

```

In[72]:= (* 300 Different Pseudocolor Schemes via One RGB Orthonormalization *)
(* Copyright May 2, 2006 Doug Youvan www.youvan.com www.pseudocolor.com *)
(* This needs to be rewritten without hard coding and hotmap can be imported *)

(* Permutaions of Tuples, then filtered for direction,
done by hand, yielding 300 non-grey-black,
rgb values for orthonormalization bases. This is equivalent to 300
different rgb axes, all of which should have nondegenerate rows *)

Tuples[{0.5, 1.0}, 3]
{{0.5`, 0.5`, 0.5`}, {0.5`, 0.5`, 1.`}, {0.5`, 1.` , 0.5`}, {0.5`, 1.` , 1.`},
{1.` , 0.5`, 0.5`}, {1.` , 0.5`, 1.`}, {1.` , 1.` , 0.5`}, {1.` , 1.` , 1.`}

(*
6,5,4,3,2,1,
5,4,3,2,1
4,3,2,1
3,2,1
2,1,
1

6,10,12,12,10,6 →56
*)

56 * 6
8! / (8 - 3) !
(*      a      b      c
d      e      f      g      h *)

(* multiples of each other can not be in the same pixel: {a , h}, no other
examples because nothing can be doubled or halved to make another triplet;
a and h can be thought of as pointing in the same direction : gray - black *)

336 - 6 * 6 (* remove 6 a,h permutations, below *)

perms = Join [
  Permutations[{a, b, c}], Permutations[{a, d, b}], Permutations[{a, b, e}],
  Permutations[{a, b, f}], Permutations[{a, b, g}], (*Permutations[{a,b,h}],*)

  Permutations[{a, c, d}], Permutations[{a, c, e}],
  Permutations[{a, c, f}], Permutations[{a, c, g}], (*Permutations[{a,c,h}],*)

  Permutations[{a, d, e}], Permutations[{a, d, f}],
  Permutations[{a, d, g}], (*Permutations[{a,d,h}],*)

  Permutations[{a, e, f}], Permutations[{a, e, g}], (*Permutations[{a,e,h}],*)

  Permutations[{a, f, g}], (*Permutations[{a,f,h}],*)

  (*Permutations[{a,g,h}],*)

```

```

Permutations[{b, c, d}], Permutations[{b, c, e}],
Permutations[{b, c, f}], Permutations[{b, c, g}], Permutations[{b, c, h}],

Permutations[{b, d, e}], Permutations[{b, d, f}],
Permutations[{b, d, g}], Permutations[{b, d, h}],

Permutations[{b, e, f}], Permutations[{b, e, g}], Permutations[{b, e, h}],

Permutations[{b, f, g}], Permutations[{b, f, h}],

Permutations[{b, g, h}],

Permutations[{c, d, e}], Permutations[{c, d, f}],
Permutations[{c, d, g}], Permutations[{c, d, h}],

Permutations[{c, e, f}], Permutations[{c, e, g}], Permutations[{c, e, h}],

Permutations[{c, f, g}], Permutations[{c, f, h}],

Permutations[{c, g, h}],

Permutations[{d, e, f}], Permutations[{d, e, g}], Permutations[{d, e, h}],

Permutations[{d, f, g}], Permutations[{d, f, h}],

Permutations[{d, g, h}],

Permutations[{e, f, g}], Permutations[{e, f, h}],

Permutations[{e, g, h}],

Permutations[{f, g, h}]
];

```

```
Tuples[{0.5, 1.0}, 3]
```

```
a = {0.5, 0.5, 0.5}
```

```
b = {0.5, 0.5, 1.}
```

```
c = {0.5, 1., 0.5}
```

```
d = {0.5, 1., 1.}
```

```
e = {1., 0.5, 0.5}
```

```
f = {1., 0.5, 1.}
```

```
g = {1., 1., 0.5}
```

```
h = {1., 1., 1.}
```

```
MatrixForm[perms];
```

```
Dimensions[perms]
```

```
Show[Graphics[RasterArray[Apply[RGBColor, perms, {2}]]], ImageSize -> {600, 300}]
```

```
Out[72]= {{0.5, 0.5, 0.5}, {0.5, 0.5, 1.}, {0.5, 1., 0.5}, {0.5, 1., 1.},
{1., 0.5, 0.5}, {1., 0.5, 1.}, {1., 1., 0.5}, {1., 1., 1.}}
```

```
Out[73]= {{0.5, 0.5, 0.5}, {0.5, 0.5, 1.}, {0.5, 1., 0.5}, {0.5, 1., 1.},
{1., 0.5, 0.5}, {1., 0.5, 1.}, {1., 1., 0.5}, {1., 1., 1.}}
```

```
Out[74]= 336
```

```
Out[75]= 336
```

```
Out[76]= 300
```

```
Out[78]= {{0.5, 0.5, 0.5}, {0.5, 0.5, 1.}, {0.5, 1., 0.5}, {0.5, 1., 1.},
          {1., 0.5, 0.5}, {1., 0.5, 1.}, {1., 1., 0.5}, {1., 1., 1.}}
```

```
Out[79]= {0.5, 0.5, 0.5}
```

```
Out[80]= {0.5, 0.5, 1.}
```

```
Out[81]= {0.5, 1., 0.5}
```

```
Out[82]= {0.5, 1., 1.}
```

```
Out[83]= {1., 0.5, 0.5}
```

```
Out[84]= {1., 0.5, 1.}
```

```
Out[85]= {1., 1., 0.5}
```

```
Out[86]= {1., 1., 1.}
```

```
Out[88]= {300, 3, 3}
```



```
Out[89]= - Graphics -
```

```
In[90]:=
```

```
perms[[1]] (* each row, can be a new r,g,b set for ortho;
           these are all nonzero, nongray, and orthogonal rgb values *)
perms[[300, 3]] (* This addresses the blue value in the 300 th, or last, row *)
```

```
Out[90]= {{0.5, 0.5, 0.5}, {0.5, 0.5, 1.}, {0.5, 1., 0.5}}
```

```
Out[91]= {1., 0.5, 1.}
```

```
In[107]:=
```

```
(* hotmap import should go here *)
```

```

ramp = Table[{
  Piecewise[{
    {1, 1 <= i < 42},
    {1, 42 <= i < 84},
    {Floor[5.8 * (i - 83)], 84 <= i < 128},
    {256, 128 <= i < 170},
    {256, 170 <= i < 212},
    {256, 212 <= i <= 256}
  ]},
  Piecewise[{
    {1, 1 <= i < 42},
    {(i - 41) * 6, 42 <= i < 84},
    {256, 84 <= i < 128},
    {256 - (i - 128) * 6, 128 <= i < 170},
    {1, 170 <= i < 212},
    {Floor[(i - 211) * 5.7], 212 <= i <= 256}
  ]},
  Piecewise[{
    {Floor[i * 5.8], 1 <= i < 42},
    {256 - ((i - 42) * 6), 42 <= i < 84},
    {1, 84 <= i < 128},
    {1, 128 <= i < 170},
    {(i - 169) * 6, 170 <= i < 212},
    {256, 212 <= i <= 256}
  ]}
],
{i, 1, 256, 1}];

ramp[[1, All]] = 1; (* This changes 1,1,5 to 1,1,
  1 at i=1 for the blue channel ; Use of All is overkill *)

(* i=1;While[(i=i+10)< 256,ramp[[i,All]]=128];
  This sets neutral gray every 10 steps in the psedocolor scheme *)

Export["D:\\Mathematica\\Data\\schroedinger3ramp.dat", ramp];

Clear[bar];
(*bar = Table[{j, j, j},{i,1,10},{j,1,256}]; *)
bar = Table[{j, j, j}, {i, 1, 1}, {j, 1, 256}];
bar[[1, 256]] = {1, 1, 1}; (* This places black in
  the bottom pixel of the last (white) column of the graybar *)

size = Dimensions[bar];
height = size[[1]];
width = size[[2]];
depth = size[[3]];
pv1 = Table[{Part[bar, i, j, 1]}, {i, 1, height}, {j, 1, width}];
pv3 = Table[{ramp[[pv1[[i, j]]]}], {i, 1, height}, {j, 1, width}];
pv4 = Flatten[pv3, 3];
pv5 = Partition[pv4, width];
Dimensions[pv5]
pv6 = Apply[RGBColor, pv5 / 256, {2}];
pv7 = RasterArray[pv6];

```

Out[120]=

```
{1, 256, 3}
```

In[123]:=

```
(* newrgbpixels= {{0.6,0.4,0.4},{0.4,0.6,0.4},{0.4,0.4,0.6}}; *)
(* newrgbpixels= {{1.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0}}; *)
newrgbpixels = {{0.0, 0.0, 1.0}, {0.0, 1.0, 0.0}, {1.0, 0.0, 0.0}};
MatrixForm[newrgbpixels];
inv = Inverse[newrgbpixels];
MatrixForm[inv];
check = newrgbpixels . inv;
MatrixForm[check];
ortho = pv5.inv;
MatrixForm[ortho];
(* absortho = Abs[ortho]/256; *) (* comment out absolute value *)
absortho = ortho / 256;

aplot = Flatten[Table[{Part[absortho, 1, j, All]}, {j, 1, 256}], 1];
Dimensions[ramp]
Dimensions[absortho]
Dimensions[aplot]
{256, 3}
{10, 256, 3}
{256, 3}
Show[
  Graphics3D[
    Line[
      Append[ramp, First[ramp] (* Append needs to
        be fixed so that the first and last point are not linked *)
      ]],
    ImageSize → {200, 200}, AxesStyle → RGBColor[0, 1, 0], BoxStyle → RGBColor[0, 1, 0]];
Show[
  Graphics3D[
    Line[
      Append[aplot, First[aplot] (* Append needs to
        be fixed so that the first and last point are not linked *)
      ]],
    ImageSize → {200, 200}, AxesStyle → RGBColor[0, 1, 0], BoxStyle → RGBColor[0, 1, 0]];

Show[Graphics[pv7, ImageSize → {400, 200}, AspectRatio → Automatic]];
Show[Graphics[RasterArray[Apply[RGBColor, bar / 256, {2}]]],
  ImageSize → {400, 200}, AspectRatio → Automatic];

Show[Graphics[RasterArray[Apply[RGBColor, absortho, {2}]]],
  ImageSize → {400, 200}, AspectRatio → Automatic]
```

Out[133]=

```
{256, 3}
```

Out[134]=

```
{1, 256, 3}
```

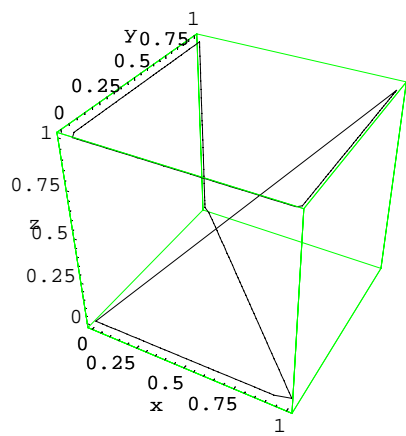
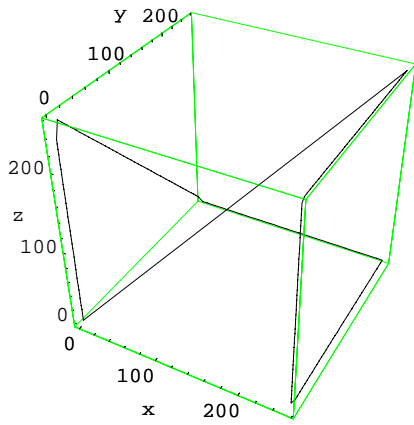
Out[135]=

```
{256, 3}
```

```
Out[136]=  
{256, 3}
```

```
Out[137]=  
{10, 256, 3}
```

```
Out[138]=  
{256, 3}
```





Out[143]=  
- Graphics -

```

In[144]:=
(*Now start making changes from pseudoortho 2 to enter
newrgbpixel values from the 300 x 3 matrix constructed at top *)

newrgbpixels = perms[[107]]

inv = Inverse[newrgbpixels];
ortho = pv5.inv;
(* absortho = Abs[ortho]/256;
either comment out absolute value or the next line *)
absortho = ortho/256;

aplot = Flatten[Table[{Part[absortho, 1, j, All]}, {j, 1, 256}], 1];

Show[Graphics3D[Line[Append[ramp, First[ramp]]], ImageSize -> {200, 200},
AxesStyle -> RGBColor[0, 1, 0], BoxStyle -> RGBColor[0, 1, 0]];

Show[Graphics3D[Line[Append[aplot, First[aplot]]], ImageSize -> {200, 200},
AxesStyle -> RGBColor[0, 1, 0], BoxStyle -> RGBColor[0, 1, 0]];

Show[Graphics[pv7, ImageSize -> {400, 200}, AspectRatio -> Automatic]];

Show[Graphics[RasterArray[Apply[RGBColor, bar/256, {2}]]],
ImageSize -> {400, 200}, AspectRatio -> Automatic];

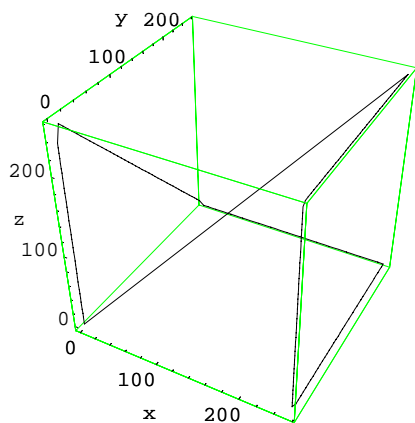
Show[Graphics[RasterArray[Apply[RGBColor, absortho, {2}]]],
ImageSize -> {400, 200}, AspectRatio -> Automatic];

```

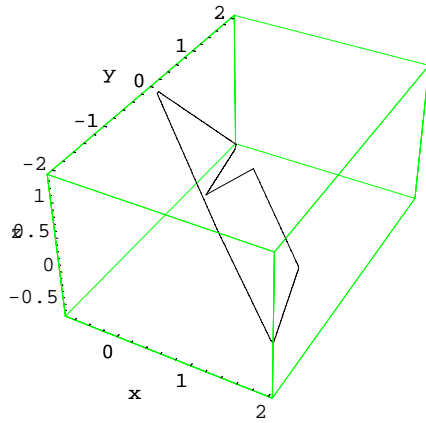
```

Out[144]=
{{1., 0.5, 1.}, {0.5, 0.5, 1.}, {0.5, 1., 0.5}}

```







```
Graphics::realu :  
Argument in RGBColor[-0.0234375, 0.078125, -0.0234375] is not a real number between 0 and 1. More...
```

```
Graphics::realu :  
Argument in RGBColor[-0.0234375, 0.078125, -0.0234375] is not a real number between 0 and 1. More...
```

```
Graphics::realu :  
Argument in RGBColor[-0.0390625, 0.125, -0.0390625] is not a real number between 0 and 1. More...
```

```
General::stop : Further output of Graphics::realu will be suppressed during this calculation. More...
```



In[154]:=

```
SetOptions[Graphics3D, AmbientLight → GrayLevel[1], AspectRatio → Automatic,
  Axes → True, AxesEdge → Automatic, AxesLabel → {x, y, z}, AxesStyle → Automatic,
  Background → Automatic, Boxed → True, BoxRatios → Automatic,
  BoxStyle → Automatic, ColorOutput → Automatic, DefaultColor → RGBColor[0, 0, 0],
  DefaultFont → $DefaultFont, DisplayFunction → $DisplayFunction, Epilog → {},
  FaceGrids → None, FormatType → $FormatType, ImageSize → Automatic,
  Lighting → True, LightSources → {{{1.`, 0.`, 1.`, RGBColor[1, 0, 0]},
  {{1.`, 1.`, 1.`, RGBColor[0, 1, 0]}, {{0.`, 1.`, 1.`, RGBColor[0, 0, 1]}}},
  Plot3Matrix → Automatic, PlotLabel → None, PlotRange → Automatic,
  PlotRegion → Automatic, PolygonIntersections → True, Prolog → {},
  RenderAll → True, Shading → True, SphericalRegion → False,
  TextStyle → $TextStyle, Ticks → Automatic, ViewCenter → Automatic,
  ViewPoint → {1.3`, -2.4`, 2.`, ViewVertical → {0.`, 0.`, 1.``};
```

```
(* now try to build a 300 line image wherein every line is original
  psedocoloring of gray bar changed by 300 orthonormalizations *)
```

```
(* I have been simulatenously working in 'string1.nb' to make an animated gif,
  but I can not change the filename in the Export command,
  even with 'StringReplace',
  so I have opted here for a single export of one image with unique rows. *)
```

```
(* I can't get this to work with the filename in Export,
  because the name of the file path is in " ". *)
```

```
(*For[start, test, incr, body] executesstart,
  then repeatedly evaluatesbodyandincruntiltestfails to give True. *)
```

In[155]:=

```
ortho = pv5.inv; (* otherwise ortho keeps growing every time this cell is run *)
For[i = 1, i < 300, ortho = Join[ortho, nextortho];
  nextortho = (pv5.(PseudoInverse[perms[[i]])) / 256; i++]
Dimensions[perms]
Dimensions[perms[[1]]]
Dimensions[pv5]
Dimensions[ortho]
```

Out[156]=

```
{300, 3, 3}
```

Out[157]=

```
{3, 3}
```

Out[158]=

```
{1, 256, 3}
```

Out[159]=

```
{300, 256, 3}
```

In[160]:=

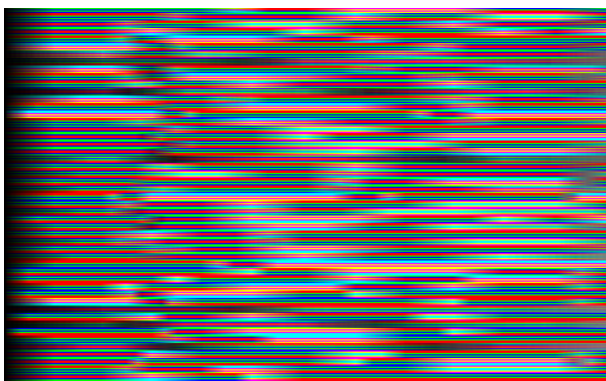
```
gortho = RasterArray[Apply[RGBColor, ortho, {2}]];
Show[Graphics[gortho], ImageSize -> {300, 256}]
```

Graphics::realu : Argument in RGBColor[-6., 20., -6.] is not a real number between 0 and 1. More...

Graphics::realu : Argument in RGBColor[-6., 20., -6.] is not a real number between 0 and 1. More...

Graphics::realu : Argument in RGBColor[-6., 20., -6.] is not a real number between 0 and 1. More...

General::stop : Further output of Graphics::realu will be suppressed during this calculation. More...



Out[161]=

- Graphics -

```
(* "C:\\Documents and Settings\\Doug\\Desktop
  \\Mathematica\\Pseudocolor\\gortho.tif" 288 x 177 ? *)
```

```
(* change 300 to 3 for testing *)
```

```
(* For[i=1,i<3,ortho = (pv5.(PseudoInverse[perms[[i]])))/256;
  gortho=RasterArray[Apply[RGBColor, ortho, {2}]];
  Show[Graphics[gortho]];Print[i, i++] *)
```

In[162]:=

```
name = Table[i, {i, 1, 301}];
(*Head[name]
  Head[name[[3]]]
  ToString[name[[3]]]
  Head[ToString[name[[3]]]])*)
```

In[163]:=

```
For[i = 1, i < 300, ortho = (pv5.(PseudoInverse[perms[[i]])) / 256;
  gortho = RasterArray[Apply[RGBColor, ortho, {2}]];
  Show[Graphics[gortho]]; Print[i]; Export[
    "D:\\Mathematica\\Data\\" <> ToString[name[[i]]] <> ".gif", gortho, "GIF"]; i++];
```

Graphics::realu :

Argument in RGBColor[-0.0703125, 0.078125, -2.51535×10<sup>-17</sup>] is not a real number between 0 and 1. More...

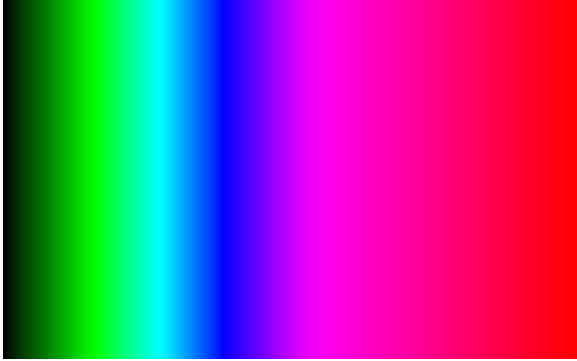
Graphics::realu :

Argument in RGBColor[-0.117187, 0.125, -4.0766×10<sup>-17</sup>] is not a real number between 0 and 1. More...

Graphics::realu :

Argument in RGBColor[-0.164062, 0.171875, -5.63785×10<sup>-17</sup>] is not a real number between 0 and 1. More...

General::stop : Further output of Graphics::realu will be suppressed during this calculation. More...



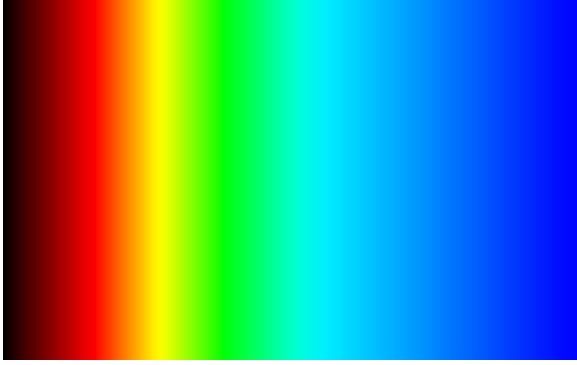
1



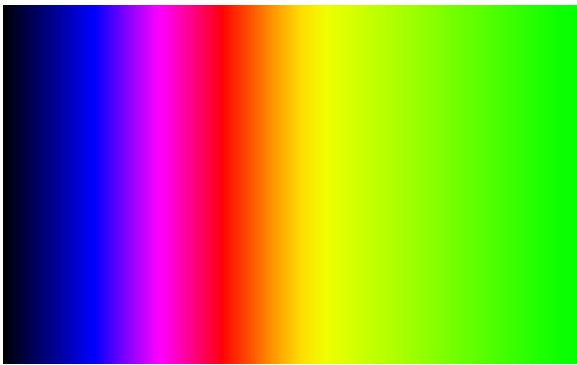
2



3



4



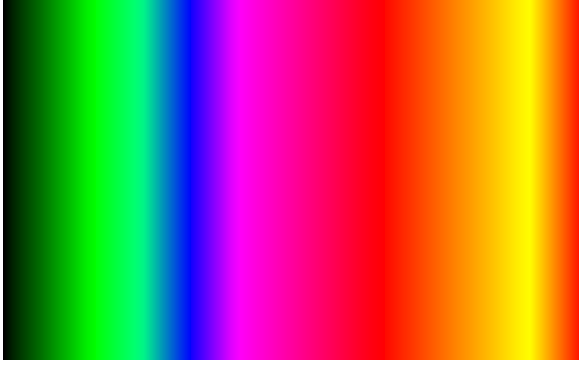
5



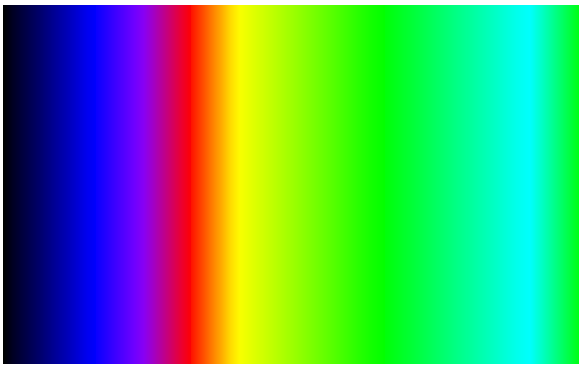
6



7



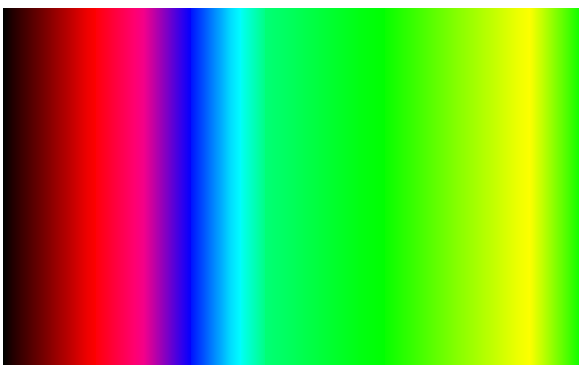
8



9



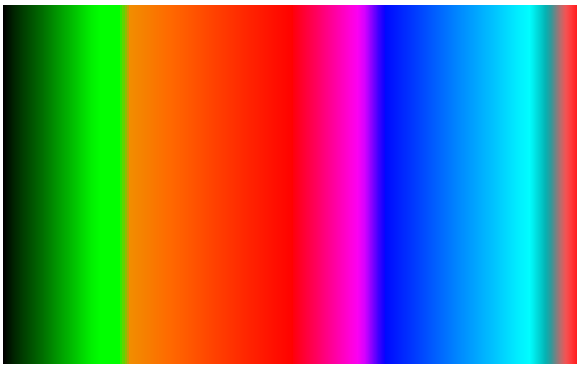
10



11



12



13

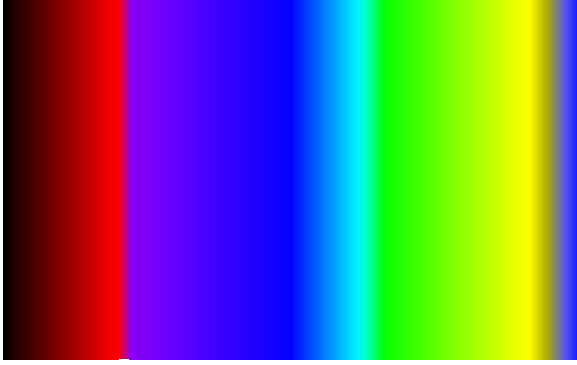


14

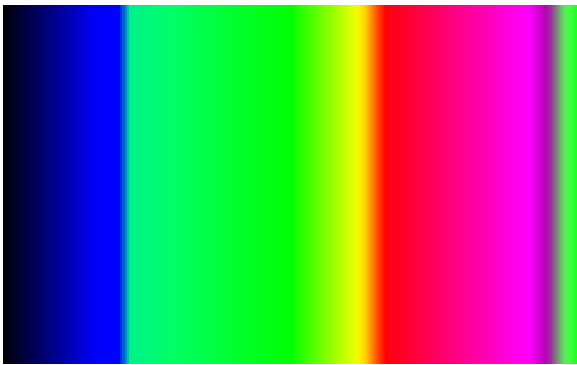


15

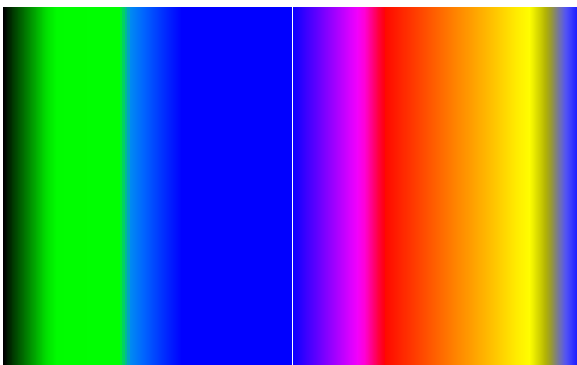




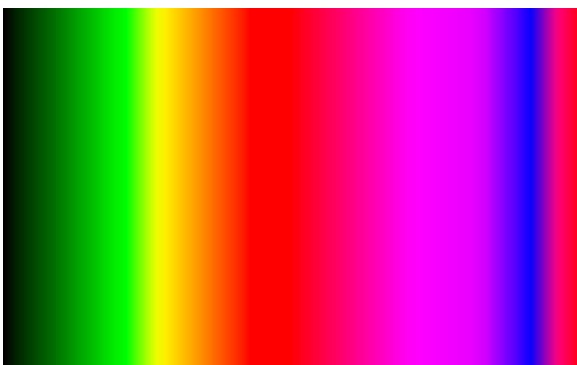
16



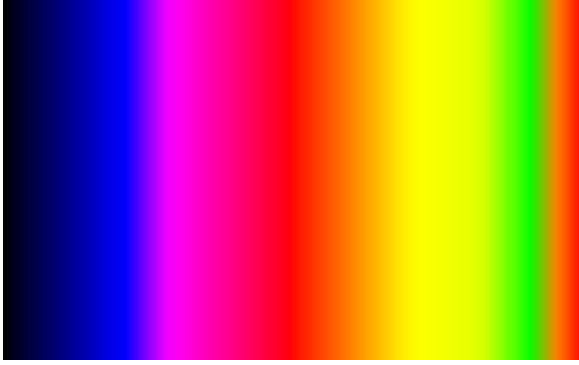
17



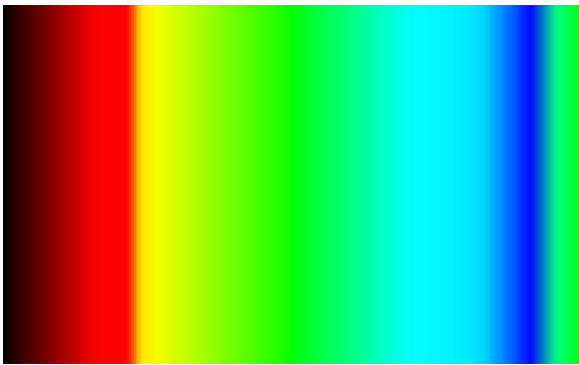
18



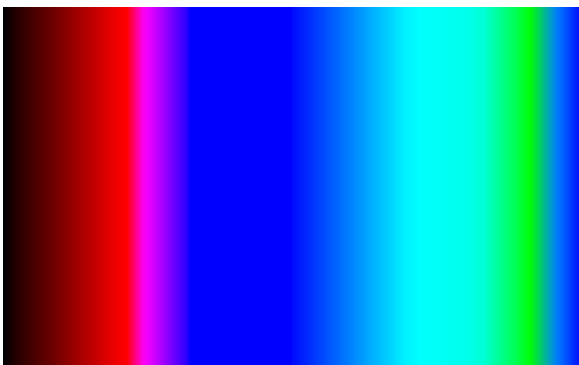
19



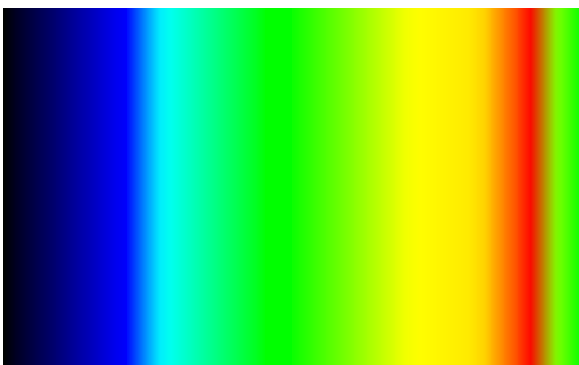
20



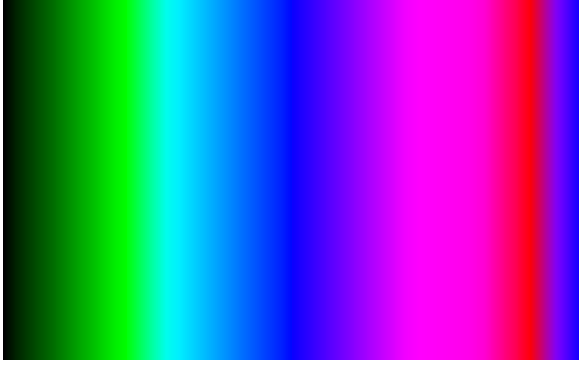
21



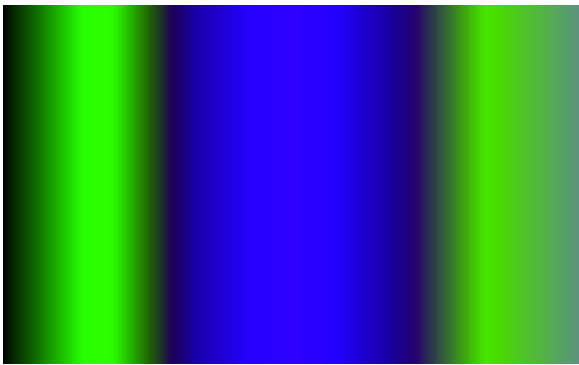
22



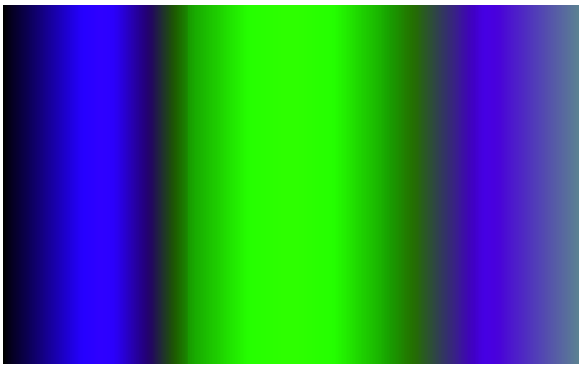
23



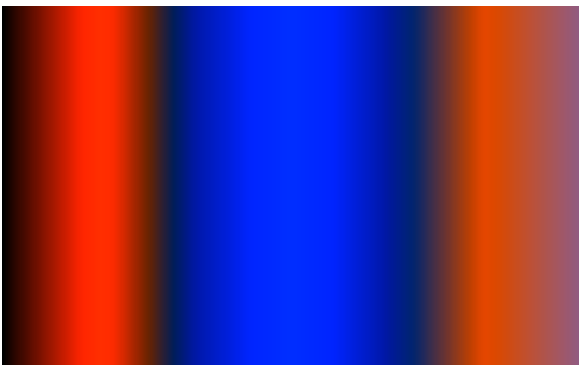
24



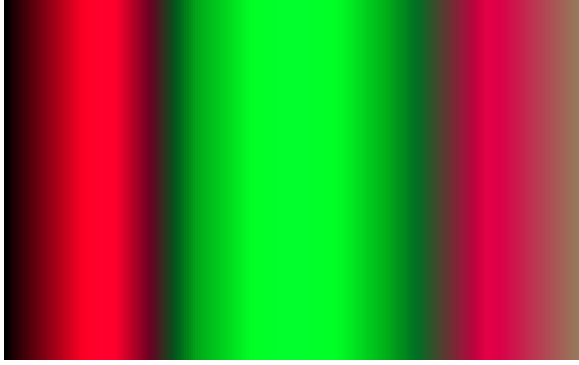
25



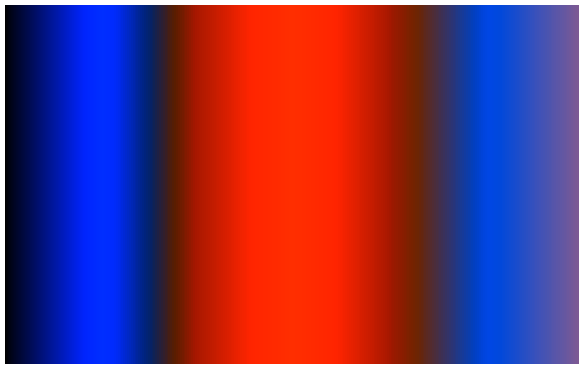
26



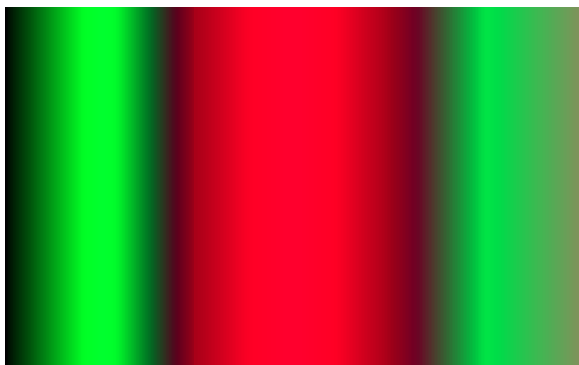
27



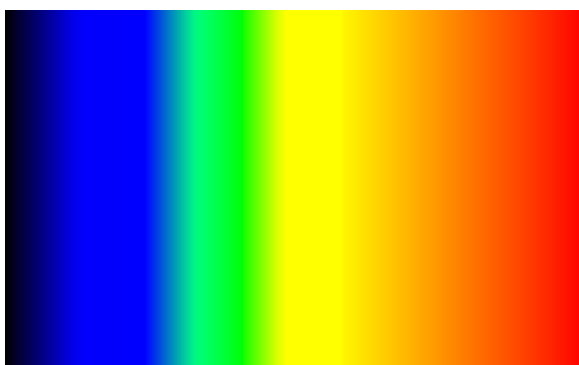
28



29



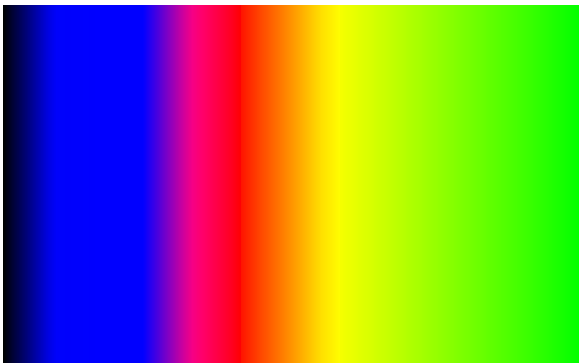
30



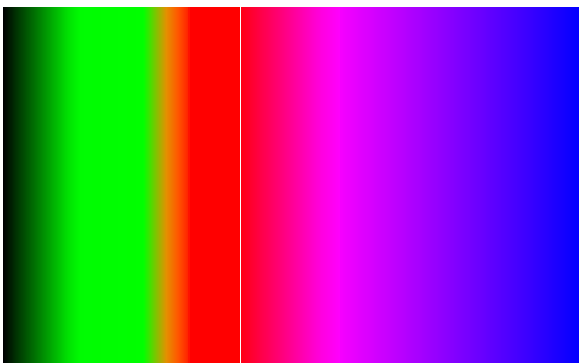
31



32



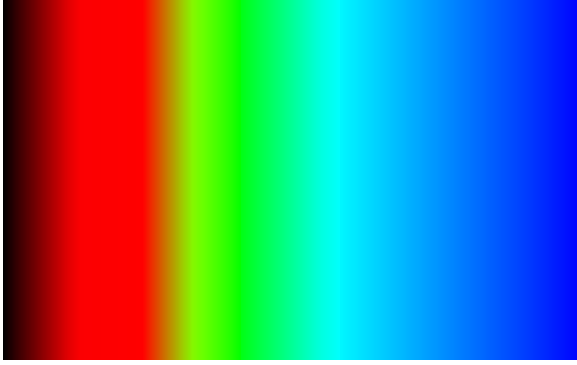
33



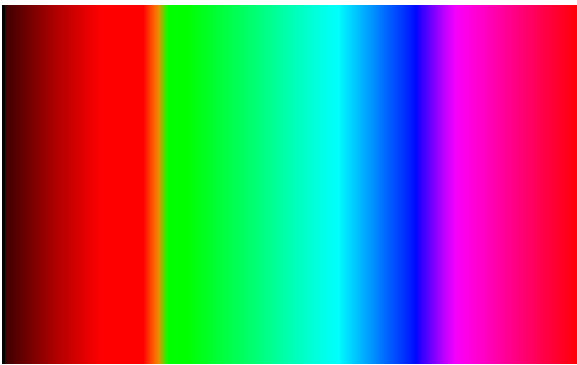
34



35



36



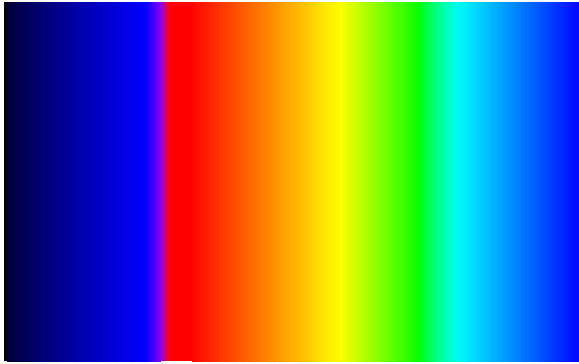
37



38



39



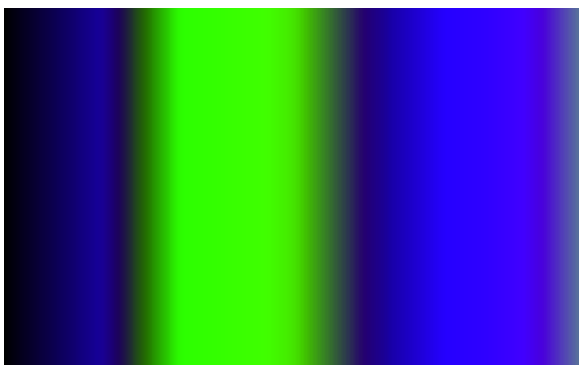
40



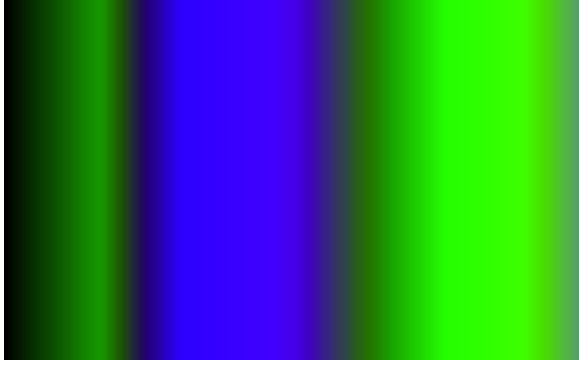
41



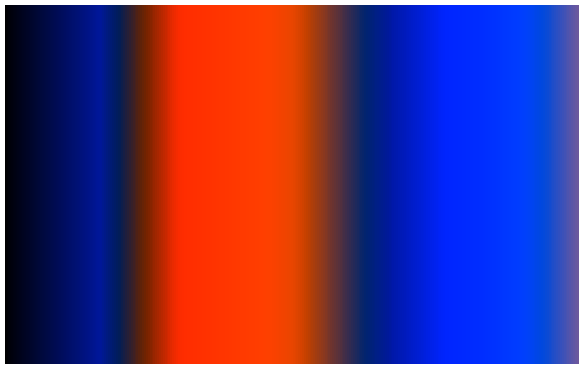
42



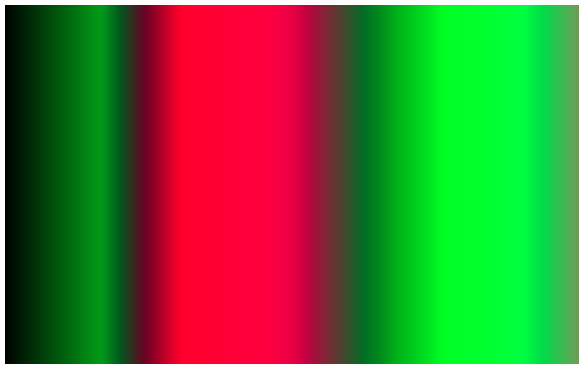
43



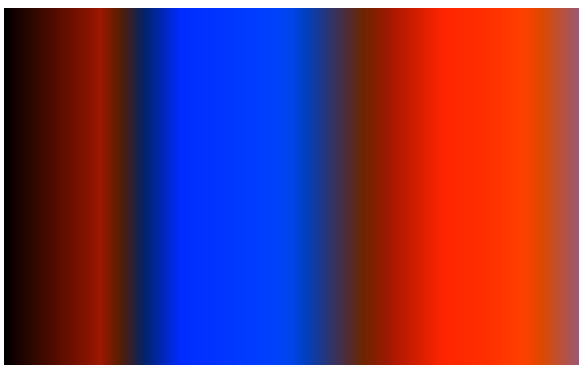
44



45

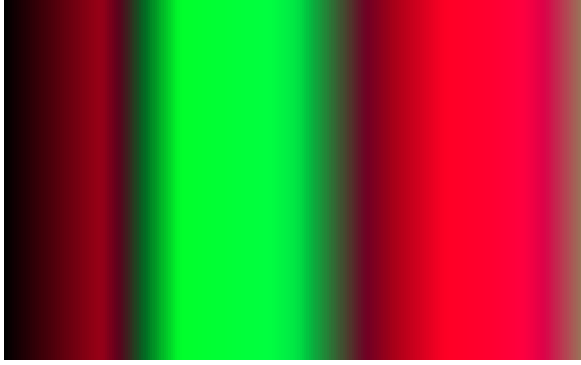


46

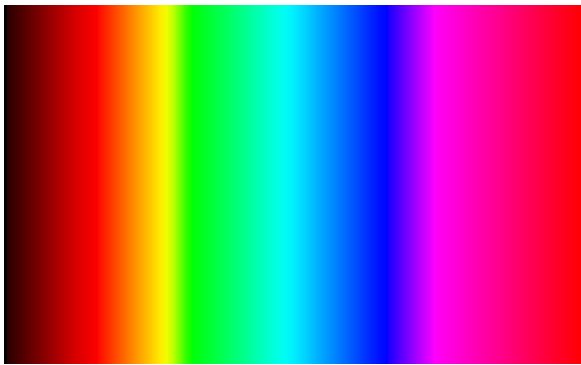


47





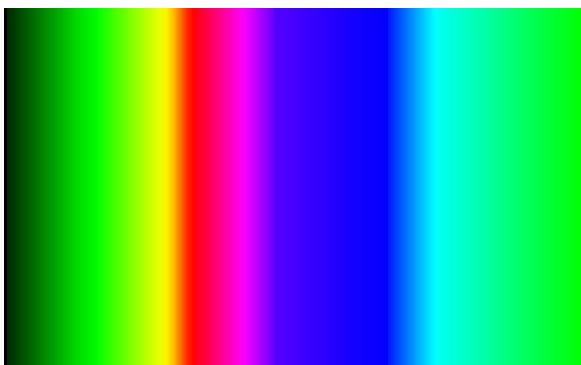
48



49



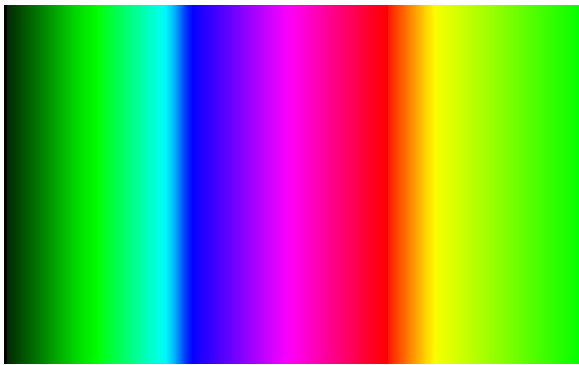
50



51



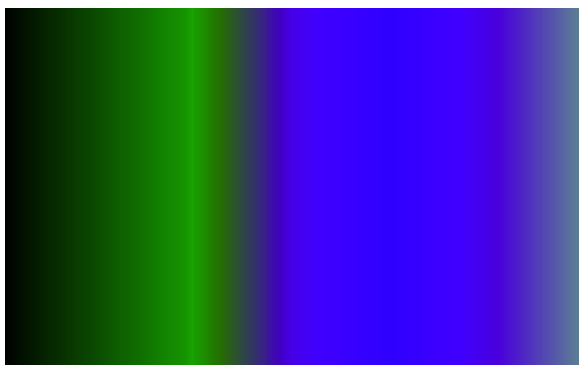
52



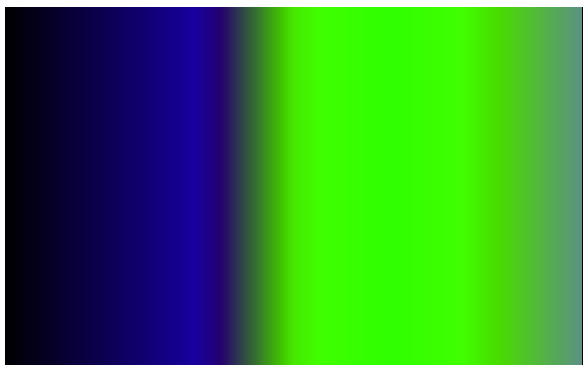
53



54



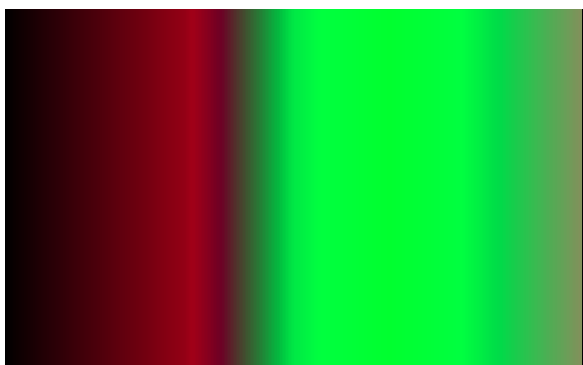
55



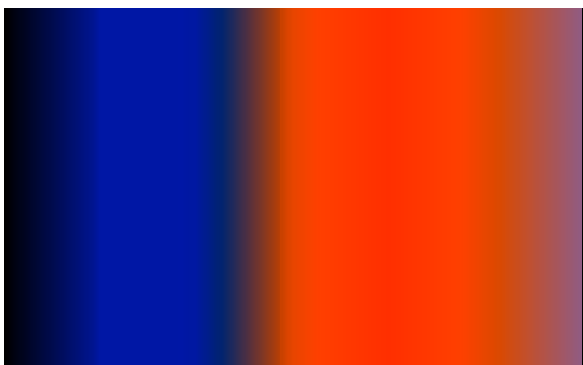
56



57



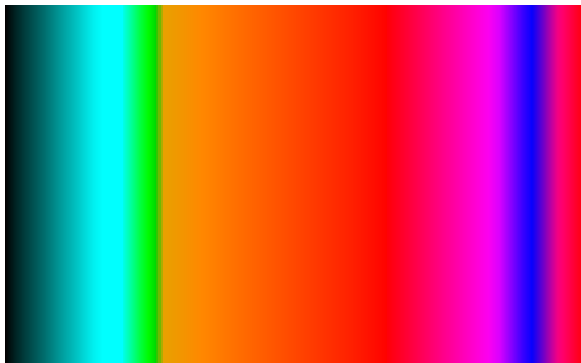
58



59



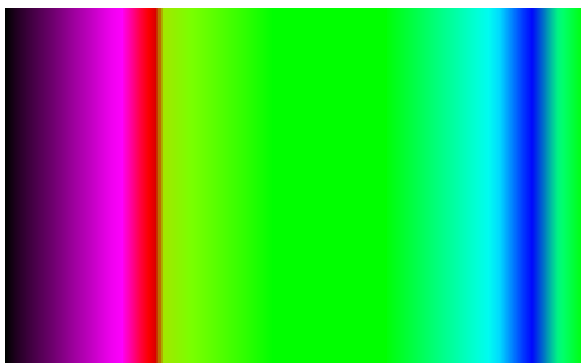
60



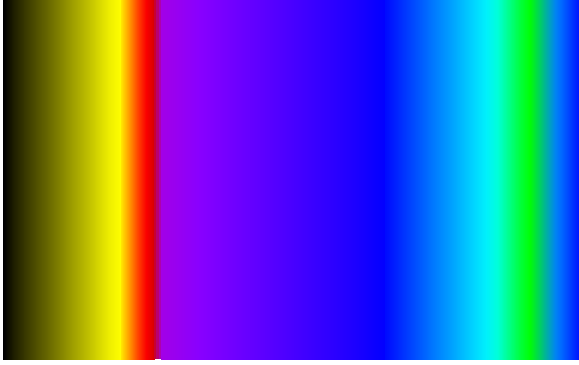
61



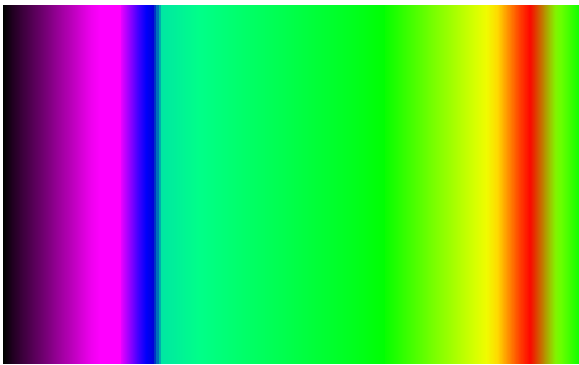
62



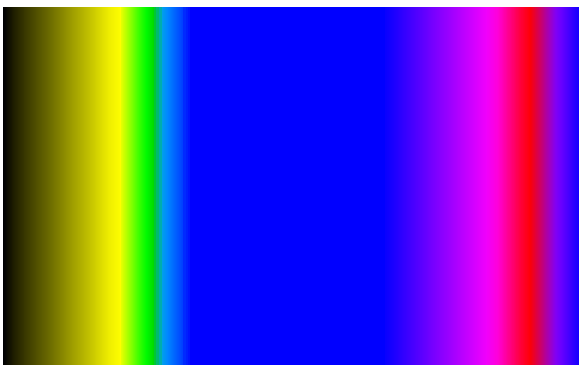
63



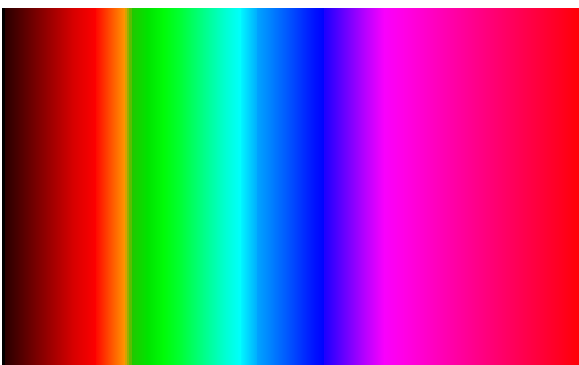
64



65



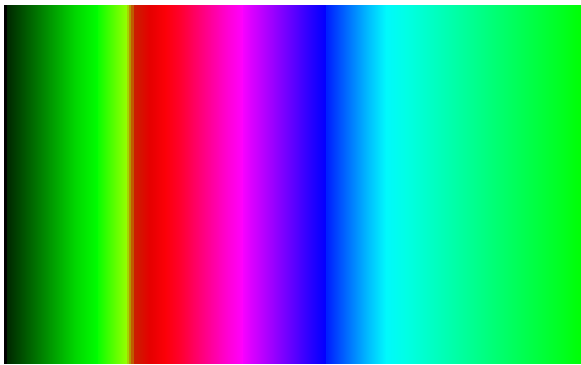
66



67



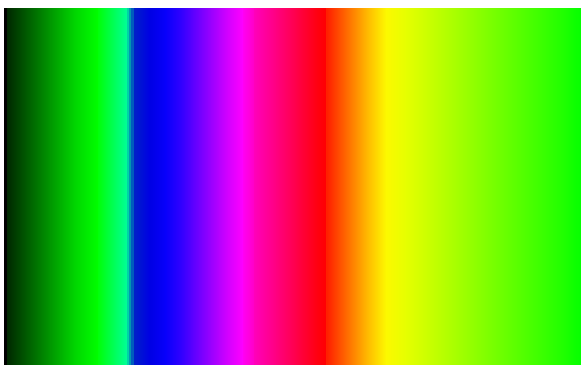
68



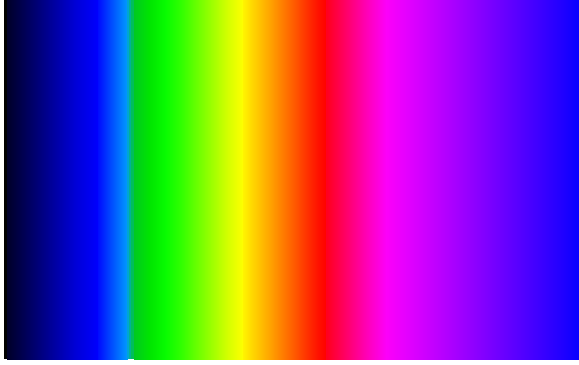
69



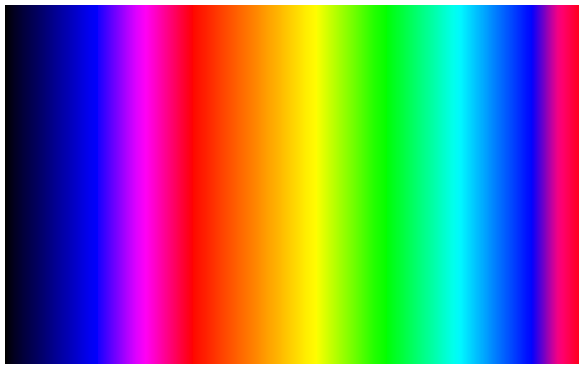
70



71



72



73



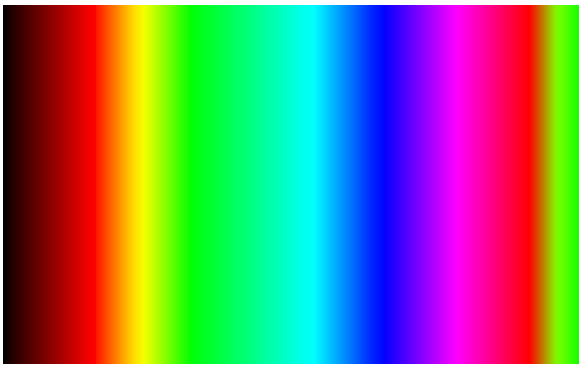
74



75



76



77



78



79





80



81



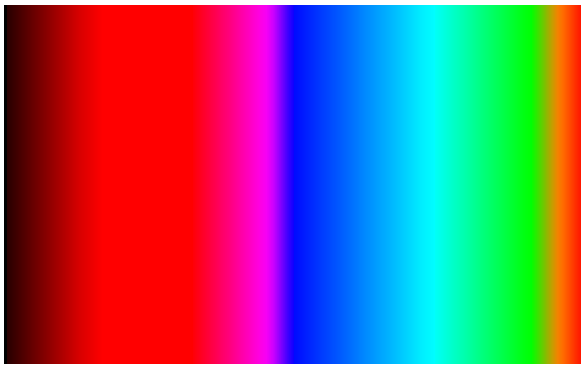
82



83



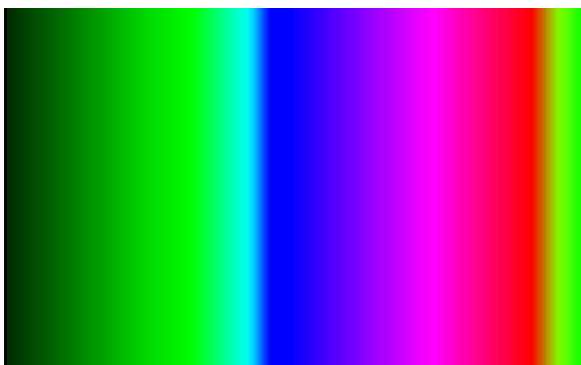
84



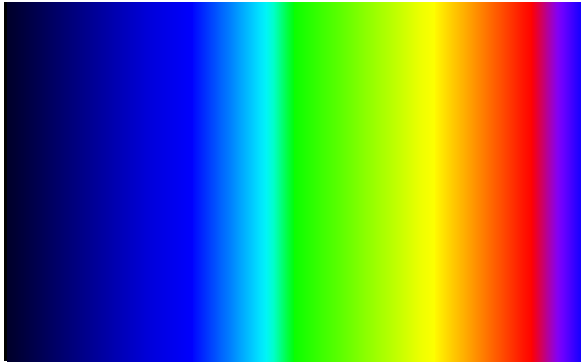
85



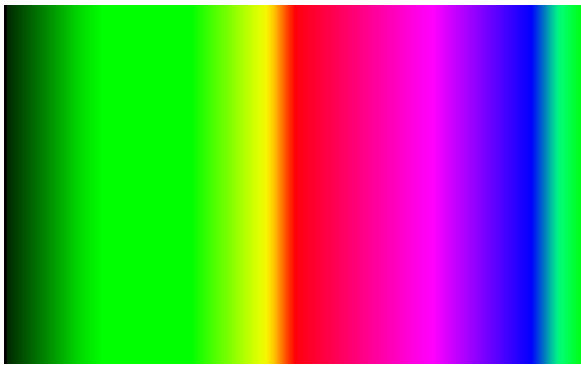
86



87



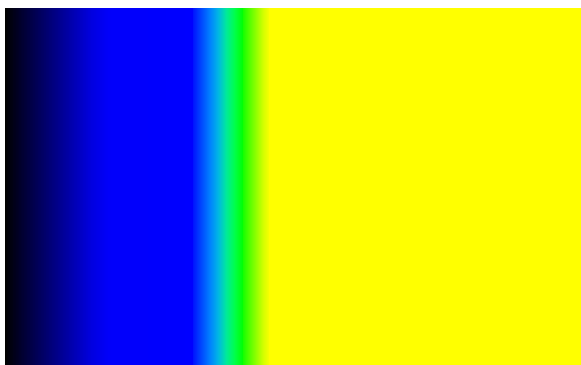
88



89



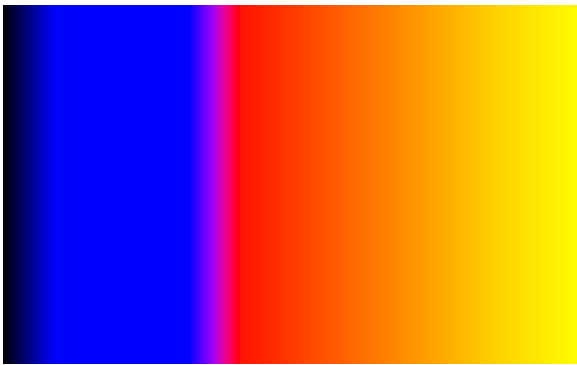
90



91



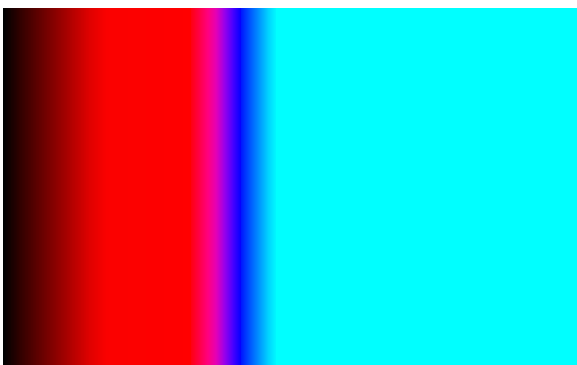
92



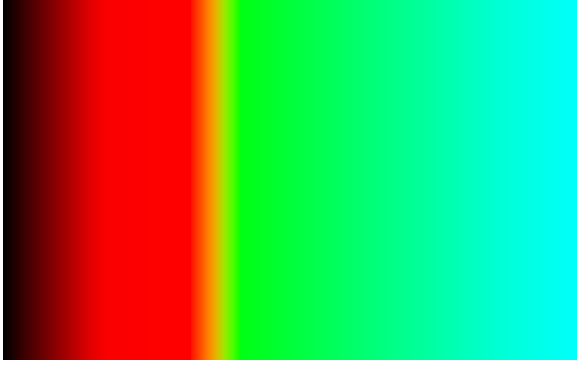
93



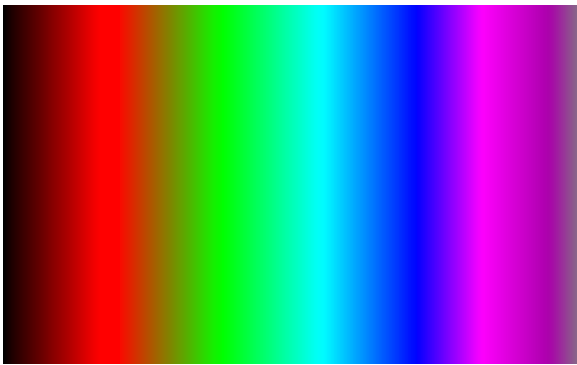
94



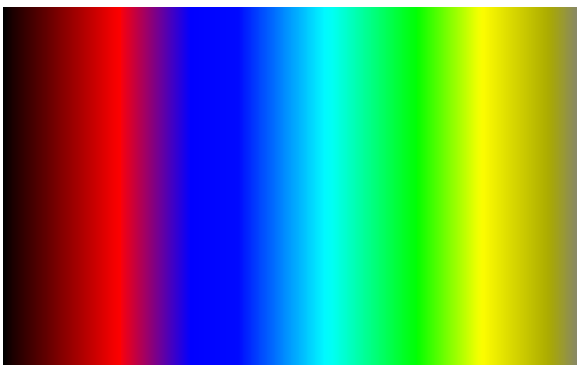
95



96



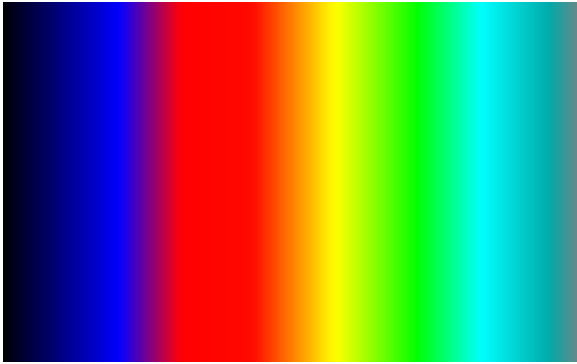
97



98



99



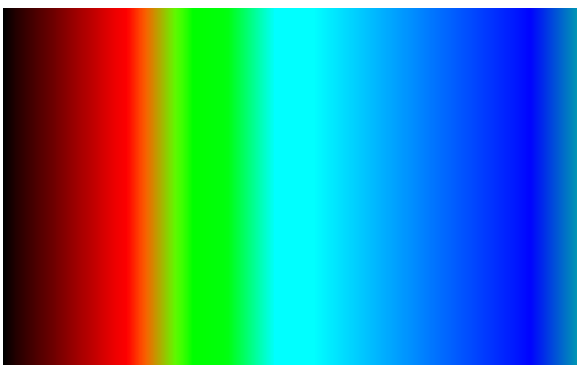
100



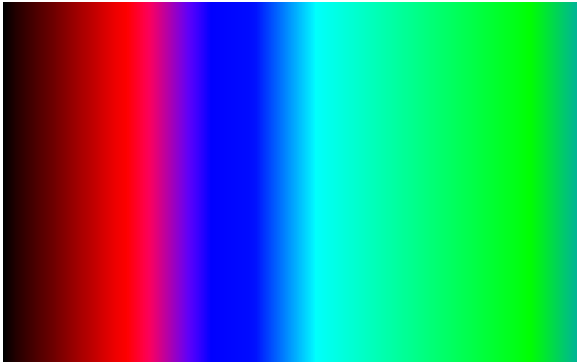
101



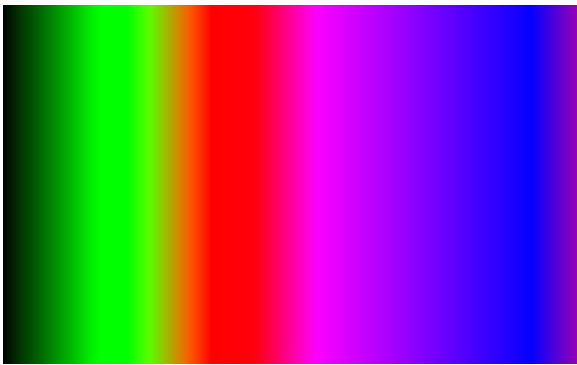
102



103



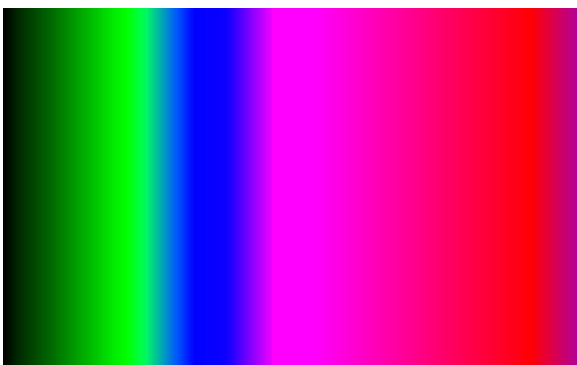
104



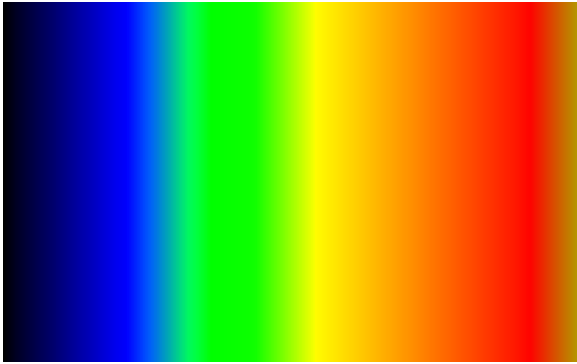
105



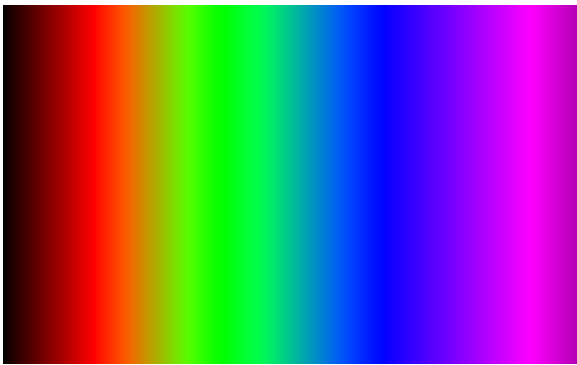
106



107



108



109

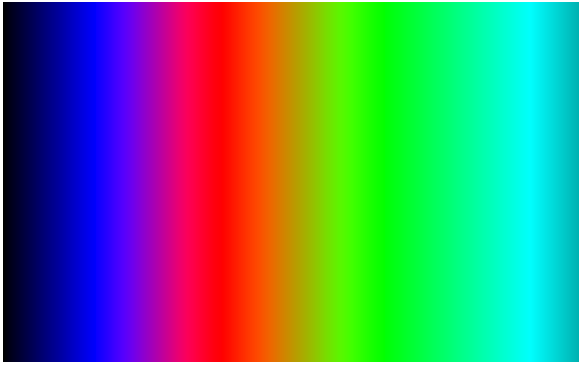


110

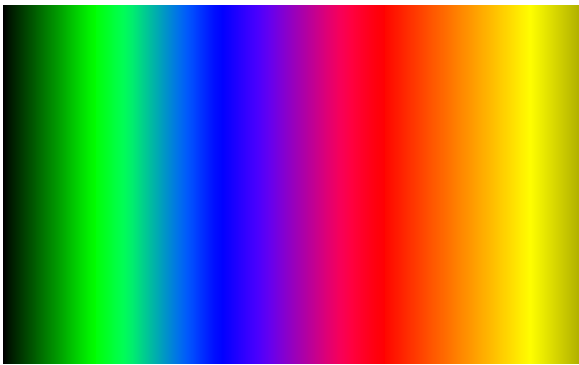


111





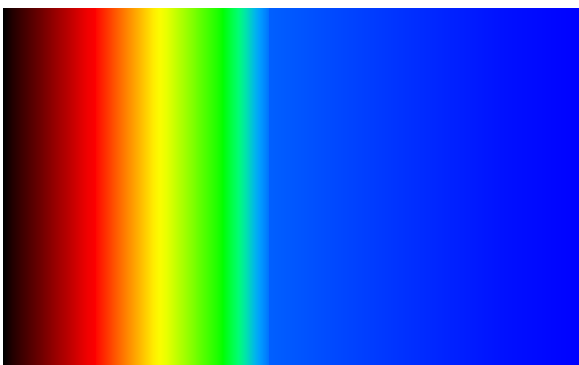
112



113



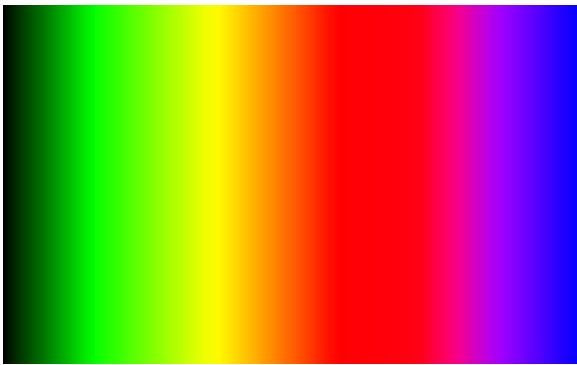
114



115



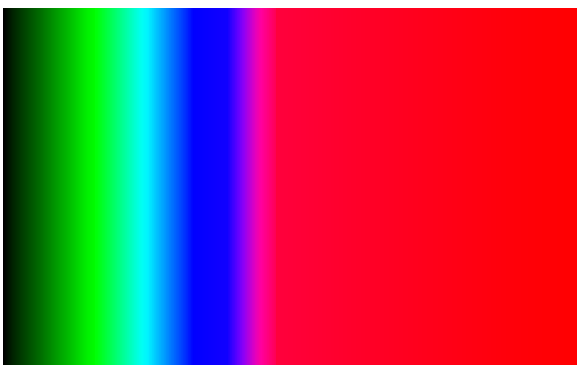
116



117



118



119



120



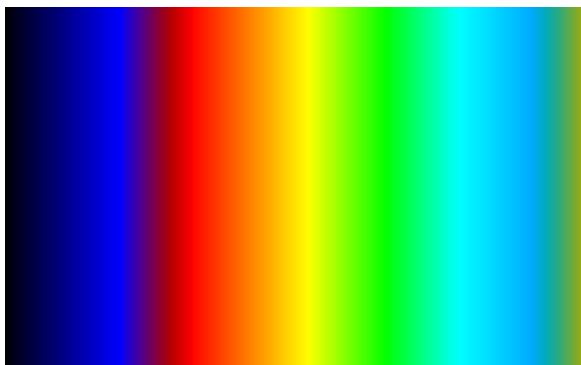
121



122



123



124



125



126



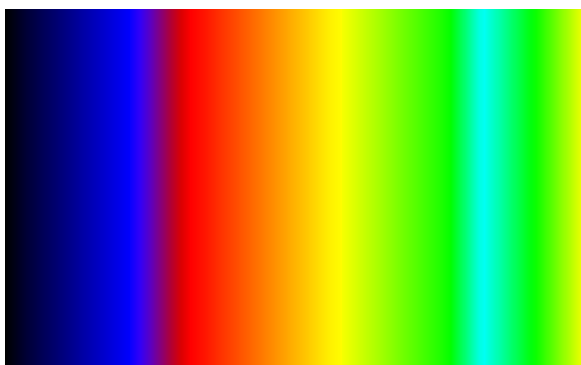
127



128



129



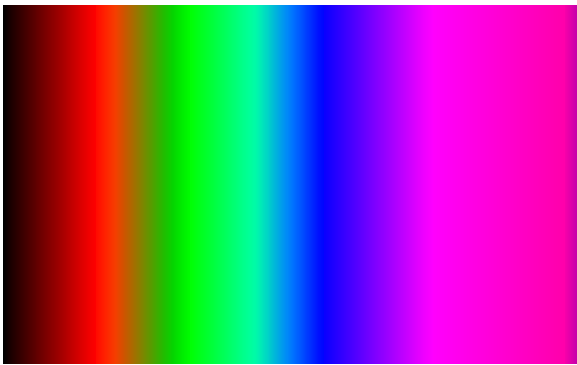
130



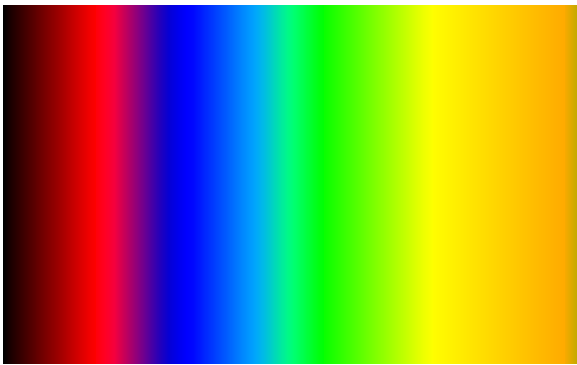
131



132



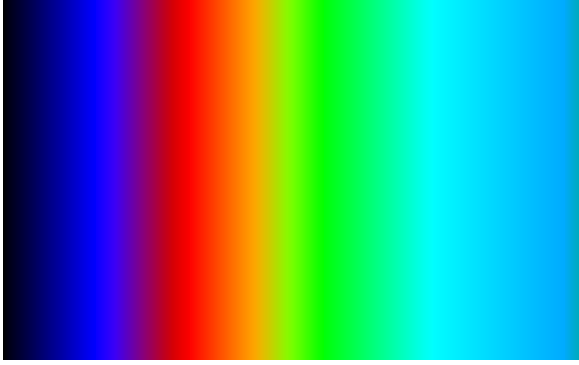
133



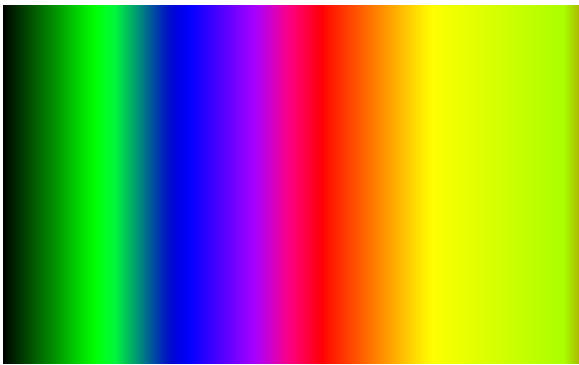
134



135



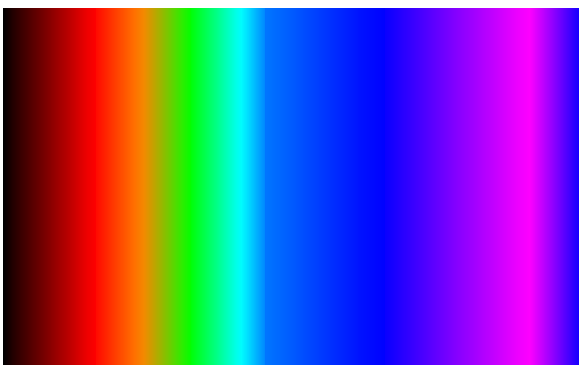
136



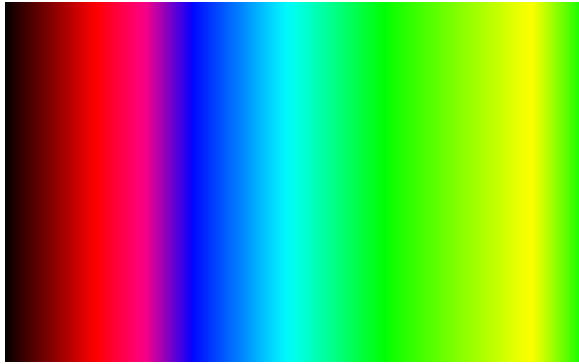
137



138



139



140



141

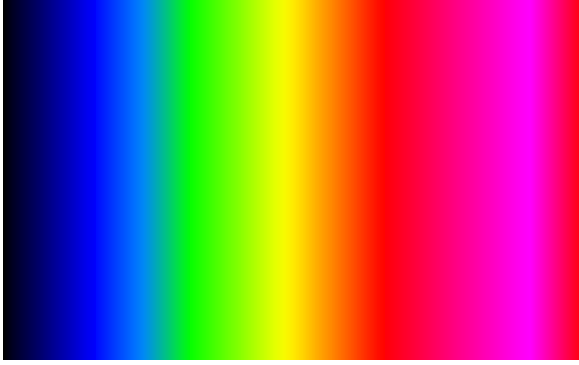


142

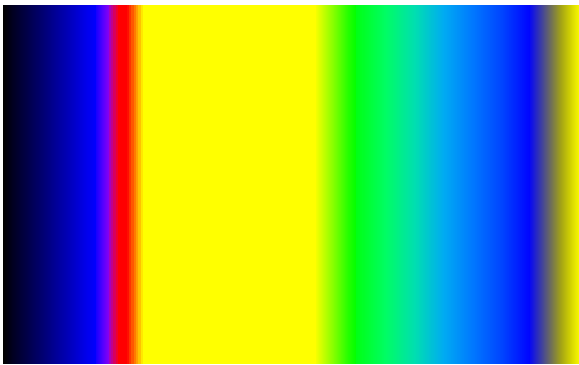


143

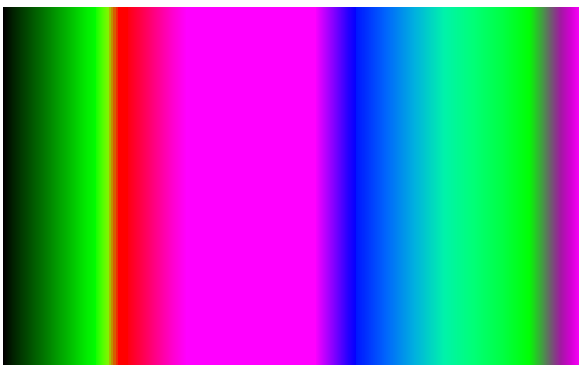




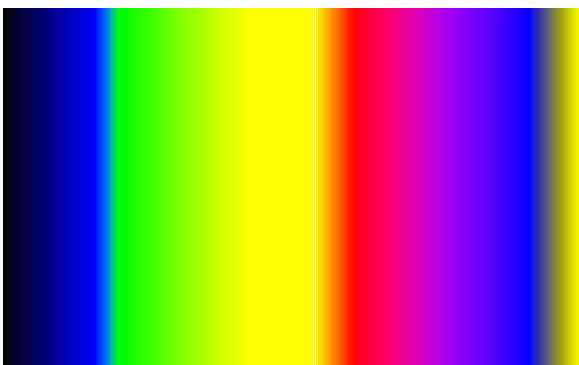
144



145



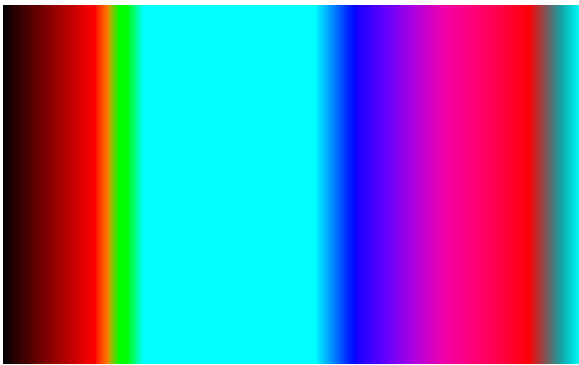
146



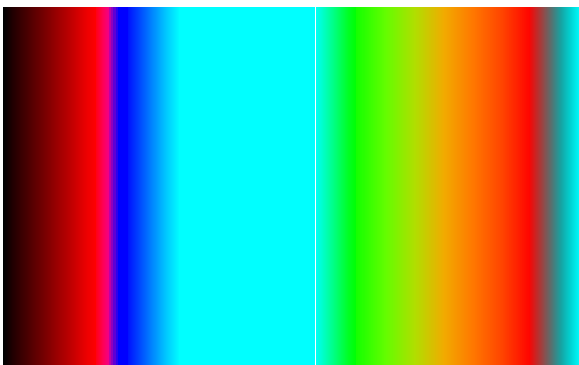
147



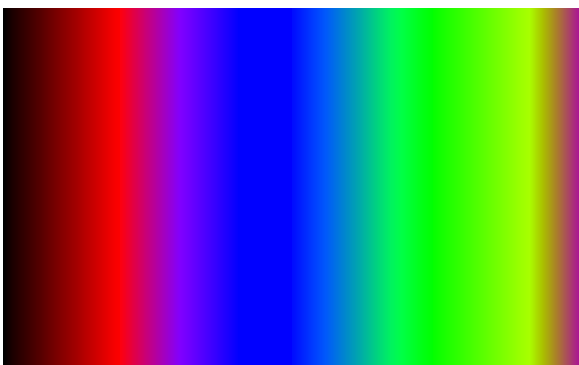
148



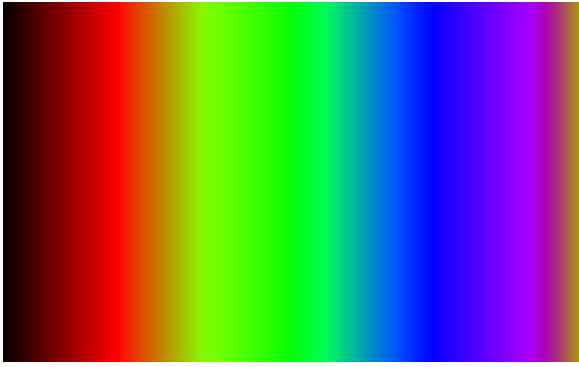
149



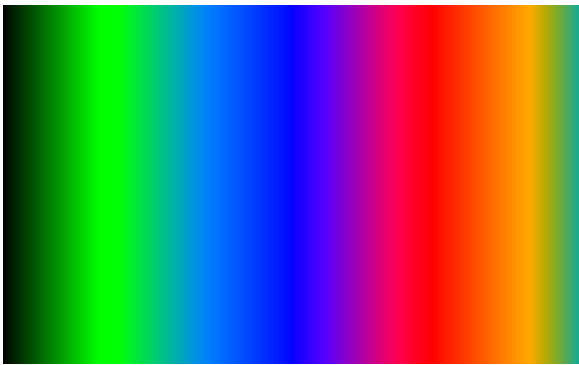
150



151



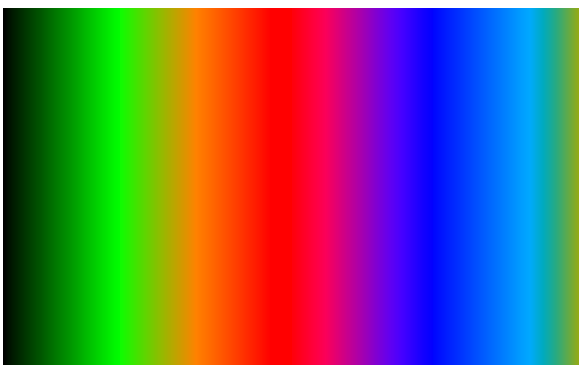
152



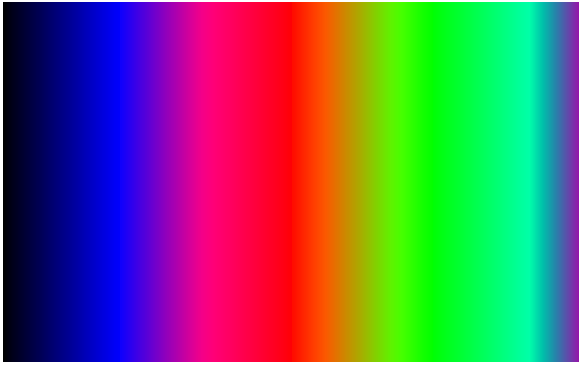
153



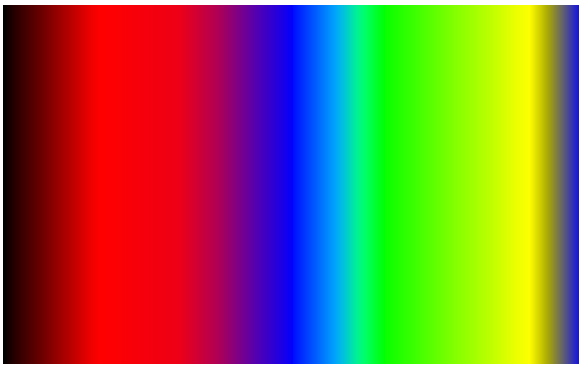
154



155



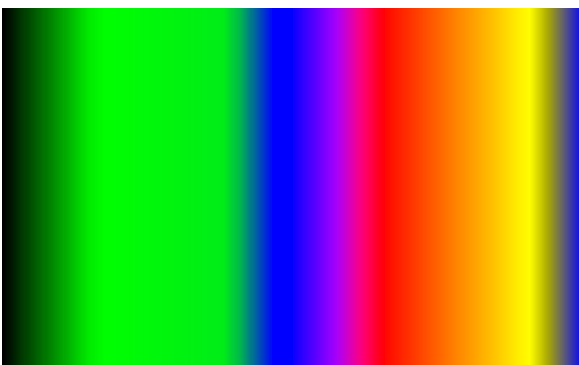
156



157



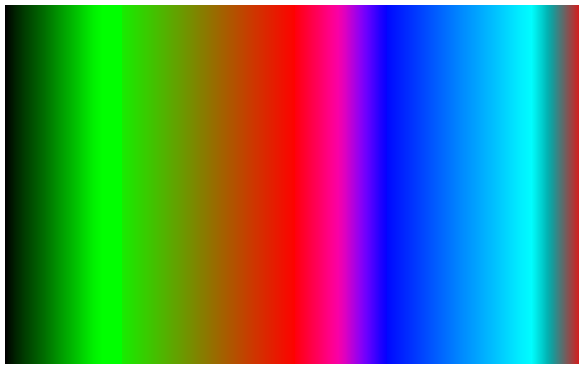
158



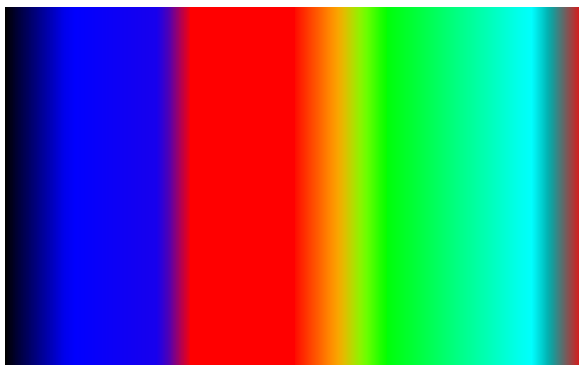
159



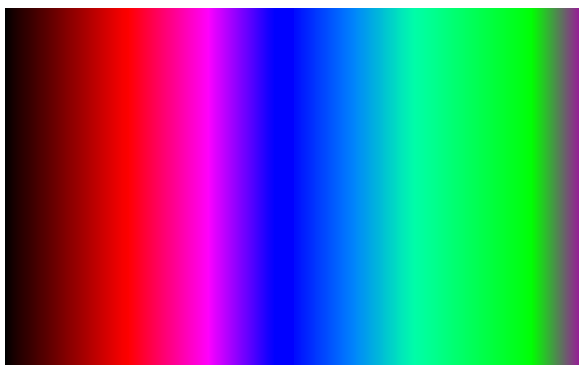
160



161



162



163



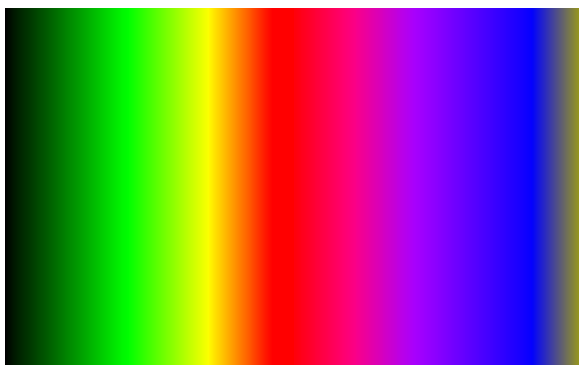
164



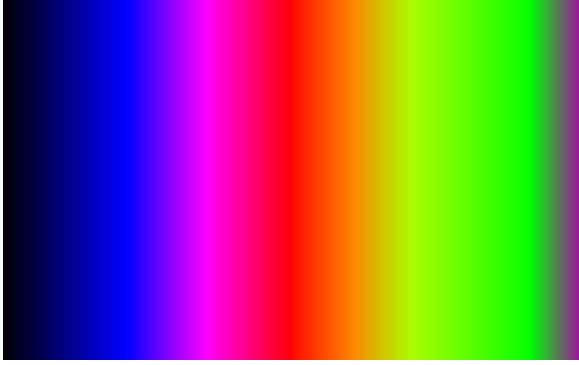
165



166



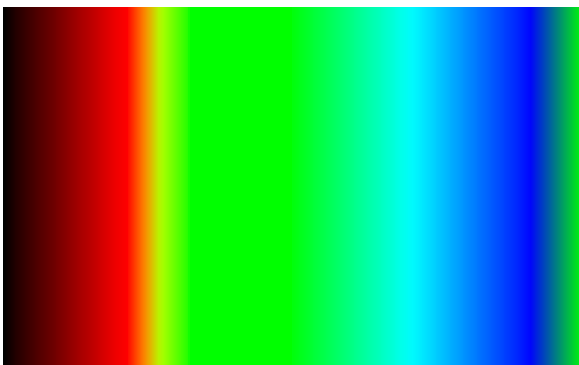
167



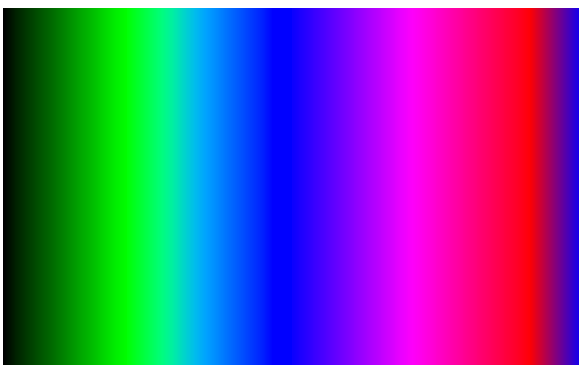
168



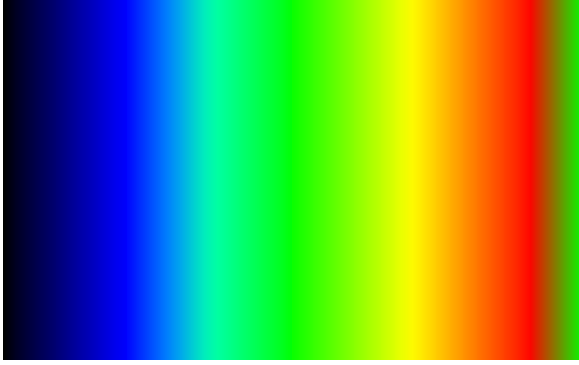
169



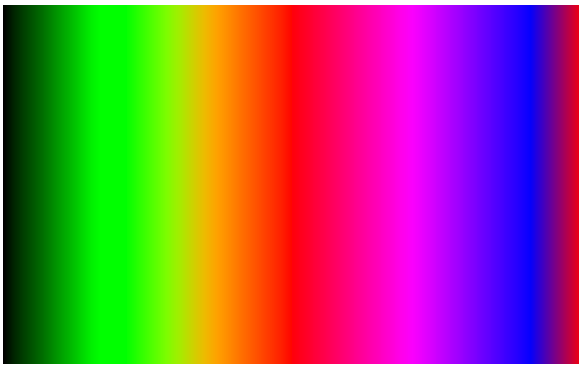
170



171



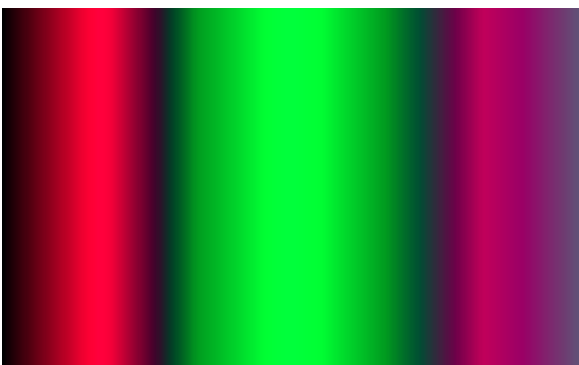
172



173

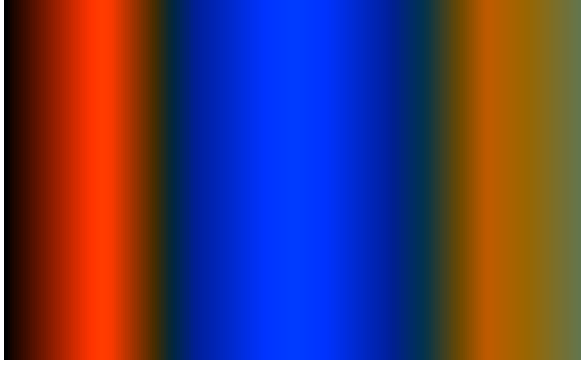


174

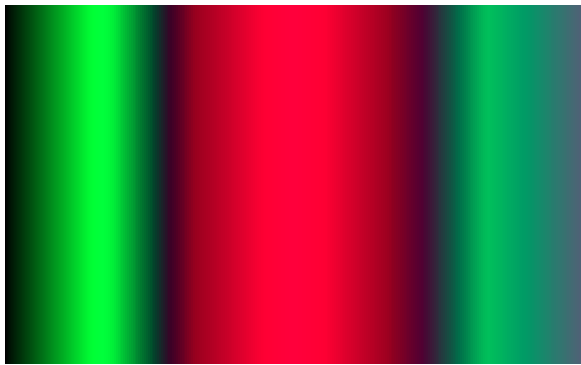


175

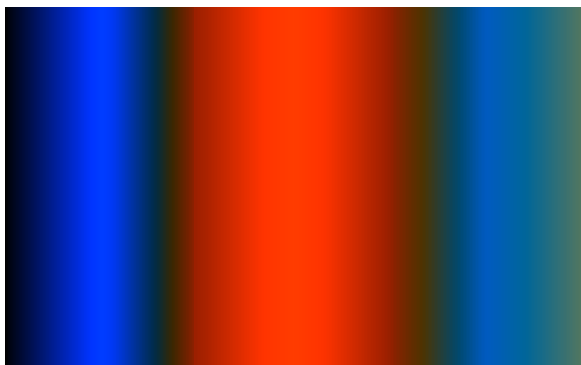




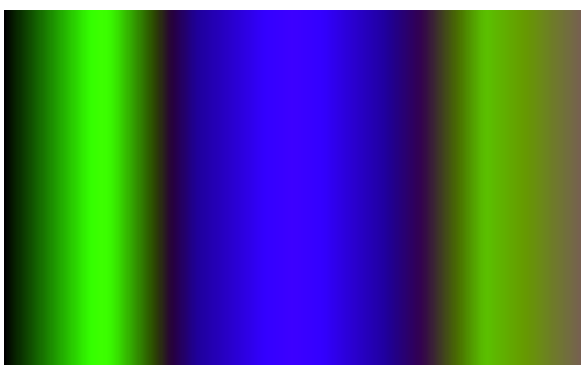
176



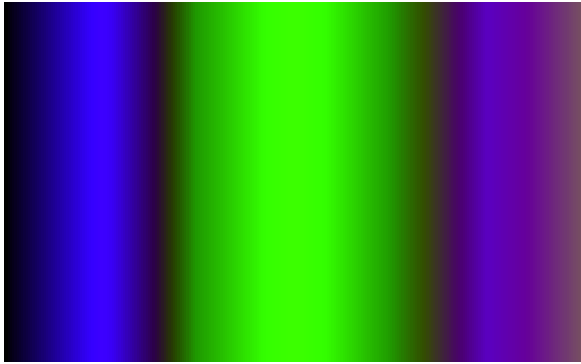
177



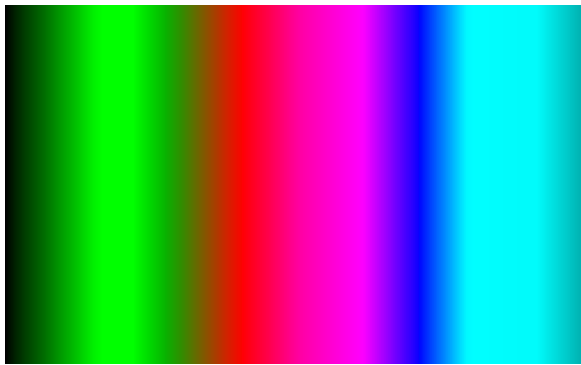
178



179



180



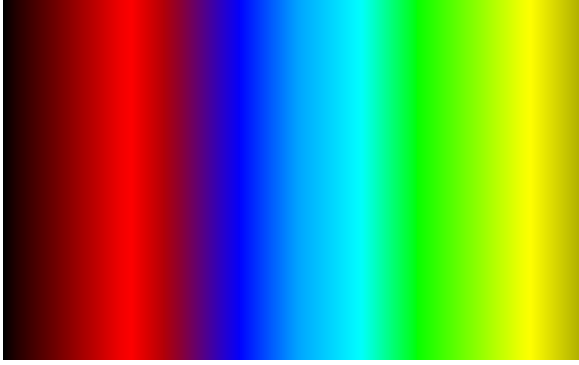
181



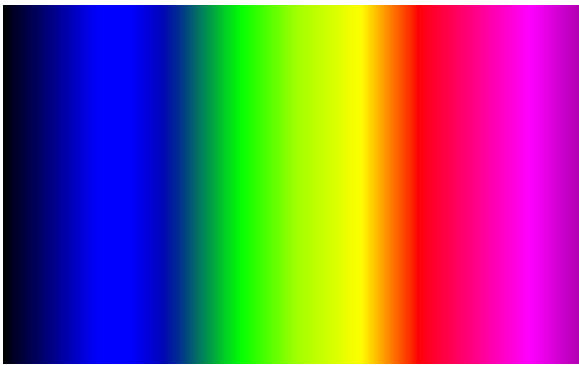
182



183



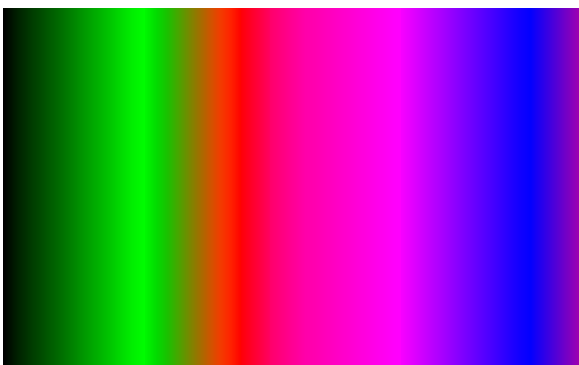
184



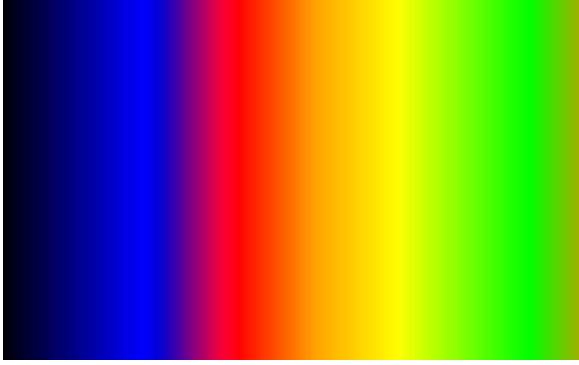
185



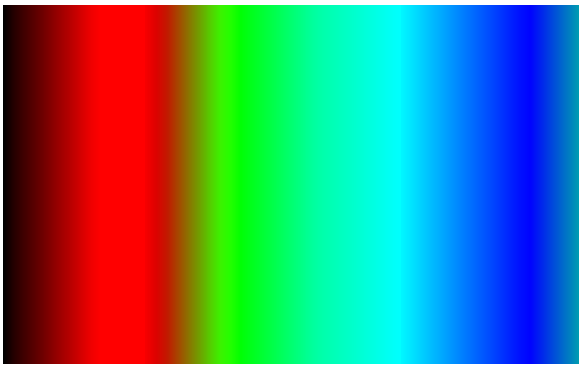
186



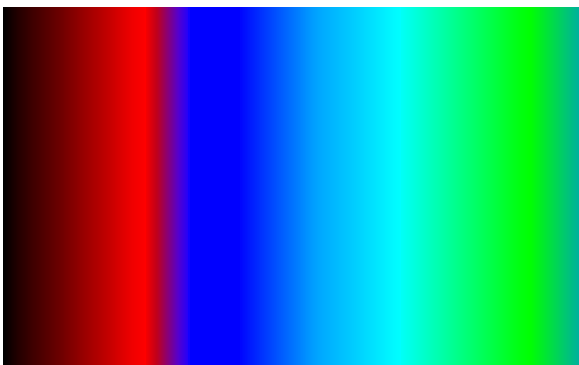
187



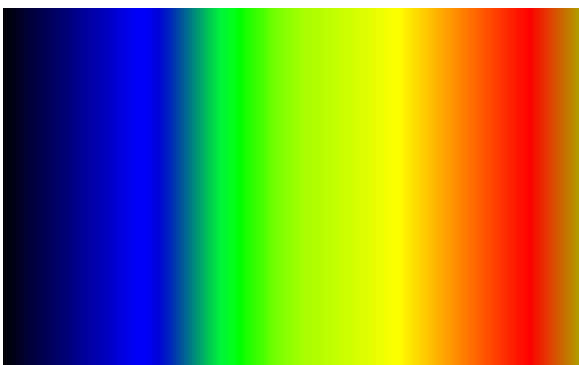
188



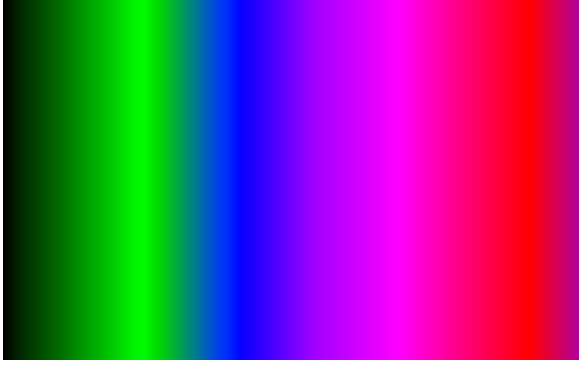
189



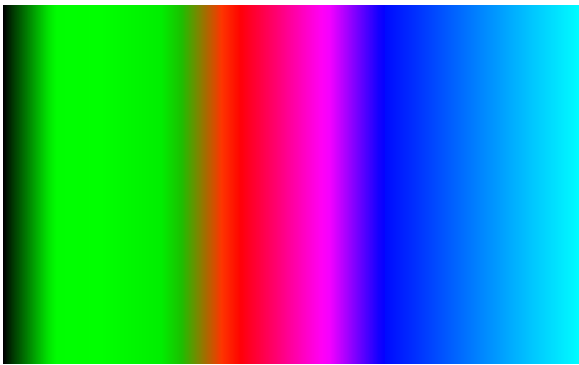
190



191



192



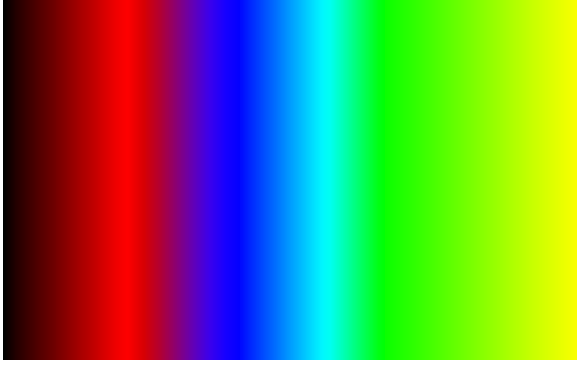
193



194



195



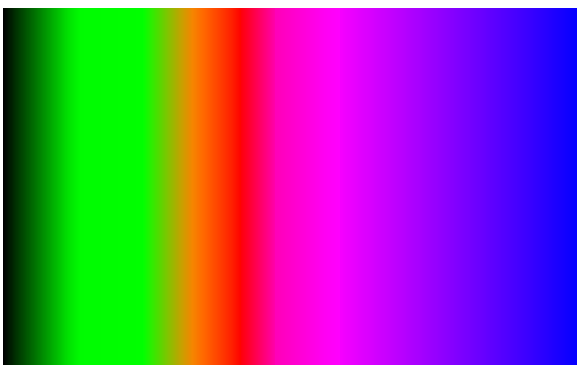
196



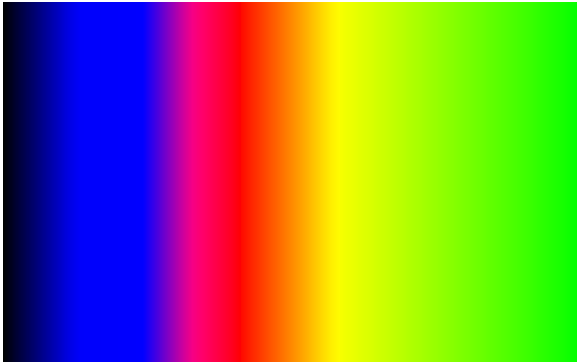
197



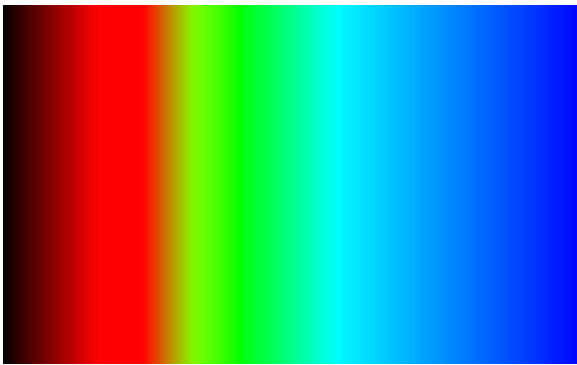
198



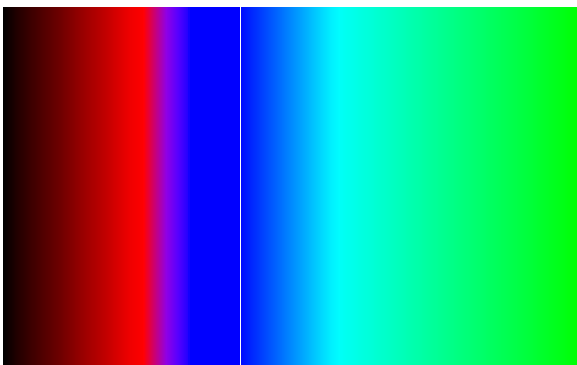
199



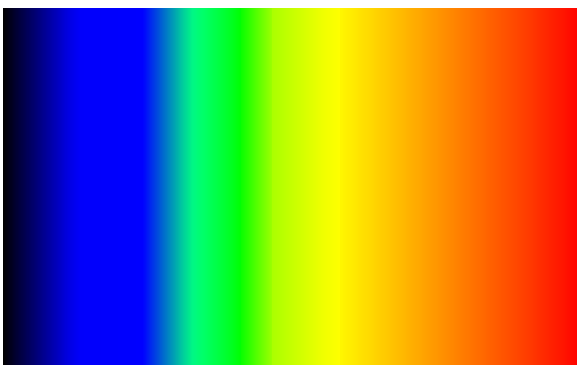
200



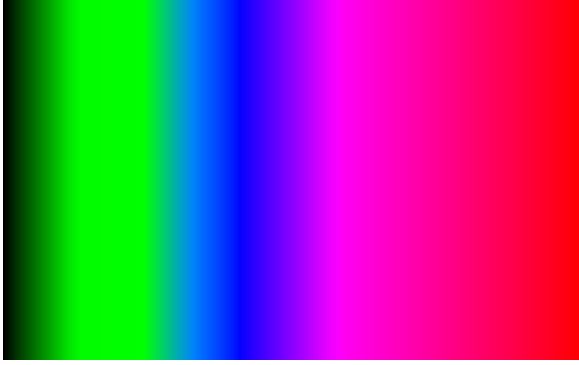
201



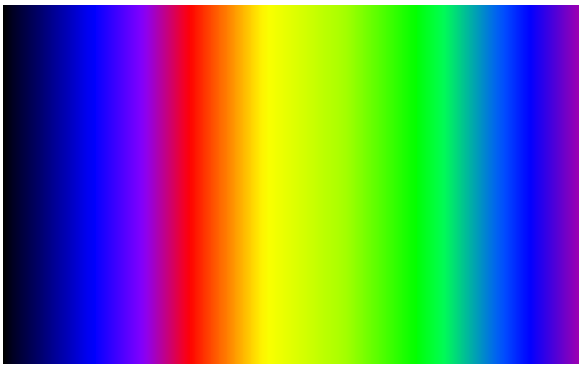
202



203



204



205



206

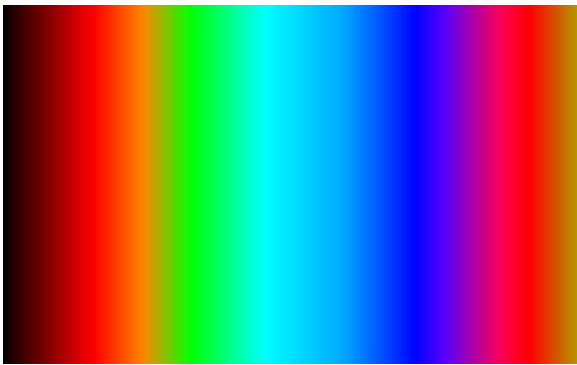


207

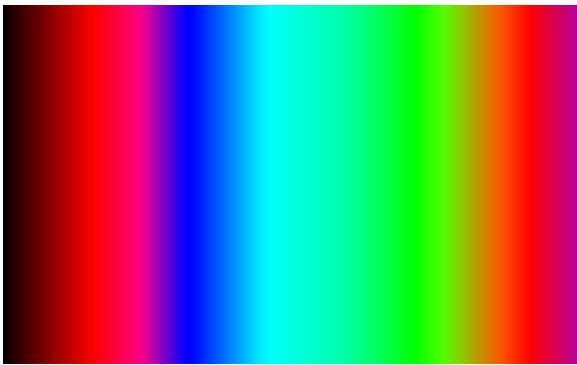




208



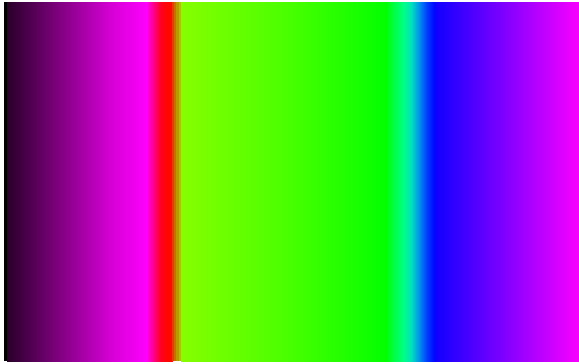
209



210



211



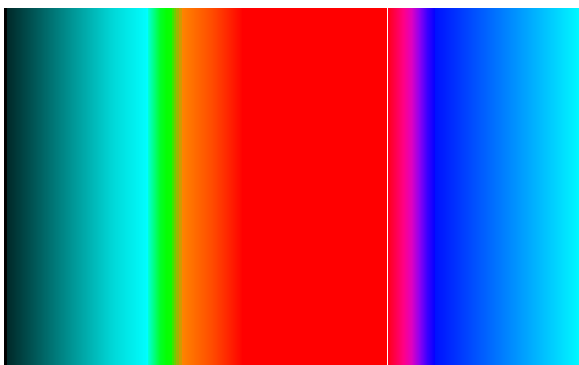
212



213



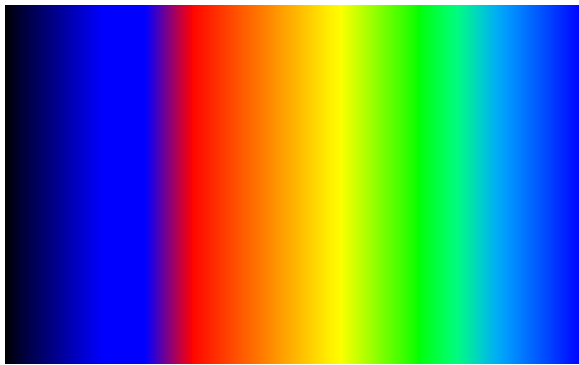
214



215



216



217



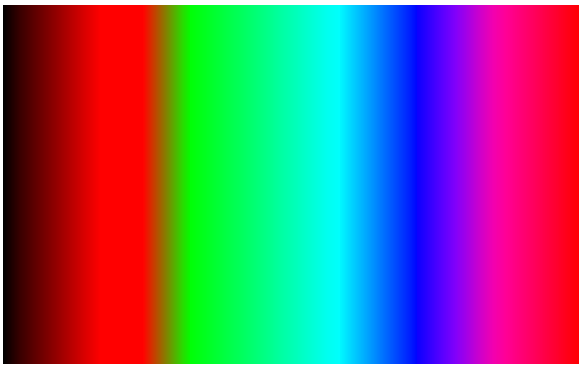
218



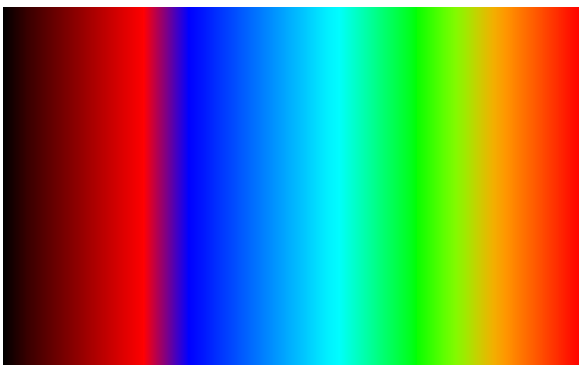
219



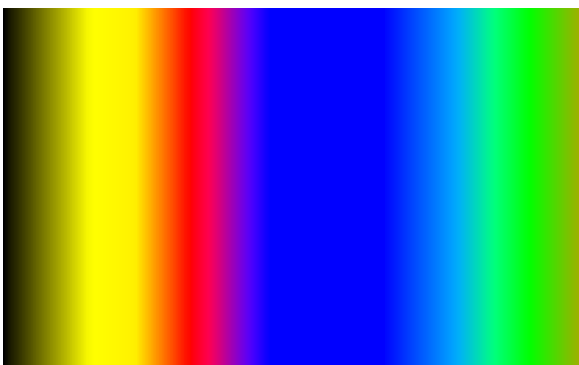
220



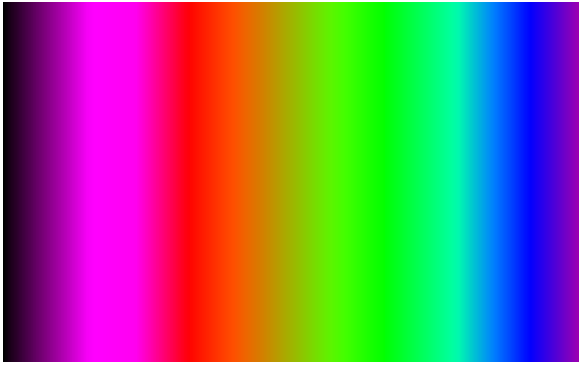
221



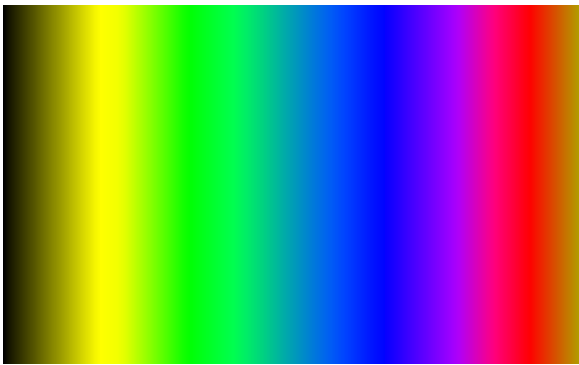
222



223



224



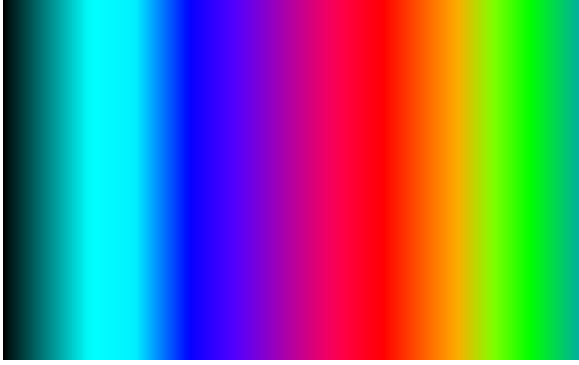
225



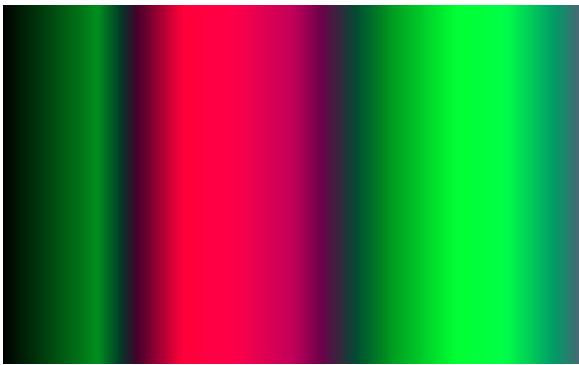
226



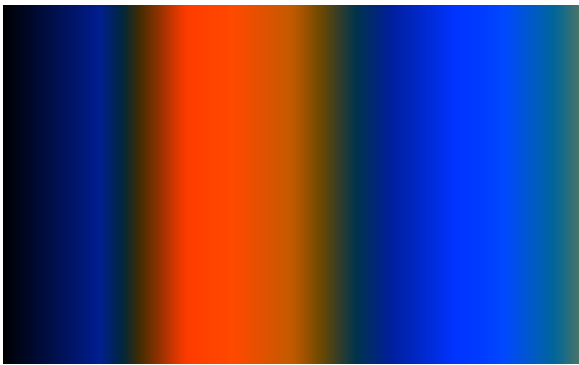
227



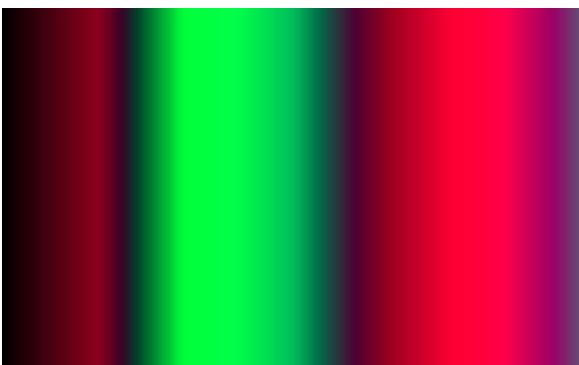
228



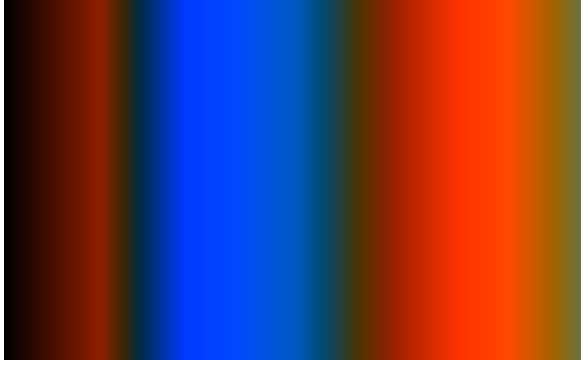
229



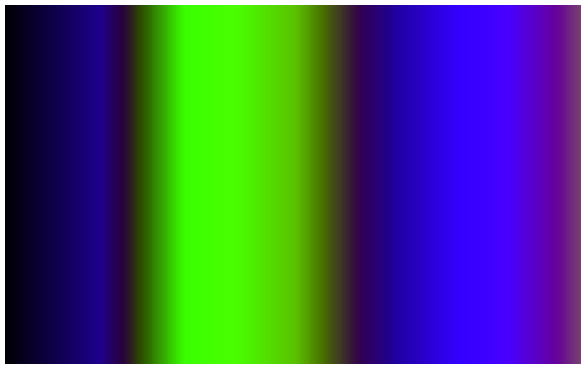
230



231



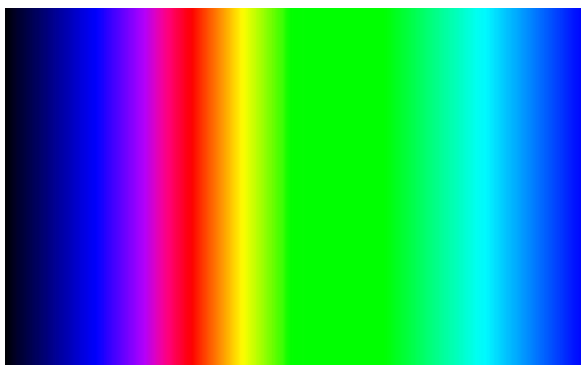
232



233



234



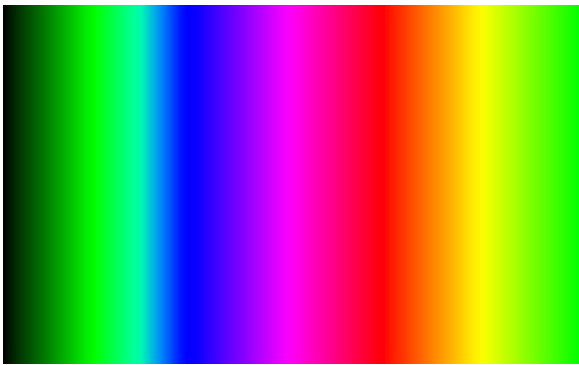
235



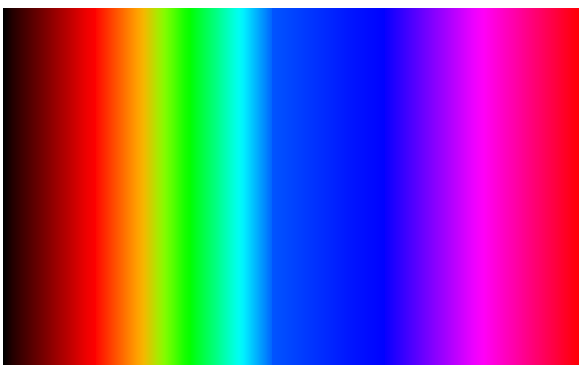
236



237



238

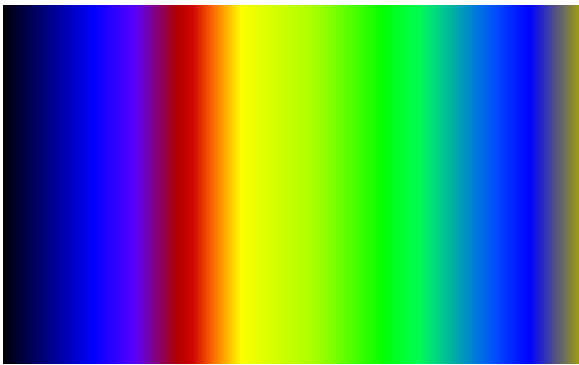


239





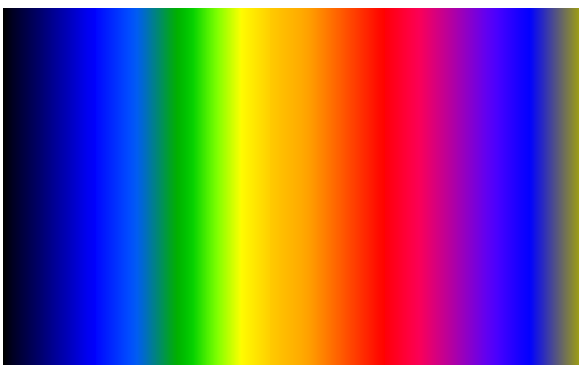
240



241



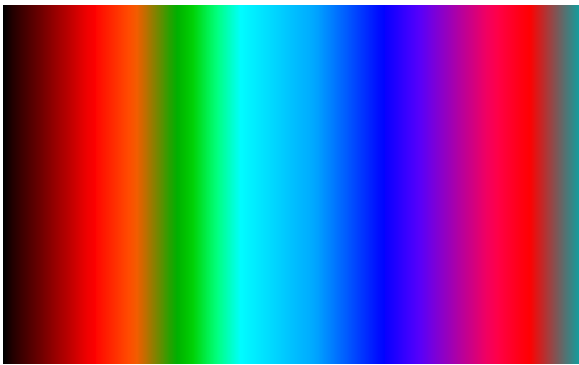
242



243



244



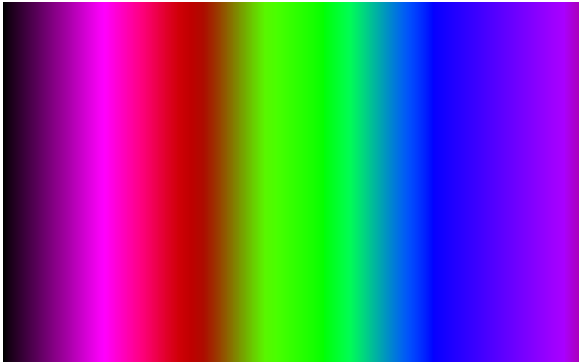
245



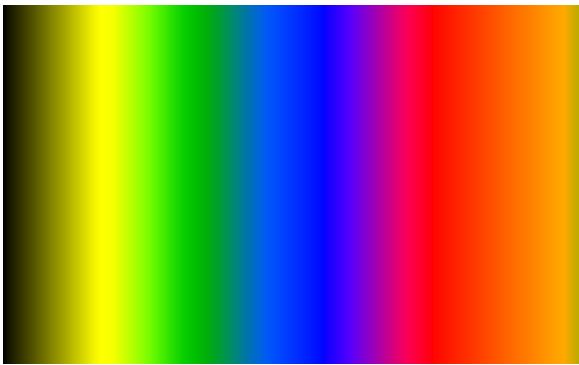
246



247



248



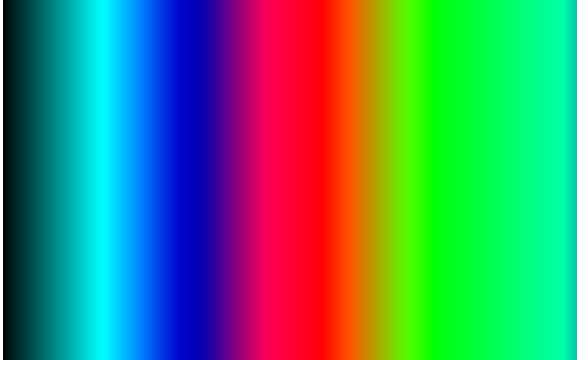
249



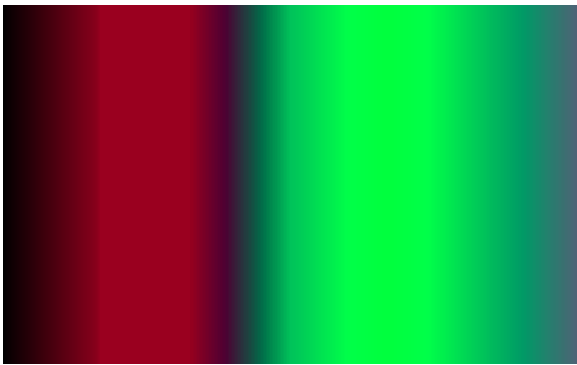
250



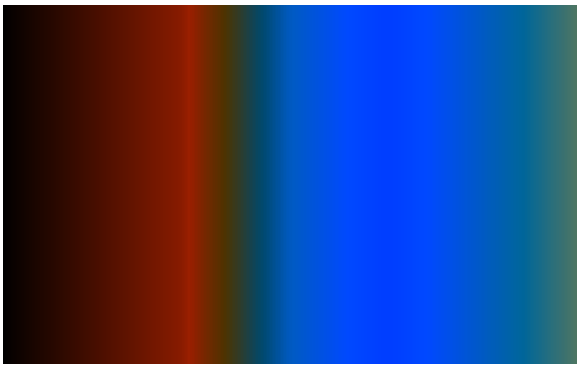
251



252



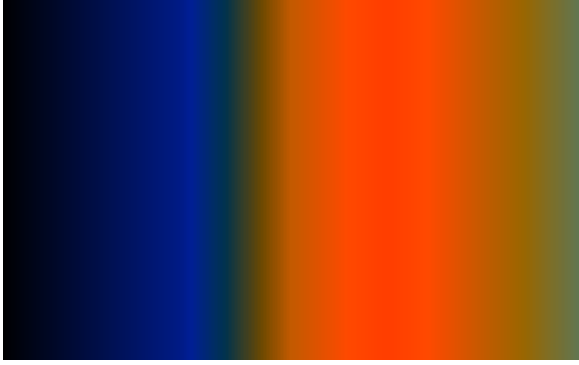
253



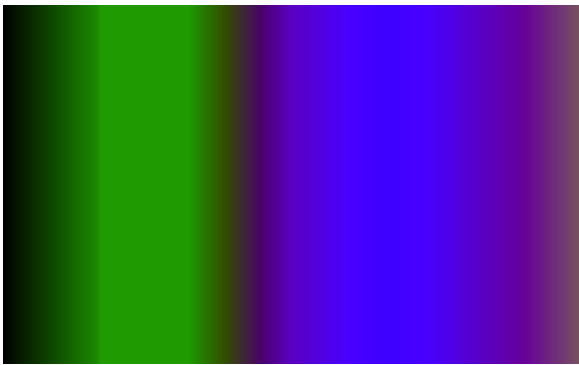
254



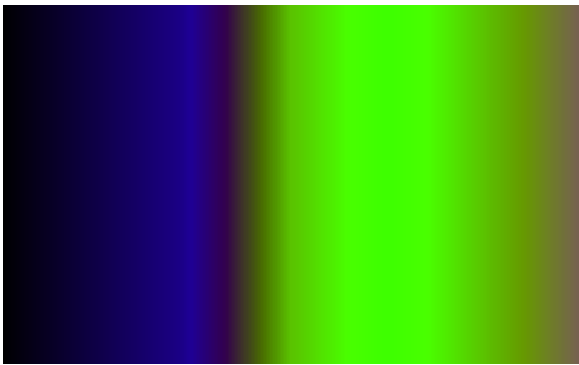
255



256



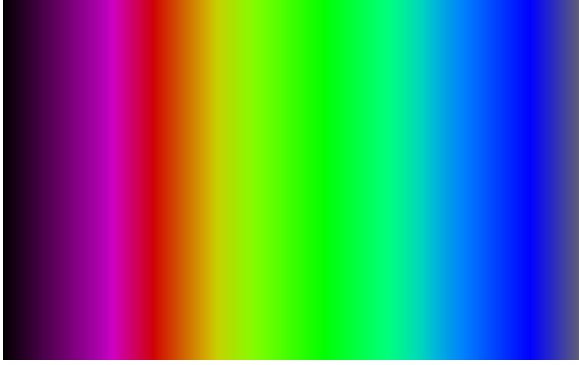
257



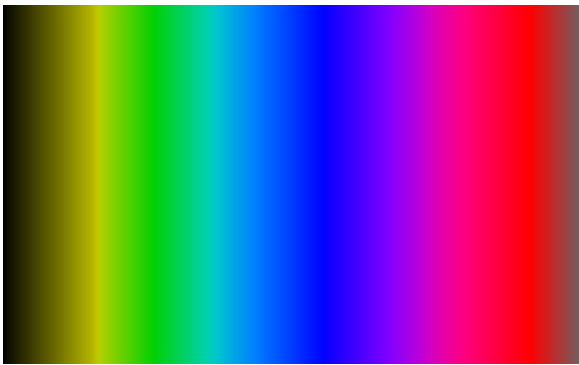
258



259



260



261



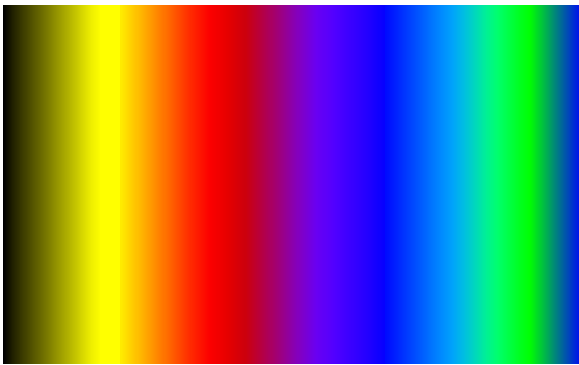
262



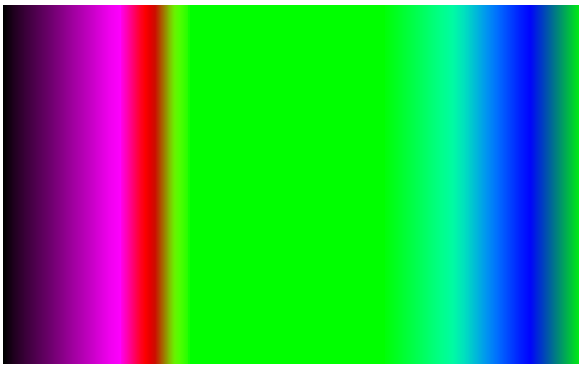
263



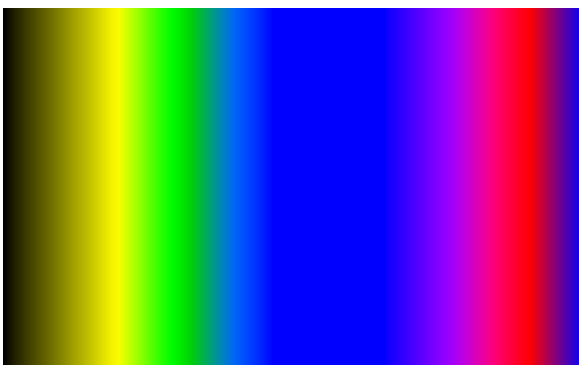
264



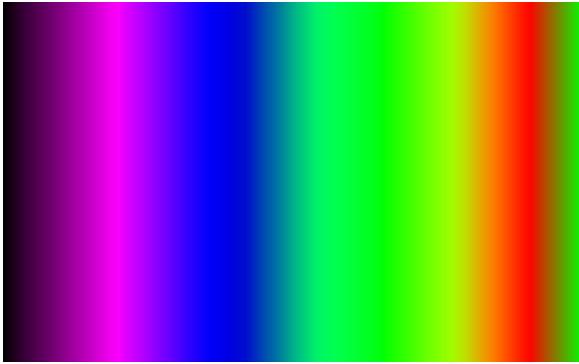
265



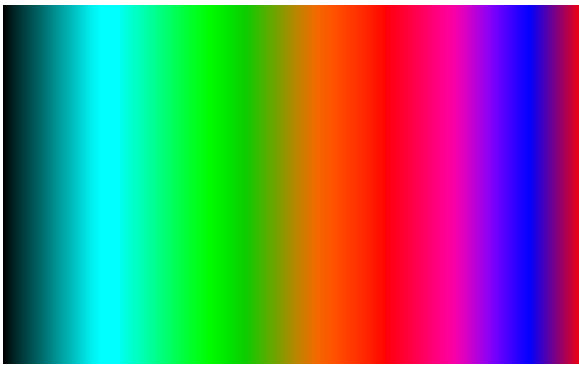
266



267



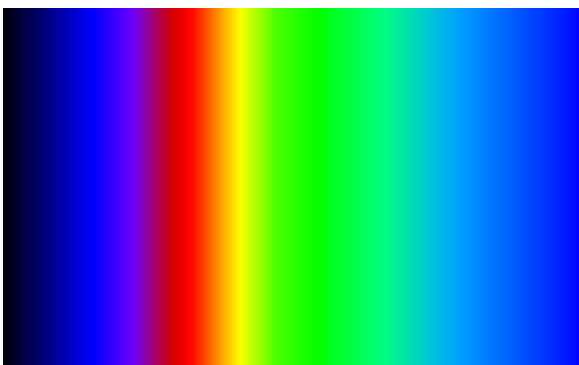
268



269



270

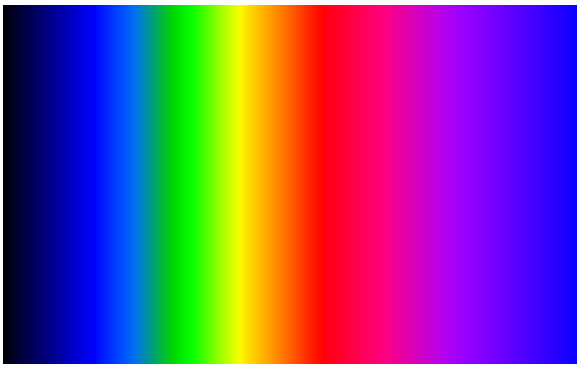


271





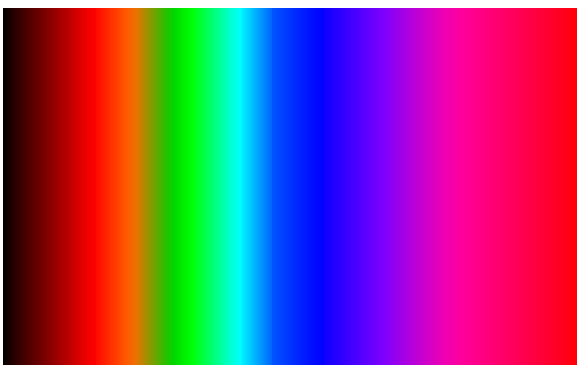
272



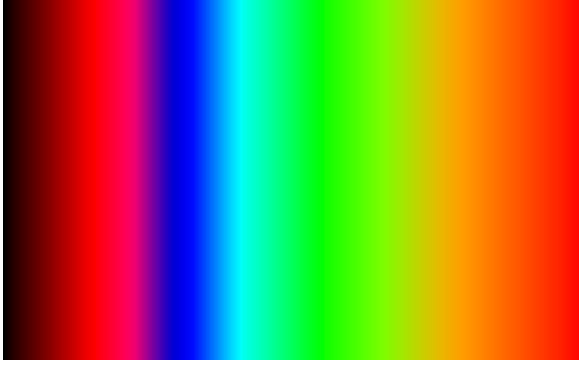
273



274



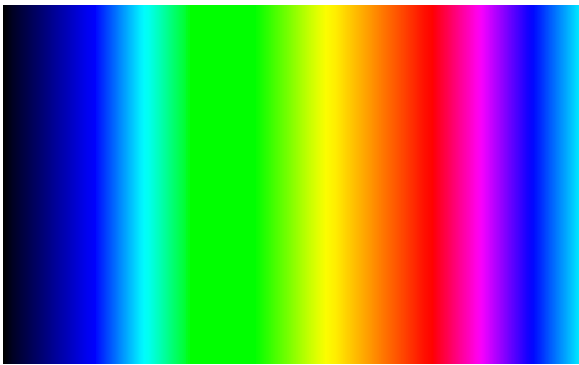
275



276



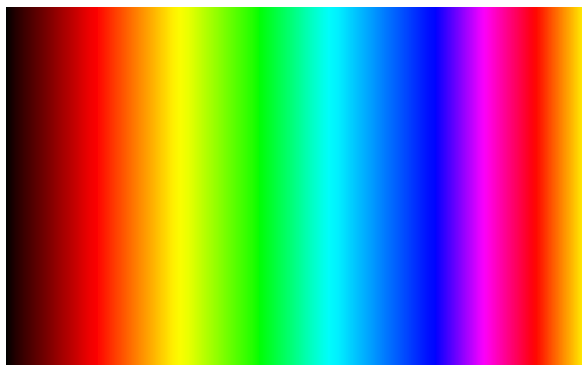
277



278



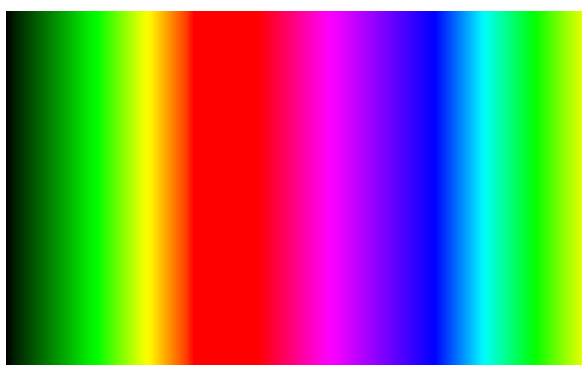
279



280



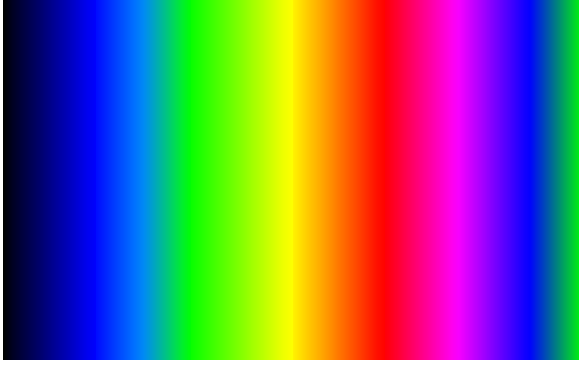
281



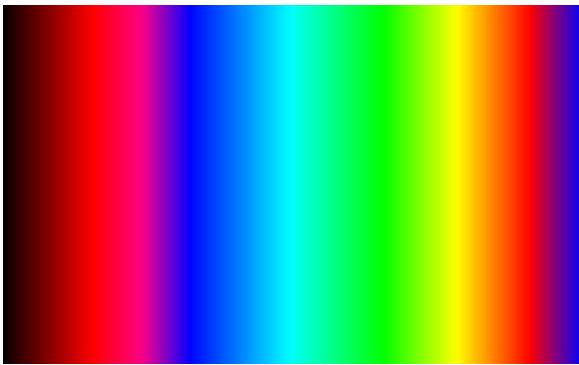
282



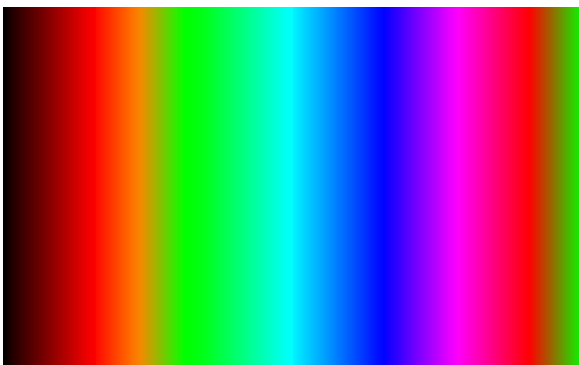
283



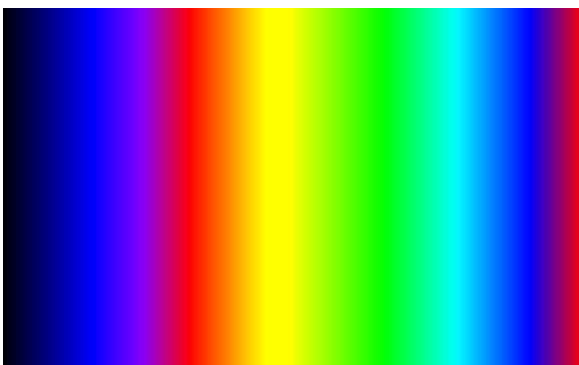
284



285



286



287



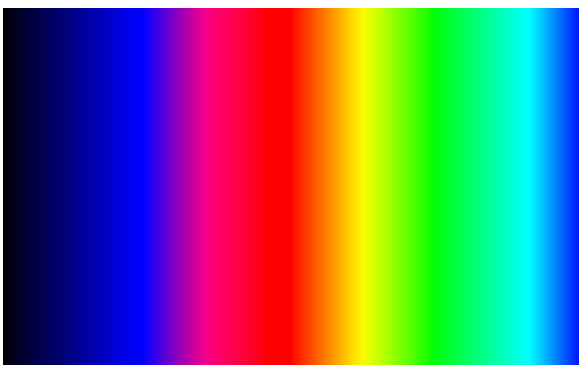
288



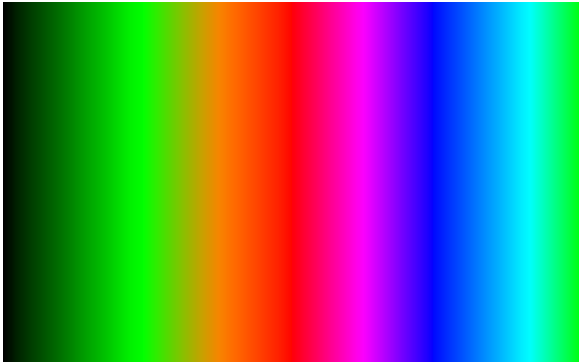
289



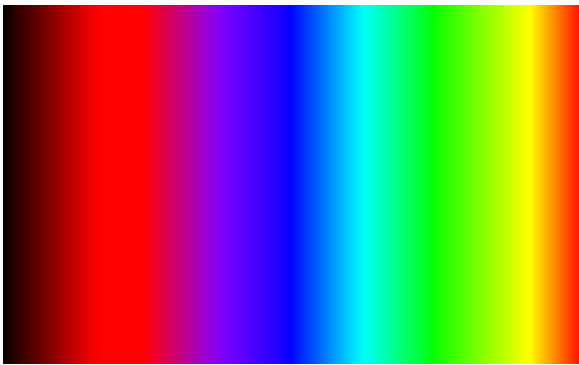
290



291



292



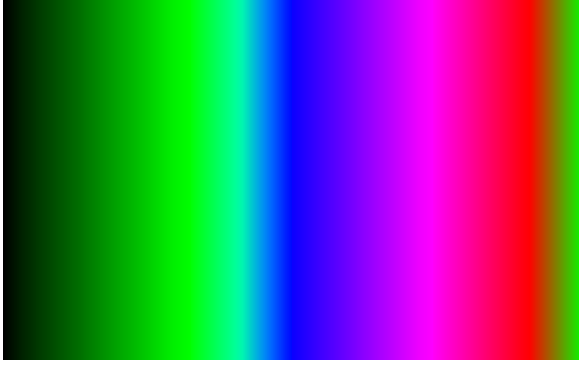
293



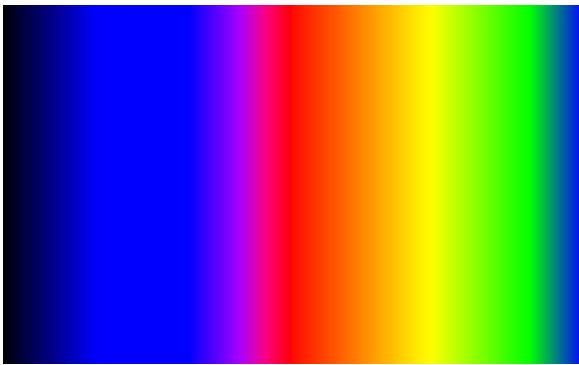
294



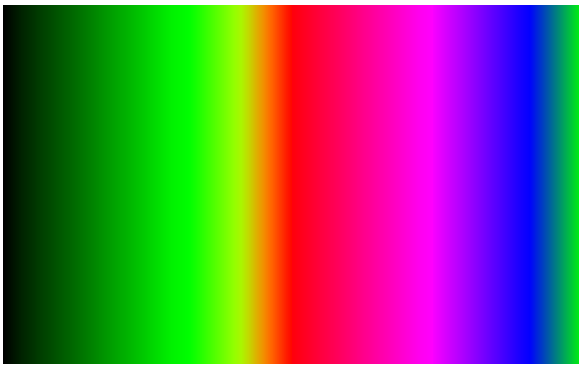
295



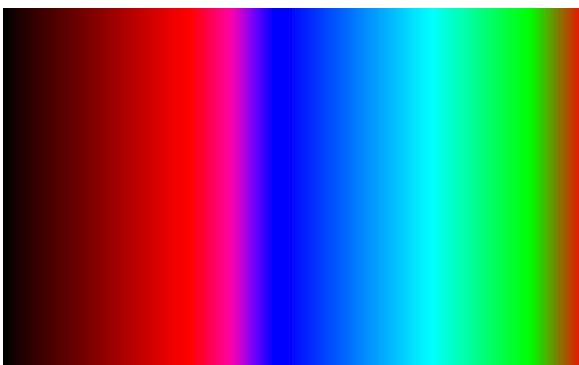
296



297



298



299