StephenGrider/FullstackReactCode

On the right side of the search results, there is a summary of the user's activity:

- 72 GitHub contributions
- 0 Projects
- 0 Packages
- 2 Stars
- 8 Followers
- 0 Following

Below the search results, there is a section for the user "John Williams" (johntango). It shows their profile picture, name, handle, and an "Edit profile" button.

One repository is highlighted: **reactBadbank01**. It is described as "React Bootstrap BadBank". The repository details are as follows:

- JavaScript
- MIT License
- Updated 3 days ago

# GitHub Cycle: Demonstration (3/12)

A screenshot of a GitHub search results page for the query "hello". The search bar at the top shows "Search · hello". The results page displays 1,887,096 repository results. On the left, there is a sidebar with categories: Repositories (1M), Code (93M+), Commits (3M), Issues (3M), Discussions (Beta) (777), Packages (535), Marketplace (23), Topics (245), Wikis (111K), and Users (68K). Below that is a Languages section showing HTML (248,605) and Java (84,006). The main area lists four repositories:

- phonegap/phonegap-start**  
PhoneGap Hello World app  
3.4k JavaScript Updated on Oct 24, 2019
- jlore/hello**  
[Fork-n-Go] A simple personal page. Demo:  
104 CSS Updated 15 days ago
- 521xueweihan/HelloGitHub**  
Find pearls on open-source seashore 分享 GitHub 上有趣、入门级的开源项目  
hellogithub github python awesome  
31.3k Python Updated 2 days ago
- blackbird71SR/Hello-World** Archived  
Hello World in all possible programming languages

# GitHub Cycle: Demonstration (4/12)

The screenshot shows a GitHub repository page for the user `jlord` named `hello`. The page includes the following details:

- Watched:** 16
- Starred:** 104
- Forked:** 439 (with a cursor icon indicating it's being interacted with)
- Issues:** 3
- Pull requests:** 9
- Actions:** 0
- Projects:** 0
- Wiki:** 0
- Security:** 0
- Insights:** 0

The repository summary section shows:

- 29 commits**
- 3 branches**
- 0 packages**
- 0 releases**
- 3 contributors**

Branch dropdown: `gh-pages`

New pull request button

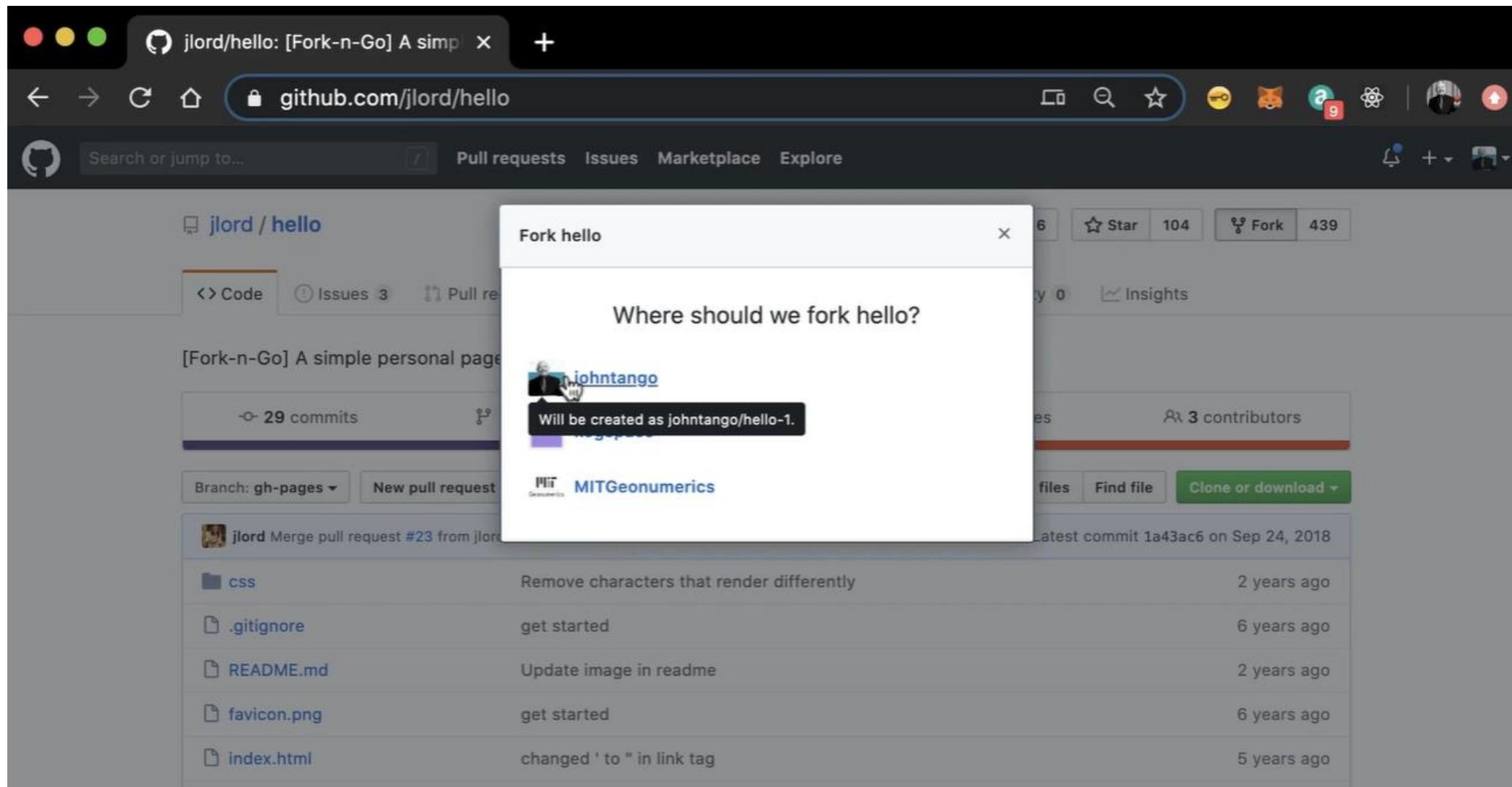
Create new file, Upload files, Find file, Clone or download buttons

Latest commit: `1a43ac6` on Sep 24, 2018

Commit history:

File	Message	Date
<code>css</code>	Remove characters that render differently	2 years ago
<code>.gitignore</code>	get started	6 years ago
<code>README.md</code>	Update image in readme	2 years ago
<code>favicon.png</code>	get started	6 years ago
<code>index.html</code>	changed ' to " in link tag	5 years ago

# GitHub Cycle: Demonstration (5/12)

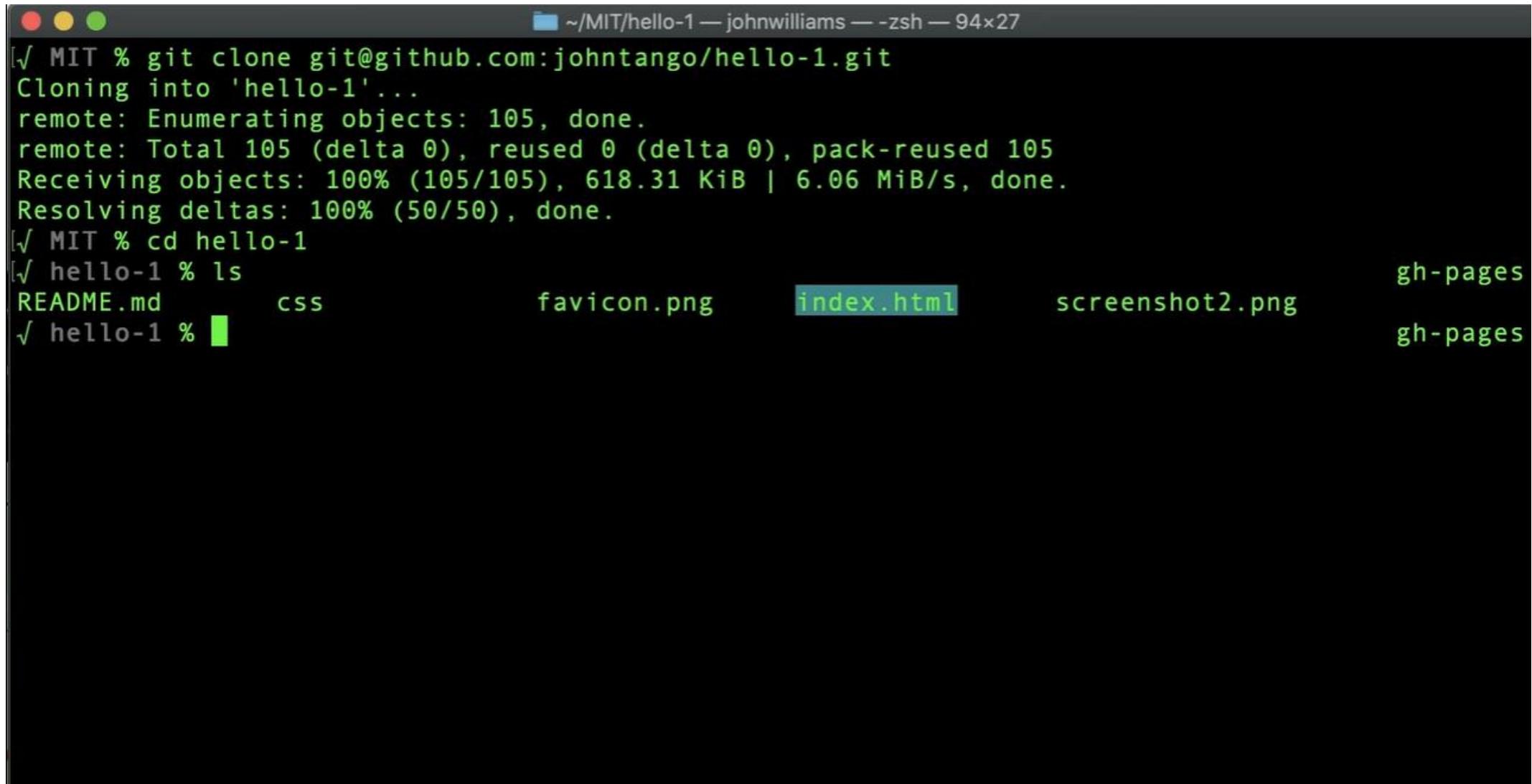


# GitHub Cycle: Demonstration (6/12)

The screenshot shows a GitHub repository page for 'johntango/hello-1'. The repository is a fork of 'jlord/hello'. The main navigation bar includes links for Code, Pull requests (0), Actions, Projects (0), Wiki, Security (0), Insights, and Settings. Below the navigation, a summary box displays: 29 commits, 3 branches, 0 packages, 0 releases, and 3 contributors. A dropdown menu for the branch 'gh-pages' is open, showing options for New pull request, Create new file, Upload files, Find file, and Clone or download. The 'Clone or download' button is highlighted in green. A tooltip for 'Clone with SSH' is visible, showing the URL 'git@github.com:johntango/hello-1.git'. The repository's file list includes: .gitignore, README.md, favicon.png, index.html, and screenshot2.png. Commit history is listed as follows:

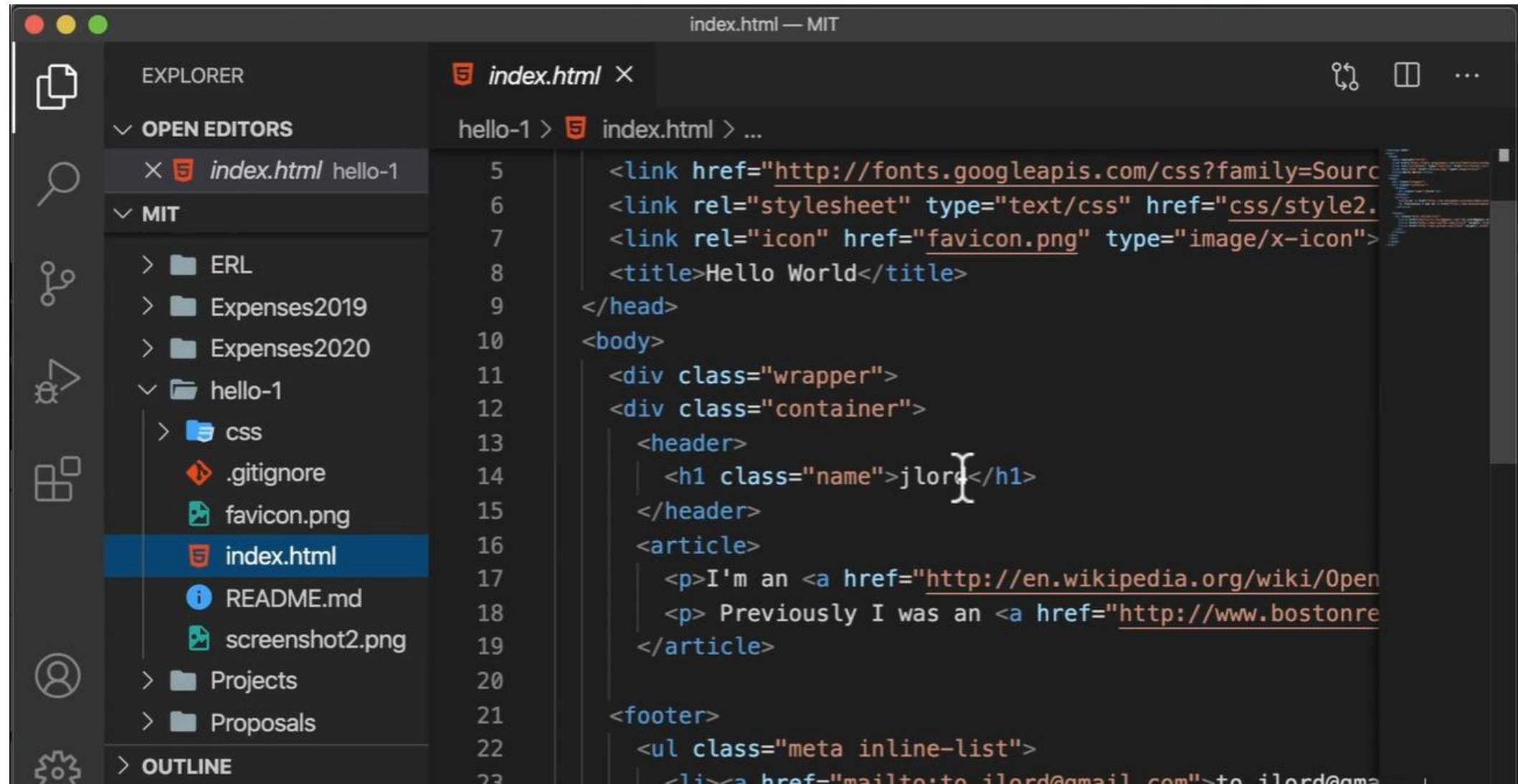
Commit	Description	Date
jlord Merge pull request jlord#23 from jlord/remove-characters	... Remove characters that render differently	2 years ago
.gitignore	get started	6 years ago
README.md	Update image in readme	5 years ago
favicon.png	get started	6 years ago
index.html	changed ' to " in link tag	6 years ago
screenshot2.png	edit readme for other style	6 years ago

# GitHub Cycle: Demonstration (7/12)



```
[✓ MIT % git clone git@github.com:johntango/hello-1.git
Cloning into 'hello-1'...
remote: Enumerating objects: 105, done.
remote: Total 105 (delta 0), reused 0 (delta 0), pack-reused 105
Receiving objects: 100% (105/105), 618.31 KiB | 6.06 MiB/s, done.
Resolving deltas: 100% (50/50), done.
[✓ MIT % cd hello-1
[✓ hello-1 % ls
README.md      css          favicon.png    index.html    screenshot2.png
[✓ hello-1 % █
```

# GitHub Cycle: Demonstration (8/12)



The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, demonstrating the GitHub Cycle workflow.

**Explorer View:** On the left, the Explorer view shows the file structure of a repository named "hello-1". The "index.html" file is currently selected and highlighted in blue. Other files visible include ".gitignore", "favicon.png", "README.md", and "screenshot2.png".

**Editor View:** The main editor area displays the content of "index.html". The code includes a link to a Google Fonts CSS file, a link to a local CSS file, and a link to a favicon. It features a title, a header section with a name, and an article section containing two paragraphs. The footer includes a meta list item with a mailto link.

```
index.html — MIT
index.html
hello-1 > index.html > ...
5 <link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro" rel="stylesheet" type="text/css" />
6 <link rel="stylesheet" type="text/css" href="css/style2.css" />
7 <link rel="icon" href="favicon.png" type="image/x-icon" />
8 <title>Hello World</title>
9 </head>
10 <body>
11   <div class="wrapper">
12     <div class="container">
13       <header>
14         <h1 class="name">jlord</h1>
15       </header>
16       <article>
17         <p>I'm an <a href="http://en.wikipedia.org/wiki/OpenSource">Open Source</a> developer.
18         <p>Previously I was an <a href="http://www.bostonreality.com">real estate agent</a> in Boston.
19       </article>
20       <footer>
21         <ul class="meta inline-list">
22           <li><a href="mailto:jlord@gmail.com">jlord@gmail.com</a></li>
23         </ul>
24       </footer>
25     </div>
26   </div>
27 </body>
28 </html>
```

# GitHub Cycle: Demonstration (9/12)

The screenshot shows a code editor interface with the following details:

- File Explorer:** On the left, under the "OPEN EDITOR" section, there is one unsaved file: "index.html".
- Project Structure:** The "EXPLORER" sidebar shows a project structure:
  - MIT
  - hello-1
    - ERL
    - Expenses2019
    - Expenses2020
    - hello-1
      - css
      - .gitignore
      - favicon.png
      - index.html
      - README.md
      - screenshot2.png
    - Projects
    - Proposals
- Code Editor:** The main area displays the content of "index.html".

```
index.html — MIT
index.html
hello-1 > index.html > html > body > div.wrapper > div.container > header
5   <link href="http://fonts.googleapis.com/css?family=Source+Sans+Pro" rel="stylesheet" type="text/css"/>
6   <link rel="stylesheet" type="text/css" href="css/style2.css"/>
7   <link rel="icon" href="favicon.png" type="image/x-icon"/>
8   <title>Hello World</title>
</head>
<body>
<div class="wrapper">
<div class="container">
<header>
  <h1 class="name">Hello World</h1>
</header>
  abc World
<article>
  abc world
    <p>I'm an <a href="http://en.wikipedia.org/wiki/Open</a></p>
    <p>Previously I was an <a href="http://www.bostonre</a></p>
</article>
<footer>
  <ul class="meta inline-list">
    <li><a href="mailto:te_ilord@gmail.com">te_ilord@gmail.com</a></li>
  </ul>
</footer>
```

# GitHub Cycle: Demonstration (10/12)

```
[✓ MIT % cd hello-1
[✓ hello-1 % ls
 README.md      css          favicon.png    index.html    screenshot2.png
[✓ hello-1 % git add .
[✓ hello-1 % git status
On branch gh-pages
Your branch is up to date with 'origin/gh-pages'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index.html

[✓ hello-1 % git commit -m 'HelloWorld Edit'
[gh-pages f45de05] HelloWorld Edit
  1 file changed, 73 insertions(+), 31 deletions(-)
    rewrite index.html (89%)
[✓ hello-1 % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.04 KiB | 1.04 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:johntango/hello-1.git
  1a43ac6..f45de05  gh-pages -> gh-pages
[✓ hello-1 % ]
```

# GitHub Cycle: Demonstration (11/12)

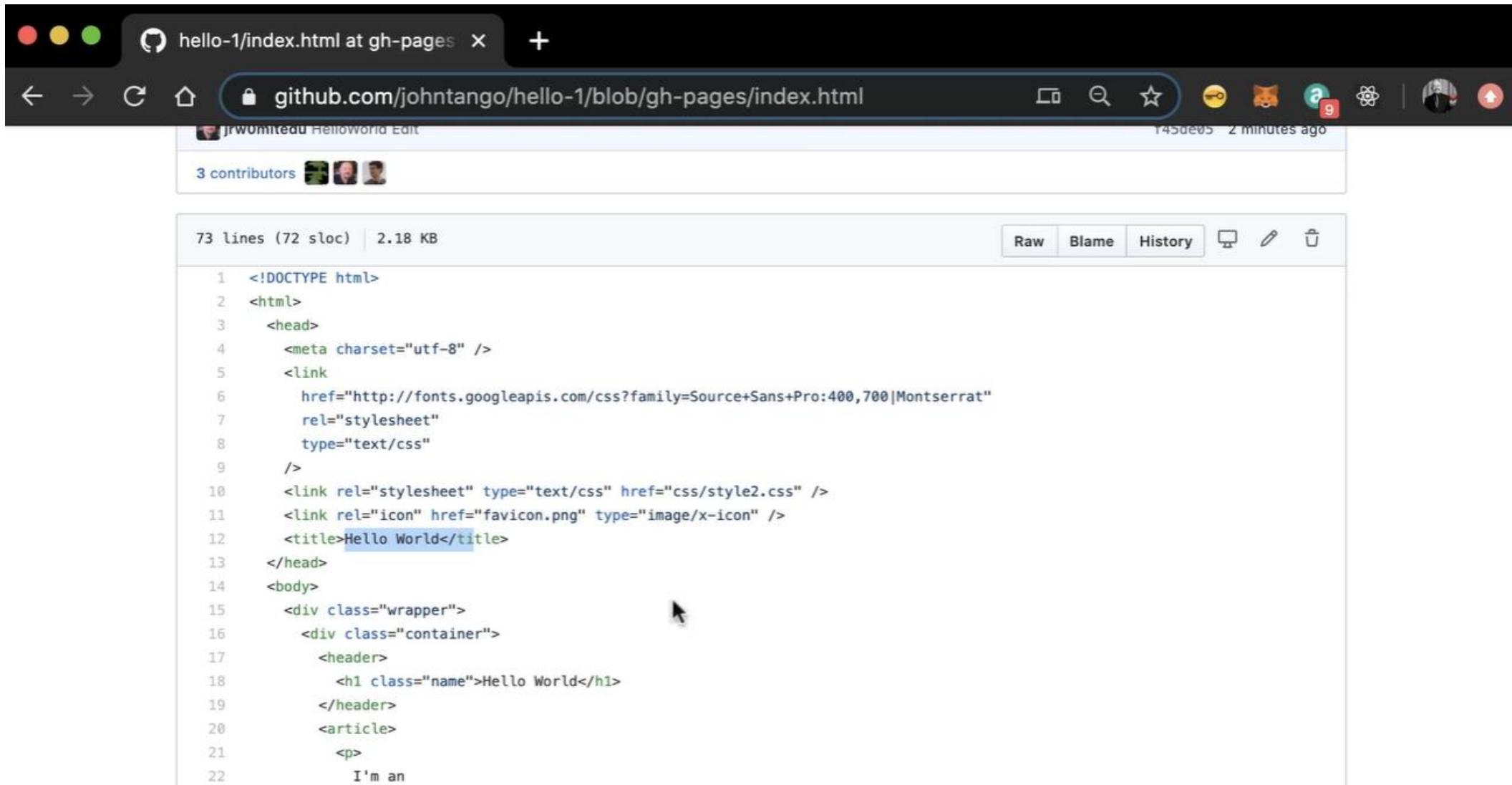
The screenshot shows a GitHub repository page for `johntango/hello-1`. The repository has 30 commits, 3 branches, 0 packages, 0 releases, 1 environment, and 3 contributors. The current branch is `gh-pages`, which is 1 commit ahead of `jlord:gh-pages`. A pull request button is available. The commit history includes:

- `jrw0mitedu` HelloWorld Edit (Latest commit f45de05 1 minute ago)
- `css` Remove characters that render differently (2 years ago)
- `.gitignore` get started (6 years ago)
- `README.md` Update image in readme (2 years ago)
- `favicon.png` get started (6 years ago)
- `index.html` HelloWorld Edit (1 minute ago)
- `screenshot2.png` edit readme for other style (6 years ago)

The `README.md` file content is displayed below:

```
Hello Page
```

# GitHub Cycle: Demonstration (12/12)

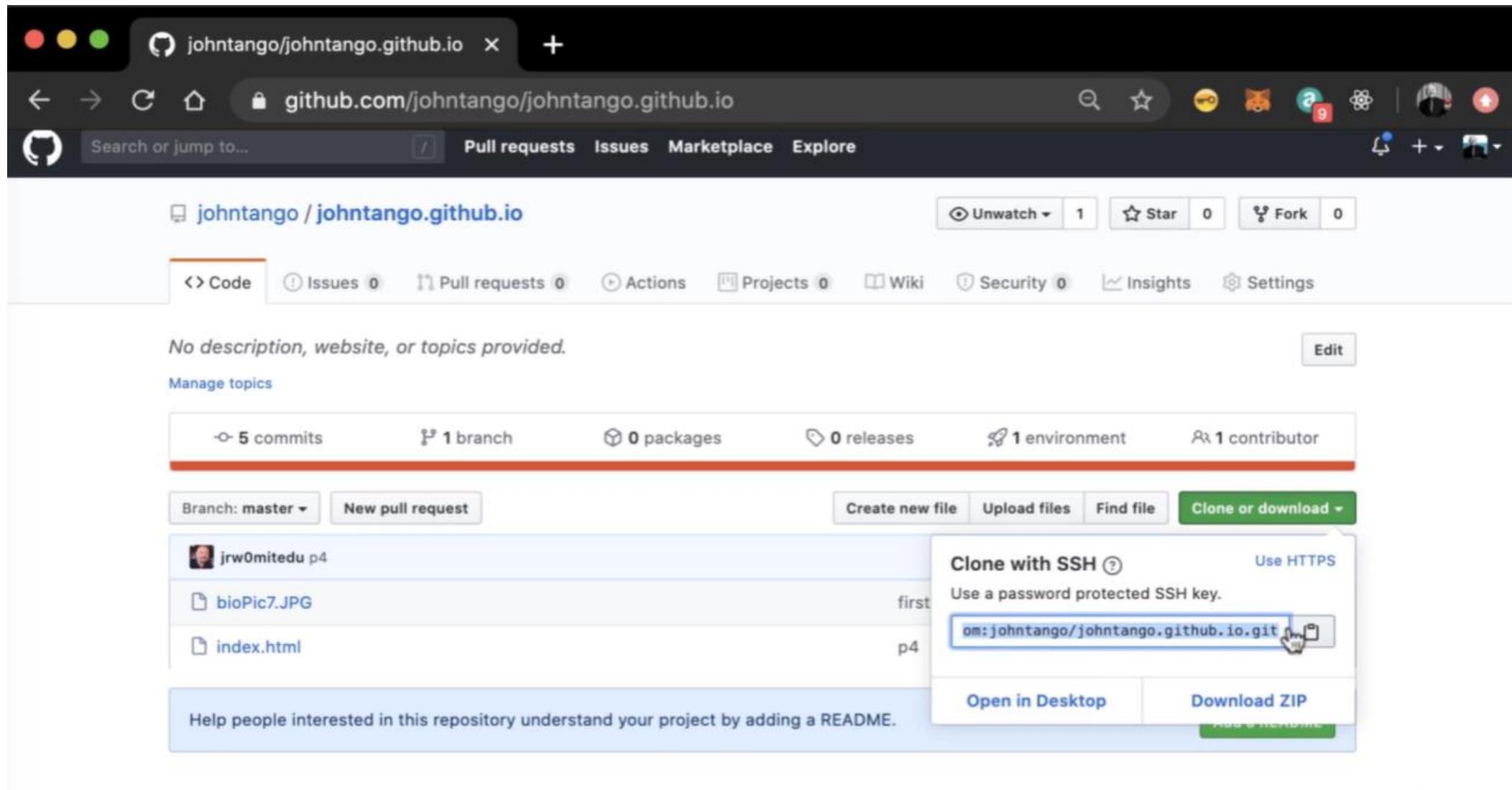


The screenshot shows a GitHub commit page for the file `index.html` in the repository `johntango/hello-1`. The commit was made by `jrwojitedu` and is titled `HelloWorld Edit`. It has 3 contributors. The commit message is `145de05 - 2 minutes ago`.

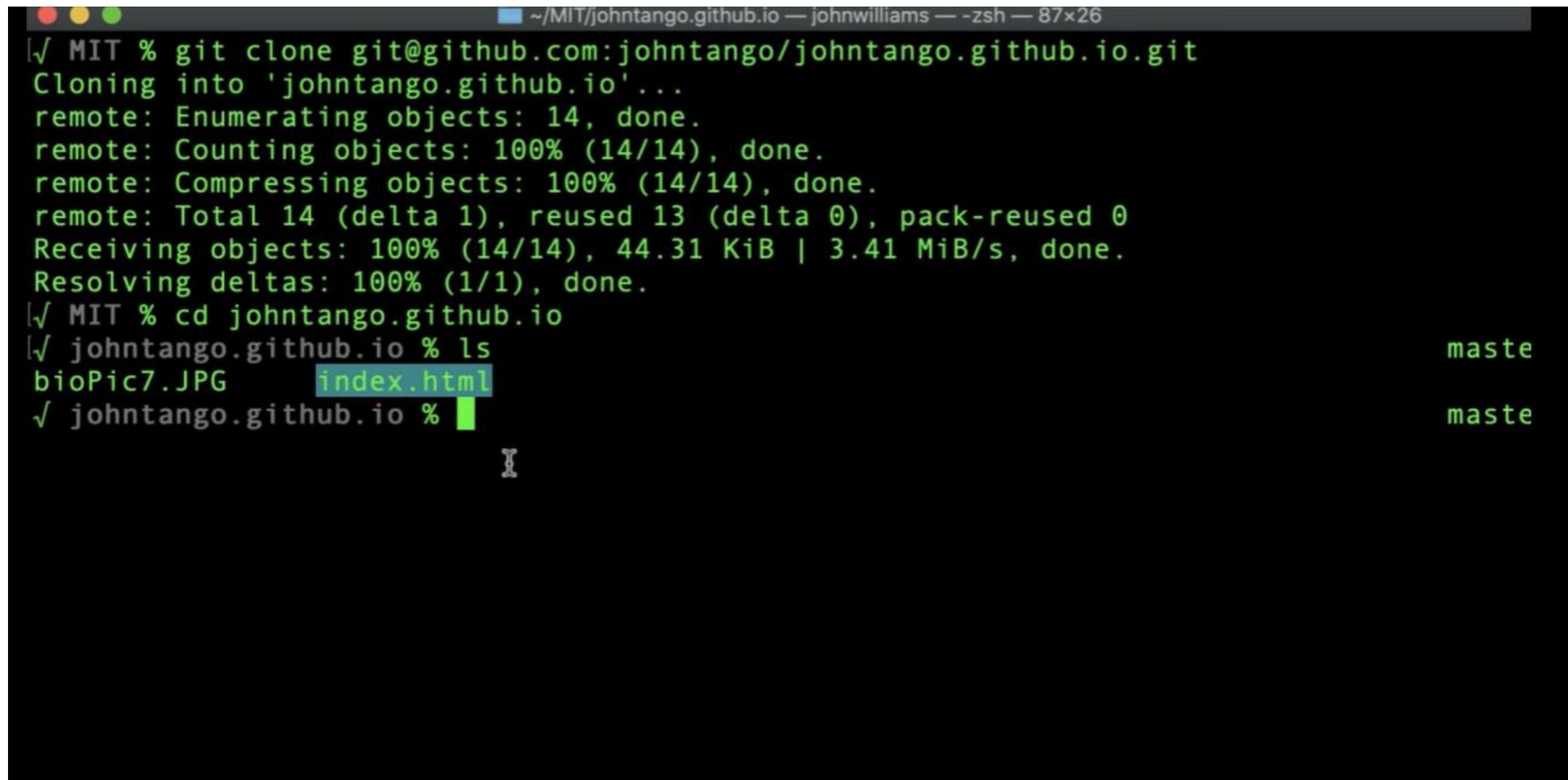
The code editor displays the following HTML content:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <link
6       href="http://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700|Montserrat"
7       rel="stylesheet"
8       type="text/css"
9     />
10    <link rel="stylesheet" type="text/css" href="css/style2.css" />
11    <link rel="icon" href="favicon.png" type="image/x-icon" />
12    <title>Hello World</title>
13  </head>
14  <body>
15    <div class="wrapper">
16      <div class="container">
17        <header>
18          <h1 class="name">Hello World</h1>
19        </header>
20        <article>
21          <p>
22            I'm an
```

# VS Code GitHub Integration (1/8)

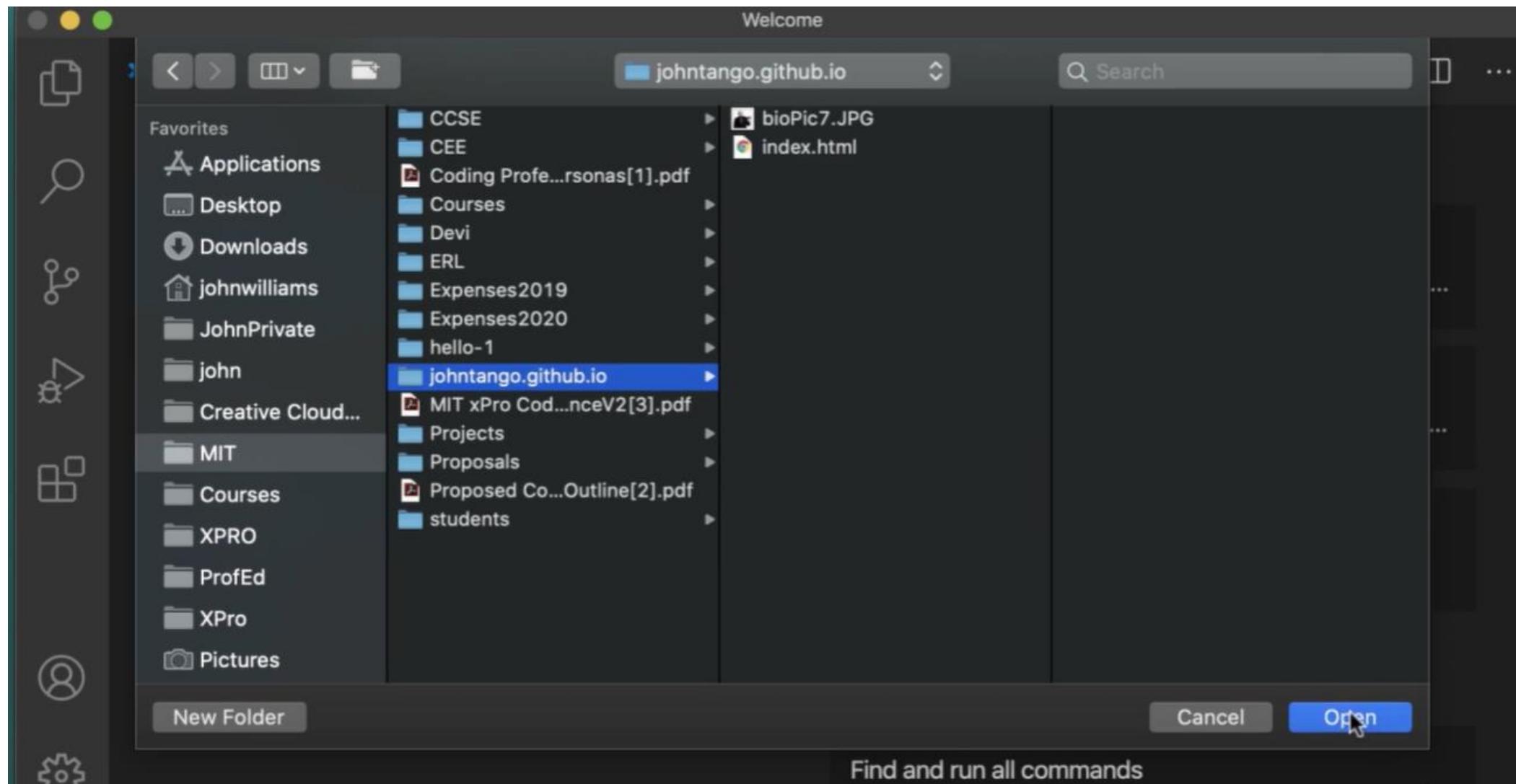


## VS Code GitHub Integration (2/8)



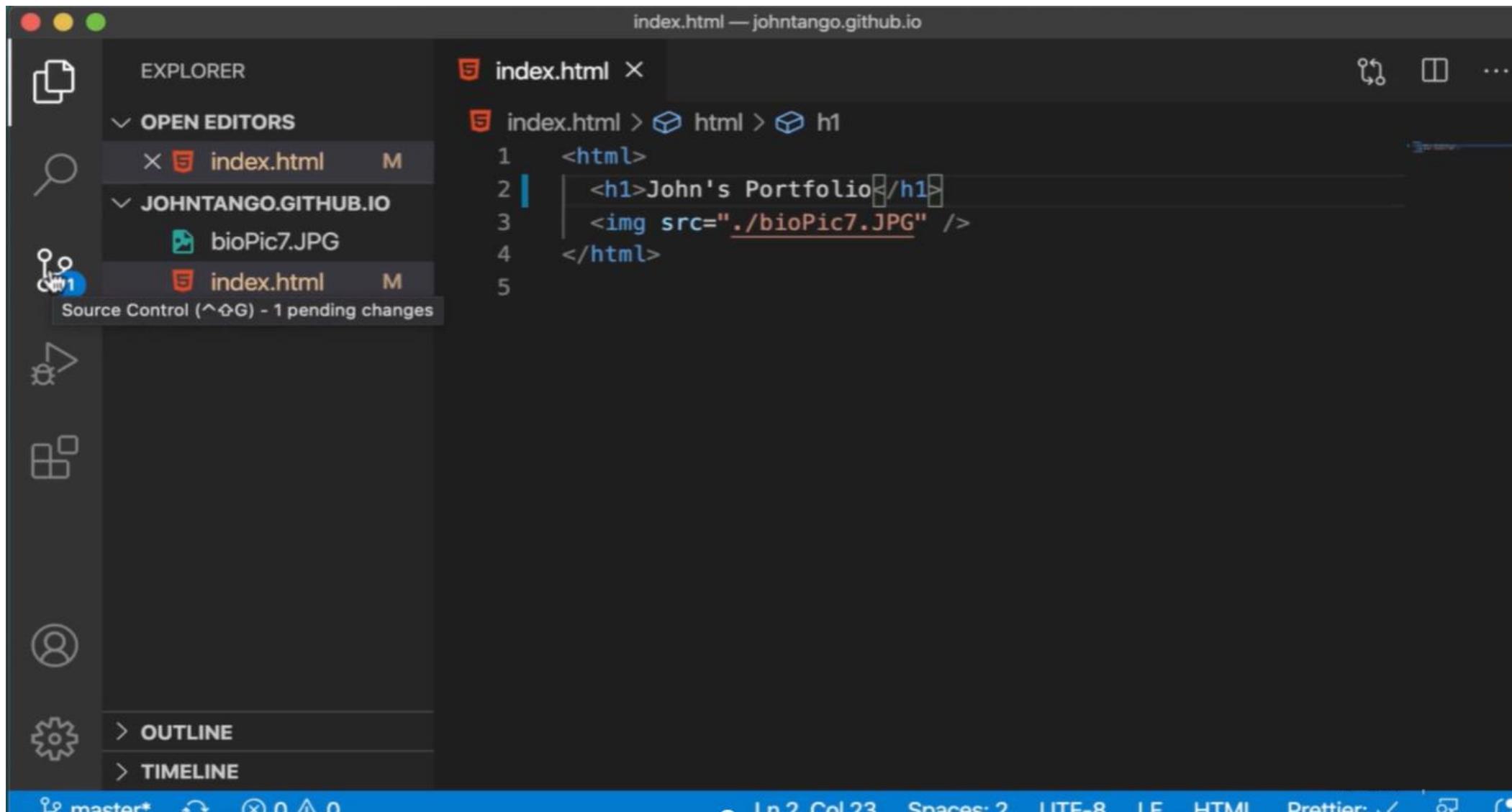
```
~/MIT/johntango.github.io — johnwilliams — -zsh — 87x26
[✓ MIT % git clone git@github.com:johntango/johntango.github.io.git
Cloning into 'johntango.github.io'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 14 (delta 1), reused 13 (delta 0), pack-reused 0
Receiving objects: 100% (14/14), 44.31 KiB | 3.41 MiB/s, done.
Resolving deltas: 100% (1/1), done.
[✓ MIT % cd johntango.github.io
[✓ johntango.github.io % ls
bioPic7.JPG      index.html
[✓ johntango.github.io %
```

## VS Code GitHub Integration (3/8)



Find and run all commands

## VS Code GitHub Integration (4/8)



The screenshot shows the Visual Studio Code interface with a dark theme. The title bar indicates the file is "index.html — johntango.github.io". The left sidebar (Explorer) shows a project structure under "JOHTANGO.GITHUB.IO" containing "bioPic7.JPG" and "index.html". The status bar at the bottom shows "master\*". The main editor area displays the following HTML code:

```
1 <html>
2 | <h1>John's Portfolio</h1>
3 | 
4 </html>
5
```

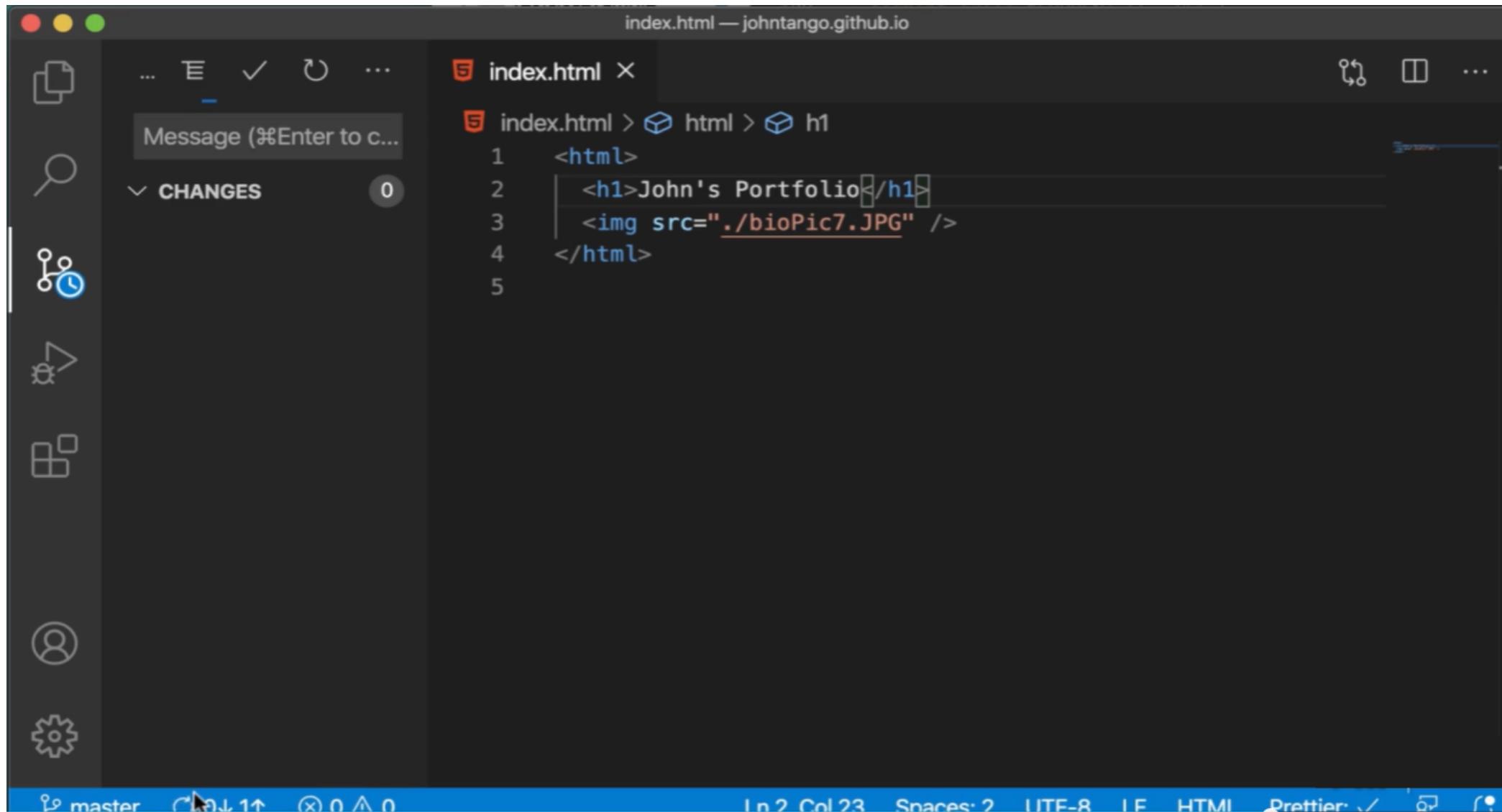
## VS Code GitHub Integration (5/8)

The screenshot shows the VS Code interface with GitHub integration. The title bar indicates the file is 'index.html — johntango.github.io'. The left sidebar has icons for file operations, search, repository, issues, pull requests, and settings. The main area shows the code editor with the following content:

```
index.html — johntango.github.io
index.html ×
index.html > html > h1
1 <html>
2 | <h1>John's Portfolio</h1>
3 | 
4 </html>
5
```

The code editor has a status bar at the bottom showing 'master\*' in the top-left, and 'Ln 2 Col 23 Spaces: 2 LITE-8 LF HTML Prettier' in the bottom-right.

## VS Code GitHub Integration (6/8)



The screenshot shows the Visual Studio Code interface with GitHub integration. The title bar indicates the file is "index.html — johntango.github.io". The left sidebar has icons for file operations, search, history, file tree, and user settings. The main editor area shows the following HTML code:

```
1 <html>
2 | <h1>John's Portfolio</h1>
3 | 
4 </html>
5
```

A message box in the center says "Message (⌘Enter to c...)" with a red "X" button. The bottom status bar shows the file is "index.html" at "Line 2 Col 23", encoding is "UTF-8", and the language is "HTML".

# VS Code GitHub Integration (7/8)

The screenshot shows a web browser window with the address bar displaying 'github.com/johntango/johntango.github.io'. The page content is a GitHub repository for 'johntango / johntango.github.io'. The repository has 6 commits, 1 branch, 0 packages, 0 releases, 1 environment, and 1 contributor. The latest commit was made 1 minute ago by user 'jrw0mitedu' with a commit message 'port1'. There are files 'bioPic7.JPG' and 'index.html' listed. A cursor is hovering over the 'port1' commit message. At the bottom, there is a call to action to 'Add a README'.

johntango/johntango.github.io

github.com/johntango/johntango.github.io

Search or jump to... Pull requests Issues Marketplace Explore

johntango / johntango.github.io

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

6 commits 1 branch 0 packages 0 releases 1 environment 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

bioPic7.JPG	first	23 hours ago
index.html	port1	1 minute ago

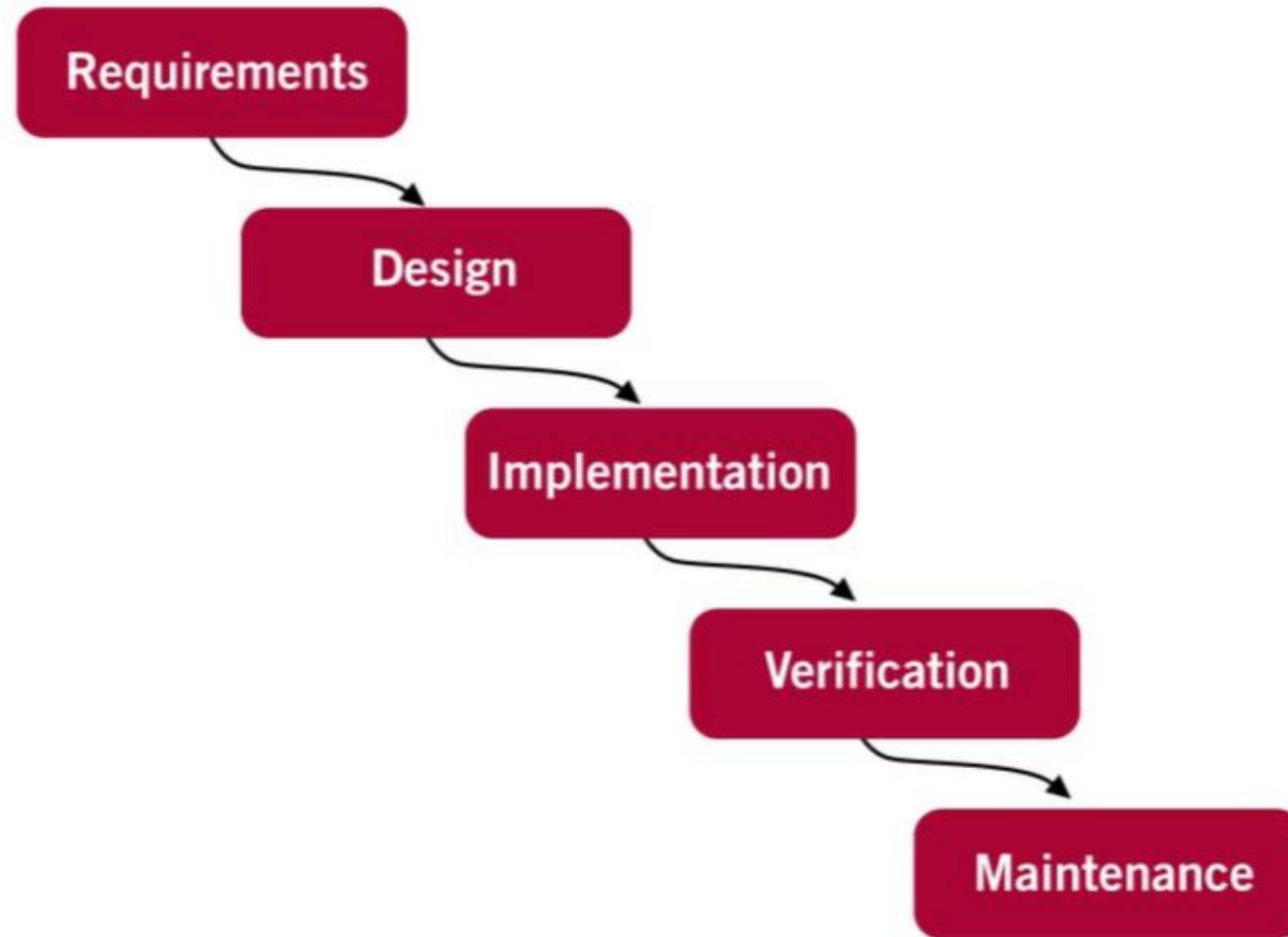
Help people interested in this repository understand your project by adding a README. Add a README

# VS Code GitHub Integration (8/8)

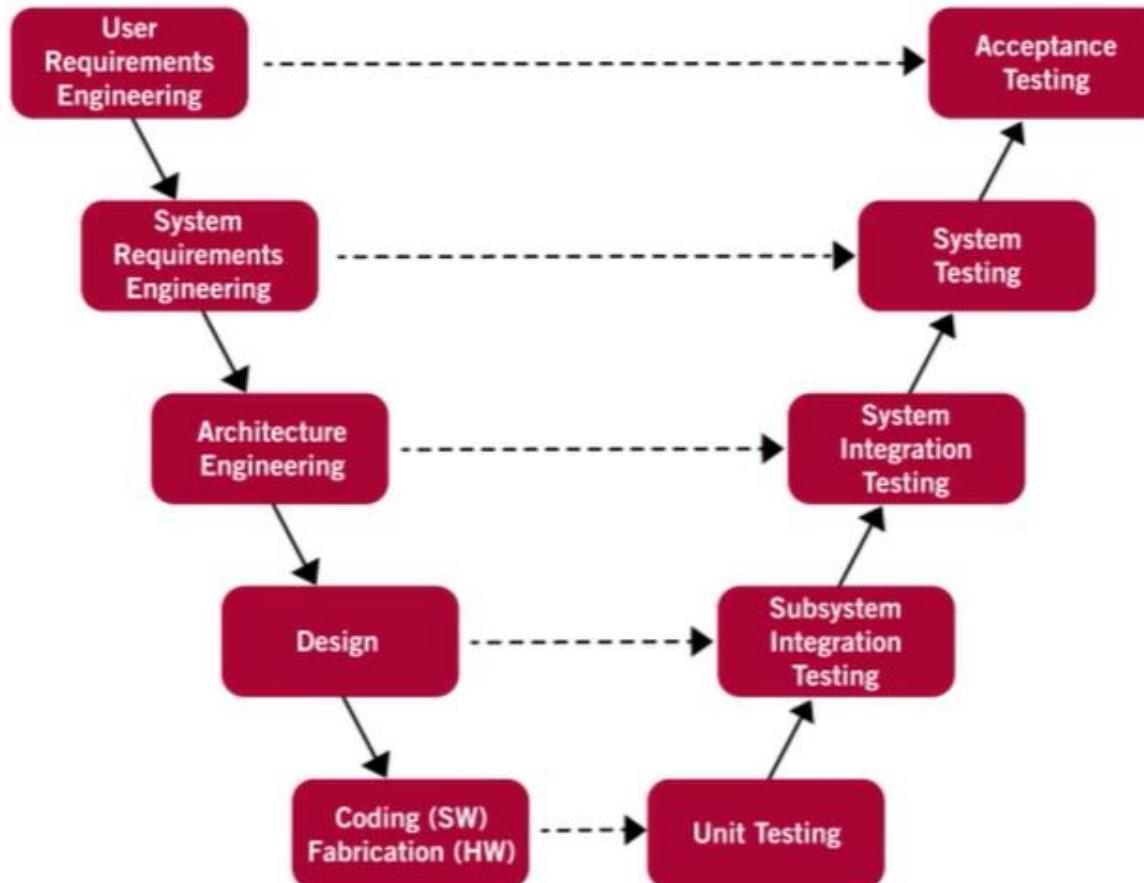
The screenshot shows a web browser window with the URL [github.com/johntango/johntango.github.io/blob/master/index.html](https://github.com/johntango/johntango.github.io/blob/master/index.html). The GitHub interface is visible, showing the repository details for `johntango / johntango.github.io`. The repository has 1 star, 0 forks, and 0 issues. The code editor view shows the contents of the `index.html` file:

```
1 <html>
2   <h1>John's Portfolio</h1>
3   
4 </html>
```

# Systems Development Life Cycle- Old School

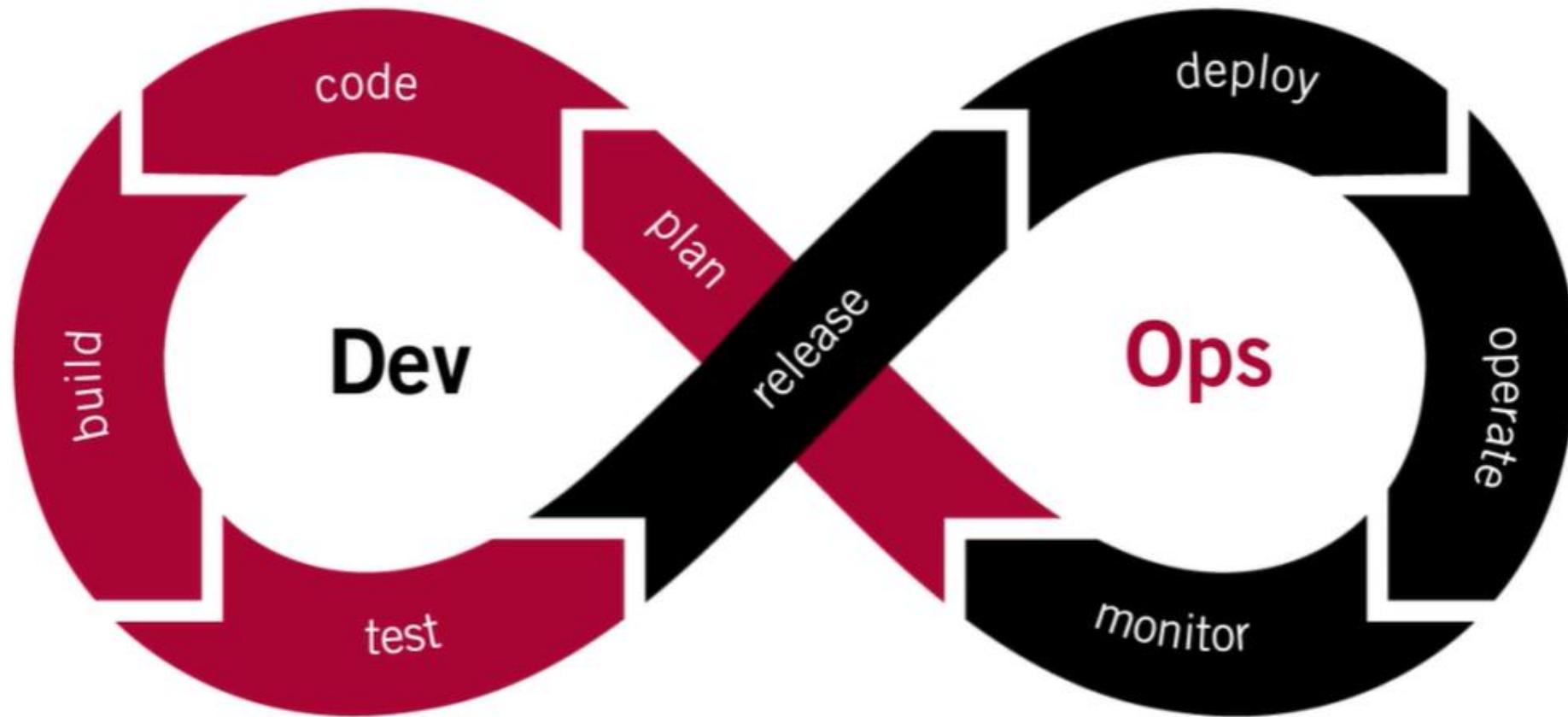


# V Models For Testing- Old School



[https://insights.sei.cmu.edu/sei\\_blog/2013/11/using-v-models-for-testing.html](https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html)

## Central To Software Creation (1/2)



Testing → QA → QE

Intelligent Testing

Start with the basics



Jest

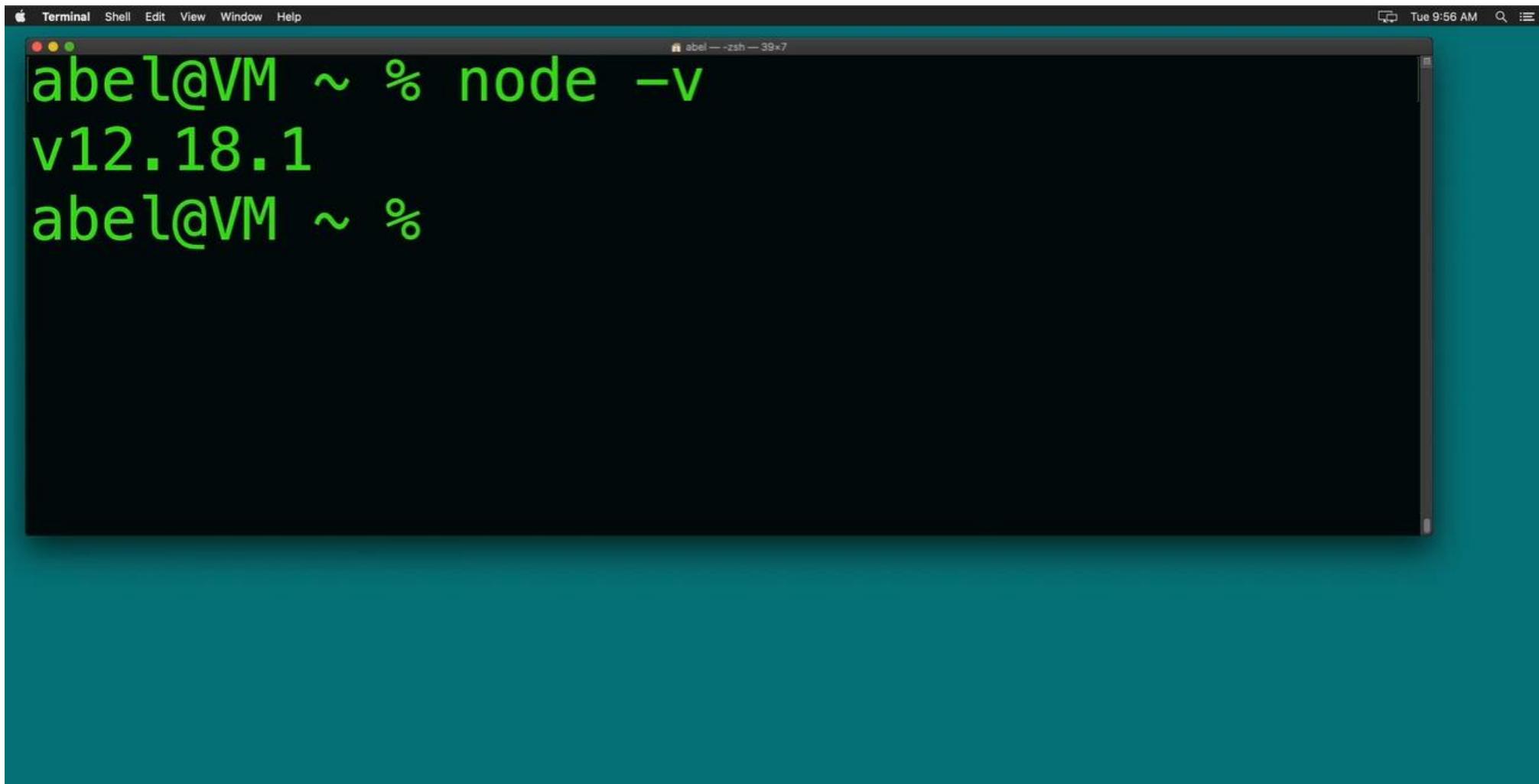
<https://jestjs.io/>

## Install Node.js (1/2)



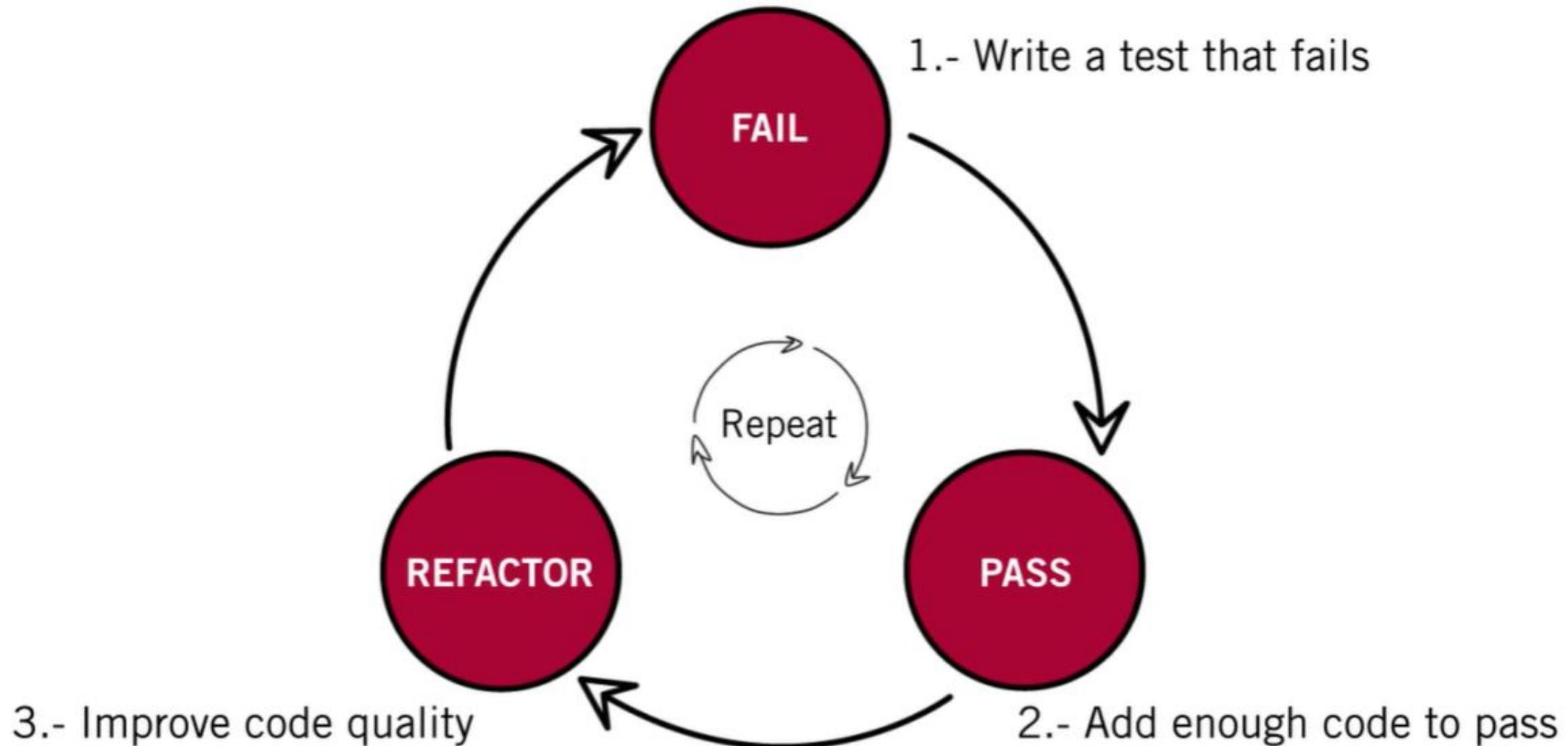
<https://nodejs.org/>

## Install Node.js (2/2)



A screenshot of a macOS Terminal window. The window title is "Terminal". The menu bar shows "Terminal", "Shell", "Edit", "View", "Window", and "Help". The status bar at the top right shows "Tue 9:56 AM". The terminal session shows the following text:  
abel@VM ~ % node -v  
v12.18.1  
abel@VM ~ %

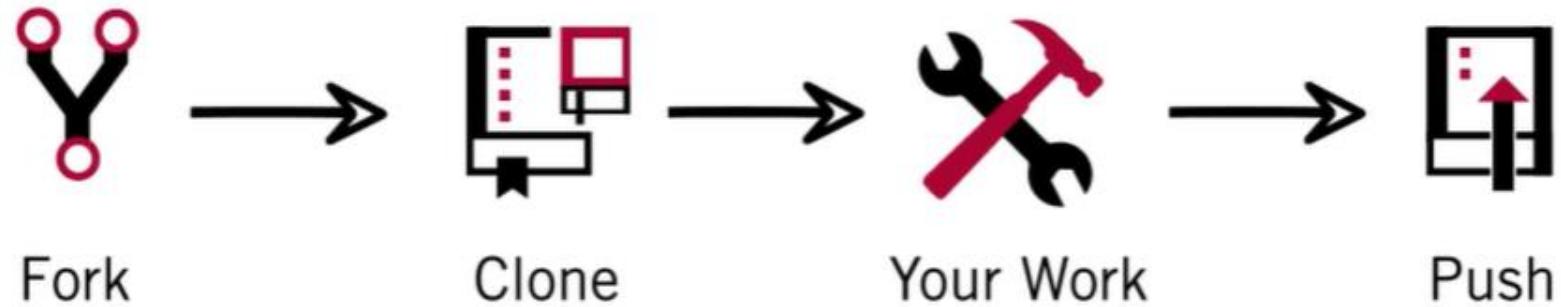
# Test Driven Development (TDD)



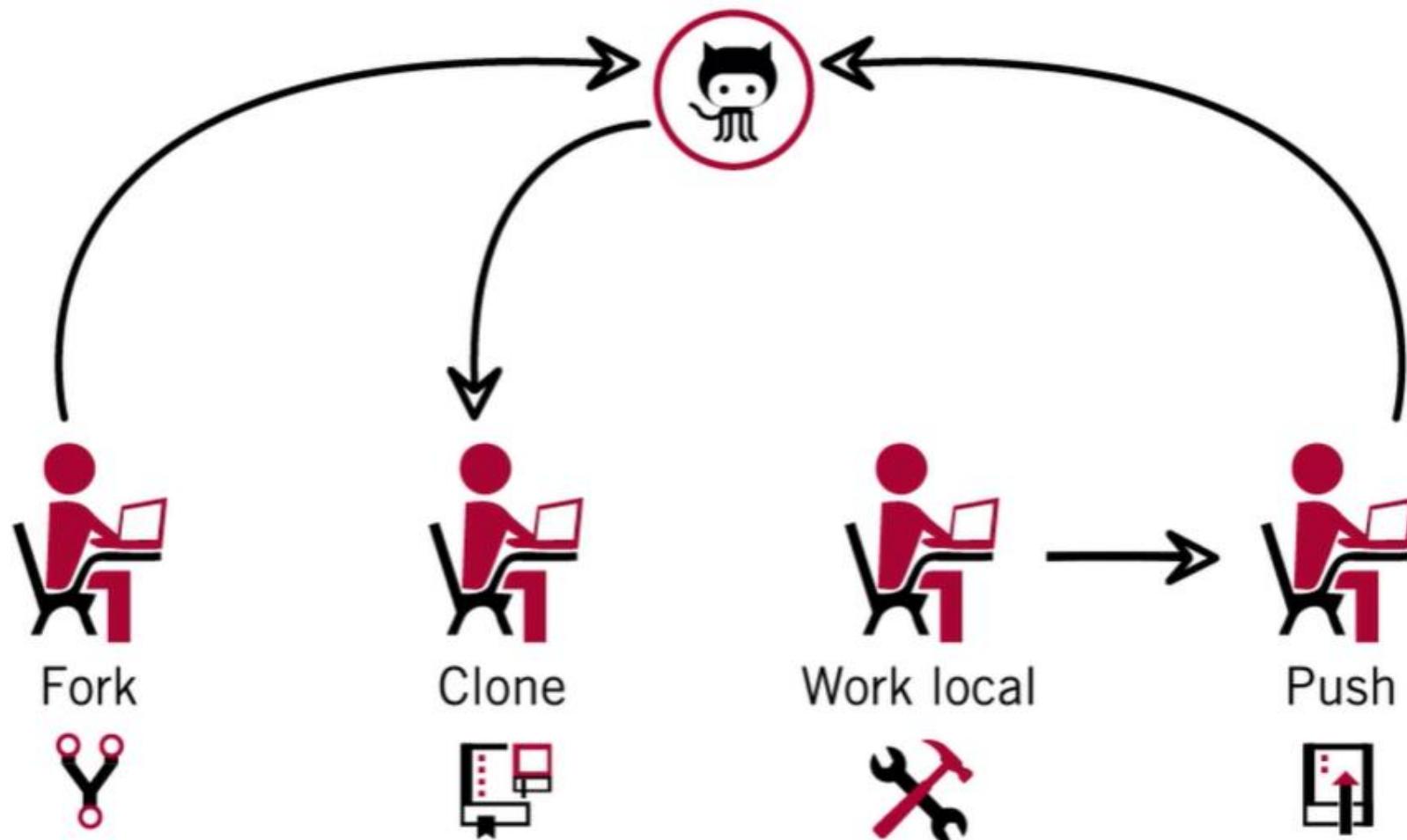
## TDD Workflow (1/3)

- Write a test for a desired software feature
- Watch it fail 
- Write the code
- Watch it fail 
- Fix
- Watch it pass 
- Feel good
- Add, commit, push

## TDD Workflow (2/3)



## TDD Workflow (3/3)



# Pushing The Changes In GitHub Repository (1/2)



<https://github.com/kogsio/greeting>

# Pushing The Changes In GitHub Repository (2/2)

The screenshot shows a web browser window with multiple tabs open. The active tab is 'peter parkerdata (peter parker)' which displays the GitHub profile of the user 'peter parker'. The profile page includes a large circular profile picture of Spider-Man's mask, the user's name 'peter parker' and handle 'peterparkerdata', a 'Edit profile' button, and follower statistics ('3 followers · 0 following · ⭐ 0'). Below the profile picture, there are sections for 'Popular repositories' (empty), '0 contributions in the last year' (empty grid), and 'Contribution settings'. The browser's address bar shows the URL 'https://github.com/kogsio/greeting'. Other tabs visible in the background include 'about:blank', 'kogsio/greeting: Testing Hello World', 'https://github.com/kogsio/greeting - Google Search', and 'peterparkerdata/greeting'.

# Creating A Fork (1/2)

The screenshot shows a GitHub repository page for the project 'kogsio/greeting'. The repository has 2 commits, 1 branch, and 0 tags. It contains files like .gitignore, LICENSE, README.md, hello.html, hello.js, and hello.test.js. The 'Code' tab is selected. On the right, there's an 'About' section with links to Readme, MIT License, Releases (none), and Languages (none).

Branch: master

Go to file Add file Clone

abelsan committed c06c565 11 hours ago

2 commits 1 branch 0 tags

File	Commit Message	Time
.gitignore	Initial commit	11 hours ago
LICENSE	Initial commit	11 hours ago
README.md	Initial commit	11 hours ago
hello.html	hello files	11 hours ago
hello.js	hello files	11 hours ago
hello.test.js	hello files	11 hours ago

About

Testing Hello World

Readme

MIT License

Releases

No releases published

Languages

# Creating A Fork (2/2)

The screenshot shows a GitHub fork page for the repository `peterparkerdata/greeting`. The page includes the following elements:

- Header:** Shows two tabs: "peterparkerdata/greeting: Test" and "about:blank". The URL is `github.com/peterparkerdata/greeting`.
- Navigation Bar:** Includes links for "Pull requests", "Issues", "Marketplace", and "Explore".
- Repository Information:** The repository name is `peterparkerdata / greeting`, and it is described as "forked from kogsio/greeting".
- Metrics:** Watch (0), Star (0), Fork (1).
- Menu Bar:** Includes "Code", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings".
- Branch Selection:** "Branch: master".
- File Operations:** Buttons for "Go to file", "Add file", and "Clone".
- Content Area:** A message stating "This branch is even with kogsio:master." followed by a "Pull request" and "Compare" button.
- Commit History:** Shows a single commit by user `abelsan` at `c06c565`, made 11 hours ago. The commit details are: "Initial commit" for files `.gitignore`, `LICENSE`, `README.md`, and `hello.html`. The commit also creates `1 branch` and `0 tags`.
- Right Sidebar:**
  - About:** Includes "Testing Hello World", "Readme", and "MIT License".
  - Releases:** No releases published. Create a new release.
  - Packages:** (empty)

# Creating A Clone (1/4)

The screenshot shows a GitHub repository page for `peterparkerdata/greeting`. The repository was forked from `kogsio/greeting`. The `Code` tab is selected. On the right, there's a **Clone** dropdown menu with options for **Clone with HTTPS** (selected) and **SSH**. Below the dropdown, there are links for **Open in Desktop** and **Download ZIP**. The repository's history shows a single commit by `abelsan` at `c06c565` 11 hours ago. The commit message is "Initial commit". The repository contains files: `.gitignore`, `LICENSE`, `README.md`, and `hello.html`.

**Code**    Pull requests    Actions    Projects    Wiki    Security    Insights    Settings

Branch: master

This branch is even with kogsio:master.

abelsan committed c06c565 11 hours ago

.gitignore    Initial commit

LICENSE    Initial commit

README.md    Initial commit

hello.html    hello files

Clone with HTTPS ⓘ    SSH

https://github.com/peterparkerdata/

Open in Desktop    Download ZIP

About

Testing Hello World

Readme

MIT License

Releases

No releases published

Create a new release

Packages

# Creating A Clone (2/4)

The screenshot shows a GitHub repository page for `peterparkerdata/greeting`. The repository is a fork from `kogsio/greeting`. The 'Code' tab is selected. On the right, there's a 'Clone' button with a dropdown menu open. The menu includes options for 'Clone with SSH' (selected) and 'Use HTTPS'. Below the dropdown, there are links for 'Open in Desktop' and 'Download ZIP'. To the right of the dropdown, there's an 'About' section with links to 'Testing Hello World', 'Readme', and 'MIT License'. At the bottom, sections for 'Releases' and 'Packages' are visible.

`peterparkerdata / greeting`  
forked from `kogsio/greeting`

Branch: master

This branch is even with `kogsio:master`.

abelsan committed c06c565 11 hours ago

`.gitignore` Initial commit  
`LICENSE` Initial commit  
`README.md` Initial commit  
`hello.html` hello files

Go to file Add file ▾ Clone ▾

Clone with SSH Use HTTPS

git@github.com:peterparkerdata/greet ↗

Open in Desktop Download ZIP

About

Testing Hello World

Readme

MIT License

Releases

No releases published

Create a new release

Packages

## Creating A Clone (3/4)

```
abel@VM ~ % git clone git@github.com:peterparkerdata/greeting.git
Cloning into 'greeting'...
Warning: Permanently added the RSA host key for IP address '140.82.112.3' to the list of known hosts.
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 1), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), 47.56 KiB | 1.08 MiB/s
, done.
Resolving deltas: 100% (1/1), done.
abel@VM ~ % cd greeting
abel@VM greeting % clear
```

## Creating A Clone (4/4)

```
abel@VM greeting % ls
LICENSE
README.md
hello.html
hello.js
abel@VM greeting % █
```

hello.test.js  
package-lock.json  
package.json

## Installing Dependencies Required For Testing (1/)

```
abel@VM greeting % ls
LICENSE
README.md
hello.html
hello.js
abel@VM greeting % npm install
added 506 packages from 348 contributors and audited 50
6 packages in 9.919s

  15 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities

abel@VM greeting %
```

## Installing Dependencies Required For Testing (2/)

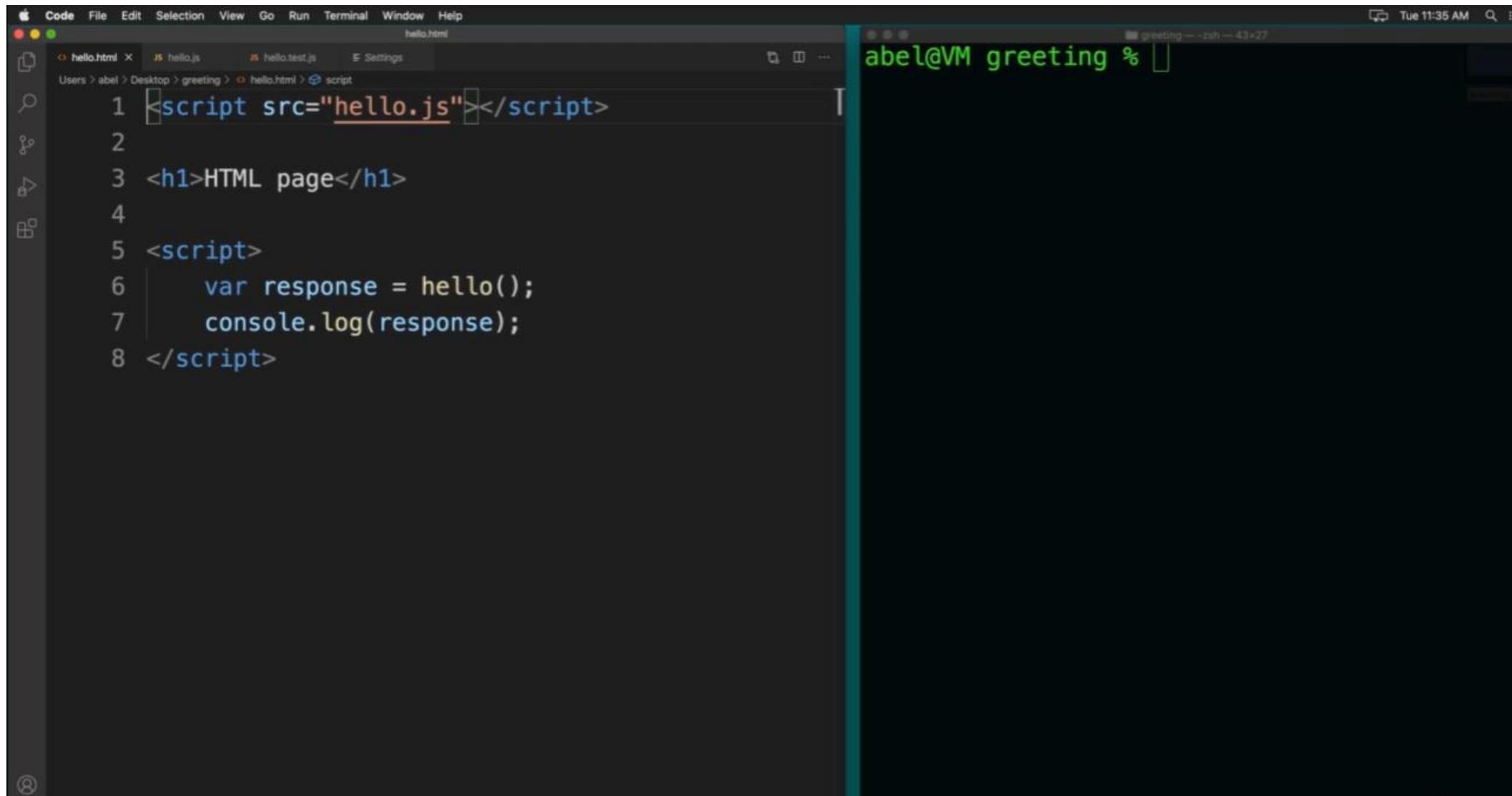
```
abel@VM greeting % npm test

> greeting@1.0.0 test /Users/abel/greeting
> jest

PASS ./hello.test.js
  ✓ outputs the correct string (4 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.962 s
Ran all test suites.
abel@VM greeting %
```

# Installing Dependencies Required For Testing: Code (1/4)

A screenshot of a Mac desktop environment. On the left, a dark-themed code editor window titled "hello.html" is open, showing the following HTML code:

```
1 <script src="hello.js"></script>
2
3 <h1>HTML page</h1>
4
5 <script>
6     var response = hello();
7     console.log(response);
8 </script>
```

The code editor's status bar shows the file path: "Users > abel > Desktop > greeting > hello.html > script". On the right, a terminal window titled "greeting" is open, showing the command prompt "abel@VM greeting %".

Tue 11:35 AM

# Installing Dependencies Required For Testing: Code (2/4)

A screenshot of a Mac desktop environment. On the left, a dark-themed code editor window titled "hello.js" is open, displaying the following JavaScript code:

```
function hello() {
  return "Hello World!";
}
```

The code editor's sidebar shows other files in the project: "hello.html", "hello.test.js", and "Settings". The file path in the sidebar is "Users > abel > Desktop > greeting > hello.js".

To the right of the code editor is a terminal window with a dark background and light-colored text. The prompt "abel@VM greeting %" is visible, indicating the user is logged in as "abel" on a virtual machine named "greeting". The terminal window has a teal vertical bar on its left side.

# Installing Dependencies Required For Testing: Code (3/4)

The image shows a Mac desktop with a dark-themed interface. On the left, a Code editor window is open, showing a file named 'hello.test.js'. The code inside is a Mocha test for a 'hello' function:

```
1 var utils = require('course-utilities');
2 var hello = utils.load('./hello.js', 'hello');
3
4 test('outputs the correct string', () => {
5   expect(hello()).toBe("Hello World!");
6});
```

On the right, a terminal window is open with the command 'greeting' entered. The output shows the user's name 'abel' and the host 'VM greeting %'.

# Installing Dependencies Required For Testing: Code (4/4)

The screenshot shows a terminal window with two panes. The left pane displays the source code for a function named 'hello' in 'hello.js'. The right pane shows the output of a test run using Jest.

```
function hello() {
  return "Hello World";
}

● outputs the correct string
  expect(received).toBe(expected) // Object.is equality
    Expected: "Hello World"
    Received: "Hello World"

  3 |   test('outputs the correct string'
, () => {
  > 5 |     expect(hello()).toBe("Hello Wor
ld!");
    |
  6 |   });
    at Object.<anonymous> (hello.test.js:5:19)

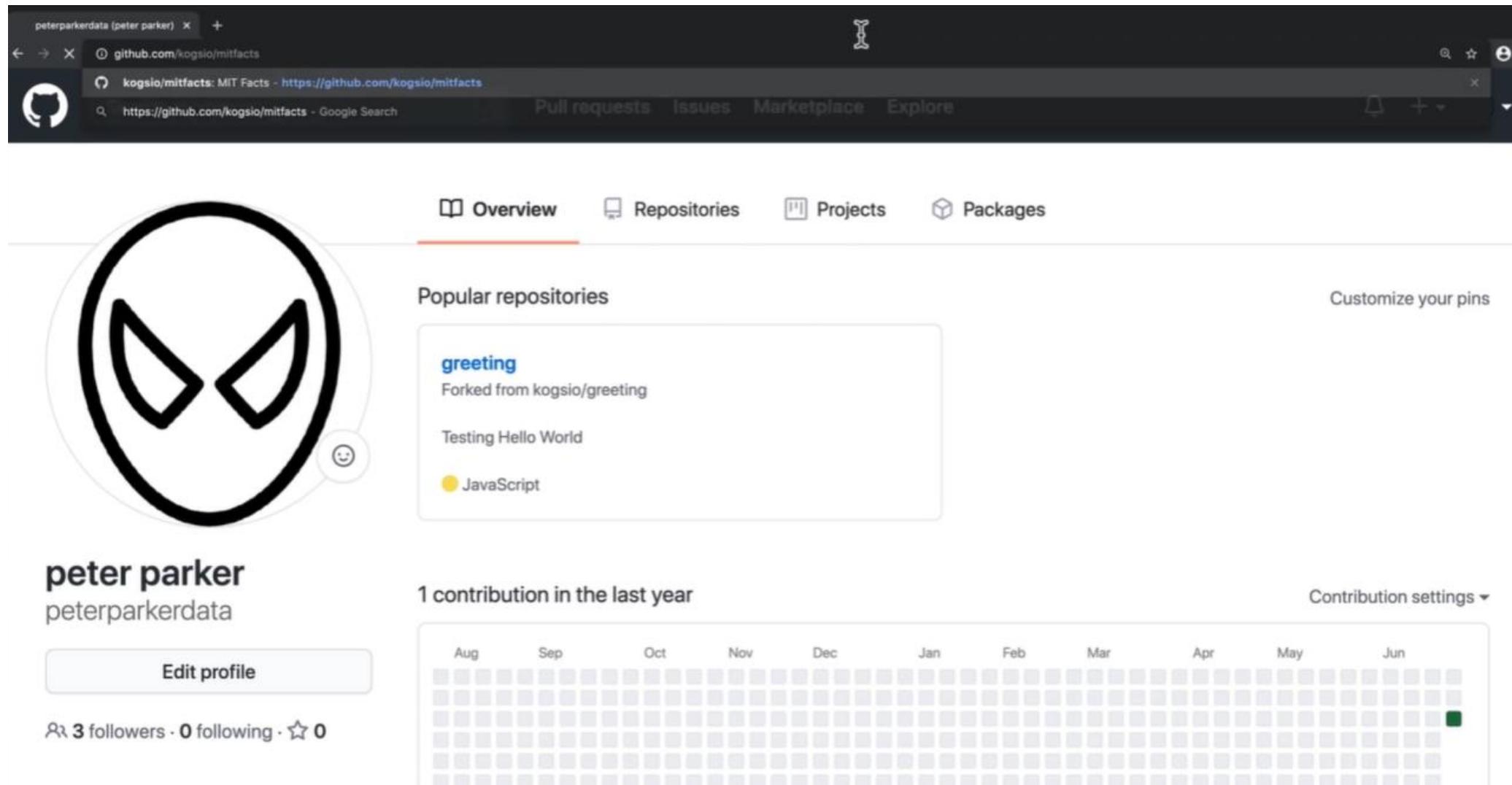
Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        0.94 s, estimated 1 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM greeting %
```

# MIT Fact Test Exercise (1/2)



<https://github.com/kogsio/mitfacts>

# MIT Fact Test Exercise (2/2)



The screenshot shows a GitHub profile page for the user `peterparkerdata`. The profile picture is a black and white graphic of Spider-Man's mask. The user's name is listed as **peter parker** and their GitHub handle as `peterparkerdata`. Below the profile picture is a button labeled "Edit profile". The user has 3 followers, 0 following, and 0 stars. The "Overview" tab is selected, showing a "Popular repositories" section with a card for the `greeting` repository, which is a fork from `kogsio/greeting`, written in JavaScript and testing Hello World. There is also a "Customize your pins" link. Below this is a "Contribution calendar" showing activity from August of the previous year to June of the current year, with a single dark green square in the June column of the current year. The top navigation bar shows the user is viewing the `kogsio/mitfacts` repository, which contains MIT Facts.

`peterparkerdata (peter parker)`

github.com/kogsio/mitfacts

`kogsio/mitfacts: MIT Facts` - <https://github.com/kogsio/mitfacts>

`https://github.com/kogsio/mitfacts` - Google Search

Pull requests Issues Marketplace Explore

**Overview** Repositories Projects Packages

Popular repositories

**greeting**  
Forked from `kogsio/greeting`

Testing Hello World

JavaScript

Customize your pins

**peter parker**  
`peterparkerdata`

Edit profile

3 followers · 0 following · 0 stars

1 contribution in the last year

Contribution settings ▾

Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun

# MIT Fact Test Exercise: Creating A Fork (1/2)

The screenshot shows a GitHub repository page for the user 'kogsio' named 'mitfacts'. The repository has 1 watch, 0 stars, and 0 forks. The 'Code' tab is selected. The master branch has 2 commits by user 'abelsan' made 13 hours ago. The commits are:

File	Message	Time
.gitignore	Initial commit	13 hours ago
LICENSE	Initial commit	13 hours ago
README.md	Initial commit	13 hours ago
mit.html	mit facts and test	13 hours ago
mit.js	mit facts and test	13 hours ago
mit.test.js	mit facts and test	13 hours ago

**About**

- MIT Facts
- Readme
- MIT License

**Releases**

No releases published

**Languages**

# MIT Fact Test Exercise: Creating A Fork (2/2)

The screenshot shows a GitHub repository page for `peterparkerdata/mitfacts`. The repository was forked from `kogsio/mitfacts`. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. On the right, there are buttons for Watch (0), Star (0), Fork (1), and a bell icon. Below the navigation, there are links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Code tab is selected. The main content area shows a dropdown for Branch: master, buttons for Go to file, Add file, and Clone, and a message stating "This branch is even with kogsio:master." It also shows a commit by `abelsan` with hash `c95fa12` made 13 hours ago, which included commits to `.gitignore`, `LICENSE`, `README.md`, and `mit.html`. The sidebar on the right contains sections for About (with a gear icon), MIT Facts (with a gear icon), Readme, and MIT License. The Releases section indicates "No releases published" and "Create a new release".

`peterparkerdata / mitfacts`  
forked from `kogsio/mitfacts`

Code Pull requests Actions Projects Wiki Security Insights Settings

Branch: master Go to file Add file Clone

This branch is even with `kogsio:master`.

`abelsan` committed `c95fa12` 13 hours ago

2 commits 1 branch 0 tags

File	Commit Message	Time
<code>.gitignore</code>	Initial commit	13 hours ago
<code>LICENSE</code>	Initial commit	13 hours ago
<code>README.md</code>	Initial commit	13 hours ago
<code>mit.html</code>	mit facts and test	13 hours ago

About MIT Facts   
Readme MIT License

Releases  
No releases published [Create a new release](#)

# MIT Fact Test Exercise: Creating A Clone (1/5)

The screenshot shows a GitHub repository page for `peterparkerdata/mitfacts`. The repository was forked from `kogsio/mitfacts`. The `Code` tab is selected. On the right, there's a green `Clone` button with a dropdown menu. The dropdown menu has two options: `Clone with SSH` and `Use HTTPS`. The `git@github.com:peterparkerdata/mitf` field contains a partially typed SSH URL. Below the dropdown are two buttons: `Open in Desktop` and `Download ZIP`.

`peterparkerdata / mitfacts`  
forked from `kogsio/mitfacts`

Branch: master

This branch is even with `kogsio:master`.

abelsan committed c95fa12 13 hours ago

`.gitignore` Initial commit  
`LICENSE` Initial commit  
`README.md` Initial commit  
`mit.html` mit facts and test

Go to file Add file `Clone` Use HTTPS

Open in Desktop Download ZIP

About MIT Facts

Readme MIT License

Releases No releases published Create a new release

## MIT Fact Test Exercise: Creating A Clone (2/5)

```
Last login: Tue Jun 30 11:21:44 on ttys000
abel@VM ~ % cd Desktop
abel@VM Desktop % git clone git@github.com:peterparker
data/mitfacts.git
Cloning into 'mitfacts'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
Receiving objects: 100% (12/12), 47.79 KiB | 3.19 MiB/
s, done.
Resolving deltas: 100% (1/1), done.
remote: Total 12 (delta 1), reused 7 (delta 0), pack-r
eused 0
abel@VM Desktop % cd mitfacts
abel@VM mitfacts % clear
```

## MIT Fact Test Exercise: Creating A Clone (3/5)

```
abel@VM mitfacts % npm install
added 506 packages from 348 contributors and audited 5
06 packages in 12.594s
```

```
15 packages are looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
abel@VM mitfacts % npm test
```

```
> mitfacts@1.0.0 test /Users/abel/Desktop/mitfacts
> jest
```



## MIT Fact Test Exercise: Creating A Clone (4/5)

FAIL ./mit.test.js

MIT

- ✓ Object properties present (2 ms)
- ✓ City match (1 ms)
- ✗ Colors match (2 ms)
- ✗ Founded range (1 ms)
- ✗ Motto

● MIT > Colors match

```
expect(received).toContain(expected) // indexOf
```

Expected value: "Silver Gray"

Received array: ["pink", "red"]

## MIT Fact Test Exercise: Creating A Clone (5/5)

```
  25 |     test('Motto', () => {
> 26 |       expect(mit.motto).toMatch('Mens');
|         ^
27 |     });
28 |   });

at Object.<anonymous> (mit.test.js:26:27)
```

```
Test Suites: 1 failed, 1 total
Tests:       3 failed, 2 passed, 5 total
Snapshots:   0 total
Time:        1.989 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM mitfacts %
```

# MIT Fact Test Exercise: Testing (1/10)

The screenshot shows a terminal window with two panes. The left pane displays a portion of an HTML file (`mit.html`) containing a script tag that points to a JavaScript file (`mit.js`). The right pane shows the output of a test run.

```
● MIT > Motto
expect(received).toMatch(expected)
Expected substring: "Mens"
Received string:    "Carpe diem"

24 |
25 |     test('Motto', () => {
> 26 |         expect(mit.motto).toMatch(
h('Mens'));
27 |     });
28 |

at Object.<anonymous> (mit.test.js:26:27)

Test Suites: 1 failed, 1 total
Tests:       3 failed, 2 passed, 5 total
Snapshots:   0 total
Time:        0.919 s, estimated 1 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM mitfacts %
```

# MIT Fact Test Exercise: Testing (2/10)

The image shows a Mac OS X desktop environment with two windows open. On the left is a dark-themed code editor (Sublime Text) with three tabs: 'mit.html', 'mit.js', and 'mit.test.js'. The 'mit.js' tab contains the following JavaScript code:

```
1 var mit = {  
2     city: 'Cambridge',  
3     colors: ['pink', 'red'],  
4     mascot: "robot",  
5     founded: 1900,  
6     motto: 'Carpe diem',  
7 };  
8
```

On the right is a terminal window titled 'mitfacts -- zsh -- 43x27' with the following output:

```
● MIT → Motto  
expect(received).toMatch(expected)  
Expected substring: "Mens"  
Received string:    "Carpe diem"  
24 |  
25 |     test('Motto', () => {  
26 |         expect(mit.motto).toMatch(  
27 |             'Mens');  
28 |     });  
     at Object.<anonymous> (mit.test.js:26:  
:27)  
  
Test Suites: 1 failed, 1 total  
Tests:       3 failed, 2 passed, 5 total  
Snapshots:   0 total  
Time:        0.919 s, estimated 1 s  
Ran all test suites.  
npm ERR! Test failed. See above for more details.  
abel@VM mitfacts %
```

# MIT Fact Test Exercise: Testing (3/10)

The terminal window shows the following output:

```
● MIT > Motto
expect(received).toMatch(expected)

Expected substring: "Mens"
Received string:    "Carpe diem"

24 |
25 |     test('Motto', () => {
> 26 |         expect(mit.motto).toMatch(
h('Mens'));
27 |     });
28 |

at Object.<anonymous> (mit.test.js:26:27)

Test Suites: 1 failed, 1 total
Tests:       3 failed, 2 passed, 5 total
Snapshots:  0 total
Time:        0.919 s, estimated 1 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM mitfacts %
```

The terminal indicates a failure in the 'Motto' test suite. The expected substring 'Mens' did not match the received string 'Carpe diem'. The failing test code is shown in the terminal output.

The left side of the image shows a code editor with the file `mit.test.js` open. The code contains several test cases for the `mit` object, including properties like `city`, `colors`, `founded`, and the `motto`. The `motto` test is currently failing, as indicated by the red error message in the terminal.

# MIT Fact Test Exercise: Testing (4/10)

The screenshot shows a terminal window with two panes. The left pane displays a file named `mit.js` containing the following code:

```
1 var mit = {
2   city: 'Cambridge',
3   colors: ['pink', 'red'],
4   mascot: "robot",
5   founded: 1861,
6   motto: 'Carpe diem',
7 };
8
```

The right pane shows the output of a test run using `npm test`. It includes the test code, a failed assertion, and the test summary.

```
● MIT > Motto
expect(received).toMatch(expected)
Expected substring: "Mens"
Received string:    "Carpe diem"

24 |
25 |
> 26 |     test('Motto', () => {
27 |       expect(mit.motto).toMatch(
28 |         'Mens');
      ^
      |
27 |     });
28 |   });

at Object.<anonymous> (mit.test.js:26:27)

Test Suites: 1 failed, 1 total
Tests:       3 failed, 2 passed, 5 total
Snapshots:  0 total
Time:        0.919 s, estimated 1 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM mitfacts %
```

# MIT Fact Test Exercise: Testing (5/10)

The screenshot shows a terminal window with two panes. The left pane displays a JavaScript file named `mit.js` containing the following code:

```
1 var mit = [
2   city: 'Cambridge',
3   colors: ['pink', 'red'],
4   mascot: "robot",
5   founded: 1861,
6   motto: 'Carpe diem',
7 ];
8
```

The right pane shows the output of a test run using Node.js's built-in testing framework, `jasmine`. The test suite is named `MIT > Motto`. It includes a single test case that fails because the expected motto ('Mens') does not match the received string ('Carpe diem'). The test code is as follows:

```
● MIT > Motto
expect(received).toMatch(expected)
Expected substring: "Mens"
Received string:    "Carpe diem"

24 |
25 |     test('Motto', () => {
> 26 |       expect(mit.motto).toMatch(
h('Mens'));
27 |     });
28 |   });

at Object.<anonymous> (mit.test.js:26:27)

Test Suites: 1 failed, 1 total
Tests:       2 failed, 3 passed, 5 total
Snapshots:  0 total
Time:        1.004 s
Ran all test suites.
npm ERR! Test failed. See above for more details.
abel@VM mitfacts %
```

## MIT Fact Test Exercise: Testing (6/10)

```
abel@VM mitfacts % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   mit.js

no changes added to commit (use "git add" and/or "git commit -a")
abel@VM mitfacts %
```

## MIT Fact Test Exercise: Testing (7/10)

```
(use "git restore <file>..." to discard changes in working directory)
modified:   mit.js
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
abel@VM mitfacts % git add .
```

```
abel@VM mitfacts % git status
```

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
modified:   mit.js
```

```
abel@VM mitfacts %
```

## MIT Fact Test Exercise: Testing (8/10)

```
abel@VM mitfacts % git commit -m 'fixed date'
[master b55a7ea] fixed date
 1 file changed, 1 insertion(+), 1 deletion(-)
abel@VM mitfacts % git push
Warning: Permanently added the RSA host key for IP address '140.82.112.3' to the list of known hosts.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 289 bytes | 289.00 KiB/s,
done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2
local objects.
To github.com:peterparkerdata/mitfacts.git
 0907d03..b55a7ea master -> master
abel@VM mitfacts %
```

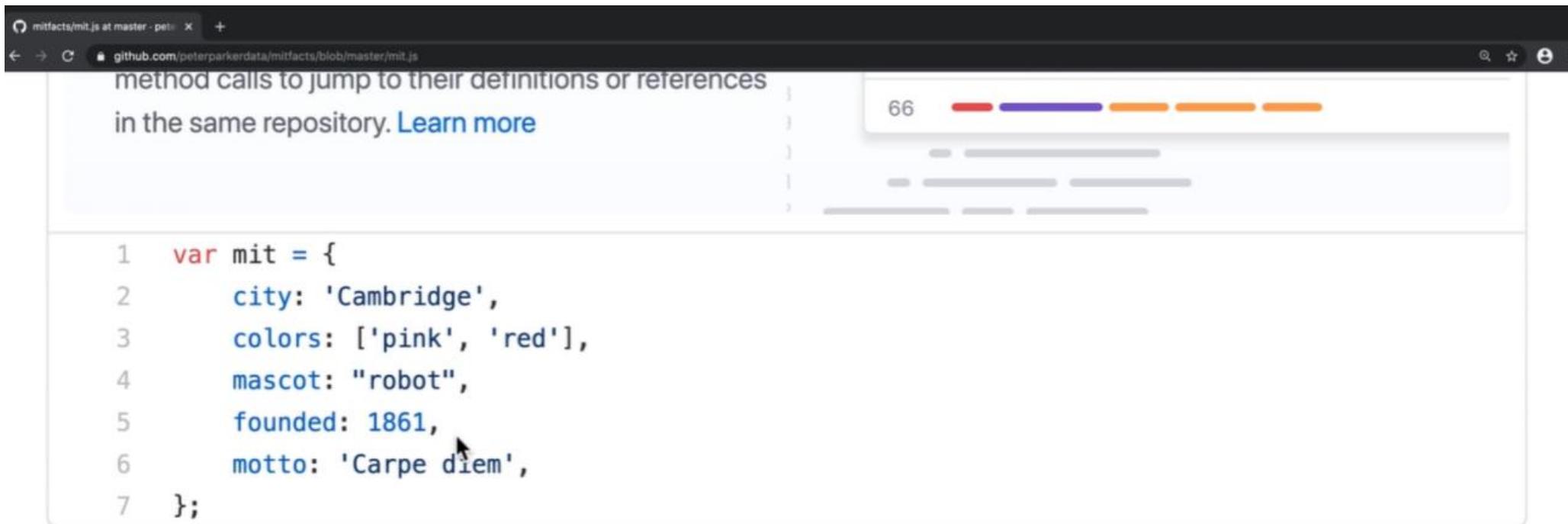
# MIT Fact Test Exercise: Testing (9/10)

The screenshot shows a GitHub repository page for 'peterparkerdata/mitfacts'. The repository has 4 commits, 1 branch, and 0 tags. The commits are:

File	Message	Time Ago
.gitignore	Initial commit	14 hours ago
LICENSE	Initial commit	14 hours ago
README.md	Initial commit	14 hours ago
mit.html	mit facts and test	14 hours ago
mit.js	<a href="#">fixed date</a>	1 minute ago
mit.test.js	mit facts and test	14 hours ago
package-lock.json	mit facts and test	14 hours ago
package.json	mit facts and test	14 hours ago

On the right side, there are sections for 'Releases' (No releases published, Create a new release), 'Packages' (No packages published, Publish your first package), and 'Languages' (A progress bar showing approximately 90% completion).

# MIT Fact Test Exercise: Testing (10/10)



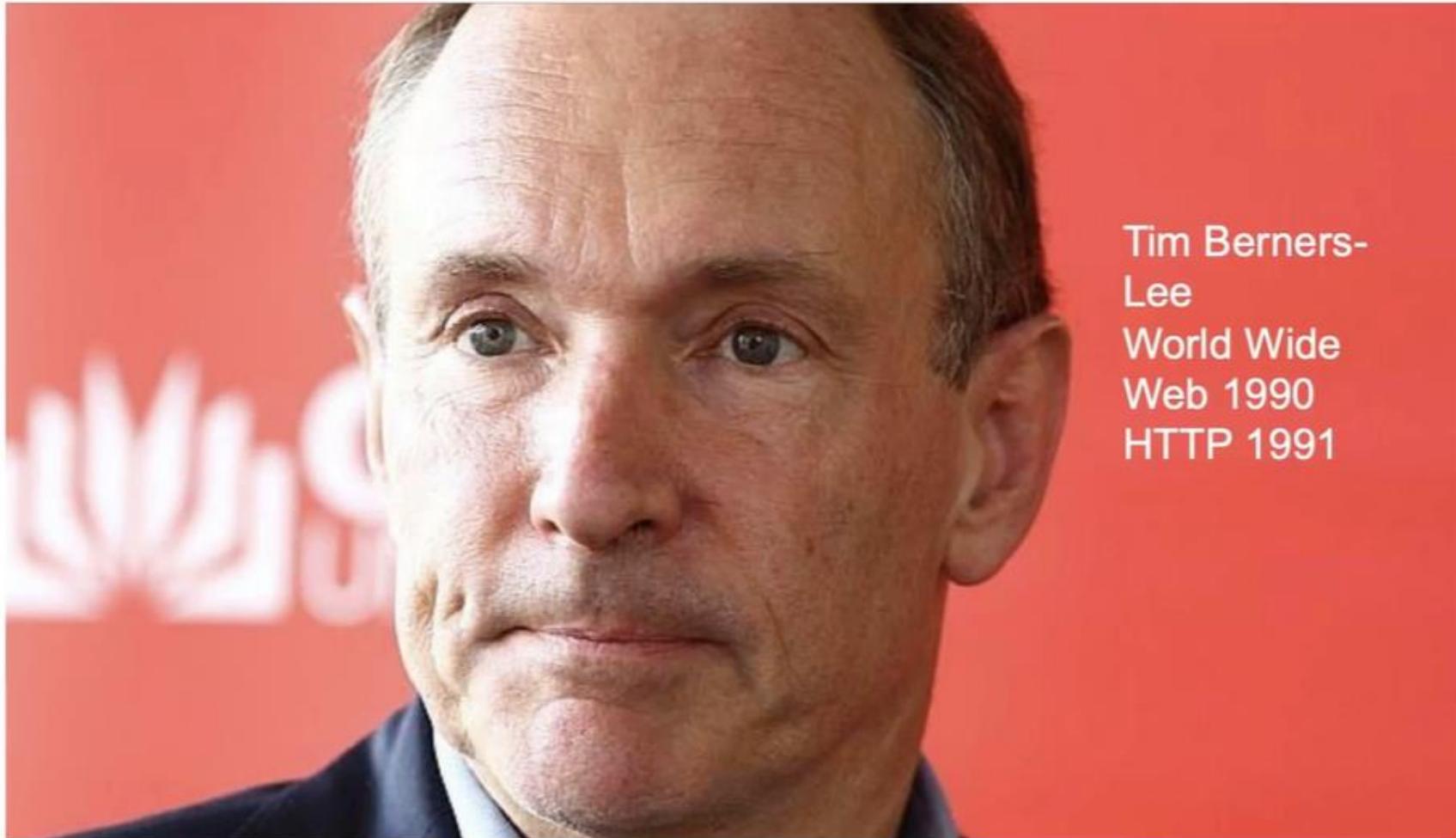
A screenshot of a GitHub code editor displaying a file named `mit.js`. The code defines an object `mit` with properties: `city`, `colors`, `mascot`, `founded`, and `motto`. The `motto` value is `'Carpe diem'`. The code is annotated with a callout box pointing to the `motto` line, containing the text: "method calls to jump to their definitions or references in the same repository. [Learn more](#)". A progress bar at the top right shows completion at 66%.

```
1 var mit = {  
2   city: 'Cambridge',  
3   colors: ['pink', 'red'],  
4   mascot: "robot",  
5   founded: 1861,  
6   motto: 'Carpe diem',  
7 };
```

© 2020 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

# Sir Tim Berners Lee Invents World Wide Web In 1990



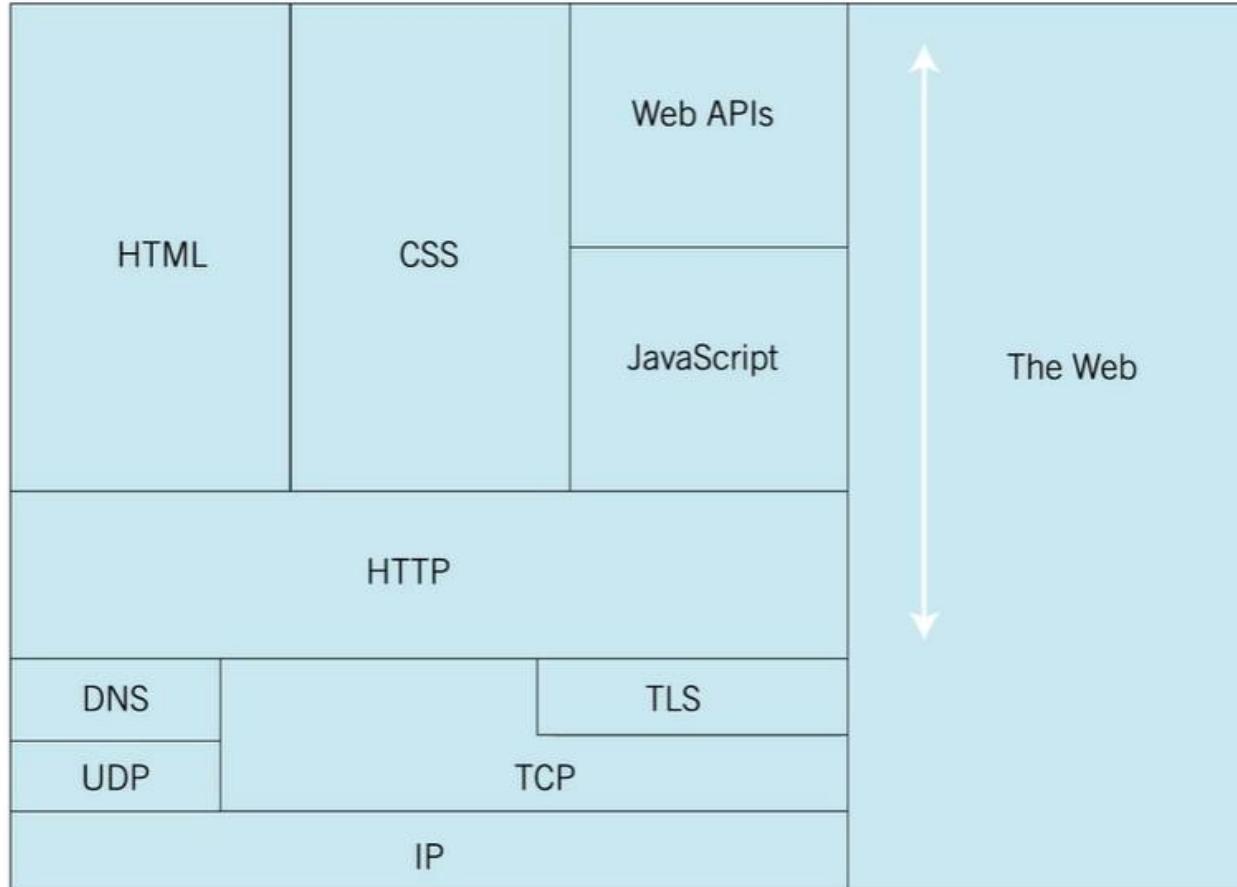
Tim Berners-  
Lee  
World Wide  
Web 1990  
HTTP 1991

# Digital Technologies (1/3)

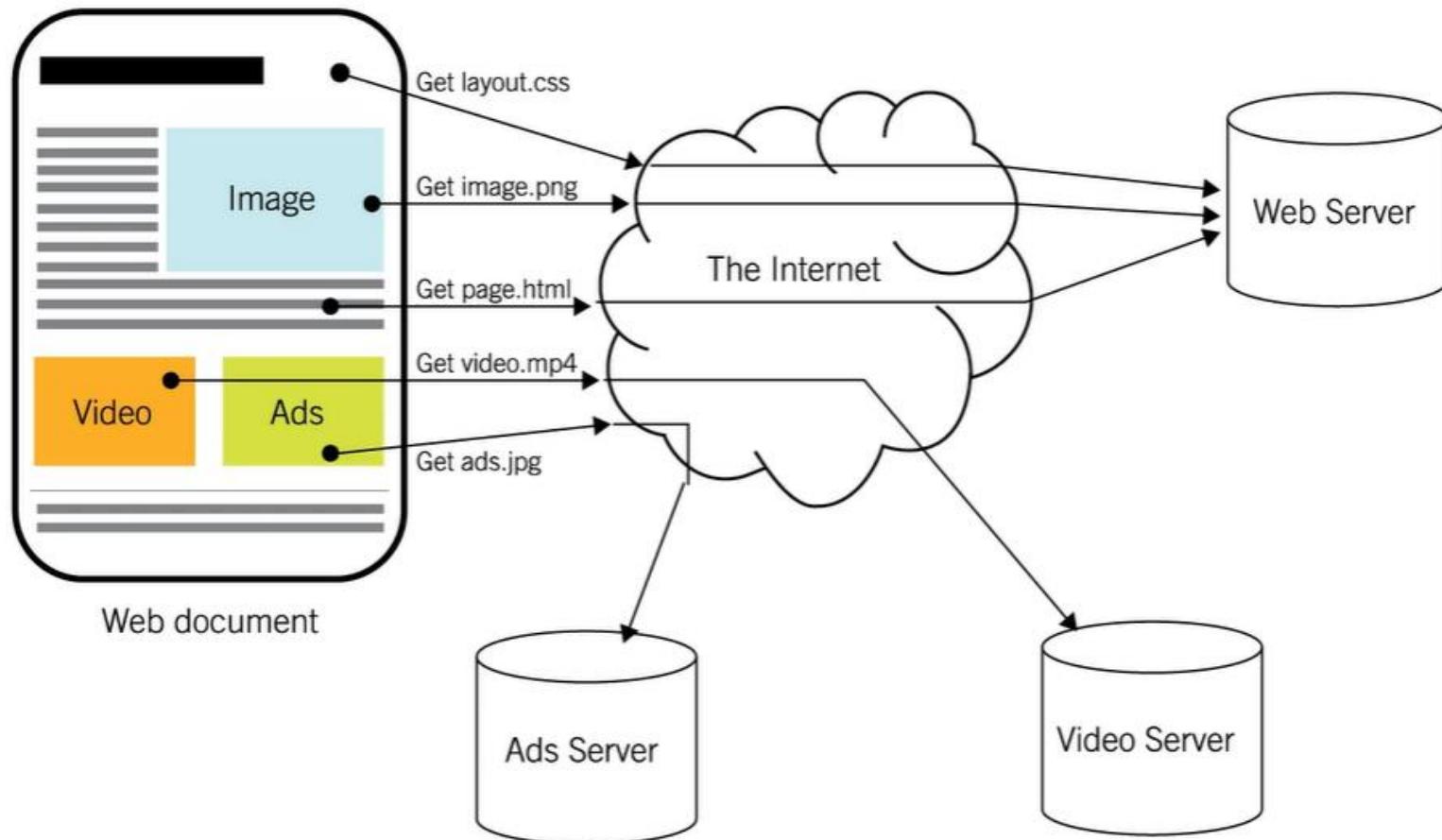
- Internet and Web 1970-2000
- Internet of Things 2001
- Smart Phone 2007
- Big Data ~2009
- Machine Learning ~2012
- Cloud Native / DevOps ~2010

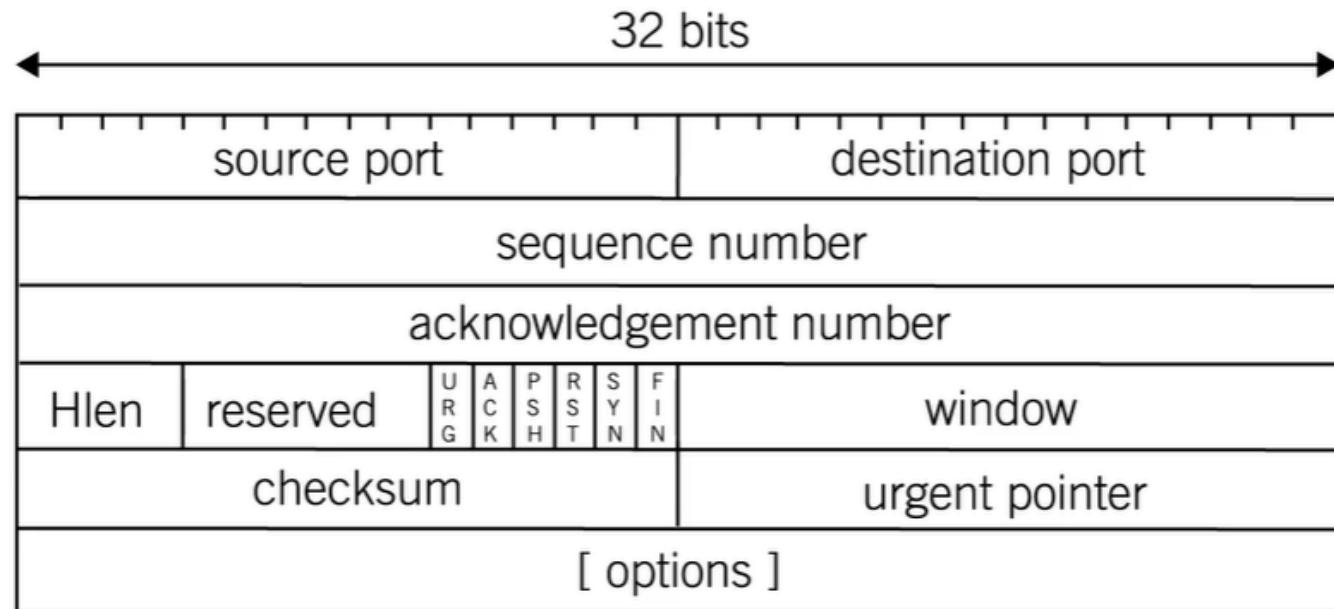
- 1983 DNS servers map URLs to IP  
<http://ww.mit.edu> ↗ 18.58.1.131
- Invented URIs (Universal Resource Identifiers)
- In 1991 HTTP 0.9 : GET, POST
- 1995 Browser - Netscape Browser

# Digital Technologies (3/3)



# Functioning Of The Browser

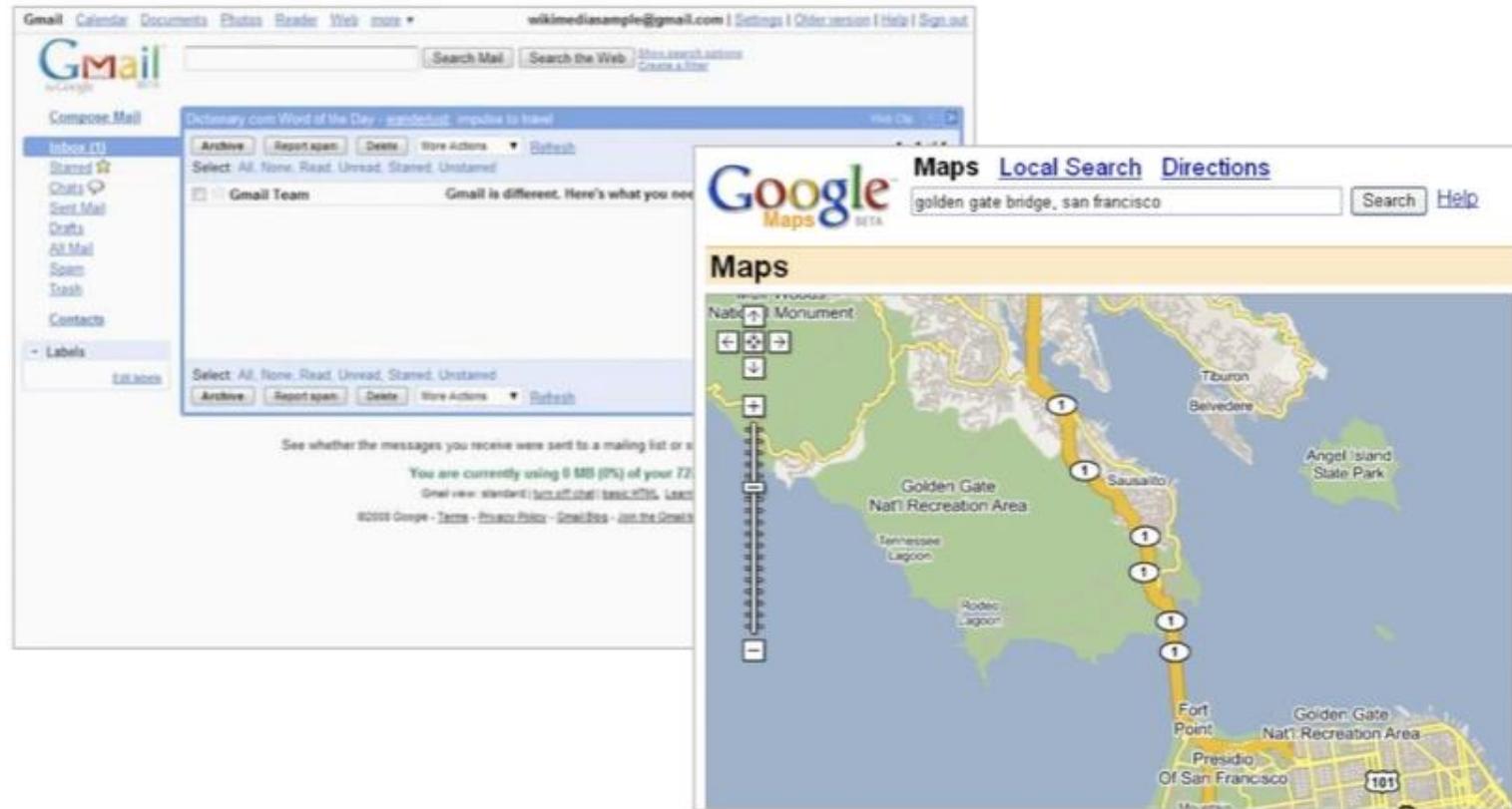




# 1995 Brenden Eich Invents JavaScript In 10 Days

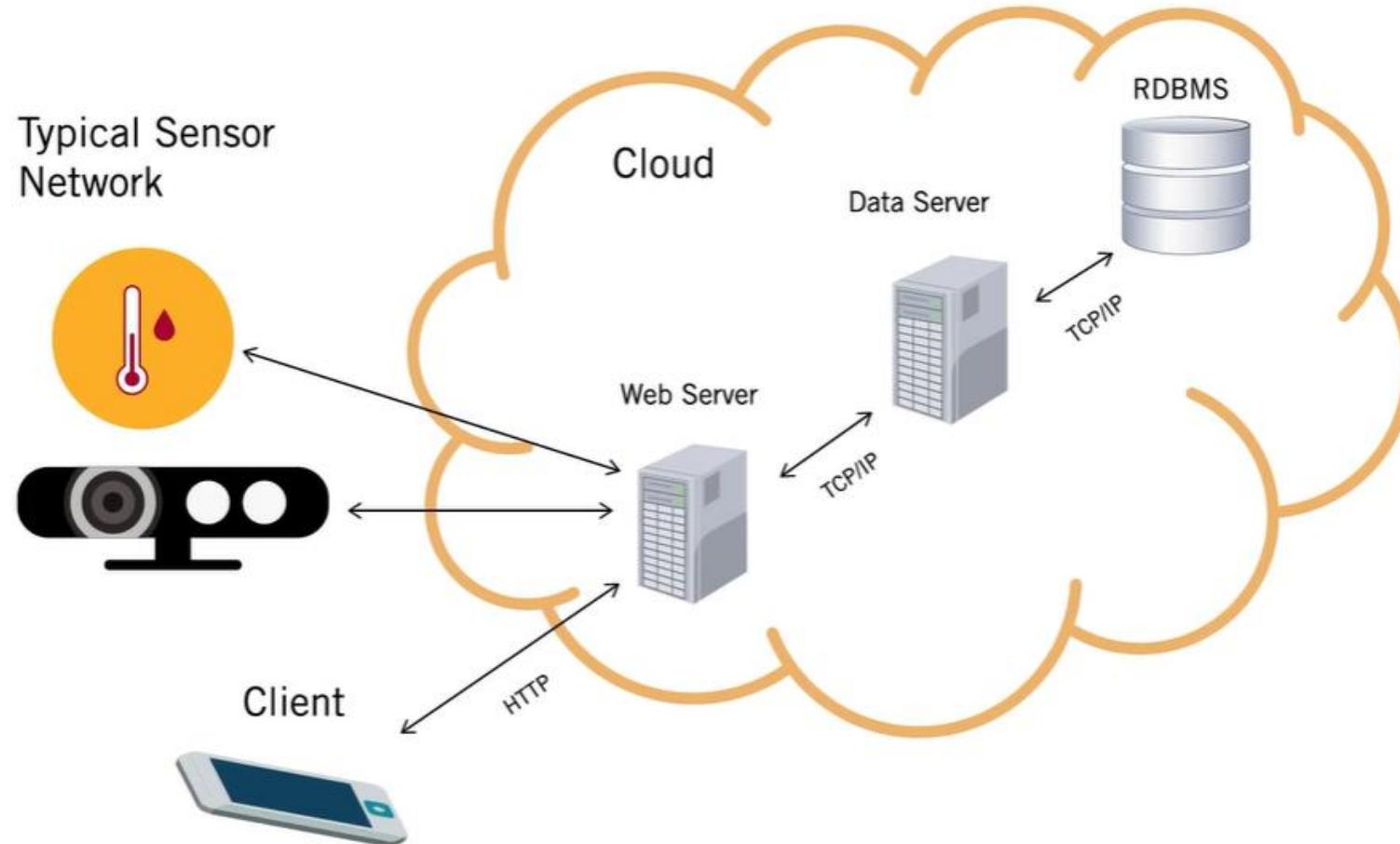


# 2004 First Web Apps – HTML 5

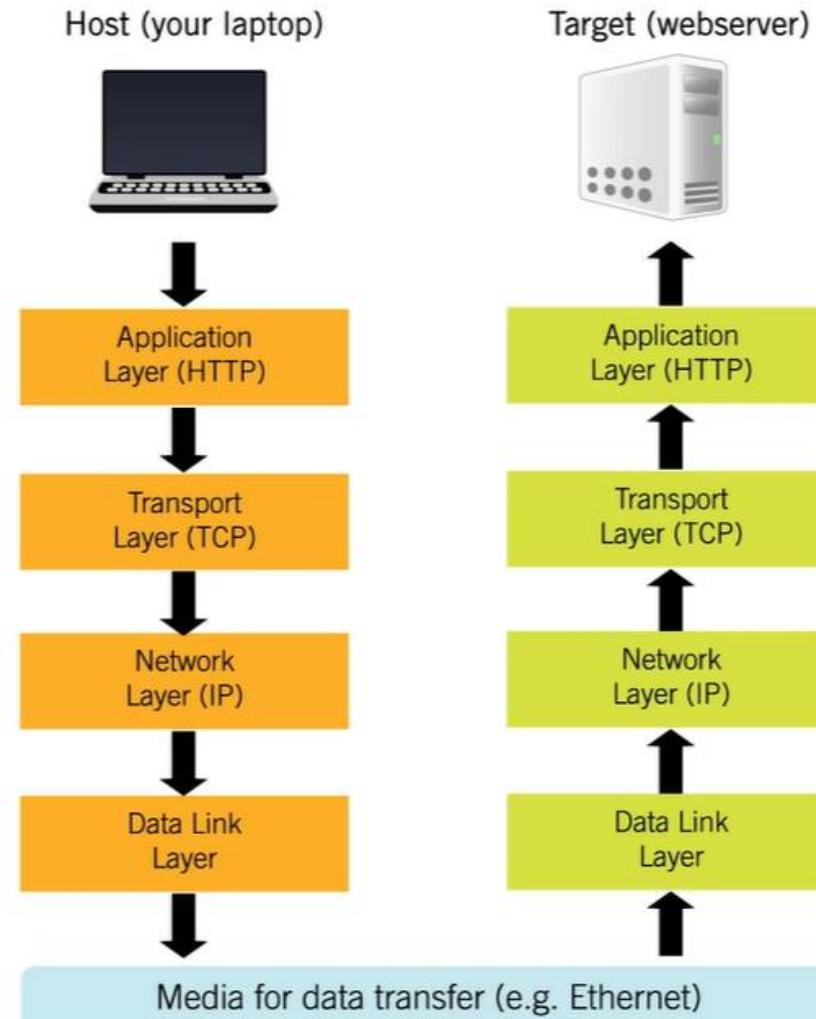




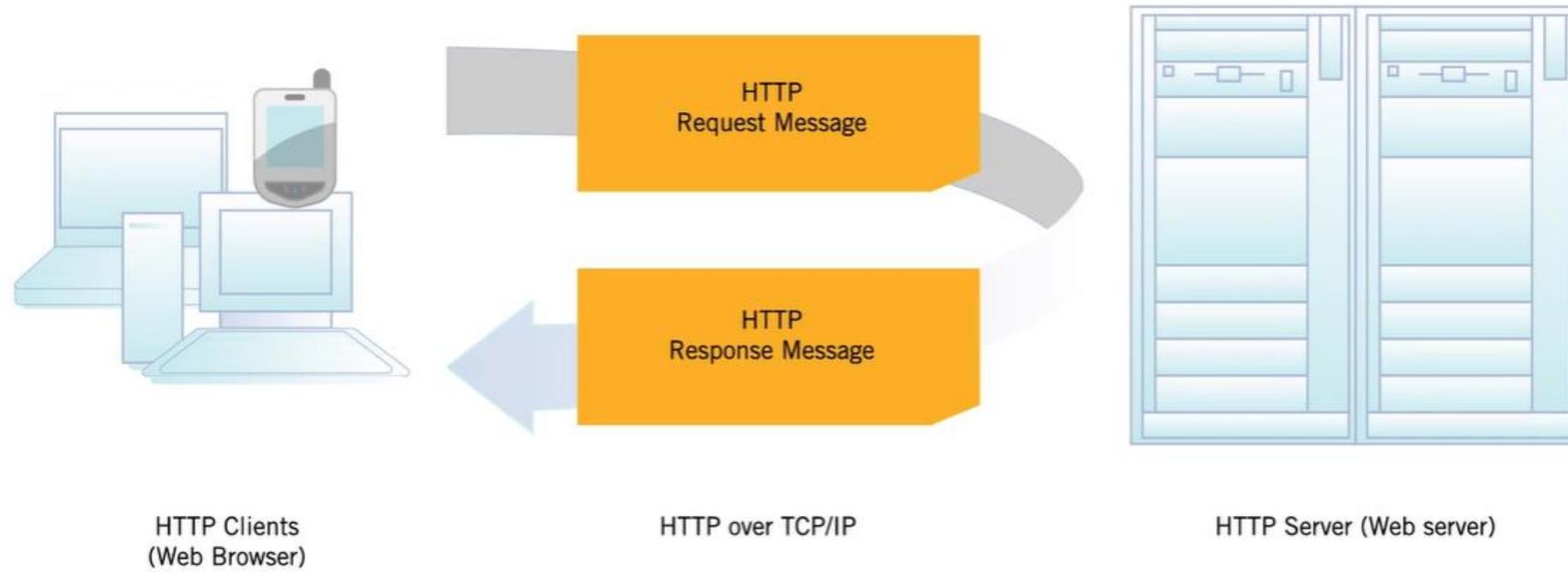
# Web Programming Environment



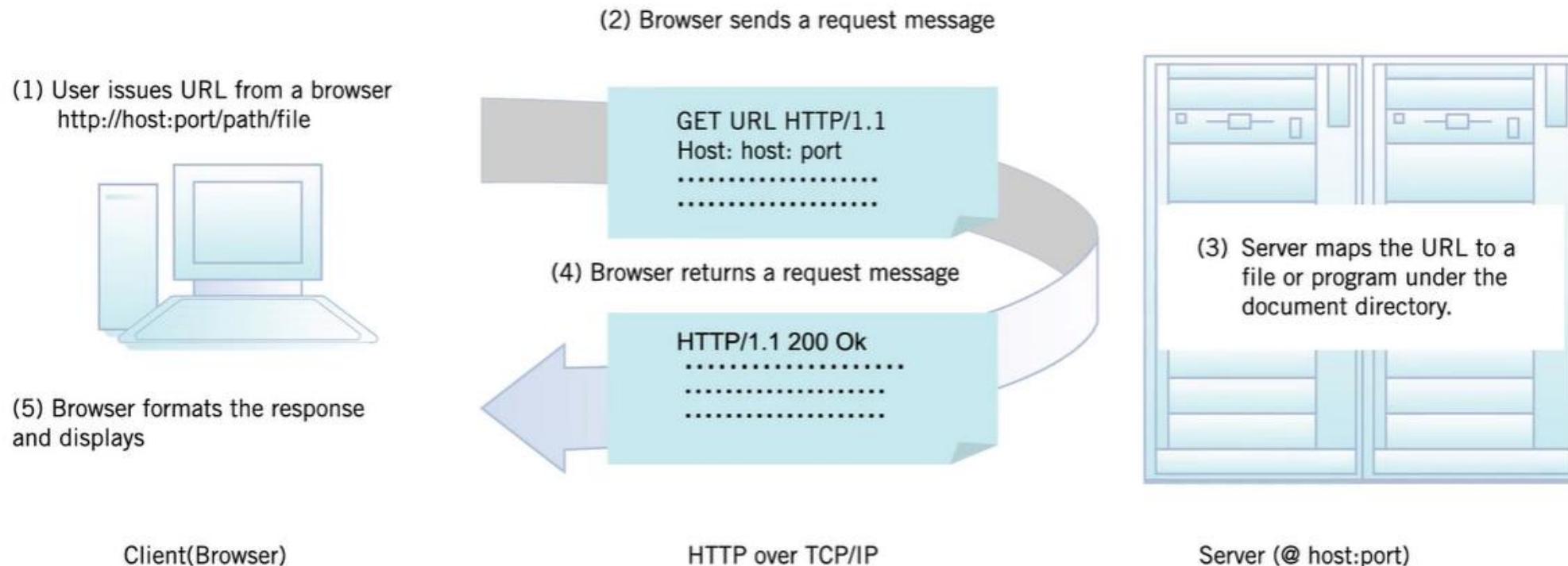
# Graphical View Of Web Programming



# HTTP Pull Mechanism (1/2)

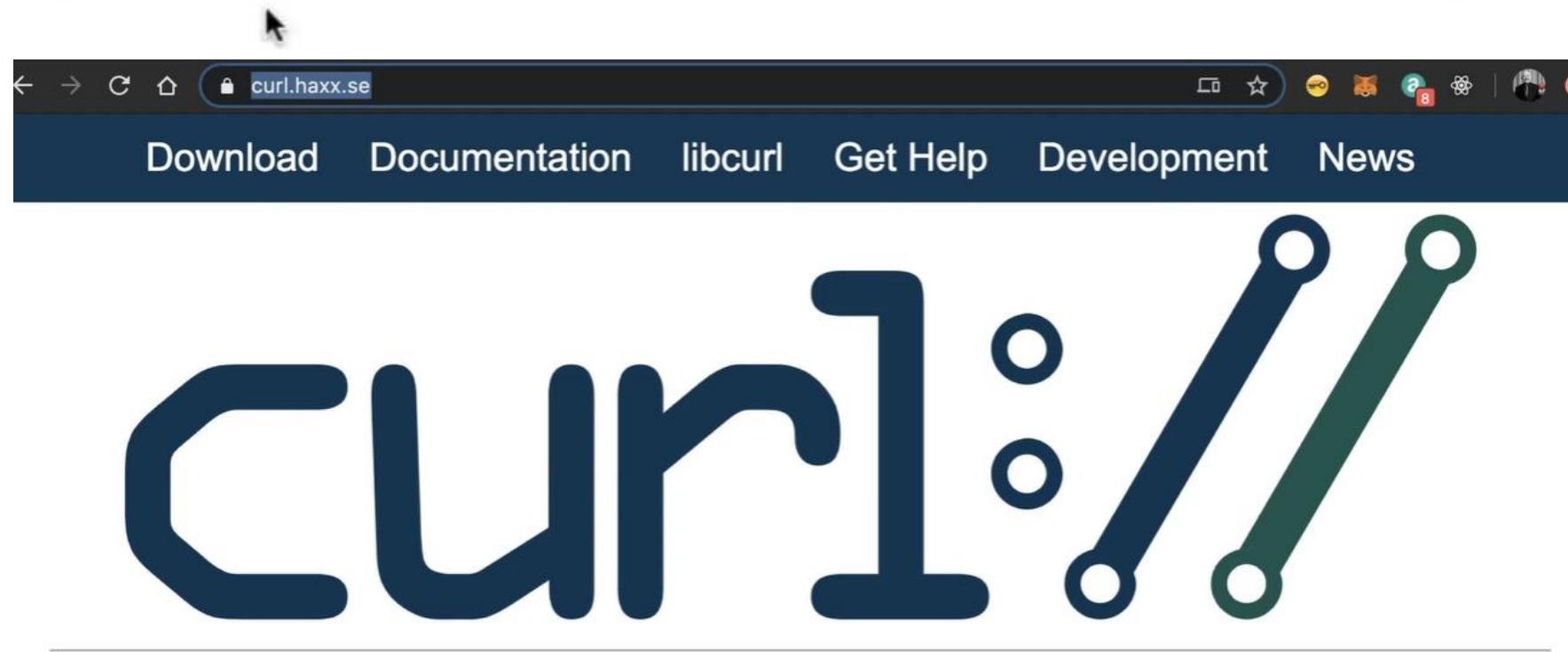


## HTTP Pull Mechanism (2/2)



## For Programming: Curl

- [curl https://pollysnips.s3.amazonaws.com/users.json](https://pollysnips.s3.amazonaws.com/users.json)



**command line tool and library**  
for transferring data with URLs

# Issuing Curl Comments (1/2)

```
johnwilliams — zsh — 80x24
?1 ~ % curl https://pollysnips.s3.amazonaws.com/contacts.json
[
  { "name": "adam", "email": "adam@mit.edu" },
  { "name": "anne", "email": "anne@mit.edu" },
  { "name": "fred", "email": "fred@mit.edu" }
]
✓ ~ % curl -o test.json https://pollysnips.s3.amazonaws.com/users.json
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent    Left  Speed
100    772  100    772      0       0  5761      0  --:--:--  --:--:--  --:--:--  5761
✓ ~ % cat test.json
```

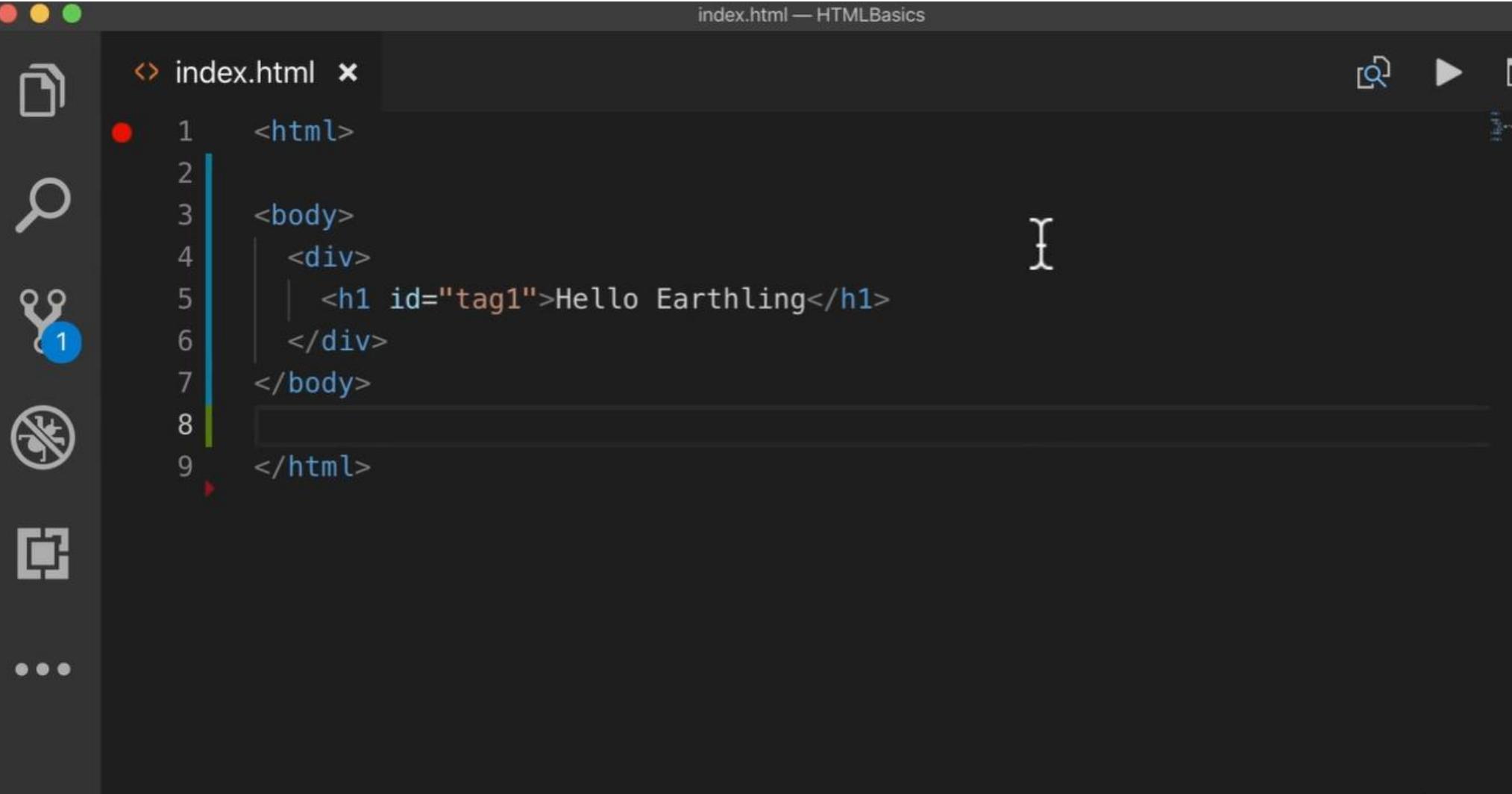
## Issuing Curl Comment (2/2)

```
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  772  100  772     0      0  5761      0 --:--:-- --:--:-- --:--:--  5761
[~ % cat test.json
[
  {
    "name"      : "peter parker",
    "age"       : 21,
    "email"     : "peter@mit.edu",
    "courses"   :
      [
        {"number" : "1.00", "name" : "engr comp"},  

        {"number" : "7.00", "name" : "intro bio"}
      ]
  },
  {
    "name"      : "bruce wayne",
    "age"       : 32,
    "email"     : "bruce@mit.edu",
    "courses"   :
      [
        {"number" : "2.00", "name" : "intro ME"},  

        {"number" : "3.00", "name" : "intro MS"}
      ]
}
```

# Injecting JavaScript Into HTML Web Pages (1/11)

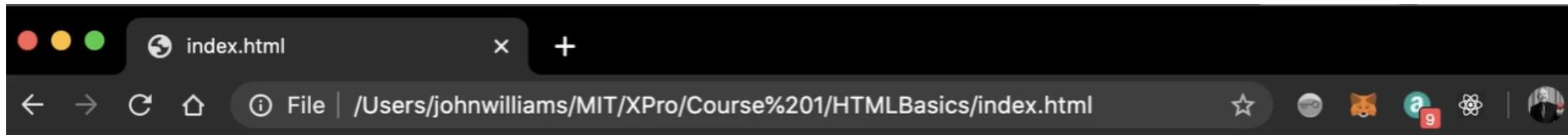


A screenshot of a code editor window titled "index.html — HTMLBasics". The file contains the following HTML code:

```
<html>
  <body>
    <div>
      <h1 id="tag1">Hello Earthling</h1>
    </div>
  </body>
</html>
```

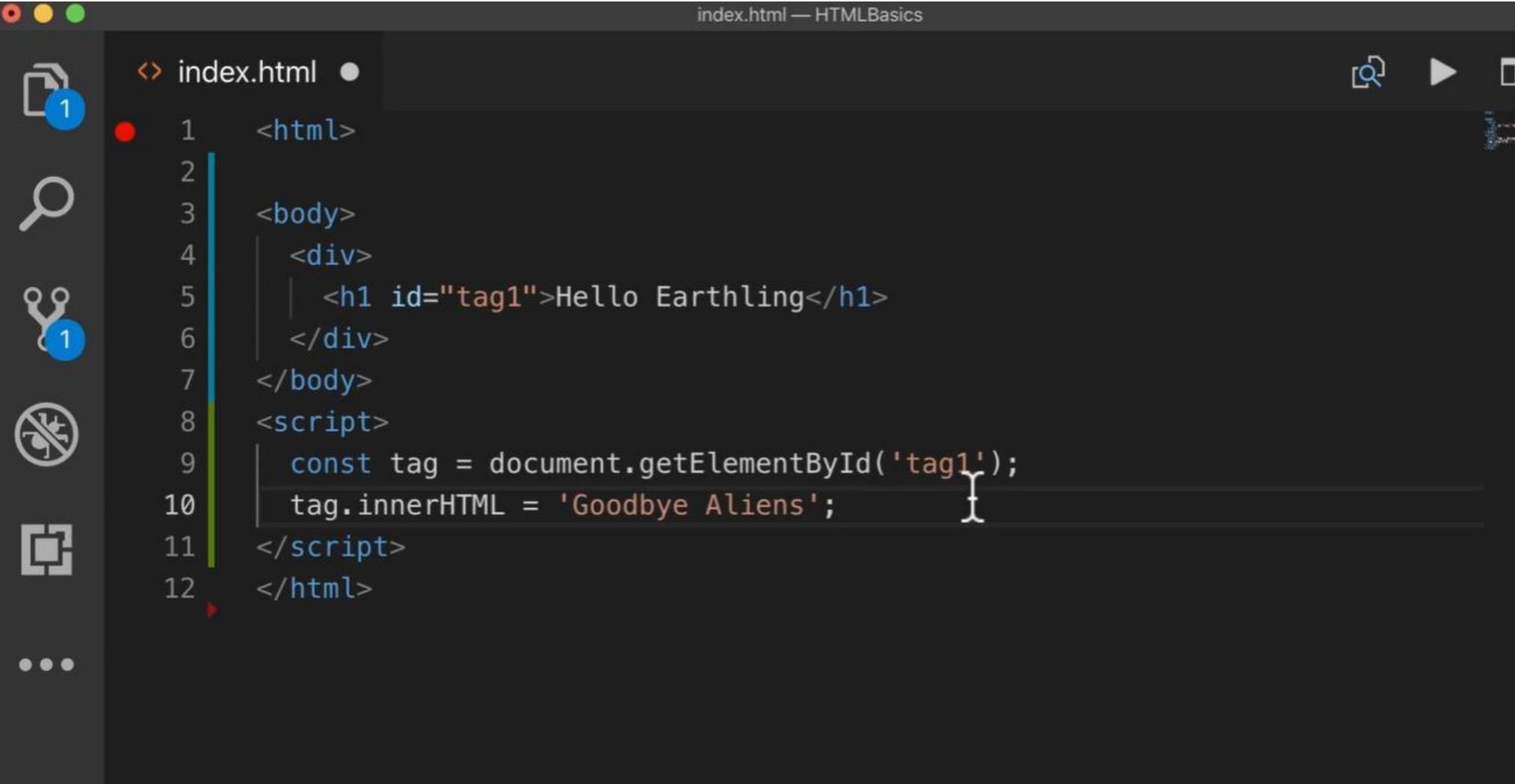
The code editor has a dark theme with light-colored syntax highlighting. A vertical teal selection bar highlights the entire body and div sections. A red dot is placed on the first line of the body section. A blue circle with the number "1" is positioned next to the magnifying glass icon in the sidebar, indicating a potential injection point or error.

# Injecting JavaScript Into HTML Web Pages (2/11)



Hello Earthling

# Injecting JavaScript Into HTML Web Pages (3/11)

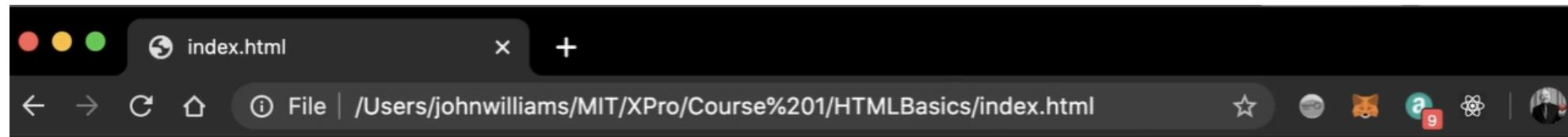


The screenshot shows a code editor window titled "index.html — HTMLBasics". The sidebar on the left has icons for file operations, search, and other tools, with a blue circle containing the number "1" indicating a pending change. The main pane displays the following HTML code:

```
<html>
  <body>
    <div>
      <h1 id="tag1">Hello Earthling</h1>
    </div>
  </body>
<script>
  const tag = document.getElementById('tag1');
  tag.innerHTML = 'Goodbye Aliens';
</script>
</html>
```

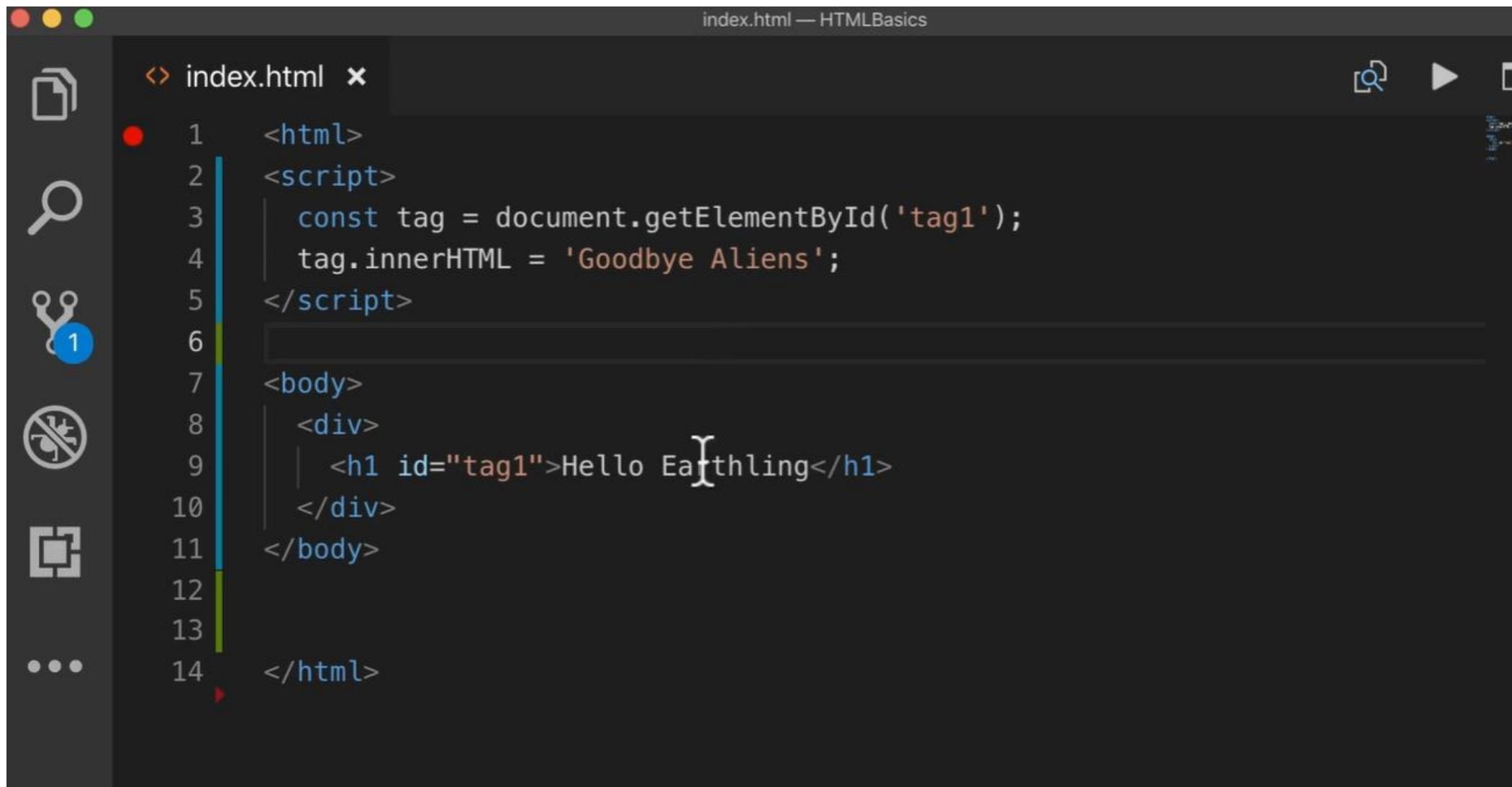
The code consists of 12 numbered lines. Lines 1 through 7 represent the static HTML structure. Lines 8 through 11 are part of a script block that injects dynamic JavaScript into the page. Line 8 defines a variable "tag" pointing to the element with id "tag1". Line 9 changes its innerHTML from "Hello Earthling" to "Goodbye Aliens". Line 12 closes the script tag.

# Injecting JavaScript Into HTML Web Pages (4/11)



**Goodbye Aliens**

# Injecting JavaScript Into HTML Web Pages (5/11)



The screenshot shows a code editor window titled "index.html — HTMLBasics". The file contains the following HTML and JavaScript code:

```
index.html
<html>
<script>
  const tag = document.getElementById('tag1');
  tag.innerHTML = 'Goodbye Aliens';
</script>
<body>
<div>
  <h1 id="tag1">Hello Earthling</h1>
</div>
</body>
</html>
```

The code editor interface includes a sidebar with icons for file operations, search, and other tools. A status bar at the bottom indicates "1 file" and "1 changes".

# Injecting JavaScript Into HTML Web Pages (6/11)

The screenshot shows a web browser window with the title bar "index.html". The address bar displays the file path: "/Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html". The main content area contains the text "Hello Earth". A developer tools sidebar is open, with the "Console" tab selected. The console output shows an error message: "Uncaught TypeError: Cannot set property 'innerHTML' of null at index.html:4". Below the error, there is a link to "file:///Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html:4". The developer tools interface includes tabs for Elements, Sources, Network, Performance, Memory, Application, and Security. At the bottom of the sidebar, there are links for "Console" and "What's New". A callout box highlights the "Emulate vision deficiencies from the Rendering tab" feature, which allows users to get a visual approximation of how people with vision deficiencies might experience their site.

DevTools - file:///Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html

Elements Console Sources Network Performance Memory Application Security

Default levels ▾

✖ Uncaught TypeError: Cannot set property 'innerHTML' of null  
at index.html:4

> | file:///Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html:4

⋮ Console What's New ×

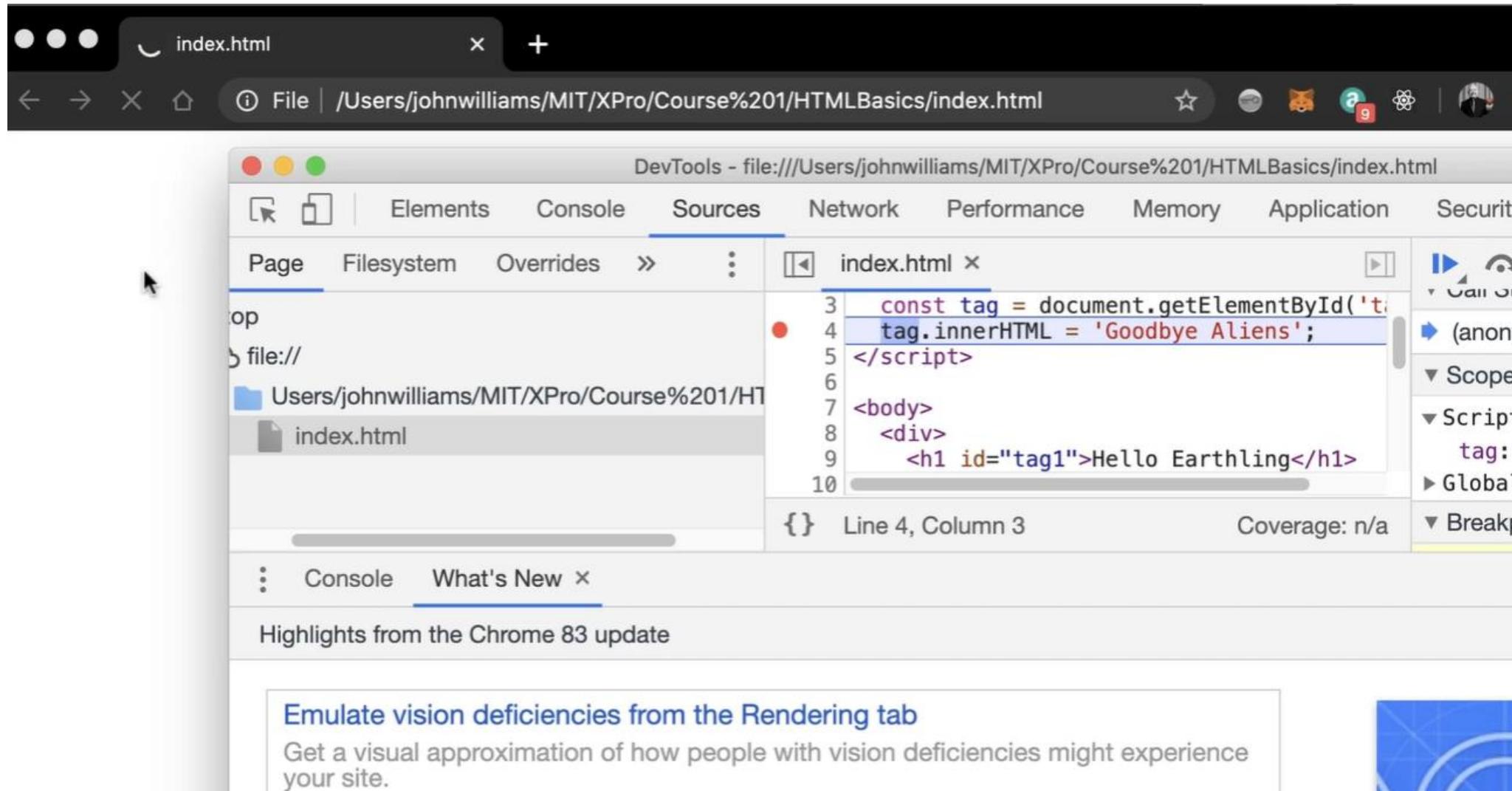
Highlights from the Chrome 83 update

Emulate vision deficiencies from the Rendering tab

Get a visual approximation of how people with vision deficiencies might experience your site.



# Injecting JavaScript Into HTML Web Pages (7/11)



The screenshot shows the Chrome DevTools interface with the "Sources" tab selected. The left sidebar lists "Page", "Filesystem", and "Overrides". The main pane displays the file "index.html" with the following code:

```
const tag = document.getElementById('tag1');
tag.innerHTML = 'Goodbye Aliens';
</script>
<body>
  <div>
    <h1 id="tag1">Hello Earthling</h1>
```

The line "tag.innerHTML = 'Goodbye Aliens';" is highlighted in red, indicating it was injected. The DevTools status bar at the bottom shows "Line 4, Column 3".

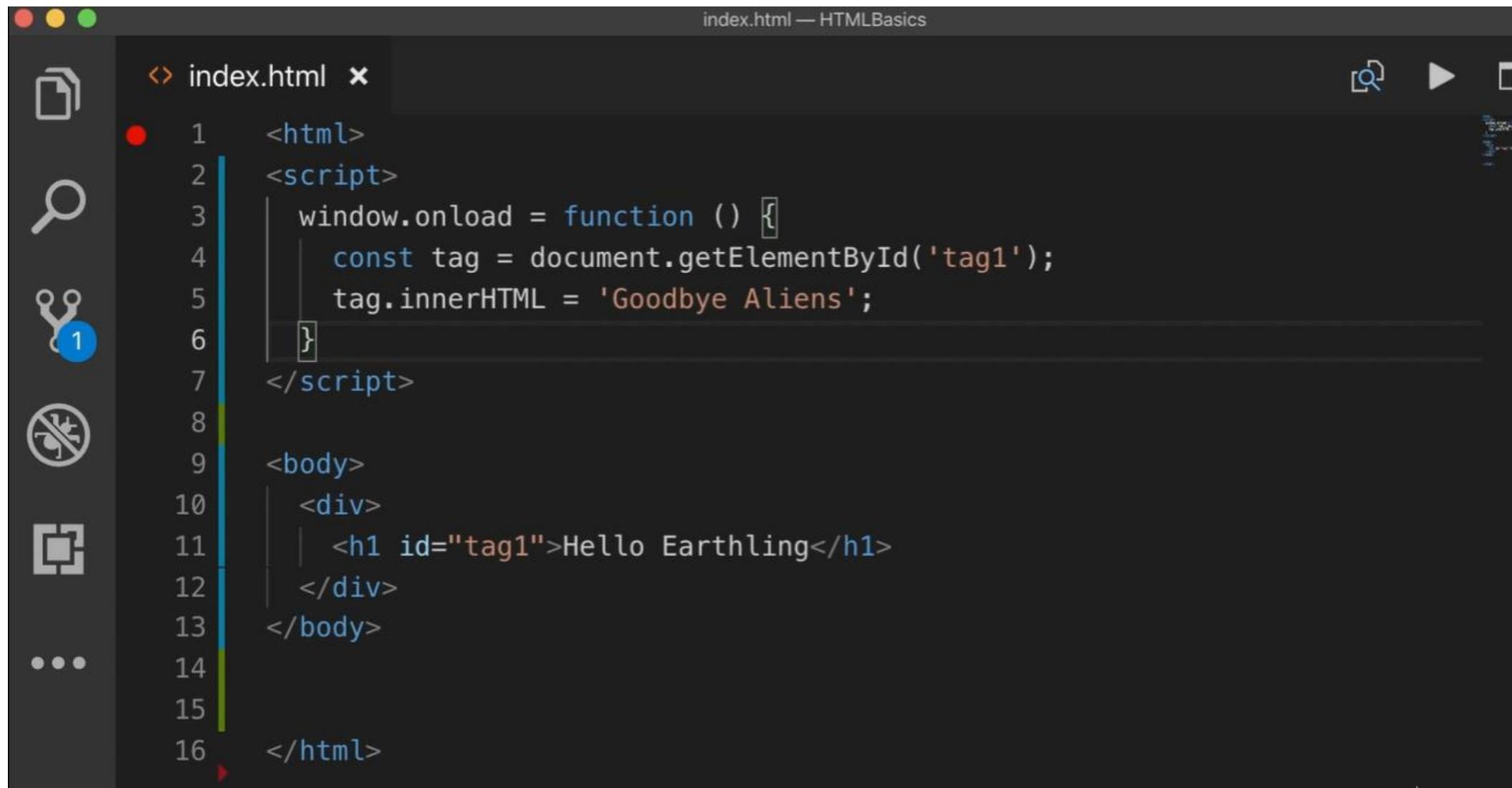
Below the DevTools, a callout box highlights the "Emulate vision deficiencies from the Rendering tab" feature.

Callout Box Content:

- Emulate vision deficiencies from the Rendering tab
- Get a visual approximation of how people with vision deficiencies might experience your site.



# Injecting JavaScript Into HTML Web Pages (8/11)



The screenshot shows a code editor window titled "index.html — HTMLBasics". The file contains the following HTML and JavaScript code:

```
index.html
<html>
<script>
  window.onload = function () {
    const tag = document.getElementById('tag1');
    tag.innerHTML = 'Goodbye Aliens';
  }
</script>
<body>
<div>
  <h1 id="tag1">Hello Earthling</h1>
</div>
</body>
</html>
```

The code is numbered from 1 to 16. A red dot icon is visible in the top-left corner of the editor area, indicating a problem or warning.

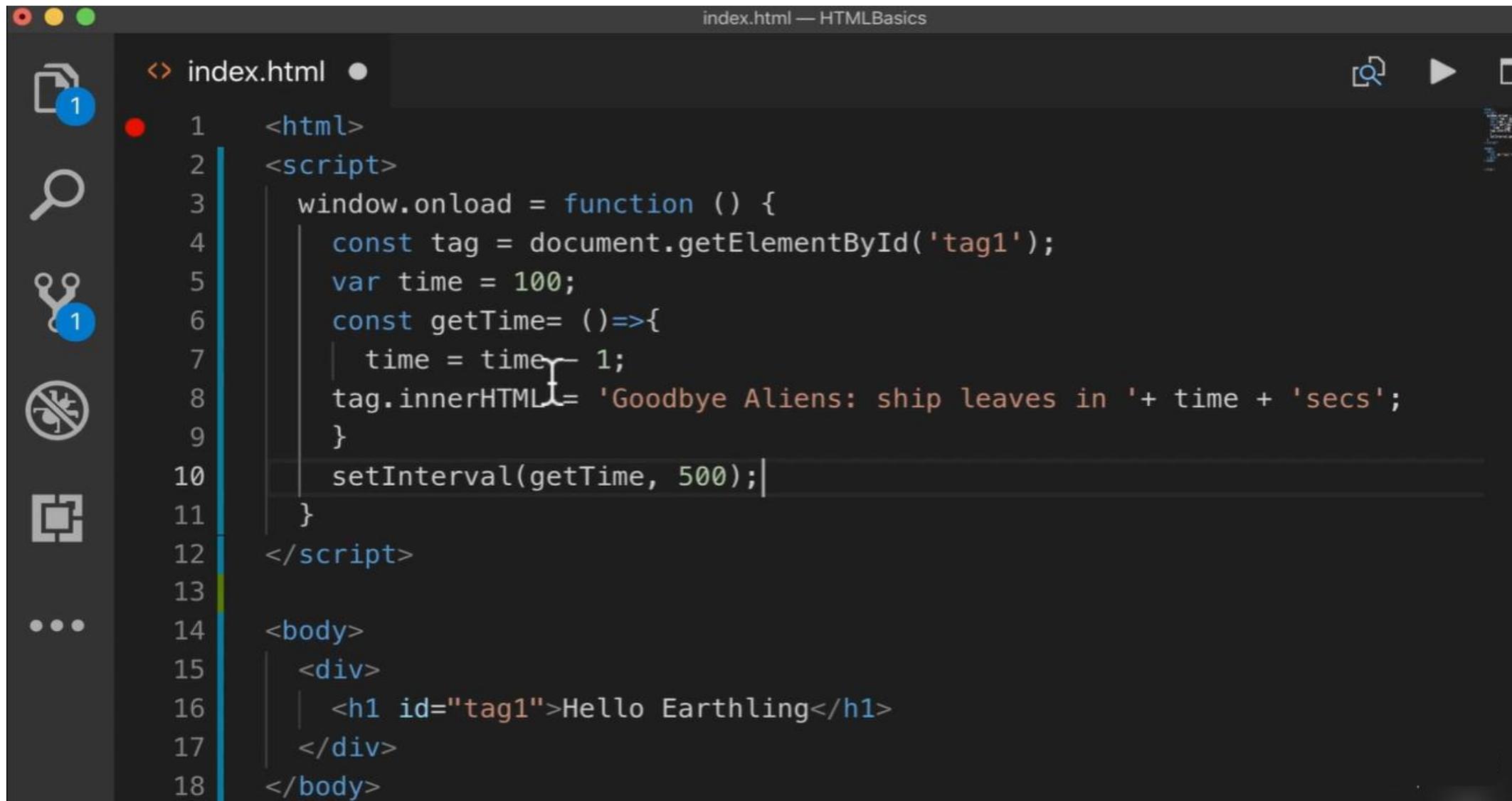
# Injecting JavaScript Into HTML Web Pages (9/11)

The screenshot shows a web browser window with the title "index.html" and the URL "/Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html". The main content area displays the text "Goodbye Aliens". Below the browser is the Chrome DevTools interface, specifically the "Sources" tab. The left sidebar shows the file structure: "file:///Users/johnwilliams/MIT/XPro/Course%201/HTMLBasics/index.html". The right pane shows the source code of the file:

```
1 <html>
2 <script>
3   window.onload = function () {
4     const tag = document.getElementById('alien');
5     tag.innerHTML = 'Goodbye Aliens';
6   }
7 </script>
```

The code is annotated with line numbers (1-7) and column numbers (23). The status bar at the bottom of the DevTools indicates "Coverage: n/a".

# Injecting JavaScript Into HTML Web Pages (10/11)

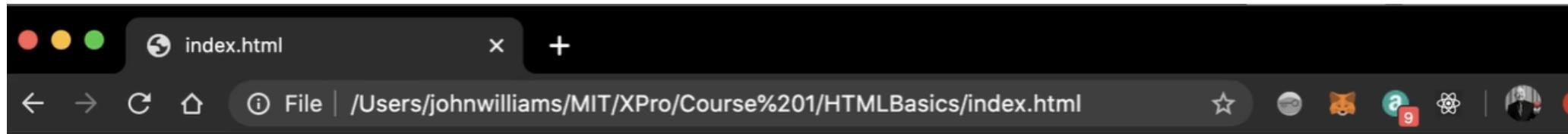


The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons: a file icon with a blue circle containing the number '1', a magnifying glass, a wrench, a trash can, a square icon, and three dots. The main area is titled 'index.html — HTMLBasics'. The code itself is as follows:

```
1 <html>
2 <script>
3   window.onload = function () {
4     const tag = document.getElementById('tag1');
5     var time = 100;
6     const getTime= ()=>{
7       time = time- 1;
8       tag.innerHTML= 'Goodbye Aliens: ship leaves in '+ time + 'secs';
9     }
10    setInterval(getTime, 500);
11  }
12 </script>
13
14 <body>
15   <div>
16     <h1 id="tag1">Hello Earthling</h1>
17   </div>
18 </body>
```

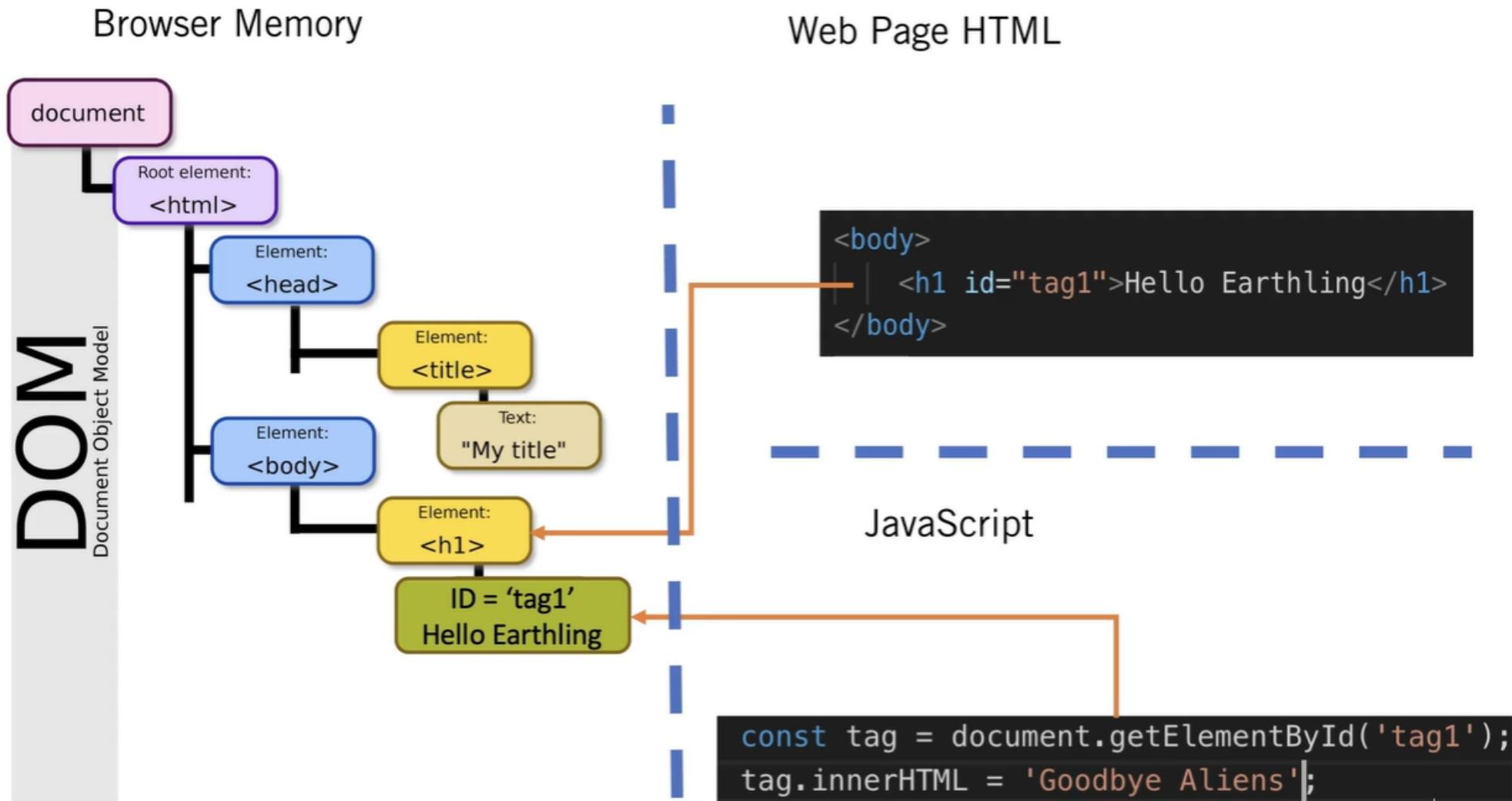
The code includes a script block that runs on page load. It finds an element with id 'tag1', sets its innerHTML to 'Goodbye Aliens: ship leaves in ' + time + 'secs', and then uses setInterval to update this value every 500ms. The body contains a single div with an h1 element that displays 'Hello Earthling'.

# Injecting JavaScript Into HTML Web Pages (11/11)

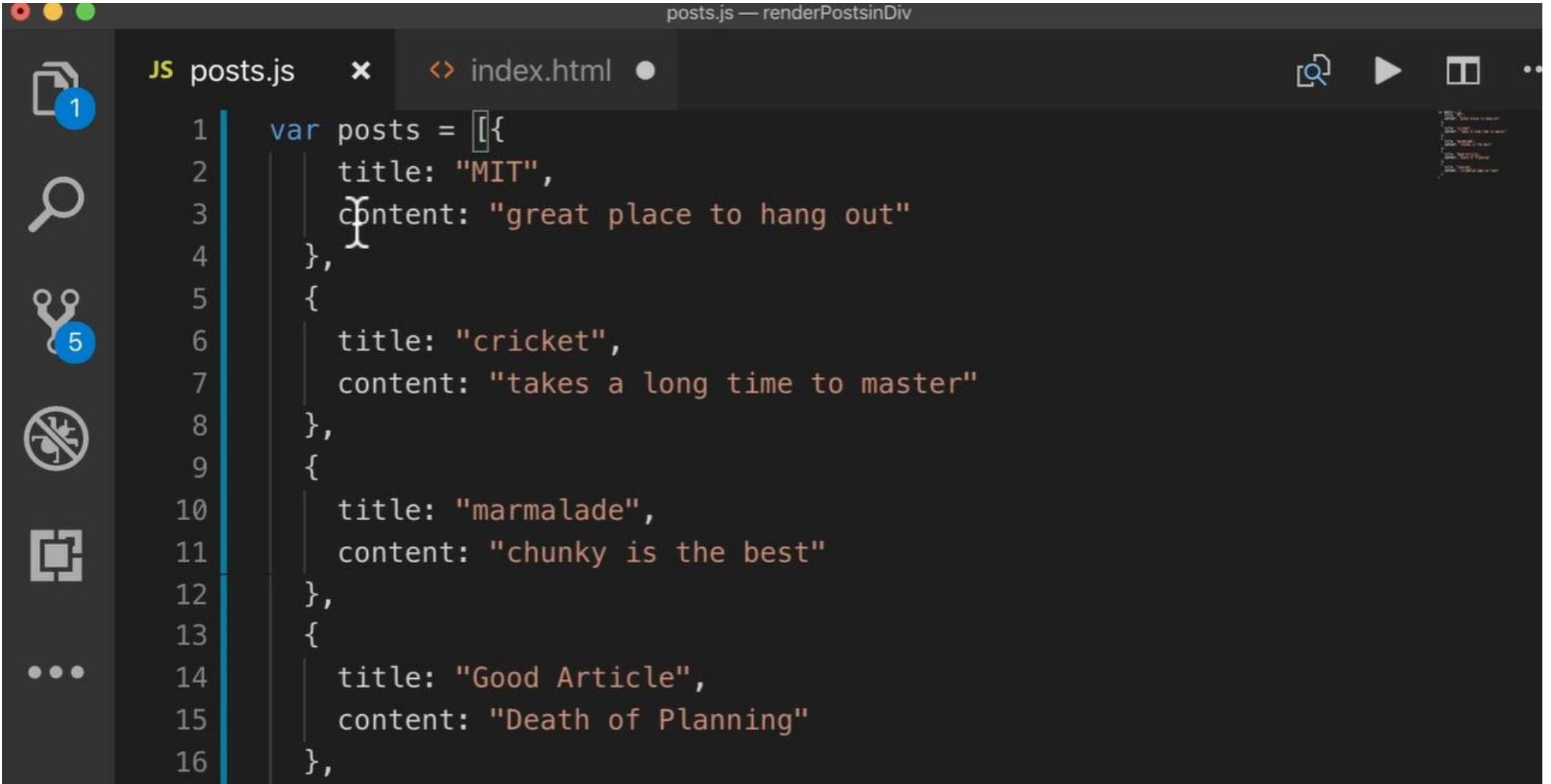


**Goodbye Aliens: ship leaves in 97secs**

# Document Object Model



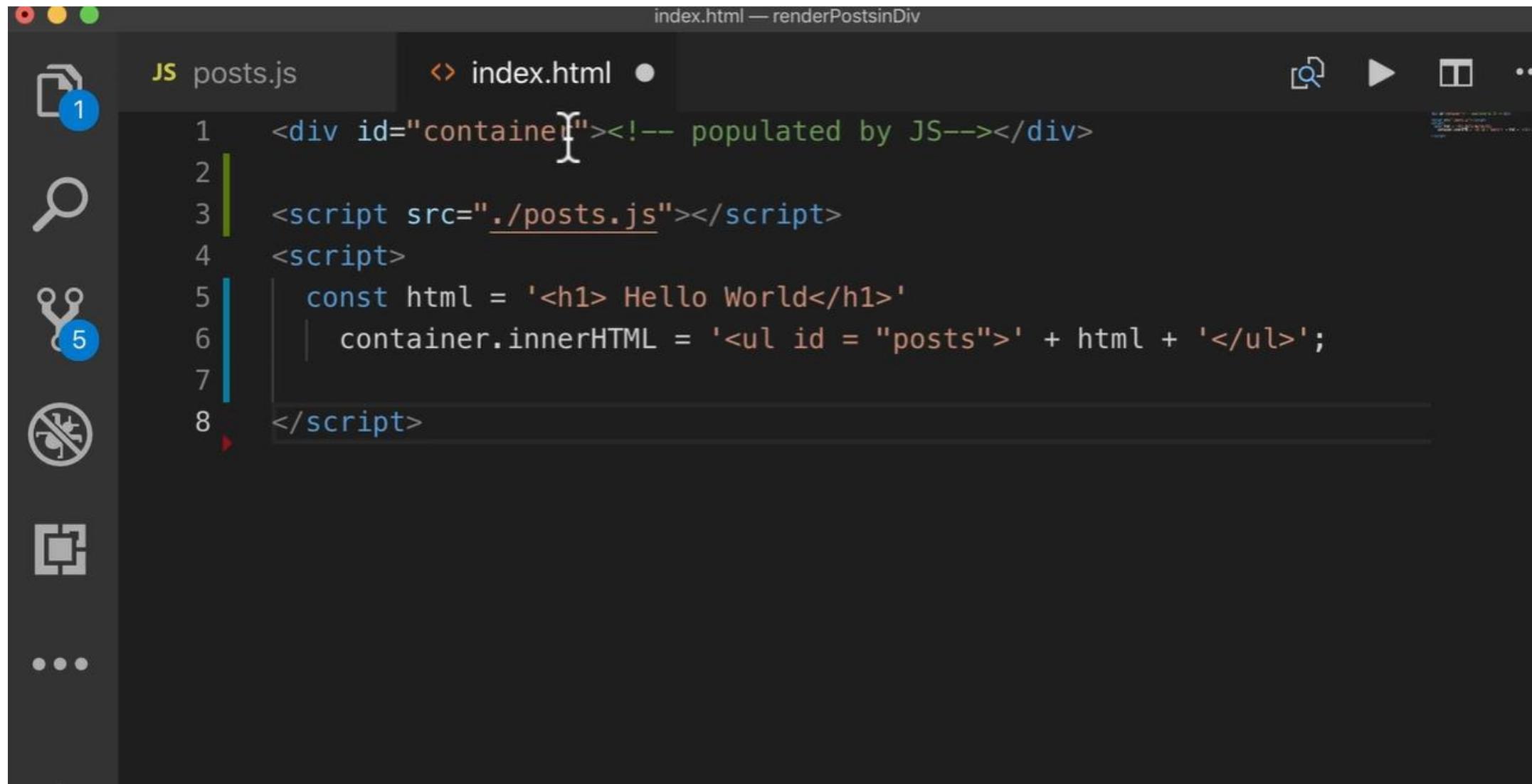
# Dynamically Inject Posts Into Div (1/8)



A screenshot of a code editor window titled "posts.js — renderPostsinDiv". The left sidebar shows icons for file operations (1 file open), search (1 search result), and other tools (5 notifications). The main editor area displays the following JavaScript code:

```
1 var posts = [{
2   title: "MIT",
3   content: "great place to hang out"
4 },
5 {
6   title: "cricket",
7   content: "takes a long time to master"
8 },
9 {
10  title: "marmalade",
11  content: "chunky is the best"
12 },
13 {
14  title: "Good Article",
15  content: "Death of Planning"
16 }]
```

## Dynamically Inject Posts Into Div (2/8)

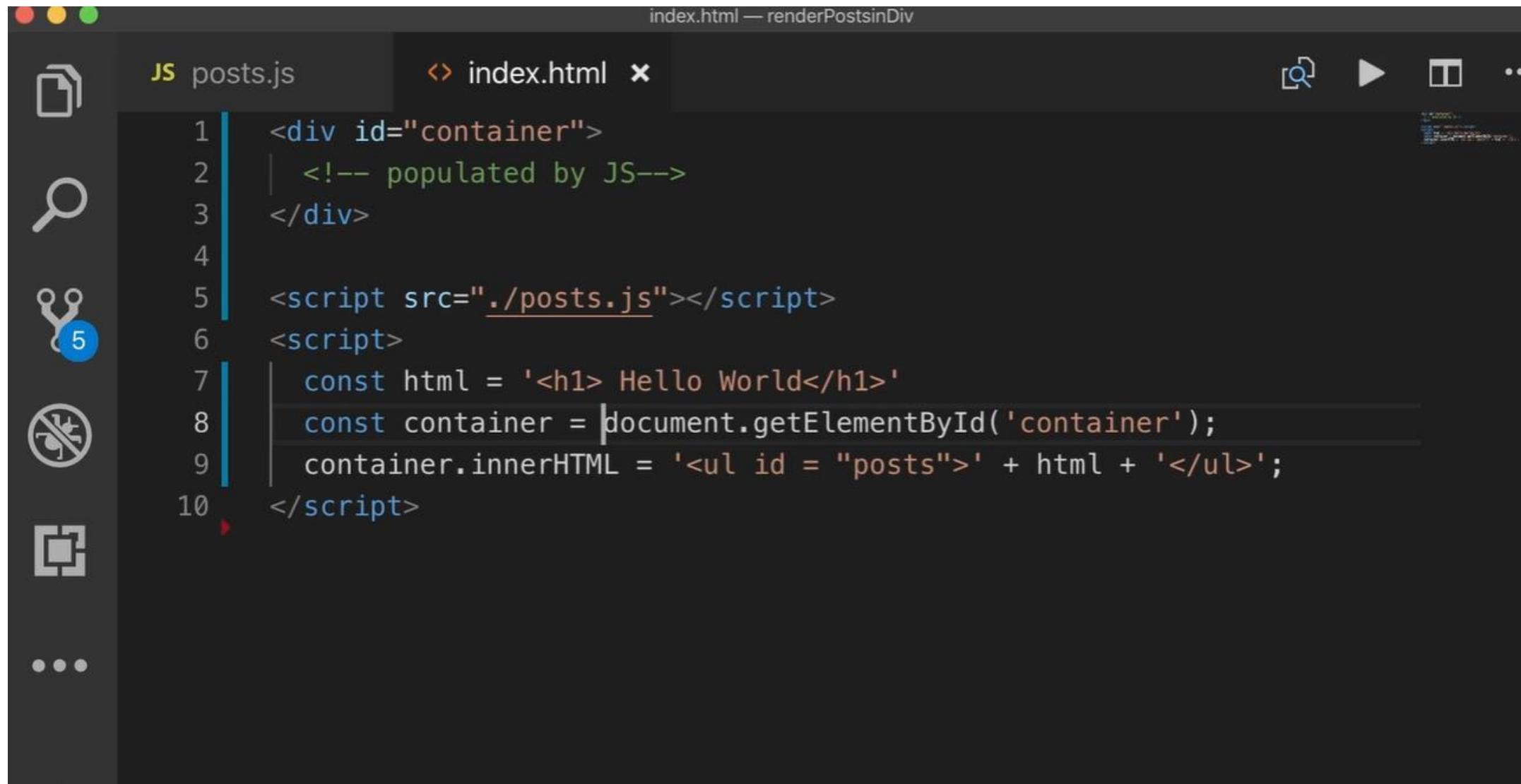


The screenshot shows a code editor interface with two tabs: "posts.js" and "index.html". The "posts.js" tab is active, displaying the following code:

```
1 <div id="container"><!-- populated by JS--></div>
2
3 <script src="./posts.js"></script>
4 <script>
5   const html = '<h1> Hello World</h1>';
6   container.innerHTML = '<ul id = "posts">' + html + '</ul>';
7
8 </script>
```

The "index.html" tab is visible in the background. The sidebar on the left contains several icons with numbers: a file icon with "1", a magnifying glass icon with "5", a wrench icon with "5", and a square icon with "...".

## Dynamically Inject Posts Into Div (3/8)



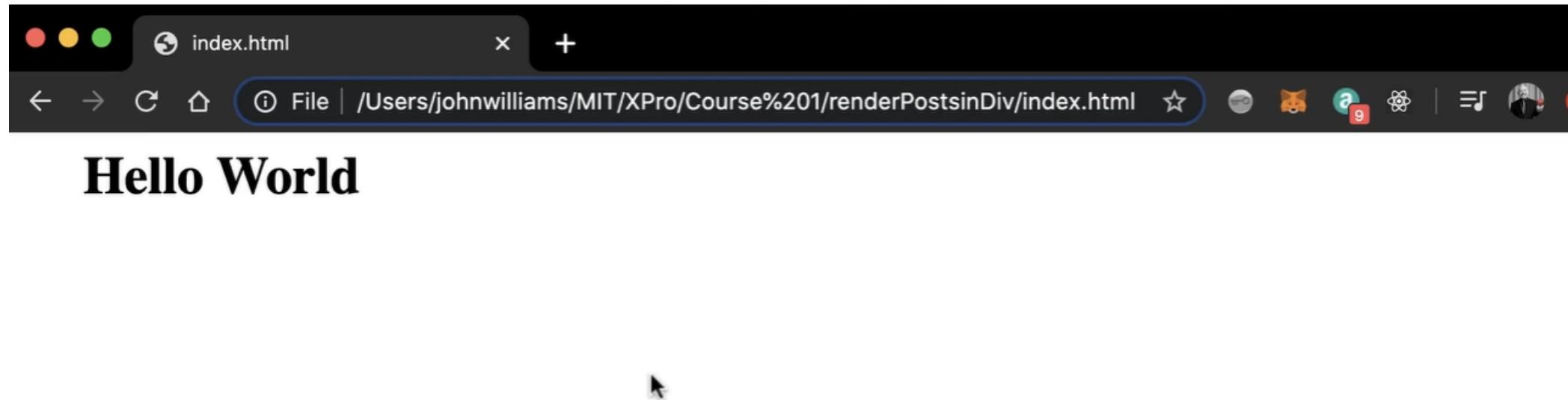
The screenshot shows a dark-themed instance of Visual Studio Code with two tabs open: `index.html` and `posts.js`. The `posts.js` tab is active, displaying the following code:

```
1 <div id="container">
2 |   <!-- populated by JS--&gt;
3 &lt;/div&gt;
4
5 &lt;script src="../posts.js"&gt;&lt;/script&gt;
6 &lt;script&gt;
7   const html = '&lt;h1&gt; Hello World&lt;/h1&gt;';
8   const container = document.getElementById('container');
9   container.innerHTML = '&lt;ul id = "posts"&gt;' + html + '&lt;/ul&gt;';
10 &lt;/script&gt;</pre>

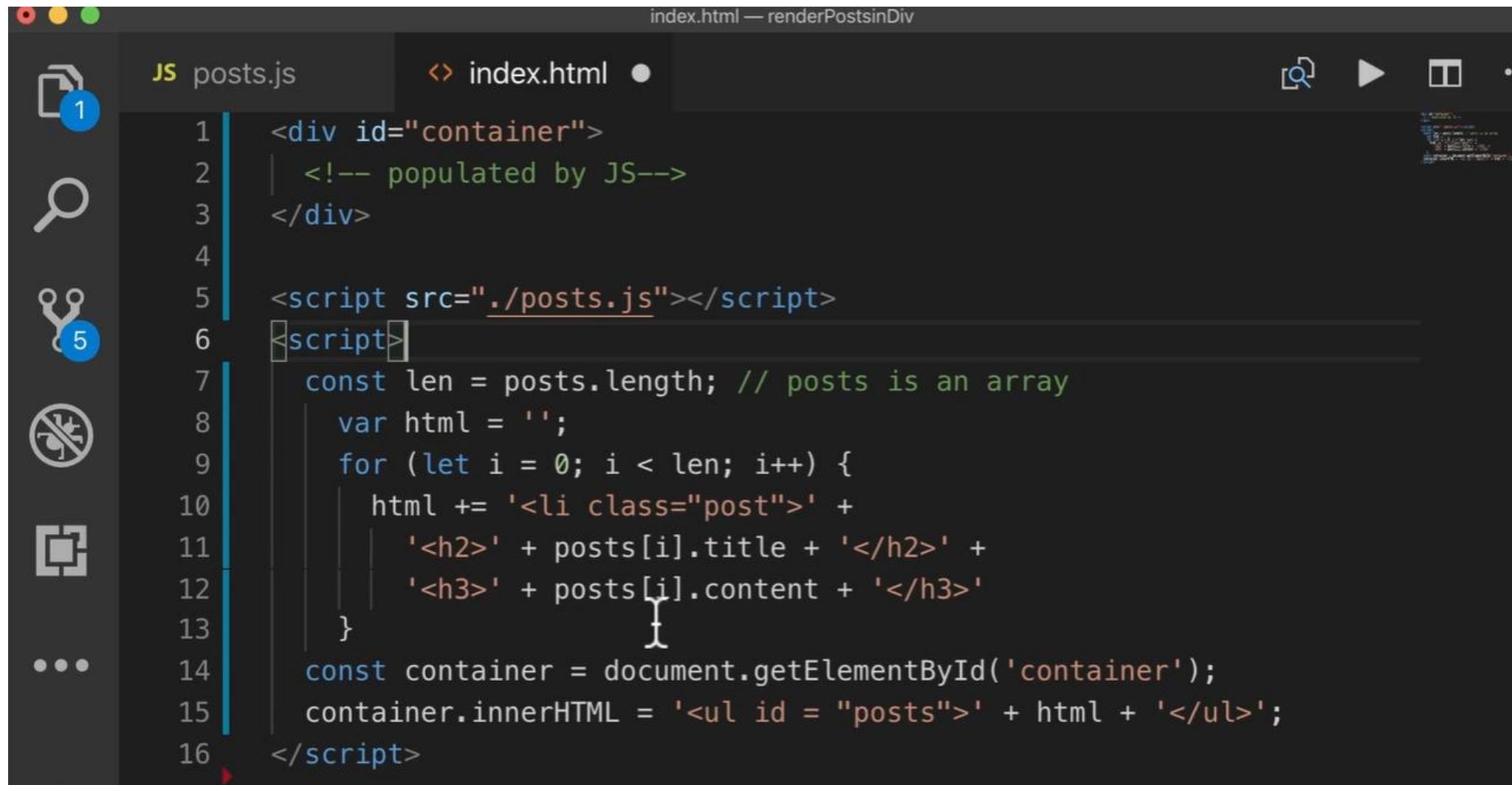
The sidebar on the left contains several icons: a file, a magnifying glass, a gear with a '5' (indicating five changes), a crossed-out bug, and a refresh/cube icon. At the bottom left, there are three dots indicating more content.


```

## Dynamically Inject Posts Into Div (4/8)



# Dynamically Inject Posts Into Div (5/8)



The screenshot shows a code editor interface with two tabs: "posts.js" and "index.html".

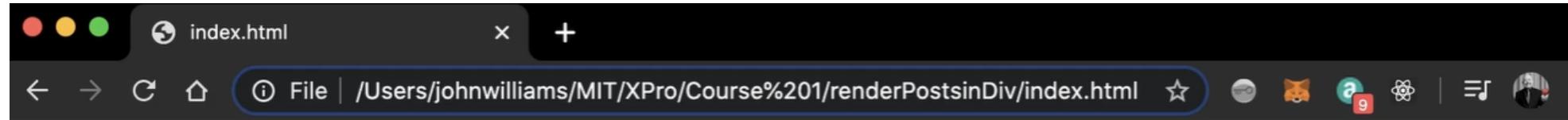
**posts.js:**

```
1 <div id="container">
2 | <!-- populated by JS-->
3 </div>
4
5 <script src=".//posts.js"></script>
6 <script>
7   const len = posts.length; // posts is an array
8   var html = '';
9   for (let i = 0; i < len; i++) {
10     html += '<li class="post">' +
11       '<h2>' + posts[i].title + '</h2>' +
12       '<h3>' + posts[i].content + '</h3>'
13   }
14   const container = document.getElementById('container');
15   container.innerHTML = '<ul id = "posts">' + html + '</ul>';
16 </script>
```

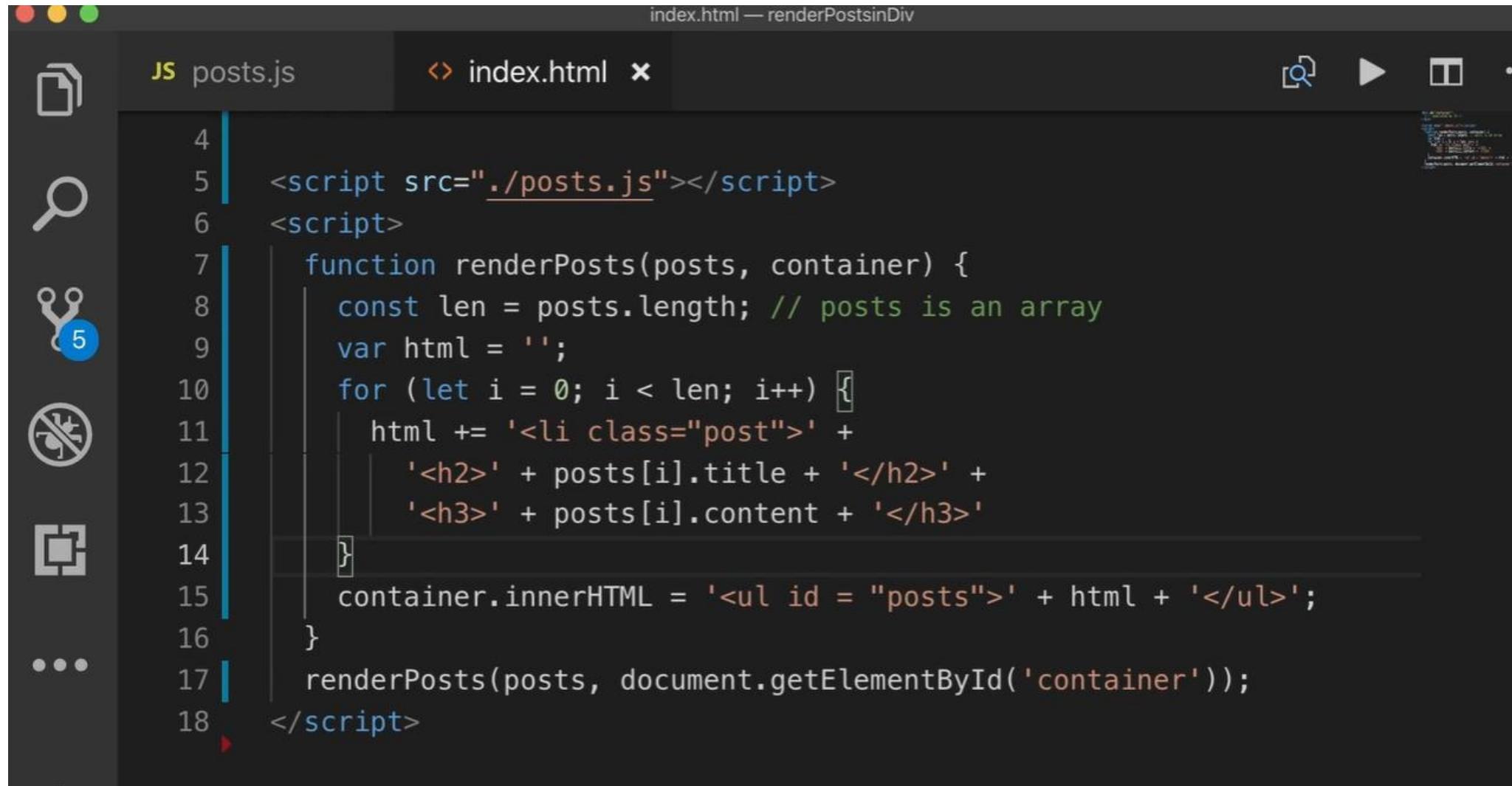
**index.html:**

```
<div id="container">
| <!-- populated by JS-->
</div>
```

# Dynamically Inject Posts Into Div (6/8)



# Dynamically Inject Posts Into Div (7/8)

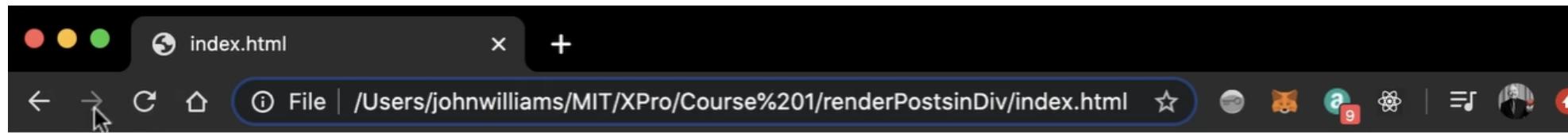


The image shows a code editor interface with two tabs: "posts.js" and "index.html". The "posts.js" tab is active, displaying the following JavaScript code:

```
4 <script src="./posts.js"></script>
5 <script>
6     function renderPosts(posts, container) {
7         const len = posts.length; // posts is an array
8         var html = '';
9         for (let i = 0; i < len; i++) {
10             html += '<li class="post">' +
11                 '<h2>' + posts[i].title + '</h2>' +
12                 '<h3>' + posts[i].content + '</h3>'
13         }
14     }
15     container.innerHTML = '<ul id = "posts">' + html + '</ul>';
16 }
17 renderPosts(posts, document.getElementById('container'));
18 </script>
```

The "index.html" tab is visible in the background. On the left side of the editor, there is a vertical toolbar with several icons: a file icon, a magnifying glass icon, a gear icon with the number "5", a circular icon with a checkmark, a square icon, and three dots at the bottom.

# Dynamically Inject Posts Into Div (8/8)



- **MIT**

- great place to hang out**

- **cricket**

- takes a long time to master**

- **marmalade**

- chunky is the best**

- **Good Article**

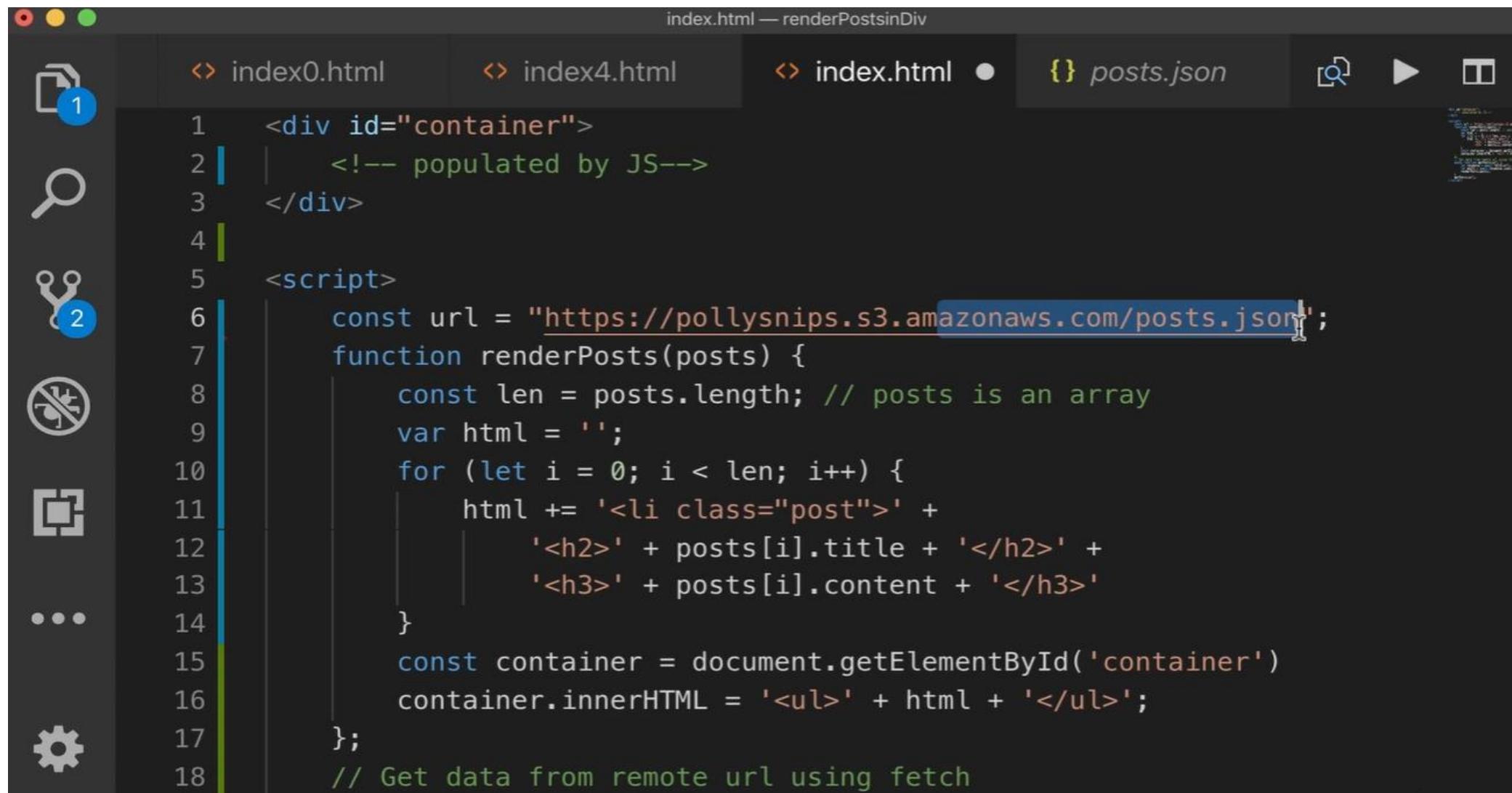
- Death of Planning**

- **Cooking**

- scrambled eggs on toast**

JEEE | xPRO

# Render Using Fetch (1/9)

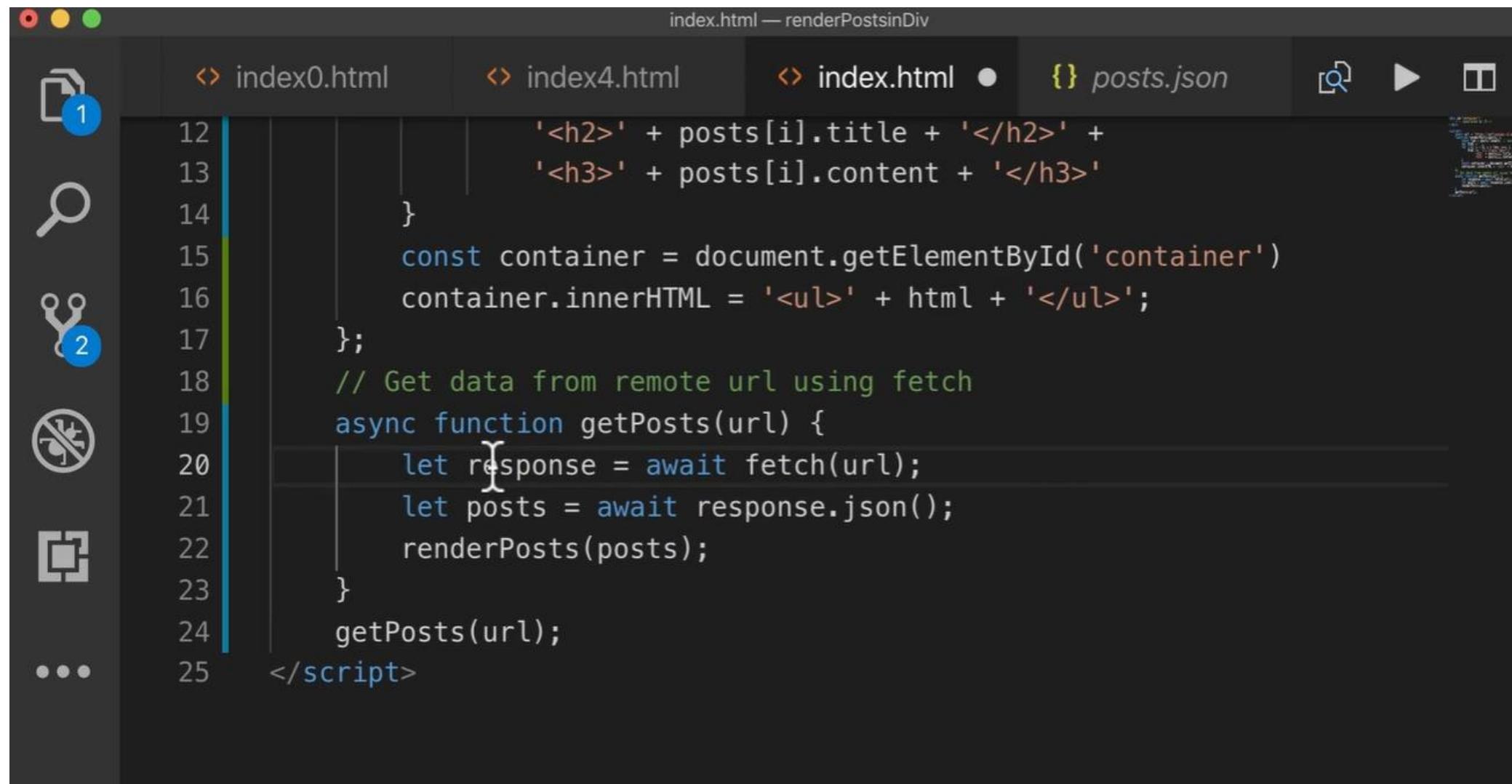


The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons: a file icon with a '1' (highlighted), a magnifying glass, a wrench, a trash can, a square icon, three dots, and a gear. The main area has tabs for 'index0.html', 'index4.html', 'index.html' (which is active, indicated by a black dot), and 'posts.json'. The 'index.html' tab contains the following code:

```
1 <div id="container">
2 |   <!-- populated by JS-->
3 </div>
4
5 <script>
6   const url = "https://pollysnips.s3.amazonaws.com/posts.json";
7   function renderPosts(posts) {
8     const len = posts.length; // posts is an array
9     var html = '';
10    for (let i = 0; i < len; i++) {
11      html += '<li class="post">' +
12        '<h2>' + posts[i].title + '</h2>' +
13        '<h3>' + posts[i].content + '</h3>'
14    }
15    const container = document.getElementById('container')
16    container.innerHTML = '<ul>' + html + '</ul>';
17  };
18 // Get data from remote url using fetch
```

The URL 'https://pollysnips.s3.amazonaws.com/posts.json' is highlighted in blue.

## Render Using Fetch (2/9)

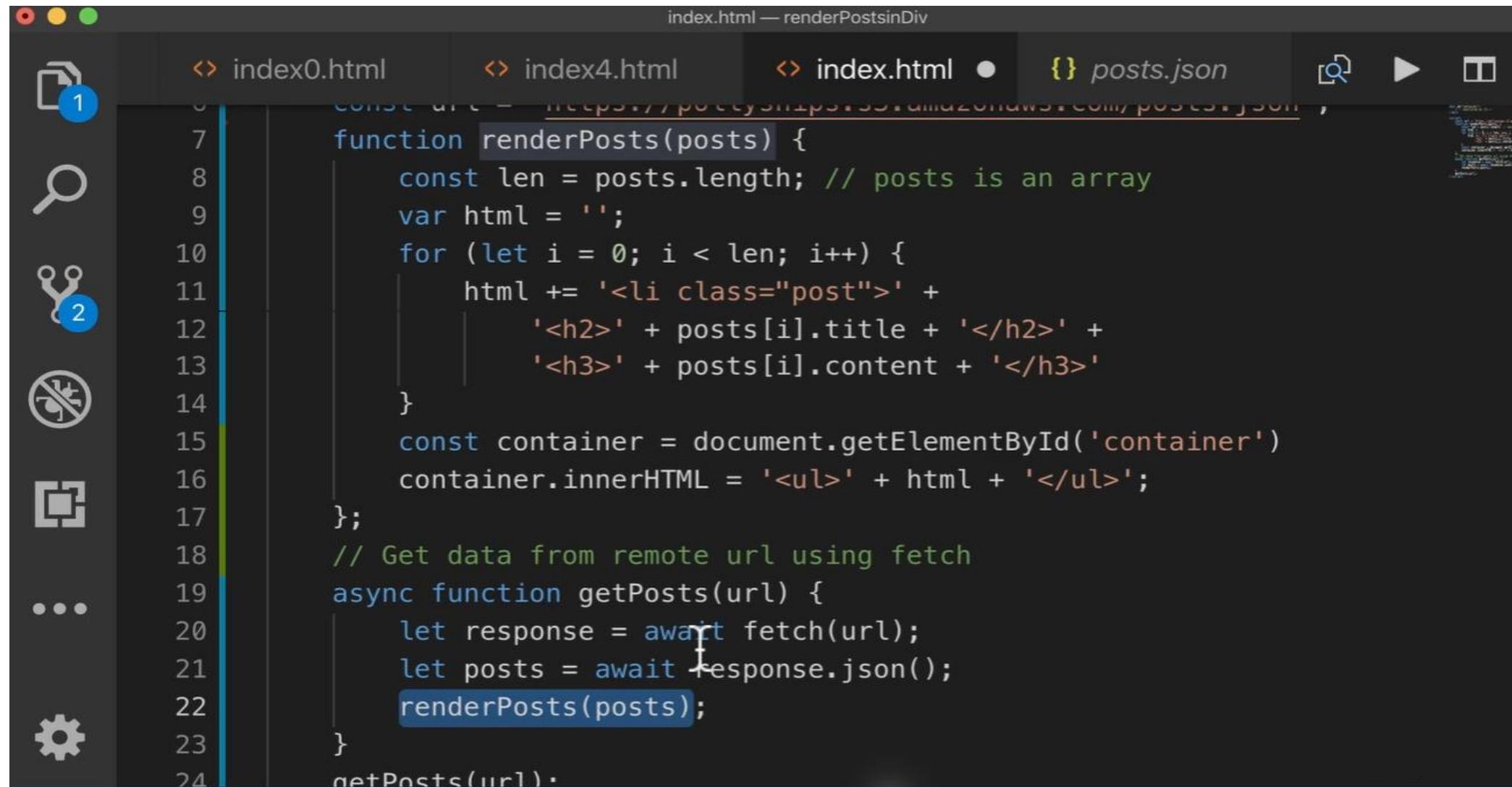


The screenshot shows a code editor window titled "index.html — renderPostsinDiv". The file contains the following JavaScript code:

```
index.html — renderPostsinDiv
index0.html index4.html index.html posts.json
12   ' <h2>' + posts[i].title + '</h2>' +
13   ' <h3>' + posts[i].content + '</h3>'
14 }
15 const container = document.getElementById('container')
16 container.innerHTML = '<ul>' + html + '</ul>';
17 };
18 // Get data from remote url using fetch
19 async function getPosts(url) {
20   let response = await fetch(url);
21   let posts = await response.json();
22   renderPosts(posts);
23 }
24 getPosts(url);
25 </script>
```

The code uses the Fetch API to retrieve data from a remote URL and then renders it into an HTML page using template literals and the `innerHTML` property.

## Render Using Fetch (3/9)



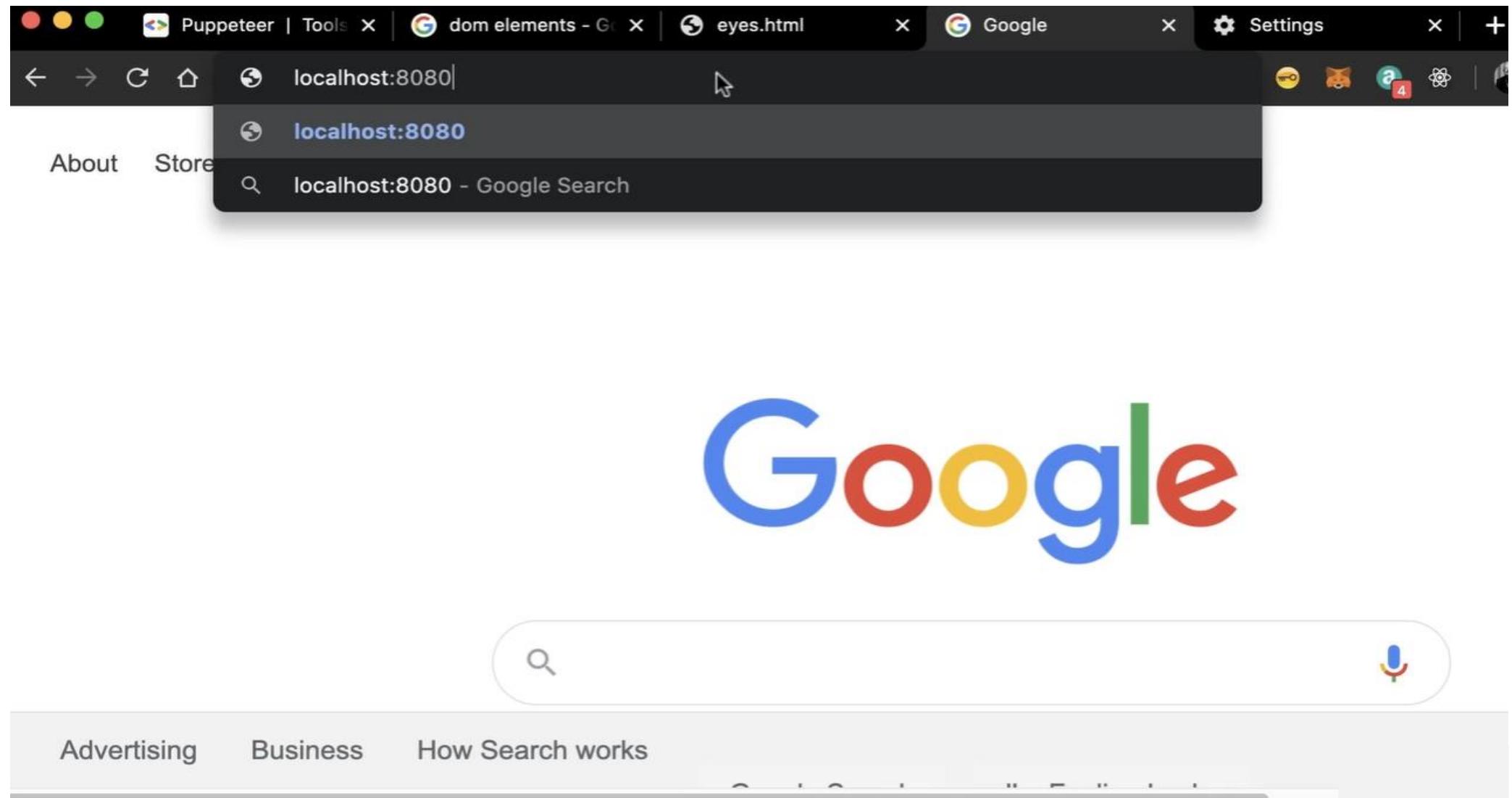
The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons: a file (1), a magnifying glass (2), a crossed-out document (3), a square (4), three dots (5), and a gear (6). The main area has tabs for 'index0.html', 'index4.html', 'index.html' (which is the active tab), and 'posts.json'. The 'index.html' tab contains the following code:

```
index.html — renderPostsinDiv
index.html
const url = "https://polycamps.s3.amazonaws.com/posts.json";
function renderPosts(posts) {
  const len = posts.length; // posts is an array
  var html = '';
  for (let i = 0; i < len; i++) {
    html += '<li class="post">' +
      '<h2>' + posts[i].title + '</h2>' +
      '<h3>' + posts[i].content + '</h3>';
  }
  const container = document.getElementById('container')
  container.innerHTML = '<ul>' + html + '</ul>';
}
// Get data from remote url using fetch
async function getPosts(url) {
  let response = await fetch(url);
  let posts = await response.json();
  renderPosts(posts);
}

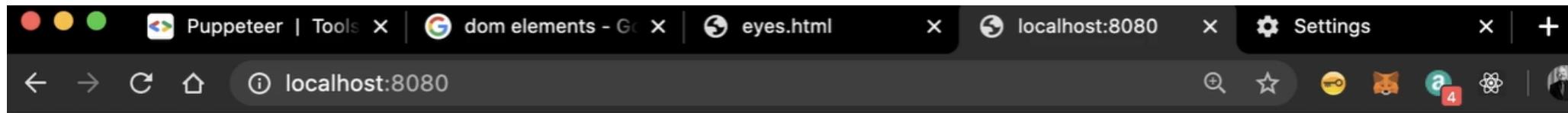
getPosts(url);
```

The 'posts.json' tab is also visible, showing an array of objects with 'title' and 'content' fields.

## Render Using Fetch (4/9)



# Render Using Fetch (5/9)



• **MIT**

**great place to hang out**

• **cricket**

**takes a long time to master**

• **marmalade**

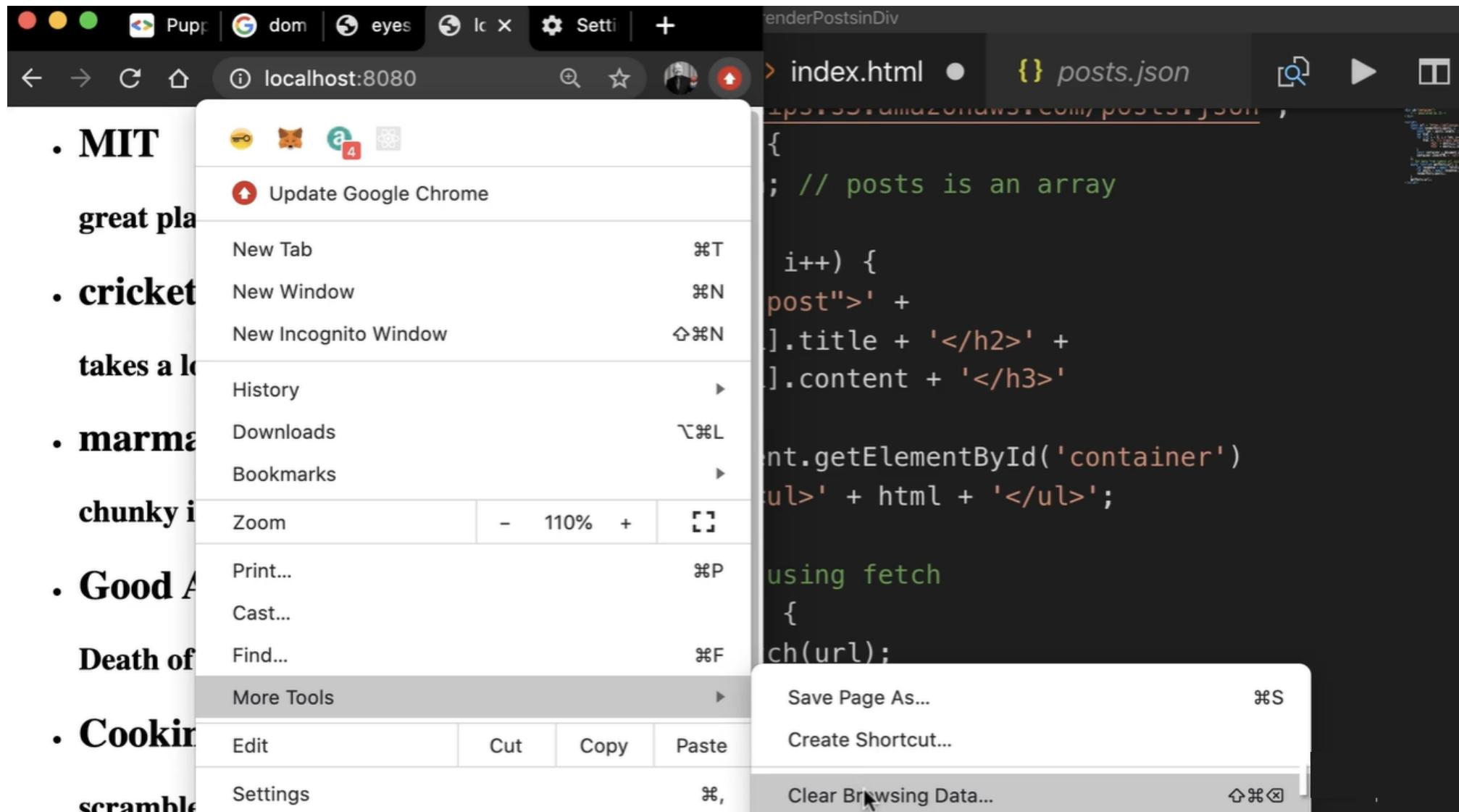
**chunky is the best**

• **Good Article**

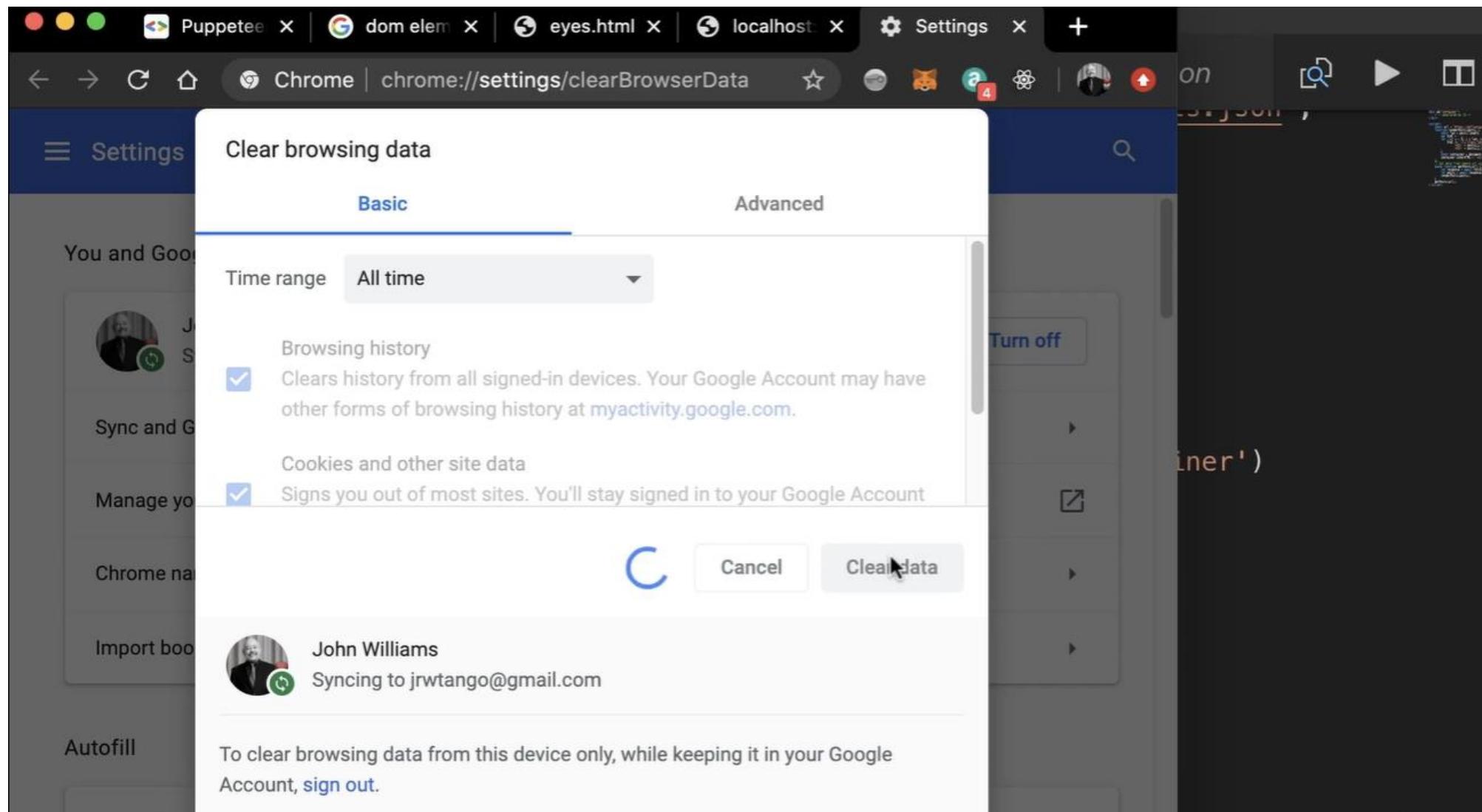
**Death of Planning**

• **Cooking**

# Render Using Fetch (6/9)



# Render Using Fetch (7/9)



# Render Using Fetch (8/9)

The screenshot shows a browser window with the address bar set to `localhost:8080`. The page content is a list of items, each preceded by a bullet point and bolded text:

- **MIT**
- great place to hang out
- **cricket**
- takes a long time to master
- **marmalade**
- chunky is the best
- **Good Article**
- Death of Planning**
- **Cooking**
- scrambled eggs on toast**

To the right of the browser window is the **Network** tab of the Chrome DevTools. It displays a timeline with three requests:

Name	Status	Type
localhost	200	document
posts.js	200	script
posts.json	200	fetch

At the bottom of the DevTools, the status bar shows: 3 requests | 2.7 kB transferred | 1.5 kB resources | Finish: 423.

# Render Using Fetch (9/9)

The screenshot shows a browser window with the URL `pollysnips.s3.amazonaws.com`. The page content displays an array of objects:

```
[{"title": "MIT", "content": "great place to hang out"}, {"title": "cricket", "content": "takes a long time to master"}, {"title": "marmalade", "content": "chunky is the best"}, {"title": "Good Article", "content": "Death of Planning"}, {"title": "Cooking", "content": "scrambled eggs on toast"}]
```

To the right, the Chrome DevTools Network tab is open, showing a timeline with four requests. The requests are color-coded: blue, orange, purple, and green. The green request for `posts.json` is currently selected. The Network tab summary at the bottom indicates 3 requests, 2.7 kB transferred, and 1.5 kB resources.

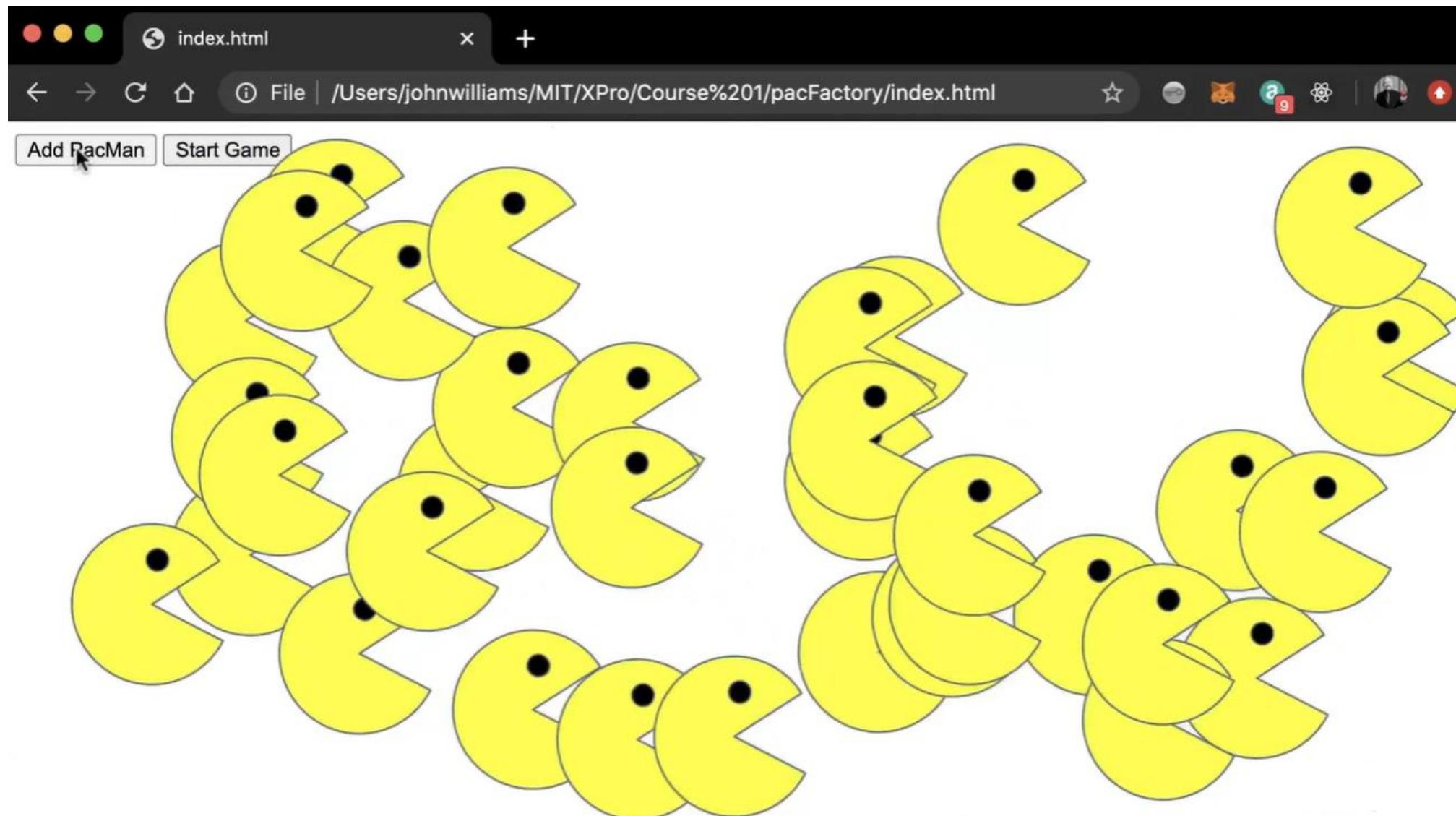
Name	Headers
localhost	
posts.js	
posts.json	

Details for the selected request (`posts.json`):

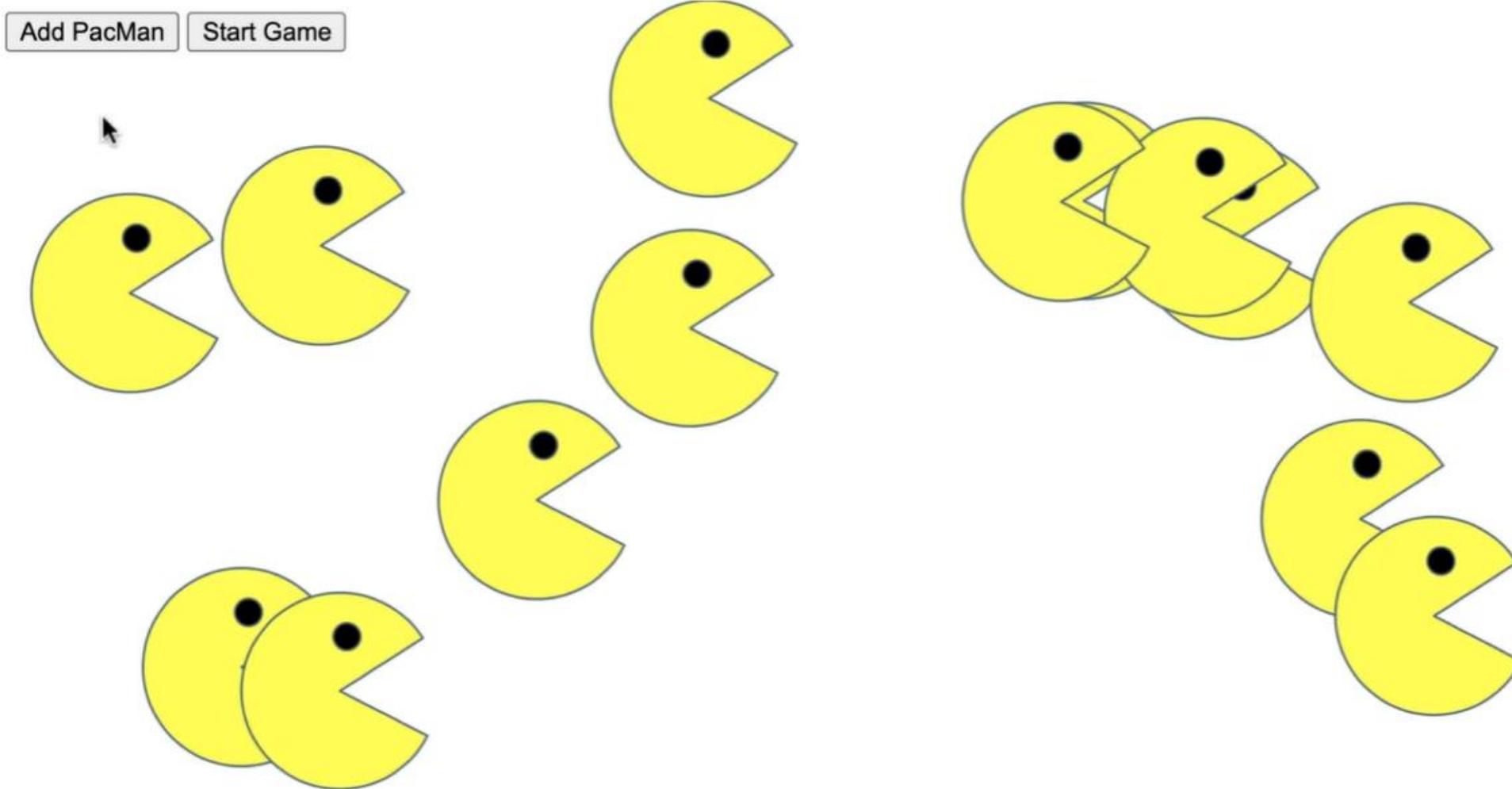
- Request ID: [REDACTED]
- Method: GET
- URL: `http://pollysnips.s3.amazonaws.com/posts.json`
- Response Headers:
  - Content-Type: application/json; charset=UTF-8
  - Date: Mon, 12 Oct 2020 18:15:00 GMT
  - Content-Length: 204
  - Content-Encoding: gzip
  - Connection: keep-alive
  - Vary: Accept-Encoding
  - Accept-Ranges: bytes
- Request Headers:
  - Host: pollysnips.s3.amazonaws.com
  - Accept: \*/\*
  - Accept-Encoding: gzip, deflate
  - User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36
  - Sec-Fetch-Site: same-origin
  - Sec-Fetch-Mode: cors
  - Sec-Fetch-Dest: empty
  - Referer: http://pollysnips.s3.amazonaws.com/
  - Accept-Language: en-US,en;q=0.9
- Timing: 50 ms - 100 ms - 150 ms - 200 ms
- Latency: 50 ms
- Time: 100 ms
- Content:

```
[{"title": "MIT", "content": "great place to hang out"}, {"title": "cricket", "content": "takes a long time to master"}, {"title": "marmalade", "content": "chunky is the best"}, {"title": "Good Article", "content": "Death of Planning"}, {"title": "Cooking", "content": "scrambled eggs on toast"}]
```

# Factory For PacMen (1/2)



## Factory For PacMen (2/2)



# Design Issues

- 1 How do we generate lots of PacMen?
- 2 We need a Factory that can produce them one at a time. We can put them into an array. Eg pacMen = [....]
- 3 How do we draw them on the screen?
- 4 With every pacMan we make we need to add an <img> tag to the DOM. We can add them as children to a <div> tag.
- 5 Each pacMan needs to keep track of its position and its velocity
  - Maybe position is an object {x: 33, y: 22}. Same for velocity = {x: 8, y:-4}
  - We need to set the style.left and style.top of the <img> tag

## Code Snippet (1/3)

```
// Factory to make a PacMan
function makePac() {
    // returns an object with values scaled {x: 33, y: 21}
    let velocity = setToRandom(10);
    let position = setToRandom(200);
    // Add image to div id = game
    let game = document.getElementById('game');
    let newimg = document.createElement('img');
    newimg.style.position = 'absolute';
    newimg.src = 'PacMan1.png';
    newimg.width = 100;
    newimg.style.left = position.x;
    newimg.style.top = position.y;
    game.appendChild(newimg);
    // new style of creating an object
    return { position, velocity, newimg }
}
```

## Code Snippet (2/3)

```
function update() {
    //loop over pacmen array and move each one and move image in DOM
    pacMen.forEach((item) => {
        checkCollisions(item)
        item.position.x += item.velocity.x;
        item.position.y += item.velocity.y;

        item.newimg.style.left = item.position.x;
        item.newimg.style.top = item.position.y;
    })
    setTimeout(update, 20);
}
```

```
function checkCollisions(item) {
    if (item.position.x + item.velocity.x + item.newimg.width > window.innerWidth ||
        item.position.x + item.velocity.x < 0) item.velocity.x = -item.velocity.x;
    if (item.position.y + item.velocity.y + item.newimg.height > window.innerHeight ||
        item.position.y + item.velocity.y < 0) item.velocity.y = -item.velocity.y;
}
```

## Code Snippet (3/3)

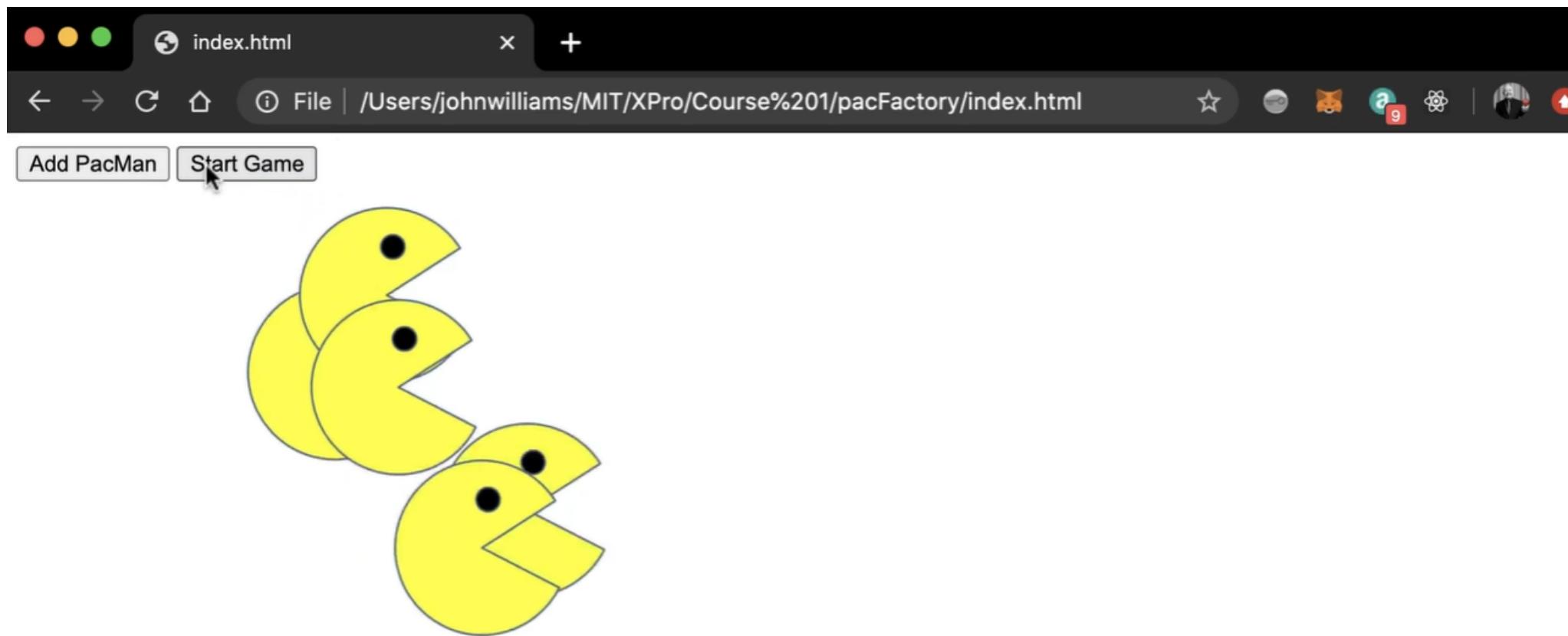
```
function makeOne() {  
    pacMen.push(makePac()); // add a new PacMan  
}
```

```
<body>  
    <div id='game'>  
        function makeOne(): void  
        <button onclick='makeOne()' width='200' height='30'>Add PacMan</button>  
        <button onclick='update()' width='200' height='30'>Start Game</button>  
    </div>  
</body>
```

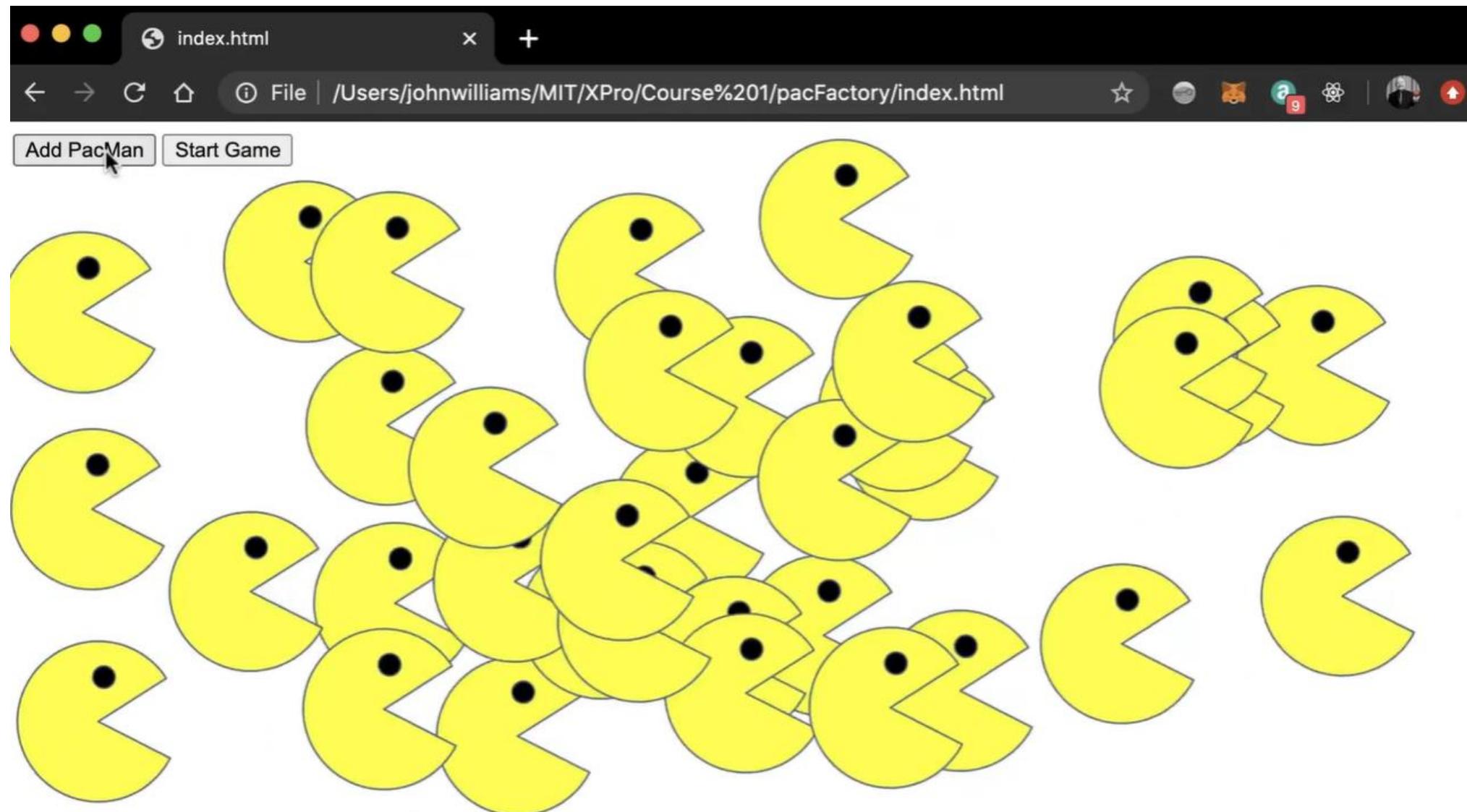
# Outputs (1/3)



## Outputs (2/3)



# Outputs (3/3)





© MIT xPRO 2021. All rights reserved.