1. Exact versus approximate arithmetics, precision

```
2 + 2         (* exact *)
2 + 2.        (* approximate *)
N[2 + 2]
2 + 2 // N
```

```
1 / 3 + 2 / 7          (* exact - rational numbers *)
% + 0.0         (* approximate - numerical value *)
1 / 3. + 2 / 7
1 / 3 + 2 / 7 // N
```

```
Pi          (* exact *)
Pi // N     (* approximate *)
N[Pi]
N[Pi, 300]
```

2. Math functions

```
Sin[1]    (* exact *)
Sin[1.]   (* approximate *)
Sin[Pi]   (* exact *)
Sin[Pi / 3]
```

```
Sin[Pi / 5]  (* exact *)
Sin[Pi / 13]
```

```
Sqrt[16]     (* exact *)
Sqrt[2]      (* exact *)
Sqrt[2] // N  (* approximate *)
Sqrt[-1]      (* imaginary *)
```

```
? Sin    (* getting help - follow the arrow at the help end (just press it :) *)
```

Sin[z] gives the sine of z. ≫

3. Recalling the results

```
%   (* the last *)
%%%  (* the last but two *)
%
%5   (* with the label Out[5] *)
```

4. Variables

```
x = 5     (* set immediately *)
y := 5    (* set delayed *)
x^2       (* no difference *)
y^2
```

```
a = 4     (* let's define *)
x = 5 + a
y := 5 + a
```

```
x         (* still no difference *)
y
```

```
a = 7;    (* and now? *)
x
y
```

5. Lists

```
{3, 5, 1}     (* Compose with braces {}, take parts with double brackets[[]] *)
{3, 5, 1}^2 + 1
{6, 7, 8} - {3.5, 3, 2.5}
{6, 7, 8, 9} - {3.5, 3, 2.5}
```

```
v = {2, 4, 3.1}
v / (v - 1)  (* element by element *)
```

6. Symbolic computations

```
3 + 62 - 1
3 x - x + 6 - 4
```

```
Clear[x, y] (* remove the value of objects -
 objects with no value are printed in blue *)
3 x - x + 6 - 4
```

```
-1 + 2 x + x^3
x^2 + x - 4 x^2
x y + 2 x^2 y + y^2 x^2 - 2 y x
```

7. Setting values

```
1 + 2 x /. x → 3
(* the arrow → is the equivalent of Rule[] function; /. means "apply rule" *)
1 + x + x^2 /. x → 2 - y
```

```
rule = x → 3 + y  (* rule as a variable value *)
x^2 - 9 /. rule
```

```
(x + y)(x - y)^2 /. {x → 3, y → 1 - a}  (* list of rules *)
```

8. Vectors and matrices

```
Clear[a, b, c, d]
m = {{a, b}, {c, d}}  (* define a matrix *)
m[[1]]              (* take a row *)
m[[1, 2]]           (* take an element *)
m[[All, 1]]         (* take a column *)
```

```
v = {x, y}       (* a vector *)
p v + q          (* combine scalars and vectors *)
v + {xp, yp} + {xpp, ypp}
```

```
{x, y}.{xp, yp} (* multiply vectors *)
m.v        (* multiply a matrix by a vector *)
m.m        (* multiply matrices *)
% // MatrixForm
```

```
s = Table[i + j, {i, 3}, {j, 3}] (* generate a matrix *)
s // MatrixForm
```

```
Array[f, 4] // MatrixForm      (* generate a vector *)
Array[p, {3, 2}] // MatrixForm (* generate a matrix *)
```

```
IdentityMatrix[3]           (* generate a matrix *)
DiagonalMatrix[{a, b, c}]   (* generate a matrix *)
```

```
m // MatrixForm
Det[m]
Transpose[m] // MatrixForm
m // Transpose // MatrixForm
```

```
Inverse[m] // MatrixForm
%.m // MatrixForm
```

```
% // Simplify // MatrixForm
```

9. Algebraic equations

```
2 + 2 == 4        (* true *)
```

```
x = 4; x == 6     (* false *)
```

```
x =.; x == 6      (* unknown *)
```

```
% /. x → 4        (* now known *)
```

```
x^2 + 2 x - 7 == 0 (* unknown *)
```

```
eqn = %
```

```
eqn
```

```
Solve[eqn, x]
```

```
% // N
```

```
result = %;
x /. result
x^2 + 3 x /. result
```

10. Defining functions

```
f[x_] := x^2
f[3]
f[a + 2]
```

11. Immediate versus delayed

```
a = 7;
gi[x_] = x^2 + a;
gd[x_] := x^2 + a
```

```
? gi
```

```
? gd
```

```
a = 11;
gi[z]
gd[z]
```

12. Conditionals

```
If[7 > 8, Print[x], Print[y]]
```

```
f[x_] := If[x > 0, 1, -1]  (* step function *)
```

```
f /@ Table[i, {i, -3, 3}]
```

```
g[x_] := 1 /; x > 0     (* alternative definition of above: the positive part *)
g[x_] := -1 /; x <= 0   (* the negative part *)
   (* one of the conditions can (should) be ommited *)
```

```
Plot[g[x], {x, -5, 5}]
```

13. Loops and looping

```
For[i = 0, i < 4, i++, Print[i^2]]
```

```
Do[Print[i^2], {i, 4}]
```

14. Basic 2D plotting

```
Plot[Sin[x], {x, 0, 6 Pi}]
Plot[{Sin[x], Sin[2 x], Sin[3 x]}, {x, 0, 2 Pi}, PlotLegends → "Expressions"]
```

```
fig1 = Plot[2 Sin[x] + x, {x, 0, 15}, Filling → Bottom]
fig2 = Plot[{Sin[x] + x / 2, Sin[x] + x}, {x, 0, 15}, Filling → {1 → {2}}]
Plot[Evaluate[Table[BesselJ[n, x], {n, 4}]], {x, 0, 10}, Filling → Axis]
```

```
Show[fig1, fig2, AxesLabel → {x, y}]
```

```
ParametricPlot[{2 Sin[t], Cos[t]}, {t, 0, 2 Pi}]
PolarPlot[Cos[t]^2, {t, 0, 2 Pi}]
ContourPlot[x^3 + x y^2 + y == 0, {x, -1, 1}, {y, -10, 10}]
```

15. Basic 3D plotting

```
Plot3D[Sin[x y], {x, -Pi, Pi}, {y, -Pi, Pi}]
Plot3D[{Sin[x y], Sin[2 x] Sin[3 y]}, {x, -Pi, Pi}, {y, -Pi, Pi}]
```

```
ParametricPlot3D[{r Cos[t], r Sin[t], Log[r^(1 / 2)]}, {r, 0.1, 1}, {t, 0, 2 Pi}]
ParametricPlot3D[{5 Cos[u], 5 Sin[u], u + Sin[u]}, {u, -2 Pi, 2 Pi}]
```

```
fig3 = SphericalPlot3D[1, {t, 0, Pi}, {p, 0, 2 Pi}, PlotStyle → Opacity[0.6]]
fig4 = ContourPlot3D[x^2 + y^2 - z^2 == 0, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]
RevolutionPlot3D[{{2 + Cos[t], Pi + 1 / 2 Sin[t]}, {Sin[t], t}}, {t, 0, 2 Pi}, {θ, 0, 2 Pi}]
```

```
Show[fig3, fig4]
```

16. Data plotting

```
sdata = Table[1. Sin[2 i], {i, 0, 2 Pi, Pi / 12}]
ListPlot[sdata]
ListLinePlot[sdata]
```

```
Manipulate[ListPolarPlot[Table[Cos[i / n]^m, {i, -k Pi, t Pi, Pi / d}],
  PlotStyle → {PointSize[s], Red}, ImageSize → {400, 400},
  Axes → None, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}}],
 {{n, 0.32}, 0.1, 5, 0.01}, {m, 1, 4, 1}, {{t, 4.14}, 2, 6}, {{d, 100}, 10, 200, 10},
 {{k, 4}, -1.5, 8}, {{s, 0.006}, 0.003, 0.01, 0.001}]
```

```
data = Table[Sin[j^2 + i^2], {i, 0, Pi, Pi / 30}, {j, 0, Pi, Pi / 30}];
ListPlot[{data[[1]], data[[19]]}]
ListPointPlot3D[data]
ListPlot3D[data]
```

17. Other plotting

```
ContourPlot[Cos[x] + Cos[y] == 1 / 2, {x, 0, 4 Pi}, {y, 0, 4 Pi}]
ContourPlot[Cos[x] + Cos[y], {x, 0, 4 Pi}, {y, 0, 4 Pi}]
```

```
ContourPlot3D[x^3 + y^2 - z^2 == 0, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]
ContourPlot3D[x^3 + y^2 - z^2, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]
```

```
VectorPlot[{y, -x}, {x, -3, 3}, {y, -3, 3}]
```

```
VectorPlot[Evaluate[D[Sin[x y], {{x, y}}]], {x, 0, 1}, {y, 0, 1}]
```

```
VectorPlot3D[{x, y, z}, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```

```
BarChart[{{1, 2, 3}, {1, 3, 2}, {5, 2}}, ChartLegends → {"a", "b", "c"}]
BarChart[Range[6], ChartElementFunction → "GlassRectangle", ChartStyle → "Pastel"]
BarChart3D[Range[6], ChartElementFunction → "ProfileCube", ChartStyle → "Pastel"]
```

```
PieChart[{1, 2, 3, 4}]
PieChart[Range[7], SectorOrigin → {Automatic, 1}, ChartStyle → 54,
 ChartBaseStyle → EdgeForm[], ChartElementFunction → "GlassSector"]
PieChart[Table[Style[1, Hue[h, s, 1]], {s, 0, 1, .1}, {h, 0, 1, .1}]]
```