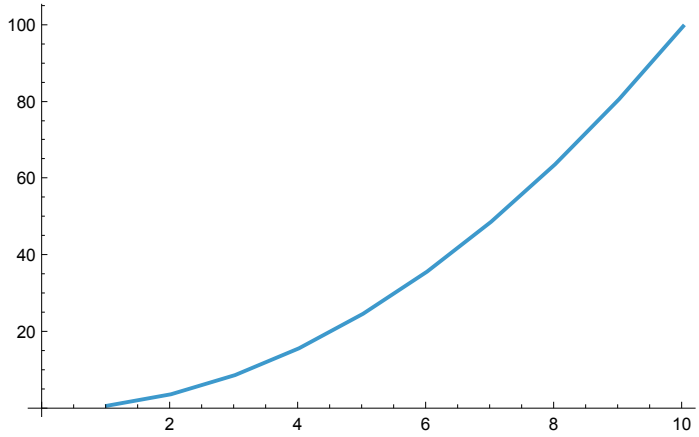
Problem Set 2, EIWL Sections 5-8 - Harper Yonago

Section 5

```
In[•]:= Reverse[Range[10]^2]
Out[•]=
{100, 81, 64, 49, 36, 25, 16, 9, 4, 1}

In[•]:= Total[Range[10]^2]
Out[•]=
385

In[•]:= ListLinePlot[Range[10]^2]
Out[•]=
```




```
In[•]:= Sort[Join[Range[4], Range[4]]]
Out[•]=
{1, 1, 2, 2, 3, 3, 4, 4}

In[•]:= Range[10, 15 + 5]
Out[•]=
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

In[•]:= {10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
(*Not sure if this was what the excersize intended,
although it is technically correct*)
Out[•]=
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

In[•]:= Sort[Join[Range[5]^2, Range[5]^3]]
Out[•]=
{1, 1, 4, 8, 9, 16, 25, 27, 64, 125}
```

Very nice. Minor comments below and on p. 3.

10/10

Definitely not what he was looking for, because it would become a real pain if you had to do 50, 100, or 1000 numbers. I did Range[10, 20, 1] and Range[11] + 9 as my two solutions.

In[•]:=
```
Length[IntegerDigits[2^128]]
```
Out[•]=
```
39
```

In[•]:=
```
First[IntegerDigits[2^32]]
```
Out[•]=
```
4
```

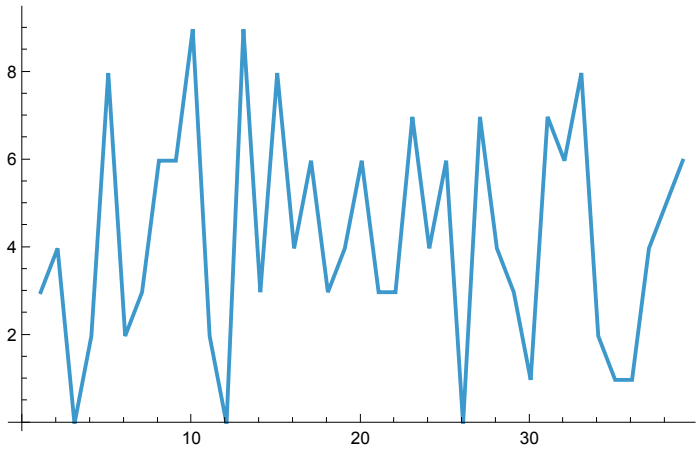
In[•]:=
```
Take[IntegerDigits[2^100], 10]
```
Out[•]=
```
{1, 2, 6, 7, 6, 5, 0, 6, 0, 0}
```

In[•]:=
```
Max[IntegerDigits[2^20]]
```
Out[•]=
```
8
```

In[•]:=
```
Count[IntegerDigits[2^1000], 0]
```
Out[•]=
```
28
```

In[•]:=
```
Part[Sort[IntegerDigits[2^20]], 2]
```
Out[•]=
```
1
```

In[•]:=
```
ListLinePlot[IntegerDigits[2^128]]
```
Out[•]=



In[•]:=
```
Drop[Take[Range[100], 20], 10]
```
Out[•]=
```
{11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

Section 6

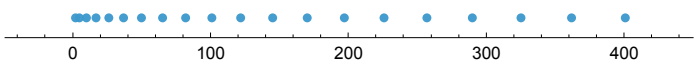
In[•]:=
```
Table[1000, 5]
```
Out[•]=
```
{1000, 1000, 1000, 1000, 1000}
```

In[ ]:=
```
Table[n^3, {n, 10, 20}]
```

Out[ ]=
```
{1000, 1331, 1728, 2197, 2744, 3375, 4096, 4913, 5832, 6859, 8000}
```

In[ ]:=
```
NumberLinePlot[Range[20]^2]
```

Out[ ]=



In[ ]:=
```
Range[2, 20, 2]
```

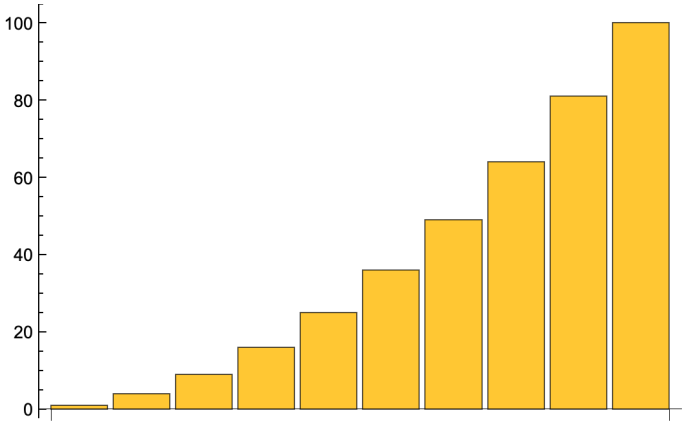
Out[ ]=
```
{2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
```

In[ ]:=
```
Table[n, {n, 10}]
```

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

In[ ]:=
```
BarChart[Range[10]^2]
```

Out[ ]=



In[ ]:=
```
IntegerDigits[Table[n^2, {2, 10}]]
```

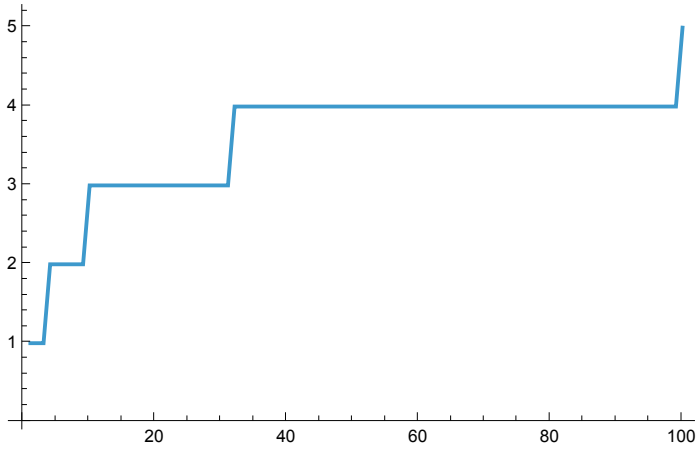
Table: Raw object 2 cannot be used as an iterator.

Out[ ]=
```
IntegerDigits[Table[n^2, {2, 10}]]
```

Oops. You meant Table[IntegerDigits[n^2, {n, 10}]

In[•]:=
```
ListLinePlot[Table[Length[IntegerDigits[n^2]], {n, 100}]]
```

Out[•]=



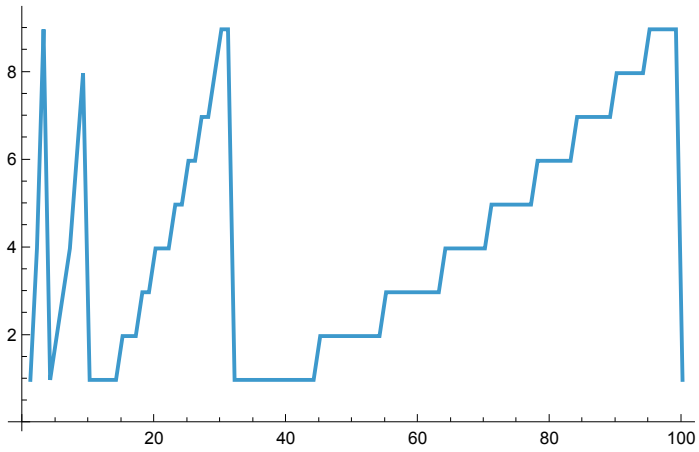
In[•]:=
```
Table[First[IntegerDigits[n^2]], {n, 20}]
```

Out[•]=

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

In[•]:=
```
ListLinePlot[Table[First[IntegerDigits[n^2]], {n, 100}]]
```

Out[•]=



Section 7

In[•]:=
```
{Red, Yellow, Green}
```

Out[•]=

{■, ■, ■}

In[•]:=
```
Column[{Red, Yellow, Green}]
```

Out[•]=

■
■
■

In[•]:=
```
ColorNegate[Orange]
```
Out[•]=

In[•]:=
```
Table[Hue[x], {x, 0, 1, 0.02}]
```
Out[•]=

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

In[•]:=
```
Blend[{Pink, Yellow}]
```
Out[•]=

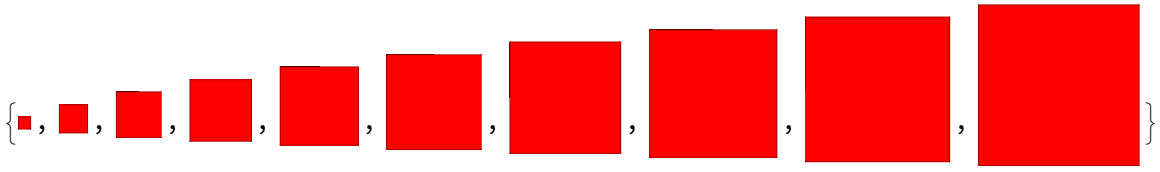
In[•]:=
```
Table[Blend[{Yellow, Hue[x]}], {x, 0, 1, 0.05}]
```
Out[•]=

In[•]:=
```
Table[Style[x, Hue[x]], {x, 0, 1, 0.1}]
```
Out[•]=

{0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.}

In[•]:=
```
Style[Purple, 100]
```
Out[•]=

In[•]:=
```
Table[Style[Red, x], {x, 10, 100, 10}]
```
Out[•]=



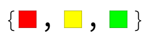
In[•]:=
```
Style[999, Red, 100]
```
Out[•]=

999

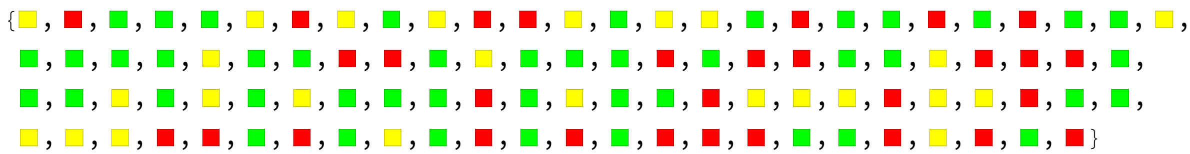
In[∘]:=
```
Table[Style[n^2, n^2], {n, 10}]
```
Out[∘]=

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

In[∘]:=
```
Part[{Red, Yellow, Green}]
```
Out[∘]=



In[∘]:=
```
Table[Part[{Red, Yellow, Green}, RandomInteger[{1, 3}]], 100]
```
Out[∘]=

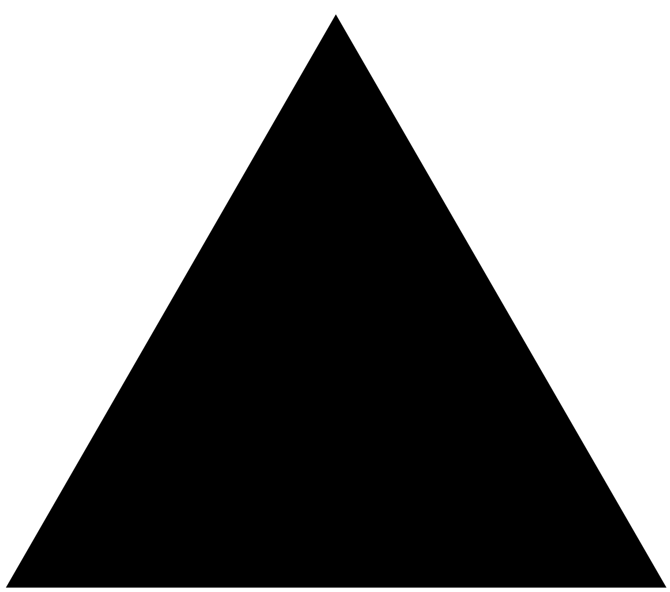


In[∘]:=
```
Table[Style[Part[IntegerDigits[2^1000], n],
  3*Part[IntegerDigits[2^1000], n]], {n, 50}]
```
Out[∘]=

{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}

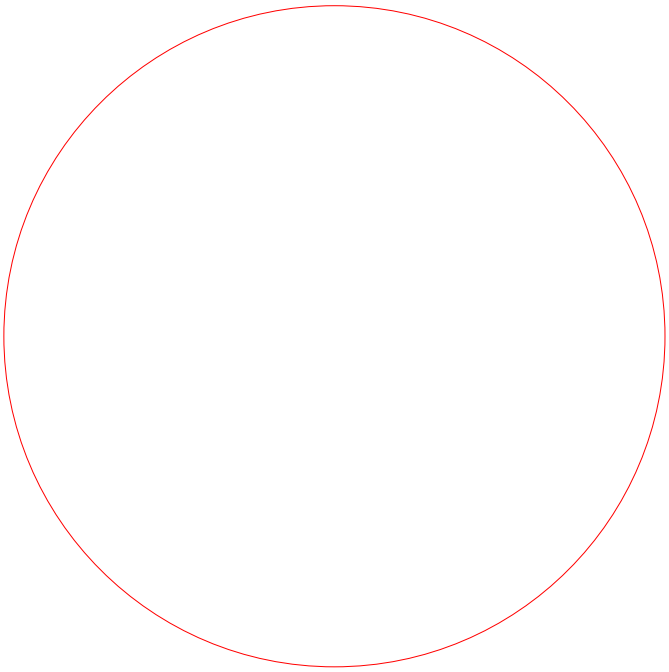
Section 8

In[∘]:=
```
Graphics[RegularPolygon[3]]
```
Out[∘]=



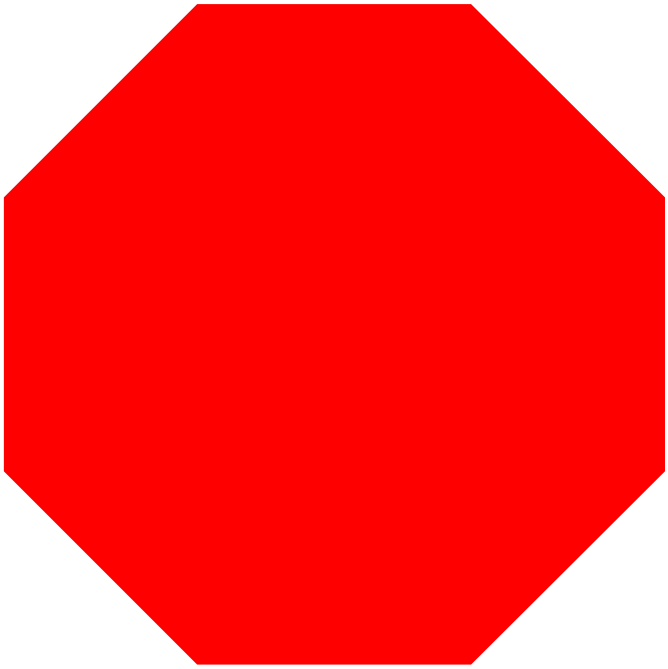
In[ ]:=
```
Graphics[Style[Circle[], Red]]
```

Out[ ]=



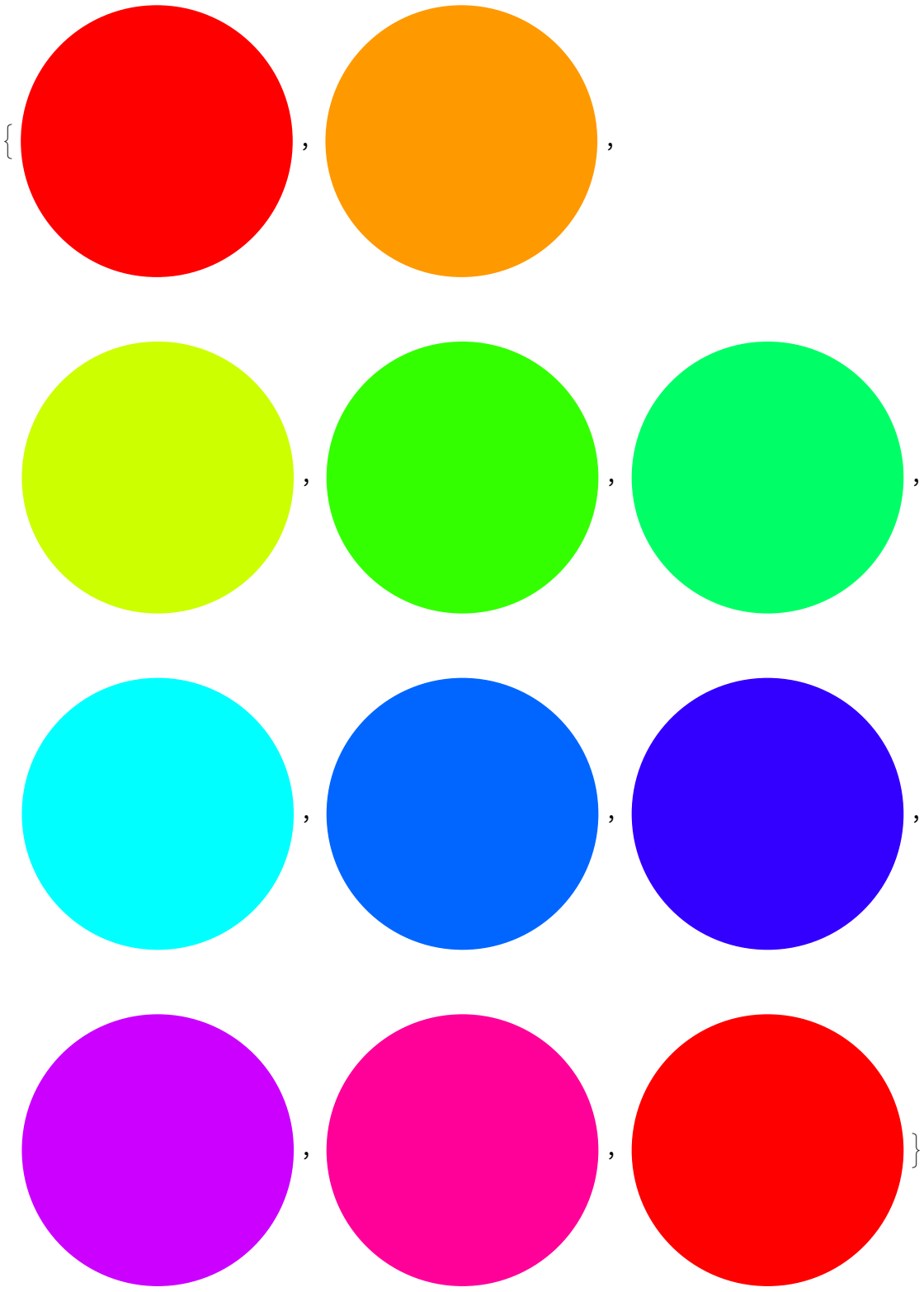
In[ ]:=
```
Graphics[Style[RegularPolygon[8], Red]]
```

Out[ ]=



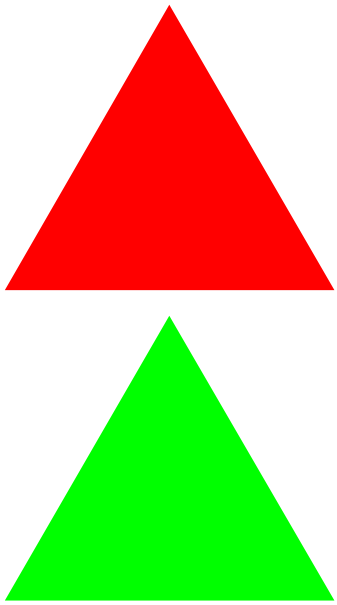
In[•]:=
```
Table[Graphics[Style[Disk[], Hue[x]]], {x, 0, 1, 0.1}]
```

Out[•]=



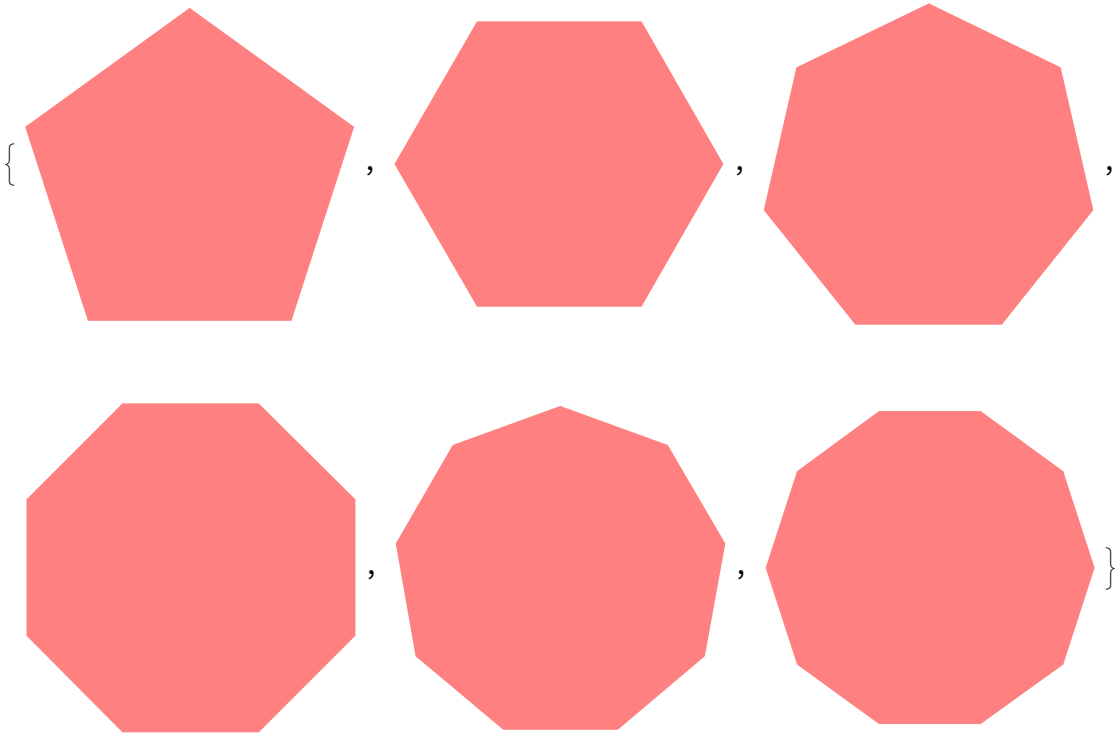
In[•]:=
```
Column[{Graphics[Style[RegularPolygon[3], Red]],
  Graphics[Style[RegularPolygon[3], Green]]}]
```

Out[•]=



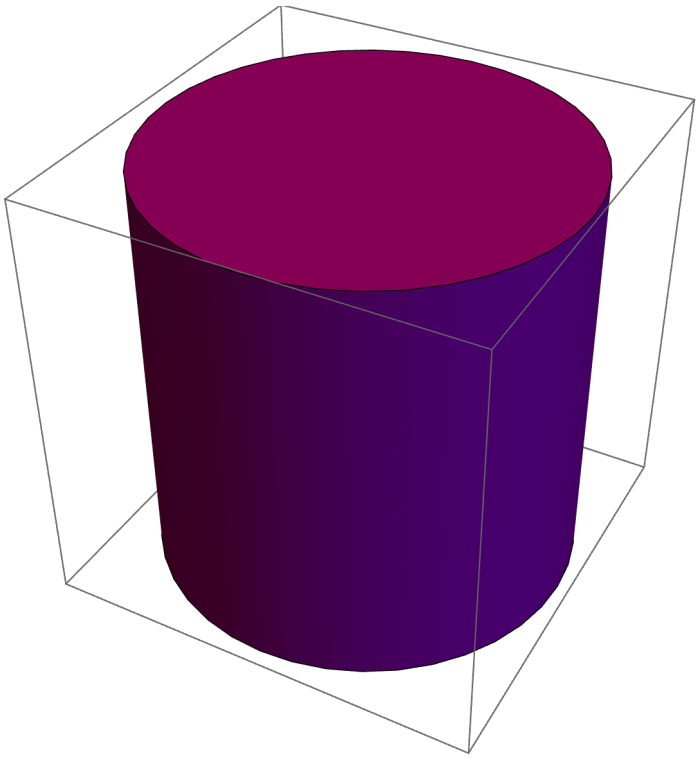
In[•]:=
```
Table[Graphics[Style[RegularPolygon[x], Pink]], {x, 5, 10}]
```

Out[•]=



In[•]:=
```
Graphics3D[Style[Cylinder[], Purple]]
```

Out[•]=



In[•]:=
```
Graphics[Reverse[Table[Style[RegularPolygon[x], RandomColor[]], {x, 3, 8}]]]
```

Out[•]=