Операторы отношения

>	больше
<	меньше
>=	больше или равно
<=	меньше или равно
==	равно
~=	не равно

```
>> 2>3
ans =     0
>> 2>1
ans =     1

>> A=rand(2,2)
A =
    0.8147    0.1270
    0.9058    0.9134

>> B=rand(2,2)
B =
    0.6324    0.2785
    0.0975    0.5469

>> A>B
ans =
     1     0
     1     1

>> A<0.5
ans =

     0     1
     0     0
```

Логические операторы

&	и(AND)
|	или(OR)
~	нет (NOT)

```
> ~1
ans =  0

>> x = [0 1 2 0];  y = [5 6 0 0]; x&y
ans =  0     1     0     0
>> x|y
ans =  1     1     1     0
```

Логические функции

Функция xor(a, b) реализует операцию ИСКЛЮЧИТЕЛЬНОЕ ИЛИ.

Таблица истинности для логического ИЛИ:

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

Таблица истинности для исключающего ИЛИ:

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

Функция all возвращает 1, если все элементы вектора истинны или отличны от нуля.
Функция any возвращает 1, если хотя бы один из элементов аргумента отличен от нуля;
Функция find определяет индексы элементов массива, которые удовлетворяют заданному логическому условию.

Оператор условия

```
if  логическое_выражение        инструкции
end

if логическое_выражение        инструкции
else
     инструкции
end

if логическое_выражение          инструкции
elseif логическое_выражение          инструкции
else
       инструкции
end
```

Оператор переключения

```
switch <выражение>
        % выражение - это обязательно скаляр или строка
    case  <значение1>
        инструкции
        % выполняются, если < выражение> =< значение1>
    case  <значение2>
        инструкции
        % выполняются, если  <выражение> = < значение2>
    ...
    otherwise
        инструкции
        % выполняются, если <выражение> не совпало ни с одним из
        %значений
    end
```

Оператор цикла с неопределенным числом операций

```
while выражение
  инструкции
end
```

break - оператор досрочного выхода из цикла
continue – оператор перехода к следующей итерации

Оператор цикла с определенным числом операций

```
for <переменная цикла> = <начальное значение>: <приращение>:<конечное
значение>
    инструкции
end
```

Использование массива в качестве переменной цикла.

```
for i = A
    инструкции
end
```

цикл выполняется столько раз, сколько столбцов в матрице А.
Для каждого шага i - это вектор, содержащий один из столбцов массива А.

Функции в Matlab

Анонимные функции

f = @(arglist)expression

```
f=@(x) 5*(exp(x)-1)-x.*exp(x);
```

Первичная функция

- функция имеет собственное имя
- имя функции и имя m-файла должны быть одинаковы
- переменные, определенные внутри функции являются локальными, то есть видны только внутри самой функции

```
function y=functionnumberone(x)
global qw;
y=2*x;
```

Подфункции

```
function y=functionnumberone(x)
y=2*functionnumbertwo(x);

function y=functionnumbertwo(x)
y=2*x;
```

В m-файлах могут быть описаны несколько функций. Это оформляется как две (или более) функций, записанных в одном файле. При вызове такого файла исполняется первая функции – ее имя должно совпадать с именем файла. Описание следующих функций локально – обычно они используются как вспомогательные для первой функции

Вложенные функции

Функция может быть описана непосредственно в теле другой функции. Такая функция называется вложенной. Вложенная функция, в свою очередь, может содержать другие вложенные функции.

function x = A(p1, p2)
...
B(p2)
  function y = B(p3)
  ...
  end
...
end