

UnitTest

1. Покажите использование следующих видов проверок, изначально для каждой из них написав функцию, которая выполняет простые операции (см. пример ниже):

- `assertTrue()`, `assertFalse()`;
- `assertIn()`, `assertNotIn()`;
- `assertGreater()`;
- `assertLess()`;
- `assertCountEqual()`.

Пример: функция принимает на вход два числа a , b , выполняет $a-b$, нужно проверить, что получившееся значение меньше a (покажите также, с какими числами эта проверка не выполняется).

Для тех, кто забыл, как объявляется функция:

```
def subtract(a, b):  
    c = a - b  
    return c
```

2. Покажите использование подтестов на своём каком-либо примере или используйте следующий пример: сгенерируйте n случайных чисел, в цикле проверьте, что каждое из них больше либо равно 0.5.

Note: Only python 3.4+

3. Напишите функцию, которая на вход принимает строку и распознаёт в этой строке даты (например, '*я родился в 1990 году*' содержит дату, а '*мне было 3 года*' - не содержит) - если есть, возвращает `True`, иначе - `False`. Для простоты не используйте знаки препинания, на слова строку разделяйте по пробелу. Для определения дат придумайте несколько правил (например, если следующее слово - '*год*', значит это дата и т.п.). Затем создайте тесты для проверки случаев, перечисленных на второй странице (ради интереса - не смотрите тестовые случаи до того, как закончите писать функцию :))

Если какие-то тесты не прошли, постарайтесь дополнить/изменить свой алгоритм распознавания даты.

Тесты для проверки задания 3:

- 'с тех пор прошло 3 года'
- 'бюджет в 2018 году'
- 'в 6 веке'
- '5 марта'
- 'почти половина мужчин не доживают до 65 лет'
- 'в 1897 г'
- '1978 до нашей эры'
- 'в мае 2018 года'