

# PYTHON

Podstawy programowania

# Tematy

- Algorytm
- Języki programowania
- Interpreter i kompilator
- Paradygmaty i języki programowania
- Wykorzystywanie języków programowania

# Algorytm

- **Algorytm** – skończony ciąg jasno zdefiniowanych czynności koniecznych do wykonania pewnego rodzaju zadań, sposób postępowania prowadzący do rozwiązania problemu
- Zadaniem algorytmu jest przeprowadzenie systemu z pewnego stanu początkowego do pożądanego stanu końcowego.
- **Algorytmika, teoria algorytmów** – nauka o algorytmach. Jest działem informatyki, cybernetyki, ekonomii i techniki. Algorytmika zajmuje się badaniem algorytmów. Jej częścią jest natomiast algorytmizacja, czyli proces budowy konkretnego algorytmu

# Przykłady algorytmów z życia codziennego

- Książka kucharska (przepis): lista składników i czynności, które muszą być wykonane aby przyrządzić potrawę.
- Nuty
- Instrukcje montażu mebli
- Nawigacja – jak dojechać do
- Choreografia – jak wykonać taniec

# Cechy algorytmów

- poprawność — algorytm powinien zwracać poprawne wyniki
- jednoznaczność — algorytm powinien przy takim samym zbiorze danych wejściowych zwracać takie same wyniki
- skończoność dla każdego zbioru poprawnych danych wejściowych algorytm powinien zwracać wyniki w skończonej liczbie kroków
- efektywność algorytm powinien prowadzić do rozwiązania problemu w jak najmniejszej liczbie kroków

# Złożoność obliczeniowa algorytmu

- określa ilość zasobów (pamięci, czasu) niezbędnych do rozwiązania problemu.
  - Złożoność czasowa algorytmu jest rozpatrywana jako miara czasu (najczęściej liczba operacji dominujących) potrzebnego do rozwiązania problemu w zależności od rozmiaru danych wejściowych.
  - Złożoność pamięciowa algorytmu określa wielkość pamięci operacyjnej komputera, która jest potrzebna do przechowywania danych wejściowych, danych pośrednich oraz ostatecznych wyników obliczeń

# Złożoność obliczeniowa algorytmu

- pesymistyczna — określa zużycie zasobów dla najgorszego przypadku,
- oczekiwana — określa zużycie zasobów dla uśrednionych wszystkich możliwych przypadków lub dla typowych przypadków.

# Przykład – przygotowanie kawy

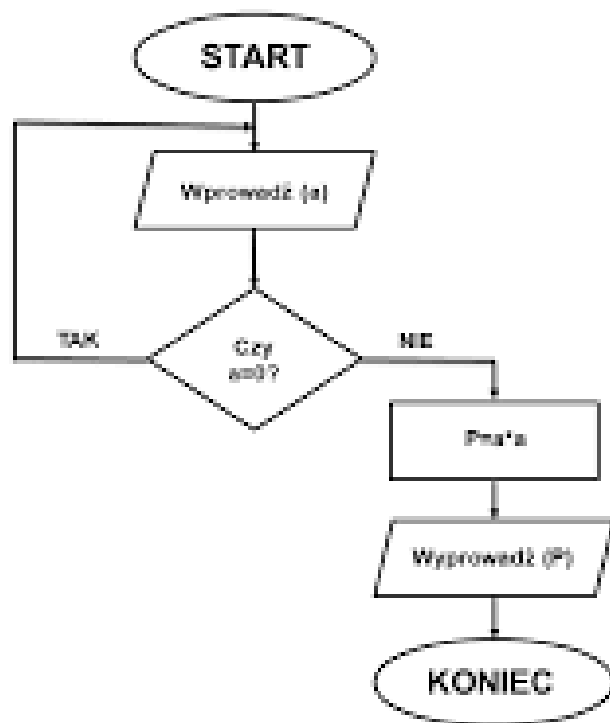
## ▶ LISTA KROKÓW:

- 1) start
- 2) przygotuj składniki
- 3) wsyp kawę do filiżanki
- 4) zagotuj wodę (2 minuty)
- 5) jeśli mam gorącą wodę przejdź do kroku 6, jeśli nie przejdź do 4
- 6) zalej kawę gorącą wodą
- 7) dodaj cukier, jeśli nie słodzysz przejdź do kroku 8
- 8) dodaj śmietankę, jeśli bez śmietanki przejdź do kroku 9
- 9) pomieszaj
- 10) stop





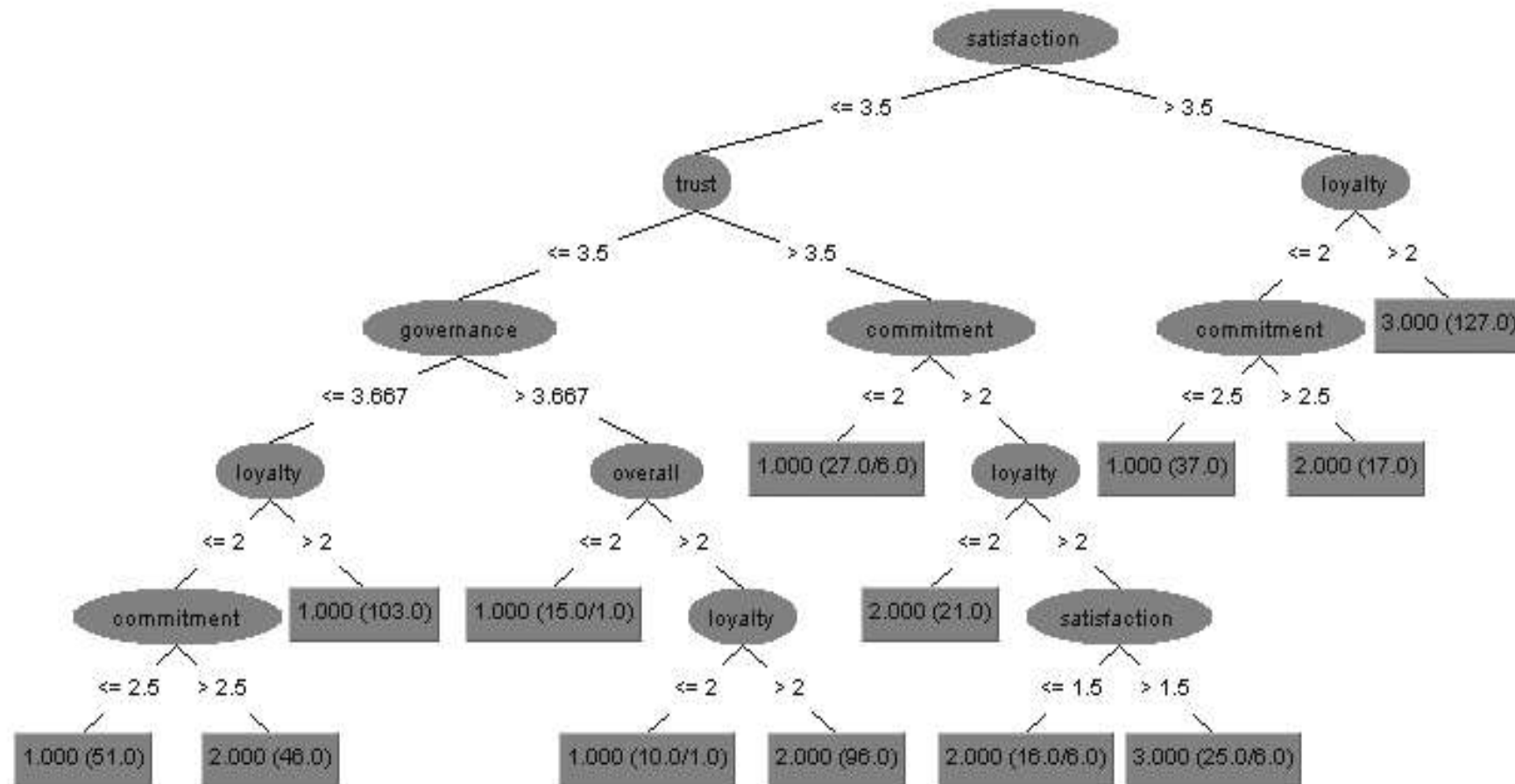
# Schemat blokowy



# Pseudokod

```
p ← 1
q ← n
dopóki p < q wykonuj
    s ← (p+q) div 2
    (*) jeżeli s*s*s < n wykonaj
        p ← s+1
    w przeciwnym wypadku
        q ← s
```

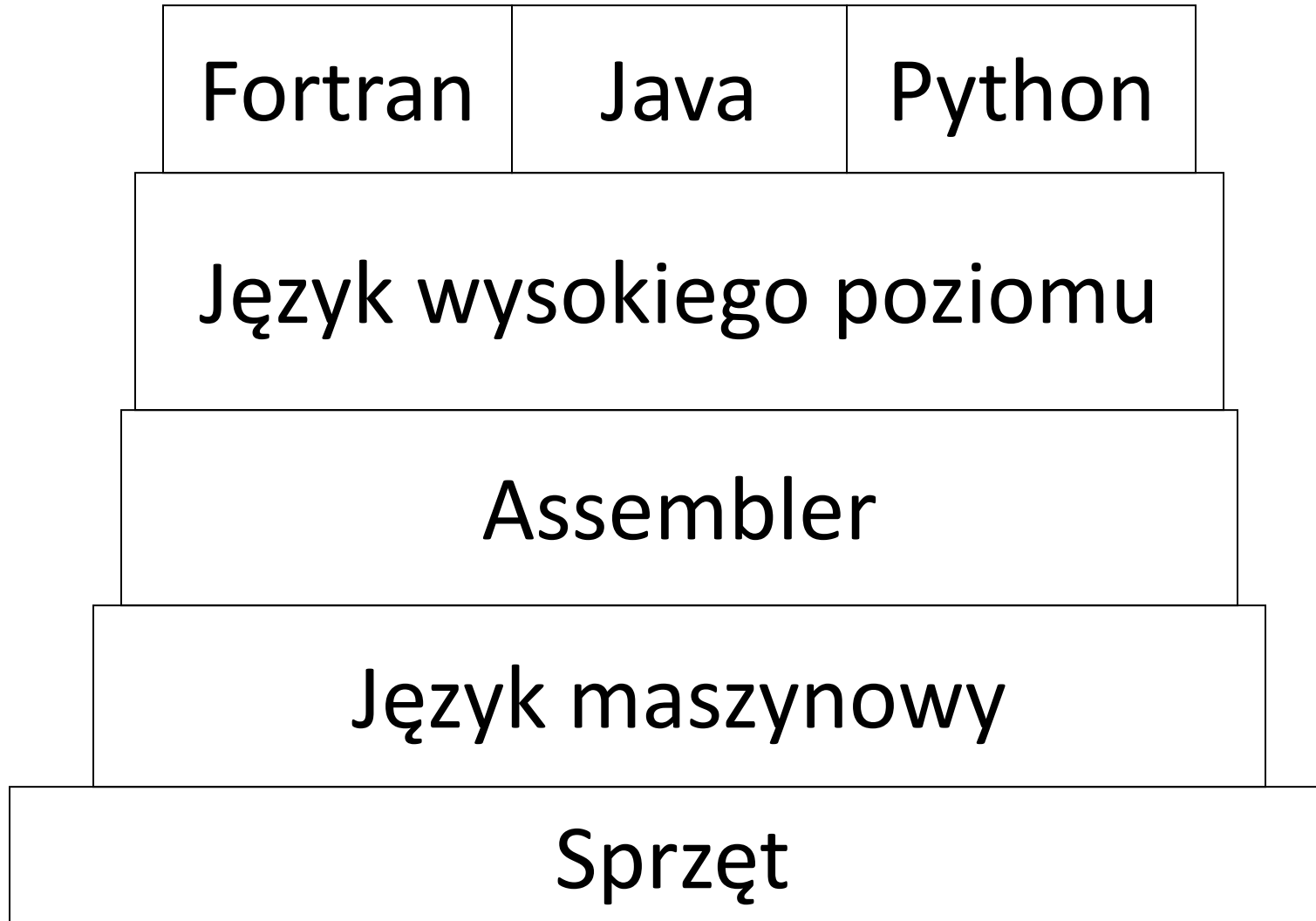
# Drzewo algorytmu przykład



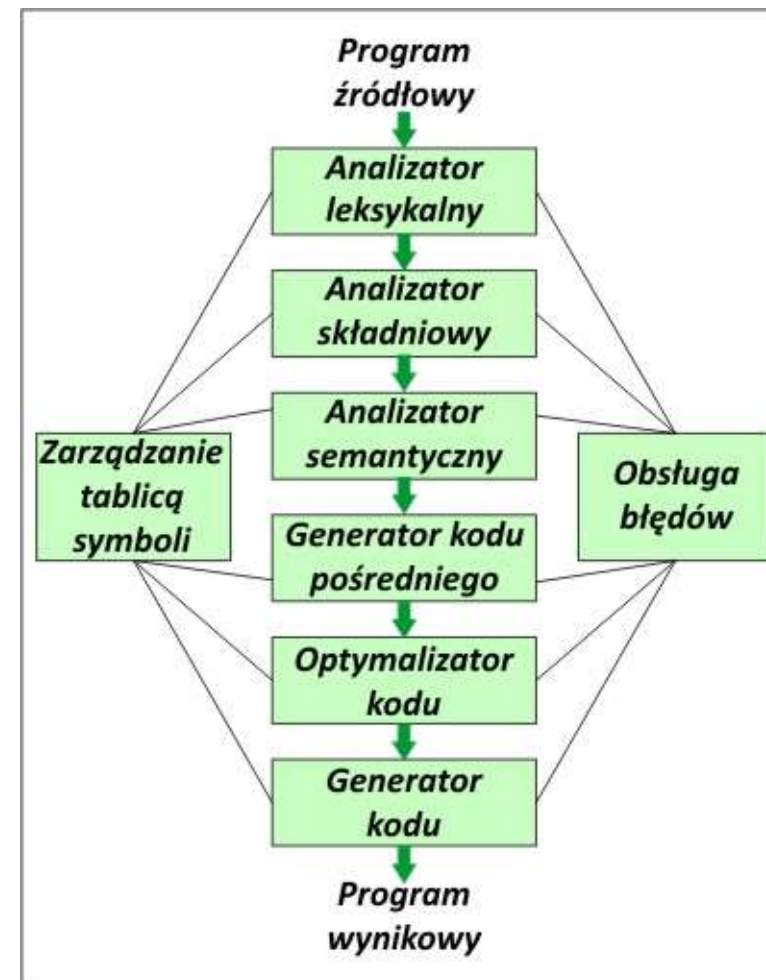
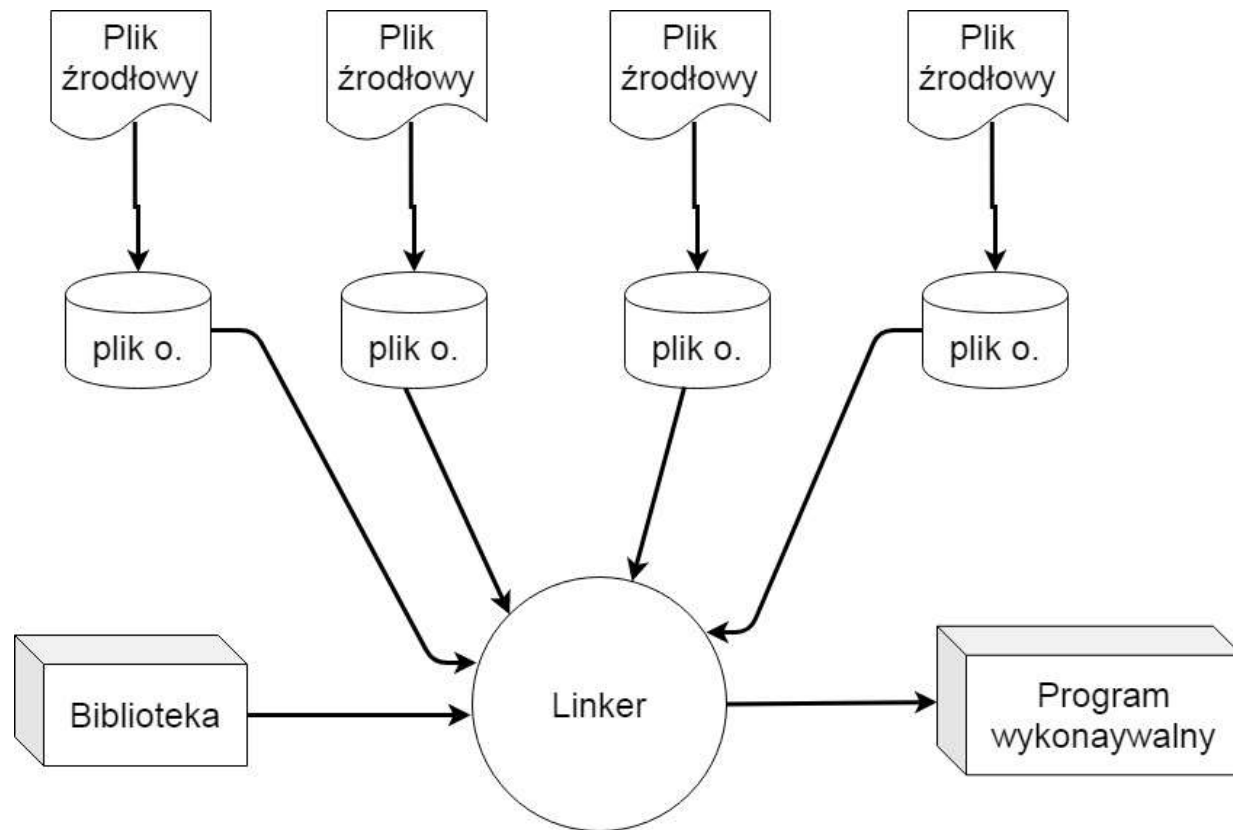
# Język programowania

- inform. narzędzie do formułowania programów dla komputerów; jest językiem formalnym, którego składnia określa zasady zapisu programów (w sposób jednoznaczny i łatwy do analizy), a semantyka przypisuje programom ich interpretację (określa efekty działania programu zapisanego w języku programowania).
- zbiór zasad określających, kiedy ciąg symboli tworzy program komputerowy oraz jakie obliczenia opisuje

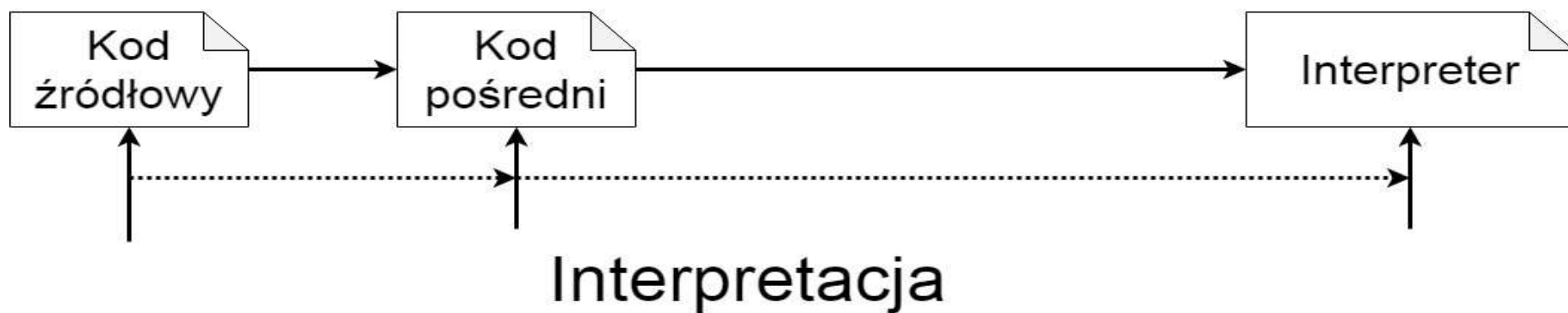
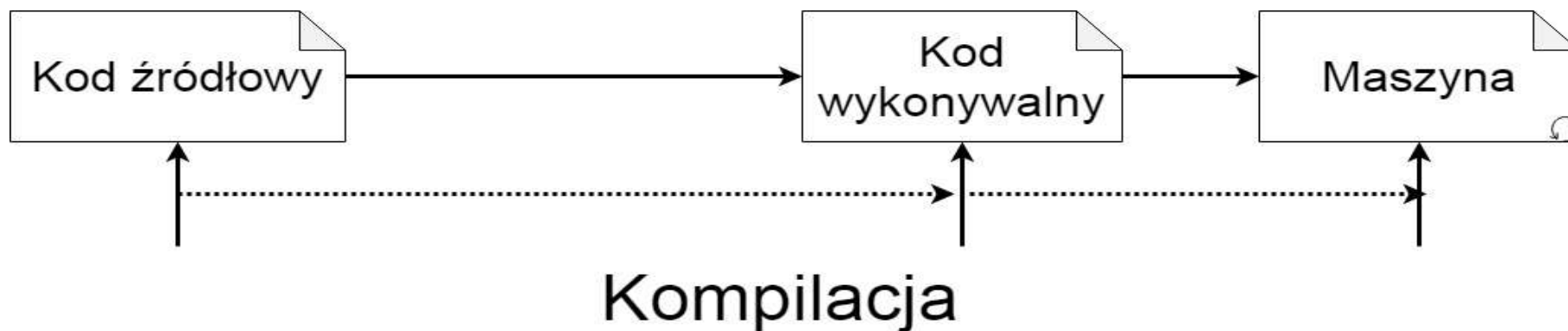
# ***Hierarchia języków programowania***



# Kompilacja



# Interpretacja vis kompilacja



\* Na podstawie <http://www.circuitstoday.com/compiler-vs-interpreters-an-overview-of-the-differences>

# Paradygmaty i języki programowania

- Języki imperatywne i deklaratywne
- Programowanie proceduralne i funkcyjne
- Programowanie obiektowe
- Programowanie uogólnione i refleksyjne
- Programowanie sterowane zdarzeniami
- Język skryptowy



# Imperatywne vs deklaratywne języki programowania

- Programowanie imperatywne – paradygmat programowania, który opisuje proces wykonywania jako sekwencję instrukcji zmieniających stan programu.
- Programy imperatywne składają się z ciągu komend do wykonania przez komputer
  - *jak to zrobić.*
- Przykłady
  - Fortran
  - Cobol
  - C++
  - Java
  - Python
- Programowanie deklaratywne
  - nie są imperatywne
- Programista opisuje warunki, jakie musi spełniać końcowe rozwiązanie
  - *co chcemy osiągnąć.*
- *Przykłady*
  - *Prolog*
  - *SQL*
  - *HTML*

# Programowanie proceduralne vis funkcyjne

- Programowanie proceduralne
- Paradygmat programowania
  - zalecający dzielenie kodu na procedury,
- Procedury
  - powinny pobierać i przekazywać wszystkie dane jako parametry wywołania.
  - nie powinny korzystać ze zmiennych globalnych
- Bazuje na wykorzystaniu
  - zmiennych
  - pętli
- Programowanie funkcyjne
- Odmiana programowania deklaratywnego
- Wartościowanie funkcji
- Raz zdefiniowana funkcja zwraca zawsze tę samą wartość (jak f. matemat.)
- Niezmienność danych
- Bazuje na wykorzystaniu
  - *Kompozycji funkcji*
  - *Rekurencji*

# Paradygmat obiektowy

- Świat widziany jako zbiór obiektów
- Dane ukryte wewnątrz obiektów
- Obiekty udostępniają interfejsy, które mogą być użyte do rozwiązania problemu
- Ułatwia powtórne wykorzystanie kodu

# Programowanie obiektowe

- Program obiektowy
  - Zbiór obiektów komunikujących się ze sobą w celu wykonania zadań
- Obiekt – element łączący
  - Stan (dane, pola)
  - Zachowanie (procedury, metody)
- Podstawowe założenia
  - Abstrakcja
  - Hermetyzacja
  - Polimorfizm
  - Dziedziczenie

# Programowanie uogólnione i refleksyjne

- Programowanie uogólnione
  - Paradygmat programowania
  - pozwala na pisanie kodu programu bez wcześniejszej znajomości typów danych, na których kod ten będzie pracował
  - ma bliski związek z metaprogramowaniem
    - ale nie wymaga od programisty generowania kodu w sposób jawny
- Programowanie refleksyjne
  - Paradygmat programowania
  - Mechanizm refleksji
  - proces, dzięki któremu program komputerowy
    - może być modyfikowany
    - w trakcie działania
    - w sposób
      - zależny od własnego kodu
      - od zachowania w trakcie wykonania

# Programowanie sterowane zdarzeniami

- Wiele drobnych programów obsługi zdarzeń
  - Wywoływanych w momencie wystąpienia zdarzenia
- Zarządca
  - Uruchamia programy obsługi zdarzeń
- Zdarzenie
  - Wywoływane przez
    - urządzenia wejścia – wyjścia (naciśnięcie klawisza, kliknięcie myszką),
    - procedury obsługi zdarzeń
  - Obsługiwane przez procedurę

# Język skryptowy

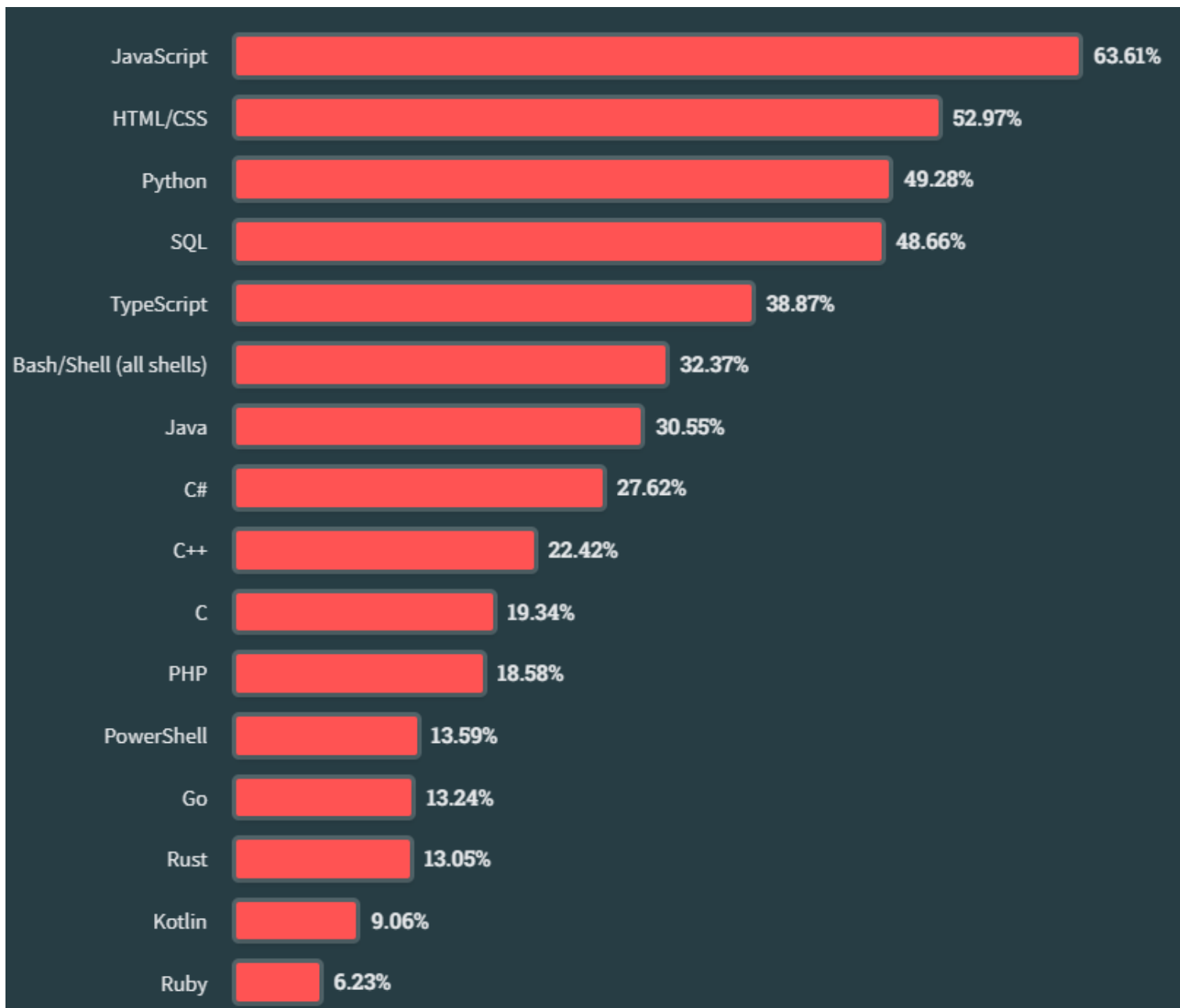
- Język programowania obsługujący skrypty
- Skrypty
  - programy napisane w językach skryptowych, najczęściej proceduralnych
  - przeznaczone do wykonywania w specjalnych środowiskach uruchomieniowych automatyzujących wykonywanie zadań,.
- Języki skryptowe
  - najczęściej języki interpretowane, najczęściej proceduralne
  - zaprojektowane do interakcji z użytkownikiem, często podejście obiektowe
  - Raczej niepełnoprawne języki programowania
  - Brak sprawdzania typów danych podczas kompilacji
  - Nawet bez zmiennych
  - Uproszczona składnia (na początek)
  - Małe projekty

Język	Wykorzystanie	Imperatywne	Obiektowe	Funkcjonalne	Proceduralne	Generyczne	Refleksyjne	Sterowane zdarzeniami
Bash	powłoka, skrypty	TAK			TAK			
C	Aplikacje, system, ogólnego przeznaczenia, operacje niskopoziomowe	TAK			TAK			
C#	Aplikacje, RAD, biznes, klient, ogólne, serwer, Sieć	TAK	TAK	TAK	TAK	TAK	TAK	TAK
C++	Aplikacje, system	TAK	TAK	TAK	TAK	TAK		
Fortran	Aplikacje, obliczenia numeryczne	TAK	TAK		TAK	TAK		
Groovy	Aplikacje, ogólne, skrypty, Sieć	TAK	TAK	TAK				
Java	Aplikacje, biznes, klient, ogólnel, mobilne development, serwer, Sieć	TAK	TAK	TAK	TAK	TAK	TAK	TAK
JavaScript	Klient, serwer, Sieć	TAK	TAK	TAK			TAK	
Lisp	Ogólne			TAK				
Objective-C	Aplikacje, ogólne, skrypty, Sieć	TAK	TAK				TAK	
Pascal	Aplikacje, edukacja	TAK			TAK			
Perl	Aplikacje, skrypty, przetwarzanie tekstu, Sieć	TAK	TAK	TAK	TAK	TAK	TAK	
PHP	Server-side, web Aplikacje, Sieć	TAK	TAK	TAK	TAK		TAK	
Python	Aplikacje, ogólne, Sieć, skrypty, sztuczna inteligencja, obliczenia naukowe	TAK	TAK	TAK	TAK		TAK	
R	Aplikacje, statystyka	TAK	TAK	TAK	TAK			
Ruby	Aplikacje, skrypt, Sieć	TAK	TAK	TAK			TAK	
Swift	Aplikacje, ogólne	TAK	TAK	TAK		TAK	TAK	TAK
Visual Basic	Aplikacje, szybkie prototypy, edukacja, biznes, ogólne, automatyzacja biura	TAK	TAK			TAK		TAK
Visual Basic .NET	Aplikacje, szybki prototyp, edukacja, Sieć, biznes, ogólne	TAK	TAK	TAK	TAK	TAK	TAK	TAK



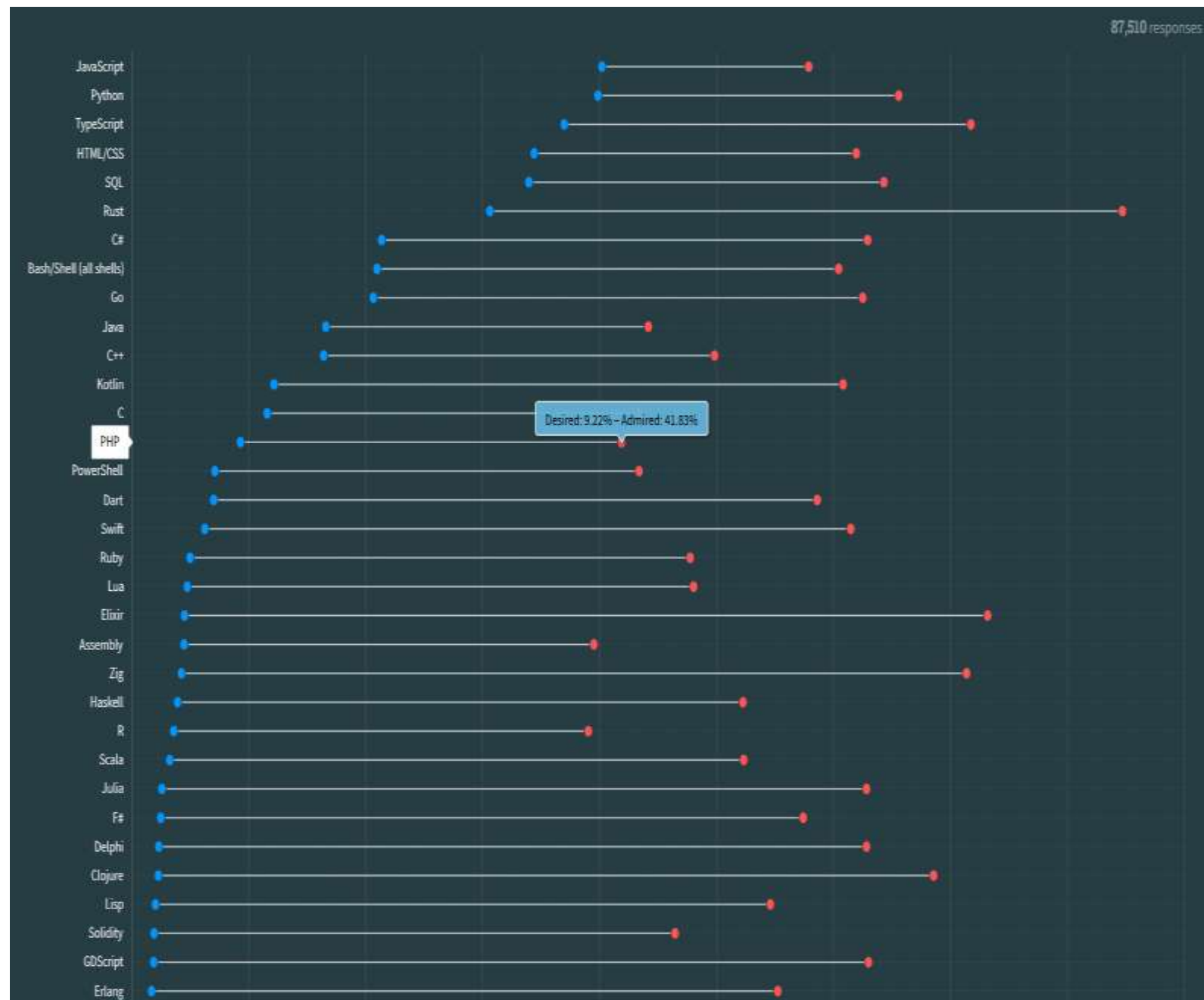
# Języki programowania

Wykorzystanie

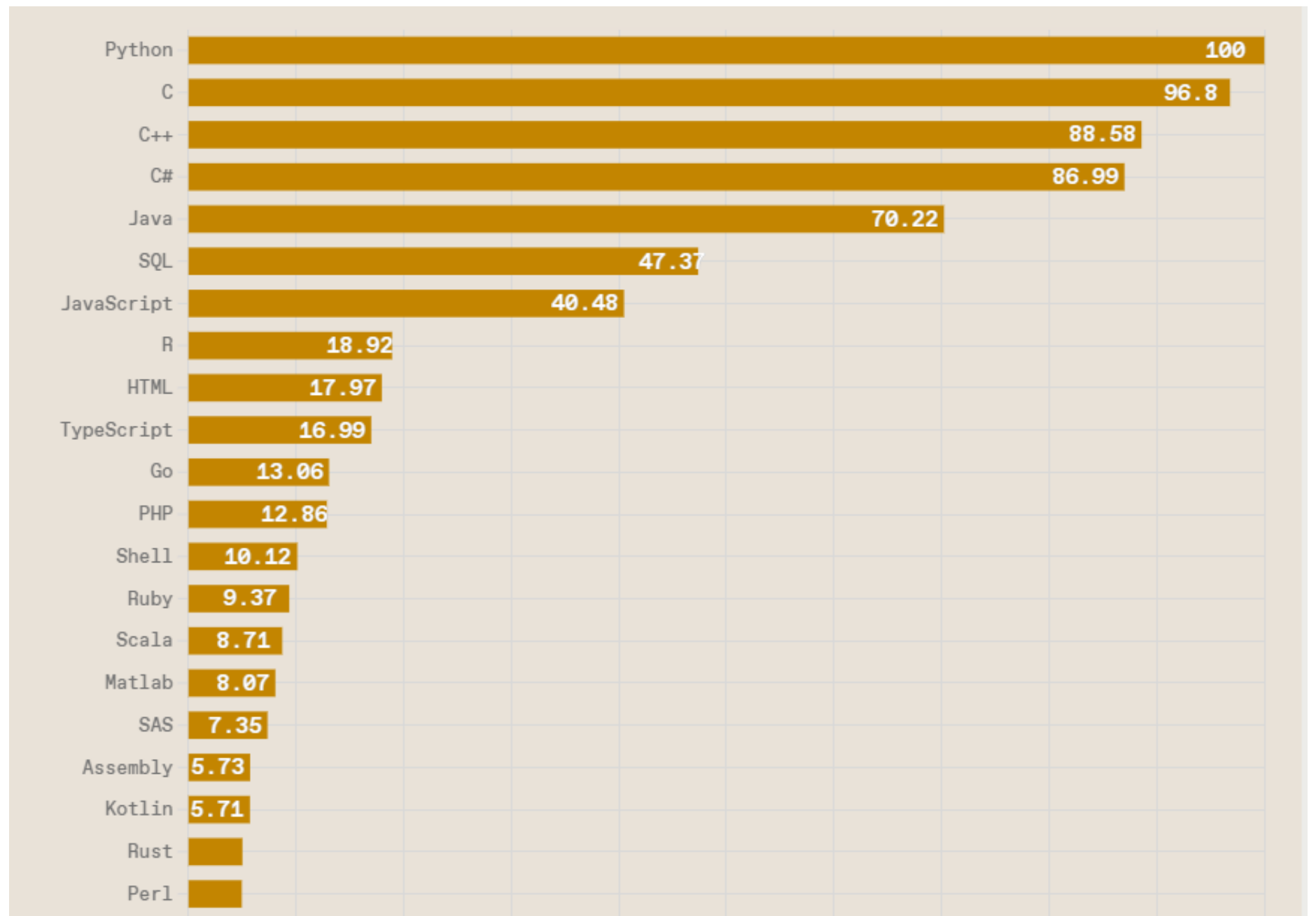


Najlepiej oceniane

# Języki programowania



# Ranking IEEE



Źródło: <https://spectrum.ieee.org/top-programming-languages-2022>

# Oferty pracy

