Bank Account Exercise (in class)

Due:

Push initial work before 13:00 today
Push final work before 22:00 today

Part 1: Code Checking Using pylint and flake8

1. Accept the Github Classroom Assignment. Link is in this week's Google Classroom material.
2. Clone it to your computer.
3. Run pylint and flake8 on the bank_account.py file. Observe the output and correct everything until you get a **perfect score** (10.0) from pylint and **no messages** from flake8.
4. Pylint does not like a class without any methods, so in the Transaction class define read-only properties for the transaction type and amount. That will provide the minimum 2 methods. You could write any public methods for this, such as __str__ and __eq__.

You *should* use complete words for names. pylint and flake8 won't complain about abbreviations of 3 or more characters, but it is good OO style to spell out words, not abbreviate.
So, use "BankAccount" instead of "BankAcct", "deposit(amount)" instead of "dep(amt)".

Part 2: Unit testing

Create a class named TestBankAccount (or BankAccountTest) in the file test_bank_account.py.

Write test methods with descriptive names, to test each of these test cases (one method for each test case):

1. *A new bank account has a balance of 0.*
2. *If you deposit a positive amount, then the balance is that amount.*
3. *If you deposit a negative or zero amount, the deposit method raises a ValueError.*
4. *After you deposit an amount you can withdraw part of it and see the balance is updated to equal the remaining amount (deposit - withdraw).*
5. *After you deposit an amount you can withdraw all of the amount and the remaining balance is 0.*
6. *If the balance is 0, you cannot withdraw anything.*

What to Submit

Push your revised **bank_account.py** and **test_bank_account.py** to Github.