Introduction to Wolfram Language

Functions

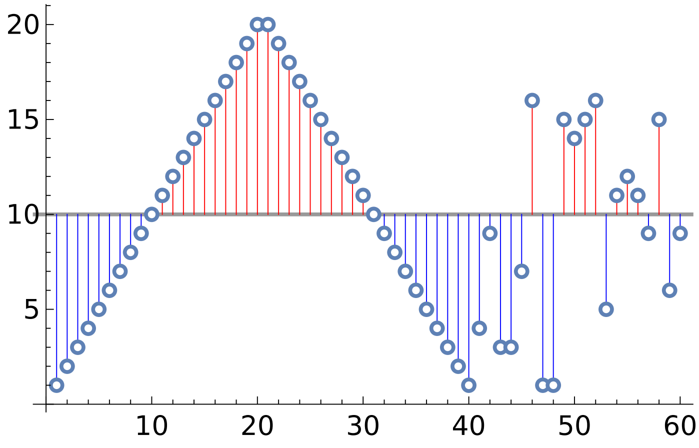
All function use square brackets, and have names that starts with capital letters.

```
In[2]:=  Times[2, Plus[3, 4]]

Out[2]=  14

In[3]:=  Times[5, 4, 3, 2]

Out[3]=  120

In[4]:=  Times[Plus[8, 7], Plus[9, 2]]

Out[4]=  165

In[5]:=  Apply[Times, Apply[Plus, {{8, 7}, {9, 2}}, 2]]

Out[5]=  165

In[6]:=  Times @@ Plus @@@ {{8, 7}, {9, 2}}

Out[6]=  165
```

Lists

```
In[7]:=  ListPlot[Join[Range[20], Reverse[Range[20]], RandomInteger[20, 20]],
          Filling → 10, PlotTheme → {"OpenMarkersThick", "LargeLabels"},
          FillingStyle → {Blue, Red}, GridLines → {None, {{10, {Black, Thick}}}}]
```

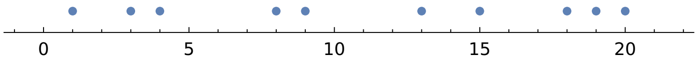
Out[7]=



In[8]:=

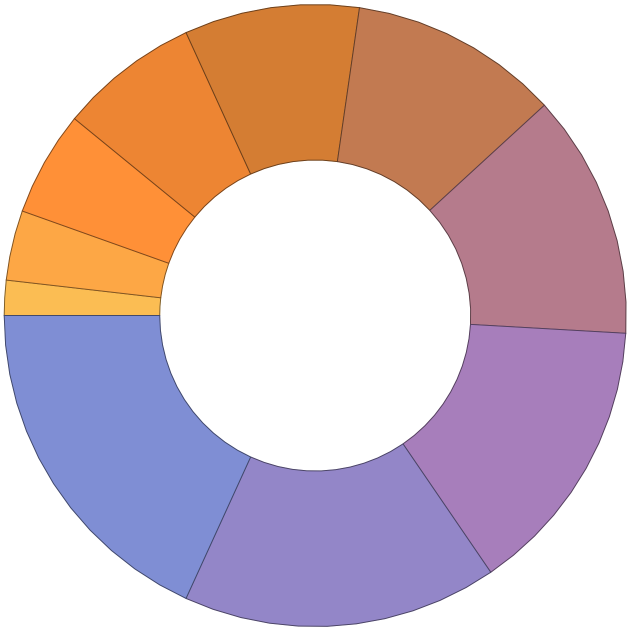
```
In[9]:=  NumberLinePlot[RandomSample[Range[20], 10]]
```

Out[9]=



```
In[10]:= PieChart[Range[10], SectorOrigin → {Automatic, 1}, PerformanceGoal → "Quality"]
```

Out[10]=



```
In[11]:= Take[Range[10], 5]

Out[11]= {1, 2, 3, 4, 5}

In[12]:= Drop[Range[10], 5]

Out[12]= {6, 7, 8, 9, 10}

In[13]:= Rest[Range[10]]

Out[13]= {2, 3, 4, 5, 6, 7, 8, 9, 10}

In[14]:= Most[Range[10]]

Out[14]= {1, 2, 3, 4, 5, 6, 7, 8, 9}

In[15]:= (* use table to make list*)
         Table[{1, 2}, 5]

Out[15]= {{1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 2}}

In[16]:= Table[n - 1, {n, 1, 10}]

Out[16]= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
In[17]:= Table[Range[n], {n, 1, 10}] // MatrixForm
Out[17]//MatrixForm=
          ⎛             {1}               ⎞
          ⎜           {1, 2}              ⎟
          ⎜         {1, 2, 3}             ⎟
          ⎜       {1, 2, 3, 4}            ⎟
          ⎜     {1, 2, 3, 4, 5}           ⎟
          ⎜   {1, 2, 3, 4, 5, 6}          ⎟
          ⎜  {1, 2, 3, 4, 5, 6, 7}        ⎟
          ⎜ {1, 2, 3, 4, 5, 6, 7, 8}      ⎟
          ⎜{1, 2, 3, 4, 5, 6, 7, 8, 9}    ⎟
          ⎝{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}⎠
```

```
In[18]:= Table[Range[n], {n, 1, 10}] // Column
         {1}
         {1, 2}
         {1, 2, 3}
         {1, 2, 3, 4}
Out[18]= {1, 2, 3, 4, 5}
         {1, 2, 3, 4, 5, 6}
         {1, 2, 3, 4, 5, 6, 7}
         {1, 2, 3, 4, 5, 6, 7, 8}
         {1, 2, 3, 4, 5, 6, 7, 8, 9}
         {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In[19]:= Table[Column[Range[n]], {n, 1, 10}]
                                                        1
                                                   1
                                              1         2
                                         1         2
                                    1         2         3
                               1         2         3
                          1         2         3         4
                     1         2         3         4
                1         2         3         4         5
Out[19]= { 1  ,    , 2  ,    , 3  ,    , 4  ,    , 5  ,     }
                2         3         4         5         6
                     3         4         5         6
                          4         5         6         7
                               5         6         7
                                    6         7         8
                                         7         8
                                              8         9
                                                   9
                                                        10
```

Colors and Graphics

```
In[20]:= Table[RGBColor[0.5, g, 0.5], {g, 0, 1, 0.05}]
Out[20]= {■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■}
```

```
In[21]:= Table[Hue[x], {x, 0, 1, 0.05}]
Out[21]= {■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■, ■}
```

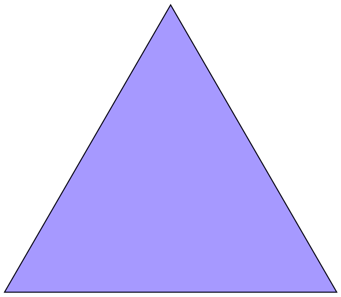
```
In[22]:= Table[Style[RandomInteger[10], RandomColor[], RandomInteger[{10, 30}]], {20}]

Out[22]= {6, 10, 0, 2, 1, 8, 10, 6, 4, 10, 1, 1, 3, 7, 0, 6, 10, 3, 8, 10}

In[23]:= ColorData["Named"]

Out[23]= {Atoms, Crayola, GeologicAges, HTML, Legacy, WebSafe}

In[24]:= ColorData["Legacy", "Image"]

Out[24]=
```

```
In[25]:= ColorData["HTML", "Image"]

Out[25]=
```

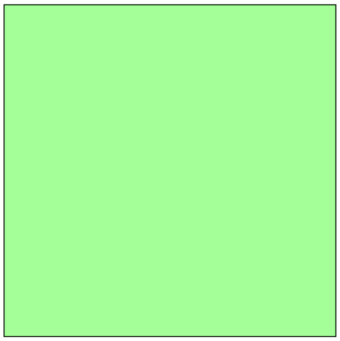
```
In[26]:= ColorData["GeologicAges", "ColorRules"]

Out[26]= {Phanerozoic → 🟩, Cenozoic → 🟨, Quaternary → 🟨, Holocene → 🟨, Pleistocene → 🟨,
  Neogene → 🟧, Pliocene → 🟨, Miocene → 🟨, Paleogene → 🟧, Oligocene → 🟧,
  Eocene → 🟧, Paleocene → 🟧, Mesozoic → 🟩, Cretaceous → 🟩, UpperCretaceous → 🟩,
  LowerCretaceous → 🟩, Jurassic → 🟩, UpperJurassic → 🟩, MiddleJurassic → 🟩,
  LowerJurassic → 🟩, Triassic → 🟩, UpperTriassic → 🟩, MiddleTriassic → 🟩,
  LowerTriassic → 🟩, Paleozoic → 🟦, Permian → 🟦, Lopingian → 🟦,
  Guadalupian → 🟦, Cisuralian → 🟦, Carboniferous → 🟦, Pennsylvanian → 🟦,
  Mississippian → 🟦, Devonian → 🟪, UpperDevonian → 🟪, MiddleDevonian → 🟪,
  LowerDevonian → 🟪, Silurian → 🟪, Pridoli → 🟪, Ludlow → 🟪, Wenlock → 🟪,
  Llandovery → 🟪, Ordovician → 🟥, UpperOrdovician → 🟥, MiddleOrdovician → 🟥,
  LowerOrdovician → 🟥, Cambrian → 🟥, UpperCambrian → 🟧, MiddleCambrian → 🟧,
  LowerCambrian → 🟥, Precambrian → 🟫, Proterozoic → 🟩, Neoproterozoic → 🟫,
  Ediacaran → ⬜, Cryogenian → 🟫, Tonian → 🟫, Mesoproterozoic → 🟫,
  Stenian → 🟫, Ectasian → 🟫, Calymmian → 🟫, Paleoproterozoic → 🟩,
  Statherian → 🟩, Orosirian → 🟩, Rhyacian → 🟩, Siderian → 🟩, Archean → ⬜,
  Neoarchean → ⬜, Mesoarchean → ⬜, Paleoarchean → ⬛, Eoarchean → ⬛, Hadean → ⬛}
```

```
In[27]:= Graphics[{EdgeForm[Black],
            FaceForm[{Opacity[0.4], Hue[RandomReal[]]}], RegularPolygon[#]}] & /@ Range[3, 10]
```

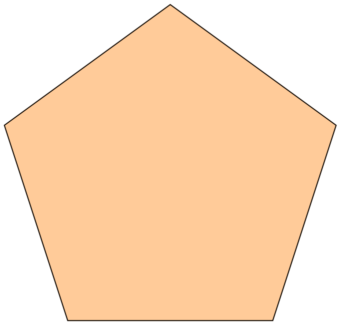
Out[27]= {



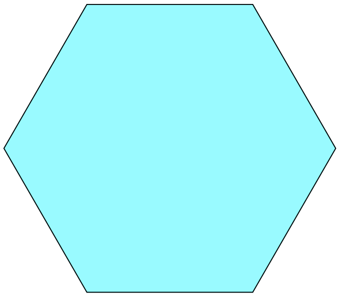
,



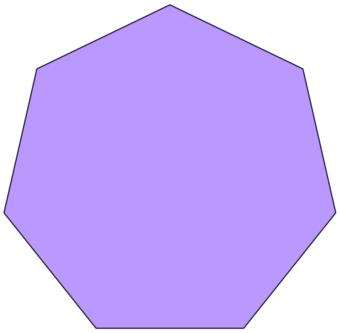
,



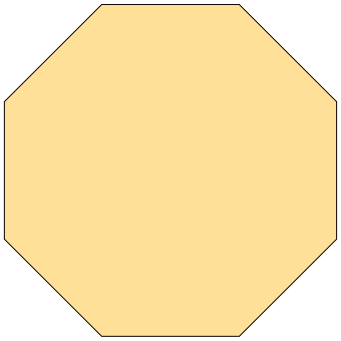
,



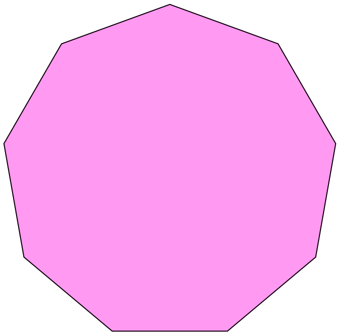
,



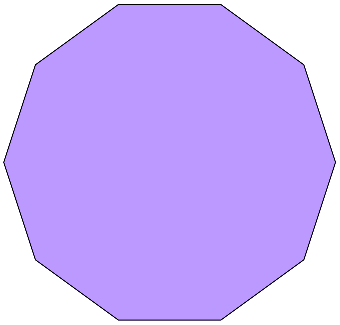
,



,



,

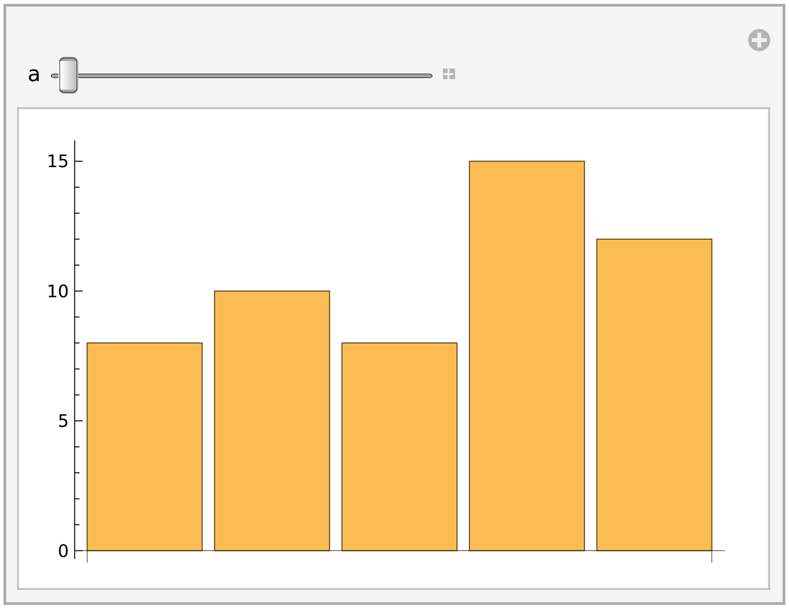


}

Manipulation

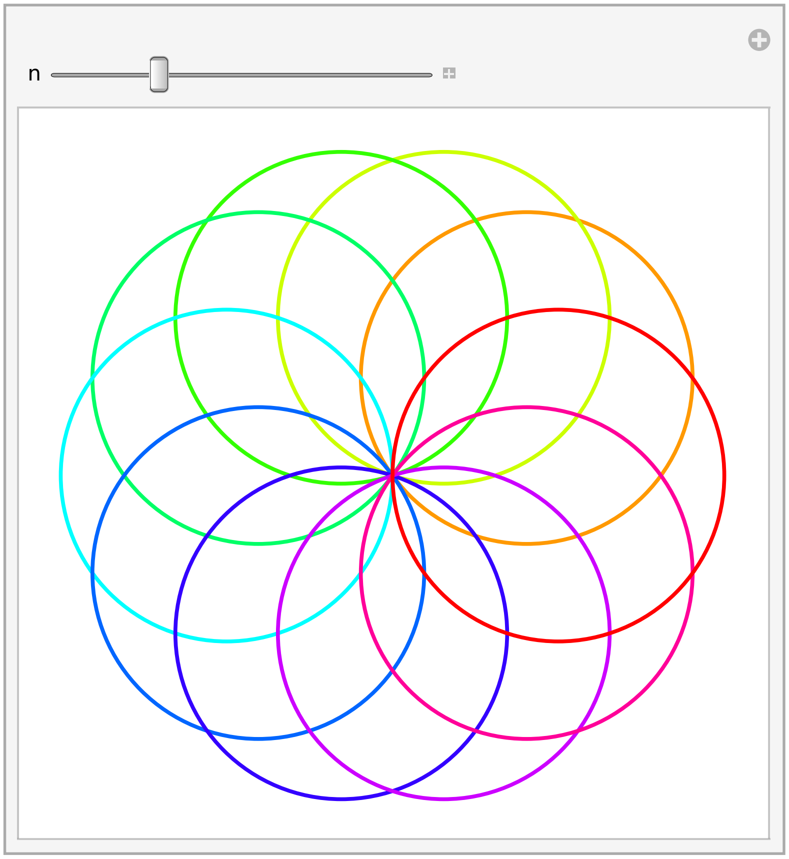
```
In[28]:=  Manipulate[BarChart[RandomInteger[{5, 20}, a]], {a, 5, 10, 1}]
```

Out[28]=



```
In[29]:= Manipulate[Graphics[Table[{Hue[t / n], Thick,
          Circle[{Cos[2 Pi t / n], Sin[2 Pi t / n]}, 1]}, {t, n}]], {{n, 10}, 3, 30, 1}]
```

Out[29]=

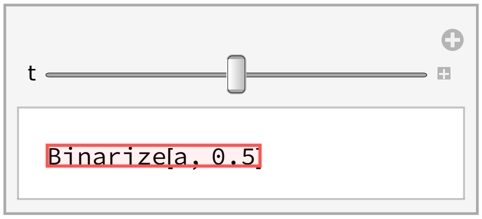


```
In[30]:= a = ExampleData[{"TestImage", "Mandrill"}]
```

Out[30]=

```
In[31]:=

In[32]:= Manipulate[Binarize[a, t], {{t, 0.5}, 0, 1}]
```

Out[32]=



```
Binarize[a, 0.5]
```

```
In[33]:=
```

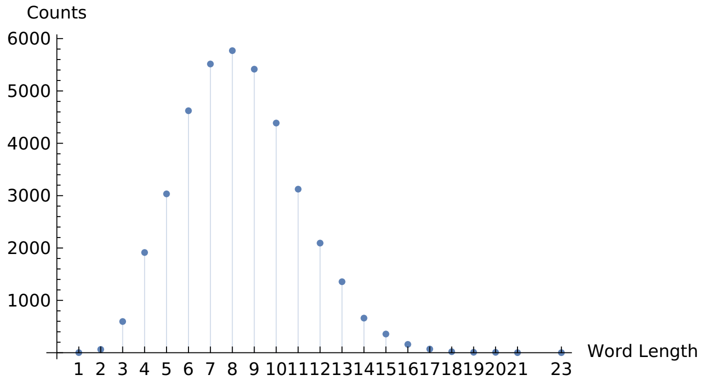
Strings and Text

```
In[34]:= Take[WordList[], 100]

Out[34]= {a, aah, aardvark, aback, abacus, abaft, abalone, abandon, abandoned,
          abandonment, abase, abasement, abash, abashed, abashment, abate, abatement,
          abattoir, abbe, abbess, abbey, abbot, abbreviate, abbreviated, abbreviation,
          abdicate, abdication, abdomen, abdominal, abduct, abducting, abduction,
          abductor, abeam, abed, aberrant, aberration, abet, abettor, abeyance, abhor,
          abhorrence, abhorrent, abidance, abide, abiding, ability, abject, abjection,
          abjectly, abjuration, abjure, ablate, ablated, ablation, ablative, ablaze,
          able, abloom, ablution, ably, abnegate, abnegation, abnormal, abnormality,
          abnormally, aboard, abode, abolish, abolition, abolitionism, abolitionist,
          abominable, abominably, abominate, abomination, aboriginal, aborigine,
          abort, abortion, abortionist, abortive, abortively, abound, abounding, about,
          above, aboveboard, abracadabra, abrade, abrasion, abrasive, abrasiveness,
          abreast, abridge, abridged, abridgment, abroad, abrogate, abrogation}
```

```
In[35]:= wordsLengthCounts = Counts[StringLength[WordList[]]]

Out[35]= <|1 → 2, 3 → 596, 8 → 5770, 5 → 3034, 6 → 4622, 7 → 5515, 9 → 5416,
          11 → 3124, 4 → 1914, 10 → 4387, 12 → 2094, 14 → 661, 16 → 159, 13 → 1356,
          15 → 357, 2 → 63, 17 → 70, 20 → 7, 18 → 19, 19 → 8, 21 → 1, 23 → 1|>
```

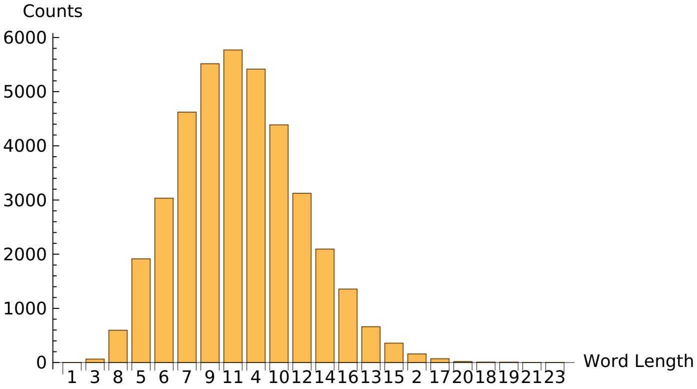
```
In[36]:= ListPlot[wordsLengthCounts, Filling → 0,
          AxesLabel → {"Word Length", "Counts"}, Ticks → {Keys[wordsLengthCounts], Automatic}]
```

Out[36]=



In[37]:=
```
BarChart[KeySort[wordsLengthCounts],
 AxesLabel → {"Word Length", "Counts"}, ChartLabels → Keys[wordsLengthCounts]]
```

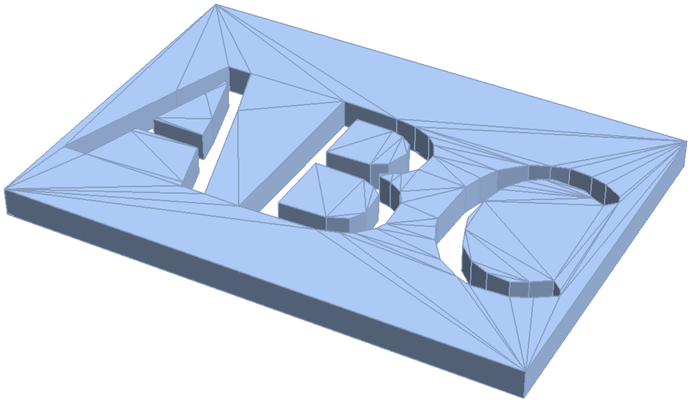
Out[37]=



In[38]:=
```
extrudeText[text_] := Block[{img, res},
img = Rasterize[Style[text, 100]];
res = ImageMesh[img];
RegionProduct[res, Line[{{0.}, {10.}}]]
]
```

In[39]:=
```
extrudeText["ABC"]
```

Out[39]=



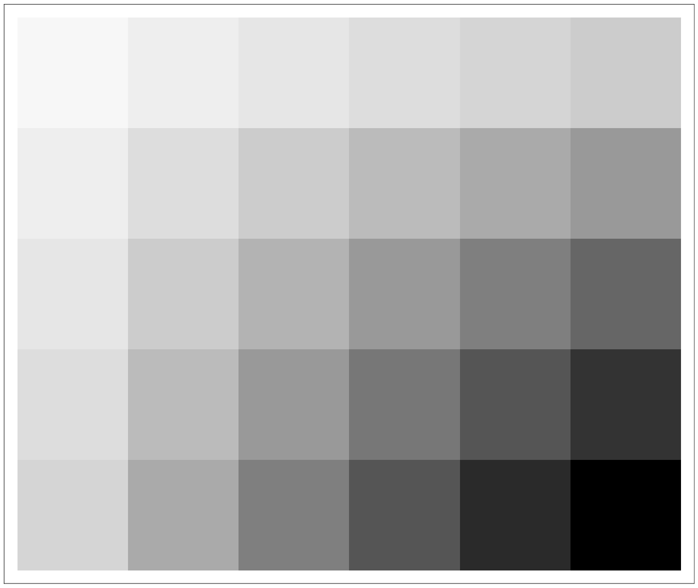
Arrays, lists of lists

In[40]:=
```
Table[i * j, {i, 5}, {j, 6}] // Grid
```

Out[40]=
```
1  2   3   4   5   6
2  4   6   8  10  12
3  6   9  12  15  18
4  8  12  16  20  24
5 10  15  20  25  30
```

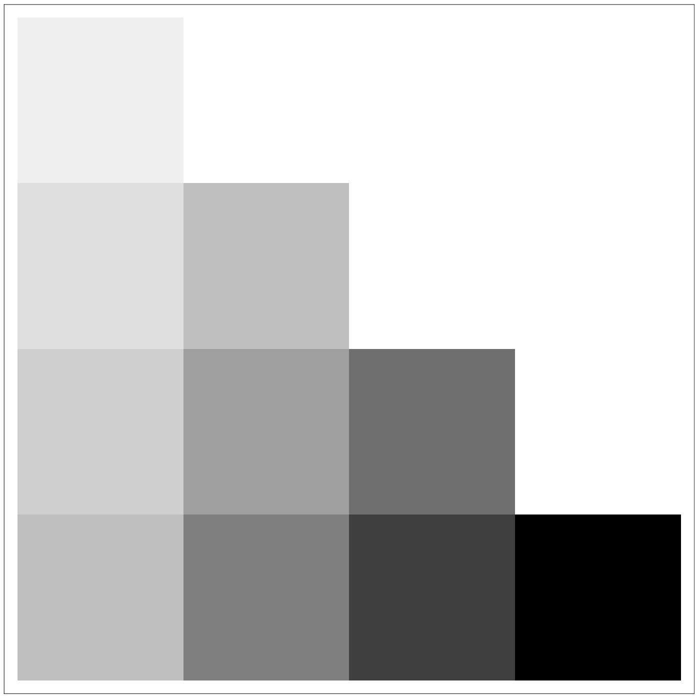
In[41]:=
```
ArrayPlot[Table[i * j, {i, 5}, {j, 6}]]
```

Out[41]=



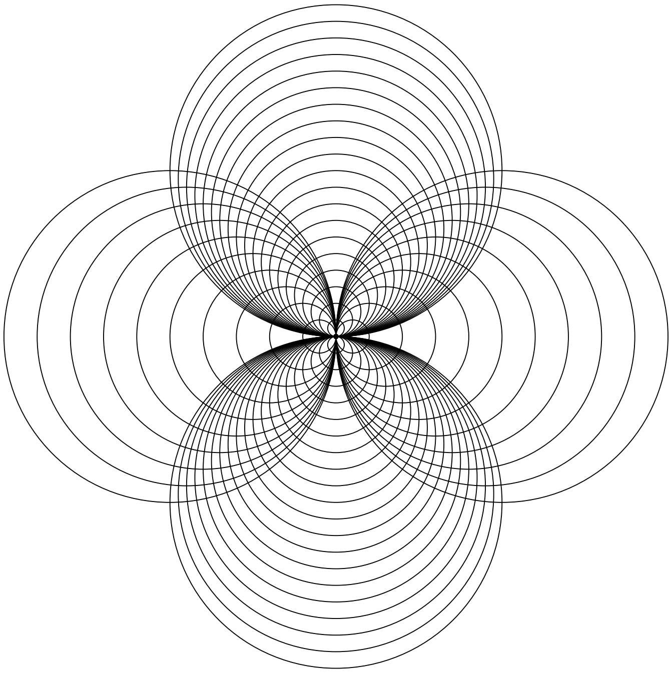
In[42]:=
```
ArrayPlot[Table[i * j, {i, 4}, {j, i}]]
```

Out[42]=



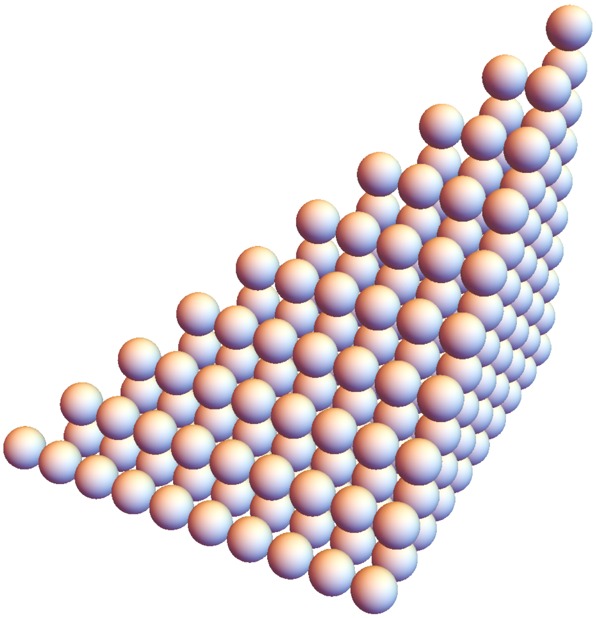
```
In[43]:= Graphics[Table[Circle[{0, x}, Abs@x], {x, -10, 10, 0.5}]~
           Join~Table[Circle[{x, 0}, Abs@x], {x, -10, 10, 1}]]
```

Out[43]=



```
In[44]:= Graphics3D[Table[Sphere[{x, y, z}, 0.5], {x, 10}, {y, x}, {z, y}], Boxed → False]
```

Out[44]=



Real-World Data

```
In[45]:=  United States COUNTRY ⋯ ✓

Out[45]=  United States

In[46]:=  InputForm[United States COUNTRY ⋯ ✓]

Out[46]//InputForm=
          "Entity[\"Country\", \"UnitedStates\"]"

In[47]:=  CountryData["UnitedStates"]

Out[47]=  United States

In[48]:=  InputForm[CountryData["UnitedStates"]]
Out[48]//InputForm=
          "Entity[\"Country\", \"UnitedStates\"]"

In[49]:=  ⊟ 2.6 h ✓

Out[49]=  2.6 h

In[50]:=  InputForm[⊟ 2.6 h ✓]

Out[50]//InputForm=
          "Quantity[2.6, \"Hours\"]"

In[51]:=  Quantity[2.6, "Hours"]

Out[51]=  2.6 h

In[52]:=  Quantity[7.5, "Feet"] + Quantity[14, "cm"]

Out[52]=  242.6 cm

In[53]:=  ⊟ 7.5 ft ✓ + ⊟ 14 cm ✓

Out[53]=  242.6 cm

In[54]:=  N@UnitConvert[Quantity[100, "lb"], "kg"]

Out[54]=  45.3592 kg

In[55]:=  EarthquakeData[Entity["Earthquake", "nc73666231"]]

Out[55]=  Missing[NotAvailable]
```

{"type":"Feature","properties":{"mag":6.2,"place":"38km W of Petrolia, CA","time":1640031019100,"updated":1640430417695,"tz":null,"url":"https://earthquake.usgs.gov/earthquakes/eventpage/nc736662

31","detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc73666231.geojson","felt":4372,"cdi":7,"mmi":6.936,"alert":"yellow","status":"reviewed","tsunami":1,"sig":1350,"net":"nc","code":"73666231","ids":",ew1640031020,nc73666231,at00r4fk16,us6000gdxq,","sources":",ew,nc,at,us,","types":",dyfi,focal-mechanism,general-text,ground-failure,impact-link,losspager,moment-tensor,nearby-cities,oaf,origin,phase-data,scitech-link,shake-alert,shakemap,","nst":118,"dmin":0.2951,"rms":0.22,"gap":224,"magType":"mw","type":"earthquake","title":"M 6.2 - 38km W of Petrolia, CA"},"geometry":{"type":"Point","coordinates":[-124.727,40.314,14.85]},"id":"nc73666231"}

```
In[56]:= data = EarthquakeData[GeoPosition[{40.31, -124.7 }], 6]
```

```
Out[56]=
⟨| centennial625301 →
    ⟨|Period → July 26, 1890 3:40 am GMT−6 , AzimuthalGap → Missing[NotAvailable],
     Depth → Missing[NotAvailable], ··· 5 ··· , Position → GeoPosition[{40.5, −124.2}],
     PositionMethod → Missing[NotAvailable]|⟩, ··· 23 ··· ,
   ncnc73666231 → ⟨|Period → ··· 1 ··· , ··· 8 ··· , ··· 1 ··· |⟩|⟩

 large output | show less | show more | show all | set size limit...
```

```
In[57]:= EarthquakeData[Entity["Earthquake", "ncnc73666231"]]
```

```
Out[57]= ⟨|Period → December 20, 2021 2:10 pm GMT−6 ,
          Magnitude → 6.2, Position → GeoPosition[{40.314, −124.727}]|⟩
```

```
In[58]:= mintemp =
           (Normal@WeatherData["KBMG", "MinTemperature", {{#, 12, 25}, {#, 12, 25}, "Day"}]) & /@
            Range[1991, 2021, 1];
```

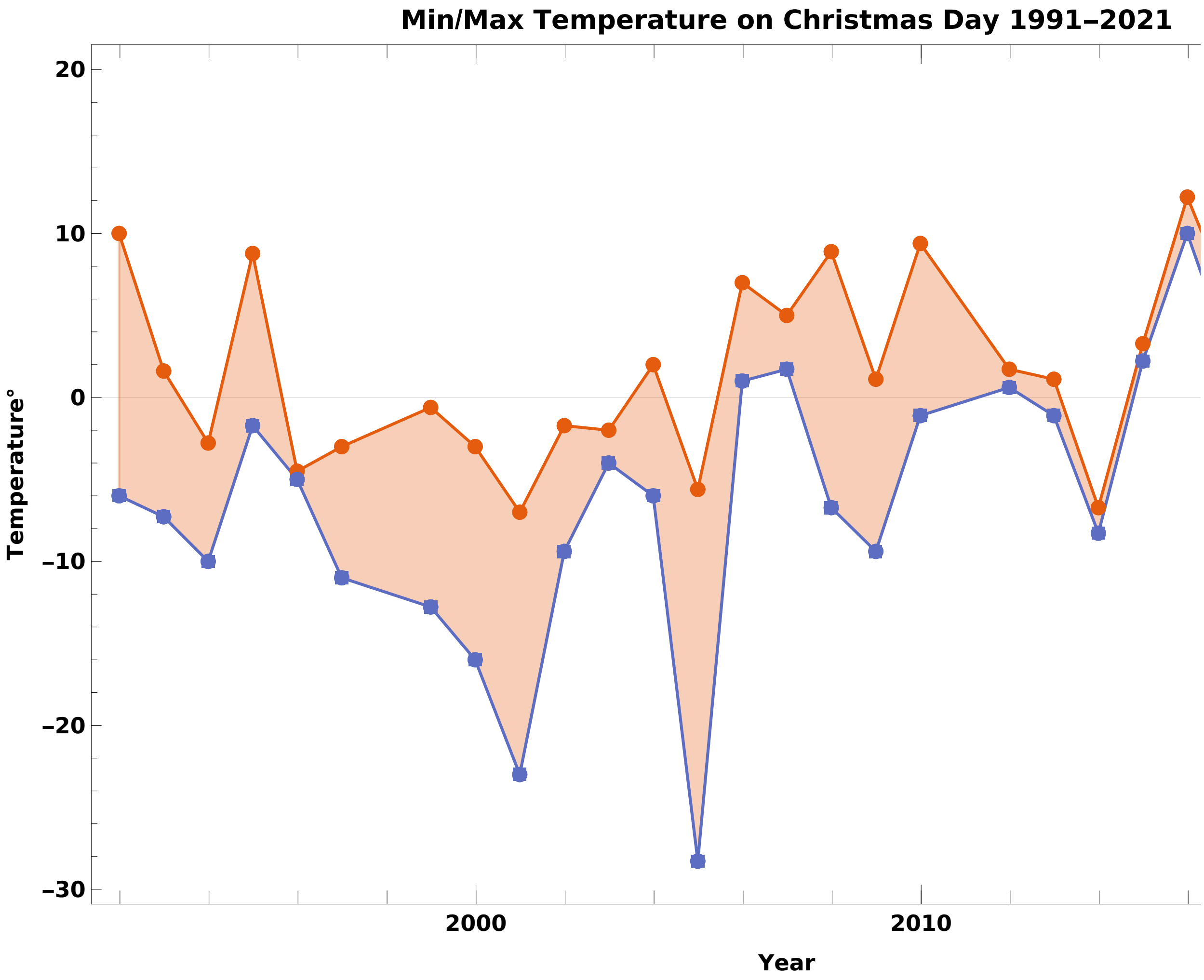
```
In[59]:= maxtemp =
           (Normal@WeatherData["KBMG", "MaxTemperature", {{#, 12, 25}, {#, 12, 25}, "Day"}]) & /@
            Range[1991, 2021, 1];
```

```
In[60]:= mintemp = DeleteMissing[Flatten[mintemp, 1]]
Out[60]= {{December 25, 1991 12:00 am GMT−6 , -6 °C},
  {December 25, 1992 12:00 am GMT−6 , -7.28 °C}, {December 25, 1993 12:00 am GMT−6 , -10 °C},
  {December 25, 1994 12:00 am GMT−6 , -1.72 °C}, {December 25, 1995 12:00 am GMT−6 , -5 °C},
  {December 25, 1996 12:00 am GMT−6 , -11 °C}, {December 25, 1998 12:00 am GMT−6 , -12.78 °C},
  {December 25, 1999 12:00 am GMT−6 , -16 °C}, {December 25, 2000 12:00 am GMT−6 , -23 °C},
  {December 25, 2001 12:00 am GMT−6 , -9.39 °C}, {December 25, 2002 12:00 am GMT−6 , -4 °C},
  {December 25, 2003 12:00 am GMT−6 , -6 °C}, {December 25, 2004 12:00 am GMT−6 , -28.28 °C},
  {December 25, 2005 12:00 am GMT−6 , 1 °C}, {December 25, 2006 12:00 am GMT−6 , 1.72 °C},
  {December 25, 2007 12:00 am GMT−6 , -6.72 °C}, {December 25, 2008 12:00 am GMT−6 , -9.39 °C},
  {December 25, 2009 12:00 am GMT−6 , -1.11 °C}, {December 25, 2011 12:00 am GMT−6 , 0.61 °C},
  {December 25, 2012 12:00 am GMT−6 , -1.11 °C}, {December 25, 2013 12:00 am GMT−6 , -8.28 °C},
  {December 25, 2014 12:00 am GMT−6 , 2.22 °C}, {December 25, 2015 12:00 am GMT−6 , 10 °C},
  {December 25, 2016 12:00 am GMT−6 , 2.22 °C}, {December 25, 2017 12:00 am GMT−6 , -7 °C},
  {December 25, 2018 12:00 am GMT−6 , 0 °C}, {December 25, 2019 12:00 am GMT−6 , -1.11 °C},
  {December 25, 2020 12:00 am GMT−6 , -12.22 °C}, {December 25, 2021 12:00 am GMT−6 , 16.72 °C}}
```

```
In[61]:= maxtemp = DeleteMissing[Flatten[maxtemp, 1]]

Out[61]= {{December 25, 1991 12:00 am GMT−6 , 10 °C},
  {December 25, 1992 12:00 am GMT−6 , 1.61 °C}, {December 25, 1993 12:00 am GMT−6 , -2.78 °C},
  {December 25, 1994 12:00 am GMT−6 , 8.78 °C}, {December 25, 1995 12:00 am GMT−6 , -4.5 °C},
  {December 25, 1996 12:00 am GMT−6 , -3 °C}, {December 25, 1998 12:00 am GMT−6 , -0.61 °C},
  {December 25, 1999 12:00 am GMT−6 , -3 °C}, {December 25, 2000 12:00 am GMT−6 , -7 °C},
  {December 25, 2001 12:00 am GMT−6 , -1.72 °C}, {December 25, 2002 12:00 am GMT−6 , -2 °C},
  {December 25, 2003 12:00 am GMT−6 , 2 °C}, {December 25, 2004 12:00 am GMT−6 , -5.61 °C},
  {December 25, 2005 12:00 am GMT−6 , 7 °C}, {December 25, 2006 12:00 am GMT−6 , 5 °C},
  {December 25, 2007 12:00 am GMT−6 , 8.89 °C}, {December 25, 2008 12:00 am GMT−6 , 1.11 °C},
  {December 25, 2009 12:00 am GMT−6 , 9.39 °C}, {December 25, 2011 12:00 am GMT−6 , 1.72 °C},
  {December 25, 2012 12:00 am GMT−6 , 1.11 °C}, {December 25, 2013 12:00 am GMT−6 , -6.72 °C},
  {December 25, 2014 12:00 am GMT−6 , 3.28 °C}, {December 25, 2015 12:00 am GMT−6 , 12.22 °C},
  {December 25, 2016 12:00 am GMT−6 , 5.61 °C}, {December 25, 2017 12:00 am GMT−6 , -5.61 °C},
  {December 25, 2018 12:00 am GMT−6 , 8.28 °C}, {December 25, 2019 12:00 am GMT−6 , 18.89 °C},
  {December 25, 2020 12:00 am GMT−6 , -7.22 °C}, {December 25, 2021 12:00 am GMT−6 , 17.78 °C}}
```

```
In[66]:= DateListPlot[{maxtemp, mintemp}, PlotTheme → "Scientific", Filling → {1 → {2}},
          PlotLabel → "Min/Max Temperature on Christmas Day 1991–2021",
          Mesh → Full, PlotMarkers → Automatic, PlotLegends → {"Max", "Min"},
          FrameLabel → {"Year", "Temperature°"}, LabelStyle → {Bold, FontSize → 12},
          ImageSize → 800, PerformanceGoal → "Quality"]
```

Out[66]=



Dates and Times

```
In[1]:= Now
```

```
Out[1]= December 27, 2021 6:40 pm GMT−6
```

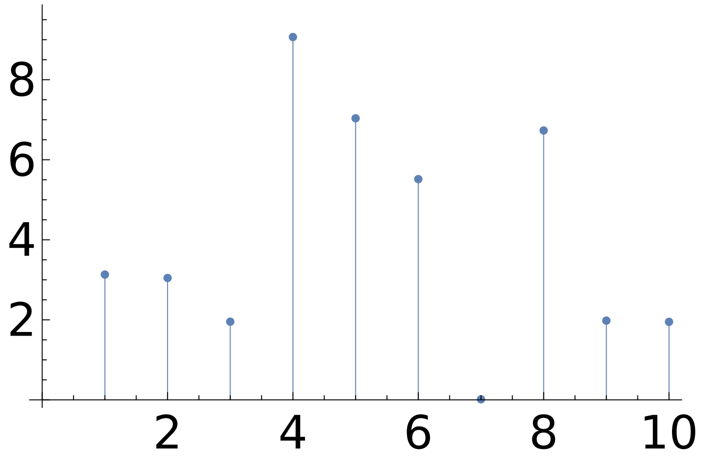
```
In[2]:=  Now - ⊟ 1 day ✓

Out[2]=  December 26, 2021 6:41 pm GMT−6

In[3]:=  Now - Quantity[1, "Day"]

Out[3]=  December 26, 2021 6:41 pm GMT−6

In[4]:=  Now - DateObject[{1988, 6, 23}]

Out[4]=  12 240. days

         DayCount[{2021, 1, 1}, {2021, 12, 27}] (* Day number shall be daycount + 1 *)

Out[8]=  360

In[9]:=  DateObject[{2021, 1, 1}] + Quantity[360, "Day"]

Out[9]=  Mon 27 Dec 2021

In[12]:= DateValue[{2021, 12, 27}, "DayOfYear"]

Out[12]= 361
```

Options

```
In[14]:= Options[ListPlot]

                                                    1
Out[14]= {AlignmentPoint → Center, AspectRatio → ─────────── , Axes → Automatic,
                                                 GoldenRatio
          AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {}, Background → None,
          BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None,
          ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic,
          ContentSelectable → Automatic, CoordinatesToolOptions → Automatic,
          DataRange → Automatic, DisplayFunction :→ $DisplayFunction, Epilog → {},
          Filling → None, FillingStyle → Automatic, FormatType :→ TraditionalForm,
          Frame → Automatic, FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic,
          FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {}, ImageMargins → 0.,
          ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic,
          InterpolationOrder → None, IntervalMarkers → Automatic,
          IntervalMarkersStyle → Automatic, Joined → False, LabelingFunction → Automatic,
          LabelingSize → Automatic, LabelStyle → {}, MaxPlotPoints → ∞, Mesh → None,
          MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic, Method → Automatic,
          MultiaxisArrangement → None, PerformanceGoal :→ $PerformanceGoal, PlotLabel → None,
          PlotLabels → None, PlotLayout → Overlaid, PlotLegends → None, PlotMarkers → None,
          PlotRange → Automatic, PlotRangeClipping → True, PlotRangePadding → Automatic,
          PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme :→ $PlotTheme,
          PreserveImageOptions → Automatic, Prolog → {}, RotateLabel → True,
          ScalingFunctions → None, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}}
```

```
In[30]:=  SetOptions[ListPlot, LabelStyle → Large, Filling → Bottom]

                                                  1
Out[30]=  {AlignmentPoint → Center, AspectRatio → ─────────── , Axes → Automatic,
                                                GoldenRatio
           AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {}, Background → None,
           BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None,
           ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic,
           ContentSelectable → Automatic, CoordinatesToolOptions → Automatic,
           DataRange → Automatic, DisplayFunction :→ $DisplayFunction, Epilog → {},
           Filling → Bottom, FillingStyle → Bottom, FormatType :→ TraditionalForm,
           Frame → Automatic, FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic,
           FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {}, ImageMargins → 0.,
           ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic,
           InterpolationOrder → None, IntervalMarkers → Automatic,
           IntervalMarkersStyle → Automatic, Joined → False, LabelingFunction → Automatic,
           LabelingSize → Large, LabelStyle → Large, MaxPlotPoints → ∞, Mesh → None,
           MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic, Method → Automatic,
           MultiaxisArrangement → None, PerformanceGoal :→ $PerformanceGoal, PlotLabel → None,
           PlotLabels → None, PlotLayout → Overlaid, PlotLegends → None, PlotMarkers → None,
           PlotRange → Automatic, PlotRangeClipping → True, PlotRangePadding → Automatic,
           PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme :→ $PlotTheme,
           PreserveImageOptions → Automatic, Prolog → {}, RotateLabel → True,
           ScalingFunctions → None, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}}

In[31]:=  ListPlot[RandomReal[10, 10] → Callouts]
```

Out[31]=

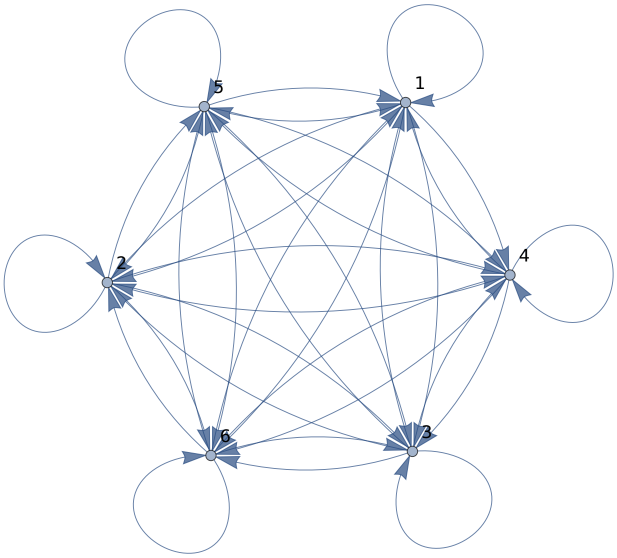


```
In[28]:= SetOptions[ListPlot, LabelStyle → Automatic, Filling → Automatic]

                                                  1
Out[28]= {AlignmentPoint → Center, AspectRatio → ─────────── , Axes → Automatic,
                                              GoldenRatio
          AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {}, Background → None,
          BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None,
          ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic,
          ContentSelectable → Automatic, CoordinatesToolOptions → Automatic,
          DataRange → Automatic, DisplayFunction :→ $DisplayFunction, Epilog → {},
          Filling → Automatic, FillingStyle → Bottom, FormatType :→ TraditionalForm,
          Frame → Automatic, FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic,
          FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {}, ImageMargins → 0.,
          ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic,
          InterpolationOrder → None, IntervalMarkers → Automatic,
          IntervalMarkersStyle → Automatic, Joined → False, LabelingFunction → Automatic,
          LabelingSize → Large, LabelStyle → Automatic, MaxPlotPoints → ∞, Mesh → None,
          MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic, Method → Automatic,
          MultiaxisArrangement → None, PerformanceGoal :→ $PerformanceGoal, PlotLabel → None,
          PlotLabels → None, PlotLayout → Overlaid, PlotLegends → None, PlotMarkers → None,
          PlotRange → Automatic, PlotRangeClipping → True, PlotRangePadding → Automatic,
          PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme :→ $PlotTheme,
          PreserveImageOptions → Automatic, Prolog → {}, RotateLabel → True,
          ScalingFunctions → None, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}}
```

Graphs and Networks

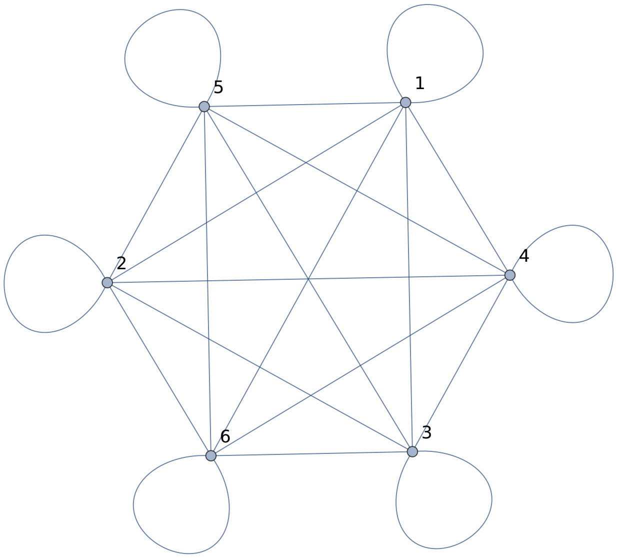
```
In[4]:=  Graph[Flatten@Table[i → j, {i, 6}, {j, 6}], VertexLabels → All]
```

Out[4]=



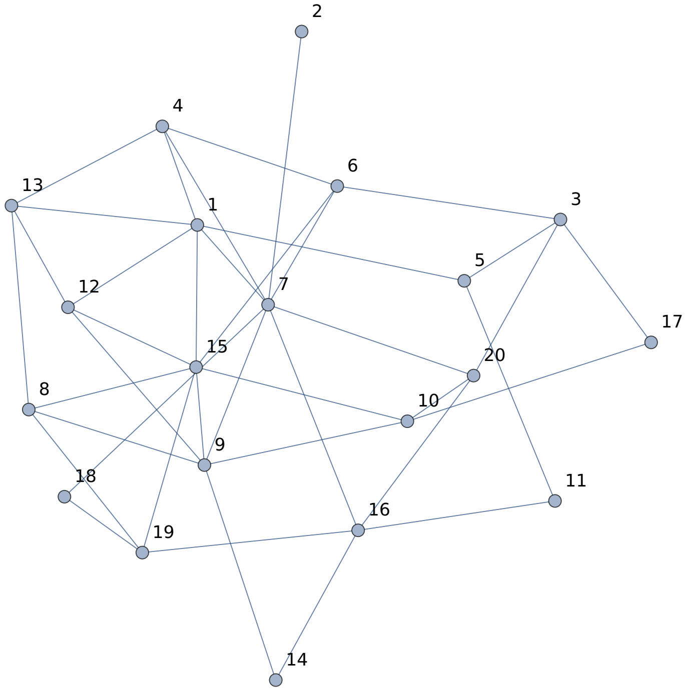
```
In[5]:=  UndirectedGraph[Flatten@Table[i → j, {i, 6}, {j, 6}], VertexLabels → All]
```

Out[5]=



```
In[8]:= aG = RandomGraph[{20, 40}, VertexLabels → All]
```

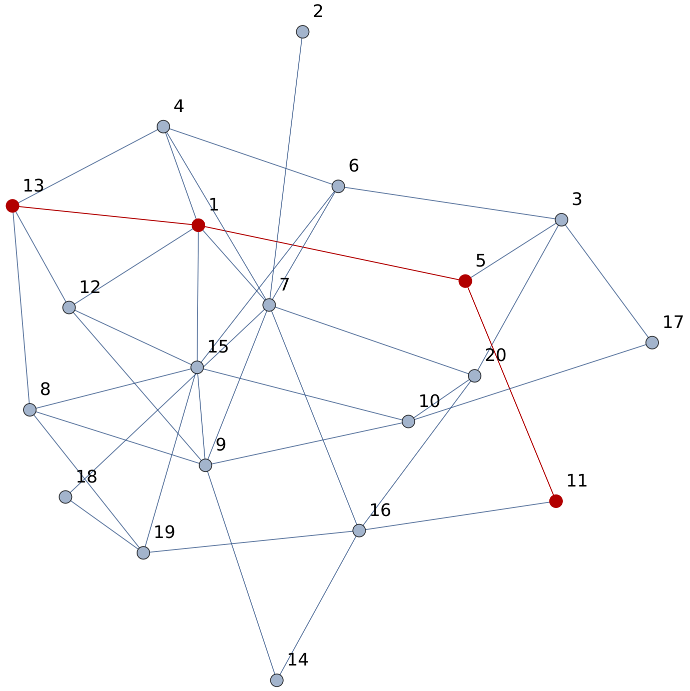
Out[8]=



```
In[11]:= apath = FindShortestPath[aG, 13, 11]
Out[11]= {13, 1, 5, 11}
```

```
In[12]:= HighlightGraph[aG, PathGraph[apath]]
```

Out[12]=

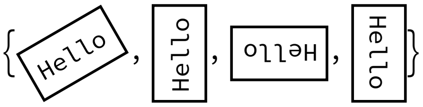


Ways to Apply Functions

```
In[13]:=  f@x
Out[13]=  f[x]

In[14]:=  f@g@h@x
Out[14]=  f[g[h[x]]]

In[15]:=  (* afterthought *)
          x // f
Out[15]=  f[x]

In[16]:=  x // h // g // f
Out[16]=  f[g[h[x]]]

In[19]:=  (f@g@h@x) === (x // h // g // f)
Out[19]=  True

          (* /@ apply to each element, map f over list *)

In[20]:=  f /@ {1, 2, 3}
Out[20]=  {f[1], f[2], f[3]}

In[21]:=  f@{1, 2, 3}
Out[21]=  f[{1, 2, 3}]

In[22]:=  f @@ {1, 2, 3}
Out[22]=  f[1, 2, 3]

In[24]:=  Framed@{x, y, z}
Out[24]=  {x, y, z}

In[25]:=  Framed /@ {x, y, x}
Out[25]=  {x, y, x}

In[26]:=  (* listable function *)
          N[{1 / 2, 1 / 3, 1 / 4}]
Out[26]=  {0.5, 0.333333, 0.25}

In[27]:=  Range[{2, 3, 4}]
Out[27]=  {{1, 2}, {1, 2, 3}, {1, 2, 3, 4}}
```

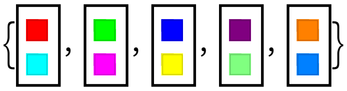
```
In[2]:= (* pure function *)
        Rotate[Framed@"Hello", #] &/@{30 Degree, 90 Degree, 180 Degree, 270 Degree}
```

Out[2]=



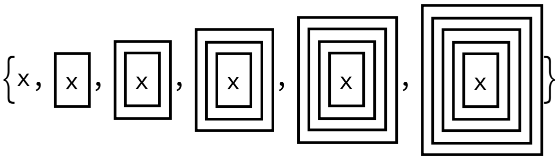
```
In[3]:= Framed[Column[{#, ColorNegate[#]}]] & /@ {Red, Green, Blue, Purple,
         Orange}
```

Out[3]=



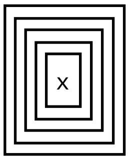
```
(* Options can often be pure functions. It's important to put parentheses
  around the whole pure function, as in ColorFunction→(Hue[#/4]&),*)


(* gives a list of the results of applying f to expr 0 through n times. *)
NestList[Framed, x, 5]
```

Out[1]=



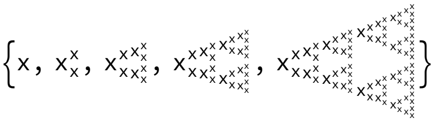
```
In[2]:= Nest[Framed, x, 5]
```

Out[2]=



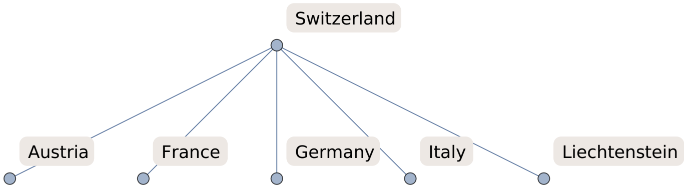
```
In[3]:= NestList[2*# &, 1, 15]

Out[3]= {1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768}

In[4]:= NestList[f, x, 5]

Out[4]= {x, f[x], f[f[x]], f[f[f[x]]], f[f[f[f[x]]]], f[f[f[f[f[x]]]]]}

In[8]:= NestList[Subsuperscript[#, #, #] &, x, 4]
```

Out[8]=



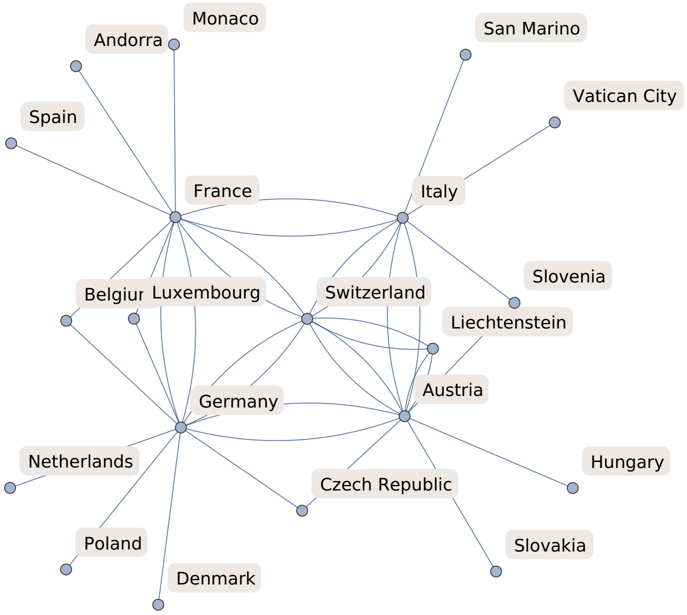
```
In[21]:= NestGraph[#["BorderingCountries"] &,
          Entity["Country", "Switzerland"], 1, VertexLabels → All, DirectedEdges → False]
```

Out[21]=



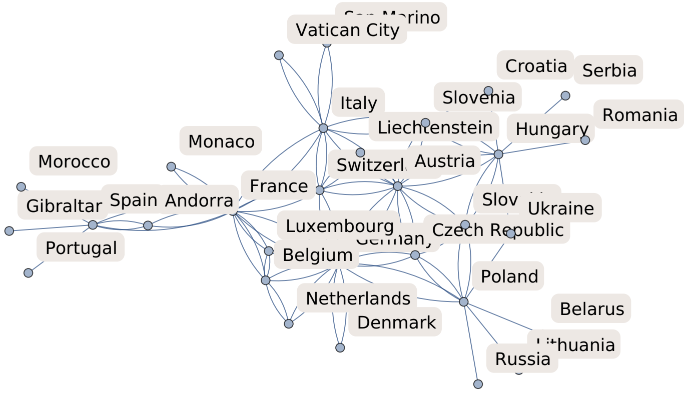
```
In[23]:= g2 = NestGraph[#["BorderingCountries"] &,
          Entity["Country", "Switzerland"], 2, VertexLabels → All, DirectedEdges → False]
```

Out[23]=



```
In[15]:= NestGraph[#["BorderingCountries"] &,
          Entity["Country", "Switzerland"], 3, VertexLabels → All, DirectedEdges → False]
```

Out[15]=



```
In[27]:= Array[f, 10]

Out[27]= {f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9], f[10]}
```

```
In[28]:=  f /@ Range[10]

Out[28]=  {f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9], f[10]}

In[29]:=  Array[f, {3, 4}] // Grid

          f[1, 1] f[1, 2] f[1, 3] f[1, 4]
Out[29]=  f[2, 1] f[2, 2] f[2, 3] f[2, 4]
          f[3, 1] f[3, 2] f[3, 3] f[3, 4]

In[30]:=  Array[Times, {3, 4}] // Grid

          1 2 3  4
Out[30]=  2 4 6  8
          3 6 9 12

In[31]:=  NestList[f, x, 5]

Out[31]=  {x, f[x], f[f[x]], f[f[f[x]]], f[f[f[f[x]]]], f[f[f[f[f[x]]]]]}

In[34]:=  FoldList[f, x, {1, 2, 3, 4, 5}]

Out[34]=  {x, f[x, 1], f[f[x, 1], 2], f[f[f[x, 1], 2], 3], f[f[f[f[x, 1], 2], 3], 4], f[f[f[f[f[x, 1], 2], 3], 4], 5]}

In[35]:=  Accumulate[{1, 2, 3, 4, 5}]

Out[35]=  {1, 3, 6, 10, 15}

In[36]:=  FoldList[Plus, {1, 2, 3, 4, 5}]

Out[36]=  {1, 3, 6, 10, 15}
```

More on Lists

```
In[41]:=  alist = {{1, 2}, {3, 4}, {5, 6}, {7, 8}}

Out[41]=  {{1, 2}, {3, 4}, {5, 6}, {7, 8}}

In[42]:=  blist = Transpose[alist]

Out[42]=  {{1, 3, 5, 7}, {2, 4, 6, 8}}

In[44]:=  MatrixForm /@ {alist, blist}
```

Out[44]=

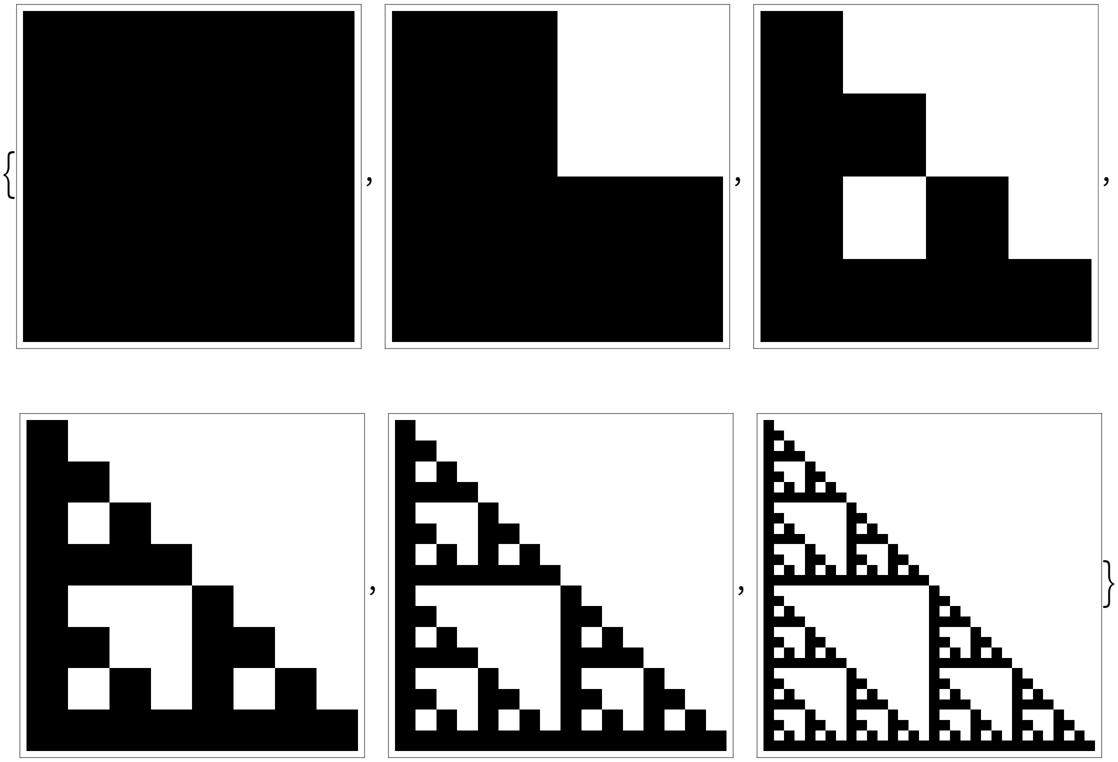
$$\left\{\begin{pmatrix}1 & 2\\ 3 & 4\\ 5 & 6\\ 7 & 8\end{pmatrix}, \begin{pmatrix}1 & 3 & 5 & 7\\ 2 & 4 & 6 & 8\end{pmatrix}\right\}$$

```
In[3]:=  Thread[{1, 3, 5, 7, 9} → {2, 4, 6, 8, 10}]

Out[3]=  {1 → 2, 3 → 4, 5 → 6, 7 → 8, 9 → 10}

In[4]:=  Thread[{1, 3, 5, 7, 9} * {2, 4, 6, 8, 10}]

Out[4]=  {2, 12, 30, 56, 90}
```

```
In[5]:= {1, 3, 5, 7, 9}*{2, 4, 6, 8, 10}

Out[5]= {2, 12, 30, 56, 90}

In[6]:= Partition[Range[10], 3]

Out[6]= {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}

In[7]:= Partition[Range[10], 2]

Out[7]= {{1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10}}

In[11]:= Partition[Range[10], 2, 1]

Out[11]= {{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 7}, {7, 8}, {8, 9}, {9, 10}}

In[12]:= Partition[Range[10], 2, 2]

Out[12]= {{1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10}}

In[13]:= Partition[Range[10], 2, 3]

Out[13]= {{1, 2}, {4, 5}, {7, 8}}

NestList[{{#, 0}, {#, #}} &, {{1}}, 2] // MatrixForm
Out[16]//MatrixForm=
(                                      {{1}}                                        )
(                       {{{{1}}, 0}, {{{1}}, {{1}}}}                                )
( {{{{{{1}}, 0}, {{{1}}, {{1}}}}, 0}, {{{{{1}}, 0}, {{{1}}, {{1}}}}, {{{{1}}, 0}, {{{1}}, {{1}}}}}} )

In[17]:= NestList[ArrayFlatten@{{#, 0}, {#, #}} &, {{1}}, 2] // MatrixForm
Out[17]//MatrixForm=
(                            {{1}}                               )
(                      {{1, 0}, {1, 1}}                          )
( {{1, 0, 0, 0}, {1, 1, 0, 0}, {1, 0, 1, 0}, {1, 1, 1, 1}} )
```

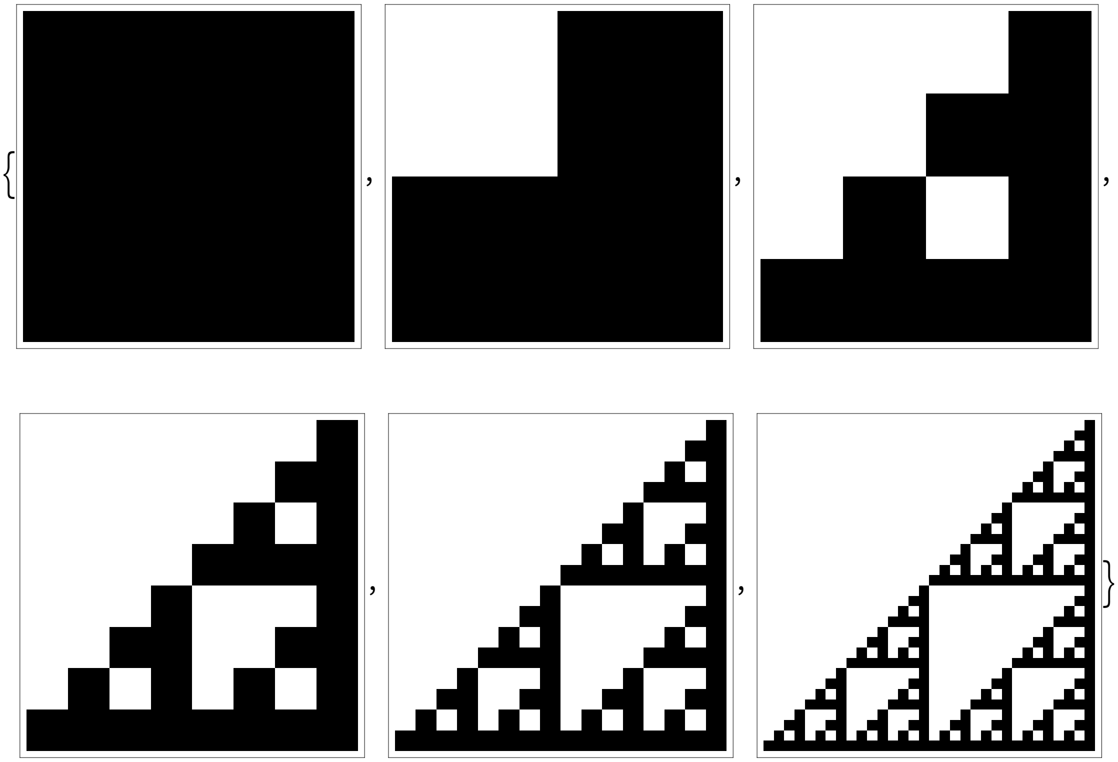
```
In[19]:= ArrayPlot /@ NestList[ArrayFlatten@{{#, 0}, {#, #}} &, {{1}}, 5]
```

Out[19]=



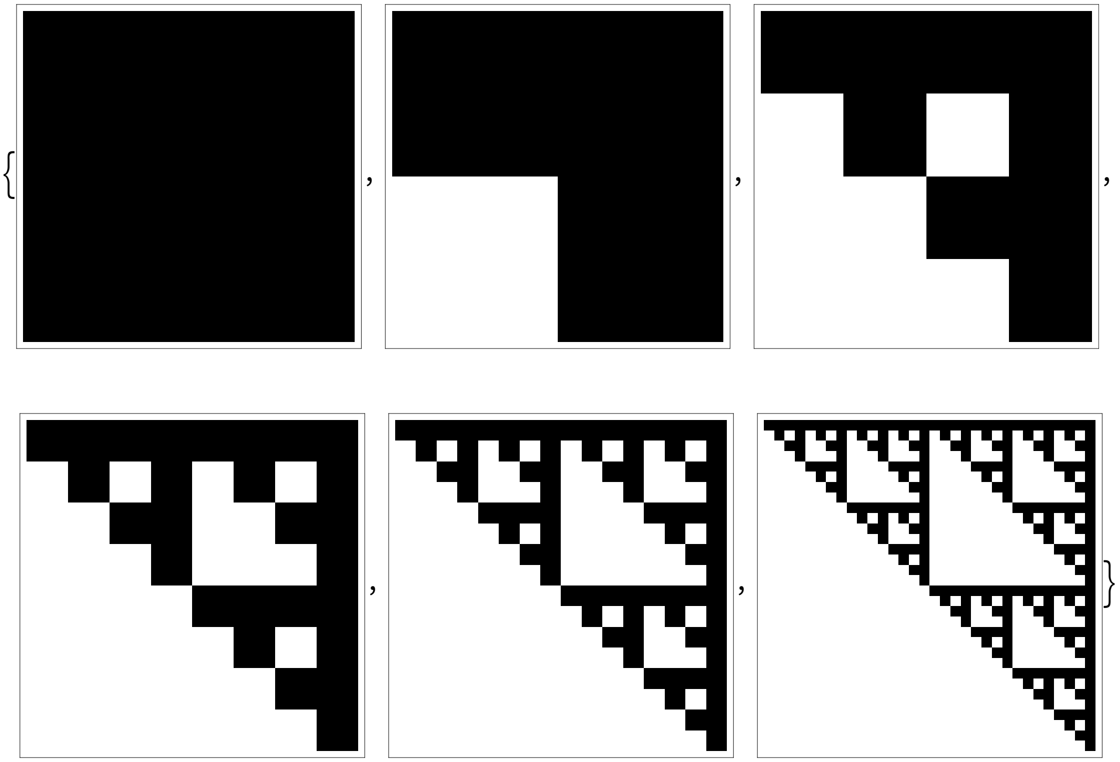
```
In[21]:= ArrayPlot /@ NestList[ArrayFlatten@{{0, #}, {#, #}} &, {{1}}, 5]
```

Out[21]=



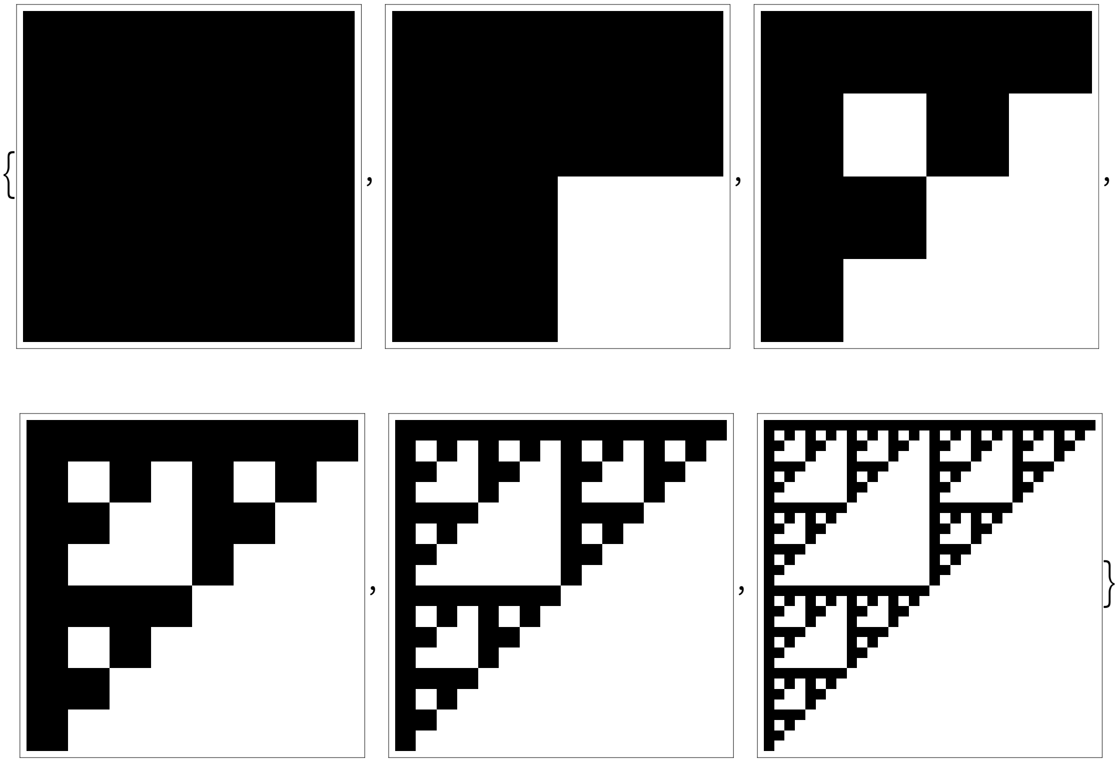
```
In[23]:=  ArrayPlot /@ NestList[ArrayFlatten@{{#, #}, {0, #}} &, {{1}}, 5]
```

Out[23]=



```
In[27]:=  ArrayPlot /@ NestList[ArrayFlatten@{{#, #}, {#, 0}} &, {{1}}, 5]
```

Out[27]=

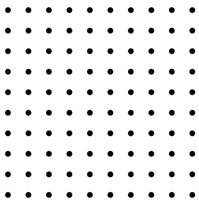


```
In[30]:= intList = RandomInteger[10, 20]

Out[30]= {3, 2, 2, 6, 6, 7, 4, 8, 2, 5, 7, 6, 6, 0, 10, 9, 3, 7, 1, 4}

In[31]:= Split[intList]

Out[31]= {{3}, {2, 2}, {6, 6}, {7}, {4}, {8}, {2}, {5}, {7}, {6, 6}, {0}, {10}, {9}, {3}, {7}, {1}, {4}}

In[32]:= Split[intList, Less]

Out[32]= {{3}, {2}, {2, 6}, {6, 7}, {4, 8}, {2, 5, 7}, {6}, {6}, {0, 10}, {9}, {3, 7}, {1, 4}}

In[33]:= Split[intList, Greater]

Out[33]= {{3, 2}, {2}, {6}, {6}, {7, 4}, {8, 2}, {5}, {7, 6}, {6, 0}, {10, 9, 3}, {7, 1}, {4}}

In[35]:= Gather[Sort[intList]]

Out[35]= {{0}, {1}, {2, 2, 2}, {3, 3}, {4, 4}, {5}, {6, 6, 6, 6}, {7, 7, 7}, {8}, {9}, {10}}

In[37]:= Gather[intList][[All, 1]]

Out[37]= {3, 2, 6, 7, 4, 8, 5, 0, 10, 9, 1}

         DeleteDuplicates[intList]
         (* DeleteDuplicates[list] does the same as Union[list], except it doesn’
          t reorder elements. *)

Out[38]= {3, 2, 6, 7, 4, 8, 5, 0, 10, 9, 1}

In[41]:= Union[intList]

Out[41]= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[40]:= Tally[intList]

Out[40]= {{3, 2}, {2, 3}, {6, 4}, {7, 3}, {4, 2}, {8, 1}, {5, 1}, {0, 1}, {10, 1}, {9, 1}, {1, 1}}

In[44]:= Permutations[{x1, x2, x3}] // MatrixForm
Out[44]//MatrixForm=
         ⎛x1  x2  x3⎞
         ⎜x1  x3  x2⎟
         ⎜x2  x1  x3⎟
         ⎜x2  x3  x1⎟
         ⎜x3  x1  x2⎟
         ⎝x3  x2  x1⎠
```

```
In[50]:= Tuples[{x1, x2, x3}, 3] // MatrixForm
Out[50]//MatrixForm=
         x1 x1 x1
         x1 x1 x2
         x1 x1 x3
         x1 x2 x1
         x1 x2 x2
         x1 x2 x3
         x1 x3 x1
         x1 x3 x2
         x1 x3 x3
         x2 x1 x1
         x2 x1 x2
         x2 x1 x3
         x2 x2 x1
         x2 x2 x2
         x2 x2 x3
         x2 x3 x1
         x2 x3 x2
         x2 x3 x3
         x3 x1 x1
         x3 x1 x2
         x3 x1 x3
         x3 x2 x1
         x3 x2 x2
         x3 x2 x3
         x3 x3 x1
         x3 x3 x2
         x3 x3 x3
```

```
In[46]:= Subsets[{x1, x2, x3}] // MatrixForm
Out[46]//MatrixForm=
              {}
             {x1}
             {x2}
             {x3}
           {x1, x2}
           {x1, x3}
           {x2, x3}
         {x1, x2, x3}
```

```
In[53]:= Graphics[Point@Tuples[Range[0, 9], 2], ImageSize → 100]
Out[53]=
```



```
In[55]:=  RandomChoice[intList, 8]

Out[55]=  {7, 6, 0, 6, 6, 5, 6, 7}

In[56]:=  RandomSample[intList, 8]

Out[56]=  {9, 7, 4, 8, 2, 7, 2, 1}

In[59]:=  intList[[{2, 4}]] == Part[intList, {2, 4}]

Out[59]=  True

In[60]:=  ReplacePart[intList, {3 → x, 5 → y}]

Out[60]=  {3, 2, x, 6, y, 7, 4, 8, 2, 5, 7, 6, 6, 0, 10, 9, 3, 7, 1, 4}

In[61]:=  Position[intList, 3]

Out[61]=  {{1}, {17}}

In[65]:=  ReplacePart[intList, Position[intList, 3] → y]

Out[65]=  {y, 2, 2, 6, 6, 7, 4, 8, 2, 5, 7, 6, 6, 0, 10, 9, y, 7, 1, 4}

In[67]:=  ReplaceAll[intList, {3 → y}]

Out[67]=  {y, 2, 2, 6, 6, 7, 4, 8, 2, 5, 7, 6, 6, 0, 10, 9, y, 7, 1, 4}

In[80]:=  intList /. 3 → y

Out[80]=  {y, 2, 2, 6, 6, 7, 4, 8, 2, 5, 7, 6, 6, 0, 10, 9, y, 7, 1, 4}

In[72]:=  TakeLargest[intList, 5]

Out[72]=  {10, 9, 8, 7, 7}

In[73]:=  TakeSmallest[intList, 5]

Out[73]=  {0, 1, 2, 2, 2}
```

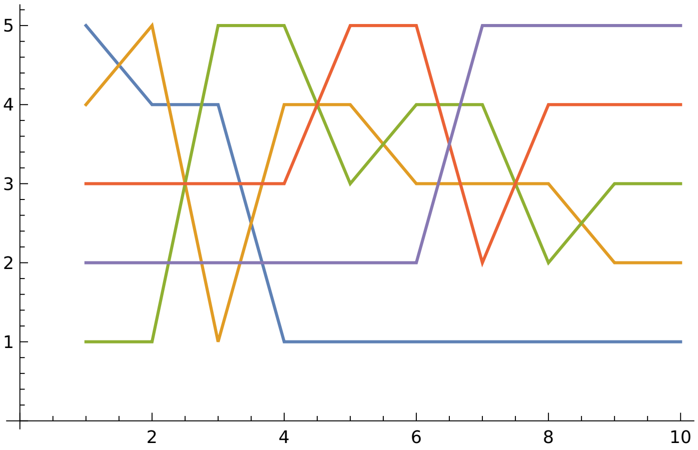
Patterns

```
          (* Patterns are a fundamental concept in the Wolfram
           Language. The pattern _ (read “blank”) stands for anything. *)

In[74]:=  MatchQ[{a, b}, {a, _}]

Out[74]=  True

In[75]:=  MatchQ[{a, b}, {_, _}]

Out[75]=  True

In[78]:=  Cases[Partition[intList, 2], {2, _}]

Out[78]=  {{2, 6}, {2, 5}}

          (* The notation __ (“double blank”) in a pattern indicates any sequence of things.*)
```

```
In[79]:=  intList /. 2 → Red

Out[79]=  {3, 🟥, 🟥, 6, 6, 7, 4, 8, 🟥, 5, 7, 6, 6, 0, 10, 9, 3, 7, 1, 4}

In[81]:=  intList /. {2 → Red, 6 → Green}

Out[81]=  {3, 🟥, 🟥, 🟩, 🟩, 7, 4, 8, 🟥, 5, 7, 🟩, 🟩, 0, 10, 9, 3, 7, 1, 4}

In[82]:=  Cases[{{a, a, a}, {a, a}, {a, b}, {a, c}, {b, a}, {b,
            b}, {c}, {a}, {b}}, {_, _}]

Out[82]=  {{a, a}, {a, b}, {a, c}, {b, a}, {b, b}}

In[87]:=  Cases[{{a, a, a}, {a, a}, {a, b}, {a, c}, {b, a}, {b,
            b}, {c}, {a}, {b}}, {x_, x_}] (* named pattern *)

Out[87]=  {{a, a}, {b, b}}

          (* _ (Blank)— any expression (a "blank" to be filled in)
          x_ — any expression, to be referred to as x
          _h anything with head h
          x_h anything with head h and call it x
          __ (BlankSequence)— any sequence of one or more expression
          ___ (BlankNullSequence)— any sequence of zero or more expressions *)

In[1]:=   digitback[n_Integer] := Framed[Reverse[IntegerDigits[n]]]

In[2]:=   digitback[2439]

Out[2]=   {9, 3, 4, 2}

In[3]:=   digitback[3.1415]

Out[3]=   digitback[3.1415]

In[8]:=   pdigitback[n_Integer /; n > 0] :=
            Framed[Reverse[IntegerDigits[n]], Background → LightGreen]

In[9]:=   pdigitback[12 346]

Out[9]=   {6, 4, 3, 2, 1}

In[10]:=  pdigitback[-12 346]

Out[10]=  pdigitback[-12 346]

In[11]:=  check[x_, y_] := Red /; x > y
          check[x_, y_] := Green /; x ≤ y

In[13]:=  {check[1, 2], check[2, 1]}

Out[13]=  {🟩, 🟥}
```

```
In[14]:= (* sorting with pattern *)
         alist = {5, 4, 1, 3, 2}
         alist /. {x___, b_, a_, y___} /; b > a → {x, a, b, y}

Out[14]= {5, 4, 1, 3, 2}

Out[15]= {4, 5, 1, 3, 2}

In[17]:= sortinglist = FixedPointList[(# /. {x___, b_, a_, y___} /; b > a → {x, a, b, y}) &, alist]

Out[17]= {{5, 4, 1, 3, 2}, {4, 5, 1, 3, 2}, {4, 1, 5, 3, 2}, {1, 4, 5, 3, 2}, {1, 4, 3, 5, 2},
          {1, 3, 4, 5, 2}, {1, 3, 4, 2, 5}, {1, 3, 2, 4, 5}, {1, 2, 3, 4, 5}, {1, 2, 3, 4, 5}}

In[20]:= ListLinePlot[Transpose@sortinglist]
```

Out[20]=



```
In[24]:= (* Transpose to find the list of elements appearing first,
         second, etc. at successive steps: *)
         Transpose[sortinglist] // MatrixForm

Out[24]//MatrixForm=
         5  4  4  1  1  1  1  1  1  1
         4  5  1  4  4  3  3  3  2  2
         1  1  5  5  3  4  4  2  3  3
         3  3  3  3  5  5  2  4  4  4
         2  2  2  2  2  2  5  5  5  5

In[26]:= (* string patter use ~~ *)

In[25]:= StringCases["the [important] word", "[" ~~ x__ ~~ "]" → Framed[x]]

Out[25]= {important}

In[27]:= StringCases["now [several] important [words]",
          "[" ~~ Shortest[x__] ~~ "]" → Framed[x]]

Out[27]= {several, words}
```

```
In[28]:=  StringReplace["now [several] important [words]",
           "[" ~~ Shortest[x__] ~~ "]" :→ ToUpperCase[x]]

Out[28]=  now SEVERAL important WORDS

          (* You can use | and .. in string patterns just like in ordinary patterns.*)

In[32]:=  StringCases["12 and 123 and 4567 and 0x456", DigitCharacter ..]

Out[32]=  {12, 123, 4567, 0, 456}

In[33]:=  StringCases["12 and 123 and 4567 and 0x456", DigitCharacter]

Out[33]=  {1, 2, 1, 2, 3, 4, 5, 6, 7, 0, 4, 5, 6}

In[34]:=  StringTemplate["`1` to the `2` is <*#1^#2*>"][2, 50]

Out[34]=  2 to the 50 is 1125899906842624

In[35]:=  (* p.. is means Repeated[p] *)
          {{}, {a, a}, {a, b}, {a, a, a}, {a}} /. {a ..} → x

Out[35]=  {{}, x, {a, b}, x, x}

In[37]:=  {{}, {a, a}, {a, b}, {a, a, a}, {a}} /. a → x

Out[37]=  {{}, {x, x}, {x, b}, {x, x, x}, {x}}
```

Expression and Associations

```
In[38]:=  (* length doesn't care about head *)
          Length[f[x, y, z]]

Out[38]=  3

In[40]:=  Length[f[{x, y, z}]]

Out[40]=  1

          (* map also skip the head *)
          Apply[f, g[x, y, z]]

Out[41]=  f[x, y, z]

In[42]:=  f /@ g[x, y, z]

Out[42]=  g[f[x], f[y], f[z]]

In[43]:=  Map[f, g[x, y, z]]

Out[43]=  g[f[x], f[y], f[z]]

In[50]:=  #1 → #2 & @@ {x, y}

Out[50]=  x → y
```

```
In[53]:=  Rule @@ {x, y}
Out[53]=  x → y

In[55]:=  Rule @@@ {{1, 10}, {2, 20}, {3, 30}}
Out[55]=  {1 → 10, 2 → 20, 3 → 30}

In[56]:=  (* f@@list → replace the head of list with f  Apply[f,expr]
             f@@@{list1,list2, ..} → replace heeds of list1,
          list2, .. with f Apply[f,expr,{1}]*)

In[57]:=  (* Associations are a kind of generalization of lists,
          in which every element has a key as well as a value. *)
          Counts[{a, a, b, c, a, a, b, c, c, a, a}]
Out[57]=  <|a → 6, b → 2, c → 3|>

In[58]:=  <|a → 6, b → 2, c → 3|>[c]
Out[58]=  3

In[59]:=  f /@ <|a → 6, b → 2, c → 3|>
Out[59]=  <|a → f[6], b → f[2], c → f[3]|>

In[60]:=  Sort[<|a → 6, b → 2, c → 3|>]
Out[60]=  <|b → 2, c → 3, a → 6|>

In[61]:=  KeySort[<|a → 6, b → 2, c → 3|>]
Out[61]=  <|a → 6, b → 2, c → 3|>

In[62]:=  Keys[<|a → 6, b → 2, c → 3|>]
Out[62]=  {a, b, c}

In[64]:=  Values[<|a → 6, b → 2, c → 3|>]
Out[64]=  {6, 2, 3}

In[65]:=  Normal[<|a → 6, b → 2, c → 3|>]
Out[65]=  {a → 6, b → 2, c → 3}

In[67]:=  Association[{a → 6, b → 2, c → 3}]
Out[67]=  <|a → 6, b → 2, c → 3|>
```

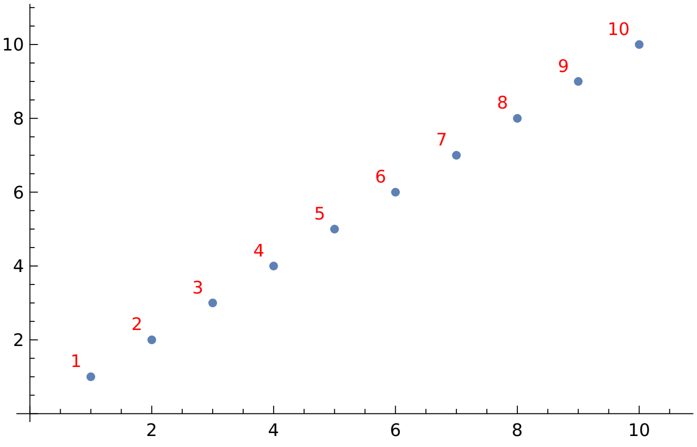
Layout and Display

```
In[25]:=  Table[Labeled[i, IntegerName[i]], {i, 1, 10}]
Out[25]=  {  1 ,  2 ,   3   ,  4  ,  5  ,  6 ,   7   ,   8   ,  9   , 10 }
            one  two  three  four  five  six  seven  eight  nine  ten
```

```
In[27]:= Table[Labeled[i, Style[IntegerName[i], Red], Right], {i, 1, 10}]

Out[27]= {1 one , 2 two , 3 three , 4 four , 5 five , 6 six , 7 seven , 8 eight , 9 nine , 10 ten}
```

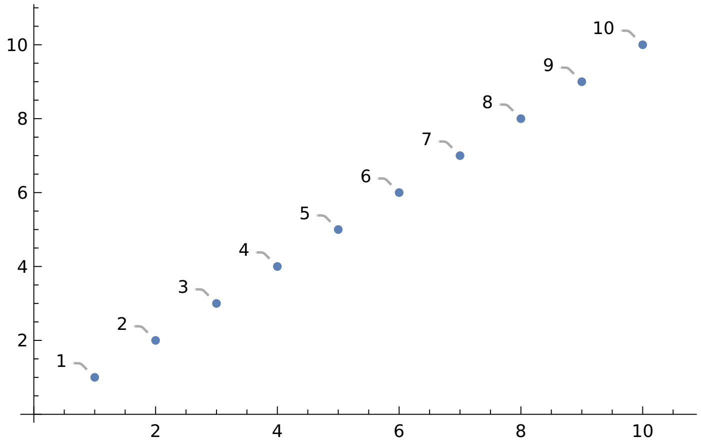
```
In[28]:= ListPlot[Table[Labeled[i, Style[i, Red]], {i, 1, 10}]]
```

Out[28]=



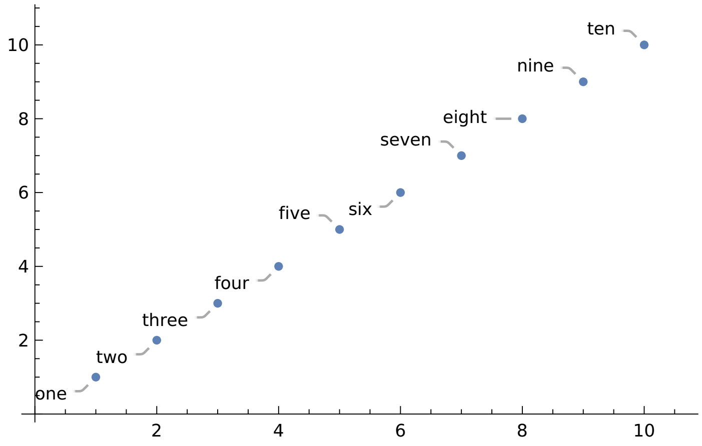
```
In[3]:= ListPlot[Table[Callout[i, i], {i, 1, 10}]]
```

Out[3]=



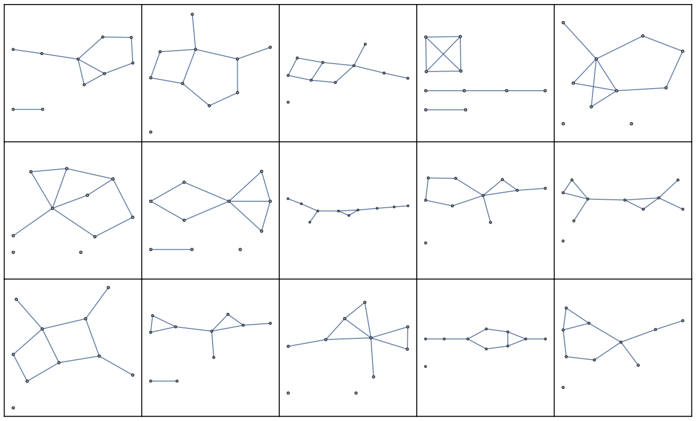
```
In[13]:= ListPlot[Flatten[{# → IntegerName[#]} & /@ Range[10]]]
```

Out[13]=



```
In[19]:= GraphicsGrid[Table[RandomGraph[{10, 10}], 3, 5], Frame → All]
```

Out[19]=



```
In[21]:= Framed[1/x + y + z, RoundingRadius → #] &/@{5, 10, 20}
```

Out[21]= $\left\{\boxed{\frac{1}{x}+y+z}, \boxed{\frac{1}{x}+y+z}, \boxed{\frac{1}{x}+y+z}\right\}$

Immediate and Delayed Values

```
In[29]:= color1 = RandomColor[]
```

Out[29]= 🟩

```
In[32]:= color2 := RandomColor[]
```

```
In[33]:= color2
```

Out[33]= 🟪

```
In[34]:= Table[color1, 10]
```

Out[34]= {🟩, 🟩, 🟩, 🟩, 🟩, 🟩, 🟩, 🟩, 🟩, 🟩}

```
In[35]:= Table[color2, 10]
```

Out[35]= {🟩, 🟧, 🟥, 🟦, 🟩, 🟦, 🟫, 🟩, 🟦, 🟥}

```
In[36]:= circles := Graphics[Table[Circle[{x, 0}, x/2], {x, n}]]
```

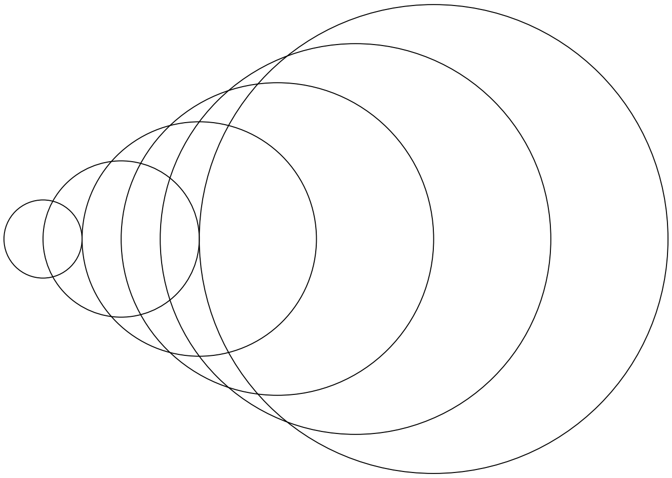
```
In[37]:= n = 6
```

```
Out[37]= 6
```

```
In[38]:= FactorInteger[n]
```

```
In[39]:= circles
```

Out[39]=



```
In[40]:= (* delayed rules *)
         {x, x, x, x} /. x → RandomReal[]

Out[40]= {0.370294, 0.370294, 0.370294, 0.370294}

In[42]:= {x, x, x, x} /. x :→ RandomReal[]

Out[42]= {0.620995, 0.24177, 0.444985, 0.976086}

In[43]:= (* clear names: Clear or name=. *)
         Clear[color1, color2]

In[44]:= circles =.
```

Functions

```
         (* Typically from most specific to least specific. If
            there are definitions that can’t be ordered by specificity,
         definitions made later are put later. When definitions are used,
         the earlier ones are tried first. ?f shows the ordering of definitions for f.*)

In[45]:= f = (#1 + #2 &)

Out[45]= #1 + #2 &

In[46]:= f[1, 2]

Out[46]= 3

In[48]:= g[x_, y_] := x + y

In[49]:= g[1, 2]

Out[49]= 3
```

Datasets

In[51]:=
```
data = Dataset[<|"a" → <|"x" → 1, "y" → 2, "z" → 3|>,
  "b" → <|"x" → 5, "y" → 10, "z" → 7|>|>]
```

Out[51]=

	x	y	z
a	1	2	3
b	5	10	7

In[52]:=
```
data["b", "y"]
```

Out[52]=
```
10
```

In[53]:=
```
data["b"]["y"]
```

Out[53]=
```
10
```

In[54]:=
```
data["b"]
```

Out[54]=

x	5
y	10
z	7

In[55]:=
```
data[All, "z"]
```

Out[55]=

a	3
b	7

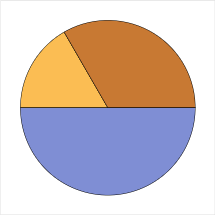
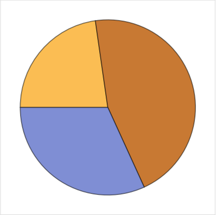
In[56]:=
```
data[All, Total]
```

Out[56]=

a	6
b	22

```
In[57]:= data[All, PieChart]
```

Out[57]=

a	
b	

```
In[59]:= data[All, #x + #y + #z &]
```

Out[59]=

a	6
b	22

```
In[60]:= data[Select[#z > 5 &]]
```

Out[60]=

b	x	5
	y	10
	z	7

```
In[62]:= Clear[f]

In[63]:= data[All, All, f]
```

Out[63]=

	x	y	z
a	f[1]	f[2]	f[3]
b	f[5]	f[10]	f[7]

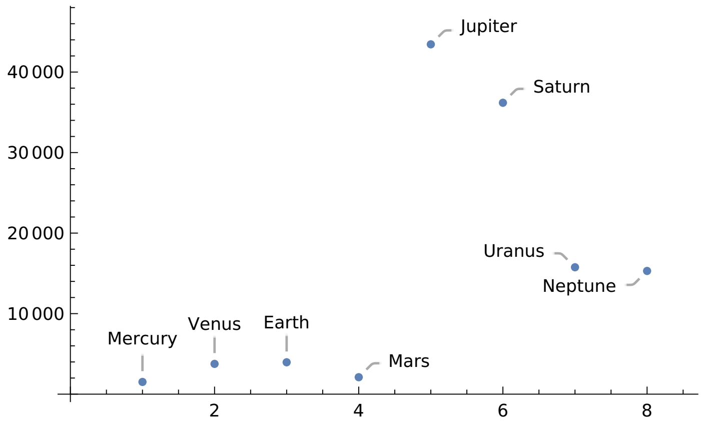
In[64]:=
```
planets = CloudGet["http://wolfr.am/7FxLgPm5"]
```

Out[64]=

	Mass	Radius	Moons		
				Mass	Radius
Mercury	3.30104×10^{23} kg	1516.0 mi			
Venus	4.86732×10^{24} kg	3760.4 mi			
Earth	5.9721986×10^{24} k	3958.761 mi	Moon	7.3459×10^{22} kg	1079.6 mi
Mars	6.41693×10^{23} kg	2106.1 mi	Deimos	1.5×10^{15} kg	3.9 mi
			Phobos	1.072×10^{16} kg	6.90 mi
Jupiter	1.89813×10^{27} kg	43 441. mi	Adrastea	$7. \times 10^{15}$ kg	5.1 mi
			Aitne	$4. \times 10^{13}$ kg	0.93 mi
			69 total		
Saturn	5.68319×10^{26} kg	36 184. mi	Aegaeon	—	0.16 mi
			Aegir	—	1.9 mi
			62 total		
Uranus	8.68103×10^{25} kg	15 759. mi	Ariel	1.35×10^{21} kg	359.7 mi
			Belinda	3.57×10^{17} kg	25.0 mi
			27 total		
Neptune	1.02410×10^{26} kg	15 299. mi	Despina	2.1×10^{18} kg	47. mi
			Galatea	3.7×10^{18} kg	55. mi
			14 total		

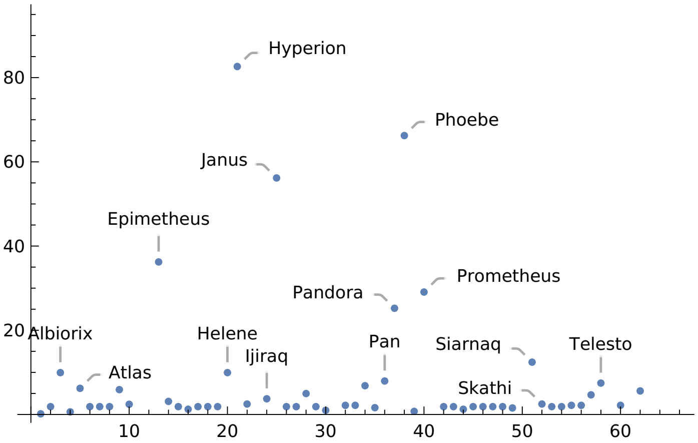
In[66]:=
```
ListPlot[planets[All, "Radius"]]
```

Out[66]=



In[68]:=
```
ListPlot@planets["Saturn", "Moons", All, "Radius"]
```

Out[68]=



In[70]:=
```
planets[All, "Moons", Length]
```

Out[70]=

Mercury	0
Venus	0
Earth	1
Mars	2
Jupiter	69
Saturn	62
Uranus	27
Neptune	14

In[71]:=
```
(f@*g@*h)[x]
```

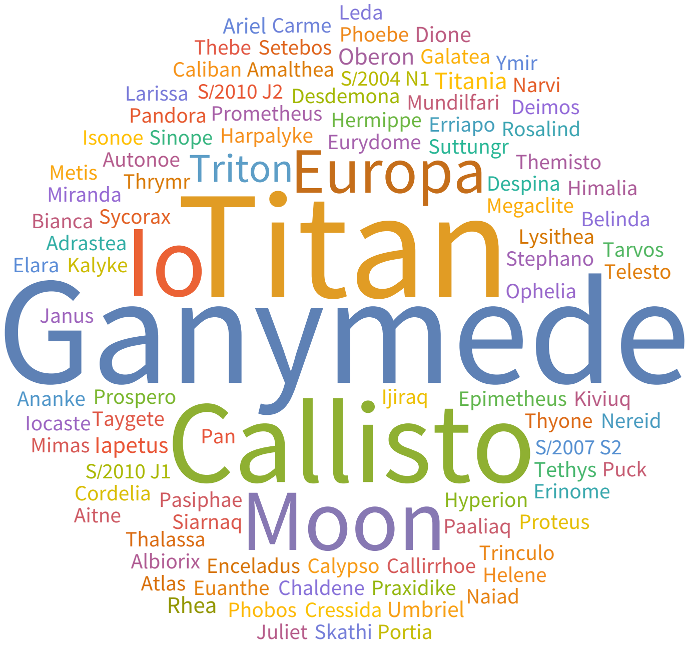
Out[71]=
```
f[g[h[x]]]
```

In[72]:=
```
(h /* g /* f)[x]
```

Out[72]=
```
f[g[h[x]]]
```

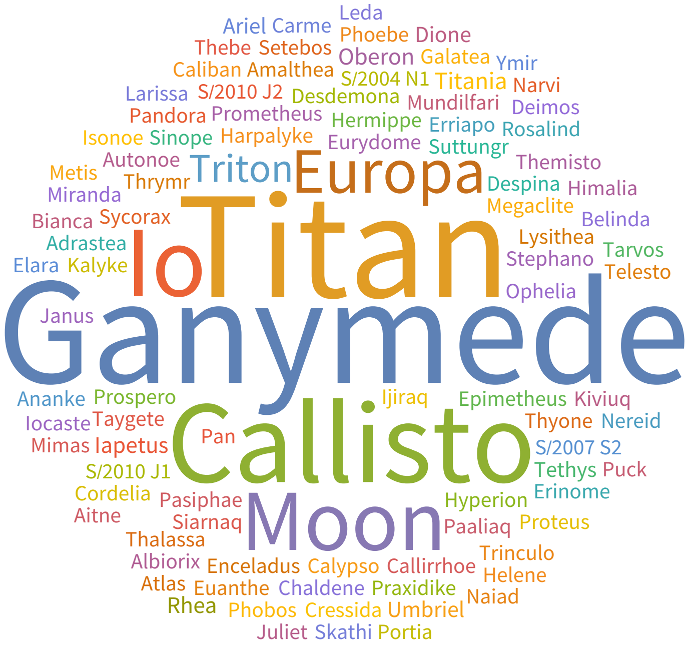
```
In[73]:= planets[WordCloud@*Association@*Values, "Moons", All, "Mass"]
```

Out[73]=



```
In[74]:= planets[Values /* Association /* WordCloud, "Moons", All, "Mass"]
```

Out[74]=



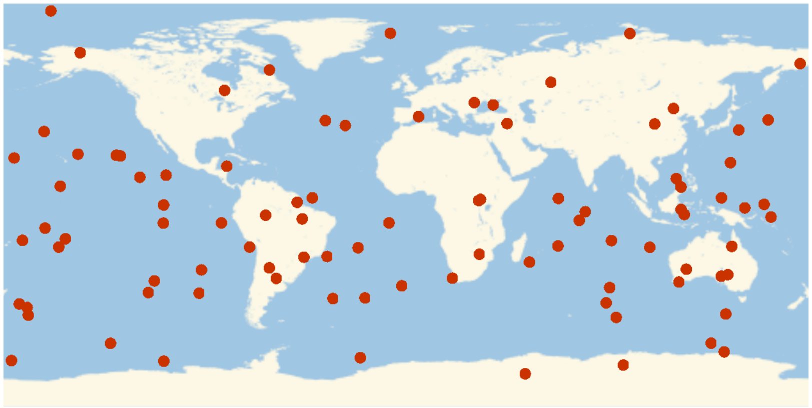
In[75]:=
```
fireballs = ResourceData["Fireballs and Bolides"]
```

PeakBrightness	Coordinates	NearestCity	Altitude	Velocity
October 8, 2009 2:	4.2°S 120.6°E	Bone	19.1 km	19.2 km/s
November 21, 200	22.0°S 29.2°E	Kobojango	38 km	32.1 km/s
December 25, 201	38.0°N 158.0°E	Kurilsk	26 km	18.1 km/s
April 21, 2012 4:08	15.8°S 174.8°W	Hihifo	—	—
April 23, 2012 10:0	36.2°N 107.4°E	Pingliang	25.2 km	—
May 4, 2012 9:54 p	76.7°N 10.6°W	Illoqqortoormiut	—	—
May 15, 2012 11:0	61.8°S 135.5°W	Owenga	33.3 km	—
May 25, 2012 11:3	41.8°S 36.2°W	Grytviken	—	—
July 25, 2012 7:48	36.4°N 41.5°E	Sinjar	26.8 km	—
July 27, 2012 4:19	63.1°N 172.3°E	Anadyr	27.2 km	—
August 26, 2012 2:	11.8°N 117.0°E	El Nido	36 km	—
August 27, 2012 6:	18.3°S 64.2°E	Quatre Cocos	38.7 km	—
September 10, 20	69.8°S 111.7°W	Rothera - permanent	23.8 km	—
September 11, 20	18.9°S 105.2°E	The Settlement	—	—
September 18, 20	1.2°N 52.2°W	Mazagao	28.1 km	—
September 28, 20	6.9°S 73.7°E	Feydhoo	—	—
October 2, 2012 4:	8.1°S 111.9°W	Hanga Roa	35 km	—
October 3, 2012 1	41.5°S 21.9°W	Edinburgh	—	—
October 9, 2012 1	51.2°N 84.6°W	Hearst	27.8 km	—
October 19, 2012	75.4°S 49.6°E	Syowa - permanent s	29.3 km	—

rows 1–20 of **92**

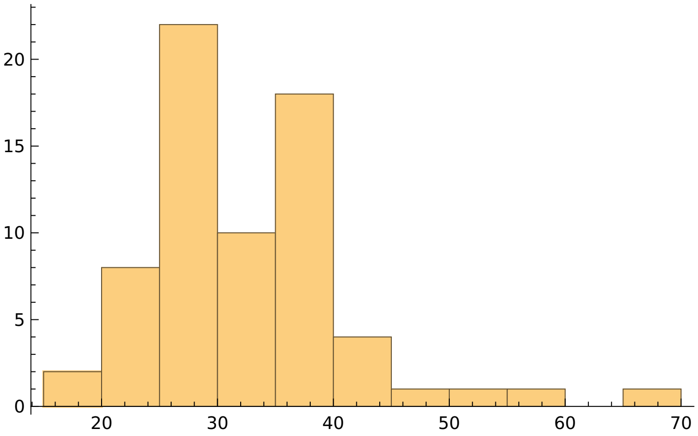
In[76]:=
```
GeoListPlot[fireballs[All, "Coordinates"]]
```

Out[76]=



In[77]:=
```
Histogram[fireballs[All, "Altitude"]]
```

Out[77]=



In[78]:=
```
(* use SemanticImport to generate a dataset object from csv file *)
```