## 2.9.3 Graphics Directives and Options

When you set up a graphics object in *Mathematica*, you typically give a list of graphical elements. You can include in that list *graphics directives* which specify how subsequent elements in the list should be rendered.

In general, the graphical elements in a particular graphics object can be given in a collection of nested lists. When you insert graphics directives in this kind of structure, the rule is that a particular graphics directive affects all subsequent elements of the list it is in, together with all elements of sublists that may occur. The graphics directive does not, however, have any effect outside the list it is in.

The first sublist contains the graphics directive `GrayLevel`.

```
In[1]:= {{GrayLevel[0.5], Rectangle[{0, 0}, {1, 1}]},
                    Rectangle[{1, 1}, {2, 2}]}
Out[1]= {{GrayLevel[0.5], Rectangle[{0, 0}, {1, 1}]},
  Rectangle[{1, 1}, {2, 2}]}
```

Only the rectangle in the first sublist is affected by the `GrayLevel` directive.

```
In[2]:= Show[Graphics[ % ]]
```

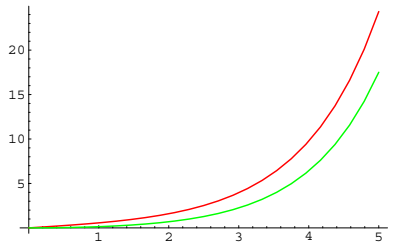
*Mathematica* provides various kinds of graphics directives. One important set is those for specifying the colors of graphical elements. Even if you have a black-and-white display device, you can still give color graphics directives. The colors you specify will be converted to gray levels at the last step in the graphics rendering process. Note that you can get black-and-white display even on a color device by setting the option `ColorOutput -> GrayLevel`.

| | |
|---|---|
| `GrayLevel[`*i*`]` | gray level between 0 (black) and 1 (white) |
| `RGBColor[`*r*`, `*g*`, `*b*`]` | color with specified red, green and blue components, each between 0 and 1 |
| `Hue[`*h*`]` | color with hue *h* between 0 and 1 |
| `Hue[`*h*`, `*s*`, `*b*`]` | color with specified hue, saturation and brightness, each between 0 and 1 |

Basic *Mathematica* color specifications.

On a color display, the two curves would be shown in color. Here they are shown in gray.

```
In[3]:= Plot[{BesselI[1, x], BesselI[2, x]}, {x, 0, 5},
          PlotStyle ->
            {{RGBColor[1, 0, 0]}, {RGBColor[0, 1, 0]}}]
```




The function `Hue[h]` provides a convenient way to specify a range of colors using just one parameter. As *h* varies from 0 to 1, `Hue[h]` runs through red, yellow, green, cyan, blue, magenta, and back to red again. `Hue[h, s, b]` allows you to specify not only the "hue", but also the "saturation" and "brightness" of a color. Taking the saturation to be equal to one gives the deepest colors; decreasing the saturation towards zero leads to progressively more "washed out" colors.

Page **??** shows examples of colors generated with various color specifications.

For most purposes, you will be able to specify the colors you need simply by giving appropriate `RGBColor` or `Hue` directives. However, if you need very precise or repeatable colors, particularly for color printing, there are a number of subtleties which arise, as discussed in Section 2.9.17 below.

When you give a graphics directive such as `RGBColor`, it affects *all* subsequent graphical elements that appear in a particular list. *Mathematica* also supports various graphics directives which affect only specific types of graphical elements.

The graphics directive `PointSize[r]` specifies that all `Point` elements which appear in a graphics object should be drawn as circles with a radius *r*. In `PointSize`, the radius *r* is measured as a fraction of the width of your whole plot.

*Mathematica* also provides the graphics directive `AbsolutePointSize[d]`, which allows you to specify the "absolute" radii of points, measured in fixed units. The units are approximately printer's points, equal to $\frac{1}{72}$ of an inch.

| | |
|---|---|
| `PointSize[r]` | give all points a radius *r* as a fraction of the width of the whole plot |
| `AbsolutePointSize[d]` | give all point a radius *d* measured in absolute units |

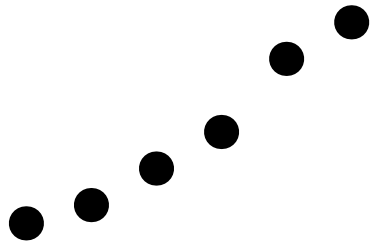
Graphics directives for points.

Here is a list of points.

```
In[4]:= Table[Point[{n, Prime[n]}], {n, 6}]
Out[4]= {Point[{1, 2}], Point[{2, 3}], Point[{3, 5}],
  Point[{4, 7}], Point[{5, 11}], Point[{6, 13}]}
```

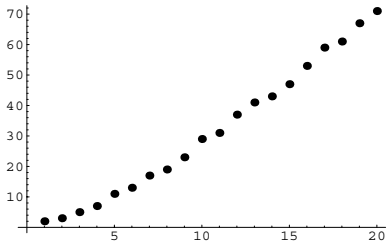
This makes each point have a radius equal to one-tenth of the width of the plot.

```
In[5]:= Show[Graphics[{PointSize[0.1], %}], PlotRange -> All]
```



Here each point has size 3 in absolute units.

```
In[6]:= ListPlot[Table[Prime[n], {n, 20}],
                 Prolog -> AbsolutePointSize[3]]
```




| | |
|---|---|
| `Thickness[`$r$`]` | give all lines a thickness $r$ as a fraction of the width of the whole plot |
| `AbsoluteThickness[`$d$`]` | give all lines a thickness $d$ measured in absolute units |
| `Dashing[{`$r_1$`, `$r_2$`, ... }]` | show all lines as a sequence of dashed segments, with lengths $r_1$, $r_2$, ... |
| `AbsoluteDashing[{`$d_1$`, `$d_2$`, ... }]` | use absolute units to measure dashed segments |

Graphics directives for lines.

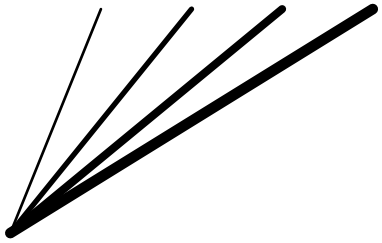
Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

This generates a list of lines with different absolute thicknesses.

```
In[7]:= Table[
          {AbsoluteThickness[n], Line[{{0, 0}, {n, 1}}]}, {n, 4}]
Out[7]= {{AbsoluteThickness[1], Line[{{0, 0}, {1, 1}}]},
  {AbsoluteThickness[2], Line[{{0, 0}, {2, 1}}]},
  {AbsoluteThickness[3], Line[{{0, 0}, {3, 1}}]},
  {AbsoluteThickness[4], Line[{{0, 0}, {4, 1}}]}}
```

Here is a picture of the lines.

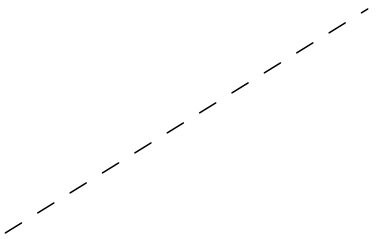
```
In[8]:= Show[Graphics[%]]
```



The `Dashing` graphics directive allows you to create lines with various kinds of dashing. The basic idea is to break lines into segments which are alternately drawn and omitted. By changing the lengths of the segments, you can get different line styles. `Dashing` allows you to specify a sequence of segment lengths. This sequence is repeated as many times as necessary in drawing the whole line.

This gives a dashed line with a succession of equal-length segments.

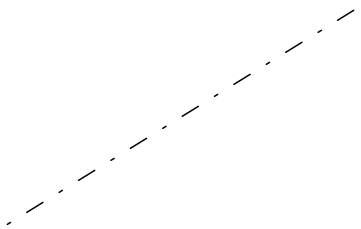
```
In[9]:= Show[Graphics[ {Dashing[{0.05, 0.05}],
                        Line[{{-1, -1}, {1, 1}}]} ]]
```



Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

This gives a dot-dashed line.

```
In[10]:= Show[Graphics[{Dashing[{0.01, 0.05, 0.05, 0.05}],
                        Line[{{-1, -1}, {1, 1}}]}]]
```



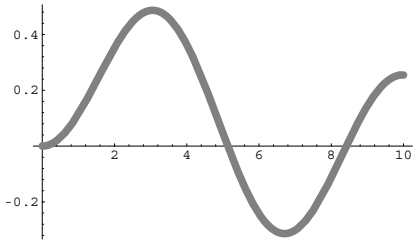
One way to use *Mathematica* graphics directives is to insert them directly into the lists of graphics primitives used by graphics objects. Sometimes, however, you want the graphics directives to be applied more globally, and for example to determine the overall "style" with which a particular type of graphical element should be rendered. There are typically graphics options which can be set to specify such styles in terms of lists of graphics directives.

| | |
|---|---|
| `PlotStyle ->` *style* | specify a style to be used for all curves in `Plot` |
| `PlotStyle -> {{`$style_1$`}, {`$style_2$`}, ... }` | specify styles to be used (cyclically) for a sequence of curves in `Plot` |
| `MeshStyle ->` *style* | specify a style to be used for a mesh in density and surface graphics |
| `BoxStyle ->` *style* | specify a style to be used for the bounding box in three-dimensional graphics |

Some graphics options for specifying styles.

This generates a plot in which the curve is given in a style specified by graphics directives.

```
In[11]:= Plot[BesselJ[2, x], {x, 0, 10},
           PlotStyle -> {{Thickness[0.02], GrayLevel[0.5]}}]
```




Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

| | |
|---|---|
| `GrayLevel[0.5]` | gray |
| `RGBColor[1, 0, 0]`, etc. | red, etc. |
| `Thickness[0.05]` | thick |
| `Dashing[{0.05, 0.05}]` | dashed |
| `Dashing[{0.01, 0.05, 0.05, 0.05}]` | dot-dashed |

Some typical styles.

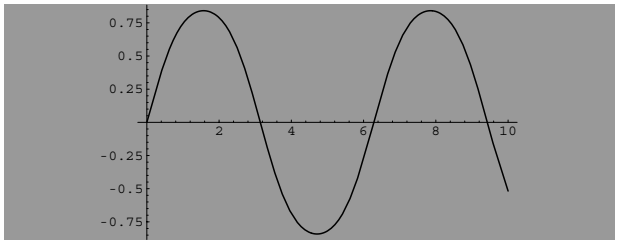
The various "style options" allow you to specify how particular graphical elements in a plot should be rendered. *Mathematica* also provides options that affect the rendering of the whole plot.

| | |
|---|---|
| `Background ->` *color* | specify the background color for a plot |
| `DefaultColor ->` *color* | specify the default color for a plot |
| `Prolog ->` *g* | give graphics to render before a plot is started |
| `Epilog ->` *g* | give graphics to render after a plot is finished |

Graphics options that affect whole plots.

This draws the whole plot on a gray background.

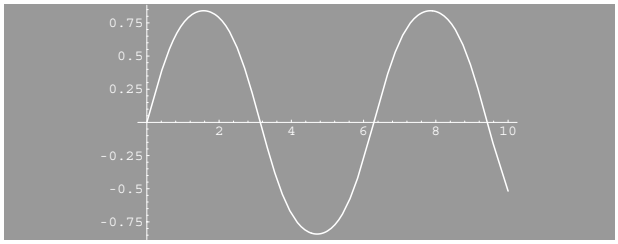
```
In[12]:= Plot[Sin[Sin[x]], {x, 0, 10},
            Background -> GrayLevel[0.6]]
```




Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.

This makes the default color white.

```
In[13]:= Show[%, DefaultColor -> GrayLevel[1]]
```




Web sample page from The *Mathematica* Book, Second Edition, by Stephen Wolfram, published by Addison-Wesley Publishing Company (hardcover ISBN 0-201-51502-4; softcover ISBN 0-201-51507-5). To order *Mathematica* or this book contact Wolfram Research: info@wolfram.com; http://www.wolfram.com/; 1-800-441-6284.