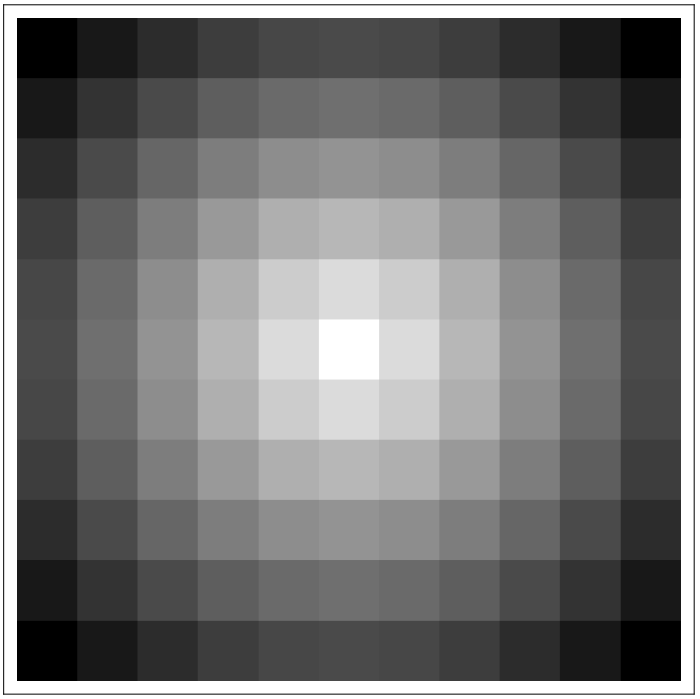
ArrayPlot

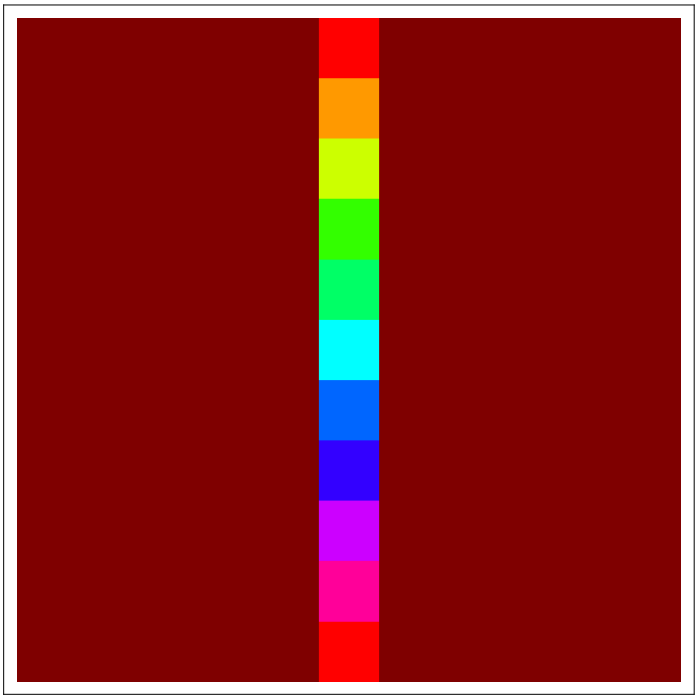
In[ ]:=
```
h = 10;
data = Table[a + b I, {a, -2, 2, 4 / h}, {b, -2, 2, 4 / h}];
       表格              虚数单位
ArrayPlot[data]
图示数组
```

Out[ ]=



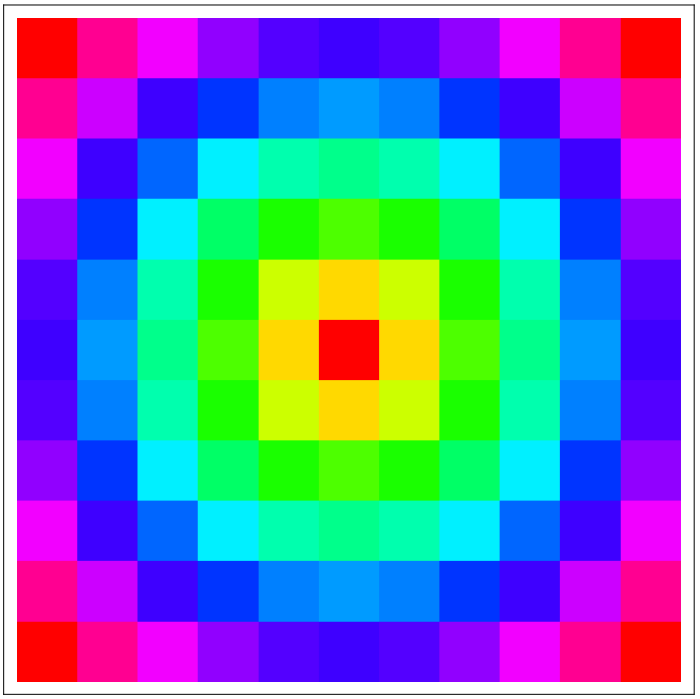
In[ ]:=
```
ArrayPlot[data, ColorFunction → Hue]
```
图示数组 颜色函数 色相

Out[ ]=



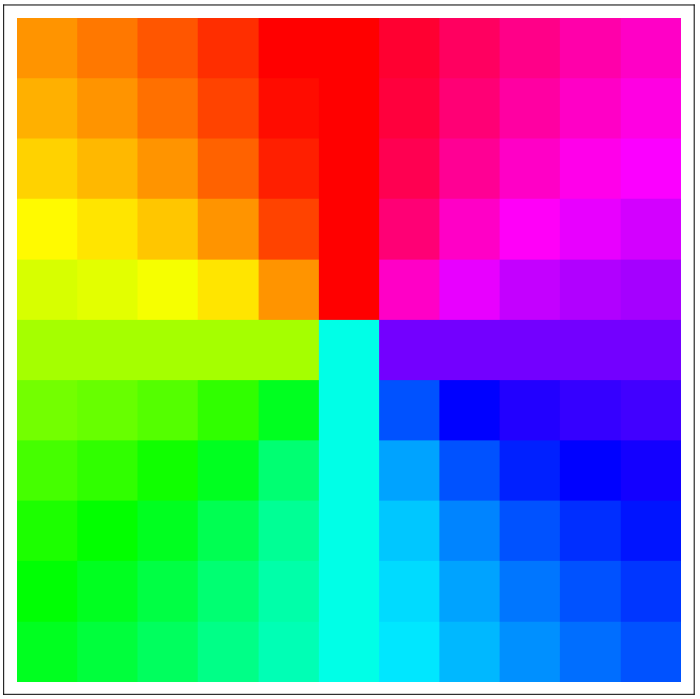
In[ ]:=
```
ArrayPlot[Abs[data], ColorFunction → Hue]
```
图示数组 绝对值 颜色函数 色相

Out[ ]=



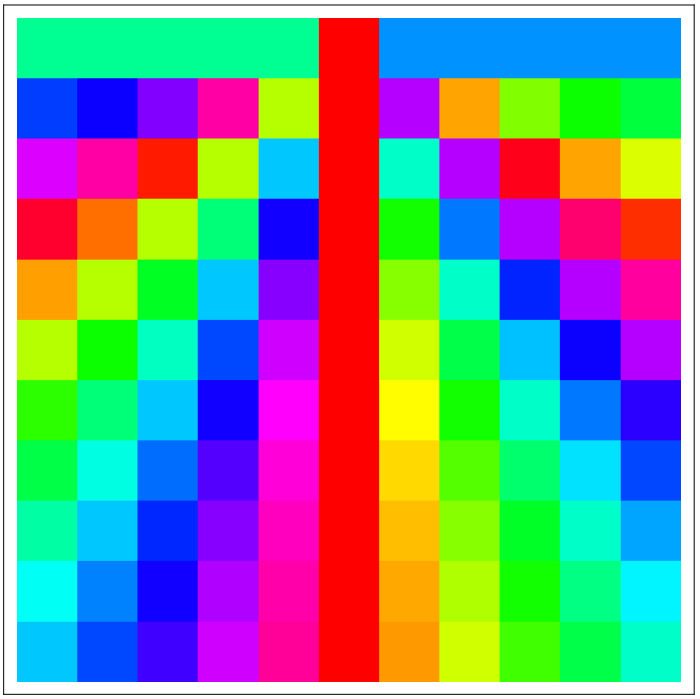
In[ ]:=
```
ArrayPlot[Arg[data], ColorFunction → Hue]
```
图示数组 辐角 颜色函数 色相

Out[ ]=



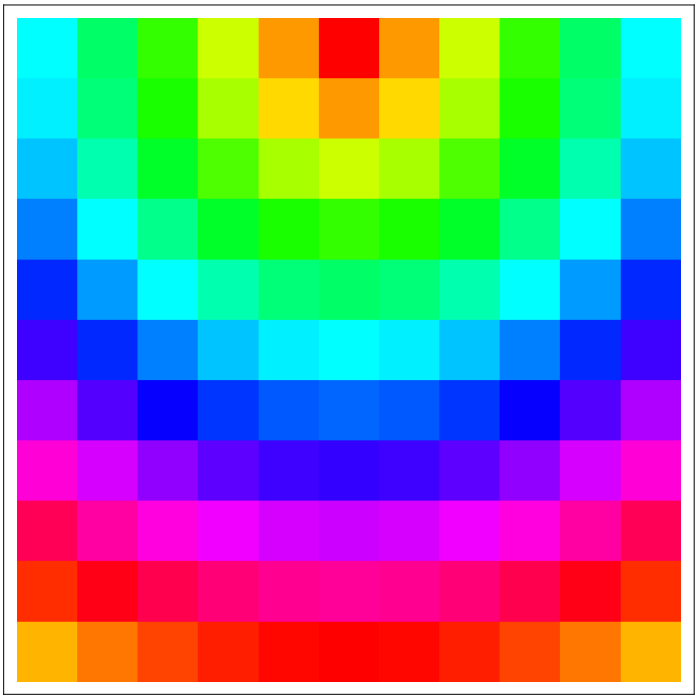
In[ ]:=
```
ArrayPlot[data, ColorFunction → (Hue[Arg[#]] &)]
```
图示数组 颜色函数 色相 辐角

Out[ ]=



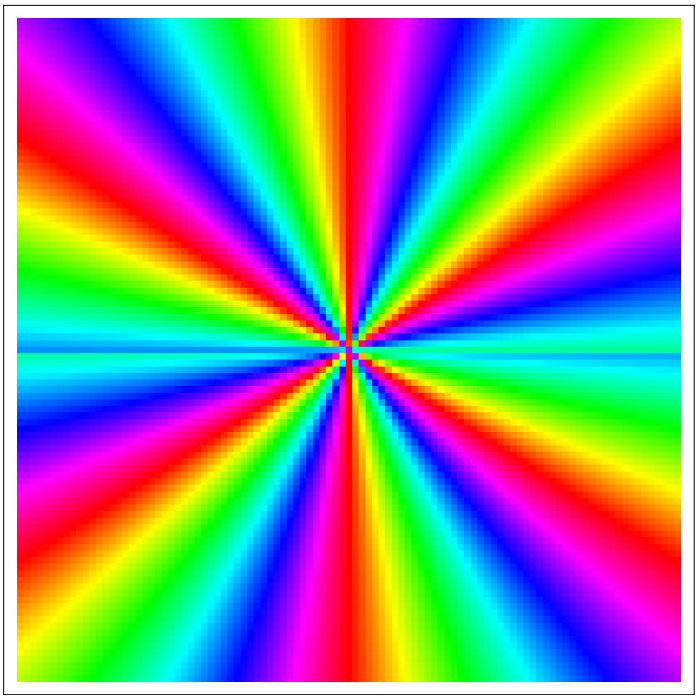
In[ ]:=
```
ArrayPlot[data, ColorFunction → (Hue[Abs[#]] &)]
```
图示数组 颜色函数 色相 绝对值

Out[ ]=



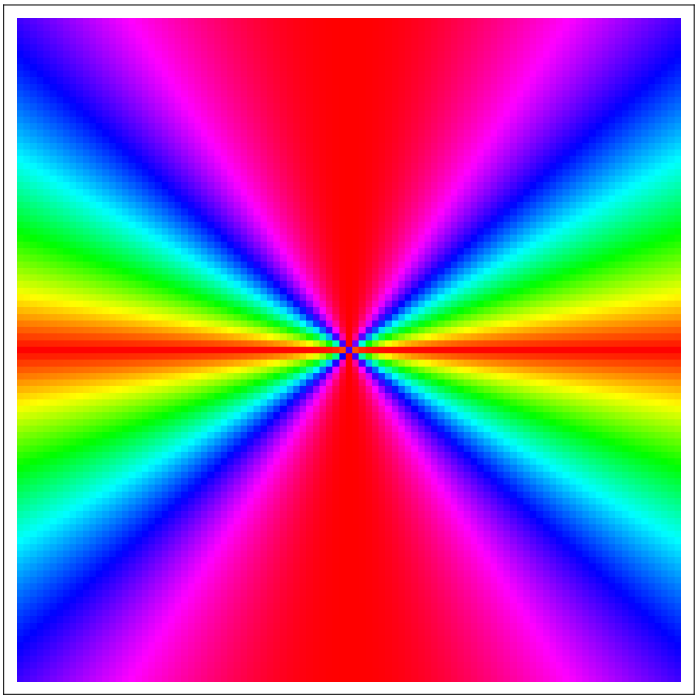
In[ ]:=
```
h = 100.1;
data = Table[(a + b i)/(a - b i), {a, -2, 2, 4/h}, {b, -2, 2, 4/h}];
ArrayPlot[data, ColorFunction → (Hue[Arg[#1]] &)]
```
表格 图示数组 颜色函数 色相 辐角

Out[ ]=



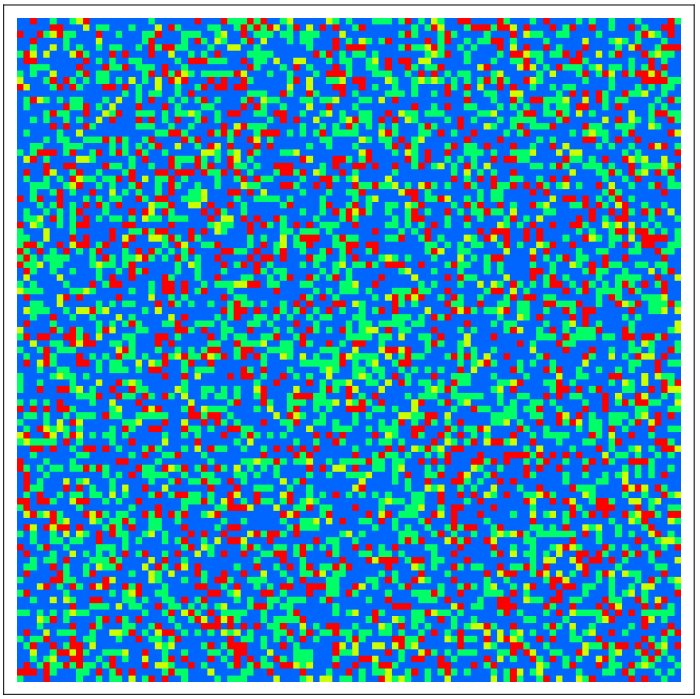
In[ ]:=
```
ArrayPlot[data, ColorFunction → (Hue[Abs[#1]] &)]
```
图示数组 颜色函数 色相 绝对值

Out[ ]=



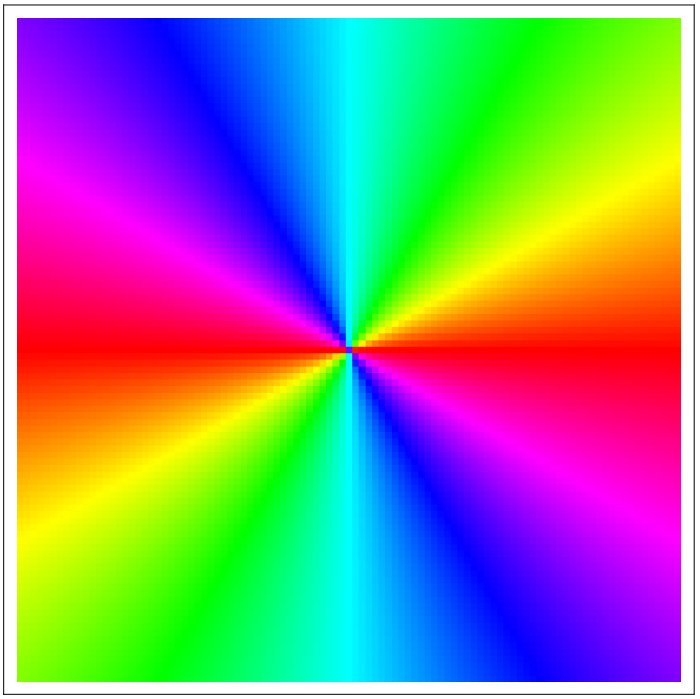
In[ ]:=
```
ArrayPlot[Abs[data], ColorFunction → Hue]
```
图示数组 绝对值 颜色函数 色相

Out[ ]=



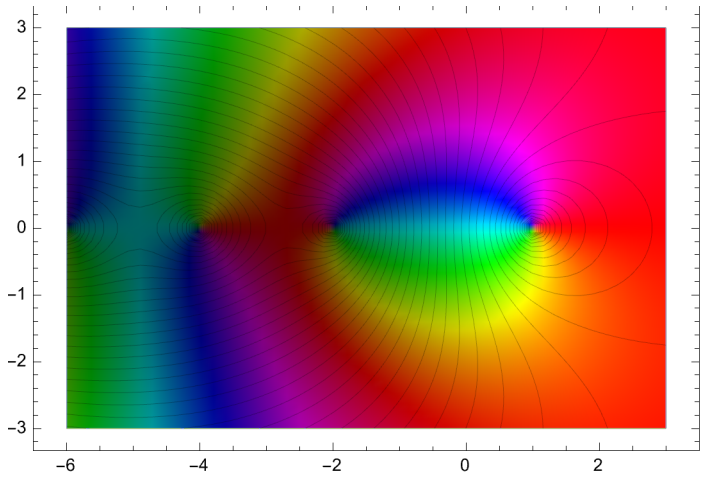
In[ ]:=
```
ArrayPlot[Arg[data], ColorFunction → Hue]
```
图示数组 辐角 颜色函数 色相

Out[ ]=



In[ ]:=
```
ParametricPlot[(*just need a vis function that will
 allow x and y to be in the color function*){x, y}, {x, -6, 3},
{y, -3, 3}, (*color and mesh functions don't trigger refinement,
so just use a big grid*) PlotPoints → 50, MaxRecursion → 0, Mesh → 50,
 (*turn off scaling so we can do computations with the actual complex values*)
ColorFunctionScaling → False,
ColorFunction → (Hue[(*hue according to argument,with shift so arg(z)==0 is red*)
    Rescale[Arg[Zeta[# + I #2]], {-Pi, Pi}, {0, 1} + 0.5], 1, (*fudge brightness a
     bit:0.1 keeps things from getting too dark,2 forces some actual bright areas*)
    Rescale[Log[Abs[Zeta[# + I #2]]], {-Infinity, Infinity}, {0.1, 2}]] &),
(*mesh lines according to magnitude,scaled to avoid the pole at z=1*)
MeshFunctions → {Log[Abs[Zeta[#1 + I #2]]] &}, (*turn off axes,
because I don't like them with frames*) Axes → False]
```

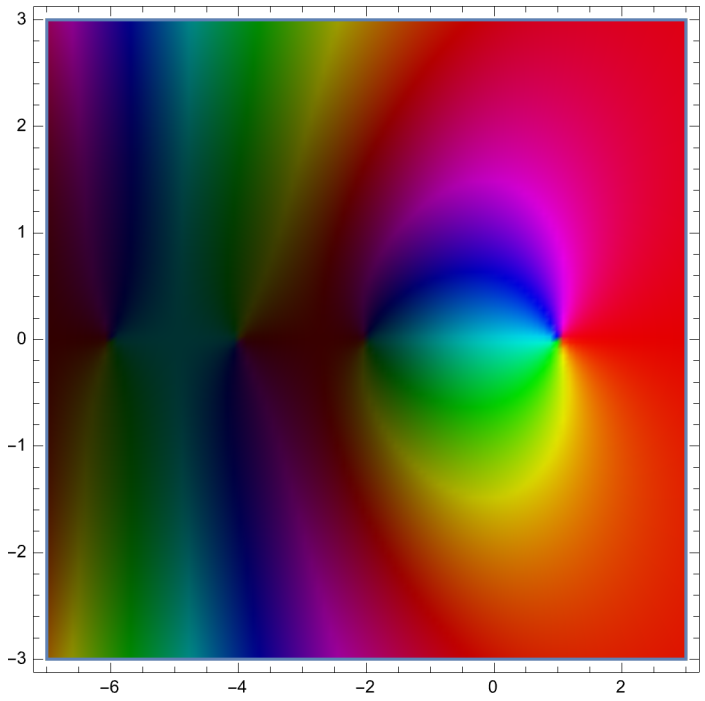
Out[ ]=



In[ ]:=
```
ComplexGraph[f_, {xmin_, xmax_}, {ymin_, ymax_}, opts : OptionsPattern[]] :=
 RegionPlot[True, {x, xmin, xmax}, {y, ymin, ymax}, opts, PlotPoints → 100,
  ColorFunctionScaling → False, ColorFunction → Function[{x, y}, With[{ff = f[x + I y]},
    Hue[(2. Pi)^-1 Mod[Arg[ff], 2 Pi], 1, 1 - (1.2 + 10 Log[Abs[ff] + 1])^-1]]]]
```

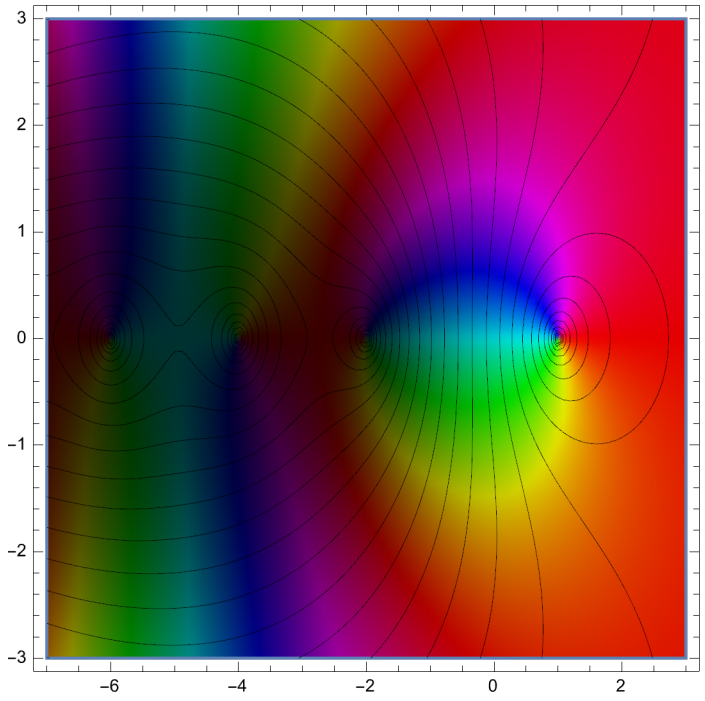
In[ ]:=
```
ComplexGraph[Zeta, {-7, 3}, {-3, 3}]
```
黎曼ζ函数

Out[ ]=



In[ ]:=
```
ComplexGraph[Zeta, {-7, 3}, {-3, 3},
  Mesh → 30, MeshFunctions → {Log[Abs[Zeta[#1 + I #2]]] &},
  MeshStyle → {{Thin, Black}, None}, MaxRecursion → 0]
```
黎曼ζ函数
网格 网格函数 对数 … 黎曼ζ函数 虚数单位
网格样式 细 黑色 无 最大递推

Out[ ]=



In[ ]:=
```
Plot3D[
Log[Abs[Zeta[x + I y]]], {x, -6, 3}, {y, -3, 3},
 (*color and mesh functions don't trigger refinement,so just use a big grid*)
 PlotPoints → 50, MaxRecursion → 0,
 Mesh → 50,
 (*turn off scaling so we can do computations with the actual complex values*)
 ColorFunctionScaling → False,
 ColorFunction → (Hue[
   (*hue according to argument,with shift so arg(z)==0 is red*)
   Rescale[Arg[Zeta[# + I #2]], {-Pi, Pi}, {0, 1} + 0.5],
   1, (*fudge brightness a bit:
      0.1 keeps things from getting too dark,
      2 forces some actual bright areas*)
   Rescale[Log[Abs[Zeta[# + I #2]]], {-Infinity, Infinity}, {0.1, 2}]] &),
 (*mesh lines according to magnitude,scaled to avoid the pole at z=1*)
 MeshFunctions → {Log[Abs[Zeta[#1 + I #2]]] &},
 (*turn off axes,because I don't like them with frames*)
 Axes → False]
```

Out[ ]=