Problem Set 1 - Functions

p1: rightJustify.py
Write a function named right_justify() that takes a string named s as a parameter and prints the string with enough leading spaces so that the last letter of the string is in column 70 of the display:

```python
>>> right_justify('monty')
 monty
```

Hint: Use string concatenation and repetition. Also, Python provides a built-in function called `len(str)` that returns the length of a string, so the value of `len('monty')` is 5.

p2: doTwice.py
A function object is a value you can assign to a variable or pass as an argument. For example, do_twice() is a function that takes a function object as an argument and calls it twice:

```python
def do_twice(f):
    f()
    f()
```

Here's an example that uses do_twice() to call a function named print_spam() twice:

```python
def print_spam():
    print('spam')
do_twice(print_spam)
```

1. Type this example into a script and test it.
2. Modify do_twice() so that it takes two arguments, a function object and a value, and calls the function twice, passing the value as an argument.

p3: grid.py
Note: This problem should be done using only the statements and other features we have learned so far.

1. Write a function that draws a grid like the following:
Hint: to print more than one value on a line, you can print a comma-separated sequence of values:

```python
print('+', '-')
```

By default, print() advances to the next line, but you can override that behavior and put a space at the end, like this:

```python
print('+', end=' ') print('-')
```

The output of these statements is: `'+ -'`

A print statement with no argument ends the current line and goes to the next line.

2. Write a function that draws a similar grid with four rows and four columns.