



podstawowe komendy w scilabie



1. "clear" - Czyści konsolę.
2. "clc" - Czyści konsolę.
3. "a = 3" - Przypisuje wartość 3 do zmiennej 'a'.
4. "b = [1, 2, 3]" - Tworzy wektor z elementami 1, 2, 3 i przypisuje go do zmiennej 'b'.
5. "c = a + b" - Dodaje zmienną 'a' i wektor 'b' i przypisuje wynik do zmiennej 'c'.
6. "plot(c)" - Rysuje wykres z danymi z zmiennej 'c'.
7. "exec('nazwa_pliku.sce')" - Uruchamia plik z kodem Scilab o nazwie "nazwa_pliku.sce".
8. "help(funkcja)" - Wyświetla pomoc dla danej funkcji.
9. "quit" - Kończy działanie Scilab.



funkcje matematyczne



1. "sin(x)" - Sinus x (x jest podawany w radianach).
2. "cos(x)" - Cosinus x (x jest podawany w radianach).
3. "tan(x)" - Tangens x (x jest podawany w radianach).
4. "asin(x)" - Arcus sinus x (zwraca wartość w radianach).
5. "acos(x)" - Arcus cosinus x (zwraca wartość w radianach).
6. "atan(x)" - Arcus tangens x (zwraca wartość w radianach).
7. "exp(x)" - E^x .
8. "log(x)" - Logarytm naturalny x.
9. "log10(x)" - Logarytm dziesiętny x.
10. "sqrt(x)" - Pierwiastek kwadratowy z x.
11. "abs(x)" - Wartość bezwzględna x.
12. "round(x)" - Zwraca najbliższą liczbę całkowitą dla x.
13. "ceil(x)" - Zwraca największą liczbę całkowitą nie większą niż x.
14. "floor(x)" - Zwraca najmniejszą liczbę całkowitą nie mniejszą niż x.
15. "mod(x, y)" - Zwraca resztę z dzielenia x przez y.



definiowanie wektorów w scilab



W Scilab wektory są tworzone przy użyciu nawiasów kwadratowych []. Można wprowadzić pojedyncze elementy wektora po przecinku lub zdefiniować zakres elementów za pomocą operatorów ":".

Przykłady:

1. `"v = [1, 2, 3]"` - Tworzy wektor v z elementami 1, 2, 3.
2. `"u = [1:5]"` - Tworzy wektor u z elementami od 1 do 5.
3. `"w = [1:0.5:2]"` - Tworzy wektor w z elementami od 1 do 2 z krokiem 0.5.
4. `"x = [1, 2, 3; 4, 5, 6]"` - Tworzy dwuwymiarowy wektor x z elementami 1, 2, 3 w pierwszym rzędzie i 4, 5, 6 w drugim rzędzie.
5. `"y = [1:3; 4:6]"` - Tworzy dwuwymiarowy wektor y z elementami od 1 do 3 w pierwszym rzędzie i od 4 do 6 w drugim rzędzie.



definiowanie macierzy



W Scilab macierze są tworzone przy użyciu nawiasów kwadratowych [], a poszczególne wiersze są oddzielane średnikiem. Można wprowadzić pojedyncze elementy macierzy po przecinku lub zdefiniować zakres elementów za pomocą operatorów ":".

Przykłady:

1. `"A = [1, 2, 3; 4, 5, 6; 7, 8, 9]"` - Tworzy macierz A o rozmiarze 3x3 z elementami 1, 2, 3 w pierwszym rzędzie, 4, 5, 6 w drugim rzędzie i 7, 8, 9 w trzecim rzędzie.
2. `"B = [1:3; 4:6; 7:9]"` - Tworzy macierz B o rozmiarze 3x3 z elementami od 1 do 3 w pierwszym rzędzie, od 4 do 6 w drugim rzędzie i od 7 do 9 w trzecim rzędzie.
3. `"C = eye(3)"` - Tworzy macierz jednostkową o rozmiarze 3x3.
4. `"D = zeros(2,3)"` - Tworzy macierz o rozmiarze 2x3 złożoną z elementów 0.
5. `"E = ones(3,2)"` - Tworzy macierz o rozmiarze 3x2 złożoną z elementów 1.
6. `"F = rand(2,2)"` - Tworzy macierz o rozmiarze 2x2 z losowymi wartościami z przedziału [0,1].

Note that the function names such as "eye", "zeros", "ones" and "rand" are Scilab functions used to create certain type of matrices.



funkcje na macierzach



1. " A^2 " - Podnosi macierz A do kwadratu.
2. " A' " - Transponuje macierz A .
3. " $\text{inv}(A)$ " - Oblicza macierz odwrotną A^{-1} dla macierzy A .
4. " $\text{det}(A)$ " - Oblicza wyznacznik macierzy A .
5. " $\text{eig}(A)$ " - Oblicza wektory i wartości własne macierzy A .
6. " $\text{rank}(A)$ " - Oblicza rząd macierzy A .
7. " $\text{size}(A)$ " - Zwraca rozmiar macierzy A .
8. " $\text{length}(A)$ " - Zwraca liczbę elementów macierzy A .
9. " $\text{sum}(A)$ " - Oblicza sumę elementów macierzy A .
10. " $\text{mean}(A)$ " - Oblicza średnią elementów macierzy A .
11. " $\text{max}(A)$ " - Zwraca element o największej wartości w macierzy A .
12. " $\text{min}(A)$ " - Zwraca element o najmniejszej wartości w macierzy A .
13. " $A(i,j)$ " - Zwraca element macierzy A z indeksem (i,j) .
14. " $A(:,j)$ " - Zwraca j -ty wiersz macierzy A .
15. " $A(i,:)$ " - Zwraca i -tą kolumnę macierzy A .

Note that these are some of the basic matrix manipulation functions in Scilab, there are many more advanced matrix operation function available in Scilab library which can be used according to the requirement.



funkcje logiczne



1. " $x < y$ " - Zwraca prawdę jeśli x jest mniejsze od y .
2. " $x > y$ " - Zwraca prawdę jeśli x jest większe od y .
3. " $x \leq y$ " - Zwraca prawdę jeśli x jest mniejsze lub równe y .
4. " $x \geq y$ " - Zwraca prawdę jeśli x jest większe lub równe y .
5. " $x == y$ " - Zwraca prawdę jeśli x jest równe y .
6. " $x \neq y$ " - Zwraca prawdę jeśli x jest różne od y .
7. " $\text{and}(x, y)$ " - Zwraca prawdę jeśli x i y są prawdziwe.
8. " $\text{or}(x, y)$ " - Zwraca prawdę jeśli x lub y jest prawdziwe.
9. " $\text{not}(x)$ " - Zwraca przeciwność logiczną x .
10. " $x \& y$ " - Zwraca prawdę jeśli x i y są prawdziwe.
11. " $x | y$ " - Zwraca prawdę jeśli x lub y jest prawdziwe.
12. " $x \text{ xor } y$ " - Zwraca prawdę jeśli x lub y jest prawdziwe, ale nie oba.

Note that in Scilab, when using the logical operator ' $==$ ' for comparison it should be used in double, single ' $=$ ' is used for assignment.



Scilab posiada kilka operatorów do wykonywania pętli:

1. "for" - Pętla for służy do wykonywania określonej liczby iteracji. Przykład:

```
for i = 1:10
    disp(i)
end
```

Copy code

2. "while" - Pętla while wykonuje pętlę dopóki warunek jest spełniony. Przykład:

```
i = 1
while i <= 10
    disp(i)
    i = i + 1
end
```

Copy code

3. "do..while" - Pętla do..while jest podobna do pętli while, ale wykonuje się przynajmniej raz, nawet jeśli warunek nie jest spełniony. Przykład:

```
i = 1
do
    disp(i)
    i = i + 1
end while i <= 10
```

Copy code

4. "break" - Operator "break" pozwala na przerwanie działania pętli.
5. "continue" - Operator "continue" pozwala na przeskoczenie do następnej iteracji w pętli.

Note that the "break" and "continue" operators can be used inside of loops in order to control the flow of the loop.



rysowanie wykresów w scilab



Scilab posiada funkcję "plot" do rysowania wykresów. Funkcja ta przyjmuje jako argument wektor lub macierz danych, które chcemy narysować.

Przykłady:

1. "x = [1, 2, 3, 4, 5]; y = [5, 4, 3, 2, 1]; plot(x, y)" - Rysuje wykres funkcji $y = f(x)$ z danymi x i y.
2. "t = [0:0.1:10]; y = sin(t); plot(t, y)" - Rysuje wykres funkcji $y = \sin(t)$ z danymi t i y.
3. "z = randn(100,1); plot(z)" - Rysuje wykres losowych danych z.
4. "x = [1, 2, 3, 4, 5]; y = [5, 4, 3, 2, 1]; plot(x, y, '*')" - Rysuje wykres funkcji $y = f(x)$ z danymi x i y, z oznaczeniem punktów na wykresie jako gwiazdki.

Funkcja plot() posiada wiele opcji, które pozwalają na kontrolę wyglądu wykresu. Można np. zmienić kolor, rodzaj linii, dodawać etykiety osi, tytuł itp.

Note that Scilab also provides functions such as plot2d(), plot3d() to plot two-dimensional and three-dimensional plots respectively.



podaj opcje funkcji plot



Funkcja "plot" w Scilab posiada wiele opcji, które pozwalają na kontrolę wyglądu wykresu. Oto kilka przykładów:

1. `"plot(x, y, 'r')"` - Rysuje wykres z danymi x i y w kolorze czerwonym.
2. `"plot(x, y, '--')"` - Rysuje wykres z danymi x i y z przerywaną linią.
3. `"plot(x, y, 'o')"` - Rysuje wykres z danymi x i y z oznaczeniem punktów na wykresie jako kółka.
4. `"plot(x, y, '-.')"` - Rysuje wykres z danymi x i y z kropkową linią.
5. `"xlabel('Oś x')"` - Dodaje etykietę "Oś x" do osi x.
6. `"ylabel('Oś y')"` - Dodaje etykietę "Oś y" do osi y.
7. `"title('Mój wykres')"` - Dodaje tytuł "Mój wykres" do wykresu.
8. `"grid()"` - Dodaje siatkę do wykresu.
9. `"legend('y = f(x)')"` - Dodaje legendę "y = f(x)" do wykresu.
10. `"xlim([0, 10])"` - Określa zakres wartości na osi x od 0 do 10.
11. `"ylim([-5, 5])"` - Określa zakres wartości na osi y od -5 do 5.

These are just some examples of the options available for the `plot()` function, additional options can be found in the Scilab documentation.



jak zmieniać umiejscowienie legendy



Aby zmienić położenie legendy na wykresie w Scilab, należy użyć opcji "legend" z odpowiednim parametrem.

Przykłady:

1. "legend('y = f(x)', 'location', 'northwest')" - Umieszcza legendę w lewym górnym rogu wykresu.
2. "legend('y = f(x)', 'location', 'northeast')" - Umieszcza legendę w prawym górnym rogu wykresu.
3. "legend('y = f(x)', 'location', 'southwest')" - Umieszcza legendę w lewym dolnym rogu wykresu.
4. "legend('y = f(x)', 'location', 'southeast')" - Umieszcza legendę w prawym dolnym rogu wykresu.
5. "legend('y = f(x)', 'location', [0.2, 0.6])" - Umieszcza legendę w wyznaczonym punkcie (0.2, 0.6) na wykresie.

Dostępne opcje dla parametru 'location' to: 'northwest', 'north', 'northeast', 'west', 'center', 'east', 'southwest', 'south' and 'southeast' or coordinates [x, y].

Note that the option for the location of the legend can be used when the legend() function is called or can be passed to the plot() function.