Editors and Integrated Development Environments

This lesson lists some IDE(s) that provide support for Go and show how the configuration of environments provides maximum ease.

WE'LL COVER THE FOLLOWING

IDE(s) providing Go support

Various development functionalities exist for editors ranging from Vim and Emacs, text editors like BBEdit, Brackets, Gedit, Kate, Komodo, TextMate, TextPad, JEdit, SublimeText, Atom and Visual Studio Code. Some more IDE-like cross-platform environments exist for Go-programming (some are plugins for existing (Java) environments):

- **GoClipse** customizes the **Eclipse** IDE for Go development (click here).
- **GoWorks** is an open-source Go IDE based on **Netbeans** (click here).
- **GoLand** is an IDE made by JetBrains (click here).
- **LiteIDE** is a simple, open-source, cross-platform Go IDE (click here).

In most IDE's you can configure that building also saves and formats the latest changes to the source file. As an example, here's a more detailed discussion of the free Visual Studio Code plugin.

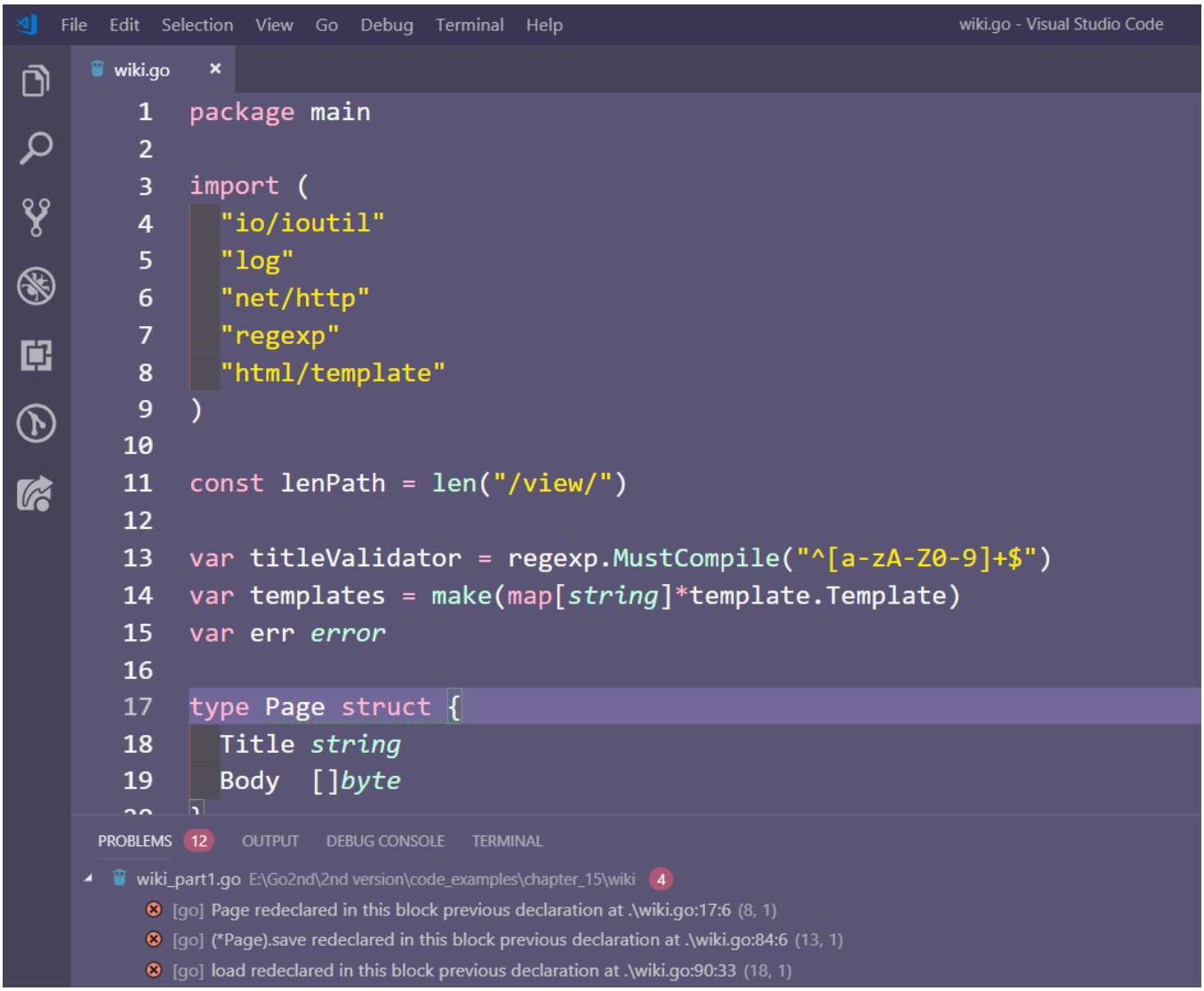
Visual Studio Code with the `vscode-go` plugin

This plugin provides rich development support, including:

- auto-completion and IntelliSense
- code navigation and definition look-up facilities
- code snippets

- formatting, renaming symbols, generating methods and structs
- diagnostic linter and error reporting at save time and even while you type code
- integrated testing, benchmarking and debugging
- installing go tools
- uploading code to Playground

Much of this support is implemented as specific Go commands in the Command Palette. Here is a screenshot using the plugin:



VSCode Go Plugin

Several cloud-based IDEs are also available, such as GitPod and Wide.

Editors and different IDE(s) provide different tools and facilities to make sure that a programmer feels at home when writing code. Another major functionality provided by IDE(s) is finding errors in the code. Let's look at this

in the next lesson!