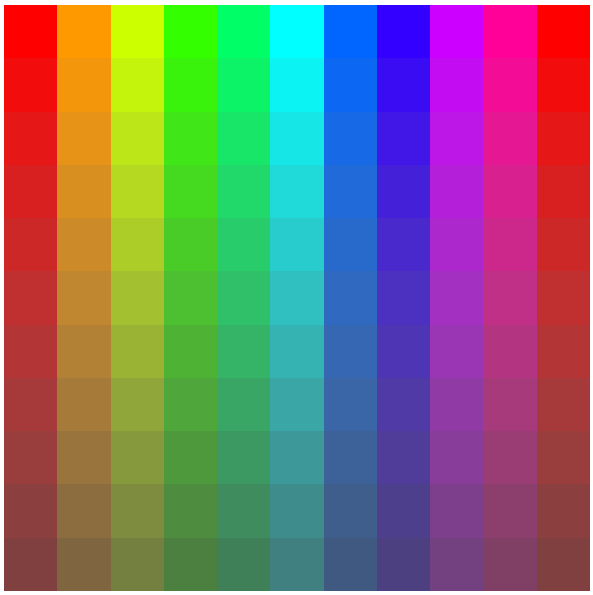
■ More about Color and Rates of Convergence

Let's look more closely at the function **Hue**.

In[1]:=
```
?? Hue
```

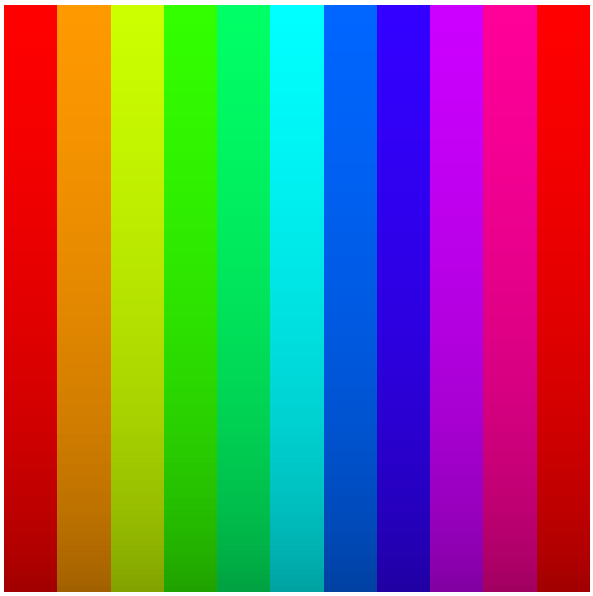
```
Hue[h] is a graphics directive which specifies that graphical objects which follow
  are to be displayed, if possible, in a color corresponding to hue h. Hue[h, s,
  b] specifies colors in terms of hue, saturation and brightness.

Attributes[Hue] = {Protected}
```

In[2]:=
```
Show[Graphics[RasterArray[
   Table[Hue[h, (s + 1) / 2, (s + 1) / 2],
         {s, 0, 1, 0.1}, {h, 0, 1, 0.1}]]],
 AspectRatio -> 1]
```



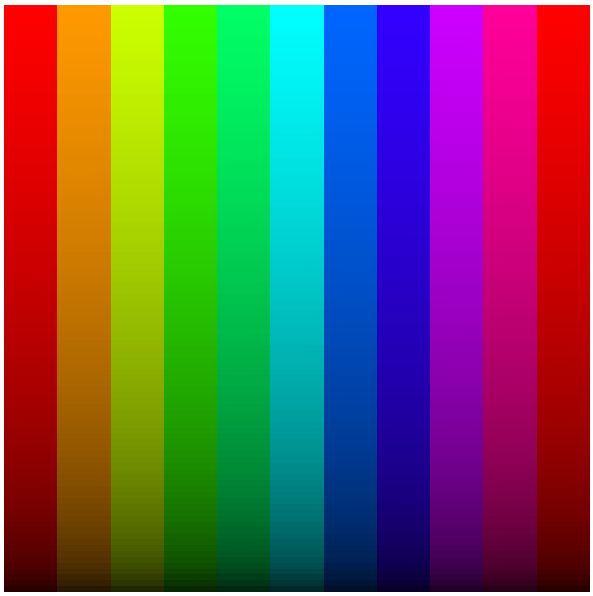
Out[2]=
```
- Graphics -
```

```
In[3]:= Show[Graphics[RasterArray[
           Table[Hue[h, 1, Log[120, s + 20]],
                 {s, 1, 100}, {h, 0, 1, 0.1}]]],
         AspectRatio -> 1]
```



```
Out[3]= - Graphics -
```

```
In[26]:= Show[Graphics[RasterArray[
            Table[Hue[h, 1, Exp[0.4 Log[s / 100]]],
                  {s, 1, 100}, {h, 0, 1, 0.1}]]],
          AspectRatio -> 1]
```



```
Out[26]= - Graphics -
```

To measure the rate of escape, there are several methods. The iterative:

```
In[8]:= FPRate[F_, x_] := Module[
          {y = x, i = 1},
          While[F[y] != y && i < 100, y = F[y]; ++i];
          {y, i}]

In[5]:= NMh[z_] := z - (z^3 - 1) / (3 z^2)

In[6]:= Simplify[NMh[z]]

Out[6]= 1/(3 z^2) + (2 z)/3

In[9]:= Timing[FPRate[NMh, -0.01 + 0.01 I]]

Out[9]= {0.03 Second, {-0.5 + 0.866025 I, 26}}
```

The recursive:

```
In[10]:= FPRateRec[F_, x_, n___] :=
          {Last[#], Length[#]}&[FixedPointList[F, x, n]]

In[11]:= Timing[FPRateRec[NMh, 0.01 + 0.01 I]]

Out[11]= {0.01 Second, {-0.5 - 0.866025 I, 27}}
```

And the compiled:

```
In[12]:= FPRc = Compile[{{z, _Complex}},
           Module[{w1, w2, i},
           w1 = z; w2 = z + 1.0; i = 0;
           While[w1 != w2 && (++i) < 100,
             w2 = w1; w1 = 1 / (3 w2^2) + 2 w2 / 3];
            {w1, i}]]


Out[12]= CompiledFunction[{z}, Module[{w1, w2, i}, w1 = z; w2 = z + 1.;
           i = 0; While[w1 ≠ w2 && ++i < 100, w2 = w1; w1 = 1/(3 w2^2) + (2 w2)/3]; {w1, i}],
          -CompiledCode-]

In[13]:= Timing[Apply[{#1, Re[#2]}&,
           FPRc[0.01 + 0.01 I]]]

Out[13]= {0.01 Second, {-0.5 - 0.866025 I, 27.}}
```

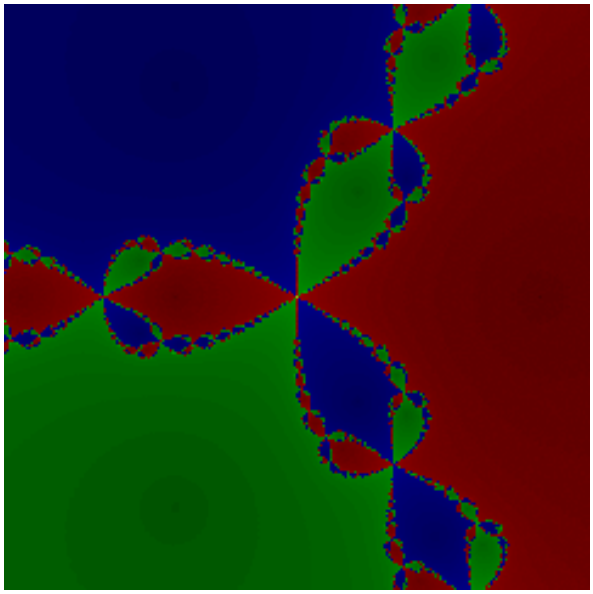
Obviously the compiled is the fastest.

```
In[14]:= Timing[data2 = Table[FPRc[x + I y],
          {y, 1.2, -1.2, -0.01}, {x, -1.2, 1.2, 0.01}];]

Out[14]= {41.9 Second, Null}


        RootToColor[{z_, n_}] :=
         Hue[Arg[z] / (2 Pi), 1,
          Exp[0.3 Log[Re[n / 100.]]]]

In[16]:= data2[[1, 1]]

Out[16]= {-0.5 + 0.866025 I, 8. + 0. I}
```

In[17]:=
```
RootToColor[%]
```

Out[17]=
```
Hue[0.333333, 1, 0.603418]
```

In[29]:=
```
Show[Graphics[RasterArray[Map[RootToColor, data2, {2}]]],
 AspectRatio -> 1]
```



Out[29]=
```
- Graphics -
```