

# Pakiety obliczeniowe

**INP001029WL**

**rok akademicki 2021/22**

**semestr zimowy**

## Wykład 5

**Karol Tarnowski**

**[karol.tarnowski@pwr.edu.pl](mailto:karol.tarnowski@pwr.edu.pl)**

**L-1 p. 220**



# Plan prezentacji

- Obsługa wejścia z konsoli - funkcja `input`
- Obsługa wypisywania informacji na ekran - funkcja `fprintf`
- Obsługa wypisywania informacji do pliku - funkcje: `fprintf`, `fopen`, `fclose`
- Odczyt danych z pliku - funkcja `textscan`



# Funkcja `input`

- Do wprowadzania danych przez użytkownika można wykorzystać funkcję `input()`
- Domyślnie funkcja `input()` interpretuje podany napis jako kod Matlaba



# Funkcja input

```
%% przykład użycia funkcji input
```

```
clear  
clc
```

```
%% przykład wczytania danych liczbowych poleceniem input
```

```
x = input('Podaj liczbę x: ') % w konsoli podaj wartość liczbową  
disp("Podałś liczbę " + x + ".")
```

```
%% przykład wczytania wyrażenia wykorzystującego zmienne
```

```
y = input('Podaj wyrażenie arytmetyczne zawierające x (np. x + 2): ')  
disp("Wartość wyrażenia to " + y + ".")
```

```
%% przykład wczytania wyrażenia wykorzystującego zmienne
```

```
z = input('Podaj wywołanie funkcji Matlaba (np. magic(3)): ')  
disp(z)
```

```
%% funkcja input potrafi także wczytać napis (bez obliczania wartości wyrażenia)
```

```
u = input('Podaj wyrażenie arytmetyczne zawierające x (np. x + 2): ','s')  
disp("Podany napis to '" + u + "'.")
```



# Funkcja `input`

- Funkcja `input()` wywołana z dodatkowym argumentem `'s'` zwraca wpisany przez użytkownika łańcuch znakowy



# Funkcja `fprintf`

- Funkcja `fprintf` wypisuje sformatowane dane do pliku lub na ekran
- Do formatowania danych wykorzystuje specyfikatory formatu



# Funkcja fprintf

**%% wypisanie dwóch liczb na ekran**

```
a = pi; b = exp(1);  
fprintf("Wartość pi to w przybliżeniu %4.2f,\n\nnatomiast liczba e to w przybliżeniu %4.2f\n",a,b);
```

**%% wypisanie dwuelementowego wektora**

```
a = [pi, exp(1)]  
fprintf("Wartość pi to w przybliżeniu %4.2f,\n\nnatomiast liczba e to w przybliżeniu %4.2f\n",a);
```

**%% wypisanie macierzy polega na wielokrotnym zastosowaniu formatu do elementów macierzy**

```
X = [ 1 1 1;  
     2 4 8;  
     3 9 27]  
fprintf("%4.2f\t%4.2f\t%4.2f\n",X);
```

**%% wypisanie macierzy polega na wielokrotnym zastosowaniu formatu do elementów macierzy**

**% przez macierz przechodzimy po kolumnach**

```
X = [ 1 1 1;  
     2 4 8;  
     3 9 27]  
fprintf("%4.2f\t%4.2f\n",X);
```



# Funkcja `fprintf`

---

%% funkcja `fprintf` umożliwia różnorodne formatowanie danych

```
fprintf("%0+12.5f\n", pi)
```

```
fprintf("%1$0+12.5f\t%2$0+12.5f\t%1$0+12.5f\n", pi, exp(1))
```





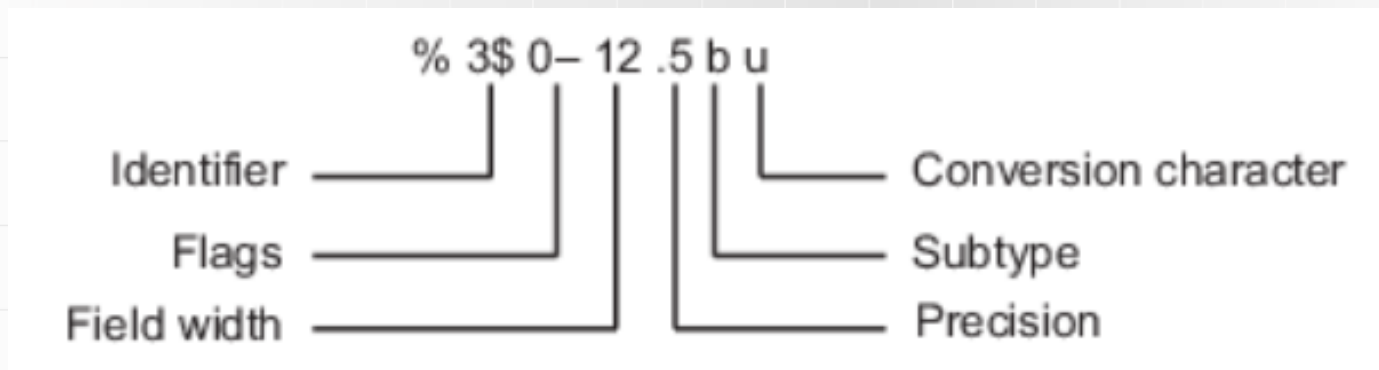
# Funkcja `fprintf`

- Specyfikator formatu rozpoczyna się znakiem %, a kończy symbolem konwersji

typ danych	symbol konwersji	opis
całkowity ze znakiem	<code>%d</code> , <code>%i</code>	liczba w systemie dziesiętnym
całkowity bez znaku	<code>%u</code>	liczba w systemie dziesiętnym
	<code>%o</code>	liczba w systemie ósemkowym
	<code>%x</code>	liczba w systemie szesnastkowym, małe litery a-f
	<code>%X</code>	liczba w systemie szesnastkowym, wielkie litery A-F
zmiennopozycyjny	<code>%f</code>	notacja zmiennopozycyjna
	<code>%e</code>	notacja wykładnicza (np. 3.141593e+00)
	<code>%E</code>	notacja wykładnicza (np. 3.141593E+00), wielka litera E
	<code>%g</code>	wybiera krótszy zapis z dwóch: <code>%i</code> , <code>%e</code> (bez zer wiodących)
	<code>%G</code>	wybiera krótszy zapis z dwóch: <code>%i</code> , <code>%E</code> (bez zer wiodących)
znakowy	<code>%c</code>	pojedynczy znak
	<code>%s</code>	wektor znakowy lub łańcuch znakowy

# Funkcja `fprintf`

- Pomiędzy znakiem % a symbolem konwersji można umieścić dodatkowe opcje



- identyfikator - wskazuje formatowany argument
- flaga - różne zachowanie w zależności od typu danej
- minimalna szerokość pola
- precyzja



# Funkcja fprintf

## %% identyfikator

```
fprintf('%3$s %2$s %1$s %2$s \n','A','B','C')
```

## %% szerokość pola

```
fprintf(":%12s:\n","napis")
```

```
fprintf(":%12f:\n",pi)
```

```
fprintf(":%*f:\n",12,pi)      %szerokość pola może być podawana jako argument
```

## %% precyzja

```
fprintf("%.4e\n",pi) %liczba cyfr po przecinku
```

```
fprintf("%.4f\n",pi) %liczba cyfr po przecinku
```

```
fprintf("%.4g\n",pi) %liczba cyfr znaczących
```



# Funkcja fprintf

%% flaga

% '-'

fprintf(":%-5.2f:\n", pi)

%wyrównanie do lewej

fprintf(":%-10s:\n", "napis")

%wyrównanie do lewej

% '+'

fprintf(":%+5.2f:\n", -pi)

%liczby zawsze ze znakiem

fprintf(":%+5.2f:\n", +pi)

%liczby zawsze ze znakiem

fprintf(":%+10s:\n", "napis")

%tekst wyrównany do prawej

% ' '

fprintf(":% 4.2f:\n", +pi)

%dodaj spację przed wartością

% '0'

fprintf(":%09.2f:\n", pi)

%dopełnij pole zerami



# Funkcja fprintf

% '#' modyfikuje wybrane formatowania numeryczne

```
fprintf("%x\n",16)
```

```
fprintf("%#x\n",16)
```

%dodaj 0x

```
fprintf("%o\n",16)
```

```
fprintf("%#o\n",16)
```

%dodaj 0

```
fprintf("%.0f\n",pi)
```

```
fprintf("%#.0f\n",pi)
```

%napisz kropkę nawet bez części ułamkowej

```
fprintf("%g\n",0.5)
```

```
fprintf("%#g\n",0.5)
```

%nie usuwaj kończących zer



# Funkcja `fprintf`

- Dodatkowe znaki specjalne

znak	Representation
apostrof	' '
znak procent	%%
ukośnik	\\
alarm	\a
backspace	\b
wysunięcie (form feed)	\f
nowa linia	\n
powrót	\r
tabulator (horizontal tab)	\t
tabulator pionowy (vertical tab)	\v
znak Unicode o podanym kodzie szesnastkowym N	<code>\xN % sprintf('\x5A') == 'Z'</code>
znak Unicode o podanym kodzie ósemkowym N	<code>\N % sprintf('\132') == 'Z'</code>



# Funkcja `fprintf`

- Pierwszym argumentem funkcji `fprintf` (poprzedzającym łańcuch sterujący) może być identyfikator pliku
- Domyślnym „plikiem” jest ekran
- Identyfikatorem pliku jest liczba zmiennopozycyjna
- Funkcja `fopen` otwiera plik i zwraca jego identyfikator
- Do zamknięcia pliku służy funkcja `fclose`



# Dostęp do pliku

%% przykład opracowany przez J. Olszewskiego

```
clear;  
clc;
```

% fprintf - zapisuje sformatowane dane do pliku tekstowego lub konsoli

```
x = 1 : 10;  
y = x + 10;  
A = [x(:), y(:)];
```

```
fid = fopen('plik_01.txt', 'wt');           % otwarcie pliku tekstowego do zapisu  
fprintf('Wynik wynosi: %d %d\n', A. ');    % wydruk w oknie konsoli  
fprintf(fid, 'Wynik wynosi: %d %d\n', A. '); % zapis do pliku  
fclose(fid);                               % zamknięcie pliku
```





# Dostęp do pliku

- Najczęściej pierwszym argumentem funkcji `fopen` jest nazwa pliku
- Nazwa pliku może zawierać ścieżkę dostępu do pliku
- Kolejnym argumentem jest sposób dostępu do pliku
- Funkcja zwraca liczbę - identyfikator pliku
- Wartości 0, 1, 2 są zastrzeżone dla standardowego wejścia, standardowego wyjścia, standardowego strumienia błędów
- Wartość -1 oznacza, że otwarcie pliku nie powiodło się



# Dostęp do pliku

```
%% przykład ze strony: https://www.mathworks.com/help/matlab/ref/fopen.html  
% pokazuje sprawdzenie, czy plik został otwarty poprawnie  
  
clear  
clc  
  
fileID = -1;  
errmsg = '';  
while fileID < 0  
    disp(errmsg);  
    filename = input('Open file: ', 's');  
    [fileID,errmsg] = fopen(filename);  
end  
  
% fclose(fileID)
```



# Funkcja `textscan`

- Funkcja `textscan` potrafi wczytać dane z pliku bądź z napisu (wektora znakowego lub łańcucha)
- Funkcja `textscan` wykorzystuje łańcuch kontrolny (analogicznie do funkcji `fprintf`)
- Wartości są odczytywane do tablicy komórkowej (`cell array`)



# Funkcja textscan

```
%% przykład opracowany przez J. Olszewskiego

clear
clc

% przed użyciem wywołaj 'fopen_fclose_example.m'
% aby wygenerować plik tekstowy 'plik_01.txt'

% fid = fopen('plik_01.txt', 'r'); % otwarcie pliku do odczytu
fid = fopen('plik_01.txt'); % otwarcie pliku do odczytu
% łańcuch formatujący jest identyczny jak w procedurze zapisu;
% oznacza to, że frazę "Wynik wynosi: " uznajemy za nieistotną,
% tzn. nie zostanie odczytana (bo i po co, skoro ją znamy ...)
% domyślny separator liczb to spacja ' '
c = textscan(fid, 'Wynik wynosi: %d %d');
% fstr = 'Wynik wynosi: %d %d';
% c = textscan(fid, fstr, 'CollectOutput', false);
fclose(fid); % zamknięcie pliku
% rozdzielanie danych
x = c{1, 1};
y = c{1, 2};
% wyświetlenie danych na ekranie konsoli
disp([x, y])
```



# Funkcja `textscan`

## Przykłady:

- odczyt liczb zmiennoprzecinkowych

```
openExample('matlab/ReadAStringExample')
```

- odczyt różnych typów danych

```
openExample('matlab/ReadDifferentTypesOfDataExample')
```

- pominięcie fragmentu tekstu

```
openExample('matlab/RemoveLiteralTextExample')
```

- pominięcie pozostałej części linii

```
openExample('matlab/SkipTheRemainderOfALineExample')
```



# Podsumowanie

- Obsługa wejścia z konsoli - funkcja `input`
- Obsługa wypisywania informacji na ekran - funkcja `fprintf`
- Obsługa wypisywania informacji do pliku - funkcje: `fprintf`, `fopen`, `fclose`
- Odczyt danych z pliku - funkcja `textscan`