

Python



План

- Практика
- Синтаксический сахар
- Принципы проектирования

Разберем прошлое задание

И git

Синтаксический
сахар

Списковые выражения (list comprehensions)

- Позволяют создавать списки "в одну строку"
- Позволяют создавать не только списки

Практика

Файл "comprehensions.py"

lambda

- Анонимные функции, позволяют создавать функцию только для одного раза
- Могут занимать только одну строку
- Если сомневаетесь, используйте `def`

Практика

Файл "lambdas.py"

Декораторы

Декораторы - очень простые

- Декоратор - такая функция
- Которая принимает функцию
- И возвращает функцию

Практика

Файл "decorators/*.py"

Полезные концепции из функционального программирования

- `map` - выполняет функцию для каждого объекта из массива входных параметров
- `filter` - оставляет из массива только что удовлетворяет условию
- `reduce` - выполняет функцию для каждого объекта из массива входных параметров, оперируя остатком

Практика

Файл "functions/fr.py"

Принципы проектирования

Для всего кода

Принципы проектирования ПО

- DRY - Don't repeat yourself ([principles/dry.py](https://principles.dry.py))
- KISS - Keep it short and simple (https://github.com/sobolevn/python-code-disasters/blob/master/python/create_objects.py)
- YAGNI - You Ain't Gonna Need It. Не создавайте что-либо, пока оно не понадобится.
- SOLID - фундаментальный принцип построения ПО

SOLID

- Single responsibility principle: На каждый класс должна быть возложена одна-единственная обязанность.
- Open/closed principle: Программные сущности должны быть открыты для расширения, но закрыты для изменения.
- Liskov substitution principle: Функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом.
- Interface segregation principle: Много специализированных интерфейсов лучше, чем один универсальный.
- Dependency inversion principle: Зависимости внутри системы строятся на основе абстракций. Модули верхнего уровня не зависят от модулей нижнего уровня. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.

Но Python отличается от
обычных ООП языков

<https://github.com/faif/python-patterns>

Шаблоны проектирования
родились, когда разработчики
решали свои проблемы

<https://github.com/kamranahmedse/design-patterns-for-humans>

Шаблоны проектирования

Для Flask