GitHub Copilot Global Bootcamp 😻

Spokane WA

IntelliTect

https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming

IntelliTect

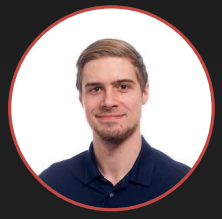
Kevin
Speaker

Phil
Host/Proctor

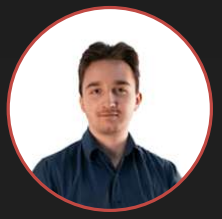
Dan
Proctor

Clayton
Proctor

Reese
Proctor



Casey
Proctor



Andre
Proctor

Mitch
Proctor

Brian
Proctor

Matthew
Proctor

AGENDA

- Introduction to GitHub
- GitHub Codespaces
- GitHub Copilot
- Hands-on Lab
- Next steps


GitHub Copilot in VS Code
FREE
https://github.com/features/copilot

Introduction to GitHub

Globally distributed, Cloud-based development, CI/CD automation

Industry-best collaborative tools for developers

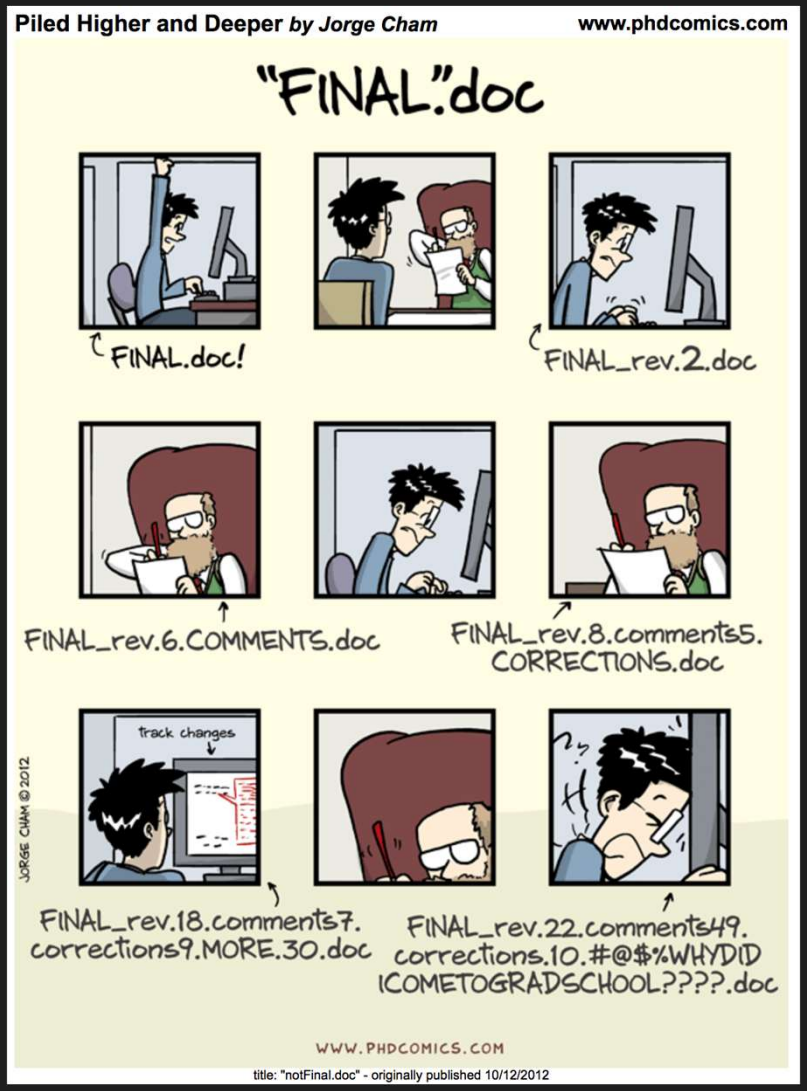
Security at every step of the workflow

Powered By AI

What is Git and GitHub?

Git is a version control system that tracks the history of changes to all files within a folder, known as a repository.

GitHub is a platform where you can store and share Git repositories.

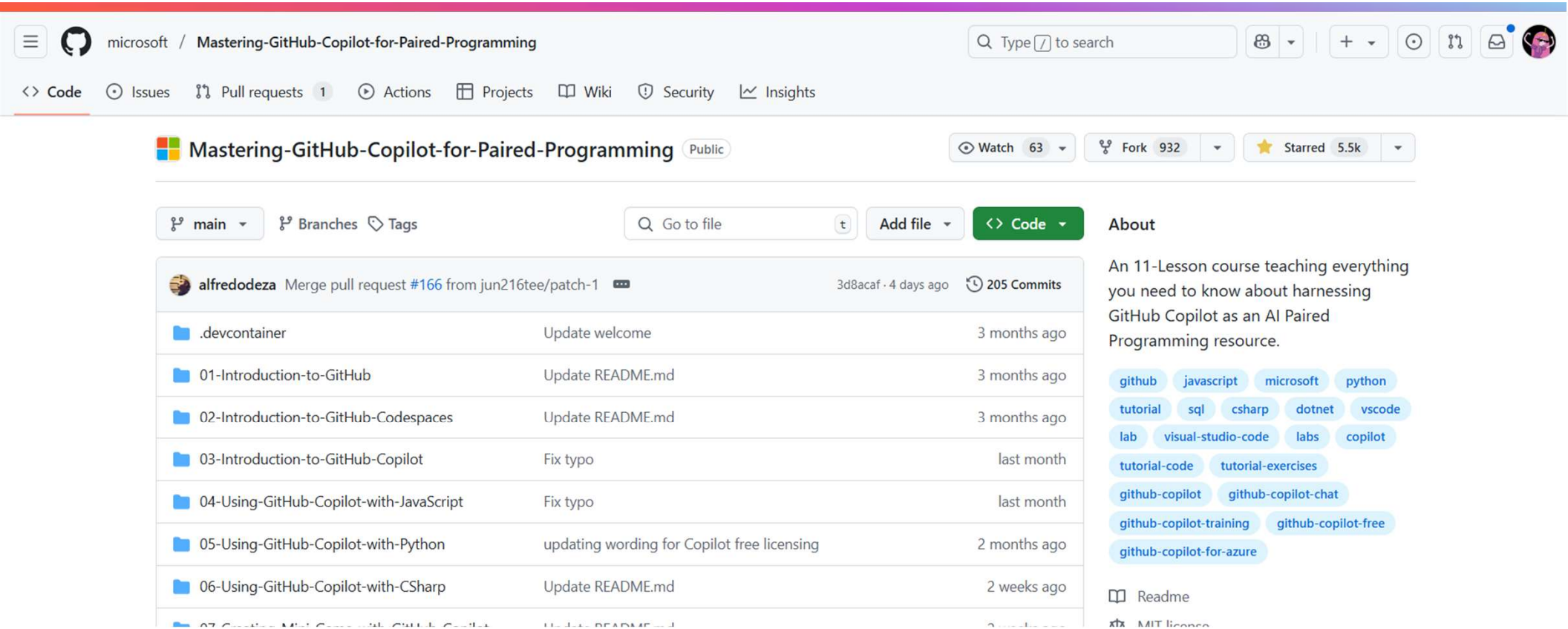


Common Term definitions

Issue	A task, bug, or request that needs to be addressed in a project.
Fork	A personal copy of someone else's repository that allows you to make changes independently.
Branch	Branches create a safe environment to make changes and try things out. Main is "special / reserved" branch.
Commit	A recorded change to a file or set of files.
Push	The action of sending your committed changes to a remote repository.
Pull Request	A way to propose and review changes before merging them into the main project.
Merge(Conflict)	The process of integrating changes from one branch into another.
Markdown	A simple markup language for formatting text and 🤣🌈🦄, commonly used in documentation and README files.

Let’s create a fork!

https://aka.ms/repo/githubcopilot



Introduction to GitHub

Go to the folder to view the instructions and click on the “Star Course” button.

01-Introduction-to-GitHub	Update README.md	3 months ago

In this exercise, you will:

- Create a branch
- Commit a file
- Open a pull request
- Merge your pull request

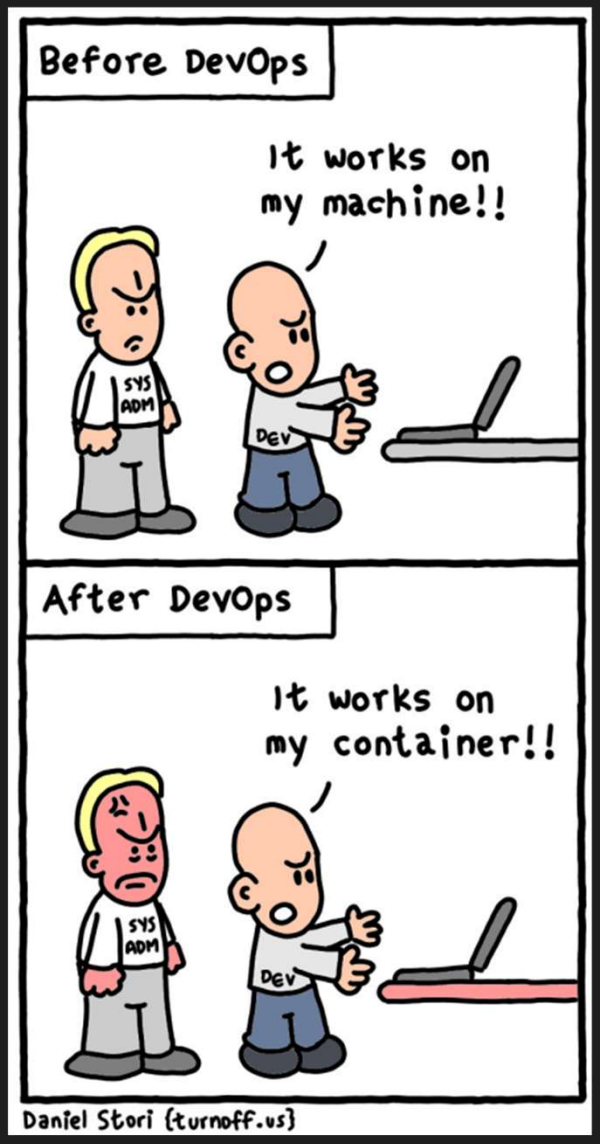
GitHub Foundations Certification

https://education.github.com/experiences/foundations_certificate

Foundations of Git Certification Course

https://learn.gitkraken.com/

Code with GitHub Codespaces



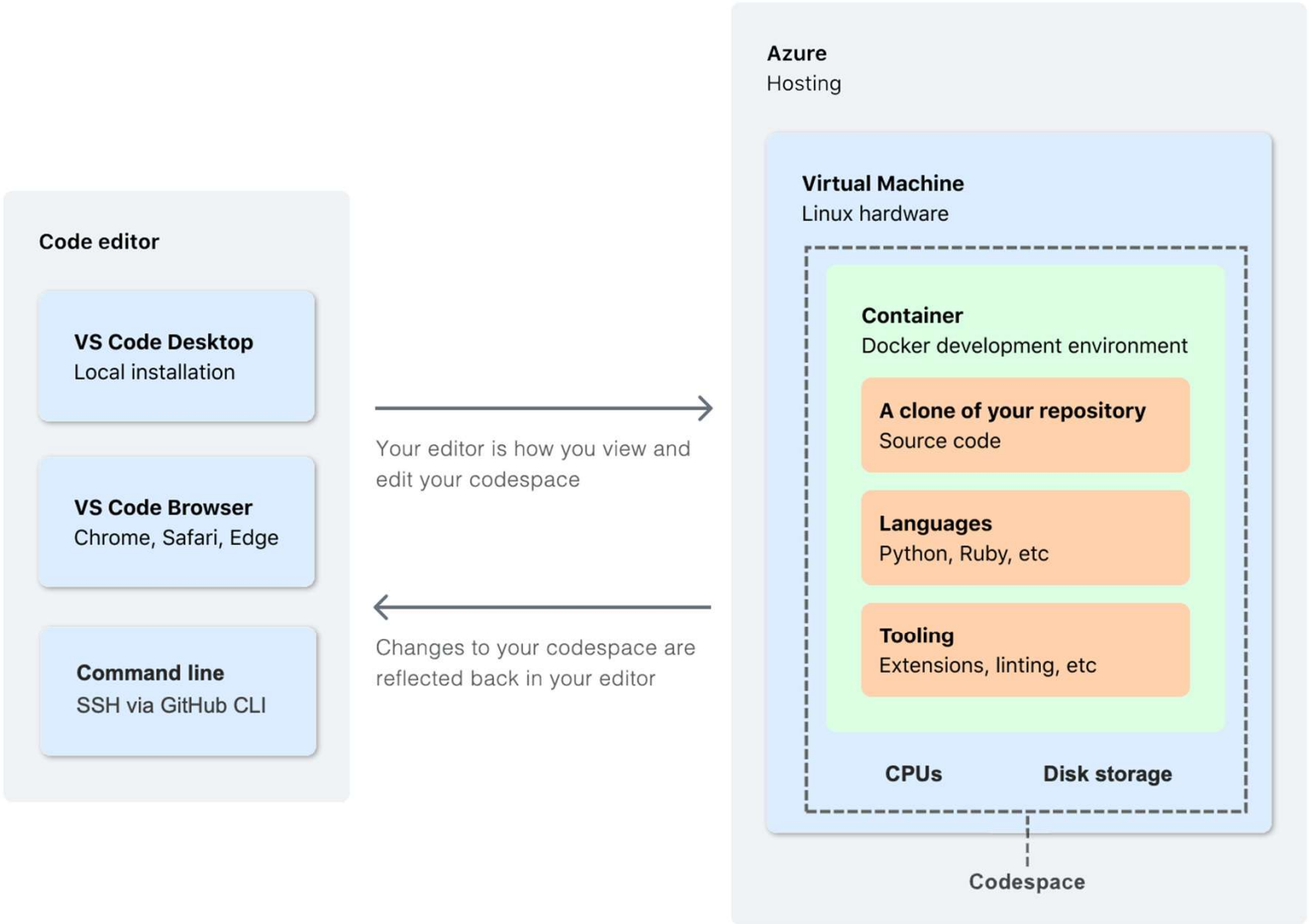
“We lost a whole day getting a QA engineer running on that project.”

“It works on my laptop. Why doesn’t it work on yours?”

“Can I have a 16-core machine?”

“I have no idea what I’ve installed over the years?”


Codespaces
mona-github-github-g59jpq2w5w7.github.dev
EXPLORER
GITHUB
.devcontainer
build-devcontainer.sh
devcontainer.json
on-create-command.sh
post-attach-command.sh
setup-devcontainer.sh
ssrh
welcome_message.txt
.github
.vscode
app
api
assets
components
controllers
helpers
jobs
mailers
models
emotion.rb
devcontainer.json
attr_reader :label
attr_reader :pronounceable_label
Public: Get the Emoji that this reaction's content represents.
#
Returns an Emoji.
attr_reader :emoji_character
def initialize(content:, label: nil, pronounceable_label: nil, emoji_character: nil)
@content = content
@label = label || @content
@pronounceable_label = pronounceable_label || @label
@emoji_character = emoji_character || Emoji.find_by_alias(@content)
@platform_enum = @pronounceable_label.gsub(" ", "_").upcase
freeze
end
end
Emotion.create(content: "+1", pronounceable_label: "thumbs up")
Emotion.create(content: "-1", pronounceable_label: "thumbs down")
CPUs up to
32 cores
Memory up to
64 GB
DEBUG CONSOLE
TestSynchronizer – Manual sync
main
Current branch
0↓ 1↑
css-changes
0↓ 6↑
tabbed-nav
2↓ 1↑


Code editor
VS Code Desktop
Local installation
VS Code Browser
Chrome, Safari, Edge
Command line
SSH via GitHub CLI
Your editor is how you view and edit your codespace
Changes to your codespace are reflected back in your editor
Azure
Hosting
Virtual Machine
Linux hardware
Container
Docker development environment
A clone of your repository
Source code
Languages
Python, Ruby, etc
Tooling
Extensions, linting, etc
CPUs
Disk storage
Codespace

Managed environments for any project

Fully-managed, on-demand dev environments that dramatically reduce setup and maintenance

Feel at home even when away

Environment can be configured, and your settings, themes, and Git identity, are roamed so the experience looks and feels like local

Develop from anywhere

Environment can be configured, and your settings, themes, and Git identity, are roamed so the experience looks and feels like local

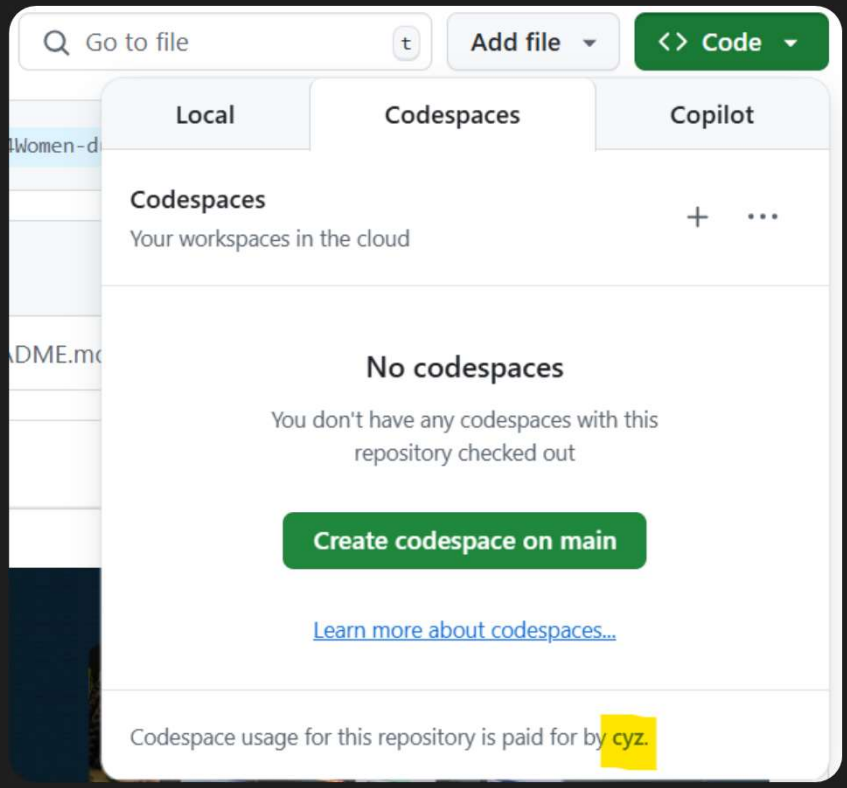
Supercharged collaborative development

Fully collaborative environment, complete with AI-assisted development support

All personal GitHub accounts have a monthly quota of free use of GitHub Codespaces.

The free plan offers 120 hours of usage for 1 core or 60 hours for 2 cores, plus 15 GB of storage per month.

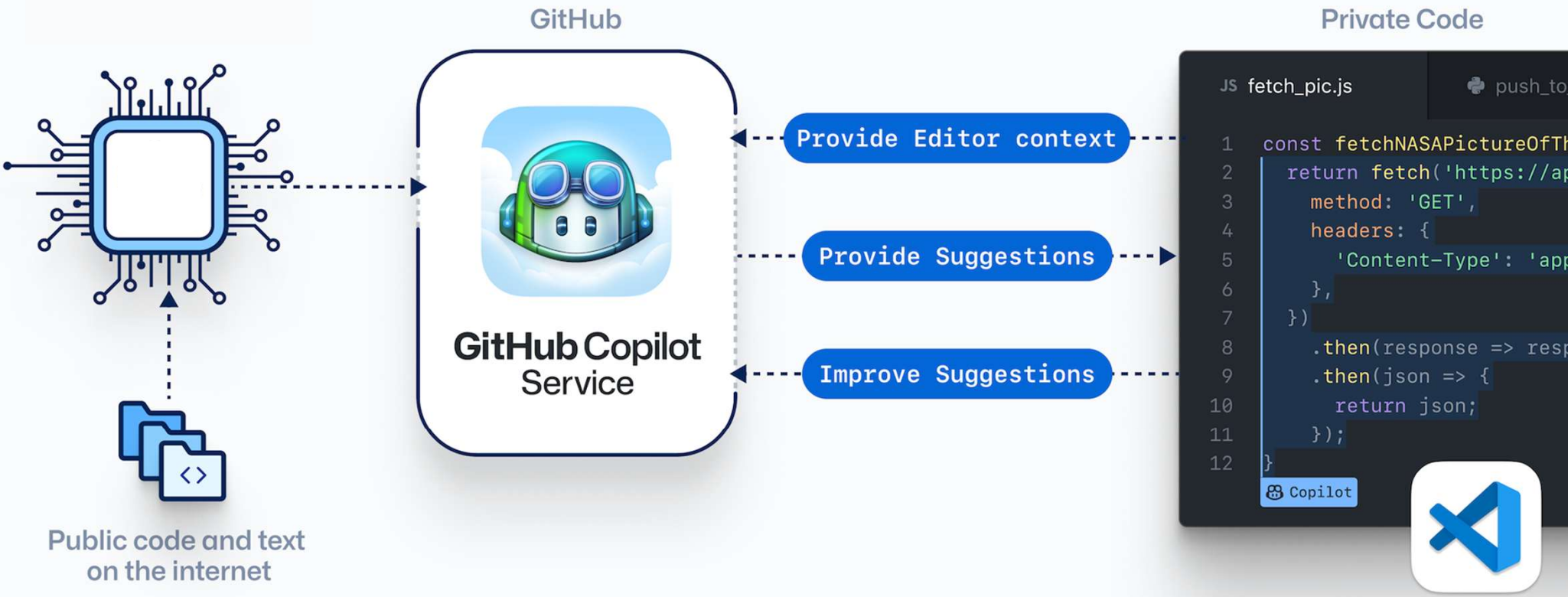
GitHub Pro offers 180 core hours of usage, plus 20 GB of storage per month.



Lab
Code with Codespaces

https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming


method: "POST",
body: `text=${text}`,
headers: {
"Content-Type": "app
},
GitHub Copilot


GitHub
Private Code
Provide Editor context
Provide Suggestions
Improve Suggestions
GitHub Copilot
Service
JS fetch_pic.js
push_to
1 const fetchNASAPictureOfTh
2 return fetch('https://ap
3 method: 'GET',
4 headers: {
5 'Content-Type': 'app
6 },
7 })
8 .then(response => resp
9 .then(json => {
10 return json;
11 });
12 }
Copilot
Public code and text
on the internet

Microsoft and GitHub's six principles of Responsible AI

Fairness

Reliability and Safety

Privacy and Security

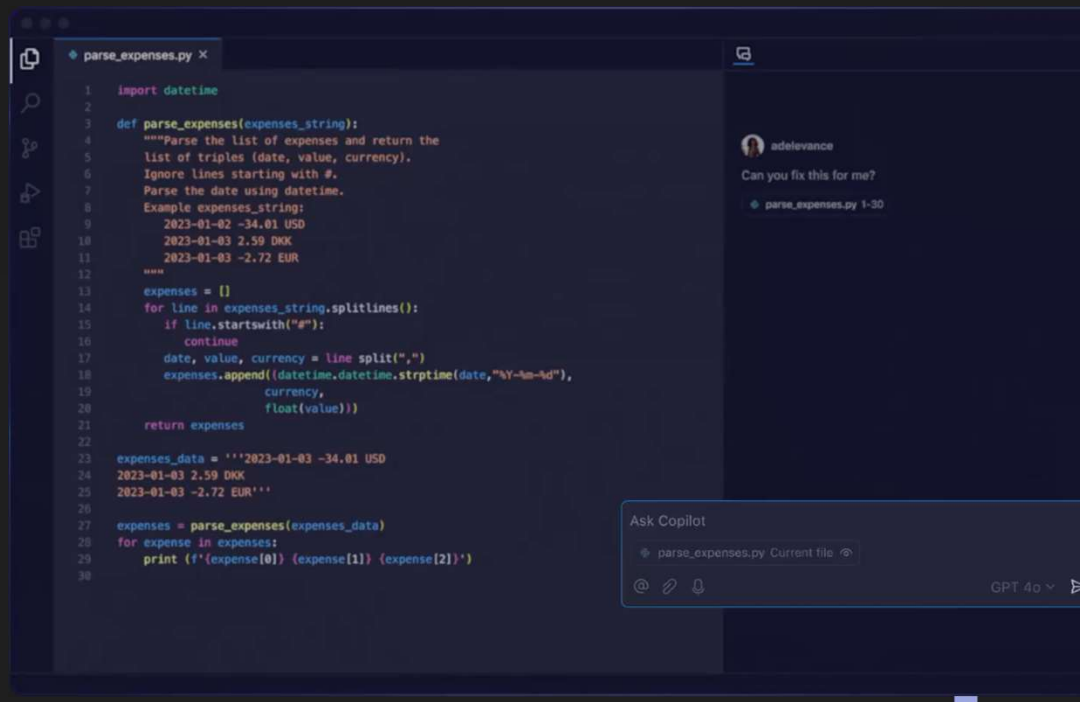
Inclusiveness

Transparency

Accountability

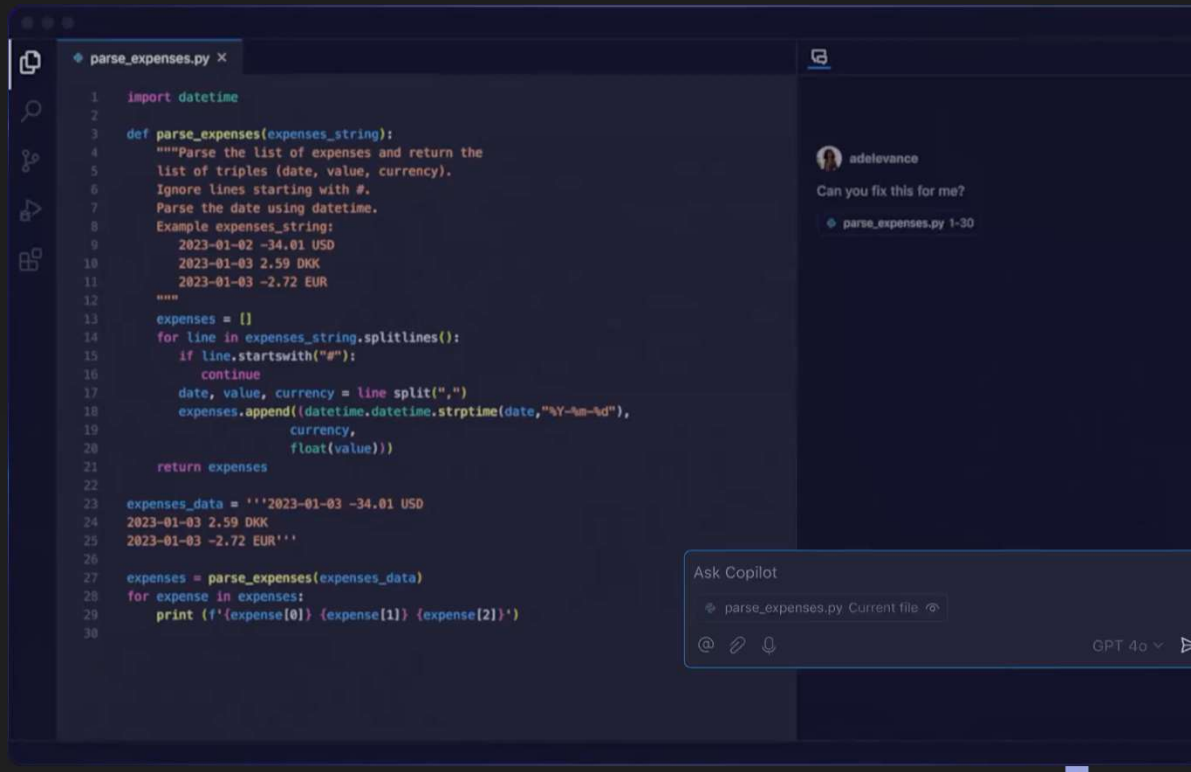
Prompt Engineering

- Your prompt should be single and specific.
- Context is EVERYTHING!
- Iterate, iterate, and iterate.



How Copilot learns from your prompts?

🤔



Zero-shot learning

```
# Zero-shot learning example with natural language processing
description = "This animal has feathers, can fly, and lays eggs."
category = model.predict(description)
print(f"The predicted category is: {category}")
```

```
# Zero-shot learning example in image classification
image = load_image("path/to/image.jpg")
category = model.predict(image)
print(f"The predicted category is: {category}")
```

One-shot learning

```python
# One-shot learning example with image recognition
reference_image = load_image("path/to/reference_image.jpg")
new_image = load_image("path/to/new_image.jpg")
is_same_class = model.compare(reference_image, new_image)
print(f"Is the new image of the same class? {is_same_class}")
```

```python
# One-shot learning example with text classification
reference_text = "This is a sample text of category A."
new_text = "This text should be classified in category A."
is_same_category = model.compare(reference_text, new_text)
print(f"Is the new text in the same category? {is_same_category}")
```

Few-shot learning

```
# Few-shot learning example with natural language processing
examples = [
    {"text": "This is a sample text of category A.", "category": "A"},
    {"text": "This is another example of category A.", "category": "A"},
    {"text": "This text belongs to category B.", "category": "B"}
]
new_text = "This new text should be classified in category A or B."
predicted_category = model.predict(new_text, examples)
print(f"The predicted category is: {predicted_category}")
```

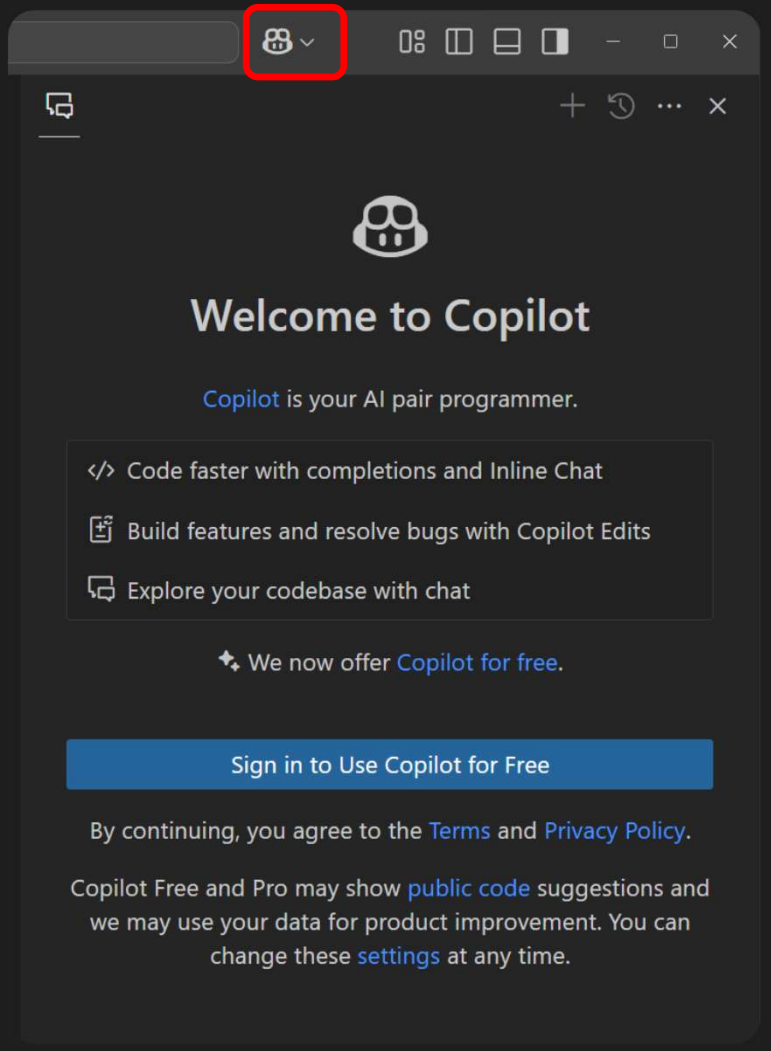
Few-shot learning

```
# Few-shot learning example with image classification
examples = [
    {"image": load_image("path/to/class_A_image1.jpg"), "category":
"A"}{"image": load_image("path/to/class_A_image2.jpg"), "category":
"A"}{"image": load_image("path/to/class_B_image1.jpg"), "category": "B"}
]
new_image = load_image("path/to/new_image.jpg")
predicted_category = model.predict(new_image, examples)
print(f"The predicted category is: {predicted_category}")
```

GitHub Copilot time! 😻

If you have not activated your account yet:

- Open the Codespaces in the lab and click on the GitHub Copilot icon.
- In the GitHub Chat panel, click the option to use GitHub Copilot for free to activate.




Lab
Getting Started with
GitHub Copilot
https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming

Slide 28

KB1 Kevin Bost, 2025-06-19T17:08:11.069

How to pick the right model? 🤔

The best model ***depends*** on your use case:

For **balance between cost and performance**:
Try GPT-4.1 or Claude Sonnet 3.7.

For **fast, low-cost support for basic tasks:**
Try o4-mini or Claude Sonnet 3.5.

For **deep reasoning or complex coding challenges:**
Try o3, GPT-4.5, or Claude Sonnet 3.7.

For **multimodal inputs and real-time performance:**
Try Gemini 2.0 Flash or GPT-4.1.

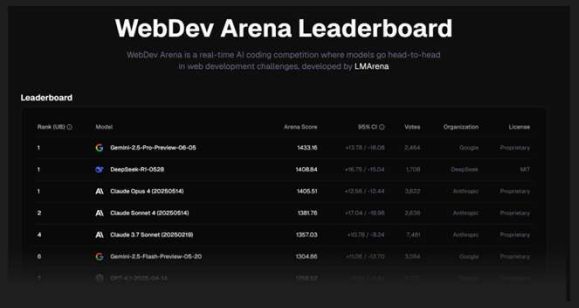
Consider cost of premium models.

https://docs.github.com/copilot/using-github-copilot/ai-models/choosing-the-right-ai-model-for-your-task

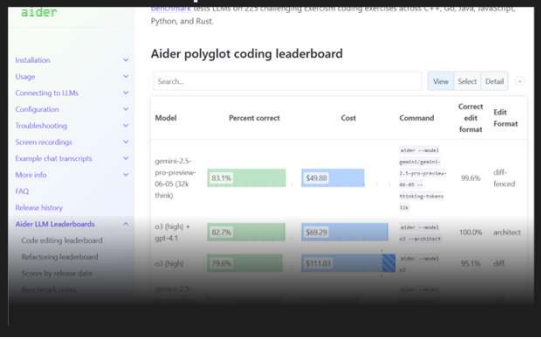
How to pick the right model? 🤔

Consider third party benchmarks:

LMArena:: https://web.lmarena.ai/leaderboard



Aider: https://aider.chat/docs/leaderboards/

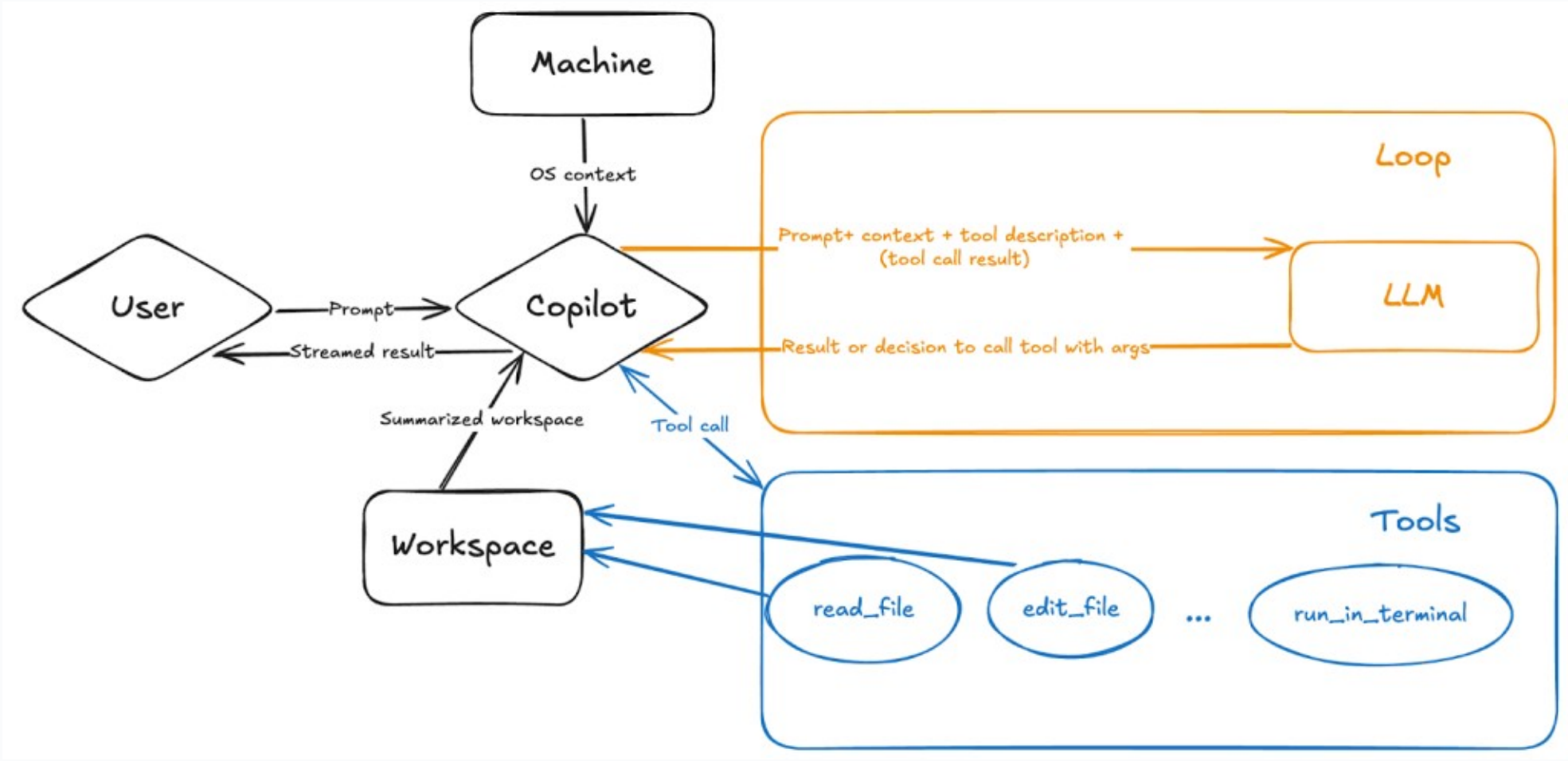


https://docs.github.com/copilot/using-github-copilot/ai-models/choosing-the-right-ai-model-for-your-task

How craft the perfect prompt 🎨

- Be clear and specific in your prompts for better results.
- Give Copilot context by referencing files, code, or errors.
- Use step-by-step instructions for complex tasks.

https://docs.github.com/copilot/using-github-copilot/copilot-chat/prompt-engineering-for-copilot-chat


Machine
OS context
Loop
Prompt+ context + tool description +
(tool call result)
LLM
User
Prompt
Copilot
Streamed result
Result or decision to call tool with args
Summarized workspace
Tool call
Workspace
Tools
read_file
edit_file
...
run_in_terminal


Lab
Creating a mini-game
https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming

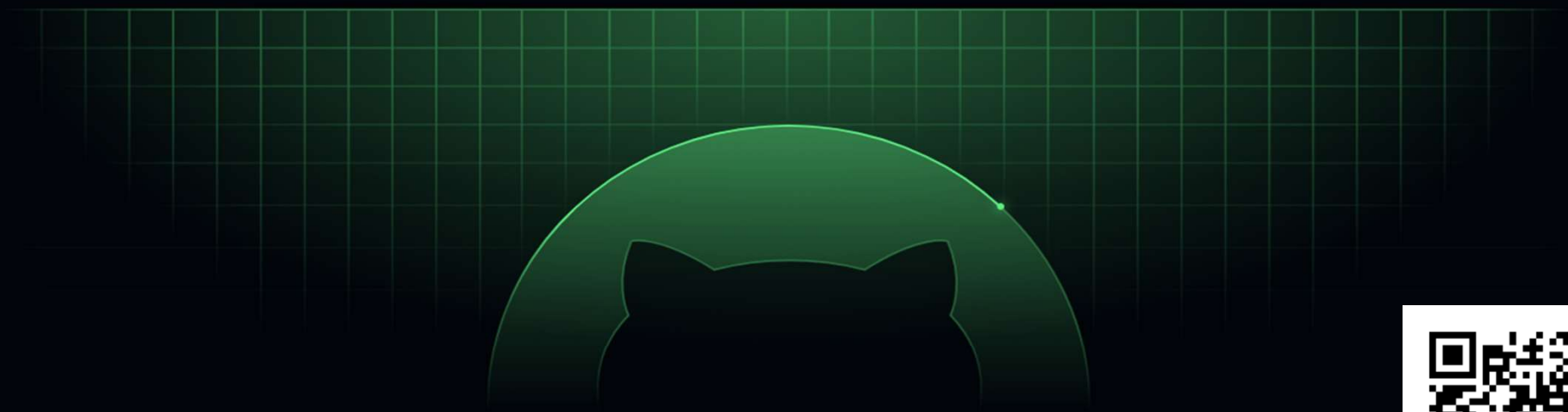

Lab
Using GitHub Copilot
with CSharp
https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming
MR1

Slide 34

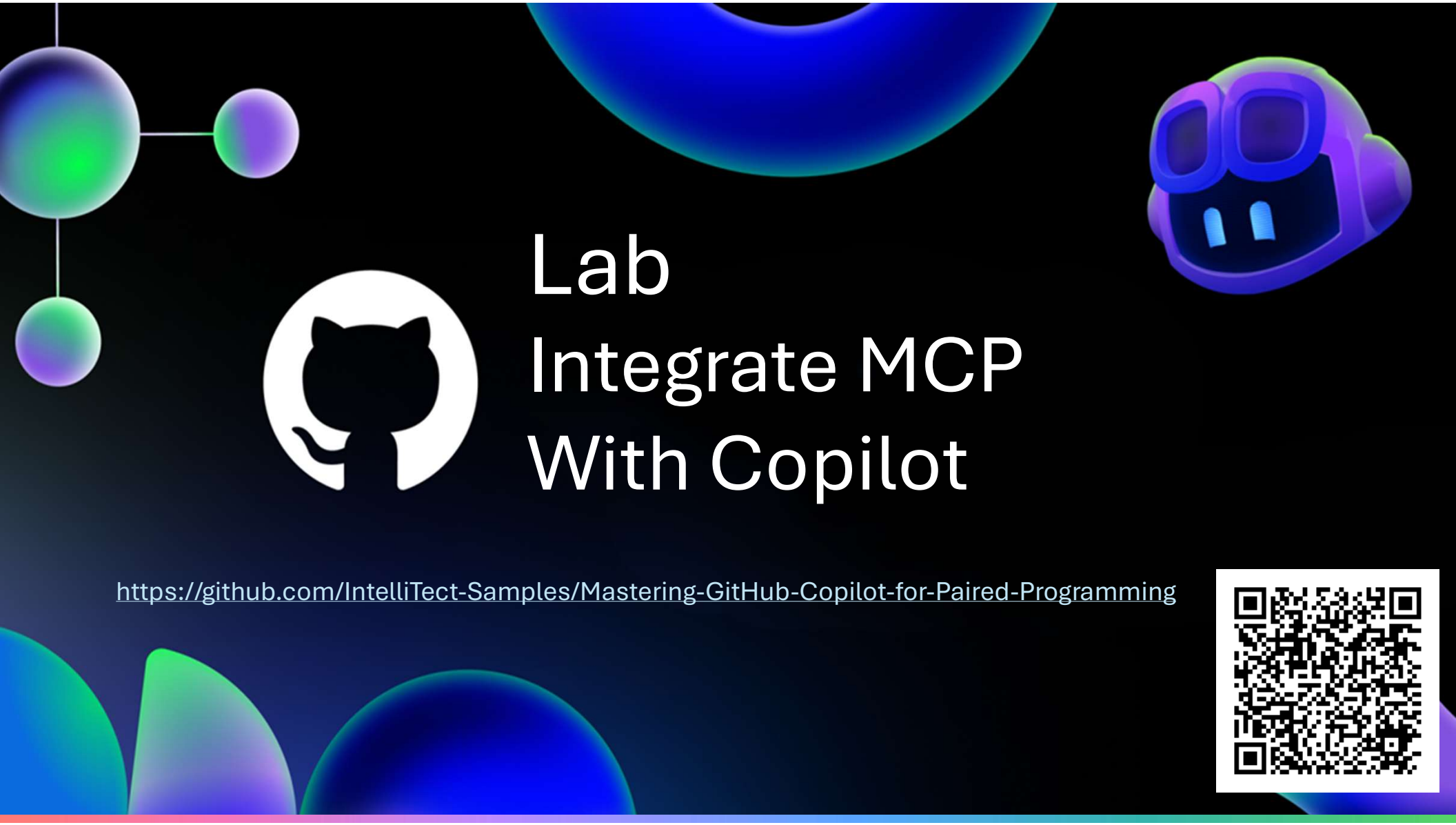
MR1 [@Kevin Bost] did you mean for this to link to https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming/tree/main/Using-GitH
Mitch Rickman, 2025-06-19T03:04:18.763

KB1 0 In this case no. I was refraining from linking directly to the sub folders and instead made all of the links go to the parent repo. It does leave it open for people to hit the wrong folder, but I also think it makes the other exercises more discoverable (and limited the number of links I had to change)
Kevin Bost, 2025-06-19T17:09:56.219

Remote GitHub MCP Server is now in public preview



https://github.com/github/github-mcp-server


Lab
Integrate MCP
With Copilot
https://github.com/IntelliTect-Samples/Mastering-GitHub-Copilot-for-Paired-Programming

GitHub CLI

Take GitHub to the command line

```
$ gh pr status

Relevant pull requests in cli/cli

Current branch
  There is no pull request associated with [

Created by you
  You have no open pull requests

Requesting a code review from you
  #1640  Add auth command and flag usage to
  ✓ Checks passing

$ gh --version
gh version 1.0.0
https://github.com/cli/cli/relea
```

https://cli.github.com/

GitHub CLI

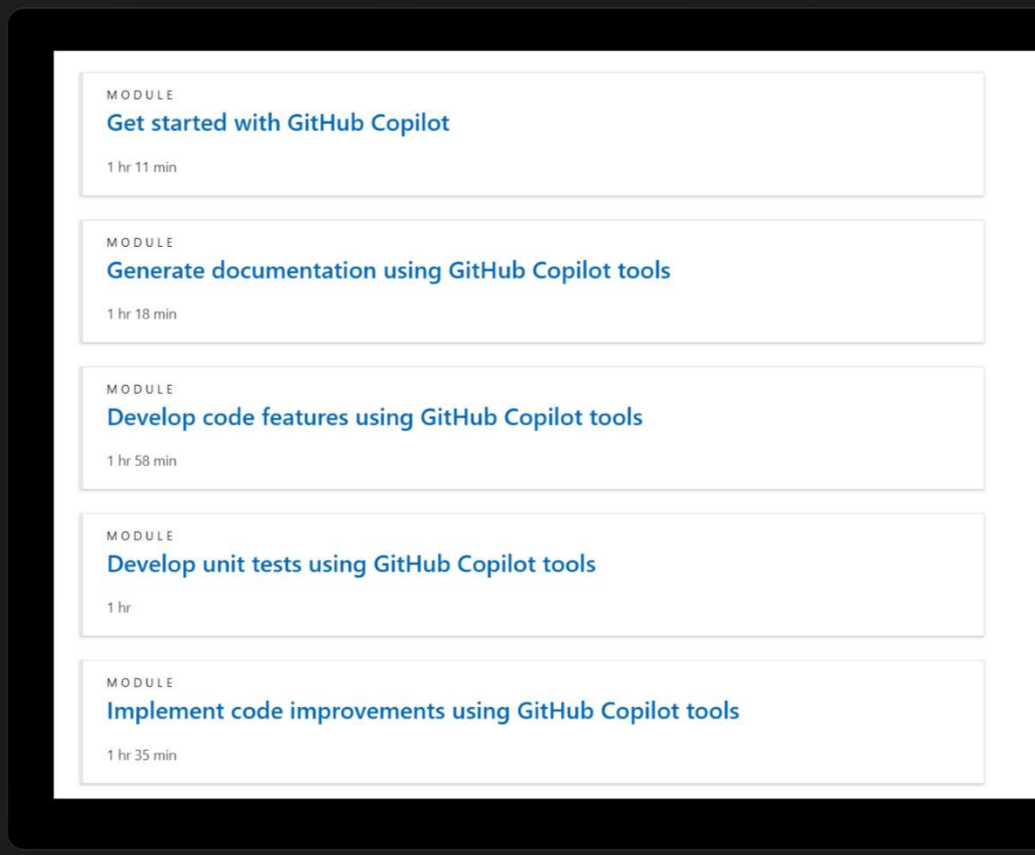
Take GitHub to the command line

```
$ gh pr status

Relevant pull requests in cli/cli

Current branch
  There is no pull request associated with [

Created by you
  You have no open pull requests

Requesting a code review from you
  #1640  Add auth command and flag usage to
  ✓ Checks passing

$ gh --version
gh version 1.0.0
https://github.com/cli/cli/relea
```

https://cli.github.com/

Keep learning!

aka.ms/learn/githubcopilot



Keep learning!

aka.ms/learn/githubcopilot