

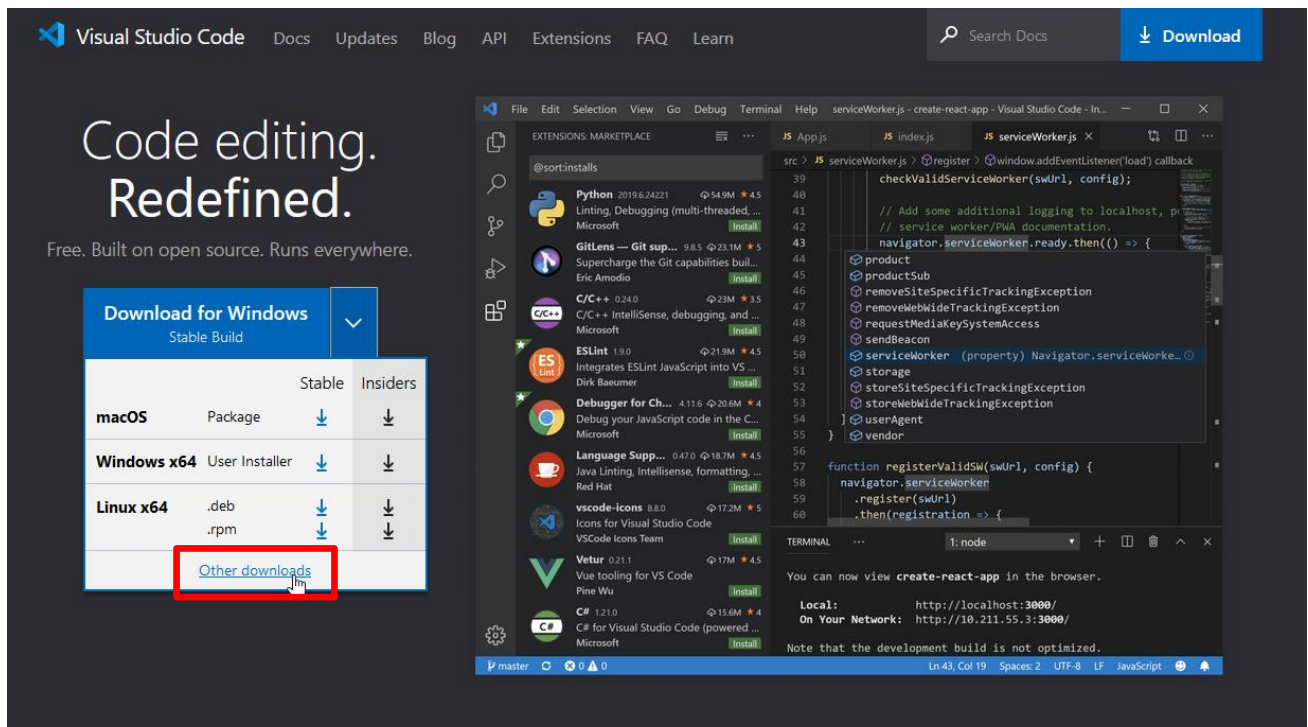
1. Środowisko programistyczne

1.1. Instalacja Visual Code

Pobieramy program **Visual Studio Code** ze strony:

<https://code.visualstudio.com/>

Wybieramy przycisk **Download for Windows**, następnie wybieramy **Other downloads** i opcję **System Installer** właściwą dla używanego systemu operacyjnego.



IntelliSense

Run and Debug

Built-in Git

Extensions



↓ Windows
Windows 7, 8, 10

↓ .deb
Debian, Ubuntu

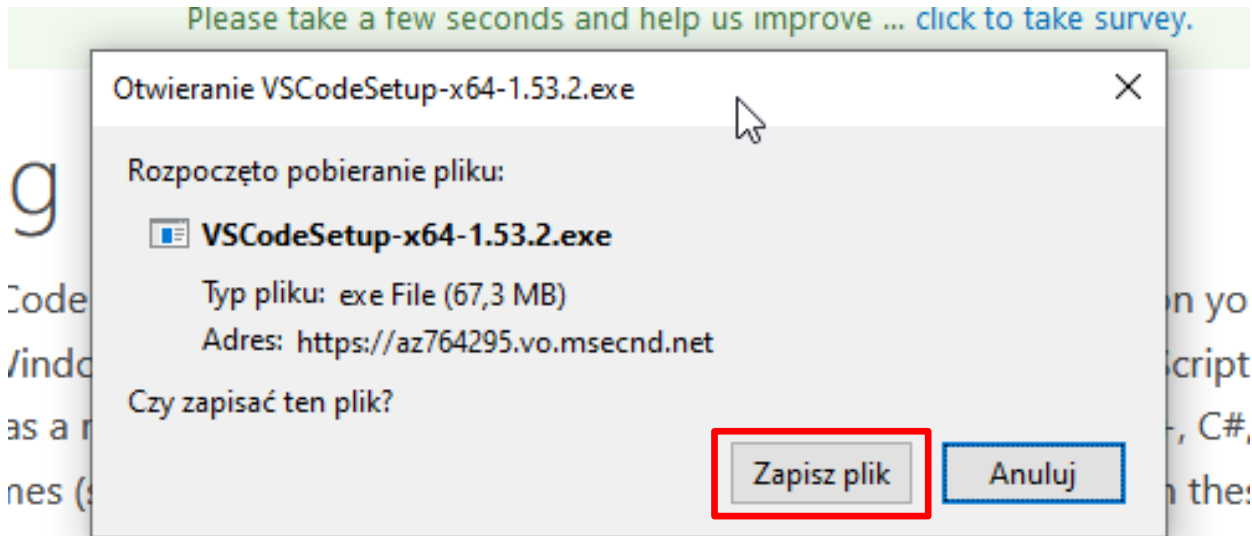
↓ .rpm
Red Hat, Fedora, SUSE

↓ Mac
macOS 10.10+

User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM

.deb 64 bit ARM ARM 64
.rpm 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store

Pojawia się okno dialogowe na którym wybieramy **zapisz**:



Wykonujemy instalację. **Podczas niej** zaznaczamy opcję **Add „Open with Code”** action to Windows Explorer

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☒ Create a desktop icon

Other:

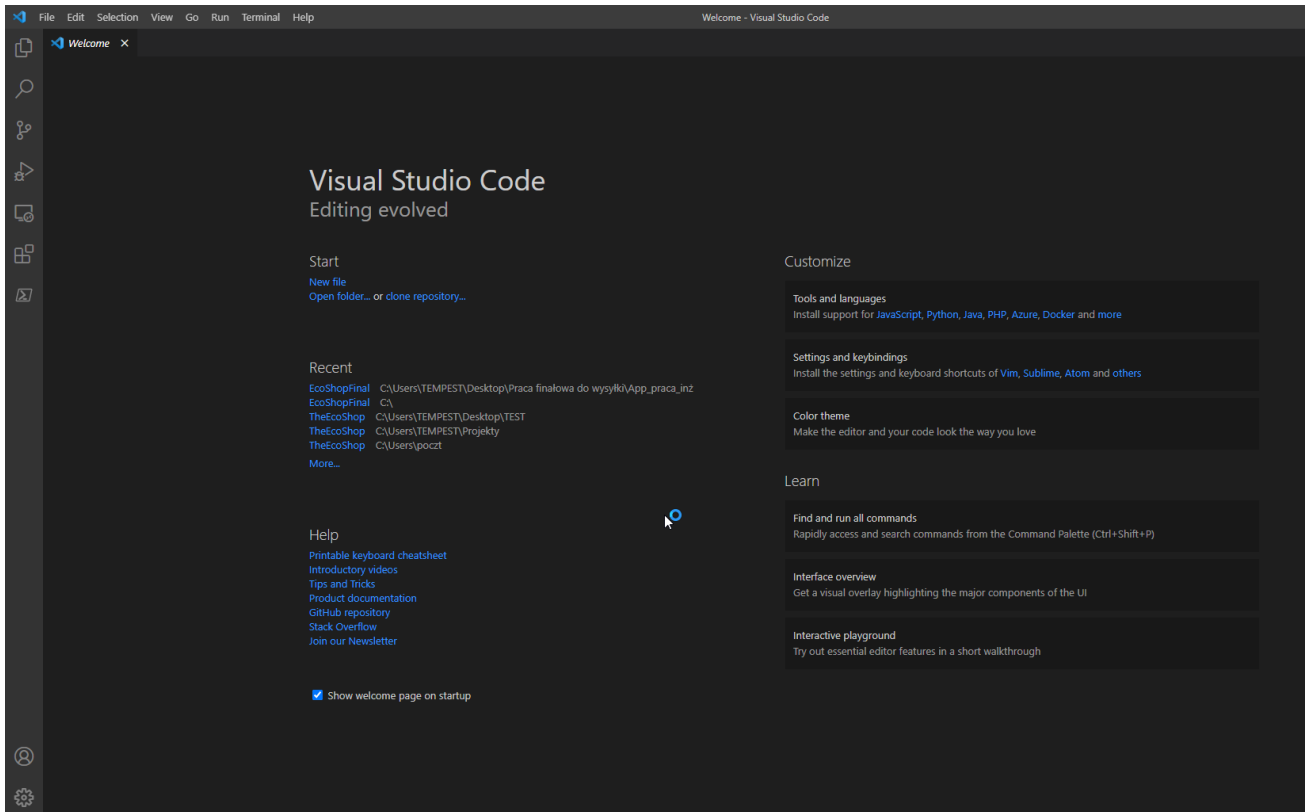
☒ Add "Open with Code" action to Windows Explorer file context menu


☒ Add "Open with Code" action to Windows Explorer directory context menu

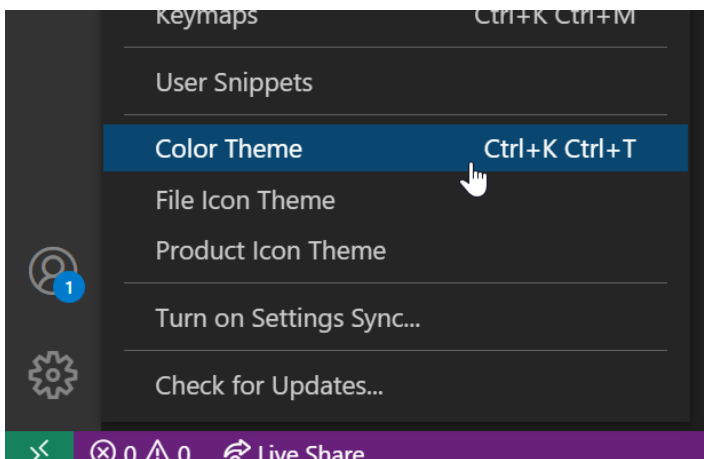
☐ Register Code as an editor for supported file types

☒ Add to PATH (requires shell restart)

Po instalacji i uruchomieniu programu **powinniśmy wiedzieć** poniższy ekran:

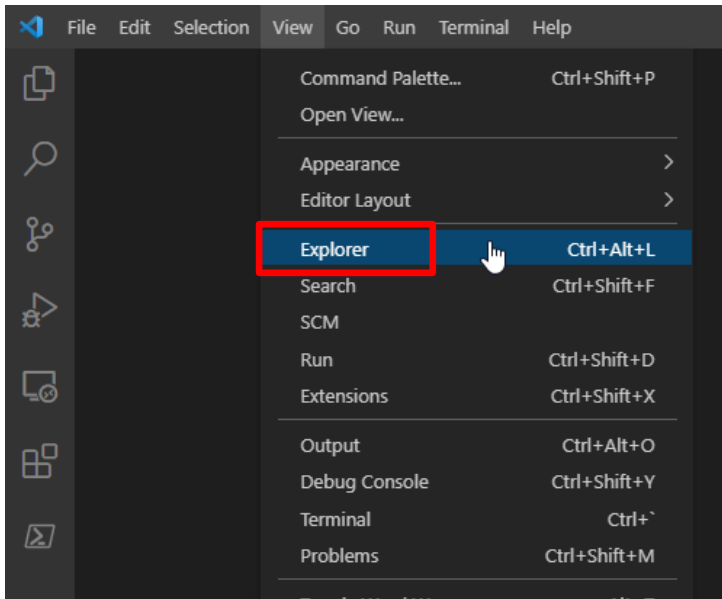


Aby zmienić tło na ciemne (polecane dla programistów) należy: ikona  w lewym, dolnym rogu -> workbench -> appearance -> color theme -> dark+ LUB color theme

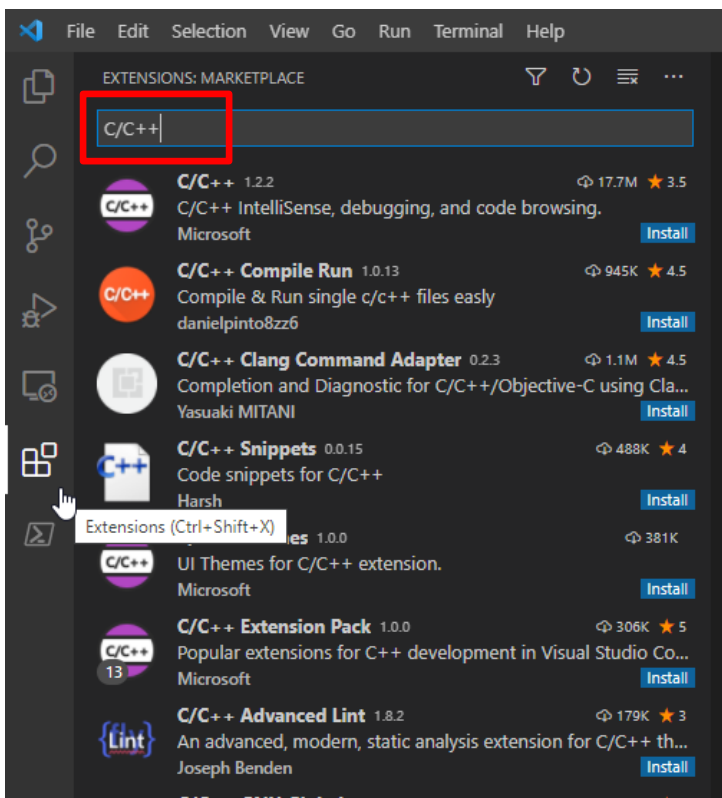


1.2. Instalacja rozszerzeń

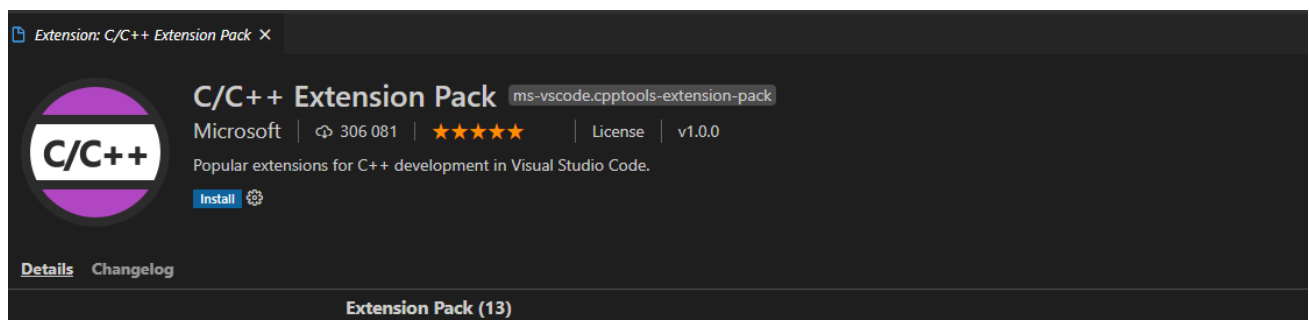
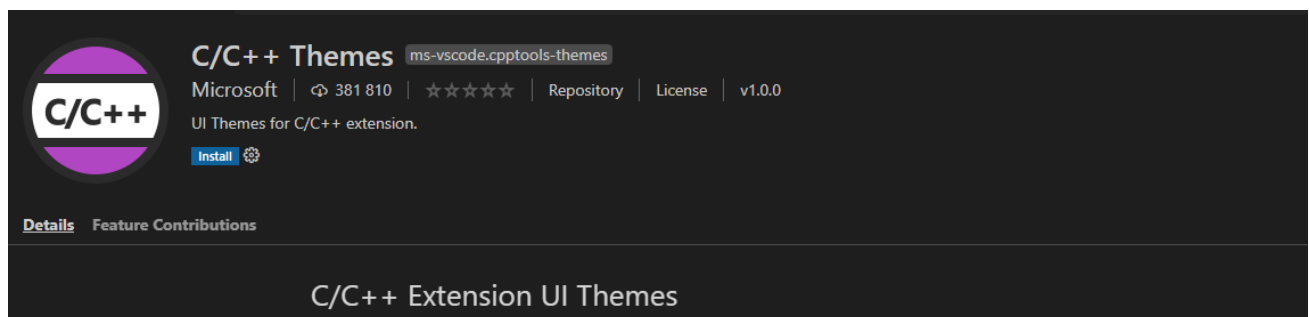
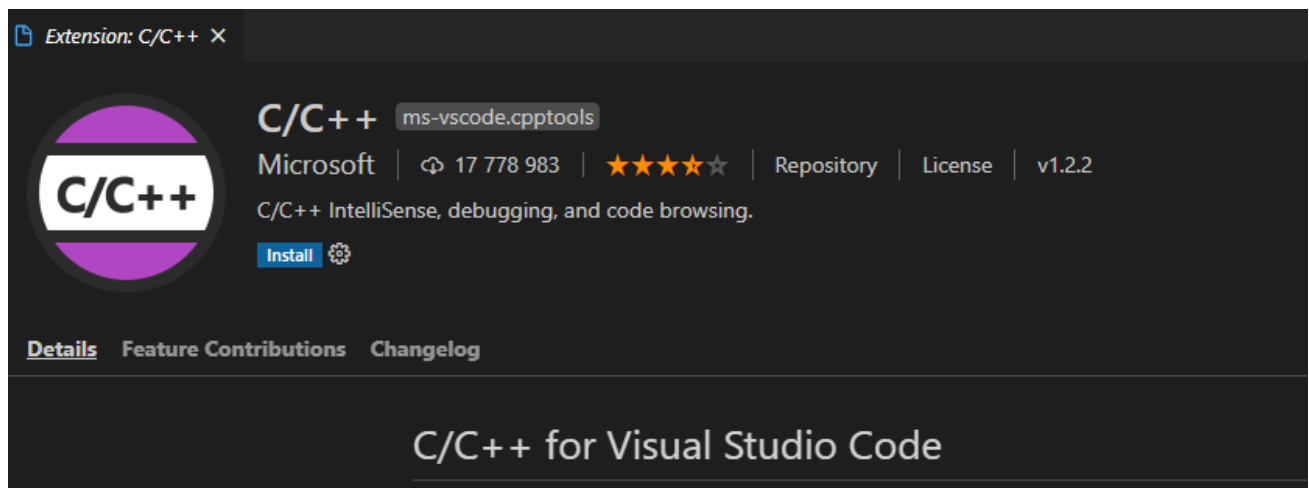
W programie Visual Code włączamy View -> Explorer



Na belce, po lewej stronie włączamy rozszerzenia i wpisujemy C/C++



Instalujemy następujące:



1.3. Instalacja kompilatora

Język C i C++ nie jest „bezpośrednio rozumiany” przez komputer. Jest czytelny dla programisty, jednak musi być przetłumaczony na język maszynowy. Proces ten nazywany jest **kompilacją**. Jednym z kompilatorów jest **GCC**, znajdujący się w paczce **MinGW-w64**.

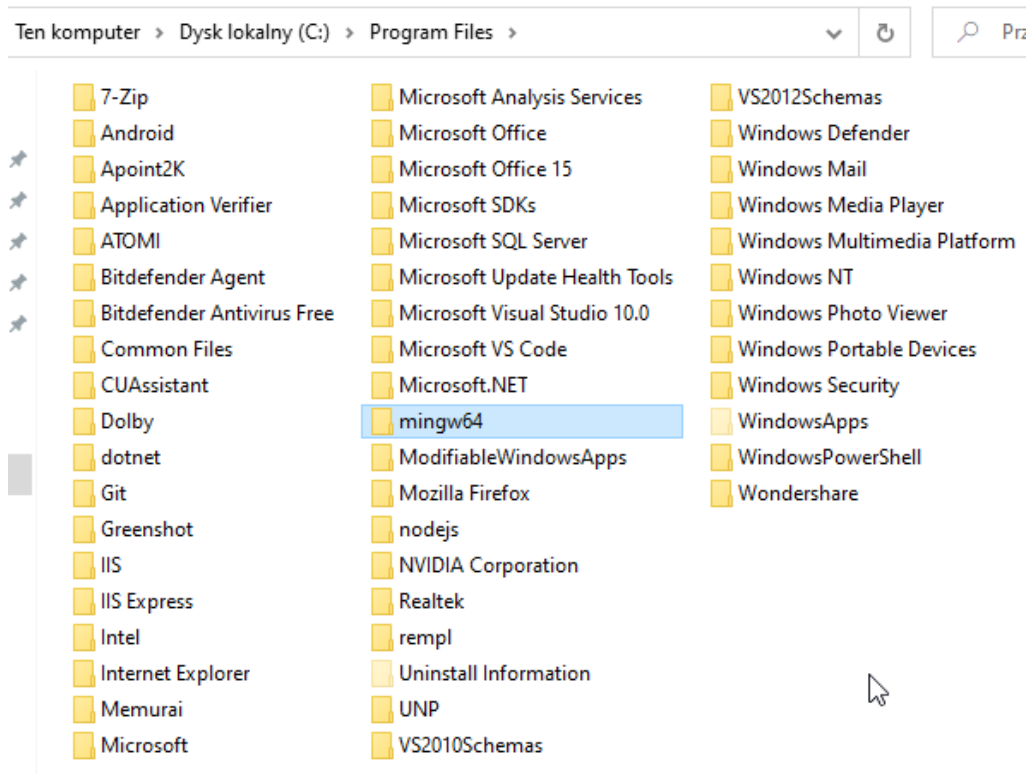
Aby zainstalować kompilator wchodzimy na link:

<https://sourceforge.net/projects/mingw-w64/>

wyberamy zakładkę **Files**, na dole jest widoczna opcja: **x86_64-posix-seh**, którą ściągamy.




Pobrany kompilator rozpakowujemy do folderu na dysku, np.: **C:\ProgramFiles\mingw64**



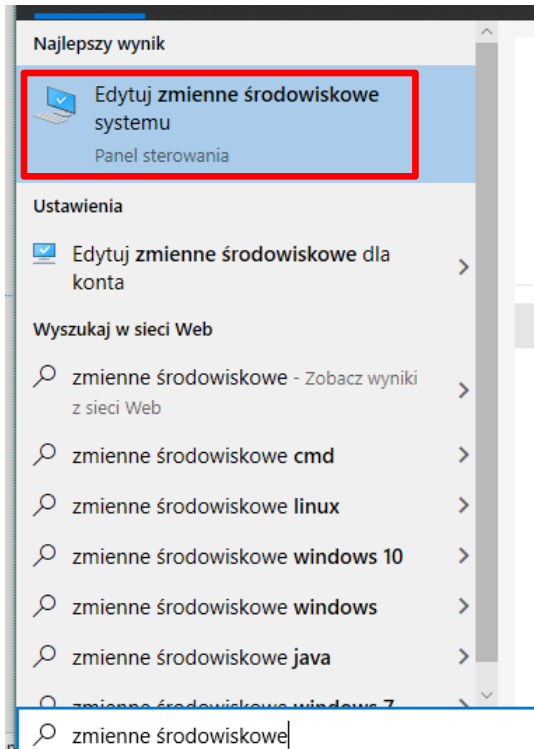
Zapisujemy ścieżkę do podfolderu bin w notatniku dla wygody (będzie nam potrzebna później) np.:

C:\Program Files\mingw64\bin

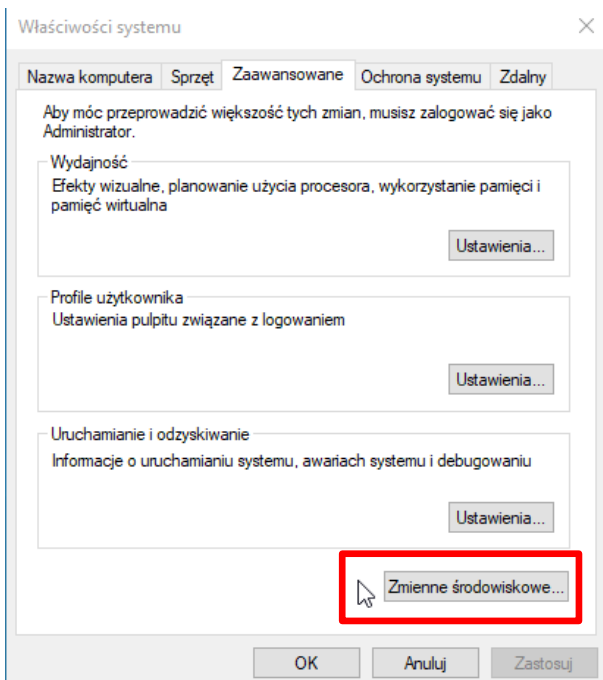
Aby to zrobić najprościej wejść do folderu i skopiować ścieżkę z paska adresu na górze.

Następnie musimy dodać ścieżkę do kompilatora do ścieżki środowiskowej **PATH** Windowsa. Klikamy ikonę Windows na belce ekranu głównego, w lewym dolnym rogu .

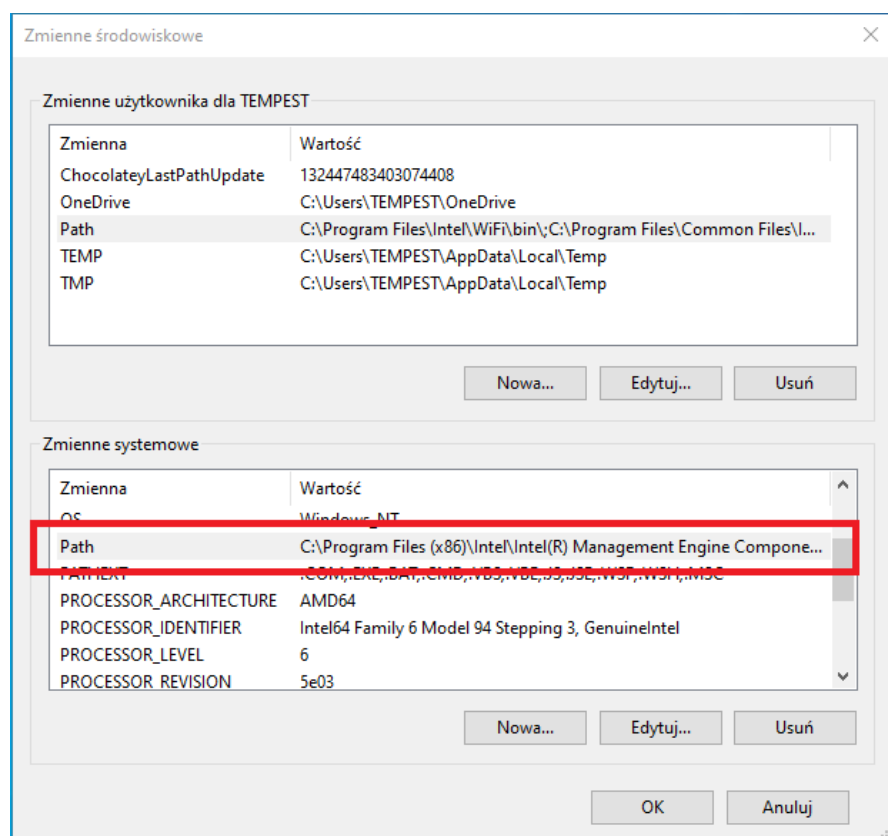
Zaczynamy pisać zmienne środowiskowe i włączamy pierwszą opcję (otwórz):



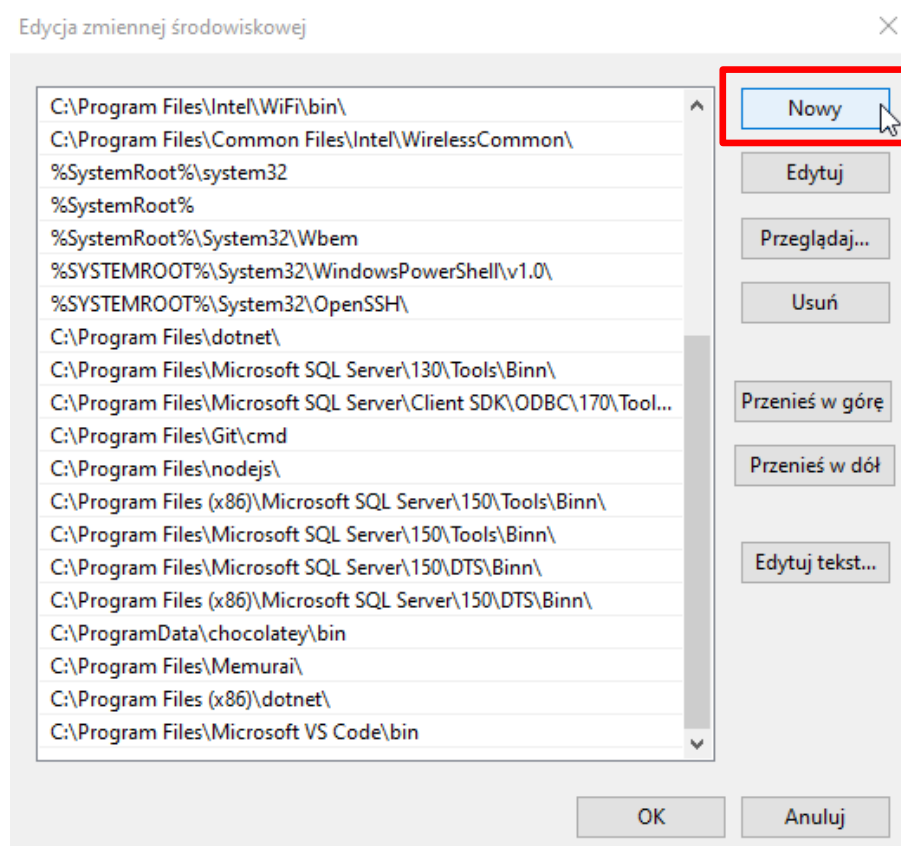
W zakładce zaawansowane wybieramy przycisk **Zmienne Środowiskowe**



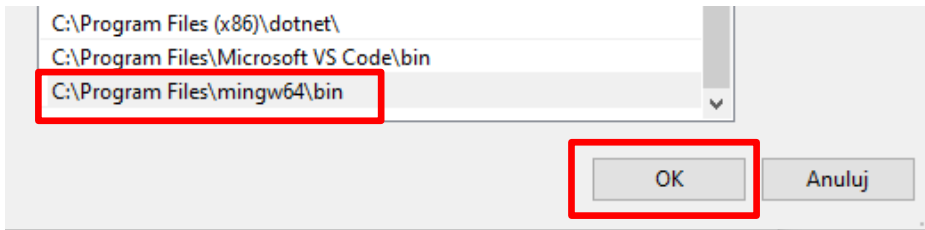
Wybieramy **zmienną systemową PATH**:



Naciskamy **Edytuj** i wybieramy przycisk **Nowy**:



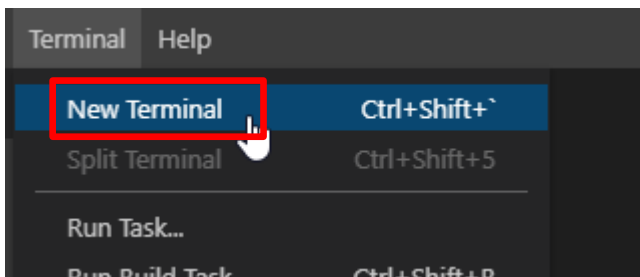
W wierszu, który się pojawił wklejamy lub wpisujemy ścieżkę do bin kompilatora, **zatwierdzamy wszystko**:



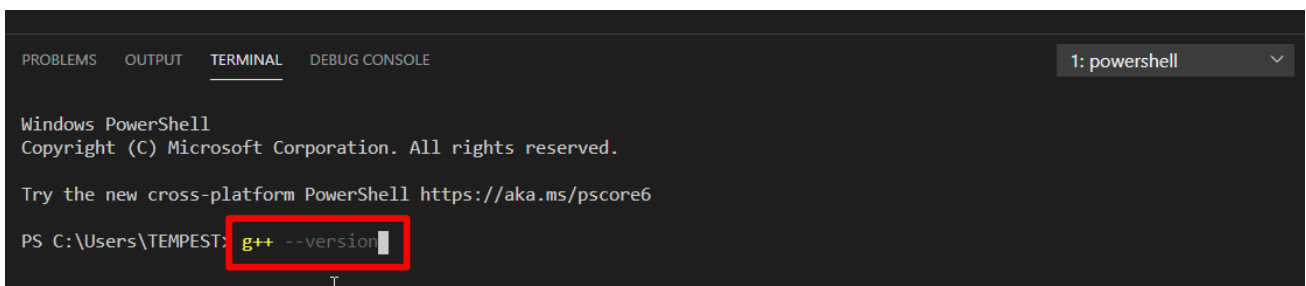
Wykonujemy restart komputera.

Aby sprawdzić czy mamy zainstalowany kompilator:

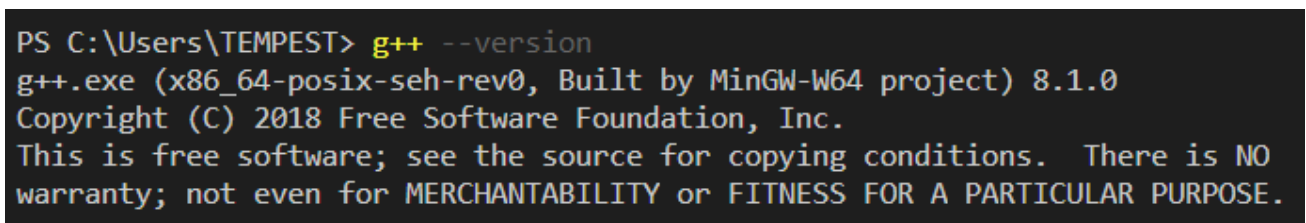
- uruchamiamy program Visual Studio Code,
- włączamy Terminal:



- w terminalu wpisujemy komendę **g++ --version**:



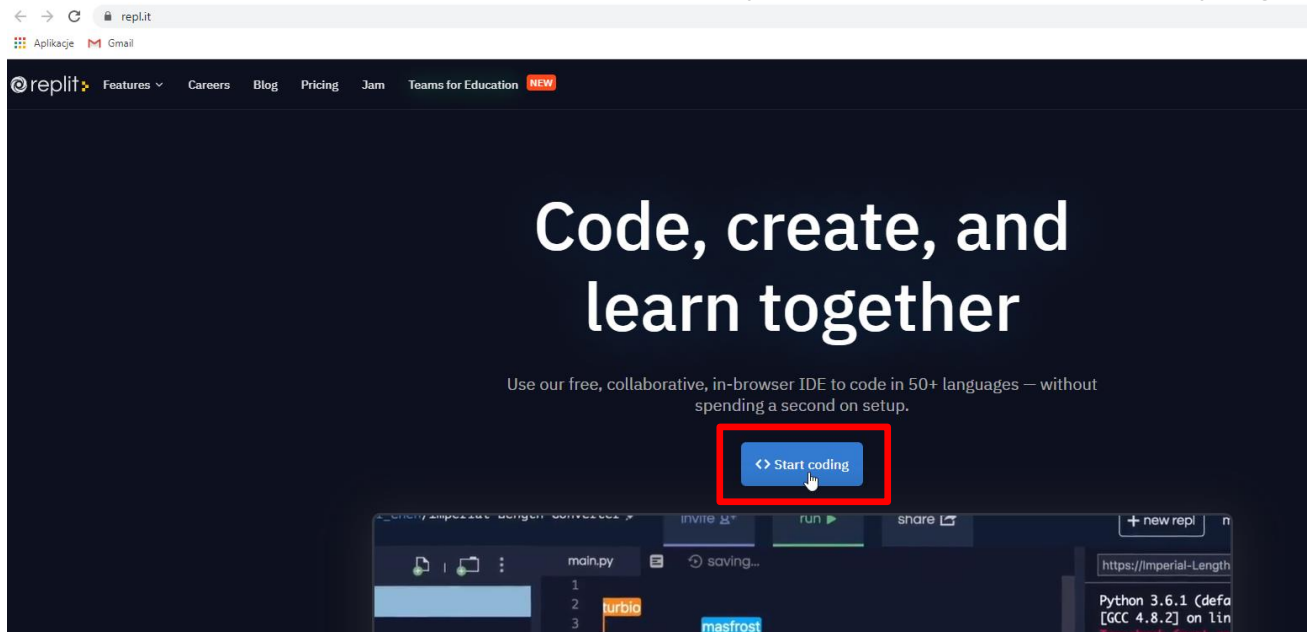
Przy poprawnej instalacji pojawia się komunikat:



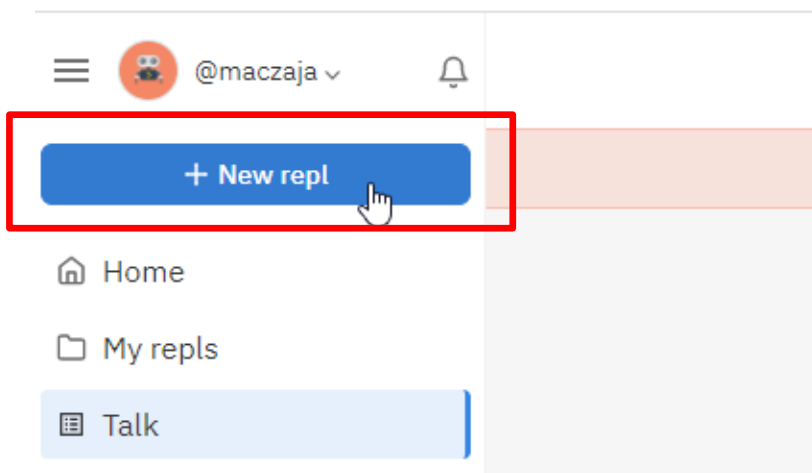
Możliwe jest używanie programu i kompilatora on-line (bez instalacji). Jest on dostępny na stronie [www](http://www.repl.it):

www.repl.it

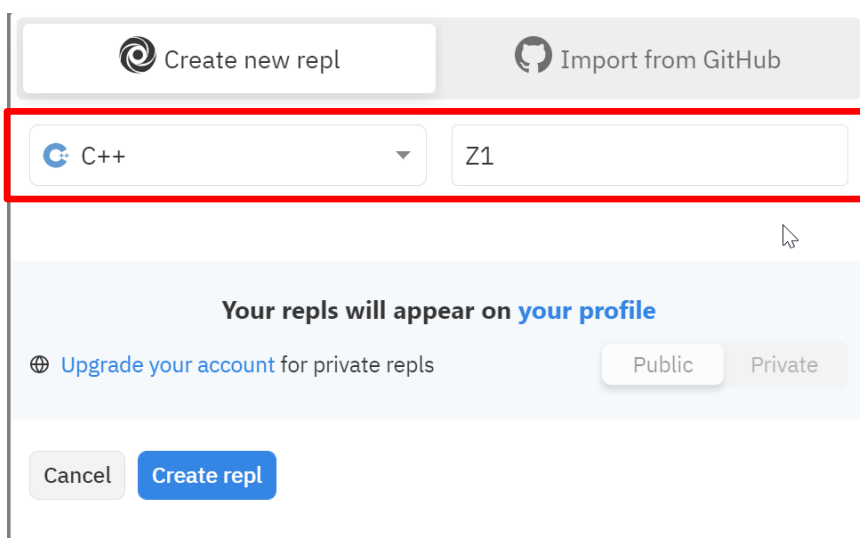
Jest to **dodatkowe** rozwiązanie, z uwagi na to, że wymagany jest dostęp do Internetu oraz dostępność samej strony.



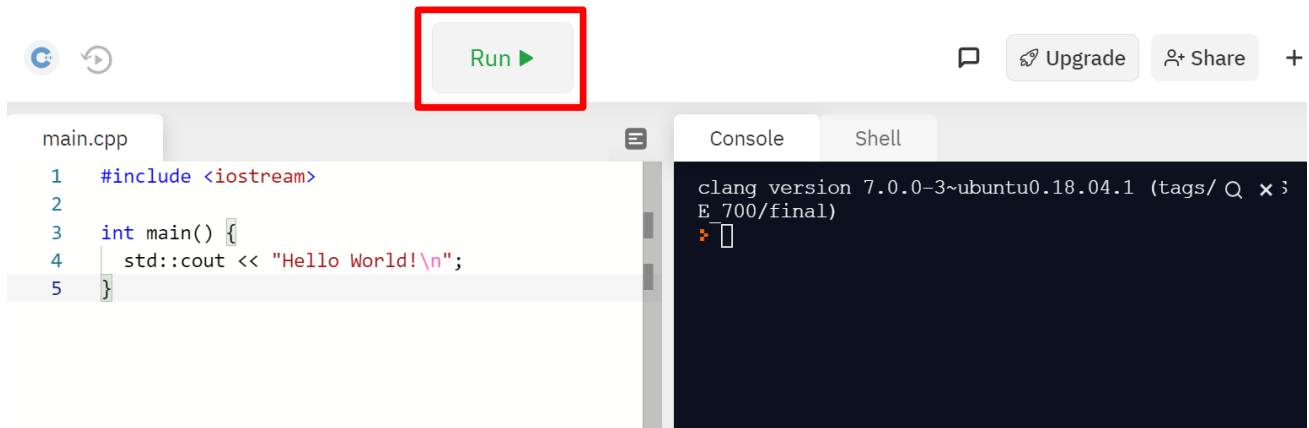
Aby pracować **wyłącznie online** należy nacisnąć przycisk **Start coding**, zarejestrować się, wybrać opcję:



Wybrać język programowania C.:



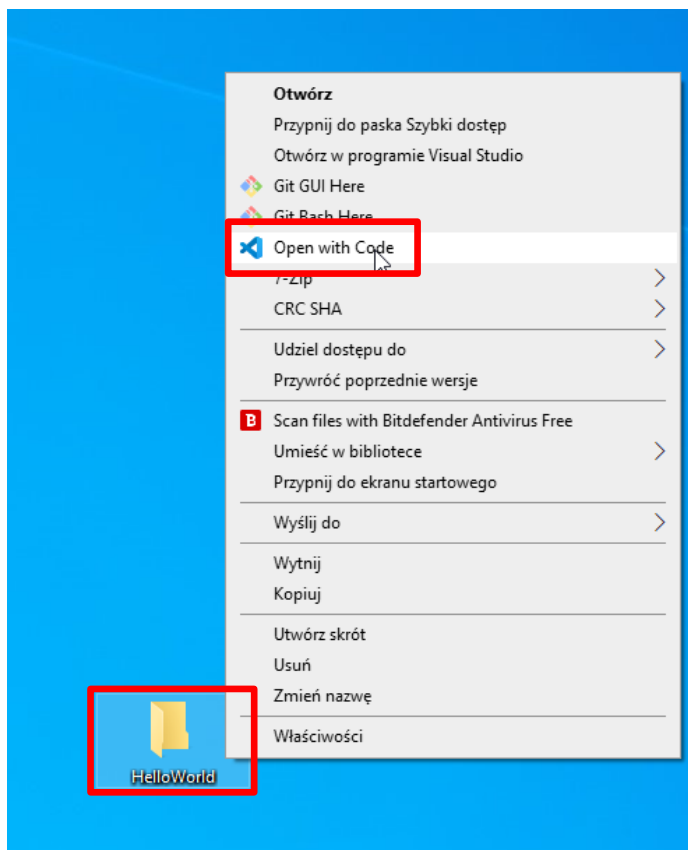
I na tym etapie jesteśmy gotowi do pracy:



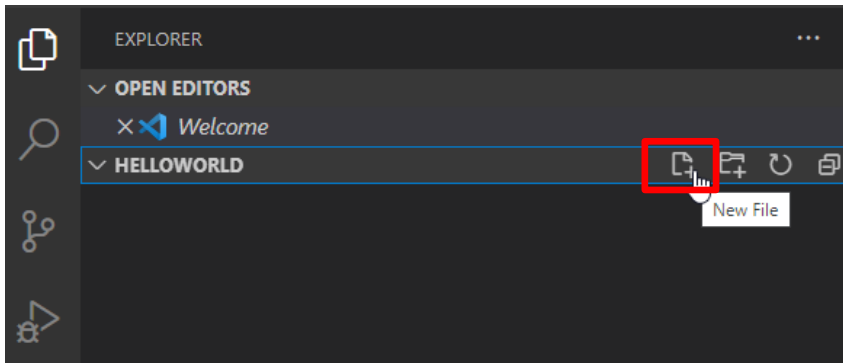
Program **kompilujemy** i uruchamiamy przyciskiem **Run**.

2. Pierwszy program

Na pulpicie (lub w innym, dowolnym miejscu **tworzymy folder** o nazwie, np.: **HelloWorld** i otwieramy Visual Studio Code pod prawym przyciskiem lub ręcznie w programie:



Z poziomu VSCode tworzymy nowy plik w folderze:



O nazwie helloworld.cpp

W treści pliku, w oknie wpisujemy kod w języku C++:

```
#include <iostream>

int main() {
    std::cout << "Hello World!\n";
}
```

Wyjaśnieniem widocznego kodu zajmiemy się w dalszej części.

W skrócie - powyższy program ma za zadanie wyświetlić powitanie po uruchomieniu.

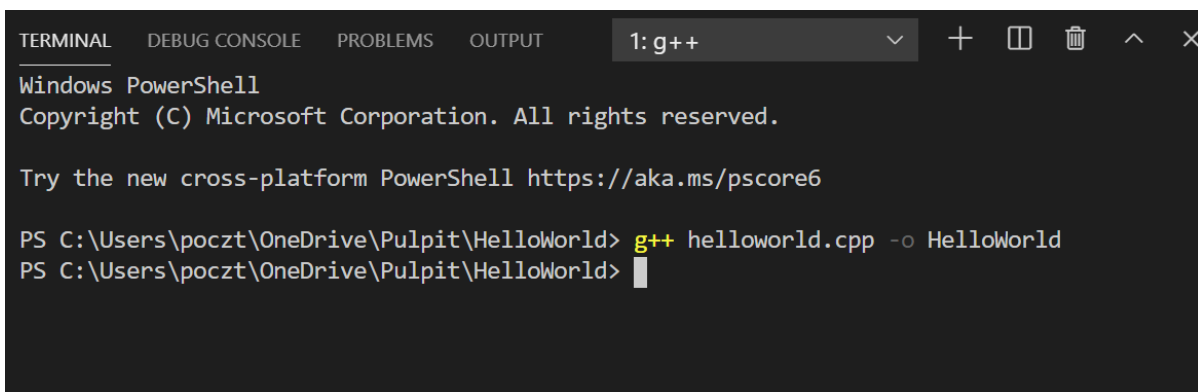
Zapisujemy zmiany.

2.1. Kompilacja programu

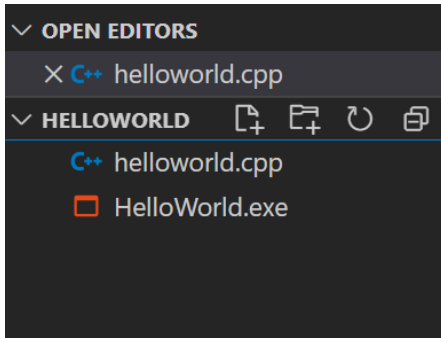
Powyższy program możemy kompilować na dwa sposoby. Używając **komend w terminalu** lub opcji dostępnych na belce nawigacyjnej.

2.1.1. Uruchomienie z poziomu komend:

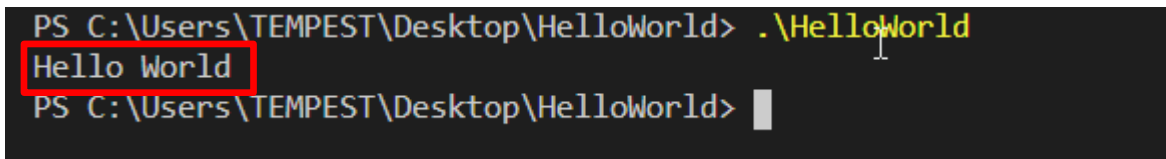
- w terminalu wpisujemy polecenie: **g++ helloworld.cpp -o HelloWorld**



Co sprawia, że utworzony zostaje plik HelloWorld.exe

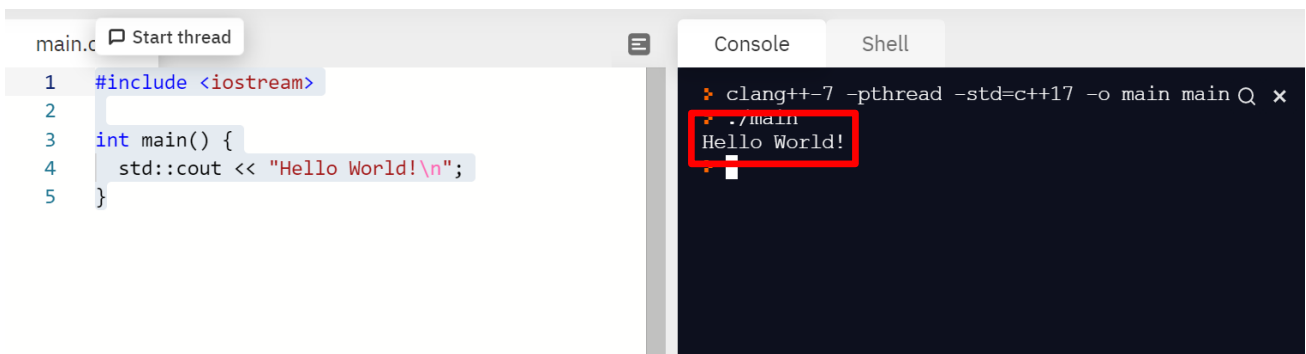


Uruchamiamy program komendą wpisywaną w terminalu: **.\HelloWorld**



Jak widać napis (efekt działania programu) pojawił się w terminalu.

Taki sam program uruchomiony w środowisku online Repl.it:



UWAGI:

- w przypadku gdy terminal **zwraca błędy kompilacji** często jest to wina antywirusa, który automatycznie kasuje (przenosi do kwarantanny pliki .exe) w momencie uruchomienia. Należy zmienić to w ustawieniach antywirusa lub wyłączyć go na czas kompilacji. Nieocenionym narzędziem wyszukiwania innych błędów jest Google 😊
- **Sugeruję wyłącznie komendy.** Celem kompilacji jest utworzenie pliku **NazwaPliku.exe**, który pozwoli na uruchomienie programu.