Terminal & HTML

Software Development Bootcamp

Topic

Terminal

What Is Code?

Code is saying one thing to mean something else. In computer science, it refers to how we write out instructions.

Code is a *tool* used to give the computer *instructions* to later follow.

These instructions are called **programs**.

Programmatic Thinking

It's easy to immediately start writing code. Don't. Figure out the instructions needed first.

- Imagination and scratch paper go a long way when planning.
- Computers follow programs exactly as written.
- Computers *sometimes* show error messages to help the programmer.
- Always **test** *before* submitting.
- Mistakes are expected and respected. Try again.

Programming Languages

Every program is written in a **language**:

- Anyone can make a computer language
- Different languages are good at different things

Computer languages are **not ambiguous**. Specific words mean specific things.

The Terminal

We use the **Terminal** application to tell the computer commands in computer language.

- The terminal gives us a **command line** that is managed by a **shell**.
- The **console** in your browser is a kind of terminal.
- *command line*, *console*, *shell*, and *terminal* are often used interchangeably.

GUI vs CLI

Most people use **GUI (Graphical User Interface)** applications that display text boxes, buttons, scrolls bars, and more for interaction.

In the terminal, you use **CLI (Command Line Interface)** applications. You can only interact with these via commands, or strings of text entered into the terminal.

Directories

Directories are locations on your computer that you will visit via the terminal.

Directories can contain other directories and any file:

- Documents
- Photos
- Videos
- MP3s
- And more

Navigation

The terminal is always referencing some directory, as if it were inside that folder.

Navigating your computer is like exploring rooms in a house. You always need to know, “Where am I?”

We can use the terminal to say, “*Print* the *working directory*,” by typing `pwd` and hitting Return

The Path

The command line is very dependant on the file **path**. The file path is the string of characters and back slashes at the beginning of your command line. The path continues on into other directories on your computer.

There are a few key commands to help you follow the path forward and backward.

- `.` means “this directory I’m currently inside of”
- `..` means “the directory containing the directory I’m currently inside of”
- `/another-directory` means “look for a directory named `another-directory` inside of the current directory”

Common Path Commands

- **“Print Working Directory”**
  - Command: `pwd`
  - Use Case: shows the name of the current directory.
- **“List”**
  - Command: `ls`
  - Use Case: shows the contents of the current directory
- **“Make Directory”**
  - Command: `mkdir`
  - Use Case: creates a new directory inside the current one
- **“Change Directory”**
  - Command: `cd`
  - Use Case: move into a different directory

Exercise

Terminal Platform

Topic

HTML

What Is Hyper Text Markup Language?

HTML is a coding language composed of various types of tags (elements) used to build web pages.

- Web browsers read HTML and render it as visual elements.
- HTML structures the content, while CSS and JavaScript enhance presentation and interactivity.

Standard Page Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <p>My house is a very fine house with two cats in the yard.</p>
   <p>Life used to be <strong>so hard</strong>; now everything is easy because of you.</p>
</body>
</html>
```

Elements

```
<p>My house is a very fine house with two cats in the yard.</p>
```

HTML elements are typically defined by sets of angle brackets, known as their **tags**.

- Content between opening and closing tags appears on the page.
    - Closing tags have a forward slash.

There are different types of elements: structural, style, and self-closing.

Structural Elements

Structural elements define the structure of the page.

- **<head>**: Contains metadata
- **<body>**: Page body
- ** **: Division
- **<h1>**: Heading (level 1)
- **<p>**: Paragraph
- ****: Text without a line break
- **<img>**: Image

Style Elements

Style elements are typically applied to format text.

- : strong
- : emphasis
-
: Line break
- <hr>: Horizontal rule
- <blockquote>: “Call-out” quotation
- <pre>: Pre-formatted text code

Self-Closing Elements

Self-closing elements are are used for elements that don't have any text content.

- <img>
-
- <input>
- <hr>

Element Attributes

- Attributes define elements and their purposes.
- Example: <img src='/images/cat-pic.jpg' title='Cat Picture' alt='Picture of a fuzzy cat'>
- Attributes include src, alt, style, title, href.

Style vs Layout vs Semantics

- Some tags describe display style (e.g., <b>) and some describe content (e.g.,).
- Semantics convey meaning and purpose.
- Screen readers and search engines rely on semantic tags to understand content structure.

HTML Has Flaws

- Difficult for humans and programs to read.
- Inconsistent whitespace rules.
- Case insensitive, except when it's not (e.g., Ç is Ç and ç is ç).
- Browsers render invalid HTML, leading to propagation of invalid HTML.
- Muddled distinctions between semantics and style.

Proper HTML5 Web Page Structure

- Modern HTML5 elements semantically structure and lay out website content.
- Not all elements are required, but each serves a specific purpose.
- Elements assist screen readers and search engines in determining website structure.

Exercise

Hello HTML