ШКОЛА ОБЛАЧНЫХ ТЕХНОЛОГИЙ

Правила хорошего кода в Python.

PEP

Python, подобно живому организму, развивается и приобретает новые возможности благодаря многочисленному международному сообществу согласно определенным правилам и стандартам PEP. PEP — Python Enhancement Proposal, предложения по развитию Python. Эти стандарты позволяют создавать унифицированную проектную документацию для новых утвержденных возможностей языка Python. Самый известный PEP имеет восьмой порядковый номер. PEP8 содержит перечень принципов написания красивого и лаконичного программного кода на языке Python.

import

Согласно правилам PEP 8, все import’ы нужно прописывать в начале файла.
Если import’ов много, то каждый пишется на новой строке.

```
import random

random = random.randint(1, 9)
print(random)
```

Отступы

Перед некоторыми «ветками» кода для лёгкой читаемости кода нужно ставить отступ

PEP 8: E302 expected 2 blank lines

PEP 8: E305 expected 2 blank lines after class or function definition

PEP 8: W292 no newline at end of file

```
import random


def random():
    rand = random.randint(1, 2)
    if rand == 1:
        print("Выпало 1")
    else:
        print("Выпало 2")


random()
```

Функция main()

Функция main() предназначена для описания логики программы.

```
#code…


def main():
    rand = random()
    say(rand)


main()
```

Конструкция if _name_ == “_main_”

Конструкция
if __name__ == “__main__”
предназначена для разделение кода, который будет выполнятся при вызове кода как модуля (при импортировании его в другой скрипт) – и при запуске самого модуля, как отдельного файла.

```
def main():
    # Какой-то код


if __name__ ==
"__main__":
    main()
```

Максимальная длина строки и разрыв строки

PEP8 предлагает ограничить длину строки 79 символами. Это рекомендуется делать, чтобы вы имели возможность открывать несколько файлов рядом друг с другом, а также избегать переноса строк.

```
def main():
    # Какой-то код


if __name__ ==
"__main__":
    main()
```

Комментарии

~~(Если реализацию трудно объяснить, это была плохая идея)~~

Используйте комментарии для документирования кода в том виде, в каком он написан. Это важно для ваших коллег и вашего понимания своего кода в будущем. Вот три важных ключевых момента, которые необходимо учитывать, при добавлении комментариев к коду:

1) Используйте длину комментариев при документации не более 72 символов;
2) Не используйте сокращения, начинайте предложения с заглавной буквы;
3) Не забывайте актуализировать комментарии, при изменении кода.

Пример простейшего комментария:

```
name = "Иван Иванович"  # Имя и Фамилия
```

Пробелы в выражениях и утверждениях

Полезность пробелов в выражениях и операторах трудно переоценить. Если пробелов недостаточно, код может быть трудночитаемым, так как все сгруппированы вместе. Если пробелов слишком много, может быть сложно визуально объединить строки кода в, логически связанные, блоки.

Окружите следующие бинарные операторы одним пробелом с каждой стороны:
1) Операторы присвоения (=, +=, -= и т. п.)
2) Сравнения (==, ! =, >, <. ≥, ≤) и (is, is not, in, not in)
3) Логические (and, or, not)

Когда = используется для присвоения значения для аргумента функции, не окружайте его пробелами.

Когда лучше проигнорировать PEP8?

Если вы безукоризненно исполняете все предписания PEP8, можно с уверенностью гарантировать «чистоту», высокий уровень читаемости кода и профессионализм программиста. Что принесет пользу всем взаимодействующим с вашим кодом, от коллег до конечного заказчика продукта. Но все же некоторые рекомендации PEP8 неприменимы в следующих случаях:

1) Если соблюдение PEP8 нарушит совместимость с существующим программным обеспечением;
2) Если код, сопутствующий тому, над чем вы работаете, несовместим с PEP8;
3) Если код нужно оставить совместимым с неактуальными версиями Python.

Задания для выполнения

Все задания выполнить в разных модулях и подключить к главному файлу. Одно из заданий нужно полностью прокомментировать.

1) Есть список a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]. Выведите все элементы, которые меньше 5.

2) Даны списки: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]; b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Нужно вернуть список, который состоит из элементов, общих для этих двух списков.

3) Найдите три ключа с самыми высокими значениями в словаре my_dict = {'a':500, 'b':5874, 'c': 560,'d':400, 'e':5874, 'f': 20}.

4) Сделайте так, чтобы число секунд отображалось в виде дни:часы:минуты:секунды.

Домашняя работа

• Повторить все предыдущие лекции.

• Напишите проверку на то, является ли строка палиндромом. Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево.