Zawartość:

Interpolacja stringa – mechanizm pozwalający mniej lub bardziej wygodnie formatować stringi tudzież wstawiać wartości do wnętrza stringów.

Chyba każdemu zdarzyło się pisać cos takiego:

```
string name = "Ala";
string count = "5";
string text = name + " ma " + count + "kotów";
```

Im dłuższy string i więcej wartości trzeba wstawić tym bardziej jest to denerwujące. Przykładem jest sklejanie zapytań SQL jeżeli nie korzystamy z ORM tylko wysyłamy gołe zapytania.

Jak sobie z tym poradzić w C#

Metoda starsza:

```
string text = string.Format("{0} ma {1} kotów. {0} lubi koty!", name, count);
```

Gdzie oczywiście {0} to pierwszy parametr podany po stringu a {1} to drugi parametr podany po stringu.

Metoda nowa:

```
string text = $"{name} ma {count} kotów. {name} lubi koty!";
```

Jedyny wymóg to znak $ przed pierwszym cudzysłowem.

Jak sobie z tym poradzić w Pythonie?

```
name = "Ala"
count = 5

text = f"{name} ma {count} kotów. {name} lubi koty!"
```

Jedyny wymóg to znak f przed pierwszym cudzysłowem.

JavaScript

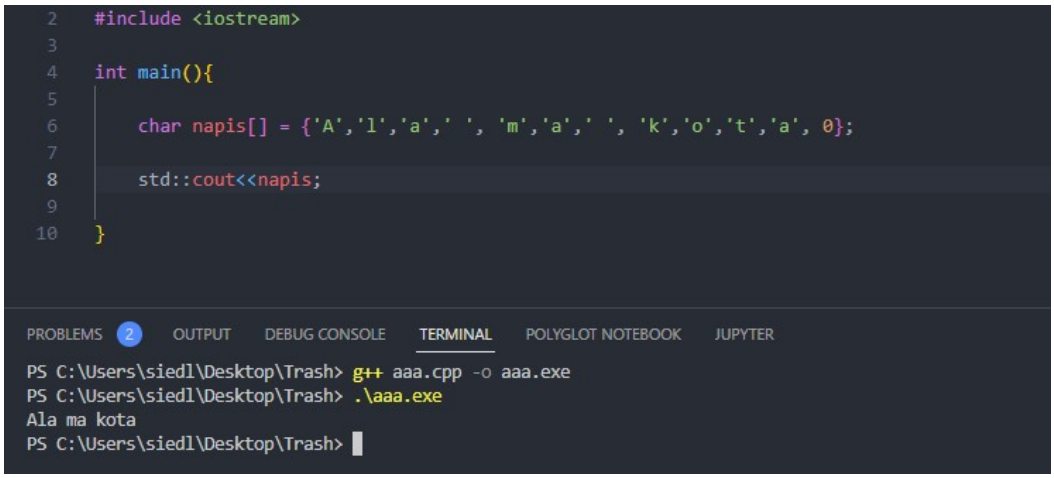
```javascript
let name = 'Ala';
let count = 5;

let text = `${name} ma ${count} kotów. ${name} lubi koty!`;
```

Tutaj składnia jest ociupinkę inna. Ważne jest by używać $ przed klamrami oraz udekorować string w tak zwane „backticki”.

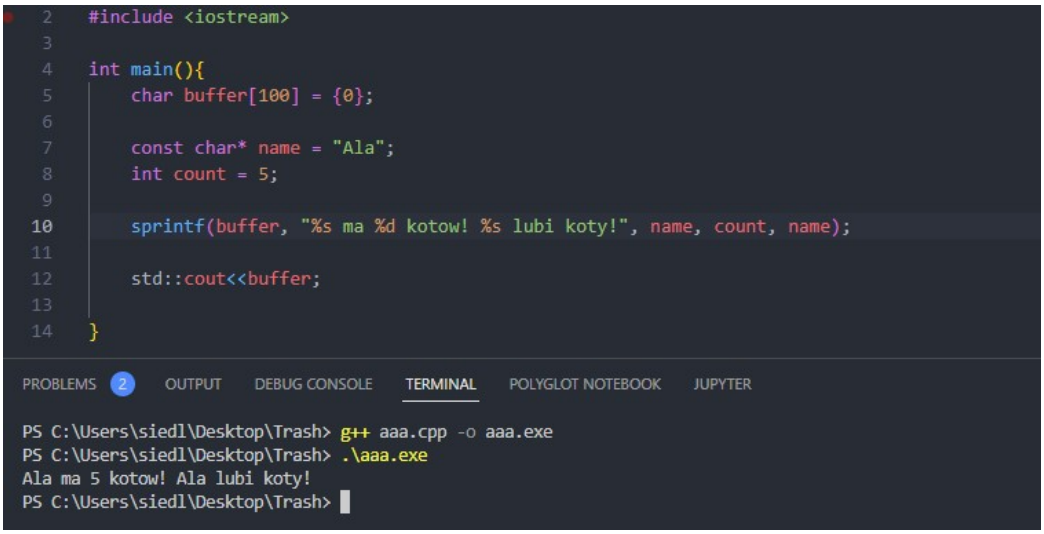
C oraz C++ (kiedy używamy stringów typu C-Style)

Jak wiadomo lub nie stringi to tablice znaków z jednym dodatkowym elementem w postaci 0 na końcu.



Takie przykładowe zapisanie C-Stringa i wypisanie na ekranie.

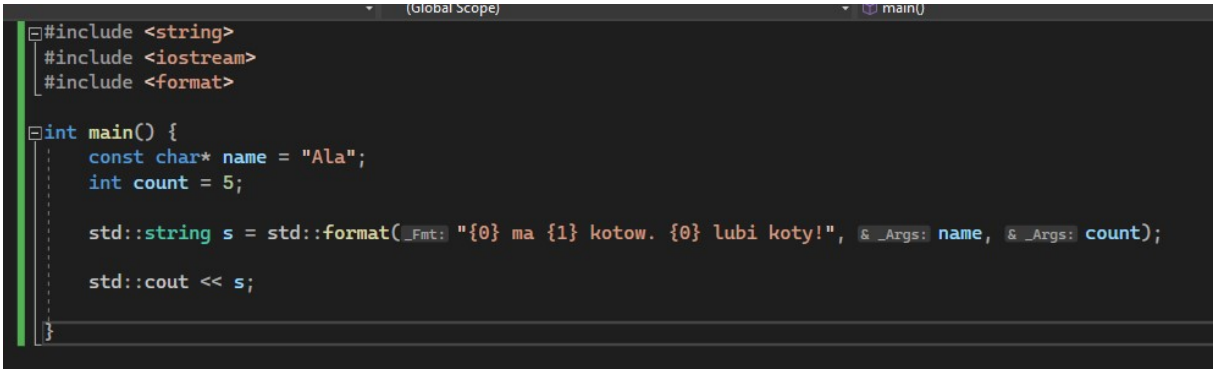
Chcemy wykonać interpolację:



1. Potrzebujemy bufor, w którym zapiszemy docelowy string. Ustawiłem mu rozmiar na 100 znaków czyli mamy do dyspozycji 99 pozycji gdyż ostatnia musi zostać 0. Dodatkowo zainicjowałem go 0, więc tam gdzie nie wstawimy znaków będą 0 – co automatycznie zakończy string.
2. Wykorzystałem funkcję sprintf(), która przyjmuje:
    a. Wskaźnik na bufor docelowy
    b. Ciąg formatujący taki sam jak podajemy do zwykłego printf
    c. Zmienne podstawiane pod kolejne %

C++ nowa metoda

Metoda ta nie jest jeszcze wspierana przez każdy kompilator. Na pewno żeby jej użyć należy ustawić kompilatorowi standard 2020 C++



Standard ustawiamy flagą (GCC, Clang): -std=C++2a

Lub w Visual Studio: