

# Домашнее задание по лекции 3 - Классы и декораторы



## задача 1

Даны объявления в формате JSON

**title** - обязательное поле

В объявлении могут присутствовать различные поля

**Пример** объявления с атрибутом 'ближайшие станции метро' (**metro\_stations**):

```
{
  "title": "iPhone X",
  "price": 100,
  "location": {
    "address": "город Самара, улица Мориса Тореза, 50",
    "metro_stations": ["Спортивная", "Гагаринская"]
  }
}
```

**Пример** объявления с атрибутом 'категория' (**class**):

```
{
  "title": "Вельш-корги",
  "price": 1000,
  "class": "dogs",
  "location": {
    "address": "сельское поселение Ельдигинское, поселок санатория  
Тишково, 25"
  }
}
```



## Напишите класс Advert , который:

1. динамически создает атрибуты экземпляра класса из атрибутов JSON-объекта
  - о не нужно фиксировать атрибуты в классе пример ниже **НЕВЕРНЫЙ**

```
# НЕВЕРНЫЙ ПРИМЕР! Создавайте атрибуты динамически
class Advert:
    def init (self, mapping):
        self.title = mapping['title']
        self.price = mapping['price']
        ...
```

- о обращаться к атрибутам можно через точку

```
# создаем экземпляр класса Advert из JSON
lesson_str = """{
    "title": "python", "price": 0,
    "location": {
        "address": "город Москва, Лесная, 7",
        "metro_stations": ["Белорусская"]
    }
}"""
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)

# обращаемся к атрибуту location.address
lesson_ad.location.address

# Out: 'город Москва, Лесная, 7'
```

**подсказка-01:** создайте отдельный класс, который будет преобразовывать JSON-объекты в python-объекты с доступом к атрибутам через точку и используйте его для разбора полей объявления и вложенного поля `location`

**подсказка-02:** в python есть функция, которая проверяет, является ли строка ключевым словом

## 2. имеет свойство price

- проверяет, что устанавливаемое значение не отрицательное

```
lesson_str = '{"title": "python", "price": -1}'
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)

# Out: ValueError: must be >= 0
```

- в случае отсутствия поля price в JSON-объекте возвращает 0

```
lesson_str = '{"title": "python"}'
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson) lesson_ad.price

# Out: 0
```

---

## задача 2

---

Добавим к классу Advert метод `__repr__`, который выводит название и цену объявления

```
class Advert:
    def repr(self):
        return f'{self.title} | {self.price} ₺'

iphone_ad = Advert('iPhone X', 100) print(iphone_ad)

# Out: iPhone X | 100 ₺
```

Напишите миксин `ColorizeMixin`, который:

- меняет цвет текста при выводе на консоль
- задает цвет в атрибуте класса `repr_color_code`

```
class Advert(ColorizeMixin):
    repr_color_code = 32 # green

    def repr (self):
        return f'{self.title} | {self.price} ₺'
```

- используйте миксин для изменения цвета вывода Advert. repr

**подсказка-01:** про изменение цвета можно почитать тут - [Add Colour to Text in Python](#)

---

## DoD (Definition of Done) - критерии, позволяющие понять, что задача сделана, как ожидается:

- написан класс, экземпляры которого позволяют обращаться к полям через точку: `iphone_ad.price`
- класс Advert не содержит атрибуты при объявлении.  
Исключение: реализация свойства `price`
- экземпляры класса Advert инициализируются из словаря
- поле `Advert.price` выбрасывает исключение при установке отрицательного значения
- выводится адрес при обращении к атрибуту через точки: `iphone_ad.location.address`
- выводит категорию при обращении через точку, например, `corgi.class_`
- при выводе объявления в консоли `print(corgi)` (в примере с вельш-корги) получаем надпись 'Вельш-корги | 1000 ₺' желтым цветом
- нет замечаний от flake8