GH-900: GitHub Foundations

Duration: 2 days (16 hours)

Course description

GH-900: GitHub Foundations provides a comprehensive introduction to Git and GitHub, designed for teams and individuals who want to learn about version control, including software developers, documentation writers, and collaborators on various projects. It begins with an overview of version control systems (VCS), explaining their importance in tracking changes to files, managing project history, and facilitating collaboration through branching and merging. The course emphasizes the differences between Git and GitHub, detailing how Git serves as a distributed version control system while GitHub provides a platform for hosting and managing Git repositories.

Participants will learn the fundamental features of GitHub, including repository management, the GitHub flow, and collaborative features such as issues and pull requests. The course covers essential Git commands like git status, git add, git commit, and git log, providing hands-on exercises to practice these commands. Additionally, it introduces GitHub's collaborative tools, explaining how to create and manage repositories, branches, and pull requests, and how to use GitHub's notification and subscription features to stay updated on project activities.

The course also delves into advanced topics such as GitHub Copilot, an AI-powered code completion tool that assists developers by providing code suggestions and explanations. Participants will learn how to configure and use GitHub Copilot within their development environment, leveraging its capabilities to enhance productivity and code quality. Furthermore, the course explores GitHub Codespaces, a cloudbased development environment that simplifies the setup and management of development environments, allowing participants to code from anywhere with ease.

By the end of the course, participants will have a solid understanding of Git and GitHub, equipped with the skills to manage version control, collaborate effectively on projects, and utilize advanced GitHub features to streamline their development workflows. The course's practical exercises and real-world examples ensure that learners can apply their knowledge immediately, making it a valuable resource for anyone looking to improve their version control and collaboration skills.

The course is designed as a blended learning experience that combines instructor-led training with online materials on the Microsoft Learn platform (https://docs.microsoft.com/learn). Students are encouraged to use the content on Learn as reference materials to reinforce what they learn in class and to explore topics in more depth.

IMPORTANT: This course is designed to be delivered in two full day. The activities are approximately 70% instructional led and 30% student interactive exercises and/or instructor demos.

Learning objectives

After completing this course, students will be able to:

- Understand what version control is and learn about distributed version control systems like Git.

- Recognize the differences between Git and GitHub and understand their roles in the software development lifecycle.
- Identify the fundamental features of GitHub, including repository management and the GitHub flow.
- Explore the collaborative features of GitHub by reviewing issues and discussions.
- Learn how to manage your GitHub notifications and subscriptions.
- Gain an understanding of how to configure and use GitHub Copilot to enhance productivity and code quality

Audience profile

Audience profile for this course is the following:

- Software developers who want to learn about version control, Git, and GitHub.
- Documentation writers who collaborate on various projects and need to manage changes effectively.
- Teams that work on collaborative projects and need to track changes, manage project history, and facilitate collaboration will benefit from this course.
- Individuals who are new to Git and GitHub and want to gain a solid understanding of their fundamental features and collaborative tools are the primary audience.

Audience prerequisites

Candidates should have the following:

- Basic understanding of software development concepts and practices.
- Familiarity with command-line interfaces and basic commands.
- Experience with version control systems, although not necessarily Git.
- Willingness to engage in hands-on exercises and practical applications.
- Experience of working with Microsoft Azure services.

To teach this 1-day course, it's necessary to move quickly through the content so that there is time for demonstrations and student hands-on activities (exercises can be used as instructor demonstrations). It is not necessary to teach each module in-depth and the instructor should decide which topics are the most suitable for the audience.

Learning Path: GitHub Foundations Part 1 of 2

Module 1: Introduction to Git

- Introduction
- What is version control?
- Exercise - Try out Git
- Basic Git commands

Module 2: Introduction to GitHub

Learning Path: GitHub Foundations Part 2 of 2

Module 9: Contribute to an open-source project on GitHub

- Introduction
- Identify where you can help
- Contribute to an open-source repository
- Exercise - Create your first pull request
- Next steps

Module 10: Manage an InnerSource program by using GitHub

- Introduction
- How to manage a successful InnerSource program
- Exercise - InnerSource fundamentals

Module 11: Maintain a secure repository by using GitHub best practices

- Introduction
- How to maintain a secure GitHub repository
- Automated security
- Exercise - Secure your repository's supply chain

Module 12: Introduction to GitHub administration

- Introduction
- What is GitHub administration?
- How does GitHub authentication work?
- How does GitHub organization and permissions work?

Module 13: Authenticate and authorize user identities on GitHub

- Introduction
- User identity and access management
- User authentication
- User authorization
- Team synchronization

Module 14: Manage repository changes by using pull requests on GitHub

- Introduction
- What are pull requests?
- Exercise - Reviewing pull requests

Module 15: Search and organize repository history by using GitHub

- Introduction
- How to search and organize repository history by using GitHub
- Exercise - Connect the dots in a GitHub repository

Module 16: Using GitHub Copilot with Python

- Introduction
- What is GitHub Copilot?
- Exercise - Set up GitHub Copilot to work with Visual Studio Code
- Use GitHub Copilot with Python

- Exercise - Update a Python web API with GitHub Copilot

Exercises and Demos (11 exercises, 4 hours)

Exercises are to be used as hands-on activities for individual students which are led by the instructor, or demonstrations led by the instructor. The decision to lead hands-on activities or perform demonstrations is the instructor's responsibility.

Module 1: Introduction to Git

- Exercise - Try out Git

Module :2 Introduction to GitHub

- Exercise - A guided tour of GitHub

Module 5: Introduction to GitHub Copilot

- Exercise - Develop with AI-powered code suggestions by using GitHub Copilot and VS Code

Module 6: Code with GitHub Codespaces

- Exercise - Code with Codespaces and Visual Studio Code

Module 8: Communicate effectively on GitHub using Markdown

- Exercise - Communicate using Markdown

Module 9: Contribute to an open-source project on GitHub

- Exercise - Create your first pull request

Module 10: Manage an InnerSource program by using GitHub

- Exercise - InnerSource fundamentals

Module 11: Maintain a secure repository by using GitHub best practices

- Exercise - Secure your repository's supply chain

Module 14: Manage repository changes by using pull requests on GitHub

- Exercise - Reviewing pull requests

Module 15: Search and organize repository history by using GitHub

- Exercise - Connect the dots in a GitHub repository

Module 16: Using GitHub Copilot with Python

- Exercise - Update a Python web API with GitHub Copilot

Course timing

The following agenda provides estimated times to complete each classroom activity. However, the estimated times may vary depending on the background of your students, which may affect whether you can move faster or slower through the course material.

Estimated times for each Module include the time to complete as part of a 2-day course:

- The module’s PowerPoint slide deck presentation (70% of course timing)
- Exercises (30% of course timing)
- Optional: Pre-defined product demonstrations, determined by instructor
- Optional: Knowledge Check questions (see the section on Additional Timing Notes below)

You should adjust the agenda accordingly based on any classroom activities that you personally created or plan to deliver that are not included in the slides for this course. For example, if you plan to present:

- ad-hoc demonstrations
- review activities
- classroom games
- and so on…

Note: Each module activity for the following agenda is the slide deck presentation for that module.

The following table provides delivery specifications for the instructor-led portion of the 2-day course.

PowerPoint slides	Presentation (minutes)
Module 1: Introduction to Git	**25**
Module 2: Introduction to GitHub	**30**
Module 3: Introduction to GitHub's products	**30**
Module 4: Configure code scanning on GitHub	**35**
Module 5: Introduction to GitHub Copilot	**15**
Module 6: Code with GitHub Codespaces	**45**
Module 7: Manage your work with GitHub Projects	**20**
Module 8: Communicate effectively on GitHub using Markdown	**55**
Module 9: Contribute to an open-source project on GitHub	**20**
Module 10: Manage an InnerSource program by using GitHub	**45**
Module 11: Maintain a secure repository by using GitHub best practices	**35**
Module 12: Introduction to GitHub administration	**20**
Module 13: Authenticate and authorize user identities on GitHub	**25**
Module 14: Manage repository changes by using pull requests on GitHub	**40**
Module 15: Search and organize repository history by using GitHub	**35**
Module 16: Using GitHub Copilot with Python	**25**
Total	**470 mins**

Exercises

IMPORTANT: The exercise activities in this technical workshop are and require intermediate level GitHub knowledge and scripting experience.

The following table provides delivery specifications for the interactive exercise portion of the 2-day course.

Exercises and/or demonstrations	Estimate (minutes)
Module 1: Introduction to Git • Exercise - Try out Git	**10**
Module :2 Introduction to GitHub • Exercise - A guided tour of GitHub	**45**

Module 5: Introduction to GitHub Copilot • **Exercise - Develop with AI-powered code suggestions by using GitHub Copilot and VS Code**	**20**
Module 6: Code with GitHub Codespaces • **Exercise - Code with Codespaces and Visual Studio Code**	**15**
Module 8: Communicate effectively on GitHub using Markdown • **Exercise - Communicate using Markdown**	**15**
Module 9: Contribute to an open-source project on GitHub • **Exercise - Create your first pull request**	**15**
Module 10: Manage an InnerSource program by using GitHub • **Exercise - InnerSource fundamentals**	**20**
Module 11: Maintain a secure repository by using GitHub best practices • **Exercise - Secure your repository's supply chain**	**15**
Module 14: Manage repository changes by using pull requests on GitHub • **Exercise - Reviewing pull requests**	**15**
Module 15: Search and organize repository history by using GitHub • **Exercise - Connect the dots in a GitHub repository**	**20**
Module 16: Using GitHub Copilot with Python • **Exercise - Update a Python web API with GitHub Copilot**	**20**
Total	**200 mins**

At the time the courses were released, the exercise instruction had been tested and the exercise steps were accurate. However, given the nature of Microsoft's cloud products and the fact that Microsoft releases UI updates on a regular basis, it's possible that at some point in time, the UI for a given feature may change so that it no longer matches the exercise instruction.

If students encounter exercise steps that don't accurately reflect the UI, they'll have to work through the UI to determine what needs to be done. Typically, UI changes are quite subtle, so hopefully you don't find yourself in a situation where a feature was completely overhauled.

However, if you do run into major UI changes, challenge your students to work through it, and only offer help if they need it. Product UI changes will be part of their daily life in today's cloud-centric world. As IT/Pros, they must learn how to work through such situations.

Portal, Cloud Shell, PowerShell, and the CLI (when necessary for the courseware)

The exercise instructions are written to use the Cloud Shell. The Cloud Shell automatically connects to Azure and provides access to PowerShell and the CLI.

If you would rather have students use PowerShell or the CLI locally, you can use these links.

- Install Azure PowerShell on Windows with PowerShellGet
- Install Azure CLI on Windows

Azure subscriptions (when necessary for the courseware)
To complete the exercises and any additional demonstration exercises in this course, students will need an Azure Subscription.

The Azure pass effectively functions in the same way as the publicly available Microsoft Azure Trial Subscription. This means there are limitations on what you can do with the pass.

Tips and tricks for teaching GH-900

The GH-900 course contains instruction and content that makes use of GitHub and Azure, which is subject to continual updates. Review the exercises prior to teaching the course to determine if any of the resources referenced in the course have changed in name or location. Every effort has been made to not tie the resources to locations in the service, but it is not possible to foresee all possible changes that may take place in cloud services such as Dependabot or Secret scanning.

Course timing

Typical comments:

The course timing is too tight, the day ran too long, the pace is too fast, the course should be an additional day, and not enough time to absorb the material.

Discussion:

GH-900 provides a lot of content for you to select based on your audience and teaching style. **It is not intended** that you cover every topic in the source content on Microsoft Learn in detail. Remember that the modules on Learn provide students with an opportunity for further study outside of the constraints of a 2-day class.

You must select the combination of course elements that fit best within the allotted classroom time. Consider your audience, consider your ability to tell the story, and consider the depth of coverage.