

Правила оформления исходного кода на языке Python

Правила именования

Для именования переменных используется нотация `snake_case`

```
Python
# Примеры объявления переменной
result = 100 + 500
max_value = 100
```

Для именования функций используется `snake_case`

```
Python
# Примеры объявления функции
def calculate(a, b):
    return a + b

def calculate_sum(a, b):
    return a + b
```

Для именования классов используется `UpperCamelCase`

```
Python
class MyClass:
    def __init__(self, value):
        self.value = value
```

Для именования констант используется `UPPER_SNAKE_CASE`

```
Python
WIDTH = 1920
WINDOW_HEIGHT = 1080
```

Пробелы и отступы

Пробелы ставятся вокруг операторов присваивания, арифметических операторов, логических операторов и операторов присваивания

Пробелы ставятся после запятых

```
Python
# Пробелы вокруг операторов
result = a + b * c
my_list = [1, 2, 3]

def my_function(arg1, arg2): # Правильно ✓
def my_function( arg1, arg2 ): # Неправильно ✗
```

Для каждого уровня отступа используется 4 пробела. Использование табуляции вместо пробелов недопустимо

```
Python
# Пример уровней отступа
def my_function():
    if condition: # 4 пробела
        print("Условие выполнено") # Ещё 4 пробела (вложенность)
```

Стараться избегать лишних пробелов

Переменные

Переменная должна сохранять свой тип на протяжении всей программы. Например, если переменная была строкой, она должна оставаться строкой, а не превращаться в число или список

Python

```
# Правильно ✓
initial_value = "10"           # Строка
numeric_value = int(initial_value) # Новое число
list_value = [numeric_value]   # Новый список

# Неправильно ✗
value = "10"                   # Строка
value = int(value)             # Теперь это число
value = [value]                # Теперь это список
```

Имя переменной должно отражать её назначение. Если переменная называется count, она должна хранить количество чего-либо

Python

```
# Правильно ✓
count = 0                       # Количество
count_label = "Ноль"           # Строка для отображения

# Неправильно ✗
count = 0                       # Количество
count = "Ноль"                 # Теперь это строка
```

Переменная не должна использоваться для хранения разных сущностей в разных частях программы

Python

```
# Правильно ✓
user_data = get_user_data()     # Данные пользователя
processed_data = process_data(user_data) # Обработанные данные

# Неправильно ✗
data = get_user_data()         # Данные пользователя
data = process_data(data)     # Теперь это обработанные данные
```

