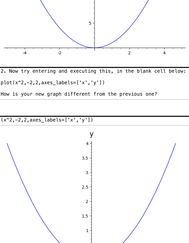
Functions

last edited Sep 8, 2009, 1:10:21 PM by kcrisp02

Functions systems sage

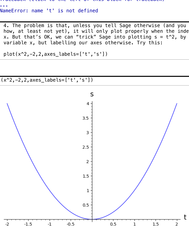
```
#  1. One of the things Sage can do is graph functions given a formula. To plot the function
#  y = x^2 on the domain [-5, 5], type the following into the empty "cell" (input box) below:
#
#
#    plot(x^2,-5,5)
#
#  and press "Evaluate" (or Shift+Enter)
```

```
plot(x^2,-5,5)
```



```
#  2. Now try entering and executing this, in the blank cell below:
#
#    plot(x^2,-2,2,axes_labels=['x','y'])
#
#  How is your new graph different from the previous one?
```

```
plot(x^2,-2,2,axes_labels=['x','y'])
```



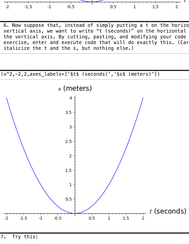
```
#  3. Now suppose that, instead, we wanted to plot the function s = t^2. Try entering
#
#
#    plot(t^2,-2,2,axes_labels=['t','s'])
#
#  and executing. What goes wrong?
```

```
plot(t^2,-2,2,axes_labels=['t','s'])
```

```
Traceback (click to the left of this block for traceback)
...
NameError: name 't' is not defined
```

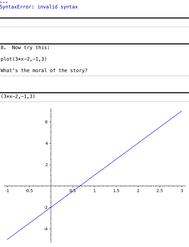
```
#  4. The problem is that, unless you tell Sage otherwise (and you can, but we won't discuss
#  how to do that yet), Sage thinks that the variable in the independent variable is called
#  "x". So that's OK, we can "trick" Sage into plotting s = t^2, by calling the independent
#  variable x, but labelling our axes otherwise. Try this:
#
#    plot(x^2,-2,2,axes_labels=['t','s'])
```

```
plot(x^2,-2,2,axes_labels=['t','s'])
```



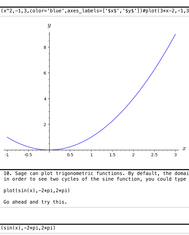
```
#  5. In mathematics, it's traditional to italicize variable names. To do this in the axes
#  labels on a Sage plot, put dollar signs around these names. Try this:
#
#    plot(x^2,-2,2,axes_labels=['$t$','$s$'])
```

```
plot(x^2,-2,2,axes_labels=['$t$','$s$'])
```



```
#  6. Now suppose that, instead of simply putting a t on the horizontal axis and an s on the
#  vertical axis, we want to write "t (seconds)" on the horizontal axis, and "s (meters)" on
#  the vertical axis. By cutting, pasting, and modifying your code from the previous
#  exercise, enter and execute code that will do exactly this. (Careful: you want to
#  italicize the t and the s, but nothing else.)
```

```
plot(x^2,-2,2,axes_labels=['$t$ (seconds)','$s$ (meters)'])
```



```
#  7. Try this:
#
#    plot(3x-2,-1,3)
#
#  What goes wrong?
```

```
plot(3x-2,-1,3)
```

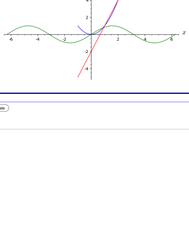
```
Traceback (click to the left of this block for traceback)
...
SyntaxError: invalid syntax
```

```

```

```
#  8. Now try this:
#
#    plot(3*x-2,-1,3)
#
#  What's the moral of the story?
```

```
plot(3*x-2,-1,3)
```



```

```

```
#  9. Plotting two or more functions is easy: just put plus signs in between "plot" commands.  #  Try it, with these two commands
# we've added some color, to distinguish the two graphs!
#
#    plot(x^2,-1,3,color='blue',axes_labels=['$x$','$y$'])
#
#    plot(3*x-2,-1,3,color='red')
```

```
plot(x^2,-1,3,color='blue',axes_labels=['$x$','$y$'])+plot(3*x-2,-1,3,color='red')
```



```
#  10. Sage can plot trigonometric functions. By default, the domain is always in radians, so
#  in order to see two cycles of the sine function, you could type
#
#
#    plot(sin(x),-2*pi,2*pi)
#
#  Go ahead and try this.
```

```
plot(sin(x),-2*pi,2*pi)
```



```
#  11. Try plotting three or more functions at a time, all on the same domain, and each in a
#  different color. You can cut and paste from the entries above, but make sure you make the
#  necessary adjustments to get the same domain but different colors. (What happens if you
#  DON'T specify the same domain for all three functions?)
```

```
plot(x^2,-1,3,color='blue',axes_labels=['$x$','$y$'])+plot(3*x-2,-1,3,color='red') + plot(sin(x),-2*pi,2*pi), color='green')
```



```

```

evaluate