

## Zapis liczb zmiennoprzecinkowych (ang. floating-point numbers)

Liczby zmiennoprzecinkowe to sposób reprezentowania liczb rzeczywistych (tj. liczb, które mogą mieć części ułamkowe) w komputerach. Komputery przechowują liczby zmiennoprzecinkowe w formacie, który jest oparty na standardzie IEEE 754. Ten format składa się z trzech kluczowych elementów:

1. **Znak:** Określa, czy liczba jest dodatnia, czy ujemna.
2. **Cecha (eksponent):** Wskazuje przesunięcie przecinka w liczbie, czyli określa, o ile miejsc należy przesunąć przecinek, aby otrzymać poprawną wartość.
3. **Mantysa (część ułamkowa):** Przechowuje właściwą wartość liczby, znormalizowaną w taki sposób, że pierwsza cyfra (niebędąca zerem) znajduje się przed przecinkiem.

### Przykład zapisu liczby zmiennoprzecinkowej:

Liczbę zmiennoprzecinkową można zapisać w postaci naukowej, np.:

- $1.234 \times 10^3$  (co odpowiada liczbie 1234)

W przypadku komputerów taki zapis jest rozbity na:

- **Znak:** 0 (dodatnia liczba)
- **Cecha (eksponent):** 3 (określa potęgę, czyli przesunięcie przecinka)
- **Mantysa:** 1.234 (określa wartość liczby)

### Jak to działa w praktyce?

Liczby zmiennoprzecinkowe są przechowywane w sposób znormalizowany, co oznacza, że mantysa jest zawsze liczbą pomiędzy 1 a 2 (w systemie binarnym). Oznacza to, że liczba jest zawsze zapisywana jako:  $\text{Liczba zmiennoprzecinkowa} = \text{znak} \times \text{mantysa} \times 2^{\text{eksponent}}$

### Problemy wynikające z liczb zmiennoprzecinkowych:

1. **Precyzja:**
  - Mantysa ma skończoną liczbę cyfr (bitów), co oznacza, że liczby nie mogą być reprezentowane z nieskończoną dokładnością. To prowadzi do **błędów zaokrągleń**.
  - Na przykład, liczba 0.1 nie ma dokładnej reprezentacji w systemie binarnym, co prowadzi do drobnych różnic przy wykonywaniu operacji arytmetycznych:

```
print(0.1 + 0.2) # Wynik: 0.30000000000000004
```

2. **Zakres liczb:**

- Liczby zmiennoprzecinkowe mogą reprezentować zarówno bardzo małe, jak i bardzo duże liczby, ale tylko w określonym zakresie. Przekroczenie tego zakresu

proceeds to problems such as **przepiętnienie** (ang. **overflow**) or **podprzepiętnienie** (ang. **underflow**).

3. **Błędy w porównywaniu liczb zmiennoprzecinkowych:**

- Z powodu ograniczonej precyzji, liczby zmiennoprzecinkowe mogą nie być dokładnie równe, nawet jeśli intuicyjnie powinny. Należy unikać porównywania ich bezpośrednio, używając np. tolerancji przy porównaniach: