Task 3

Горелова Анастасия

September 2021

Ниже будут описаны некоторые особенности синтаксиса языка Python, про которые я раньше не знала или знала, но не в полной мере, и которые показались мне наиболее интересными :)

Pattern matching

Во многих языках программирования присутствует конструкция switch/case, которая используется для удобного и красивого разбора вариантов значения переданной переменной. Так называемыя конструкция pattern matching. Удивительно, но в питоне она появилась, только начиная с Python 3.10. И выглядит она следующим образом:

```
match expression:
    case pattern_1:
        ...
    case pattern_2:
        ...
```

Как же люди имитировали поведение switch/case до Python 3.10? Есть несколько способов. Самый простой выглядит следующим образом:

```
colour = "blue"
if colour == "red":
    print("The colour is red")
elif colour == "blue":
    print("The colour is blue")
elif colour in ("black", "grey"):
    print("The colour is dark")
else:
    print("The colour is unknown")
```

Однако есть более красивый и "питонический" способ реализовать то же самое, который я раньше не использовала. Для него нам понадобятся словари - неупорядоченные коллекции произвольных объектов с доступом по ключу.

```
colours = {
    "red": "The colour is red",
    "blue": "The colour is blue",
    "black": "The colour is dark",
    "grey": "The colour is dark"
}
mycolour = "blue"
print(colours.get(mycolour, "The colour is unknown"))
```

Ссылка на спецификацию: https://docs-python.ru/tutorial

Присваивание с распаковкой

Следующая конструкция носит интересное название - "Присваивание с распаковкой". Ее суть в том, что в левой части выражения мы перечисляем переменные, которым хотим присвоить значения, а в правой части - итерируемый объект. Важное условие - он должен содержать ровно столько элементов, сколько находится в левой части. Когда конструкция раскрывается, происходит попарное присваивание переменных.
Так, мы, например, можем сделать циклический сдвиг значений переменных:

```
a = 1
b = 2
c = 3
a, b, c = c, a, b
print(a, b, c)  # 3 1 2
```

Более сложный пример:

```python
array = (2, 1, 3, "blue")
a, b, c, colour = array
print(colour, a, b, c)  # blue 2 1 3
```

Ссылка на спецификацию: https://docs-python.ru/tutorial