

# Randomizing Problem Order

*Common Sage Functions and Details*

This isn't what I would necessarily call convenient - it is more of a proof of concept and useful for when it is definitely necessary. Hopefully a more convenient/natural method will be available in future releases.

## The problem...

No matter how well randomized a specific problem is<sup>1</sup>, if the student does enough on a page, they already know what mechanics/techniques are involved for each problem - since the order of the problems don't change, just the way it was generated.

Luckily, since we can make strings in Sage, we can technically randomize the prompt along with the answer box. This allows us to randomize question order, along with the questions themselves.

## Example

Consider the problems below - note how the text matches correctly to the actual style of problem. Hit the "Try Another" button and see how they randomize, but move together properly. (Note, only 3 items, so you may need to hit the button a few times to get an actual different order).

**Problem 1** *Factor the following polynomial via grouping. Remember you must factor it completely.*

$$-5x^3 - 4x^2 + 80x + 64 = \boxed{-(5x + 4)(x + 4)(x - 4)}$$

**Problem 2** *Factor the following polynomial with the AC-method.*

$$-20x^2 - 20x = \boxed{-20(x + 1)x}$$

---

Learning outcomes:

<sup>1</sup>Ok, if it's randomized well *enough* this isn't quite true, but that is usually pretty tough

---

**Problem 3** Factor the following polynomial.

Note that its leading coefficient is one, so you should use the coefficient method.

$$x^2 - x - 12 = \boxed{(x + 3)(x - 4)}$$


---

## The Code

### How the code works

It's a bit tricky because we can't call sage code within the latex string formation. Nonetheless we can build a list of prompt/display/answer components, and then randomize the elements of that list. It's important to notice that calling "shuffle" on the list shuffles the list in place (it doesn't work to assign a new variable to be the shuffled result) but it only shuffles it on the outermost layer, so the elements stay in the [prompt,disp,answer] format, which makes calling them easier.

### The Code Itself

```
\begin{sagesilent}
#####Define default Sage variables.
#Default function variables
var('x,y,z,X,Y,Z,r,R')
#Default function names
var('f,g,h,dx,dy,dz,dh,df')

def RandInt(a,b):
    """ Returns a random integer in ['a','b']. Note that 'a' and 'b' should be integers
    """
    return QQ(randint(int(a),int(b)))
    # return choice(range(a,b+1))

def NonZeroInt(b,c, avoid = [0]):
    """ Returns a random integer in ['b','c'] which is not in 'av'.
        If 'av' is not specified, defaults to a non-zero integer.
    """
    while True:
        a = RandInt(b,c)
```

## Randomizing Problem Order

```
        if a not in avoid:
            return a

#### Problem p1
p1ans = (x-RandInt(-5,5))*(x-RandInt(-5,5))
p1disp = expand(p1ans)

#### Problem p2
p2ans = (NonZeroInt(-5,5)*x-RandInt(-5,5))*(NonZeroInt(-5,5)*x-RandInt(-5,5))
p2disp = expand(p2ans)

#### Problem p3
p3c1 = NonZeroInt(-5,5)
p3c2 = NonZeroInt(-5,5)
p3c3 = NonZeroInt(-5,5)
p3ans = (p3c1*x-p3c2)*(x-p3c3)*(x+p3c3)
p3disp = expand(p3ans)

p1prompt = LatexExpr(r'\text{Factor the following polynomial. } \\\
\text{Note that its leading coefficient is one, so you should use the coefficient method.}')

p2prompt = LatexExpr(r'\text{Factor the following polynomial with the AC-method. }')
p3prompt = LatexExpr(r'\text{Factor the following polynomial via grouping. } \\\
\text{Remember you must factor it completely. }')

problemVec = [[p1prompt,p1disp,p1ans],[p2prompt,p2disp,p2ans],[p3prompt,p3disp,p3ans]]
shuffle(problemVec)

\end{sagesilent}

\begin{problem}
    $\sage{problemVec[0][0]}$ \\\
    \[
        \sage{problemVec[0][1]} = \answer{\sage{problemVec[0][2]}}
    \]
\end{problem}

\begin{problem}
    $\sage{problemVec[1][0]}$ \\\
    \[
        \sage{problemVec[1][1]} = \answer{\sage{problemVec[1][2]}}
    \]
\end{problem}

\begin{problem}
    $\sage{problemVec[2][0]}$ \\\
    \[
        \sage{problemVec[2][1]} = \answer{\sage{problemVec[2][2]}}
```

`\]`  
`\end{problem}`