

# JĘZYKI PROGRAMOWANIA

Aby przedstawić algorytm w postaci programu komputerowego, należy zapisać go jako ciąg instrukcji języka programowania. Każda instrukcja odpowiada określonej operacji lub ciągowi operacji.

Język programowania posiada swój:

- Zbiór instrukcji (w tym słowa kluczowe oznaczające określony rozkaz, instrukcję lub polecenie)
- Zasady składni
- Właściwe słownictwo.

Instrukcje każdego języka programowania realizują czynności takie, jak:

- Wprowadzanie danych,
- Wyprowadzanie wyników,
- Wykonywanie obliczeń,
- Sprawdzanie warunków,
- Powtarzanie operacji.

Zasady dotyczące języka programowania:

- Podlega konkretnym regułom,
- Precyzja - instrukcje błędne nie mogą być przetłumaczone, komputer nie wykona programu,
- Kolejność zapisywania instrukcji odpowiada kolejności operacji realizujących dany algorytm.

Języki programowania dzielimy ze względu na poziom wykonywania programu na:

- Języki wysokiego poziomu (np. Java, C++, Python, Pascal, PHP)
- Języki niskiego poziomu (wewnętrzne) (np. assembly).

**Kod źródłowy** można zapisać w dowolnym edytorze tekstu, najwygodniej jest korzystać z edytora tekstowego. W edytorze piszemy zgodnie ze składnią wybranego języka programowania.

**Program źródłowy** nie może być wykonywany przez procesor. Procesor wykonuje kod maszynowy, a dokładniej wykonuje program w kodzie maszynowym (w języku wewnętrznym procesora).

## Program komputerowy to:

- program (kod) źródłowy - instrukcje zapisane w języku programowania,
- program (kod) wynikowy (maszynowy), zrozumiały dla komputera, ciąg instrukcji zrozumiałych dla procesora.

Napisany przez siebie program sprawdzamy w kompilatorze (C++) lub interpreterze (Python) - sprawdzamy poprawność zapisania poleceń, zgodnie z regułami składni tego języka.

**Implementacja** – powstanie programu (kodu) źródłowego za pomocą algorytmu z ciągiem instrukcji języka programowania (kodu źródłowego)

**Translacja** – przetłumaczenie programu źródłowego napisanego w języku wysokiego poziomu na program w języku niskiego poziomu (język wewnętrzny komputera).

Translacja przebiega w formie kompilacji (C, C++) lub interpretacji (LOGO, Scratch, Python), wykonywanych w translatorze (programie do tłumaczenia programu na kod maszynowy).

**Interpretacja** – tłumaczenie programu tworzonego w języku programowania instrukcja po instrukcji, każdorazowo przy uruchomieniu programu. Piszemy program składający się z wielu poleceń (skrypt), zapisujemy do pliku, uruchamiamy. Np. Python - język interpretowany.

**Kompilacja** – przetłumaczenie programu w całości na język zrozumiały dla procesora, tak by mógł go wykonać komputer. Jeśli kompilacja przebiegnie poprawnie, można uruchomić program (nie trzeba go już powtórnie tłumaczyć). Piszemy program, zapisujemy w pliku, kompilujemy, uruchamiamy poprawną (!) wersję. Przykładem jest C++ - język kompilowany.

## Ćwiczenie 1.

1. Utwórz nowy plik źródłowy i napisz program wyświetlający na ekranie napis „Hello world!” - `print("Hello world!")`
2. Zapisz program w pliku pod nazwą **Hello**.
3. Uruchom program.
4. Jeśli interpreter wykrył błędy, popraw je (literówka lub np. brak nawiasu). Zapisz plik pod tą samą nazwą i ponownie uruchom
5. Zapamiętaj gdzie na komputerze program został zapisany.

## ZMIENNE W JĘZYKU PYTHON

Zasady dotyczące nazw zmiennych w języku Python:

1. Wielkie i małe litery w nazwach traktowane są odmiennie (np. suma i Suma oznaczać będą różne zmienne). Pisząc program, należy zwracać uwagę na poprawne używanie małych i wielkich liter.
2. W nazwach zmiennych powinno się używać liter, znaku podkreślenia i cyfr.
3. Nazwa nie może zaczynać się od cyfry. Przyjęte jest stosowanie małych liter i niestosowanie polskich liter.
4. W nazwach zmiennych nie wolno stosować spacji. W przypadku nazw kilkuczłonowych zamiast spacji stosujemy znak podkreślenia.
5. Należy nadawać nazwy, które określają znaczenie danej zmiennej, np. suma, liczba\_elementow.

Zmiennej stosowanej w programie możemy nadać konkretną wartość za pomocą instrukcji przypisania. W instrukcji przypisania zmiennej podanej po lewej stronie instrukcji zostanie przypisana obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji. ZMIENNA = WYRAŻENIE.

Typ zmiennej całkowitej (int) i jej wartość: waga = 88, rok = 2021, dlugosc = 100

Typ zmiennej rzeczywistej (float) i jej wartość: a = 1,5, srednia = 4,75, wys = 180

## INSTRUKCJA WEJŚCIA

Instrukcja wejścia (wartość wpisywana z klawiatury): **input()**

Aby dana wprowadzona za pomocą instrukcji input() została zapamiętana nie jako tekst tylko jako liczba, musimy dodać instrukcję, która zamieni ciąg znaków na liczbę:

**int()** - w przypadku liczb całkowitych, **float()** - w przypadku liczb rzeczywistych.

```
miasto = input("Wprowadź nazwę miasta: ")
```

- jeśli wpiszemy z klawiatury nazwę miasta „Kraków”, w zmiennej miasto zostanie zapamiętany ciąg znaków „Kraków”.

a = input("Wprowadź liczbę: ")

- jeśli wpiszemy z klawiatury liczbę 346, w zmiennej a zostanie zapamiętany ciąg znaków „346”, a nie liczba 346,

a = int(input("Wprowadź liczbę: "))

- jeśli wpiszemy z klawiatury liczbę 346, w zmiennej a zostanie zapamiętana liczba całkowita 346,

srednia = float(input("podaj średnią ocen: "))

- jeśli wpiszemy z klawiatury liczbę 4.3, w zmiennej średnia zostanie zapamiętana liczba rzeczywista 4,3,

### Podstawowe operatory arytmetyczne w języku Python:

Operator	Działanie	Przykład instrukcji przypisania	Wynik działania dla danych: a = 11 i b = 4
+	dodawanie	suma = a + b	11 + 4 = 15
-	odejmowanie	roznica = a - b	11 - 4 = 7
*	mnożenie	iloczyn = a * b	11 * 4 = 44
//	dzielenie całkowite (z zaokrągleniem części ułamkowej „w dół” – do największej liczby całkowitej mniejszej od wyniku dzielenia)	iloraz = a // b	11 // 4 = 2
/	dzielenie zmiennoprzecinkowe (z zachowaniem części ułamkowej)	iloraz = a / b	11 / 4 = 2.75
%	obliczenie reszty z dzielenia dwóch liczb całkowitych	reszta = a % b	11 % 4 = 3

## Ćwiczenie 2.

1. Utwórz nowy plik źródłowy.
2. Wprowadź polecenia:

```
a = int(input("Podaj pierwszą liczbę: "))
b = int(input("Podaj drugą liczbę: "))
iloczyn = a * b
print(iloczyn)
```

3. Zapisz program pod nazwą **iloczyn1** i uruchom program

## Ćwiczenie 3.

1. Utwórz nowy program obliczający iloczyn  $1.5 * 2.5$
2. Zapisz program pod nazwą **iloczyn2** i uruchom program

## Ćwiczenie 4.

1. Do programu Iloczyn2 dopisz linijkę z odpowiedzią tekstową:

```
print("Iloczyn" , a , "*" , b , "wynosi" , iloczyn)
```

2. Zobacz co się zmieniło, jak można łączyć słowa ze zmiennymi.
3. Zapisz plik jako **iloczyn3**