

1.

```
octave:1> sqrt(98)
ans = 9.8995
octave:2> 98^0.5
ans = 9.8995
octave:3> 98^1/2
ans = 49
[octave:4> 98^(1/2)
ans = 9.8995
```

2.

```
octave:5> y = 16
y = 16
octave:6> mod(sqrt(y), 1) == 0
ans = 1
```

3.

```
octave:9> iskeyword("if")
ans = 1
[octave:10> iskeyword("when")
[ans = 0
[octave:11> iskeyword("while")
[ans = 1
octave:12> iskeyword("how")
[ans = 0
octave:13> iskeyword("which")
ans = 0
octave:14> iskeyword("catch")
ans = 1
octave:15> iskeyword("try")
ans = 1
[octave:16> iskeyword("until")
ans = 1
[octave:17> iskeyword("spmd")
ans = 1
[octave:18> iskeyword("spot")
ans = 0
[octave:19> iskeyword("partfor")
ans = 0
[octave:20> iskeyword("for")
ans = 1
[octave:21> iskeyword("global")
ans = 1
[octave:22> iskeyword("else")
ans = 1
[octave:23> iskeyword("e")
ans = 0
[octave:24> iskeyword("pi")
ans = 0
[octave:25> iskeyword("__FINE__")
ans = 0
[octave:26> iskeyword("__LINE__")
ans = 1
[octave:27> iskeyword("break")
ans = 1
[octave:28> iskeyword("broke")
ans = 0
[octave:29> iskeyword("function")
ans = 1
```

4.

```
octave:30> who
Variables visible from the current scope:

ans  y

octave:31> whos
Variables visible from the current scope:

variables in scope: top scope

  Attr      Name      Size      Bytes  Class
  ====      ==
      ans      1x1         1  logical
      y       1x1         8   double
```

who: Lists the names of variables currently in the workspace.

whos: Provides detailed information about each variable, including size, type, and memory usage.

5.

Returns [1 1] for scalars.

```
octave:32> var1 = 3
var1 = 3
octave:33> size(var1)
ans =

     1     1
```

Returns [m n] for an $m \times n$ matrix.

```
[octave:36> var2 = [3,3;2,2;5,7]
var2 =

     3     3
     2     2
     5     7

[octave:37> size(var2)
ans =

     3     2
```

6.

```
octave:38> Inf/3
[ans = Inf
octave:39> Inf/0
ans = Inf
octave:40> Inf/-3
ans = -Inf
[octave:41> Inf/3i
ans = NaN - Inf i
```

Division by 0 returns Inf in Octave. Division involving imaginary numbers leads to undefined behavior (NaN).

7.

```
octave:42> Inf/Inf  
ans = NaN
```

8.

```
octave:48> Inf^2  
ans = Inf
```

9.

```
octave:44> sqrt(Inf)  
ans = Inf
```

10.

```
octave:47> Inf + Inf  
ans = Inf
```

11.

```
octave:45> Inf*Inf  
ans = Inf
```

12.

```
octave:49> Inf - Inf  
ans = NaN
```

13.

```
octave:50> sqrt(-Inf)  
ans = 0 + Inf i
```

14.

```
octave:51> Inf^Inf  
ans = Inf
```

15.

```
octave:52> Inf/i  
ans = NaN - Infi
```

16.

Typing `clc` clears the command window but does not affect the workspace variables.

```
octave:54> 
```

17.

Typing `clear` removes all variables from the workspace. To clear a specific variable, use `clear variable_name`

```
[octave:54> clear  
[octave:55> who  
octave:56> 
```

18.

format short: Displays numbers with 4 decimal places.

format long: Displays numbers with 15 decimal places.

```
[octave:61> format short
[octave:62> pi+e
ans = 5.8599
[octave:63> format long
[octave:64> pi+e
ans = 5.859874482048838
```

19.

```
[octave:56> X = [2,3; 4,1]
X =

     2     3
     4     1

[octave:57> A = [3,4,10; 70,1,30]
A =

     3     4    10
    70     1    30

[octave:58> Z = [2,3; 50,49; 0,1]
Z =

     2     3
    50    49
     0     1

[octave:59> Y = [1,0,0; 0,1,0; 0,0,1]
Y =

     1     0     0
     0     1     0
     0     0     1
```

20.

```
octave:60> X', A', Z', Y'
ans =

     2     4
     3     1

ans =

     3    70
     4     1
    10    30

ans =

     2    50     0
     3    49     1

ans =

     1     0     0
     0     1     0
     0     0     1
```

21.

```
octave:65> fliplr(X), fliplr(A), fliplr(Z), fliplr(Y)
ans =

     3     2
     1     4

ans =

    10     4     3
    30     1    70

ans =

     3     2
    49    50
     1     0

ans =

     0     0     1
     0     1     0
     1     0     0
```


22.

```
octave:66> flipud(X), flipud(A), flipud(Z), flipud(Y)
ans =

     4     1
     2     3

ans =

    70     1    30
     3     4    10

ans =

     0     1
    50    49
     2     3

ans =

     0     0     1
     0     1     0
     1     0     0
```

23.

```
octave:67> x = 7;
y = x^2 - 6*x + 5
y = 12
```

24.

```
octave:69> linspace(5, 150, 11)
ans =

Columns 1 through 4:

    5.000000000000000e+00    1.950000000000000e+01    3.400000000000000e+01    4.850000000000000e+01

Columns 5 through 8:

    6.300000000000000e+01    7.750000000000000e+01    9.200000000000000e+01    1.065000000000000e+02

Columns 9 through 11:

    1.210000000000000e+02    1.355000000000000e+02    1.500000000000000e+02
```

25.

```
octave:70> logspace(log10(10), log10(1000), 7)
ans =

Columns 1 through 4:

    1.000000000000000e+01    2.154434690031883e+01    4.641588833612777e+01    1.000000000000000e+02

Columns 5 through 7:

    2.154434690031885e+02    4.641588833612777e+02    1.000000000000000e+03
```

26.

Typing rand(3,4) multiple times generates different random 3x4 matrices.

```
octave:71> rand(3,4)
ans =

    8.286478970026836e-01    2.355666367620324e-02    2.119169322725361e-01    6.762577754320931e-02
    1.747996414339855e-01    7.685929956553248e-01    8.164340203585055e-01    4.011508322917033e-01
    4.490510505470446e-02    4.916548466649906e-03    1.549005317634722e-02    8.919391776355519e-01

octave:72> rand(3,4)
ans =

    1.786659193967575e-01    1.291588708000582e-01    5.570197392295395e-02    4.597310726468391e-01
    6.848614374221249e-01    7.334469176507336e-01    4.727391814468030e-01    3.940098537645358e-01
    5.192569567138474e-01    5.650987961946521e-01    4.455819314649851e-01    1.967677529619682e-01

octave:73> rand(3,4)
ans =

    7.762185079337302e-01    5.228098040337840e-01    1.840091451137379e-01    5.533261456554833e-01
    2.316788762429357e-01    3.490202923160286e-01    8.469825750811500e-01    5.018353316473478e-01
    5.115615455565097e-01    7.466040269961921e-01    2.523181838449018e-03    6.382852143573081e-02

octave:74> rand(3,4)
ans =

    8.180624731062351e-01    8.736892874922005e-01    5.392398752813266e-02    2.190959281511141e-01
    9.827155092675915e-01    7.560363469533782e-02    5.534596142546525e-01    3.200614801439338e-01
    6.534153909200839e-01    6.275679168826516e-01    1.748494738980282e-01    2.142410099582945e-01

octave:75> rand(3,4)
ans =

    6.651869488884961e-01    7.225656083909171e-01    1.721280249215071e-01    9.566546997463607e-01
    5.055468510532725e-01    6.176109812099907e-01    5.435758391447876e-01    1.176717313627973e-02
    1.766621758482730e-01    2.769957428438760e-01    8.677319090122483e-01    9.969699953843632e-01
```

27.

```
octave:76> A = rand(3, 4)
A =

    7.080049663065336e-01    4.039026508946556e-01    7.457550534490653e-02    9.185246954366817e-01
    3.440106589725921e-01    3.667156738061936e-01    4.075535145581873e-01    5.125217025277434e-01
    5.966076236867102e-01    1.991090451877243e-01    4.752186369771108e-01    9.011382473996887e-02

octave:77> A
A =

    7.080049663065336e-01    4.039026508946556e-01    7.457550534490653e-02    9.185246954366817e-01
    3.440106589725921e-01    3.667156738061936e-01    4.075535145581873e-01    5.125217025277434e-01
    5.966076236867102e-01    1.991090451877243e-01    4.752186369771108e-01    9.011382473996887e-02
```

28.

```
[octave:81> x = 0
x = 0
[octave:82> x = x + 25
x = 25
[octave:83> x = x + 25
x = 50
[octave:84> x = x + 25
x = 75
[octave:85> x = x + 25
x = 100
[octave:86> x = x + 25
x = 125
[octave:87> x = x + 25
x = 150
[octave:88> x = x + 25
x = 175
```

29.

```
[octave:120> a = 2
a = 2
[octave:121> z = 8
z = 8
[octave:122> a = 2;
z = 8;
for i = 1:14
    a = a + 1;
    z = z + 2;
end
[octave:125> a
a = 16
[octave:126> z
z = 36
[octave:127> a = 1;
z = 6;
for i = 1:14
    a = a + 1;
    z = z + 2;
end
[octave:130> a
a = 15
[octave:131> z
z = 34
[octave:132> █
```

30.

```
octave:106> inv([2, 3; 4, 5])
ans =

-2.5000000000000000    1.5000000000000000
 2.0000000000000000   -1.0000000000000000
```

31.

```
octave:107> y = eye(12);
[octave:108> y
y =

Diagonal Matrix

  1   0   0   0   0   0   0   0   0   0   0   0
  0   1   0   0   0   0   0   0   0   0   0   0
  0   0   1   0   0   0   0   0   0   0   0   0
  0   0   0   1   0   0   0   0   0   0   0   0
  0   0   0   0   1   0   0   0   0   0   0   0
  0   0   0   0   0   1   0   0   0   0   0   0
  0   0   0   0   0   0   1   0   0   0   0   0
  0   0   0   0   0   0   0   1   0   0   0   0
  0   0   0   0   0   0   0   0   1   0   0   0
  0   0   0   0   0   0   0   0   0   1   0   0
  0   0   0   0   0   0   0   0   0   0   1   0
  0   0   0   0   0   0   0   0   0   0   0   1
```

32.

```
[octave:109> det_y = det(y)
det_y = 1
```

33.

```
octave:110> az = 2:7:98
az =

  2   9  16  23  30  37  44  51  58  65  72  79  86  93
```

34.

```
[octave:116> k = [2, 3, 7; 8, 3, 4]
k =
```

```
    2    3    7
    8    3    4
```

```
[octave:117> m = rot90(k, 3)
m =
```

```
    8    2
    3    3
    4    7
```

35.

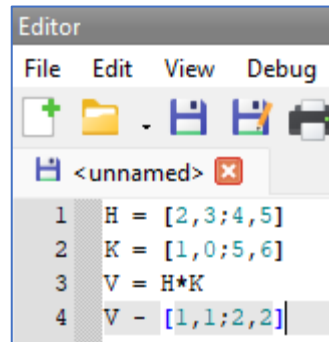
```
[octave:118> w = [1, 0, 1; 2, 3, 5]
w =
```

```
    1    0    1
    2    3    5
```

```
[octave:119> reshape(w, [], 1)
ans =
```

```
    1
    2
    0
    3
    1
    5
```

36.



```

1 H = [2,3;4,5]
2 K = [1,0;5,6]
3 V = H*K
4 V = [1,1;2,2] + V

```

Name	Class	Dimension	Value	Attribute
H	double	2x2	[2, 3; 4, 5]	
K	double	2x2	[1, 0; 5, 6]	
V	double	2x2	[17, 18; 29, 30]	