

Visible Python code

This is the first of two-part example. In this first part the Python code and tags are made visible using `\PySetup{action=show}` (this is the default). In the second part the Python code will be hidden using `\PySetup{action=hide}`. In both cases the Python output is identical (apart from the different tag names).

```
from sympy import *

x, y = symbols('x y')

ans = limit((sin(x)-x)/x**3, x, 0)           # py(ans.101,ans)
ans = (1/cos(x)).series(x, 0, 6)             # py(ans.102,ans)
ans = diff(cos(x**2)**2 / (1+x), x)          # py(ans.103,ans)
ans = integrate(x**2 * cos(x), x)            # py(ans.104,ans)
ans = integrate(x**2 * cos(x), (x, 0, pi/2)) # py(ans.105,ans)

f = Function('f')

eqn = Eq(Derivative(f(x),x,x) + 9*f(x), 1)    # py(ans.106,eqn)
sol = dsolve(eqn, f(x))                      # py(ans.107,sol)

eqn = Eq(f(x).diff(x, x) + 9*f(x), 1)         # py(ans.108,eqn)
sol = dsolve(eqn, f(x))                      # py(ans.109,sol)

my_int = Integral(cos(x), (x,0,2))            # py(ans.110,my_int)
my_ans = my_int.doit()                       # py(ans.111,my_ans)
```

Visible Python code output

Here is the output from the above Python code. This line of text is here for no other reason than to fill out the line so that we have two lines of text before the Python output. This makes it easier to compare the vertical spacing between this and the next example.

$$\begin{aligned}\text{ans.101} &:= -\frac{1}{6} \\ \text{ans.102} &:= 1 + \frac{x^2}{2} + \frac{5}{24}x^4 + O(x^6) \\ \text{ans.103} &:= -\frac{4x}{x+1} \sin(x^2) \cos(x^2) - \frac{\cos^2(x^2)}{(x+1)^2} \\ \text{ans.104} &:= x^2 \sin(x) + 2x \cos(x) - 2 \sin(x) \\ \text{ans.105} &:= -2 + \frac{\pi^2}{4} \\ \text{ans.106} &:= 9f(x) + \frac{d^2}{dx^2}f(x) = 1 \\ \text{ans.107} &:= f(x) = C_1 \sin(3x) + C_2 \cos(3x) + \frac{1}{9} \\ \text{ans.108} &:= 9f(x) + \frac{d^2}{dx^2}f(x) = 1 \\ \text{ans.109} &:= f(x) = C_1 \sin(3x) + C_2 \cos(3x) + \frac{1}{9} \\ \text{ans.110} &:= \int_0^2 \cos(x) dx \\ \text{ans.111} &:= \sin(2)\end{aligned}$$

Hidden Python code and output

In this example the Python code is hidden and consumes no vertical space on the page. Compare the gap between this pair of lines and the following output text against that seen on the previous page.

$$\text{ans.201} := -\frac{1}{6}$$

$$\text{ans.202} := 1 + \frac{x^2}{2} + \frac{5}{24}x^4 + O(x^6)$$

$$\text{ans.203} := -\frac{4x}{x+1} \sin(x^2) \cos(x^2) - \frac{\cos^2(x^2)}{(x+1)^2}$$

$$\text{ans.204} := x^2 \sin(x) + 2x \cos(x) - 2 \sin(x)$$

$$\text{ans.205} := -2 + \frac{\pi^2}{4}$$

$$\text{ans.206} := 9f(x) + \frac{d^2}{dx^2}f(x) = 1$$

$$\text{ans.207} := f(x) = C_1 \sin(3x) + C_2 \cos(3x) + \frac{1}{9}$$

$$\text{ans.208} := 9f(x) + \frac{d^2}{dx^2}f(x) = 1$$

$$\text{ans.209} := f(x) = C_1 \sin(3x) + C_2 \cos(3x) + \frac{1}{9}$$

$$\text{ans.210} := \int_0^2 \cos(x) dx$$

$$\text{ans.211} := \sin(2)$$

Visible Python markup

There is nothing special in this particular example. Its purpose is to provide a contrast with the following example where the tags have been hidden from view.

```
from sympy import *  
  
x = Symbol('x')  
  
ans = diff(cos(x)**2, x) # py(ans.301,ans)  
ans = integrate(2*sin(x)*exp(-x), x) # py(ans.302,ans)
```

$$\text{ans.301} := -2 \sin(x) \cos(x)$$
$$\text{ans.302} := -e^{-x} \sin(x) - e^{-x} \cos(x)$$

Hidden Python markup

This example creates the impression that there are no tags in the following block. But looks can be deceptive, the truth is revealed on the following page.

```
from sympy import *  
  
x = Symbol('x')  
  
ans = diff(cos(x)**2, x)  
ans = integrate(sin(x)*exp(-x), x)
```

$$\text{ans.401} := -2 \sin(x) \cos(x)$$
$$\text{ans.402} := -e^{-x} \sin(x) - e^{-x} \cos(x)$$

How to hide the tags

The easiest way to hide the tags from view is to include two blocks, one containing the tags but hidden from view using `\PySetup{action=hide}`, the other block is a copy of the first but with all tags stripped out. That second block is made visible using `\PySetup{action=verbatim}`. This is a bit cumbersome and prone to errors (the second block must be a faithful copy of the first block). Better solutions could be devised but they all appear to require significant changes to the current preprocessors – so much so that the this cloning trick was deemed a reasonable compromise.

Here is the LaTeX code from the previous example.

```
\PySetup{action=hide}

\begin{python}

    from sympy import *

    x = Symbol('x')

    ans = diff(cos(x)**2, x)                # py(ans.401,ans)
    ans = integrate(2*sin(x)*exp(-x), x)    # py(ans.402,ans)

\end{python}

\PySetup{action=verbatim}

\begin{python}

    from sympy import *

    x = Symbol('x')

    ans = diff(cos(x)**2, x)
    ans = integrate(sin(x)*exp(-x), x)

\end{python}

\begin{align*}
&\&\texttt{py}\{ans.401\}\&\&
&\&\texttt{py}\{ans.402\}
\end{align*}

\end{align*}
```