Как переменную назовёшь, так она и...

Уважаемые коллеги, некоторых из вас я уже, наверное, замучил требованиями называть переменные какими-то, извините, "говорящими" названиями. Что это вообще значит?

Попробуем ~~использовать машиннное обучение~~ понять это на примерах.

- однобуквенные переменные, сравните два примера (чем одно из них лучше, чем другое? что вам не нравится в этих решениях?):

```
for i in x:
    z = 0
    k = 0
    for f in x:
        if f < i:
            z += 1
    for f in x:
        if f <= i:
            k += 1
    if k - 1 >= len(x) // 2 and z <= len(x) // 2:
        return i
```

```
bigger = 0
smaller = 0
for j in range(l):
    if arr[j] > arr[i]:
        bigger += 1
    if arr[j] < arr[i]:
        smaller += 1
if bigger <= l // 2 and smaller <= l // 2:
    print(arr[i])
    break
elif bigger <= l // 2:
    m_max = arr[i]
else:
    m_min = arr[i]
```

NB: к вопросу о том, что не нравится — прочитав пример выше спустя пару недель я сам удивился, зачем я сравниваю переменную `bigger` с нулём (целочисленно деля единицу на двойку)!

- имена логических переменных (мало сделать её логической, дайте нормальное имя, чтобы проверка не выглядела издёвкой!), сравните:

```
count = 0
for i in range(0, n):
    a = True
    j = i + 1
    while j < n and a:
        if x[i] == x[j]:
            a = False
        else:
            j += 1
    if a:
        count += 1
```

```
flag = 1
num = 1
for i in range(1, len(x)):
    flag = 1
    for idx in range(i - 1, - 1, - 1):
        if x[idx] == x[i]:
            flag = 0
            break
    if flag == 1:
        num += 1
```

Да, переменная `flag` выглядит лучше, чем `a`, но это переименование не решает основную задачу — сделать текст программы понятнее.
Слово `flag` (варианты: `indicator`, `check`) подчёркивает то, что переменная работает своего рода "переключателем", т.е. является логической. Но это и так видно из присваивания вида `flag = False` ики `flag = True`.
При этом ни одно имя никак не поясняет, сигналом (проверкой) **ЧЕГО** является эта переменная. А ведь именно это является основной целью разумного именования функций и переменных.
А теперь вот такой вариант. Раз уж совершенство, то поменяем внутренний цикл, который теперь выглядит не так устрашающе. Нам в этом случае всё равно как искать — слева направо или справа налево.

```
num = 1
for i in range(1, len(x)):
    seenBefore = False
    for idx in range(i):
        if x[idx] == x[i]:
            seenBefore = True
            break
    if not seenBefore:
        num += 1
```

Непонятно, зачем это всё нужно, пока вы пишете программы длиной 10-20 строк. Такую программу, как правило, легко целиком "удержать" в голове и помнить смысл каждой переменной.
Но как только размер программы станет хотя бы 100+ строк, помнить смысл всех переменных вроде `flag`, `flag_2`, `check_it`, `indicator` будет непросто, если вообще возможно.
Кстати, чтобы не гадать, какие имена не стоит выбирать в качестве имён переменных и функций, чтобы они не совпали с именами встроенных в `Python` функций, можно посмотреть вот сюда.