

CustomTicks package

Mark A. Caprio, Department of Physics, University of Notre Dame

Version 1.82 (July 15, 2012)

Introduction

Mathematica provides a powerful system for generating graphics but does not provide, in built-in form, the fine formatting control necessary for the preparation of publication quality figures. The CustomTicks package provides detailed customization of tick mark placement and formatting. The flexibility achieved matches or exceeds that available with most commercial scientific plotting software. Linear, logarithmic, and general nonlinear axes are supported. Some tick mark manipulation functions, for use in graphics programming, are also provided by the CustomTicks package.

The CustomTicks package was developed as part of the LevelScheme system for preparing publication-quality scientific figures [Comput. Phys. Commun. **171**, 107 (2005)].

Basic use for linear axes

The default tick marks produced by *Mathematica*'s plotting functions are typically not ideal for publication. It is often desirable to be able to change the tick spacing from that selected by *Mathematica*. The tick marks are also often too short to be easily visible.

It is possible to override the default *Mathematica* ticks by specifying a list of tick marks, complete with formatting information, as the value for the `Ticks` or `FrameTicks` option (see the *Mathematica* documentation for basic plotting options). It is prohibitively tedious to construct such lists by hand. The CustomTicks package provides functions to automatically construct lists of tick marks, with detailed control over formatting.

<code>LinTicks[x1 , x2]</code>	Produces linear tick specifications, with automatically chosen major and minor tick intervals
<code>LinTicks[x1 , x2 , interval , subdivs]</code>	Produces linear tick specifications, with manually chosen major and minor tick intervals

Tick specification function.

<i>option name</i>	<i>default value</i>	
<code>TickRange</code>	<code>{-Infinity, Infinity}</code>	Limits the drawing of ticks (and their labels) to given coordinate range
<code>ShowMinorTicks</code>	<code>True</code>	Controls whether or not the minor ticks are drawn; mainly for use with <code>LogTicks</code> (see below)

Options controlling the coordinates at which tick marks are displayed.

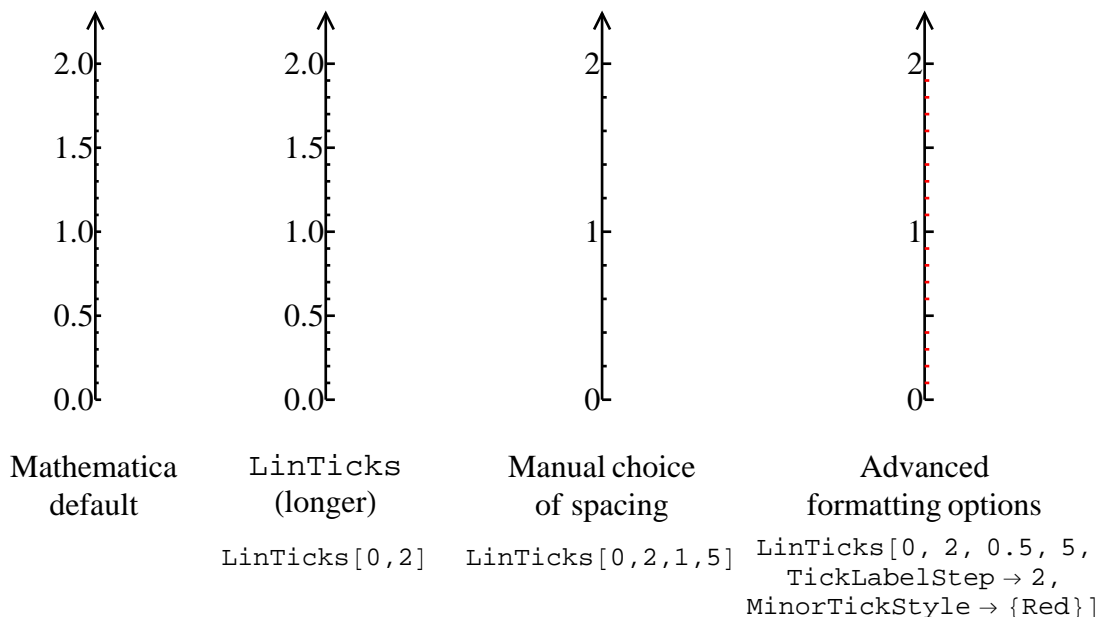
<i>option name</i>	<i>default value</i>	
ShowTickLabels	True	Controls whether or not major tick labels are printed
TickLabelRange	{-Infinity, Infinity}	Limits printing of major tick labels to given coordinate range
ShowFirst	True	Controls whether or not first major tick label is printed
ShowLast	True	Controls whether or not last major tick label is printed
TickLabelStep	1	Limits printing of major tick labels to one in every TickLabelStep major ticks
TickLabelStart	0	Used in conjunction with TickLabelStep chooses which subset of major tick labels are printed

Options controlling which tick marks are accompanied by labels.

<i>option name</i>	<i>default value</i>	
MajorTickLength	0.01	Length for the major ticks (may also be given as a list of two lengths, into and out of the frame, as described in the <i>Mathematica</i> documentation for Ticks)
MinorTickLength	0.005	Length for the minor ticks (may also be given as a list of two lengths, into and out of the frame, as described in the <i>Mathematica</i> documentation for Ticks)
TickDirection	In	Orientation of tick marks (In for tick marks into the frame, Out for tick marks out of the frame, or All for tick marks both into and out of the frame)
TickLengthScale	1	Additional scale factor by which to lengthen both the major and minor ticks, relative to the lengths given by the options MajorTickLength and MinorTickLength above
MajorTickStyle	{}	List specifying the line style for the major ticks
MinorTickStyle	{}	List specifying the line style for the minor ticks
DecimalDigits	Automatic	Sets number of digits after decimal place for major tick labels; if Automatic, the maximum number of digits needed for any label is used

Options specifying the appearance of tick marks and labels.

The `LinTicks` function, in its simplest form, is given a starting and an ending coordinate value as its arguments. It then generates the same tick marks *Mathematica* would have for this coordinate range, except that the tick marks lines are somewhat longer. Alternatively, arguments may be given to manually specify the coordinate interval between major tick marks and the number of minor subdivisions. Several further options, listed above, can be specified. These control which tick marks are drawn, which major ticks have labels, and the formatting of the tick lines and labels. Some examples are shown below.



In Mathematica versions 5 and earlier, Mathematica dropped trailing zeros after the decimal point in its default tick marks, leading to a series of ticks of “ragged” lengths (e.g., “0.”, “0.5”, “1.”, ...), which were particularly unsuited for publication. The CustomTicks package therefore provided not simply finer control and customization but also much “neater” formatting than the Mathematica default.

If you are using Mathematica 5 or an earlier version, the rightmost example assumes the Mathematica color names package (Graphics`Colors`) has been loaded.

The tick specifications generated with the CustomTicks package can be used either with two-dimensional plots (through the options Ticks or FrameTicks) or with three-dimensional plots (through the option Ticks). These options are described further in the basic Mathematica documentation on plotting.

Logarithmic axes

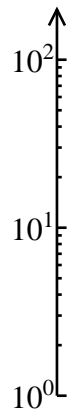
The function LogTicks generates tick marks for logarithmic axes. LogTicks can produce tick marks for an arbitrary logarithmic base (10 is the default, but e and 2 are other commonly useful bases). Unlike the Mathematica LogPlot function, which produces cumbersome decimal labels (e.g., “0.0000001”, “0.000001”, ...), LogTicks produces true exponential labels (e.g., “ 10^{-7} ”, “ 10^{-6} ”, ...).

```

LogTicks[ n1 , n2] Produces logarithmic tick specifications, base 10
LogTicks[ base , n1 , n2] Produces logarithmic tick specifications, arbitrary base
    
```

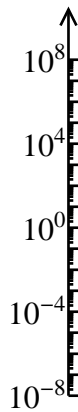
Logarithmic tick specification function.

LogTicks must be given the starting power, ending power, and, optionally, the logarithmic base b . For base 10, a total of eight minor ticks are produced in each major interval, at 2×10^n through 9×10^n . For an arbitrary base b , $[b]-2$ minor ticks are produced, at $2 \times b^n$, $3 \times b^n$, ... Display of the minor ticks may be suppressed by specifying the option ShowMinorTicks -> False. Some examples are shown below.



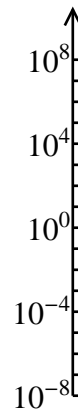
Base 10

```
LogTicks[0, 2]
```



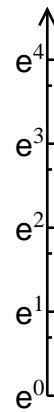
Skipped labels

```
LogTicks[-8, 8,  
TickLabelStep -> 4]
```



No minor ticks

```
LogTicks[-8, 8,  
ShowMinorTicks -> False,  
TickLabelStep -> 4]
```



Base e

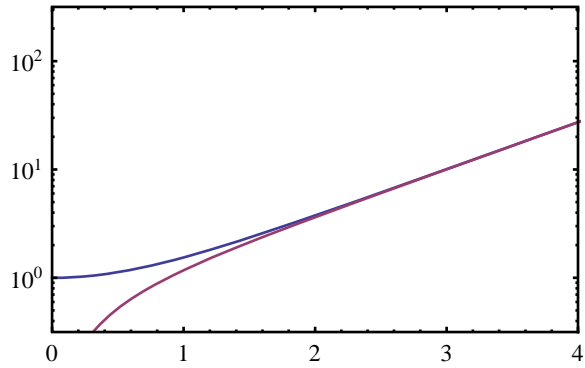
```
LogTicks[E, 0, 4]
```

Note that plots with *logarithmic* axes are actually generated as *linear* plots, but where the logarithm has been taken of either the x -axis or y -axis variable. Specifically, for base 10,

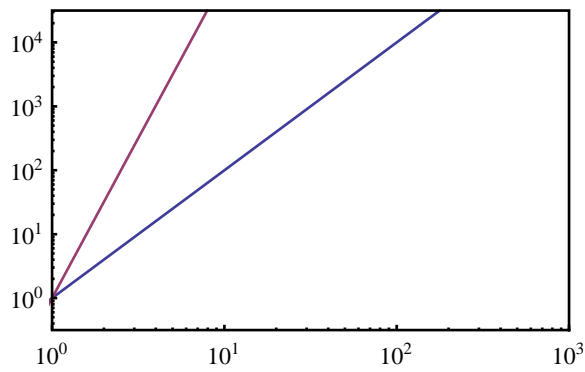
- (1) a *logarithmic* (or *linear-log*) plot of f is obtained by plotting $\log_{10} f(x)$,
- (2) a *log-linear* plot of f is obtained by plotting $f(10^x)$, and
- (3) a *log-log* plot of f is obtained by plotting $\log_{10} f(10^x)$

on ordinary linear axes. A similar procedure holds for bases other than 10. Examples of a logarithmic plot and a log-log plot follow.

```
Plot[  
  {Log[10, Cosh[x]], Log[10, Sinh[x]]}, {x, 0, 10},  
  PlotRange -> {{-0.0001, 4}, {-0.5, 2.5}},  
  FrameTicks -> {  
    LinTicks[0, 4],  
    LogTicks[10, -1, 3],  
    LinTicks[0, 4, ShowTickLabels -> False],  
    LogTicks[10, -1, 3, ShowTickLabels -> False]  
  },  
  Axes -> False, Frame -> True, ImageSize -> 72*3  
]
```



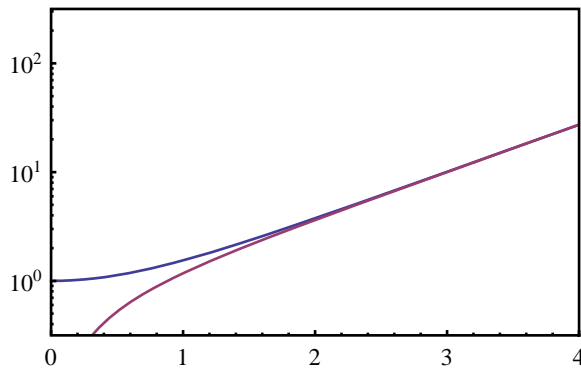
```
Plot[
  {Log[10, (10^x)^2], Log[10, (10^x)^5]}, {x, -1, 3},
  PlotRange -> {{-0.0001, 3}, {-0.5, 4.5}},
  FrameTicks -> {
    LogTicks[10, 0, 3],
    LogTicks[10, -1, 5],
    None,
    None
  },
  Axes -> False, Frame -> True, ImageSize -> 72*3
]
```



Automatic ticks for *Mathematica* plot functions

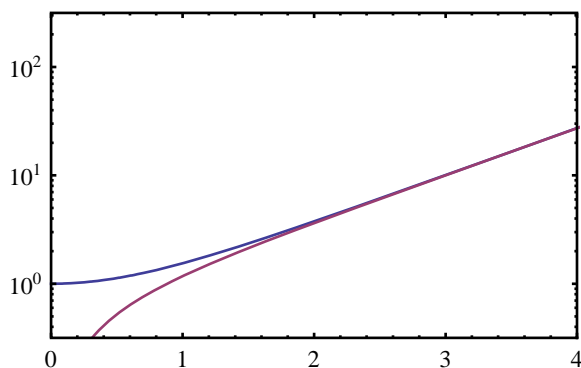
The functions `LinTicks` and `LogTicks` can also be specified as automatic tick generation functions for the *Mathematica* plotting functions. This saves you typing the plot range explicitly each time, at least if you do not wish to specify details such as the number of subdivisions.

```
Plot[
  {Log[10, Cosh[x]], Log[10, Sinh[x]]}, {x, 0, 10},
  PlotRange -> {{-0.0001, 4}, {-0.5, 2.5}},
  FrameTicks -> {LinTicks, LogTicks, None, None},
  Axes -> False, Frame -> True, ImageSize -> 72*3
]
```



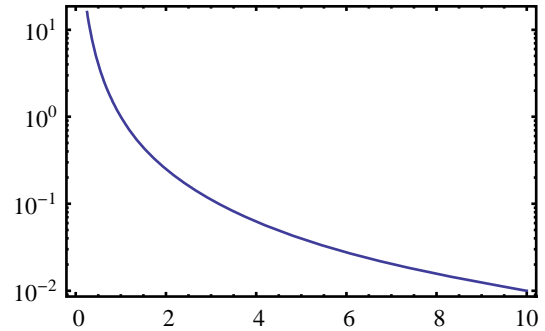
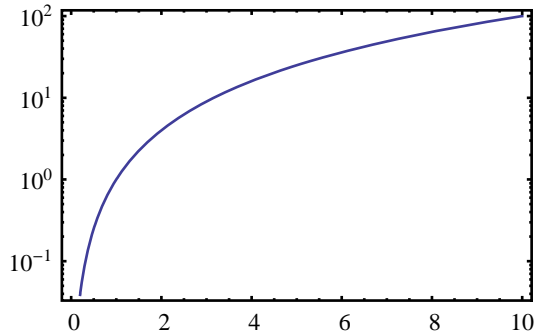
What if you wish to have tick marks on the top and right as well, but without labels on them? Simply specifying `FrameTicks -> {LinTicks, LogTicks}` or `FrameTicks -> {LinTicks, LogTicks, LinTicks, LogTicks}` would unfortunately result in ticks with unsightly and redundant labels. This can be avoided with the `CustomTicks StripTickLabels` function, as shown below.

```
Plot[
  {Log[10, Cosh[x]], Log[10, Sinh[x]]}, {x, 0, 10},
  PlotRange -> {{-0.0001, 4}, {-0.5, 2.5}},
  FrameTicks ->
  {LinTicks, LogTicks, StripTickLabels[LinTicks], StripTickLabels[LogTicks]},
  Axes -> False, Frame -> True, ImageSize -> 72*3
]
```



If you are doing many such plots, it is easiest to set the necessary options as the default options for `Plot`.

```
SetOptions[Plot, Axes->False, Frame->True, FrameTicks->
{LinTicks, LogTicks, StripTickLabels[LinTicks], StripTickLabels[LogTicks]};
GraphicsGrid[{{Plot[Log[10, x^2], {x, 0, 10}], Plot[Log[10, x^-2], {x, 0, 10}]}}
```

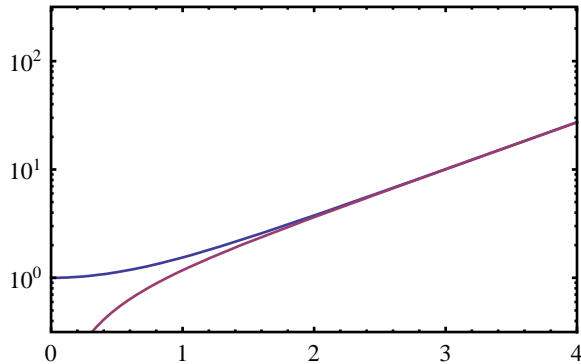


option name	default value	
LogPlot	False	Controls whether LogTicks operates as it should for standalone use or for automatic tick mark generation with <i>Mathematica</i> 's LogPlot

LogTicks special option.

If you wish to use LogPlot or related *Mathematica* functions directly for your logarithmic plots, you may also do so. However, LogTicks must then be instructed to adjust its output accordingly. (LogPlot expects tick coordinates to be specified as the true coordinate value, not the logarithm of the coordinate value. LogTicks must also interpret its *n1* and *n2* arguments differently in this case.) Automatic use with LogPlot can be accomplished by setting the option LogPlot->True.

```
SetOptions[LogTicks, LogPlot -> True];
Plot[
{Cosh[x], Sinh[x]}, {x, 0, 10},
PlotRange -> {{-0.0001, 4}, {10^-0.5, 10^2.5}},
FrameTicks -> {LinTicks, LogTicks, StripTickLabels[LinTicks],
StripTickLabels[LogTicks]},
Frame -> True, ImageSize -> 72*3
]
```



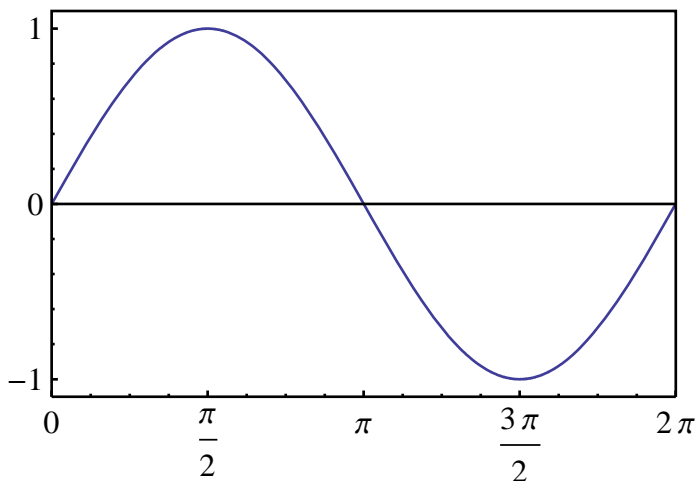
Further control over tick placement and advanced customization

option name	default value	
TickLabelFunction	Automatic	Function used to generate major tick labels (first argument is the numerical coordinate, second argument is the <code>LinTicks</code> default formatted label); <code>Automatic</code> gives the default label
TickPreTransformation	Identity	Function to be applied to tick coordinates, before range tests and label generation
TickPostTransformation	Identity	Function to be applied to tick coordinates, after range tests and label generation
MinorTickIndexTransformation	Identity	Function to be applied to minor tick indices (originally 1, 2, ..., <i>subdivs</i> - 1) before minor tick coordinate is obtained by linear interpolation between major tick positions
MinorTickIndexRange	{1, Infinity}	Limits drawing of minor ticks to those with indices (before transformation) in given range

Advanced customization options.

`LinTicks` accepts several options for advanced customization, allowing fully customizable labels and general nonlinear axis scales. The option `TickLabelFunction` is used to specify the function to be used to construct tick labels (see the *Mathematica* documentation for `Function` for information on defining functions). The label function is given as arguments both the raw numerical tick coordinate and the `LinTicks` default formatted label, so it can work with whichever is more convenient. The label function may be used for simple tasks, such as attaching a prefix or suffix to the usual default label, or for more sophisticated formatting. In the following example, tick values are formatted as rational multiples of π .

```
LinTicks[0, 2*Pi, Pi/2, 4, TickLabelFunction -> (Rationalize[#/Pi]*Pi&)]
```

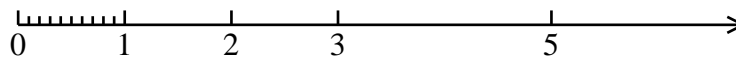
Nonlinear axes are constructed using the coordinate transformation functions. For instance, the `LogTicks` function provided by the `CustomTicks` package is actually implemented as a special case of `LinTicks`, with transformed minor tick positions. A simplified version (base 10 logarithm only) is given below for illustration.

```
Log10Ticks[p1_Integer, p2_Integer, Opts___?OptionQ] := LinTicks[
  p1, p2, 1, 9,
  TickLabelFunction -> (DisplayForm[SuperscriptBox[10, IntegerPart[#]]] &),
  MinorTickIndexTransformation -> (Log[10, # + 1] * 9 &),
  Opts
];
```

`LinTicks[
majorticklist, minorticklist]` Produces major and minor ticks at the specified coordinate values

Form of tick specification function for ticks at arbitrary locations.

Ticks may be placed at arbitrary coordinate locations by using the most flexible form of `LinTicks`, in which all major and minor tick coordinates are specified explicitly in two lists. All the usual coordinate-transformation and customization options described above (except `MinorTickIndexRange`) are still applicable.



Manual choice of tick coordinates

```
LinTicks[{0, 1, 2, 3, 5}, Range[0.1, 0.9, 0.1]]
```

<i>option name</i>	<i>default value</i>	
ExtraTicks	{}	Additional coordinate values at which to insert tick marks
TickTest	(True &)	Logical test to be applied to coordinate values, to select the coordinates at which tick marks are displayed (provides further control beyond the simple range test provided by TickRange above)
TickLabelTest	(True &)	Logical test to be applied to coordinate values, to select the coordinates at which tick labels are displayed (provides further control beyond the simple range test provided by TickLabelRange above)

Further options controlling the placement of tick marks and labels.

Or, even without taking complete manual control of the choice of tick positions, some further control over tick placement is provided through the options above.

Tick mark programming utilities

LimitTickRange[<i>{ x1, x2 }, ticks</i>]	Selects those tick mark with coordinates in the range specified; approximate equality testing is used to avoid dropping ticks at the ends of the interval due to roundoff; ticks must be specified as lists rather than bare numbers
TransformTicks[<i>coordfcn, lengthfcn, ticks</i>]	Applies the specified transformation functions to the tick coordinates and tick lengths, respectively; ticks must be specified with an explicit pair of in and out lengths
StripTickLabels[<i>ticks</i>]	Removes any text labels from ticks; ticks must be specified as lists rather than bare numbers
AugmentTicks[<i>labelfcn, { l1, l2 }, stylelist, ticks</i>]	Upgrades all tick specifications to full specifications, complete with labels, lengths into and out of the frame (default 0 for out), and style directives
AugmentAxisTickOptions[<i>numaxes, tickoptions</i>]	Given a list of tick options (themselves lists of tick specifications) for several axes, replaces any None entries with null lists and appends additional null lists as needed to make <i>numaxes</i> entries; a value None for <i>tickoptions</i> is replaced by a list of null lists
TickQ[<i>x</i>]	Tests whether or not <i>x</i> is a valid tick mark specification
TickListQ[<i>x</i>]	Tests whether or not <i>x</i> is a list of valid tick mark specifications

Tick manipulation utilities.

Several utility functions for tick mark manipulation and testing are provided. These are mainly intended for use in graphics programming rather than for direct use by someone wishing to specify tick marks. They are used internally by the LevelScheme figure preparation system.

FractionDigits[<i>x</i>]	Returns the number of digits to the right of the point in the decimal representation of <i>x</i>
----------------------------	--

Decimal digit counting function.

FractionDigits determines the number of digits to the right of the point in the decimal representation of a number. It is of use in constructing fixed-point tick labels. It will, naturally, return large values, determined by *Mathematica's* Precision, for some numbers, such as non-terminating rationals. It accepts the option FractionDigitsBase, by default 10, for work with non-decimal representations. Some examples follow:

```

FractionDigits[100]
FractionDigits[1.25]
FractionDigits[1 / 3]

0

2

17

```

```

FixedPointForm[ x, r ] Formats x as a fixed-point number with r digits to the right of the decimal point.
FixedPointForm[ x, { l, r } ] Formats x as a fixed-point number with l digits
(or spaces) to the left and r digits to the right of the decimal point.

```

Decimal digit counting function.

`FixedPointForm` returns a string, consisting of the real number x formatted in fixed-point representation. It is used internally by CustomTicks in constructing fixed-point tick labels, hence its inclusion in this package, but it may be used to format numbers in many other contexts as well. Some examples follow:

```

FixedPointForm[Pi // N, 5]
FixedPointForm[Pi // N, 2]
FixedPointForm[Pi // N, 0]

3.14159

3.14

3

FixedPointForm[Pi // N, {2, 3}]
FixedPointForm[4 * Pi // N, {2, 3}]

3.142

12.566

```

By default, for positive numbers a blank padding space appears at left, where a minus sign would be for negative numbers, to improve alignment with negative numbers. However, with `NumberSigns -> Automatic` this space is suppressed.

```

FixedPointForm[Pi // N, 3]
FixedPointForm[-Pi // N, 3]
FixedPointForm[Pi // N, 3, NumberSigns -> Automatic]

3.142

-3.142

3.142

```

Technical notes: `FixedPointForm` allows as many digits as necessary to the left of the decimal point, thereby avoiding the rounding problem associated with `PaddedForm[x, {n, f}]` when n is specified too small (`PaddedForm` zeros out some of the rightmost digits of the number). It also suppresses any trailing decimal point when $r=0$.

© Copyright 2012, Mark A. Caprio.