

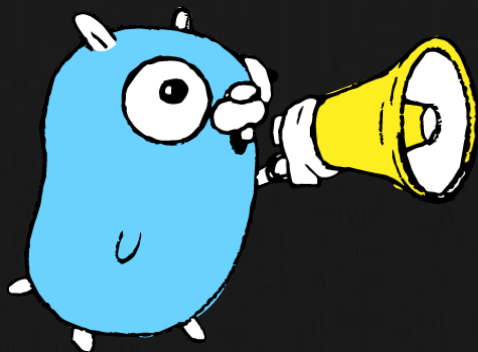


Go programming course

Lecture 1

Introduction to Go





Today's (glorious) blather.

Introduction to Go	01
Written popular services	02
Why Go?	03
Installing compiler and IDE	04
First program	05



SECTION ONE

Introduction to Go

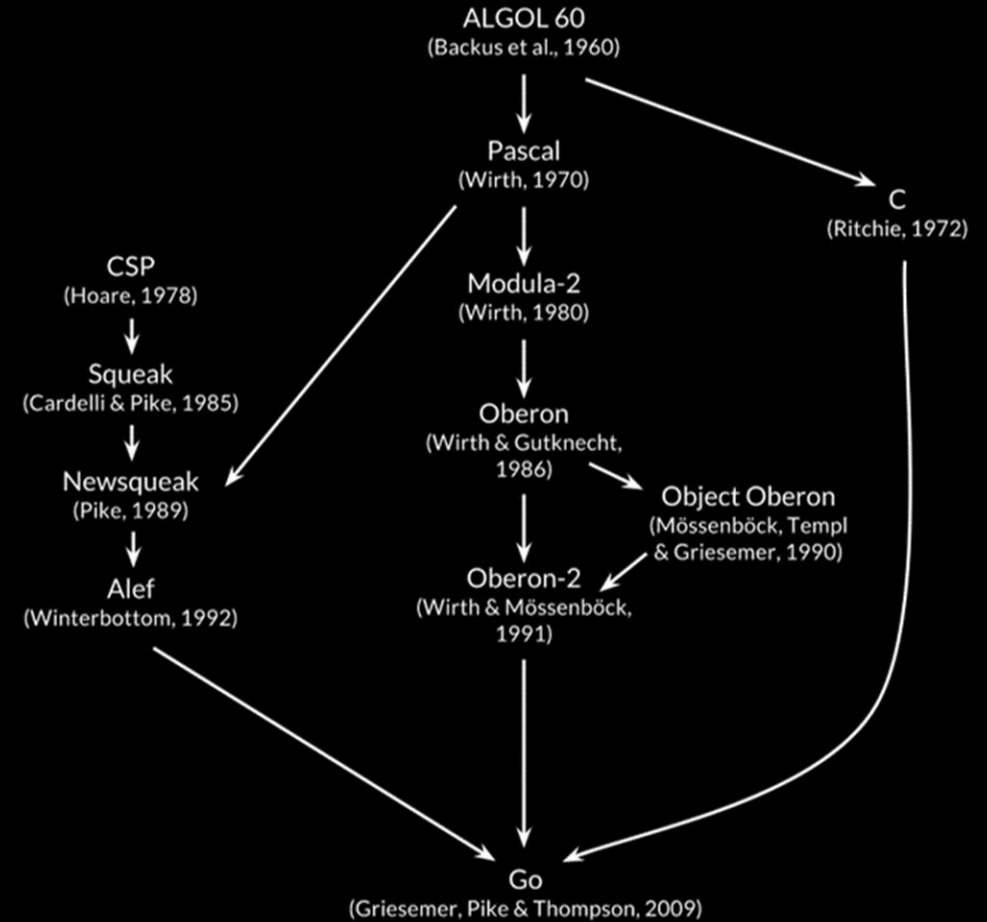
- Go is a statically typed, compiled programming language designed at Google by Robert Griesemer, Rob Pike, and Ken Thompson. It is syntactically similar to C, but with memory safety, garbage collection, structural typing, and CSP-style concurrency
- Was designed at Google in 2007
- First appeared on 10 Nov. 2009
- Latest stable release: 1.20 / 1 February 2023

[https://en.wikipedia.org/wiki/Go_\(programming_language\)](https://en.wikipedia.org/wiki/Go_(programming_language))

- Golang is one of the youngest programming languages. It was created to meet the needs of programmers working on large projects.
- Key take-away: Go was meant to be easy to use and perfectly suited to creating large programming systems. Because of that, many programmers working together could efficiently debug complex projects.

THE ORIGINS

- From Modula-2:
 - package concept
- From Oberon-2:
 - syntax for packages
 - Imports
 - declarations
- From Object Oberon:
 - syntax for method declarations
- From Alef:
 - CSP
- From APL:
 - iota
- From Scheme:
 - lexical scope with nested functions
- New:
 - slices
 - defer



Donovan, Kernighan - The Go Programming Language

SECTION TWO

Written popular services

USE CASES



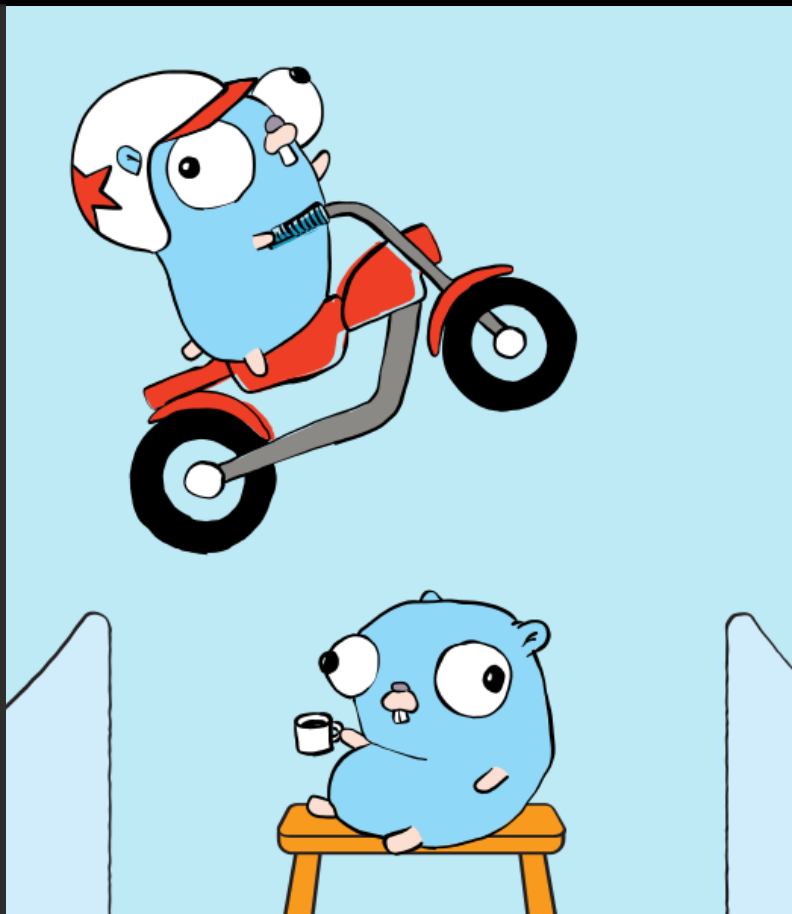
Cloud & Network Services

With a strong ecosystem of tools and APIs on major cloud providers, it is easier than ever to build services with Go.



Web Development

With enhanced memory performance and support for several IDEs, Go powers fast and scalable web applications.



Command-line Interfaces

With popular open source packages and a robust standard library, use Go to create fast and elegant CLIs.



DevOps & Site Reliability

With fast build times, lean syntax, an automatic formatter and doc generator, Go is built to support both DevOps and SRE.

<https://go.dev>

COMPANIES USING GO



<https://github.com/golang/go/wiki/GoUsers>

SECTION THREE

Why Go?

“

Go's purpose is to
make its designers'
programming lives better.

”

ROB PIKE



REASONS TO CODE IN GO



- Simple code
- Created for large projects
- Easy to learn
- One problem – one solution
- Ease of maintenance
- Similar to C

<https://brainhub.eu/library/companies-using-golang>

REASONS TO CODE IN GO



- Designed for multi-core processors
- Designed for the Internet
- Quick compilation
- Small application size
- Open Source model

<https://brainhub.eu/library/companies-using-golang>

SECTION FOUR

Installing compiler and IDE

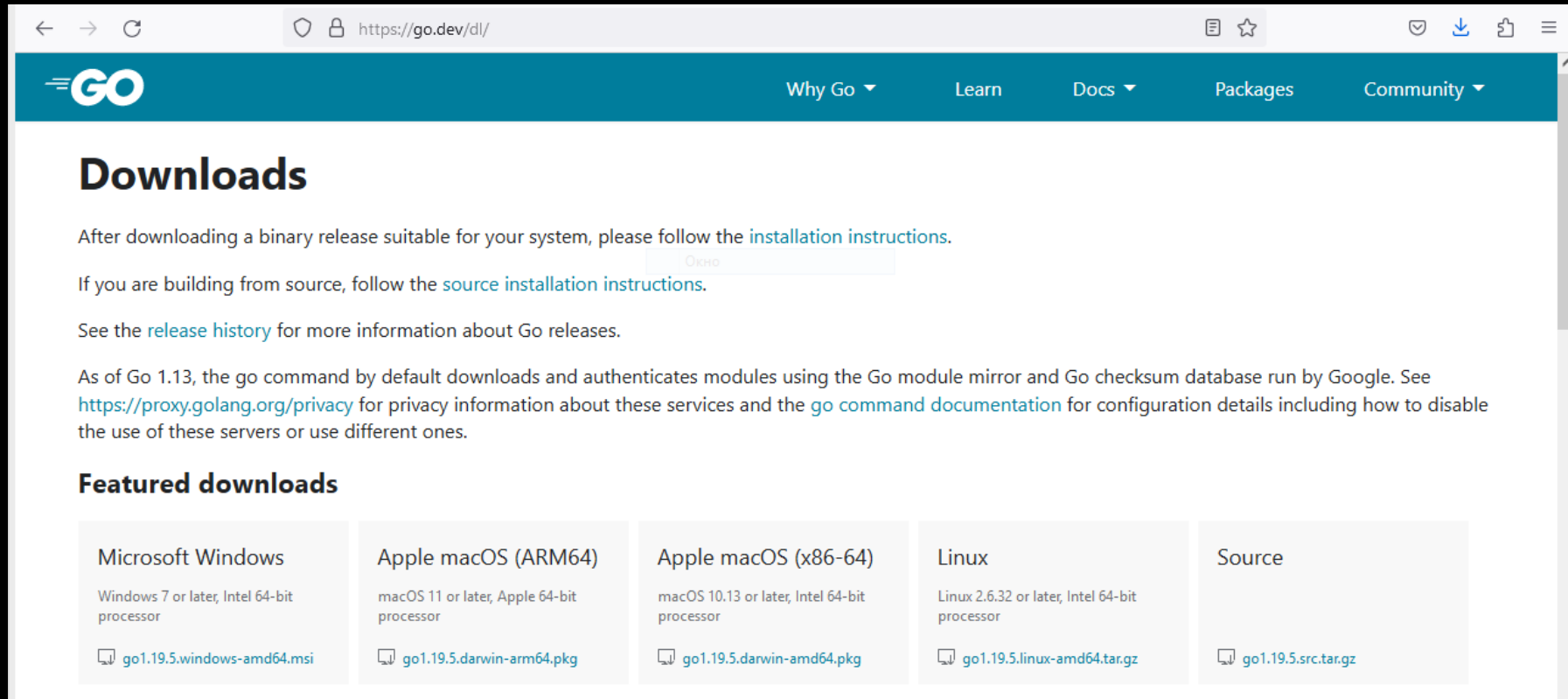
DOWNLOADING AND INSTALLING TOOLS



To get started with Go, we need any text editor to write code and a Go compiler to compile our code to binary executable file and then run it.

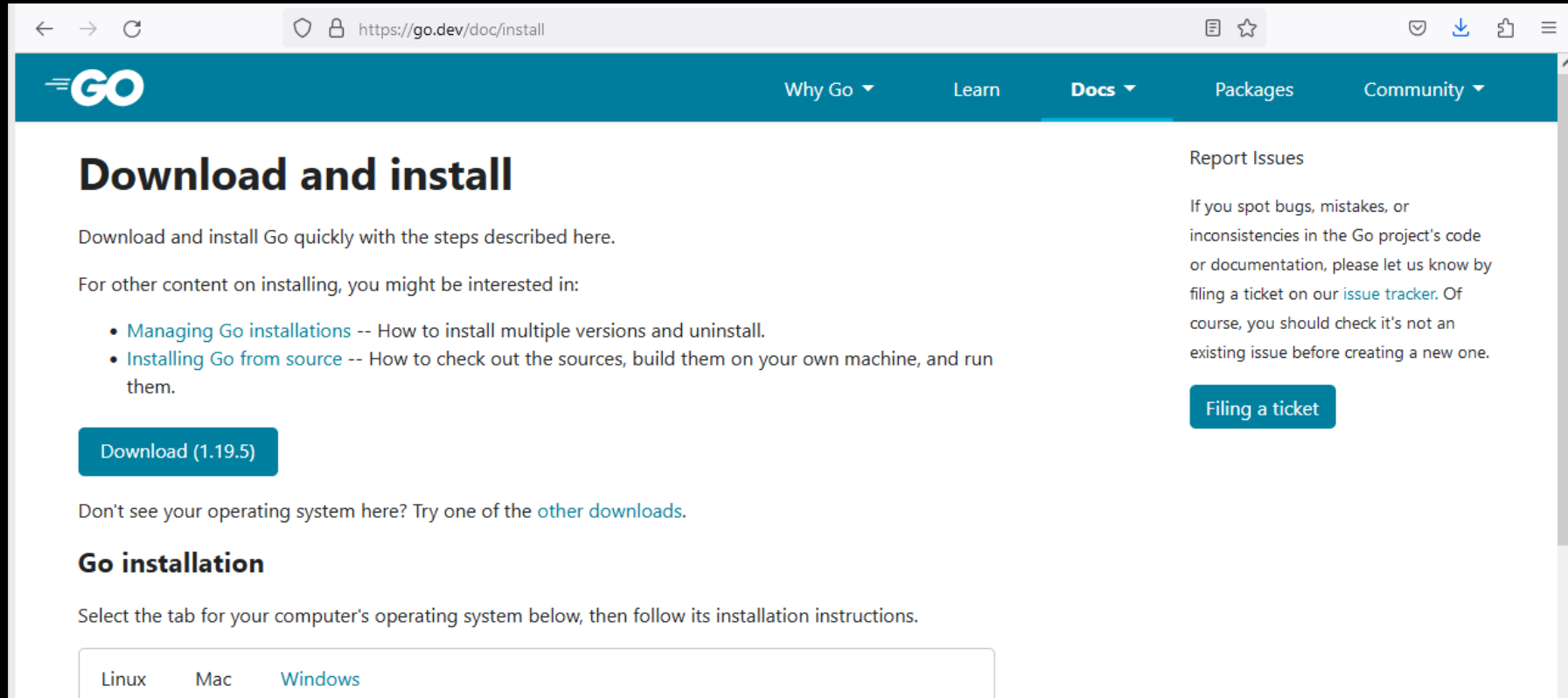
Instead of any editor, we recommend using an IDE - **Integrated Development Environment**, that will help you code more productively.

Go ahead and download the Go installer from official website



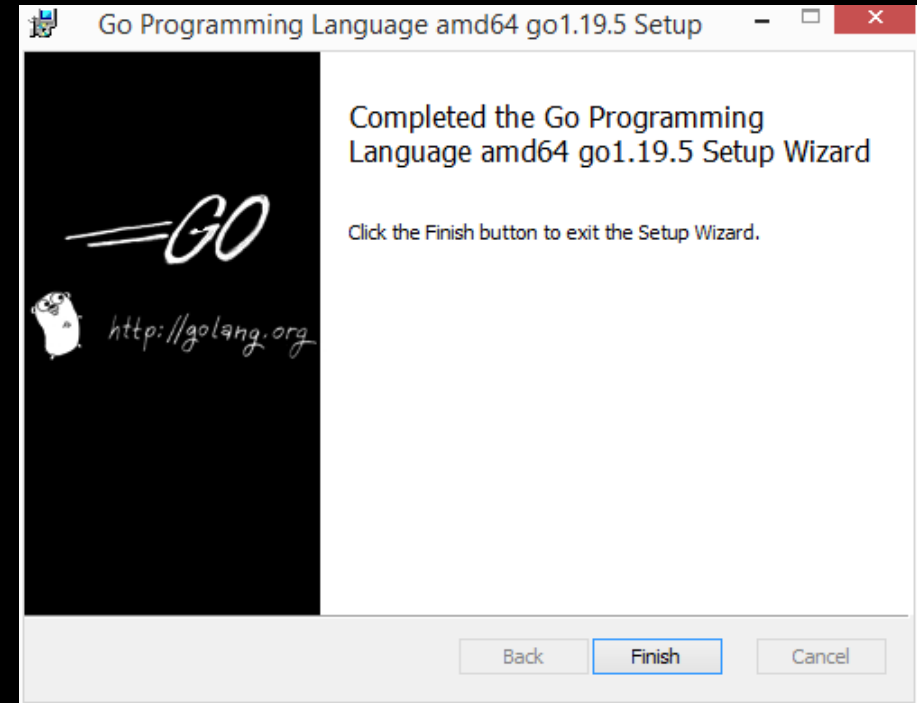
<https://go.dev/dl>

You can go to instructions page to get info about further steps



<https://go.dev/doc/install>

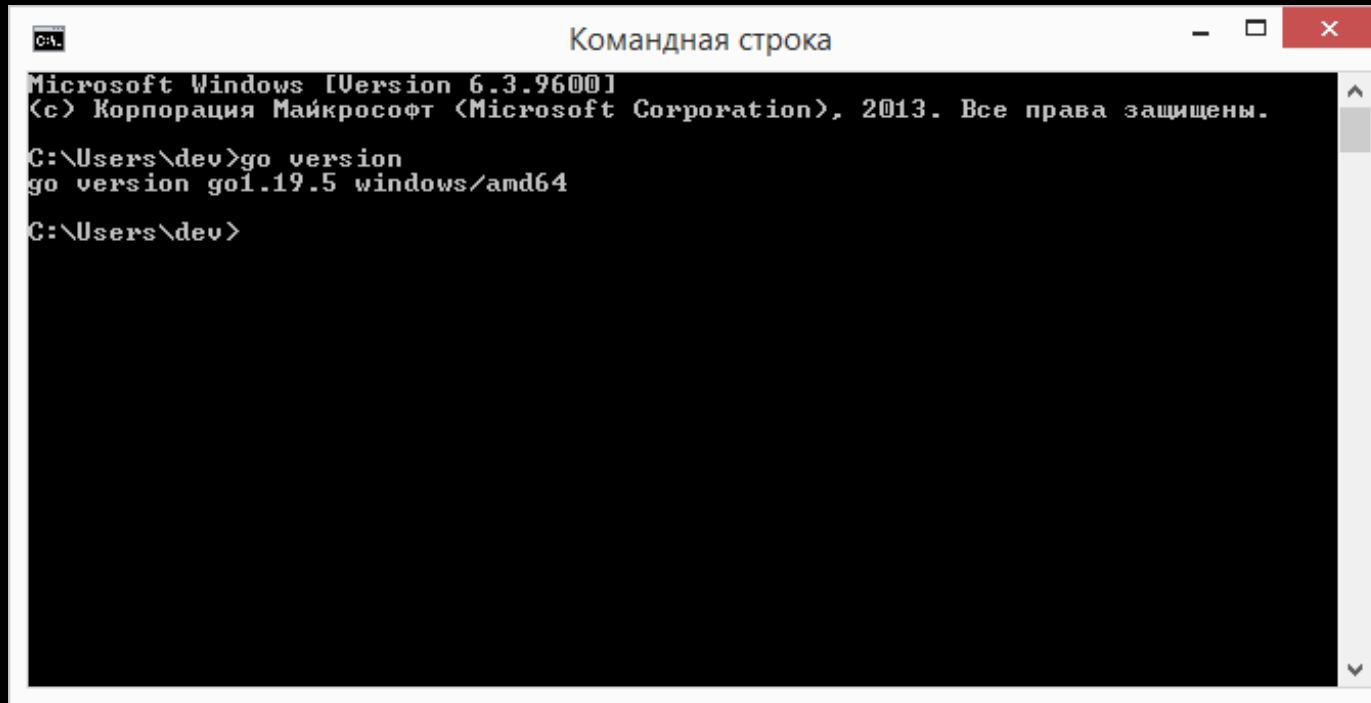
For Windows: run the MSI Installer that you have downloaded, follow the instructions and install Go



COMPILER



To verify that you've installed Go successfully, open command prompt, type the following command, and check the output:

A screenshot of a Windows Command Prompt window. The title bar is labeled 'Командная строка' (Command Prompt). The window shows the following text: 'Microsoft Windows [Version 6.3.9600] (c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.' followed by the command 'C:\Users\dev>go version' and its output 'go version go1.19.5 windows/amd64'. The prompt 'C:\Users\dev>' is shown again on the next line.

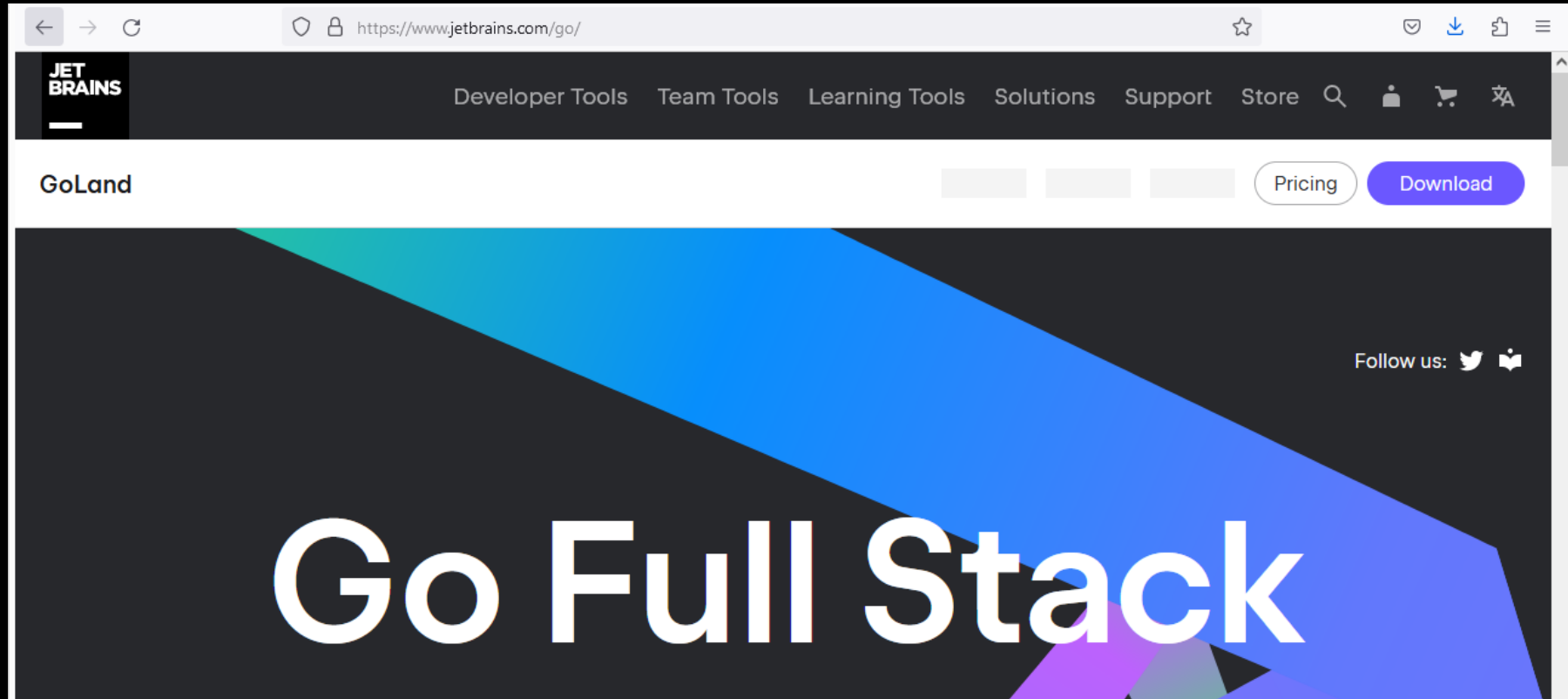
```
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.

C:\Users\dev>go version
go version go1.19.5 windows/amd64

C:\Users\dev>
```

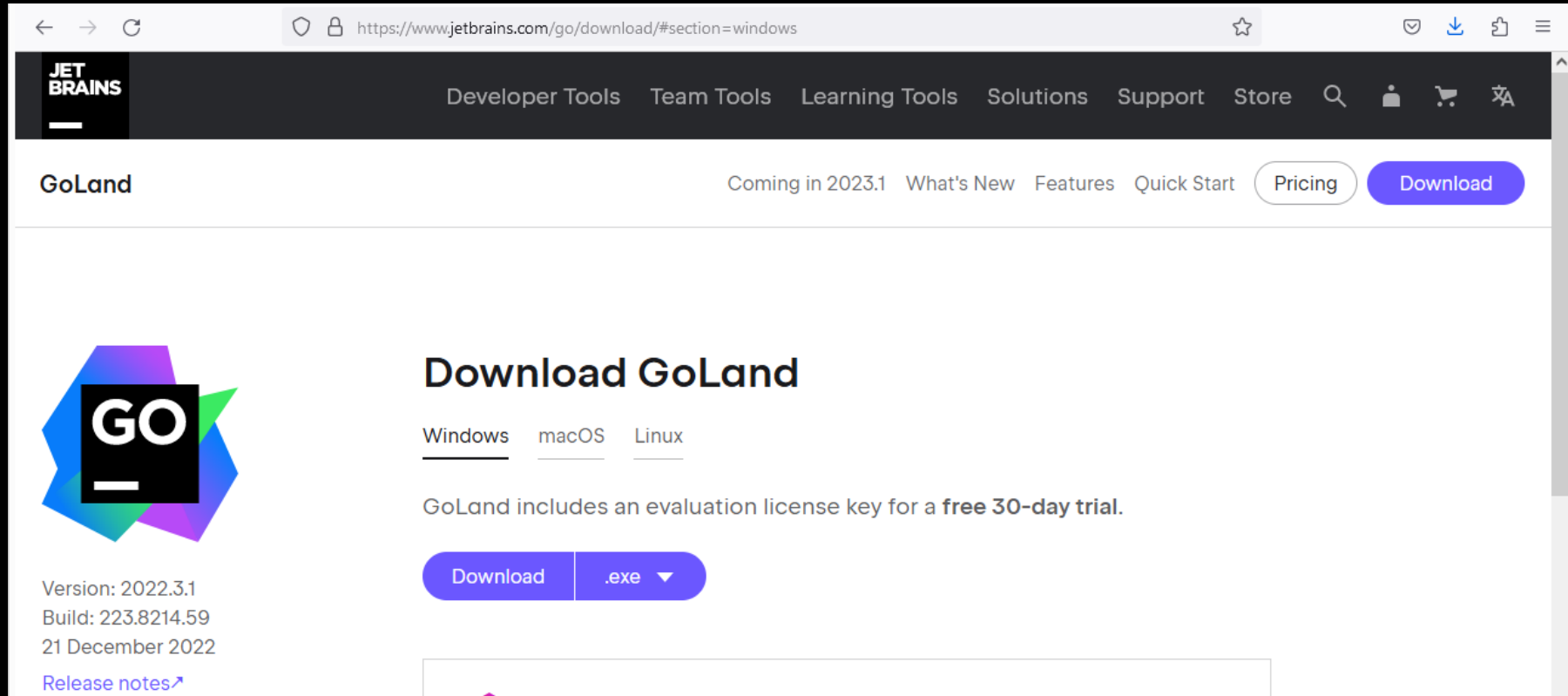
Start menu > Search for “cmd” > Press Enter > type “go version” > Press Enter

In this course, we will be using the GoLand IDE from JetBrains



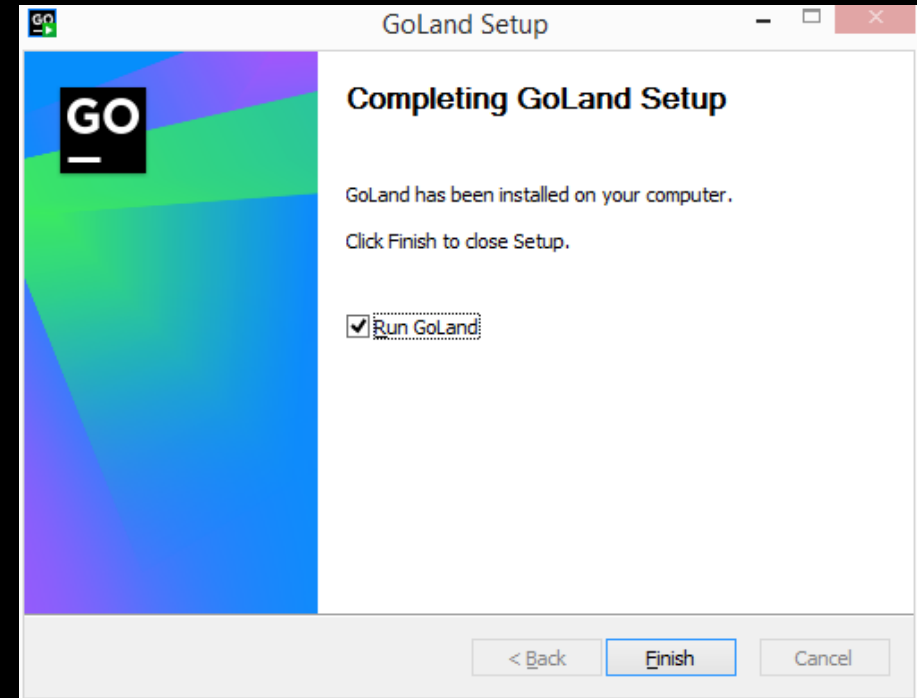
<https://www.jetbrains.com/go/>

Go to download page and choose your platform



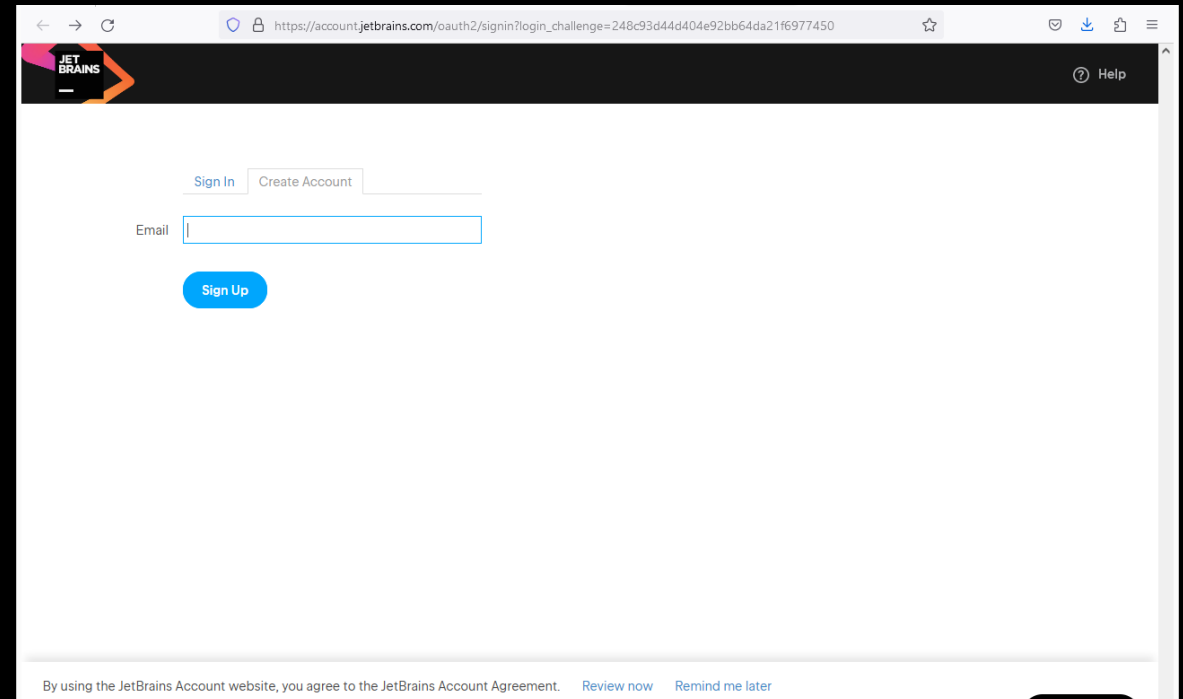
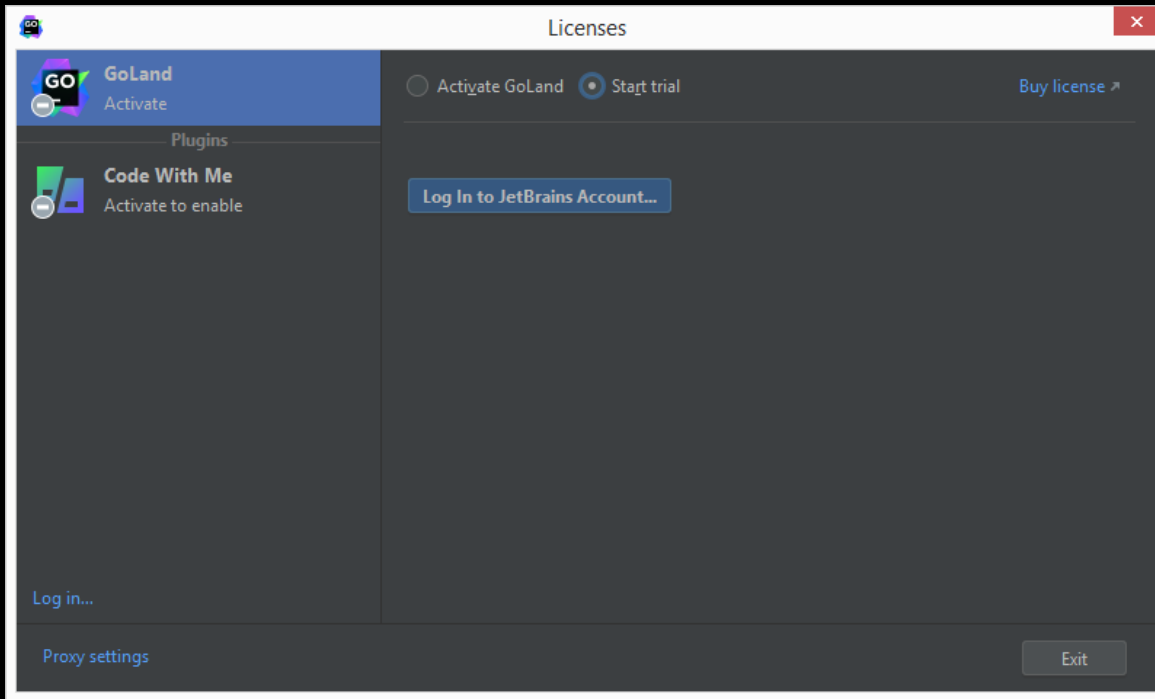
<https://www.jetbrains.com/go/download>

For Windows: run the exe file that you have downloaded, follow the instructions and install GoLand IDE



IDE LICENSE ACTIVATION

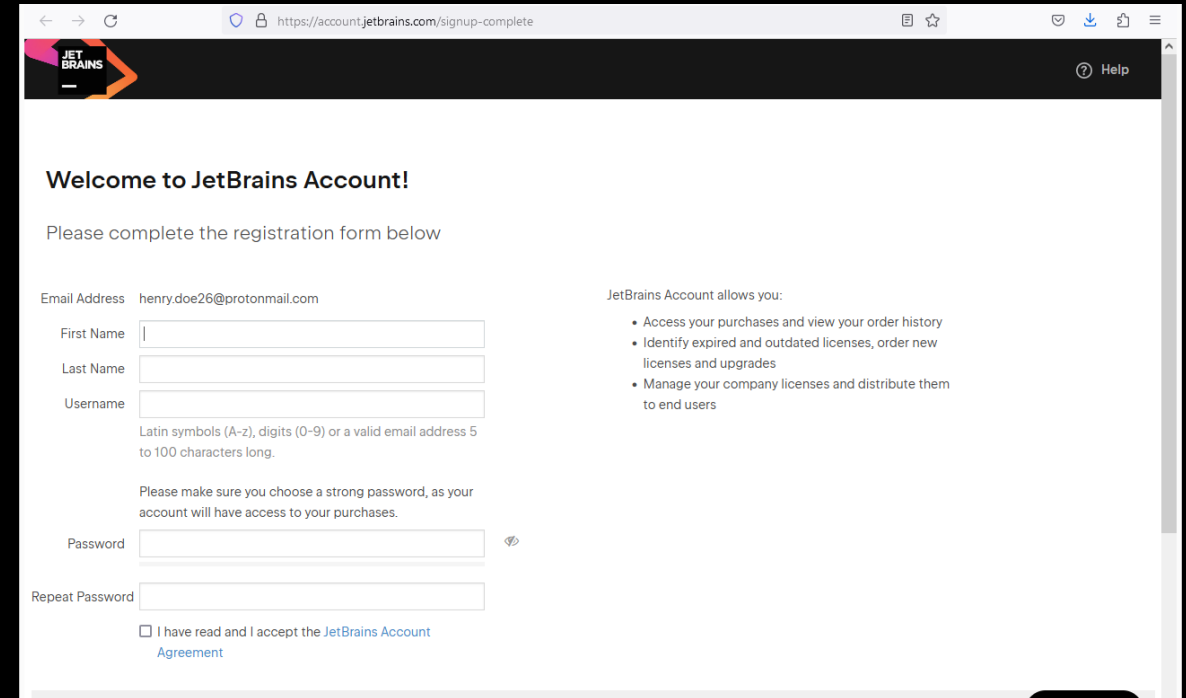
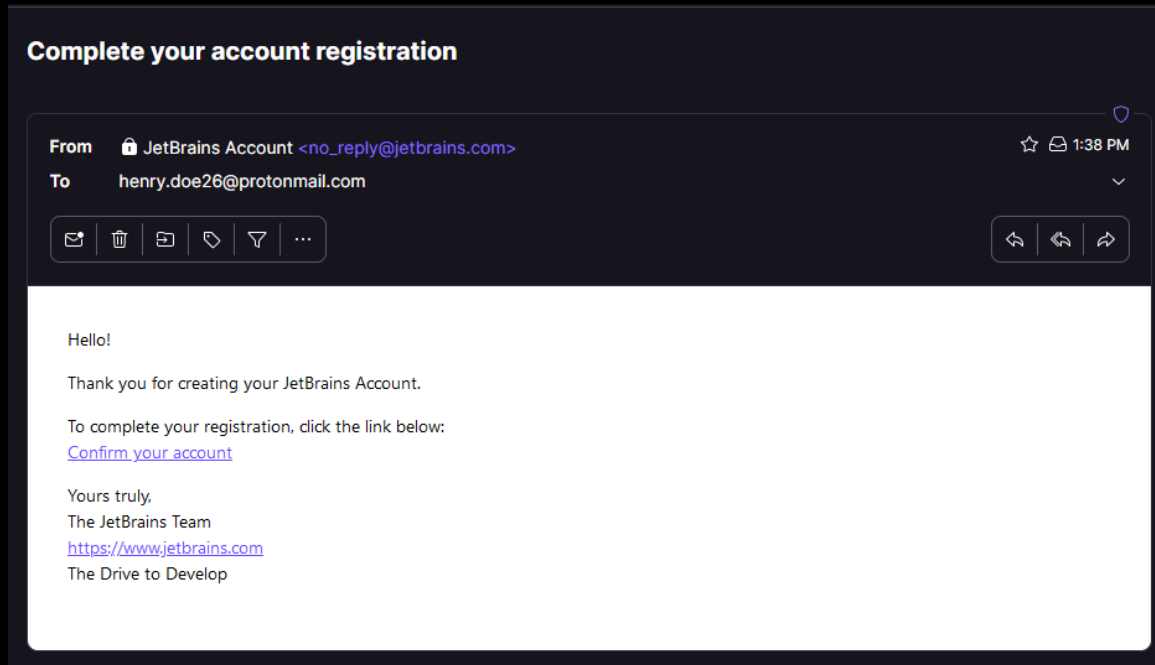
JetBrains offers a 30-day free trial license. But you have to have an account to activate it. Choose trial mode and click the Log in button



It will open your browser at JetBrains account sign in page. If you don't have an account – create one. Enter your email address and click the Sign Up button

IDE LICENSE ACTIVATION

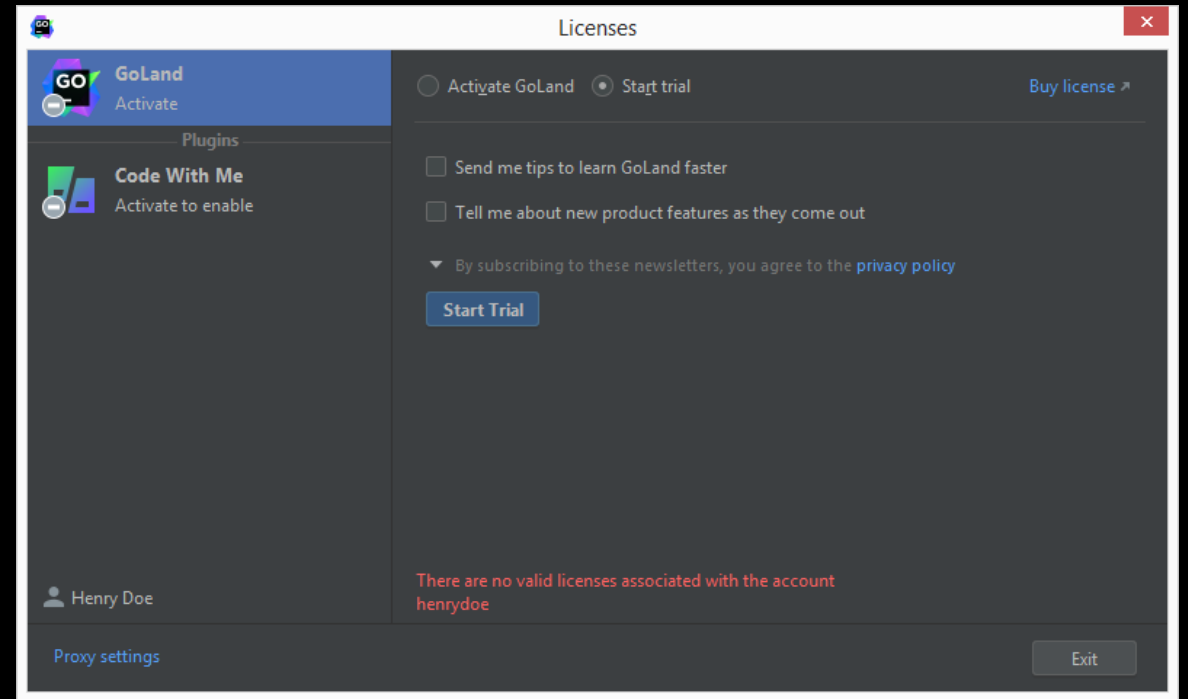
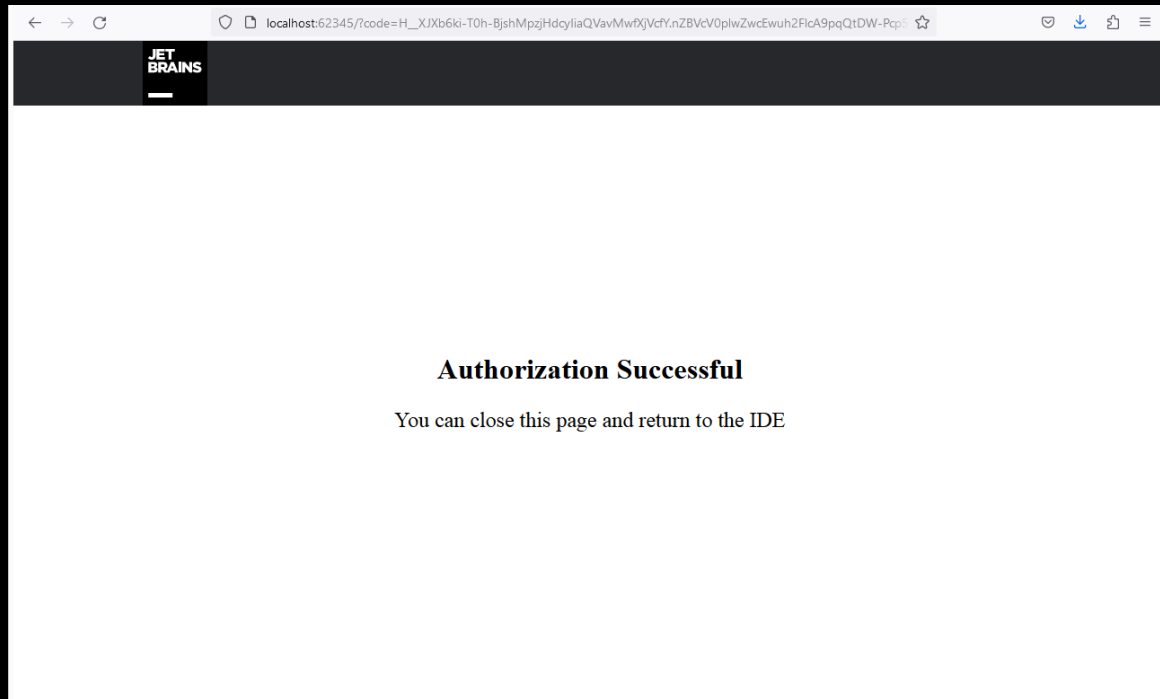
Go to your email and check your inbox. Click the link in the mail from JetBrains to confirm your registration.



It will open your profile page. Fill up the form, create a password and complete your profile.

IDE LICENSE ACTIVATION

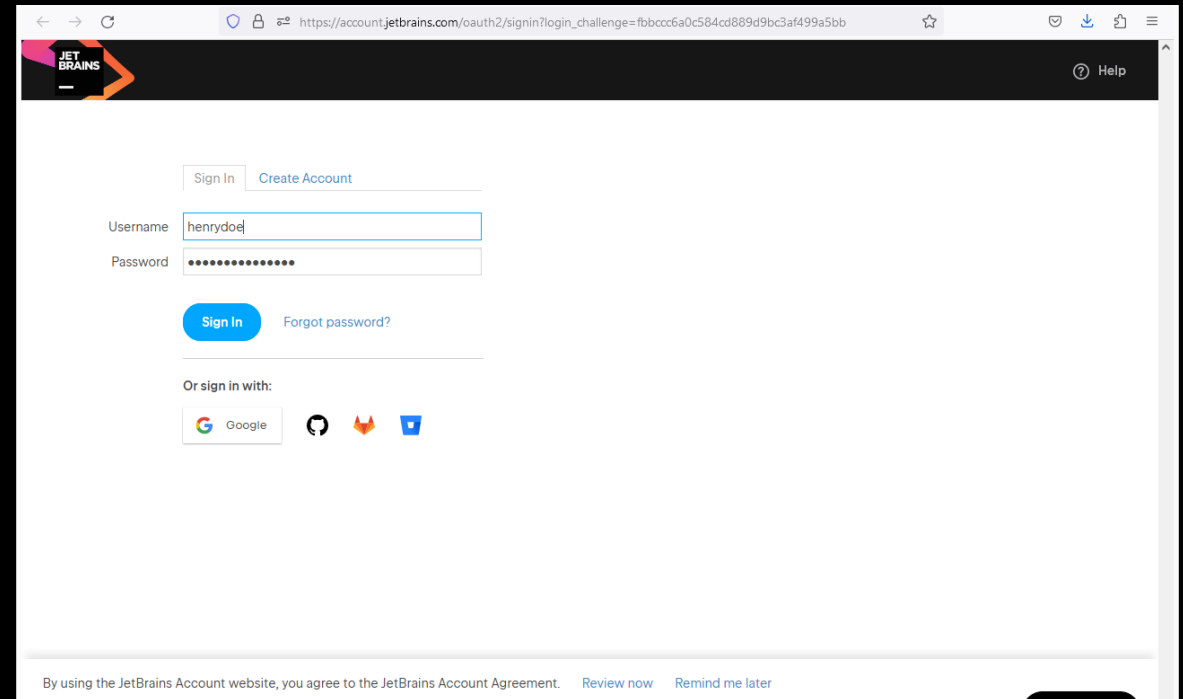
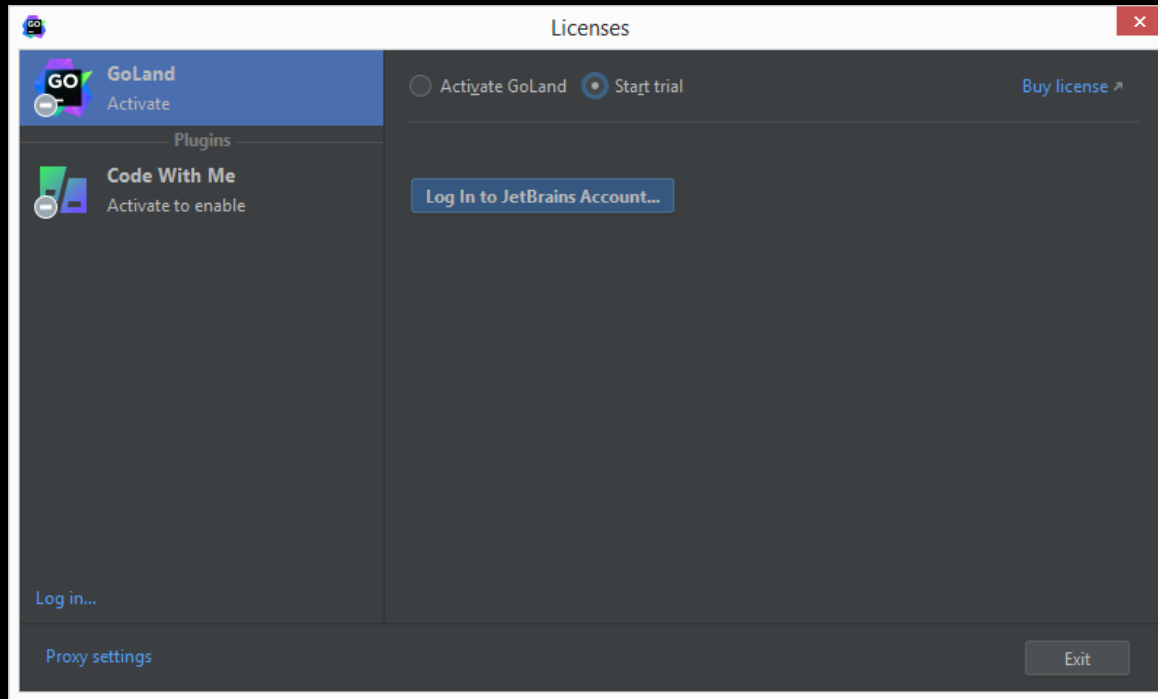
After completing your registration you'll see the success message that says to go back to the IDE. If your IDE looks like the picture on the right, then you're done and activation is completed.



But if not, click the Log In button once again, and log in to your newly created JetBrains account.

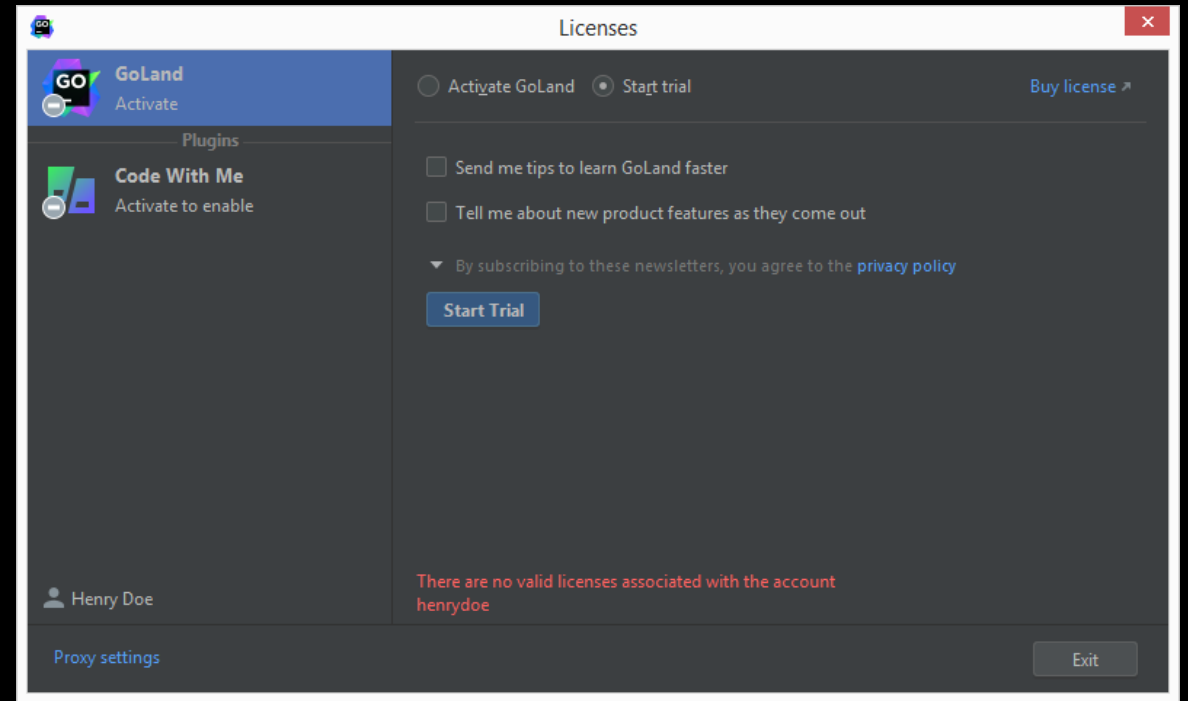
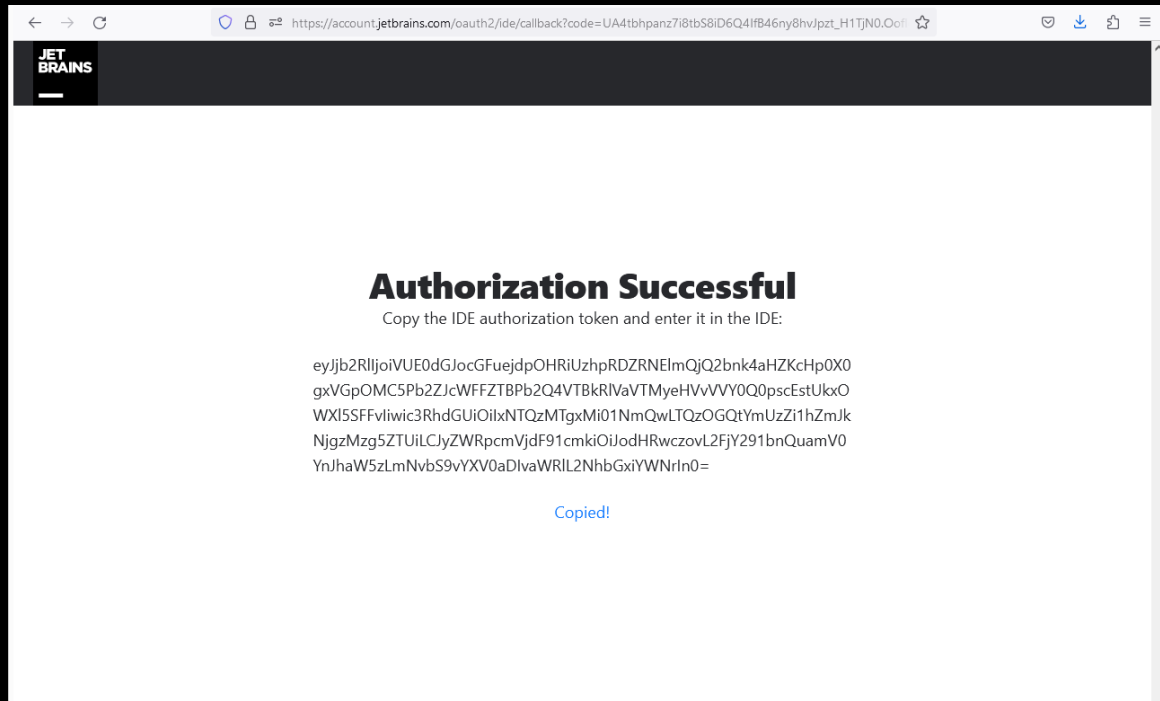
IDE LICENSE ACTIVATION

Log in to your JetBrains account



IDE LICENSE ACTIVATION

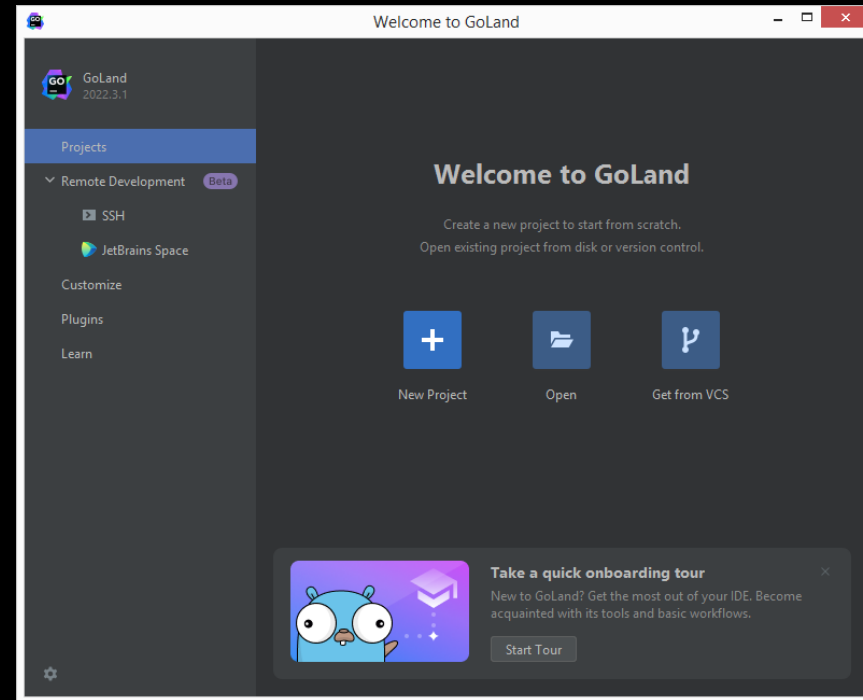
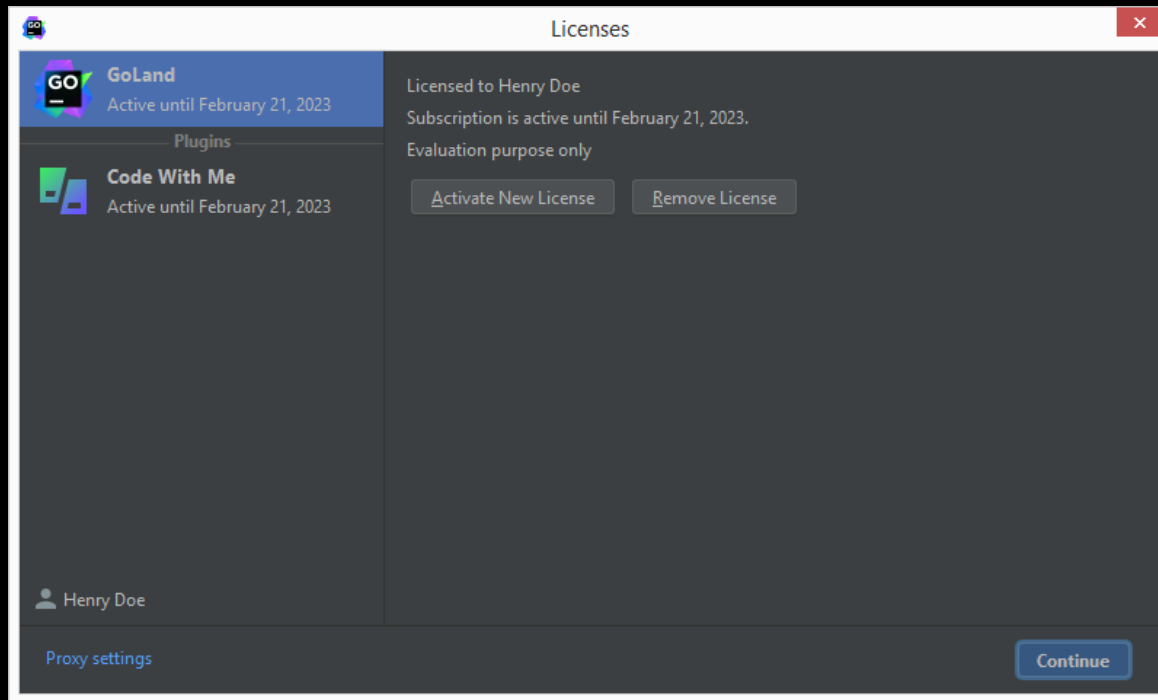
Copy the authorization token, enter it in the IDE and click the Check Token button



After successful authorization you should see the window like the picture on the right. Click the Start Trial button.

IDE LICENSE ACTIVATION

Click Continue button.



Now you are all set up and can freely work with GoLand IDE for the next 30 days

SECTION FIVE

First program

HELLO WORLD!



Source code:

lectures-part-1/lecture1/firstProgram.go

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Салом дунё!")
7 }
```

Console command:

```
1 .../lecture1 (main)$ go run firstProgram.go
2 Салом дунё!
```

Try typing this code and running it

HELLO WORLD!



Explanation

lectures-part-1/lecture1/firstProgram.go

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Салом дунё!")
7 }
```

Package name block

Imports block

Entry point of the program

Body of the main function

Try typing this code and running it

