Rania — PS 2 — 2025-01-21

Rania — PS2 (2025-01-21)

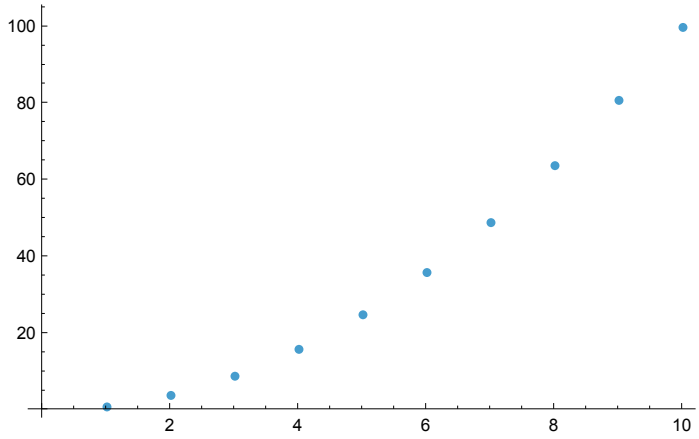
Exercises from EIWL3 Section 5

In[•]:=
```
(*5.1*)Reverse[Range[10]^2]
(*5.2*)Total[Range[10]^2]
(*5.3*)ListPlot[Range[10]^2]
(*5.4*)Sort[Join[Range[4], Range[4]]]
(*5.5*)Range[10, 20] (*what do they mean by +?*)
(*5.6*)Sort[Join[Range[5]^2, Range[5]^3]]
(*5.7*)IntegerLength[2^128]
(*5.8*)First[IntegerDigits[2^128]]
(*5.9*)Take[IntegerDigits[2^100], 10]
(*5.10*)Max[IntegerDigits[2^20]]
(*5.11*)Count[IntegerDigits[2^1000], 0]
(*5.12*)Part[Sort[IntegerDigits[2^20]], 2]
(*5.13*)ListLinePlot[IntegerDigits[2^128]]
(*5.14*)Drop[Take[Range[100], 20], 10]
```

Out[•]=
```
{100, 81, 64, 49, 36, 25, 16, 9, 4, 1}
```

Out[•]=
```
385
```

Out[•]=



Out[•]=
```
{1, 1, 2, 2, 3, 3, 4, 4}
```

Out[•]=
```
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

Wolfram was looking for Range[11]+9 which gets the same result.

Please put one exercise per cell. It makes it a load easier for me to compare with my solution. Yours uses impressively less space though! :)

Very nice. See comments above and on next two pages.

10/10

Out[•]=
```
{1, 1, 4, 8, 9, 16, 25, 27, 64, 125}
```

Out[•]=
```
39
```

Out[•]=
```
3
```

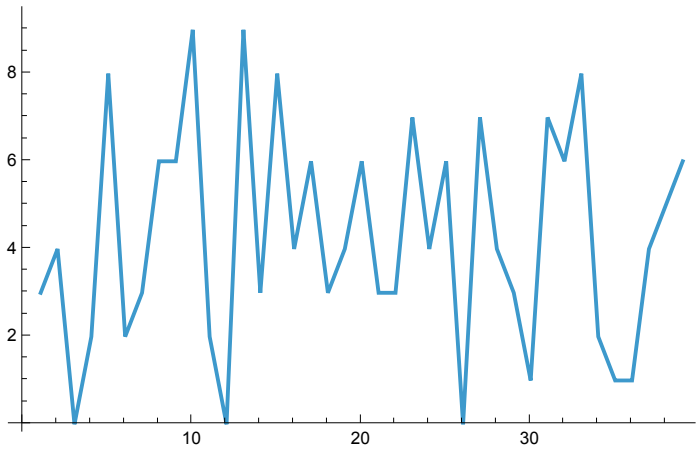
Out[•]=
```
{1, 2, 6, 7, 6, 5, 0, 6, 0, 0}
```

Out[•]=
```
8
```

Out[•]=
```
28
```

Out[•]=
```
1
```

Out[•]=



Out[•]=
```
{11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

Exercises from EIWL3 Section 6

In[•]:=
```
(*6.1*)Table[1000, 5]
(*6.2*)Table[n^3, {n, 10, 20}]
(*6.3*)NumberLinePlot[Table[n^2, {n, 20}]]
(*6.4*)Range[2, 20, 2]
(*6.5*)Table[n, {n, 10}] == Range[10]
(*6.6*)BarChart[Table[n^2, {n, 10}]]
(*6.7*)IntegerDigits[Table[n^2, {n, 10}]]
(*6.8*)ListLinePlot[Table[Length[IntegerDigits[n^2]], {n, 100}]]
(*6.9*)Table[First[IntegerDigits[n^2]], {n, 20}]
(*6.10*)ListLinePlot[First[IntegerDigits[n^2]], {n, 100}]
```

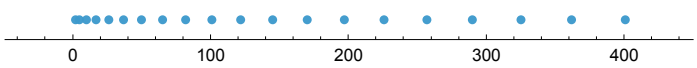
Out[•]=
```
{1000, 1000, 1000, 1000, 1000}
```

For 6.10 you need a Table[] function call.

Out[ ]=

{1000, 1331, 1728, 2197, 2744, 3375, 4096, 4913, 5832, 6859, 8000}

Out[ ]=



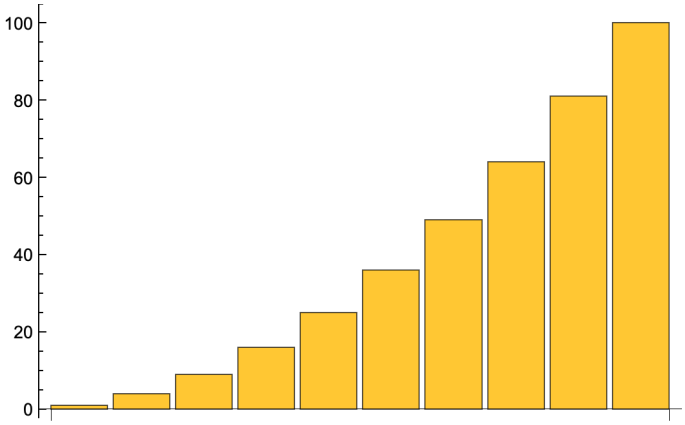
Out[ ]=

{2, 4, 6, 8, 10, 12, 14, 16, 18, 20}

Out[ ]=

True

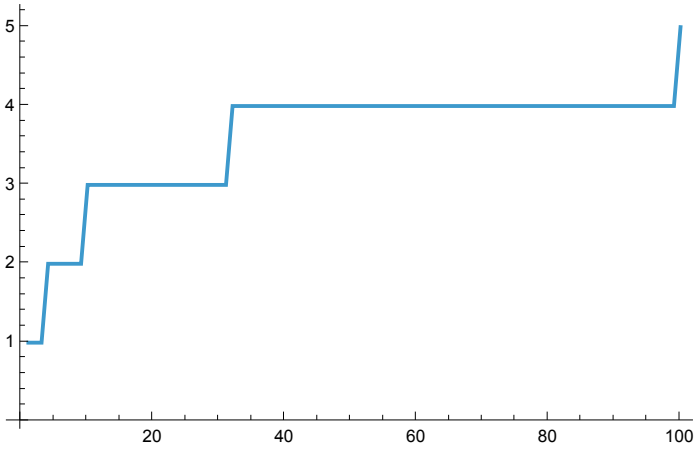
Out[ ]=



Out[ ]=

{{1}, {4}, {9}, {1, 6}, {2, 5}, {3, 6}, {4, 9}, {6, 4}, {8, 1}, {1, 0, 0}}

Out[ ]=



Out[ ]=

{1, 4, 9, 1, 2, 3, 4, 6, 8, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 4}

ListLinePlot: Options expected (instead of {n, 100}) beyond position 1 in ListLinePlot[n^2, {n, 100}]. An option must be a rule or a list of rules.

Out[ ]=

ListLinePlot[n^2, {n, 100}]

6.10 barfed an error message.

Exercises from EIWL3 Section 7

In[ ]:=
```
(*7.1*){Red, Yellow, Green}
(*7.2*)Column[{Red, Yellow, Green}]
(*7.3*)ColorNegate[Orange]
(*7.4*)Table[Hue[n], {n, 0, 1, 0.02}]
(*7.5*)Table[RGBColor[1, G, 1], {G, 0, 1, 0.05}]
(*7.6*)Blend[{Pink, Yellow}]
(*7.7*)Table[Blend[{Yellow, Hue[n]}], {n, 0, 1, 0.05}]
(*7.8*)Table[Style[n, Hue[n]], {n, 0, 1, 0.1}]
(*7.9*)Style[Purple, 100]
(*7.10*)Table[Style[Red, x], {x, 10, 100, 10}]
(*7.11*)Style[999, 100, Red]
(*7.12*)Table[Style[x^2, x^2], {x, 10}]
(*7.13*)Part[{Red, Yellow, Green}, RandomInteger[{1, 3}, 100]]
(*7.14*) Table[
 Style[Part[IntegerDigits[2^1000], n], 3 Part[IntegerDigits[2^1000], n]], {n, 50}]
```

Out[ ]=

Out[ ]=

Out[ ]=

Out[ ]=

Out[ ]=

Out[ ]=

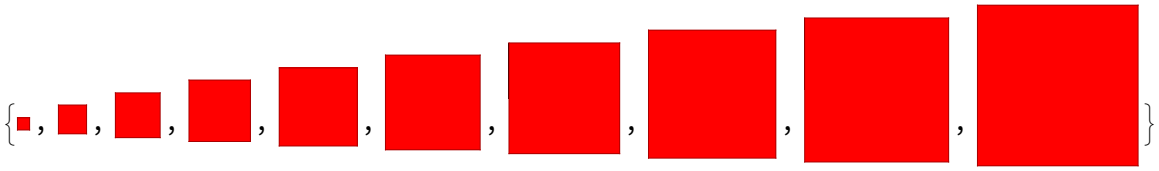
Out[ ]=

Out[ ]=

Out[ ]=

Out[ ]=

Out[•]=



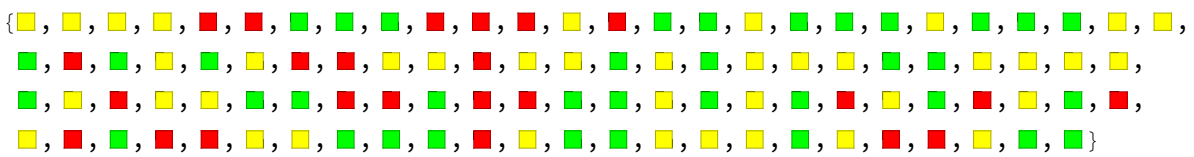
Out[•]=

999

Out[•]=

{, 4, 9, 16, 25, 36, 49, 64, 81, 100}

Out[•]=



Out[•]=

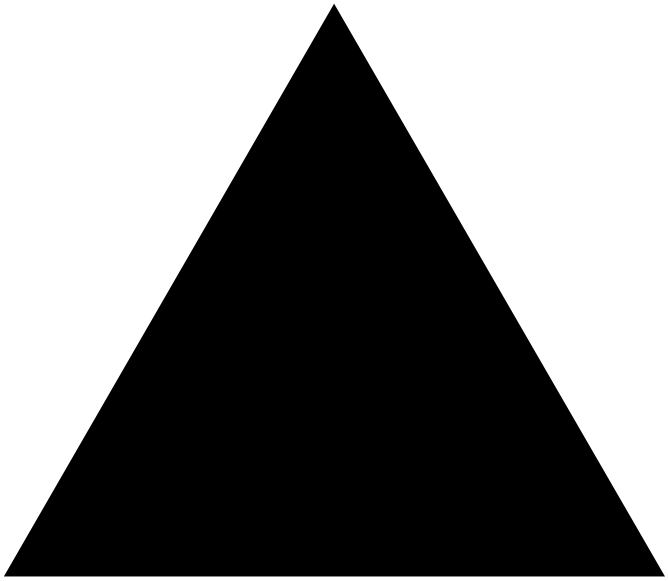
{1, , 7, 1, 5, , 8, 6, , 7, 1, 8, 6, 2, 6, 7, 3, 2, , 9, 4, 8, 4, 2, 5,
, 4, 9, , 6, , , , 1, 8, 1, , 5, 6, 1, 4, , 4, 8, 1, 1, 7, , 5, 5}

Exercises from EIWL3 Section 8

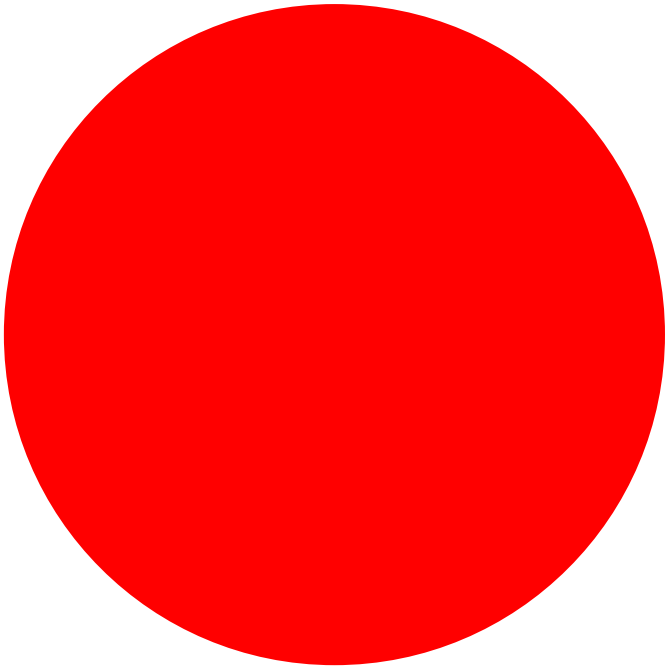
In[•]:=

```
(*8.1*)Graphics[RegularPolygon[3]]
(*8.2*)Graphics[{Red, Disk[]}]
(*8.3*)Graphics[{Red, RegularPolygon[8]}]
(*8.4*)Table[Graphics[Style[Disk[], Hue[n]]], {n, 0, 1, 0.1}]
(*8.5*)Column[{Graphics[Style[RegularPolygon[3], Red]],
  Graphics[Style[RegularPolygon[3], Green]]}]
(*8.6*)Table[Graphics[Style[RegularPolygon[n], Pink]], {n, 5, 10}]
(*8.7*)Graphics3D[{Purple, Cylinder[]}]
(*8.8*)Graphics[Reverse[Table[Style[RegularPolygon[n], RandomColor[]], {n, 3, 8}]]]
```

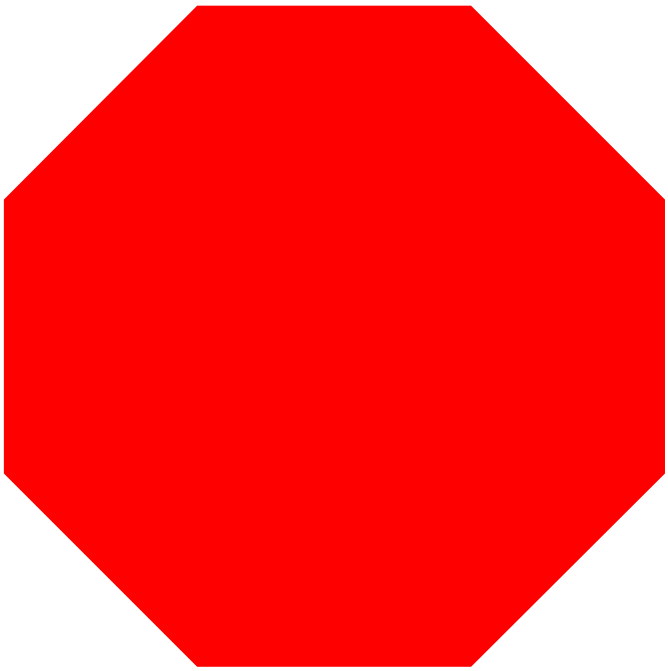
Out[ ]=



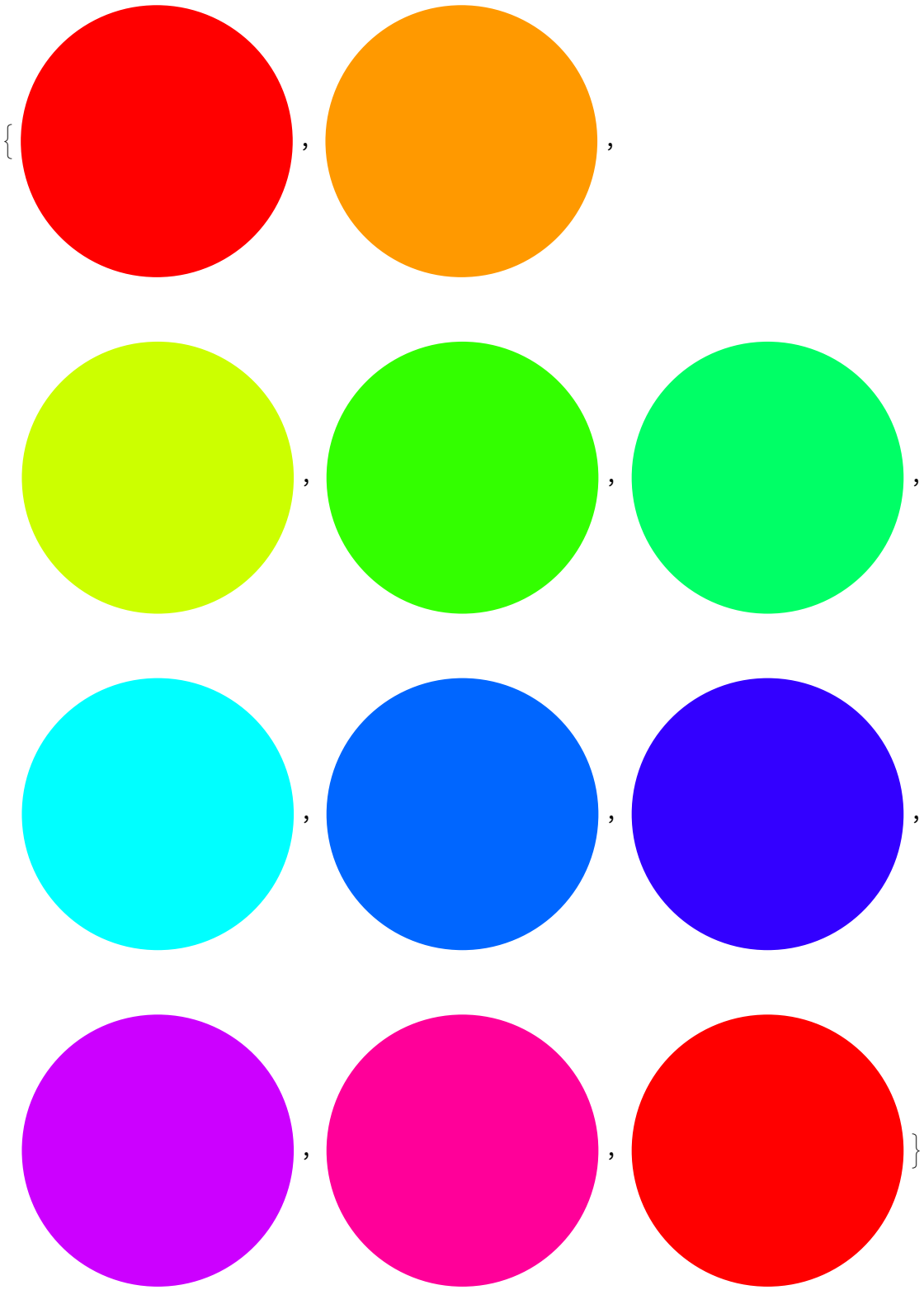
Out[ ]=



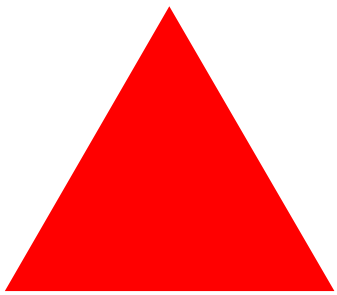
Out[ ]=

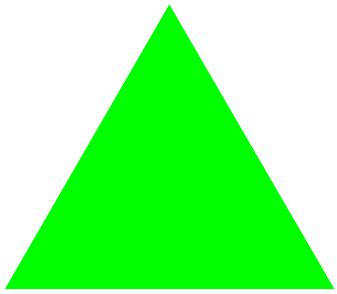


Out[ ]=

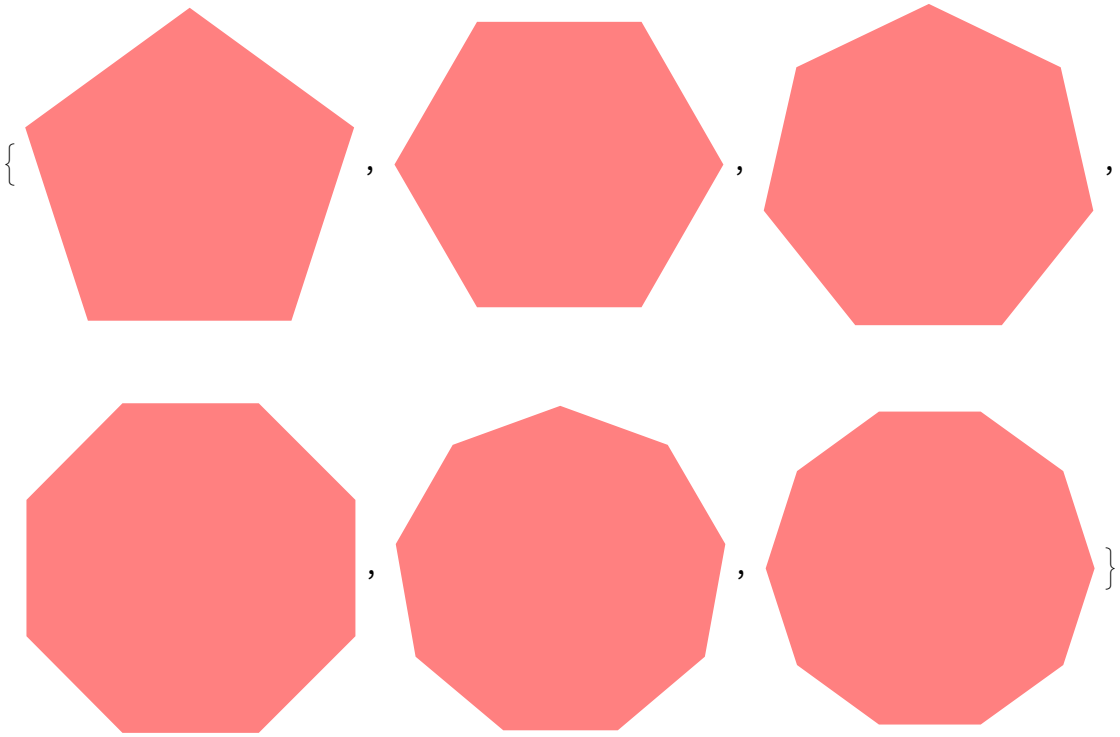


Out[ ]=

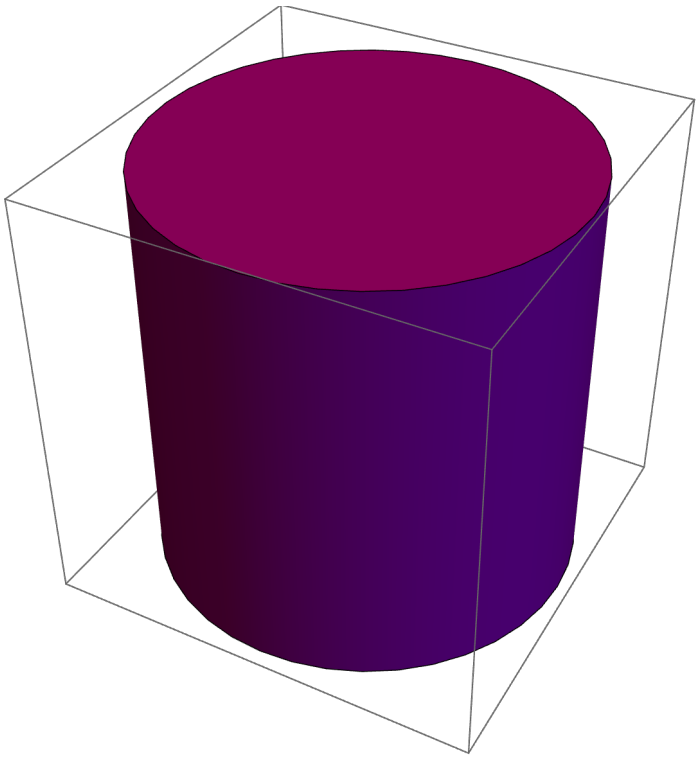




Out[ ]=



Out[ ]=



Out[ ]=