**Introduction to GitHub by Dongmo Dassi Raïssa**

**Sources:** [**https://code.visualstudio.com/docs**](https://code.visualstudio.com/docs)

**Sources :** [**https://www.w3schools.com/**](https://www.w3schools.com/)

GitHub is a web-based platform that leverages Git, a version control system, to provide a collaborative environment for software development. It hosts source code, manages version control, and facilitates collaboration among developers.

**Primary Functions and Features:**

* **Repository Hosting**: Store and manage code repositories.
* **Version Control**: Track changes to code over time.
* **Collaboration**: Enable multiple developers to work on the same project.
* **Pull Requests**: Propose changes to codebases and discuss them before integration.
* **Issues and Project Management**: Track bugs, tasks, and project progress.
* **GitHub Actions**: Automate workflows, such as CI/CD pipelines.
* **Code Review**: Facilitate peer reviews of code changes.

**Repositories on GitHub**

**What is a GitHub Repository?**

A GitHub repository is a storage space where a project's files and their revision history are kept. It allows developers to collaborate on code and track changes over time.

**Creating a New Repository:**

1. **Sign in to GitHub**: Go to [GitHub](https://github.com/) and sign in.
2. **Create a New Repository**:
   * Click on the "+" icon in the top-right corner and select "New repository."
   * Fill in the repository name, description (optional), and choose visibility (public or private).
   * Initialize with a README (optional) and add a .gitignore and license if needed.
3. **Essential Elements**:
   * **README.md**: Provides an overview and documentation of the project.
   * **.gitignore**: Specifies which files and directories to ignore.
   * **LICENSE**: Defines the project's licensing terms.
   * **src/**: Directory containing source code.
   * **docs/**: Documentation files.

**Version Control with Git**

**Concept of Version Control:**

Version control systems (VCS) manage changes to source code over time. Git, a distributed VCS, allows multiple developers to work on a project simultaneously without overwriting each other's changes.

**GitHub and Version Control:**

GitHub enhances version control by providing a remote repository, enabling collaboration, and offering features like pull requests, issue tracking, and project management.

**Branching and Merging in GitHub**

**What are Branches?**

Branches are independent lines of development within a repository. They allow developers to work on features or fixes separately from the main codebase.

**Creating and Merging Branches:**

1. **Create a Branch**:

***git checkout -b feature-branch***

1. **Make Changes**: Modify code and commit changes to the new branch.
2. **Merge Branch**:
   * Push the branch to GitHub.
   * Open a pull request to merge changes into the main branch.
   * Review and merge the pull request.

**Pull Requests and Code Reviews**

**What is a Pull Request?**

A pull request (PR) is a method to propose changes to a codebase. It allows team members to review, discuss, and approve changes before merging them.

**Steps to Create and Review a Pull Request:**

1. **Create a Pull Request**:
   * Push your branch to GitHub.
   * Navigate to the repository and click "New pull request."
   * Select the branch with your changes and the base branch (e.g., main).
   * Add a title and description.
2. **Review a Pull Request**:
   * Review code changes, add comments, and request changes if needed.
   * Approve and merge the PR if it meets the requirements.

**GitHub Actions**

**What are GitHub Actions?**

GitHub Actions automate workflows directly within GitHub. They can be used for continuous integration (CI), continuous deployment (CD), and other automated tasks.

**Example of a CI/CD Pipeline:**

1. **Create a Workflow File**: .github/workflows/ci.yml

**Introduction to Visual Studio**

**What is Visual Studio?**

Visual Studio is an integrated development environment (IDE) by Microsoft. It supports a wide range of programming languages and tools for software development.

**Key Features:**

* Advanced debugging and profiling tools.
* IntelliSense code completion.
* Integrated Git support.
* Support for multiple programming languages.
* Extensive library of extensions.

**Difference from Visual Studio Code:**

Visual Studio Code (VS Code) is a lightweight, open-source code editor, while Visual Studio is a full-featured IDE with comprehensive tools for large-scale software development.

**Integrating GitHub with Visual Studio**

**Steps to Integrate GitHub:**

1. **Install GitHub Extension**: Ensure the GitHub extension is installed in Visual Studio.
2. **Clone Repository**:
   * Open Visual Studio and go to "File" > "Clone Repository."
   * Enter the repository URL and clone it.
3. **Manage Changes**: Use Visual Studio's Git tools to commit, push, pull, and create branches.

**Enhancements to Development Workflow:**

* Seamless integration with GitHub for version control.
* Easy access to pull requests, issues, and code reviews.
* Enhanced collaboration features.

**Debugging in Visual Studio**

**Debugging Tools:**

* **Breakpoints**: Pause code execution at specific points.
* **Watch Windows**: Monitor variables and expressions.
* **Call Stack**: View the sequence of function calls.
* **Immediate Window**: Execute code and inspect variables.

**Using Debugging Tools:**

* Set breakpoints to pause execution.
* Step through code to understand flow and logic.
* Inspect variables and modify values during debugging.

**Collaborative Development using GitHub and Visual Studio**

**Using GitHub and Visual Studio Together:**

GitHub and Visual Studio together provide a powerful environment for collaborative development. GitHub handles version control and collaboration, while Visual Studio offers advanced coding and debugging tools.

**Real-World Example:**

A team developing a web application:

* **Developers** use branches in GitHub to work on features.
* **Pull Requests** are created for code review.
* **GitHub Actions** automate testing and deployment.
* **Visual Studio** is used for coding, debugging, and integrating with GitHub for seamless version control.