Языки программирования:

Низкоуровневые – они представляют собой символьную интерпретацию машинного кода (Асемблер)

Высокоуровневый – используются понятные человеку слова(англ) функции (Python)

Программа переводчик – транслятор.



Переводят исходный код программы на высокоуровневом языке в машинный код, понятный процессору.

Транслятор 2 вида:

* Компилятор – сначала анализирует весь код, затем создает исполняемый файл, после запускает. Если есть ошибки файл не будет создан.
* Интерпретатор – исполняет программу сразу построчно. Если натыкается на ошибку останавливается. (Python)

Также языки программирования классифицируются по **парадигмам**. Парадигма программирования — это совокупность идей и понятий, определяющих стиль написания компьютерных программ. Выделяют следующие парадигмы программирования:

* Императивное программирование
* Процедурное программирование
* Объектно-ориентированное программирование
* Декларативное программирование
* Функциональное программирование
* Логическое программирование

Для **императивного программирования** характерна запись программы в виде последовательно выполняемых инструкций (команд). К нему обычно относят процедурные и объектно-ориентированные языки. В **процедурном программировании** программа состоит из шагов, которые могут объединяться в более крупные структурные единицы - подпрограммы. К таким языкам относятся Basic, C, Pascal. В **объектно-ориентированной** парадигме программы представляются в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы могут наследовать друг друга. Считается, что объектно-ориентированное программирование наиболее абстрактное и приближенное к мышлению человека.

В **декларативном программировании** задаётся спецификация решения задачи, то есть описывается, что представляют собой проблема и ожидаемый результат. К подвидам декларативного программирования также зачастую относят **функциональное** (процесс вычисления трактуется как вычисление значений функций в математическом понимании) и **логическое программирование**(основано на теории и аппарате математической логики с использованием математических принципов резолюций).

Языки можно разделить на 2 группы:

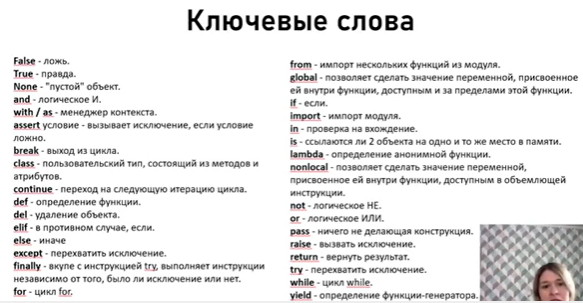
* Императивные (программа состоит из команд/процедур/объектов)
* Декларативные (программа состоит из описаний)

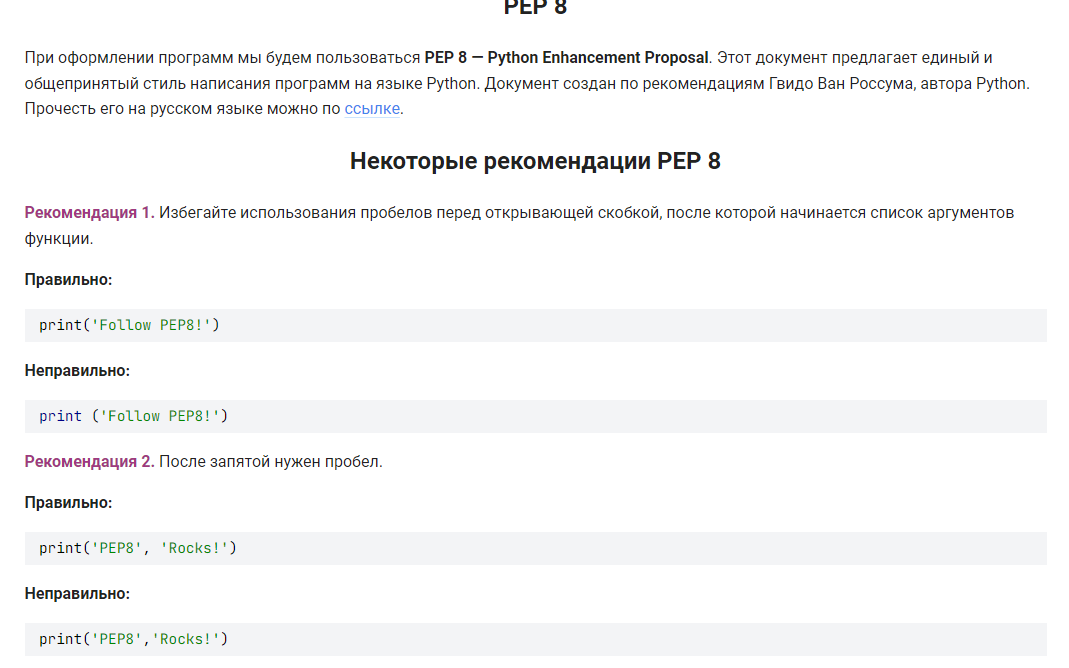
Императивные делятся на:

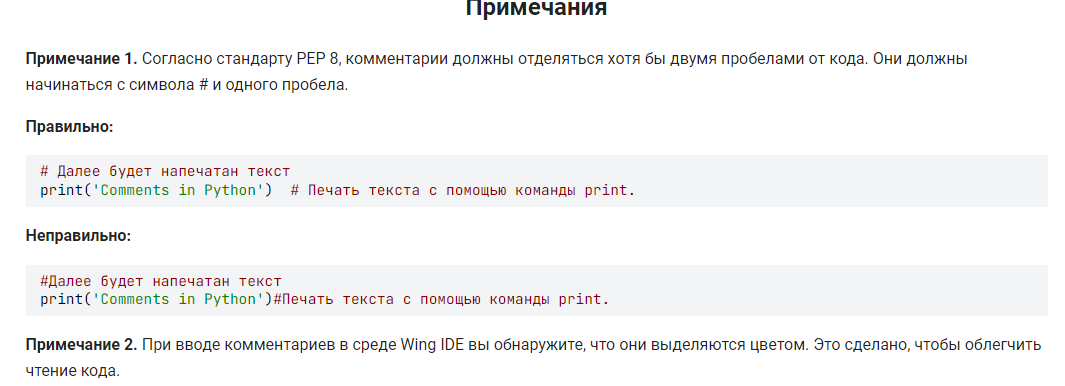
* Процедурные (Паскаль, С)
* Объектно-ориентированные (С++, Java, Python)

ООП (Объектно-ориентированное программирование)

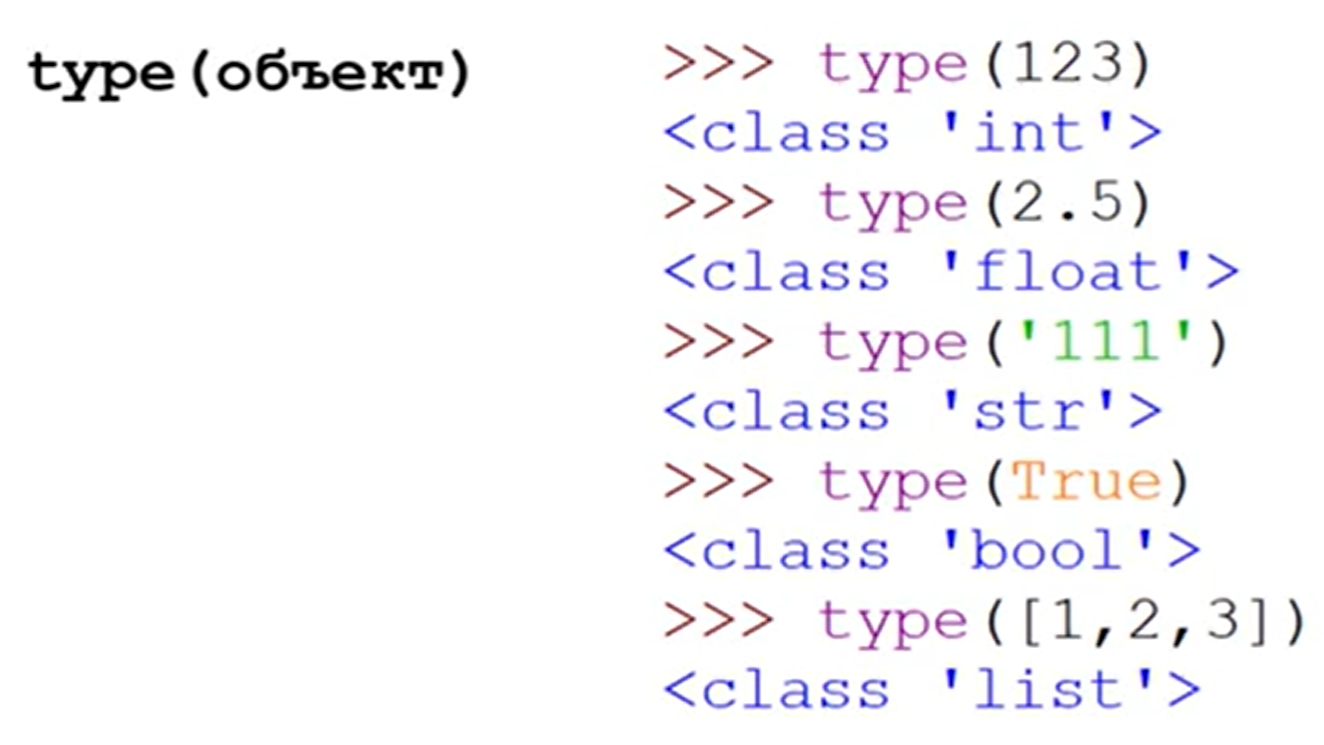
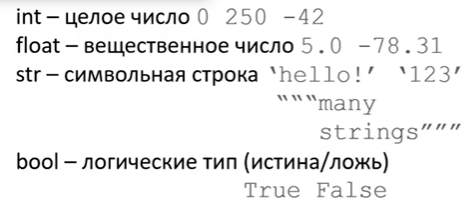
* Все состоит из объектов
* Объект принадлежит определенному классу
* Классы могут наследовать друг друга
* У объекта есть свойства (характеристики) и методы(действия)
* Объекты можно создавать и удалять



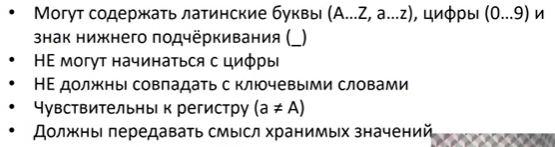
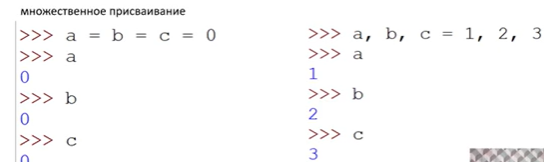


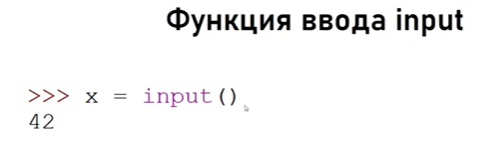


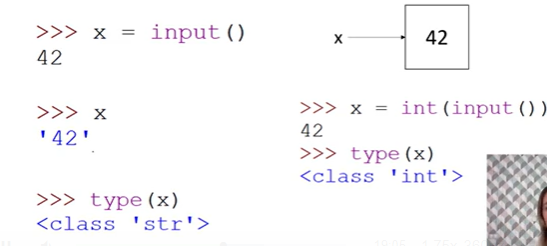
Типы данных. Переменные ввод-вывод.

Типы данных 

С помощью функции **type** мы можем определить, к какому типу относится то или иное значение. Результатом выполнения функции будет строка, начинающаяся со слова class и затем название типа данных.

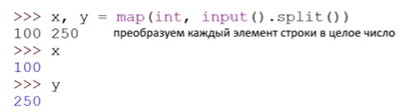
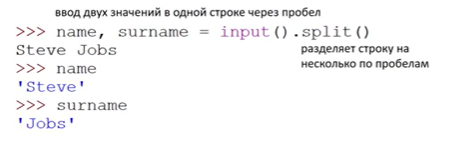


Функция input всегда имеет строковое значение.

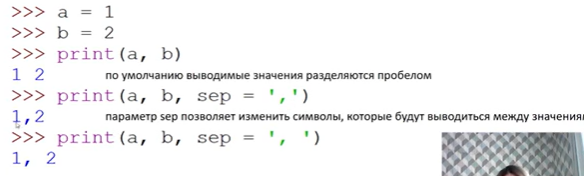
 Преобразование значение при помощи функции type

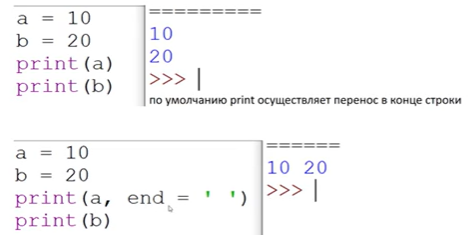
Перенос на новую строку \n

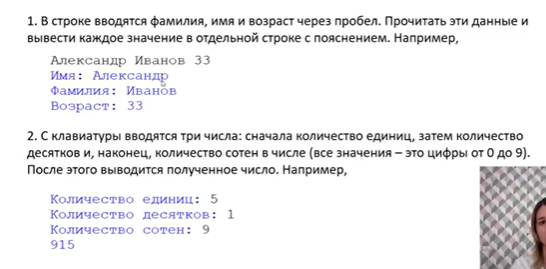
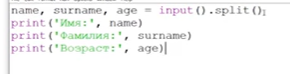
Часто несколько значений вводятся в одной строке и разделяются пробелом. Для того, чтобы присвоить каждое значение отдельной переменной, воспользуйтесь методом **split**. Он вызывается через точку после вызова функции input. Слева от оператора присваивания нужно перечислить через запятую имена переменных в том порядке, в котором необходимо присвоить прочитанные значения.

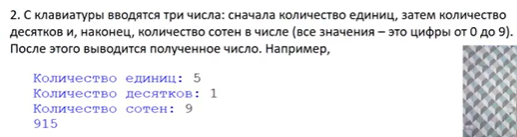


Если значения имеют числовой тип, то для преобразования нам понадобится функция **map**. Первым её параметром мы укажем, в какой тип нужно преобразовать каждое из считанных значений, а вторым параметром – саму считанную строку, разделенную по пробелам.



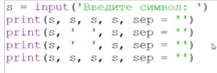


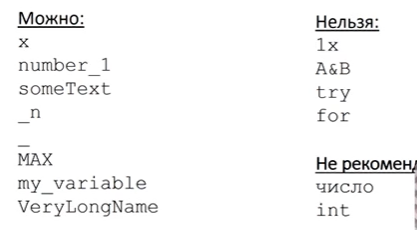
Задача: Решение: 

Задача:  Решение: 

Задача:  Решение: 

Задача: 

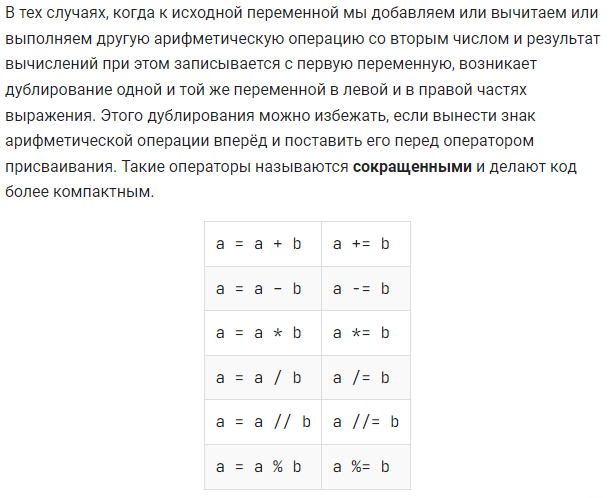
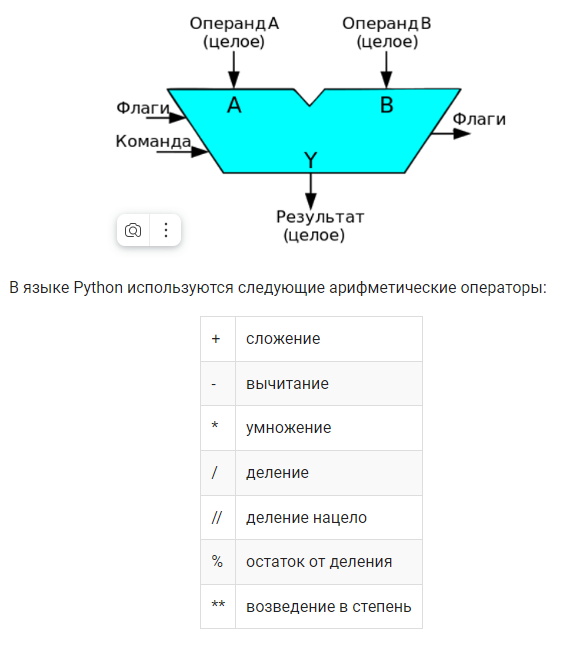
Решение: 

Имена переменных: 

Арифметические операторы

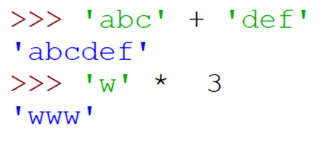
Выполнение арифметических действий – это основное назначение процессора компьютера. Изначально компьютер предназначен именно для выполнения вычислений. И даже если вы редактируете изображение на мониторе, для процессора это превращается в операции сложения, вычитания, умножения или деления, применяемые к двоичным числам.

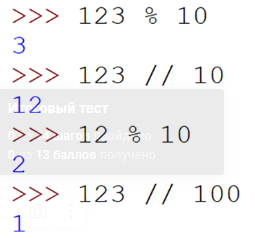
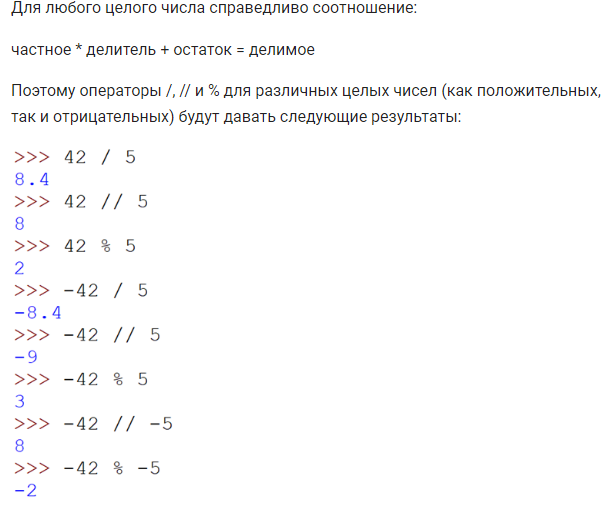
Основной вычислительный блок процессора – это АЛУ, арифметико-логическое устройство. Оно принимает на вход до двух значений операндов, а также код команды, которую необходимо выполнить. После выполнения команды, мы получаем на выходе результат вычислений.



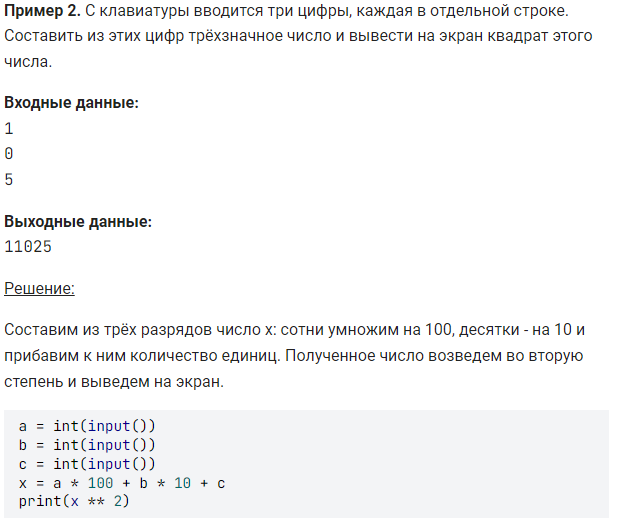
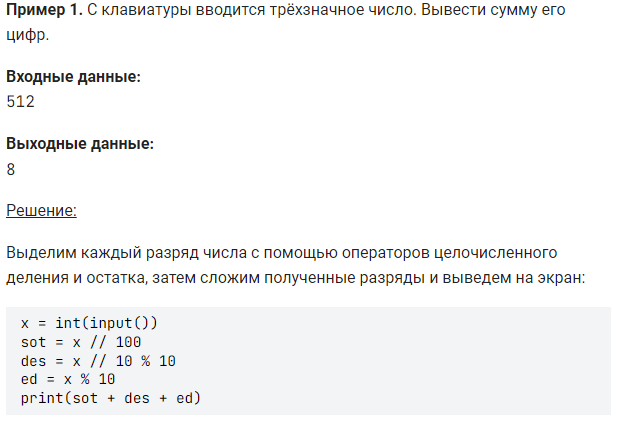
Арифметические операции имеют следующий приоритет выполнения:

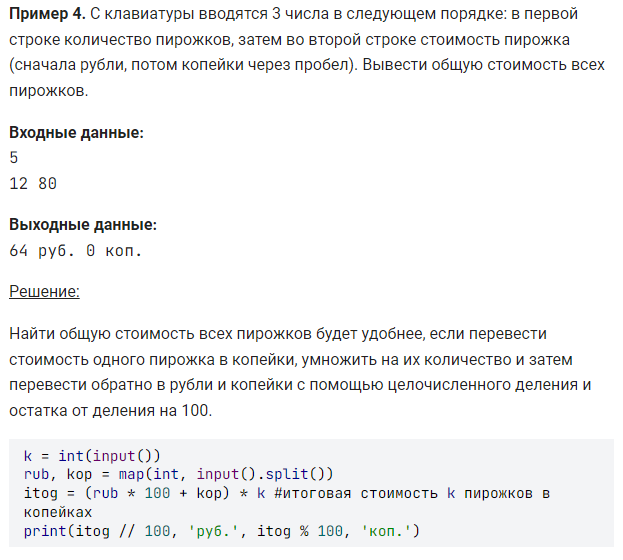
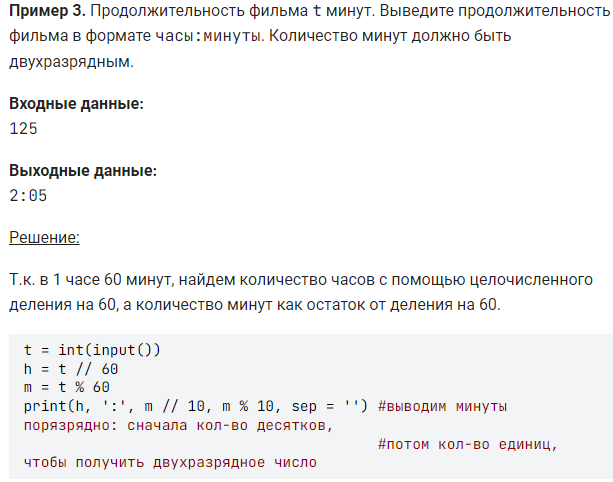
1. Скобки
2. Степень
3. Унарный минус
4. Умножение, деление
5. Сложение, вычитание

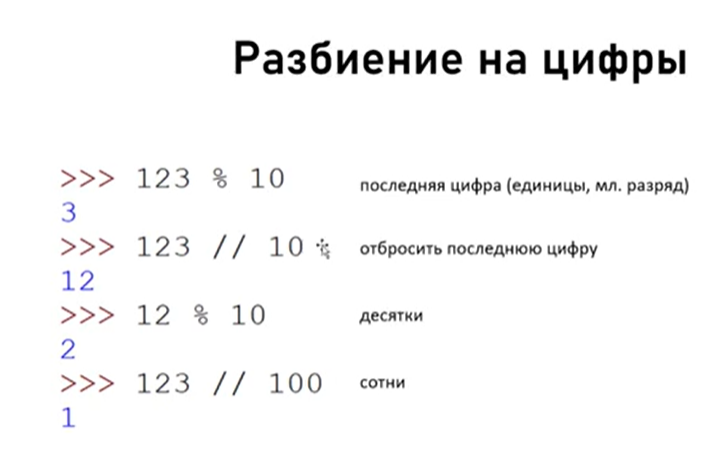
Несмотря на то, что арифметические операторы в основном предназначены для работы с числами, некоторые из них можно использовать и со строками. Оператор сложения выполняет конкатенацию строк, то есть соединяет их в одну строчку. Порядок сложения строк имеет значение, если записать слагаемые наоборот, то и результат получится другой. Также строки можно умножать на целые числа. При умножении на положительное число происходит дублирование строки указанное количество раз. При умножении на 0 или на отрицательное число, всегда получаем пустую строку. 



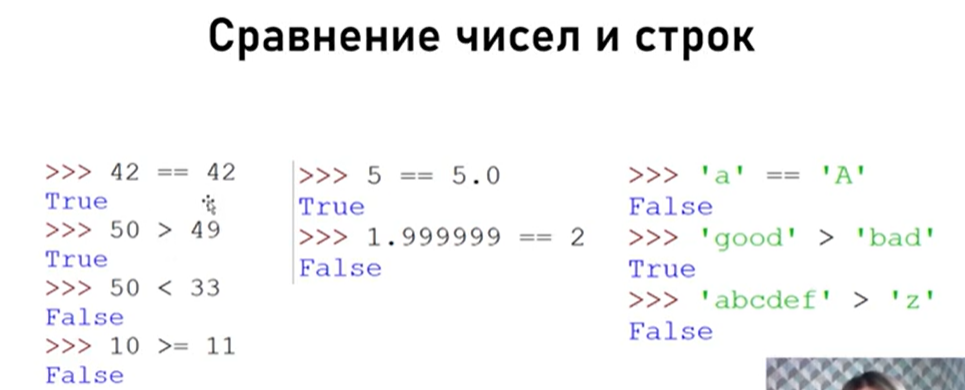
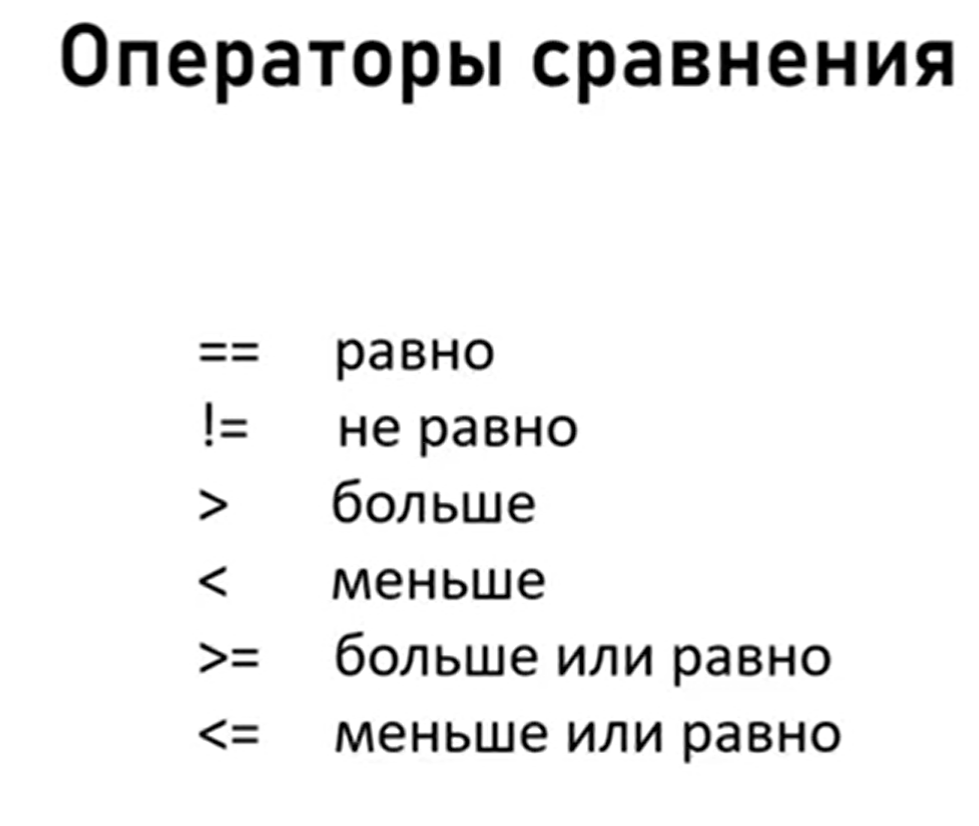
Одним из основных применений целочисленного деления в алгоритмах можно назвать выделение разрядов числа. Например, для любого десятичного числа справедливо утверждение, что остаток от деления на 10 – это его младший разряд, а частное – это все цифры без младшего разряда. Последовательное применение операций деления и деления с остатком позволит выделить все разряды из числа.

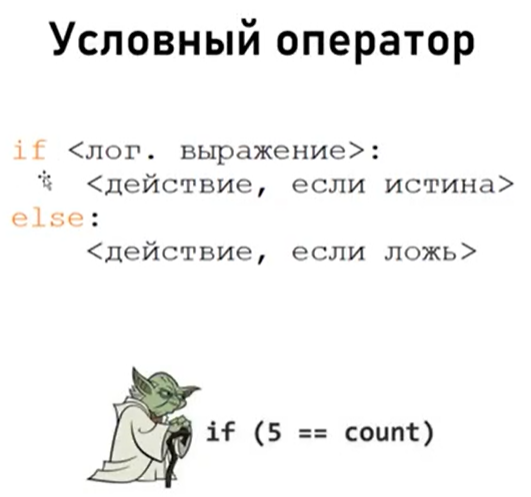


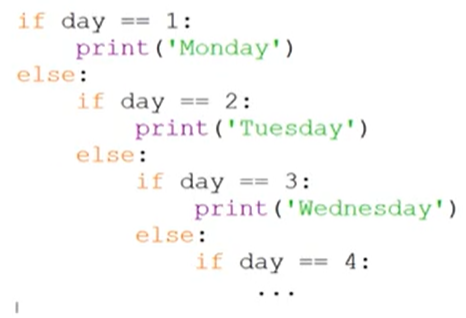
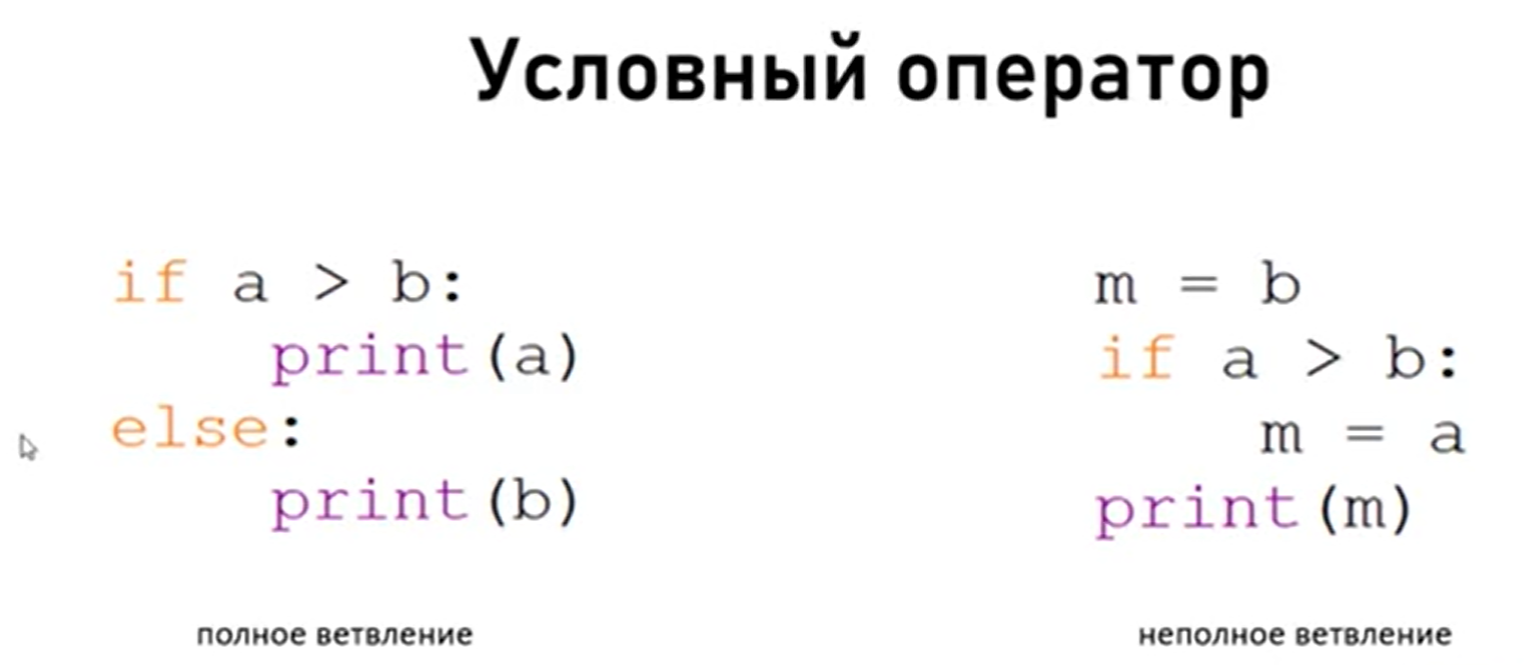


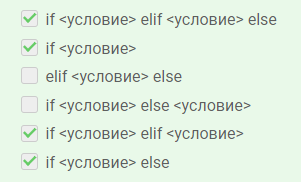
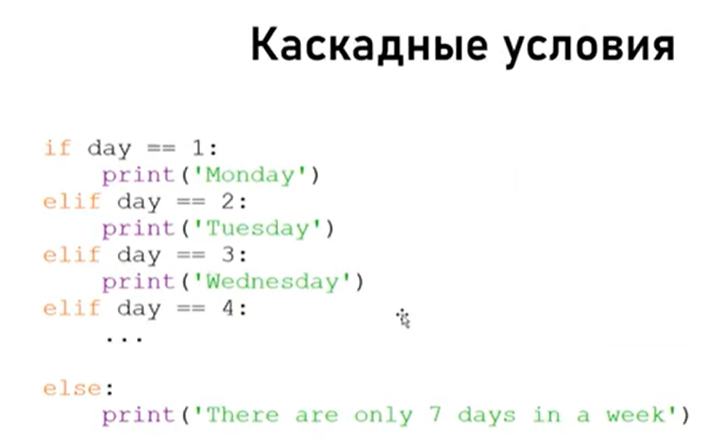


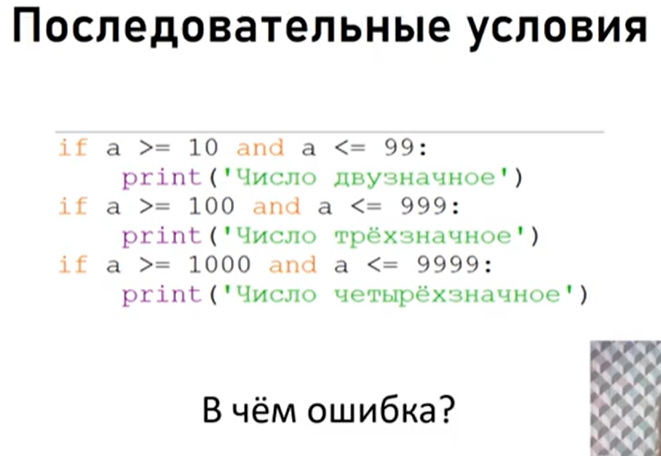
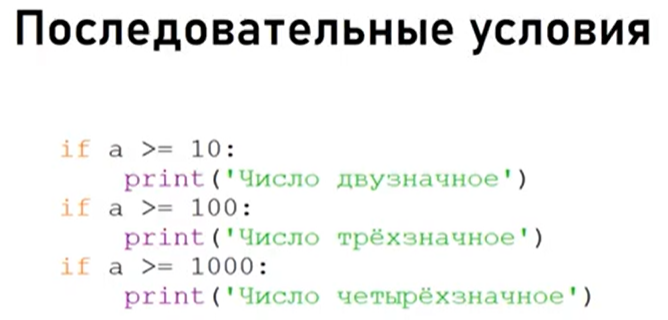
Условия

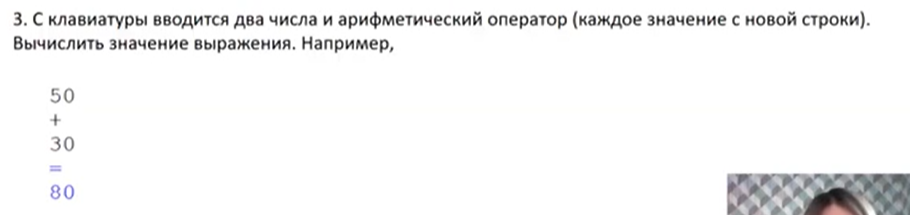
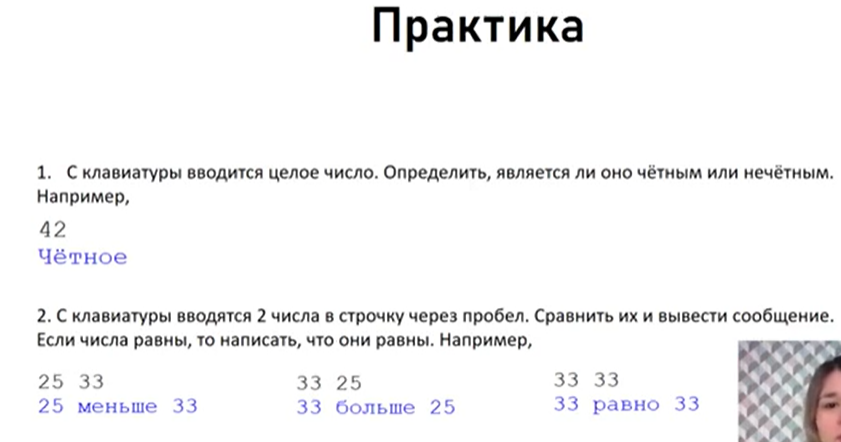




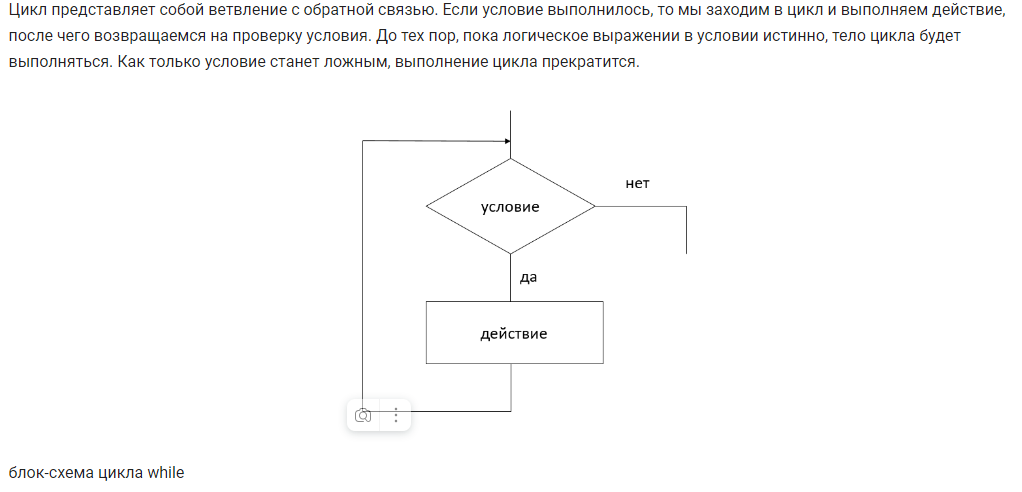
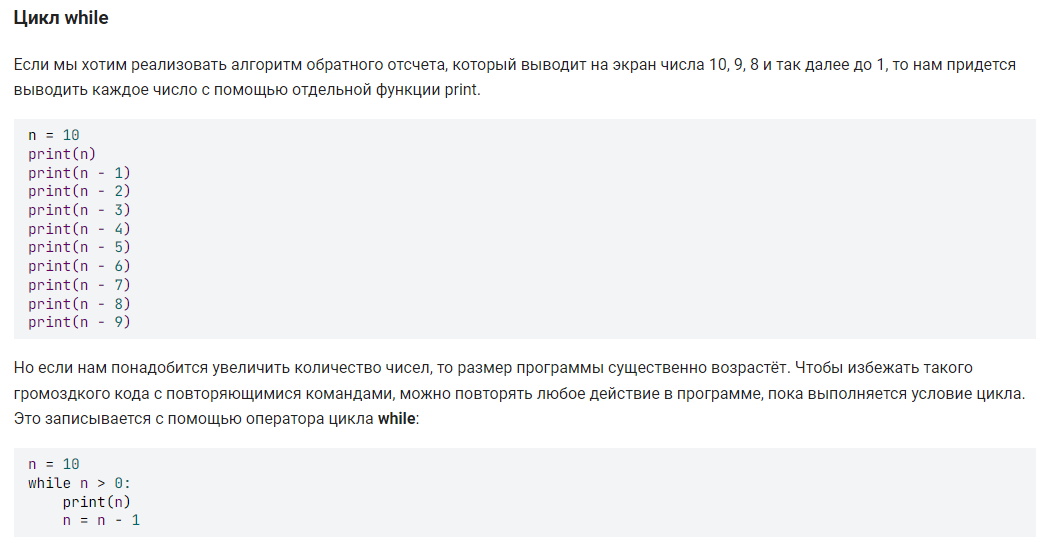


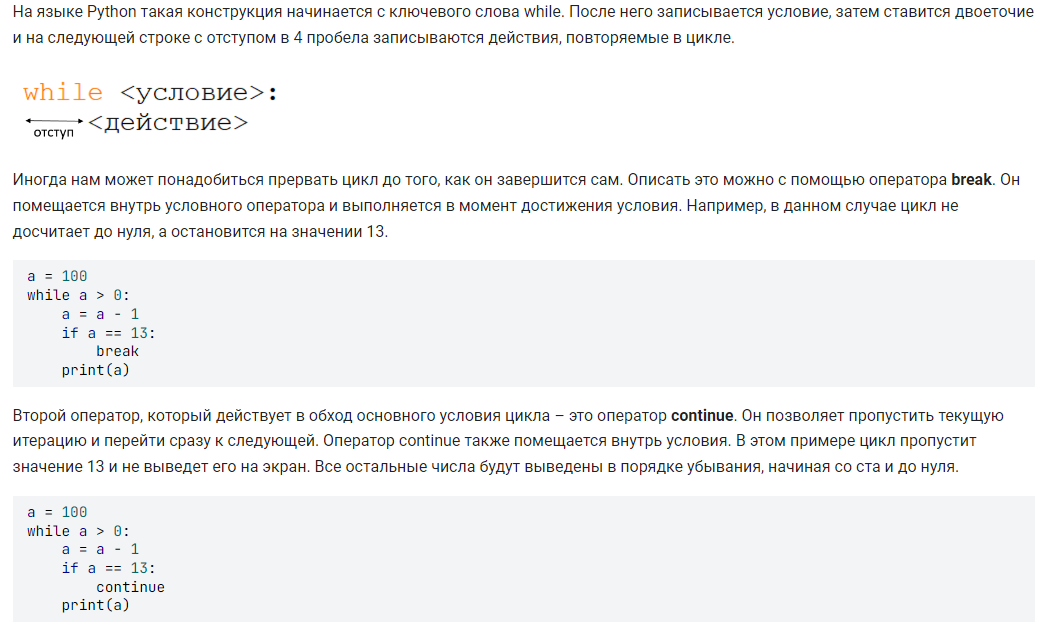


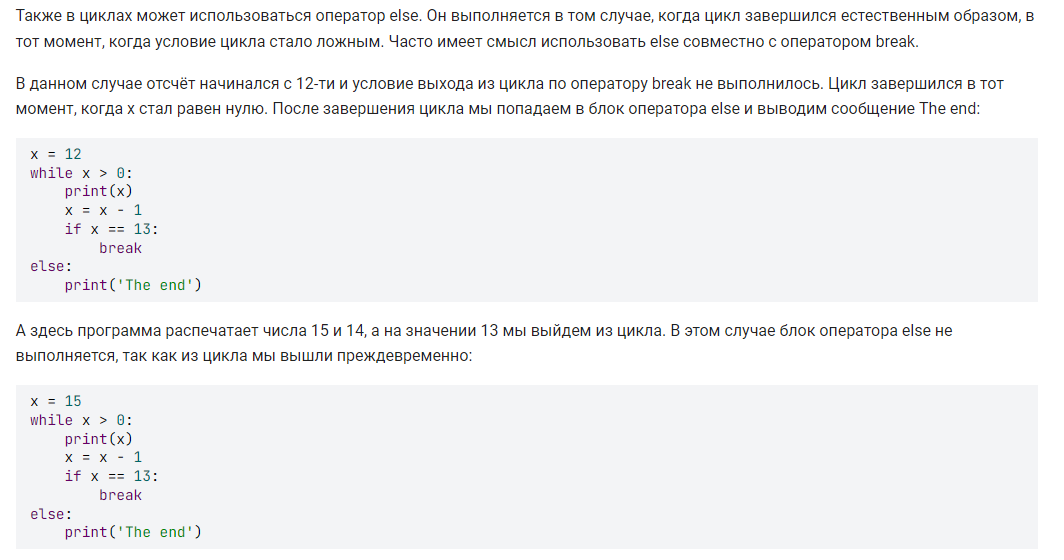


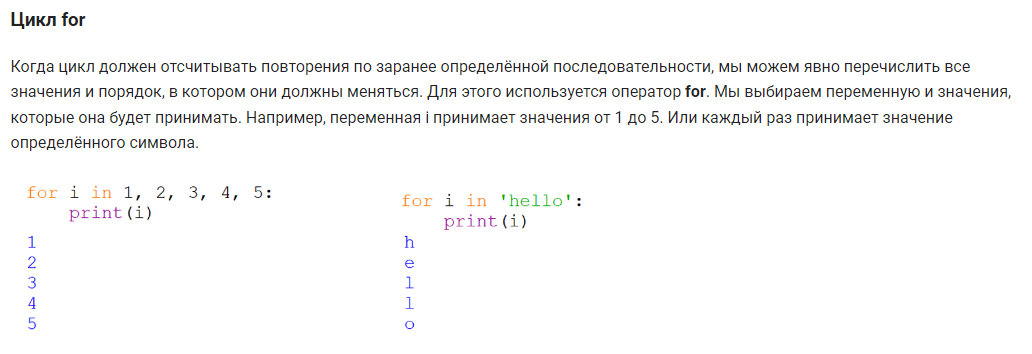


Циклы



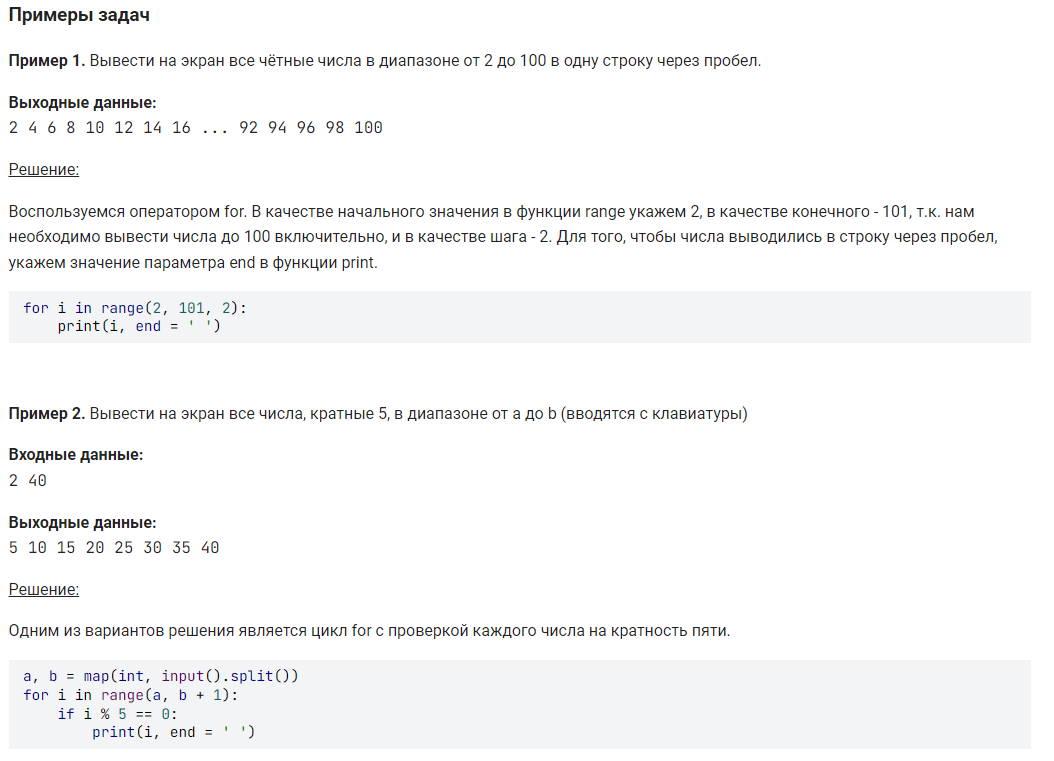


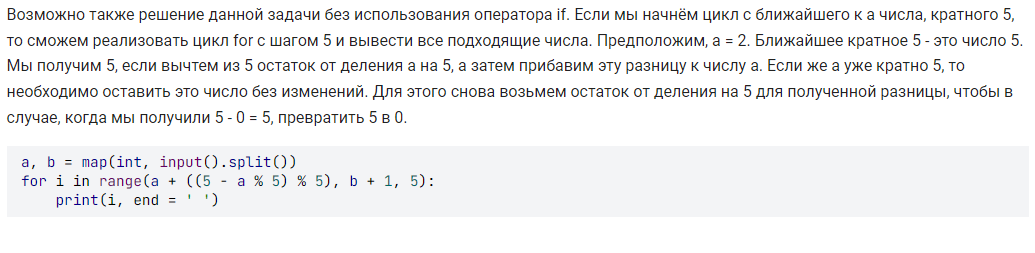






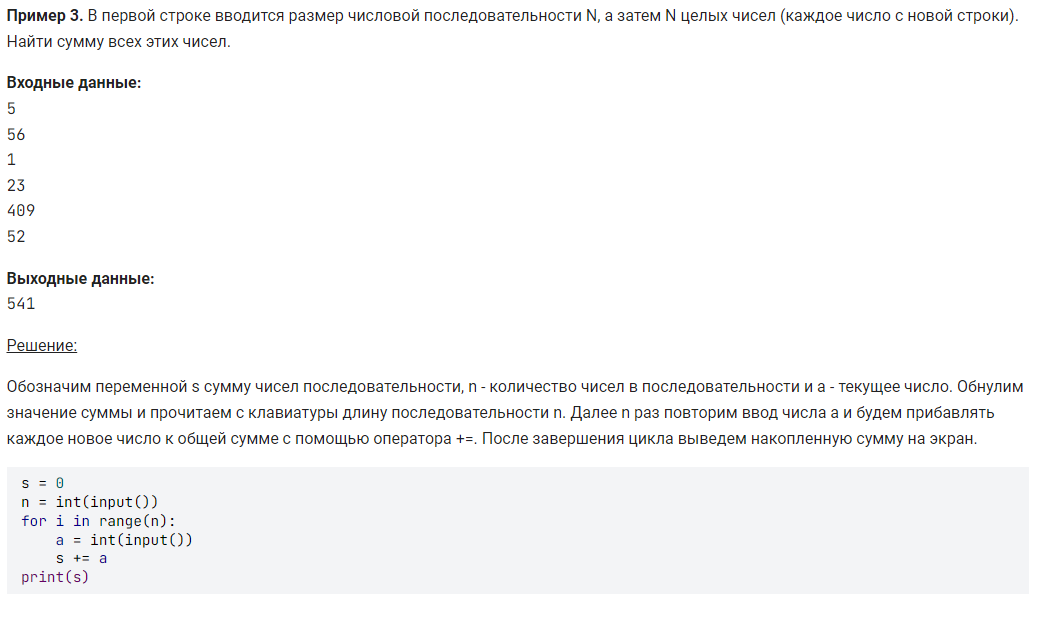
Какой параметр функции range является обязательным? Stop

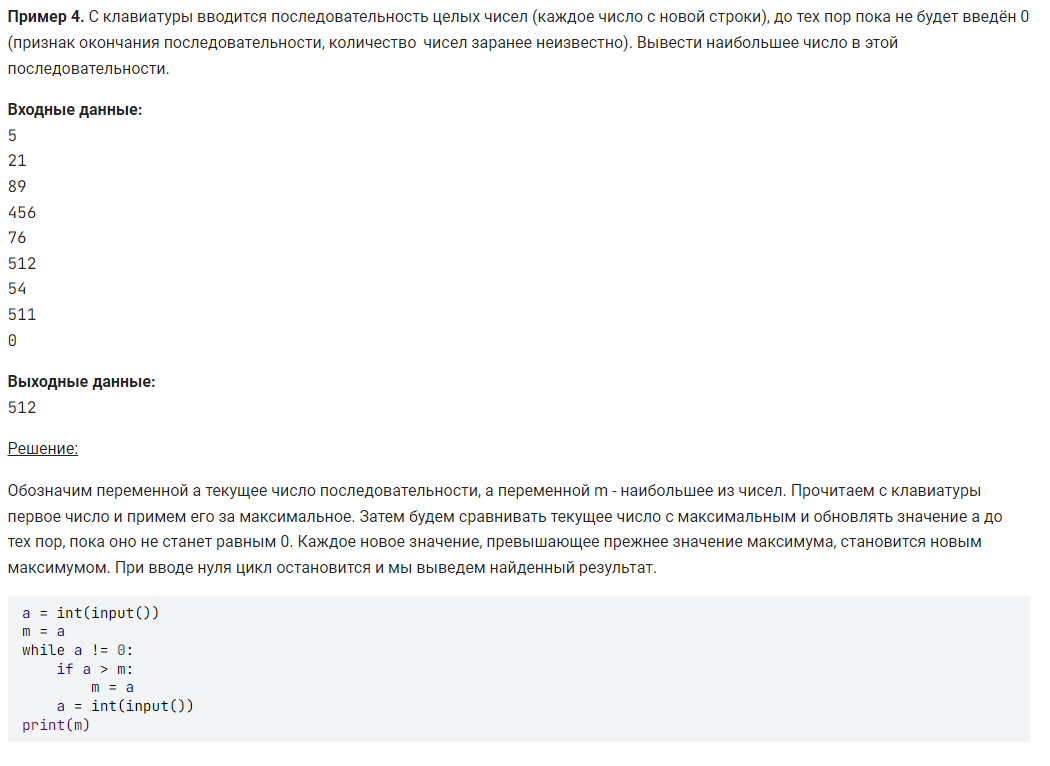




Какие виды циклов существуют в Python?

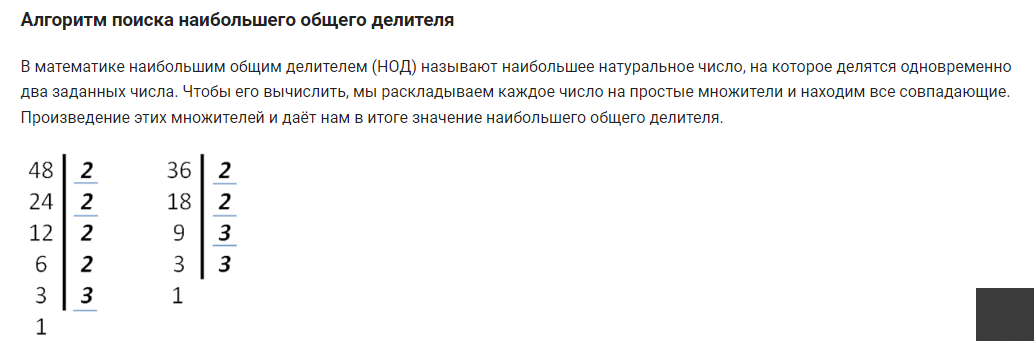
С параметром и с предусловием

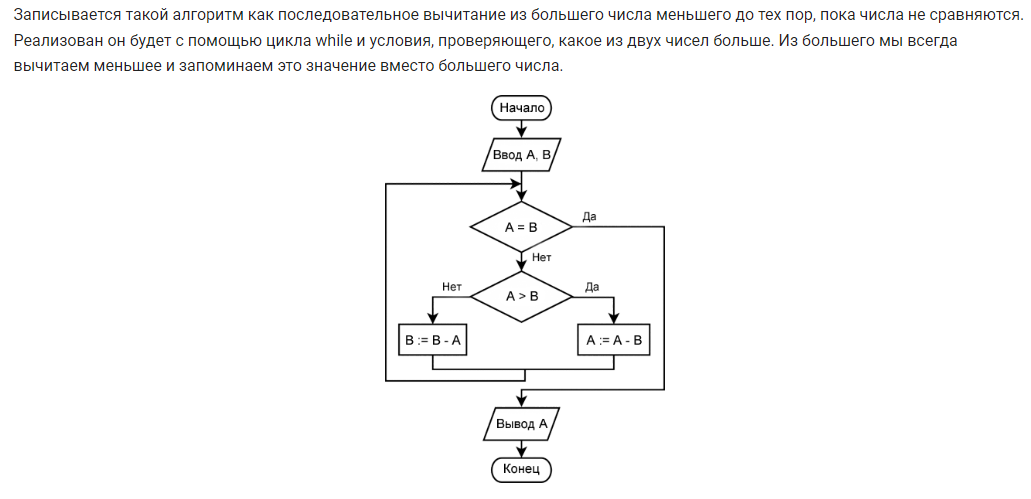


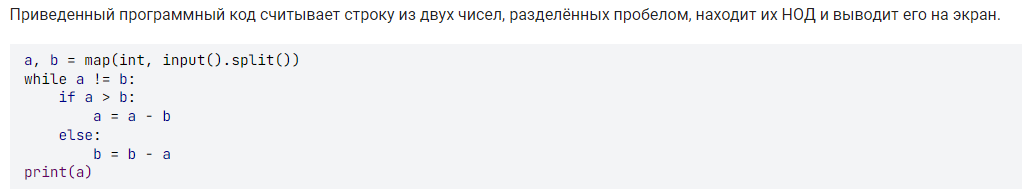


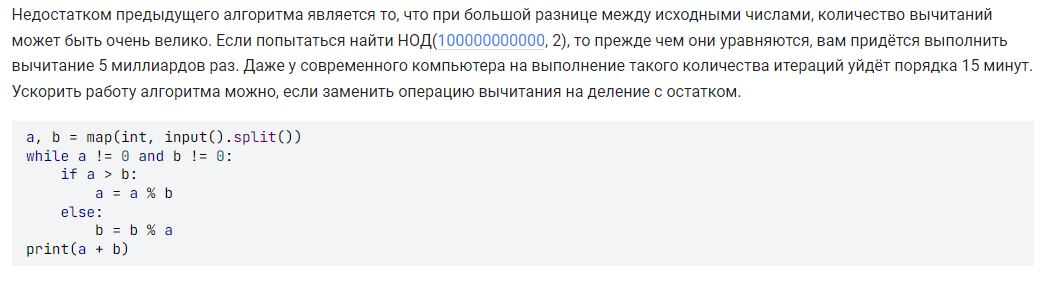
Алгоритмы

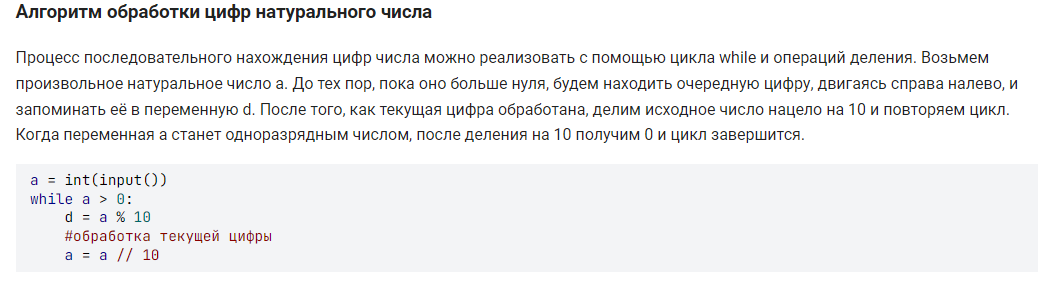
Алгоритмом называют некоторую последовательность действий, которая преобразует входные данные в результат. В большинстве программ мы сначала вводим данные, затем происходят определенные преобразования, после которых мы получаем нужный нам результат и выводим его.

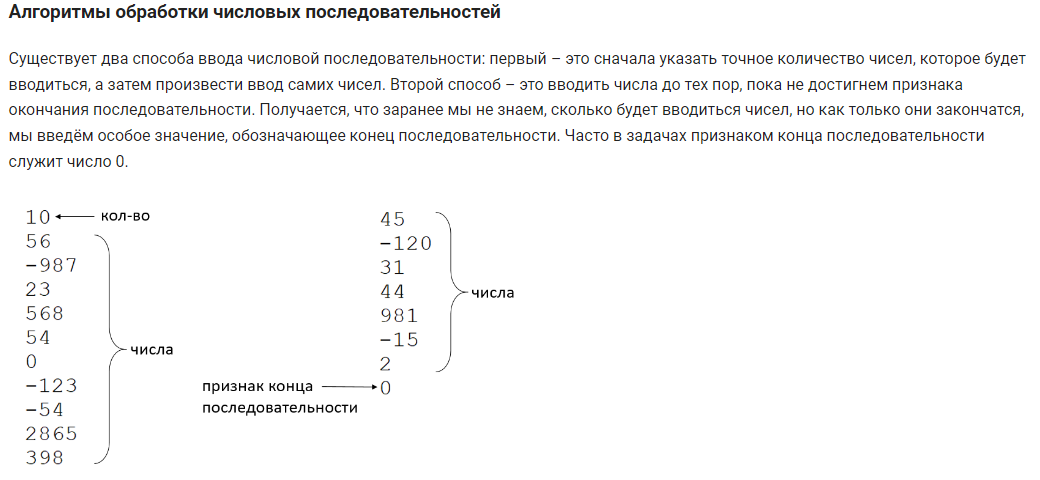


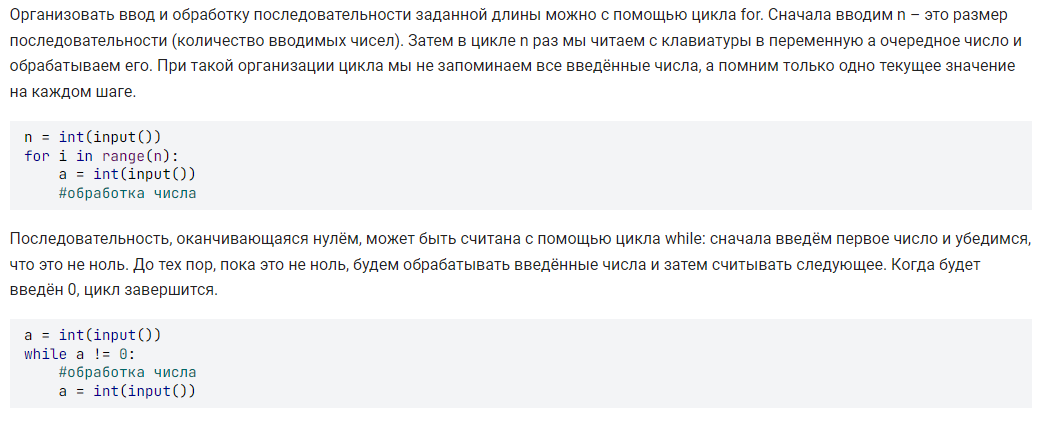


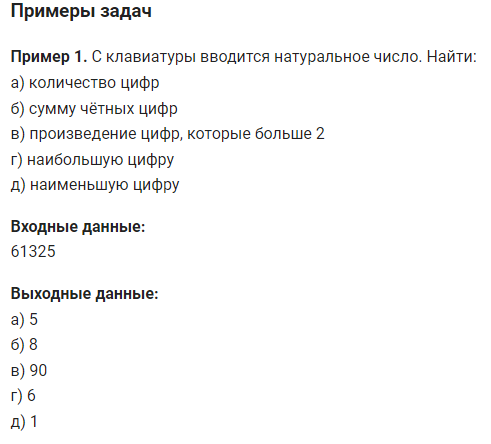
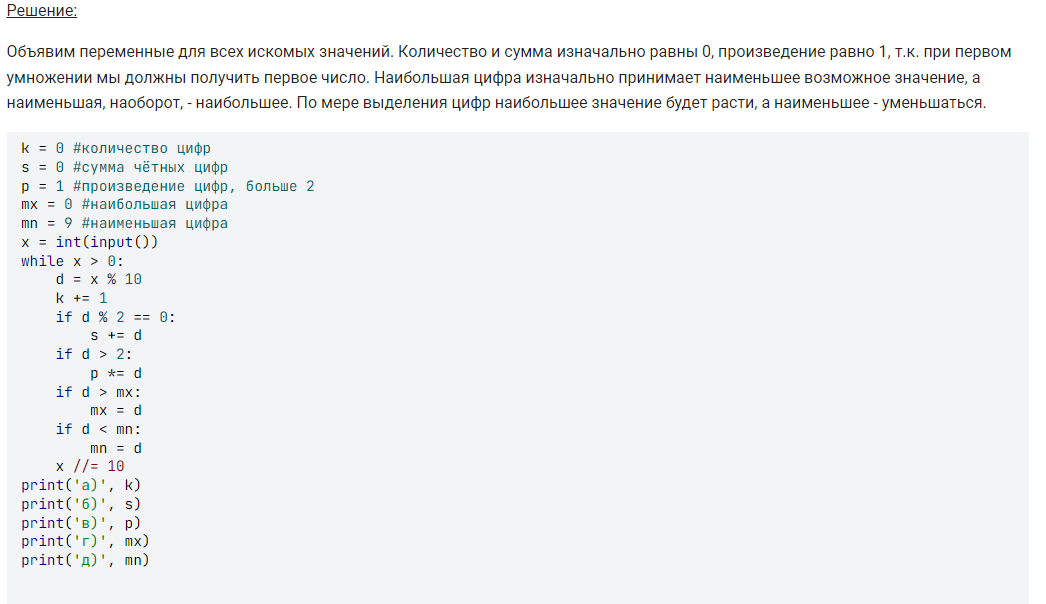










1

s = 0  
p = 1  
k = 0  
x = int(input())  
while x != 0:  
 d = x % 10  
 if x % 10 == d:  
 k += 1  
 if d % 10 > 5:  
 s += d  
 if d < 7:  
 p \*= d  
 x //= 10  
print(k)  
print(s)  
print(p)

k5 = 0  
mx = 0  
mn = 9  
for i in range (1):  
 a = int(input())  
 d = a % 10  
 k5 += 1  
 if k5 == 1:  
 i5 = i + 1  
 if a > mx:  
 mx = a  
 if a < mn:  
 mn = a  
print("Максимальная: ", mx)  
print("Минимальная", mn)

