Check-in 3: Fix unit-tested code

Once you’ve written tests for the `numberfmt` module, and used them to expose the three bugs,

Try fixing the code for `formatInt` so that all the tests pass.

1. For the `size=0` bug, it’s okay to require `size>0` as a precondition, and then test for the `AssertionError`.

2. For the leading-zero bug, the problem is that `str(0)` produces `"0"` rather than `"000"`, or `"48"` when it should be `"048"`. Instead of using `str()`, you can use Python format strings like this:

   ```
   >>> "%0*d" % (8, 415)
   '00000415'
   >>> "%0*d" % (5, 415)
   '00415'
   ```

   where the first number in the pair is the `size`.

3. The trickiest is probably the negative number bug. I suggest checking for a negative number explicitly, and either setting a boolean flag, or else a string to represent the `sign`.

   ```
   if num < 0:
      sign = "-"
      num = - num                  # Make it positive
   else:
      sign = ""
   ```

   Then you can proceed with doing the formatting for the positive number, and when done, stick the `sign` out front.

Before committing, you should create a file at the top of your repository called `.gitignore` (notice the leading dot), with the following lines:

```
__pycache__/
.coverage
htmlcov/
```

That will ensure that temporary object files and coverage reports do not get committed to the repository.