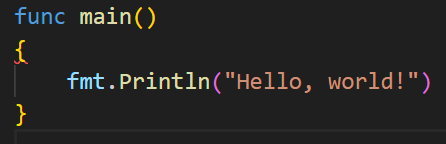
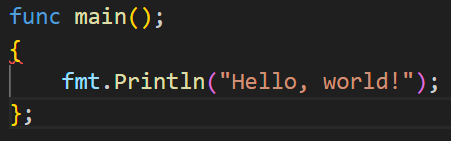
FMT

I the semicolon (virgule) Insertion rule

* If the last token before a new line is any of the following : a semicolon will be inserted after the token
* An identifier (includes words like ***float64***)
* A basic literal (number or string constant)
* One of the tokens ***break***, ***continue***, ***fallthrough***, ***return***, ***++***, ***--***, ***)***, or ***}***
* If we write our code like this :



* The semicolon insertion rule sees the ) at the end of func main () and turns that into:

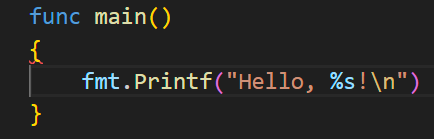


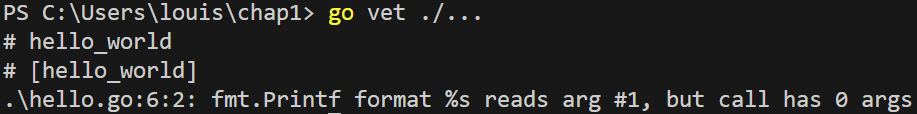
→ That’s not valid Go

* The semicolon insertion rule and the resulting brace (brace = curly brackets = accolade) placement
* Makes the Go compiler simpler and faster while enforcing a coding style

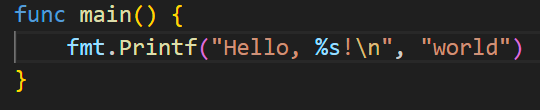
II Go vet

* The go tool includes a command called ***go vet*** to detect errors





* Now that we found the error we can easily fix it



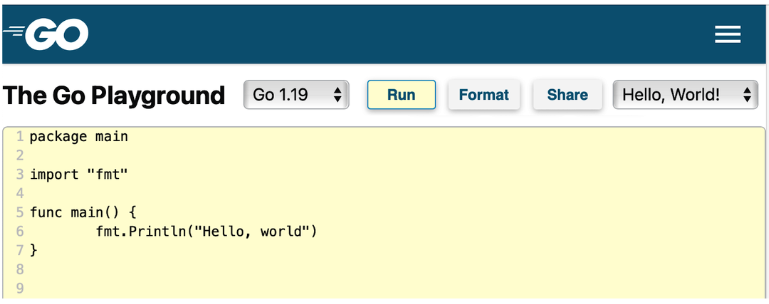
* While go catches several common programming errors, there are things that it can’t detect
* Third-part Go code-quality tools can close the gap

III Goland

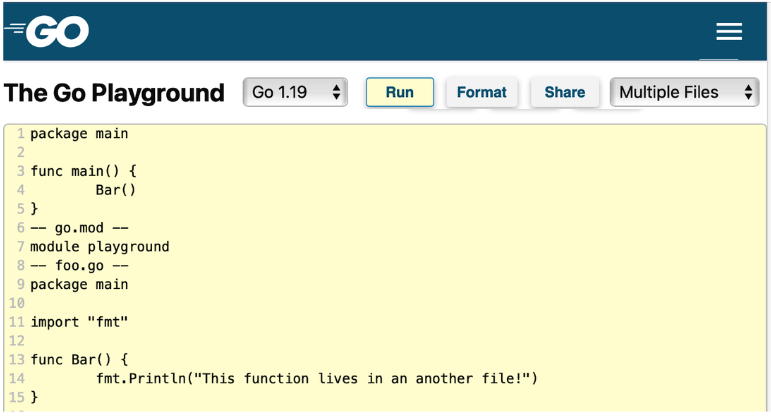
* It’s the specific ***IDE*** from ***JetBrains*** : refactoring, syntax highlighting, code completion/navigation, debugger…
* Includes ***JS***/***HTML***/***CSS***/***SQL*** database tools

IV The Go Playground (<https://go.dev/play/>)

* It’s a place to try and share small programs
* The format button runs go ***fmt*** on our program and updates your imports
* The share button creates a unique URL that you can send to someone else to take a look at your program

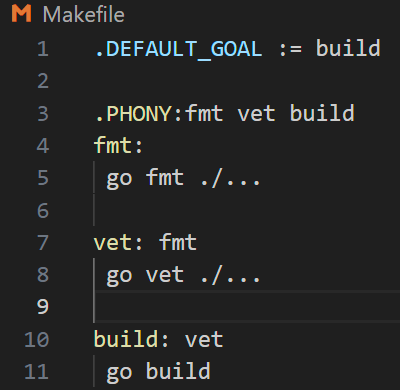


* We can only make connections to localhost
* Processes that run for too long/use too much memory are stopped
* The Go Playground supports multiple files



V Makefiles

* Lets developers specify a set of operations that are necessary to build a program and the order in which the stapes must be performed



* Each possible operation is called a ***target***
* ***.DEFAULT\_GOAL*** defines which target is run when no target is specified
* In this case : the default is the ***build*** target
* The world before the colon (***:***) is the name of the target
* Any word after the colon (like ***vet***) are the other targets that must run before the specified target runs
* The tasks that are performed by the target are on the intended line after the target