




Функциональное программирование



- Ленивое вычисление — это стратегия, реализованная в некоторых языках программирования, при которой определенные объекты не создаются до тех пор, пока они не понадобятся ; эта стратегия часто используется в сочетании с функциями, которые создают коллекции объектов.
- [itertools — Functions creating iterators for efficient looping — Python 3.13.0 documentation](#)




Императивное программирование Декларативное программирование

- Императивное программирование: структурное, ООП.
 - Концентрация на последовательности действий, чтобы получить результат
- Декларативное программирование: функциональное
 - Фиксация результата, без определения алгоритма



Свойства

- Функции - объекты первого класса. Работа с функциями как с данными: присваивание переменным, передавать в качестве аргументов
 - Нет состояний- нет переменных – нет операции присваивания
 - Рекурсия – основной подход , не циклы
 - Есть функции высшего порядка – функции, которые в рамках аргументов принимают другие функции
 - Функция чистая – нет побочных эффектов
 - Концентрация на результате, не на методе
- 



Функции как объекты первого класса

- Пример



Рекурсия

- Вызов функции самой себя

Функции высшего порядка

- Принимают в качестве аргументов иные функции
- Мар: принимает функцию и список, применяет функцию к каждому элементу списка и возвращает новый модифицированный список.



lambda

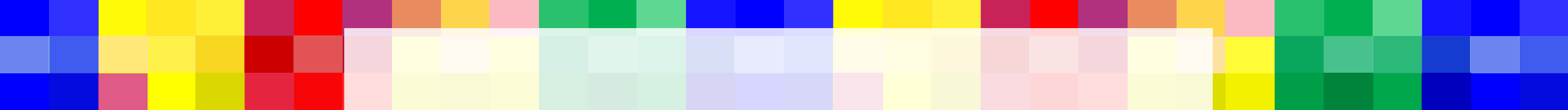
- lambda-функции (в программировании): lambda-функция – это безымянная функция с произвольным числом аргументов, вычисляющая одно выражение

Замыкание, каррирование

- Замыкание – в теле присутствуют ссылки на переменные, объявленные вне тела этой функции
- Каррирование – преобразование функции многих переменных в набор функций, каждая из которых является функцией одной переменной
- Частичное применение: возвращение функции от меньшего количества аргументов

- 
- Не равны как объекты, но результат гарантированно одинаков на одном наборе данных



- 
- Применил функцию – потом применяешь аргументы (нормальный подход)
 - Вычислил аргументы – вычисляешь функцию (аппликативный подход, большинство языков программирования)

Итераторы

- Навигация по объектам
- Объект-перечислитель
- Если вы проектируете класс, объекты которого будут итераторами, то в него необходимо добавить методы `__iter__()`, который возвращает `self`, и `__next__()`, который возвращает какие-либо значения.
- Генератор – это функция, которая будучи вызванной в функции `next()` возвращает следующий объект согласно алгоритму ее работы. Вместо ключевого слова `return` в генераторе используется `yield`.
- при вызове `yield` функция не прекращает свою работу, а “замораживается” до очередной итерации, запускаемой функцией `next()`.

Ленивые вычисления

- *map(), filter(), range(), zip()* возвращают итератор
- Не возвращают наборы вычислений над всем списком, выдача результата по необходимости