**2023/09/24 00:00|Лекция. Списки. Индексация и методы списков**

## Списки. Индексация и методы списков

Сегодня мы рассмотрим такой тип данных, как списки. А также ознакомимся с их методами и индексацией. Начнем!

### Что такое список?

Список представляет собой **упорядоченный набор элементов**, каждый из которых имеет свой порядковый номер, то есть индекс.

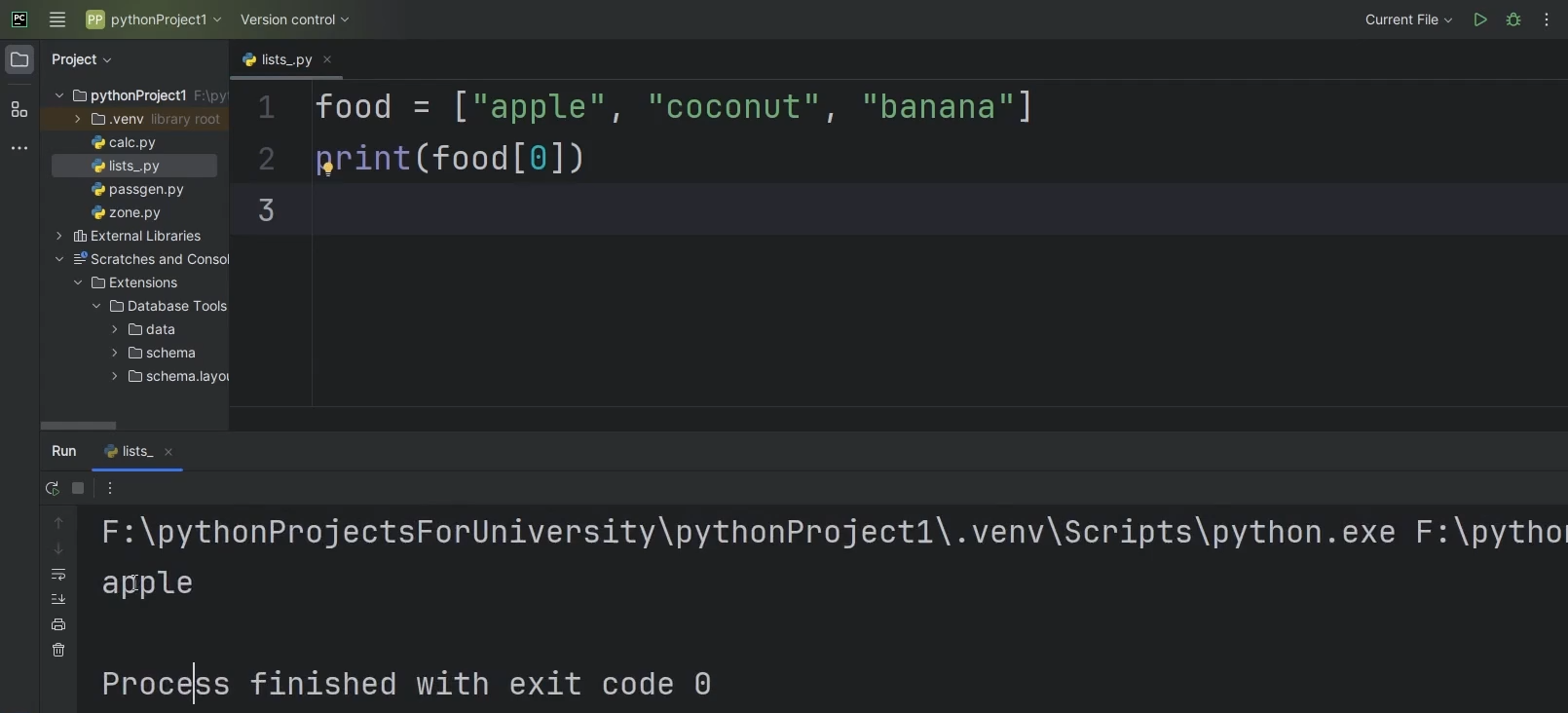
Чтобы создать список нужно просто написать имя переменной и через равно написать некие элементы в квадратных скобках. Например, внесем в скобки яблоко, кокос и банан (Рис.1). Если мы дадим команду print, то увидим содержимое нашего списка.



(Рис.1)

### Индексация списков

Мы можем также перебрать все элементы списка, то есть мы можем, например, работать с каждым элементом по отдельности. Для этого после имени нашего списка нужно открыть квадратные скобки и написать порядковый номер нужного элемента. Допустим нам нужно яблоко, поэтому впишем в скобки 0 (Рис.2), ведь как мы знаем индексация начинается с нуля.



(Рис.2)

Теперь давайте попробуем заменить этот элемент. Для этого пропишем, что элемент под номером 0 это персик (Рис.3). Даем команду print и видим, что теперь в списке вместо яблока находится персик.



(Рис.3)

Если мы заменим число в скобках на 1, то персик у нас займет место уже кокоса, а не яблока (Рис.4).



(Рис.4)

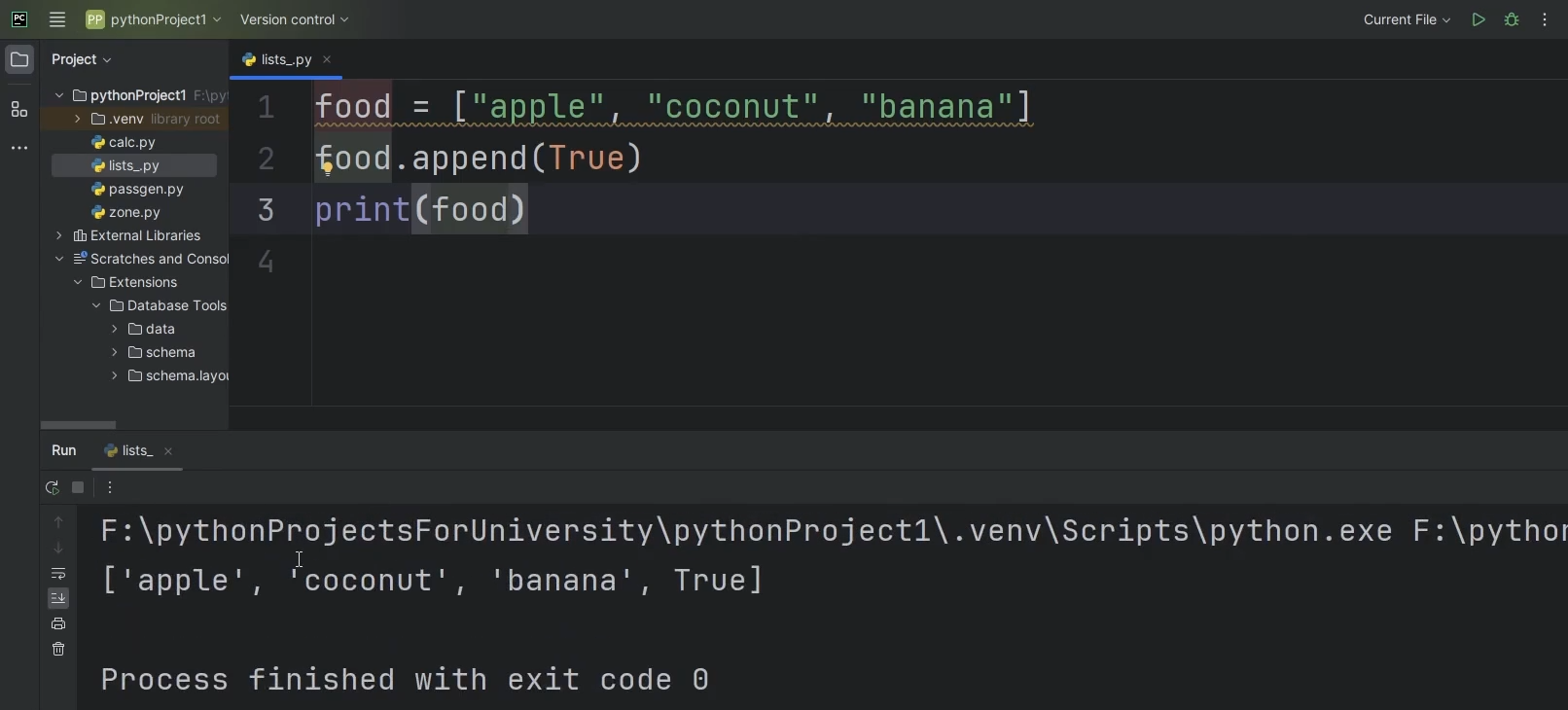
Подобные возможности в работе со списком говорят нам о том, что он относится к изменяемым объектам.

Вообще сами по себе списки являются коллекцией не самих элементов, а ссылок на эти элементы в памяти.

### Методы списков

С изменением элементов разобрались, а что, если нам нужно добавить элемент? Здесь уже нам могут помочь два метода. Методы записываются через точку после имени переменной.

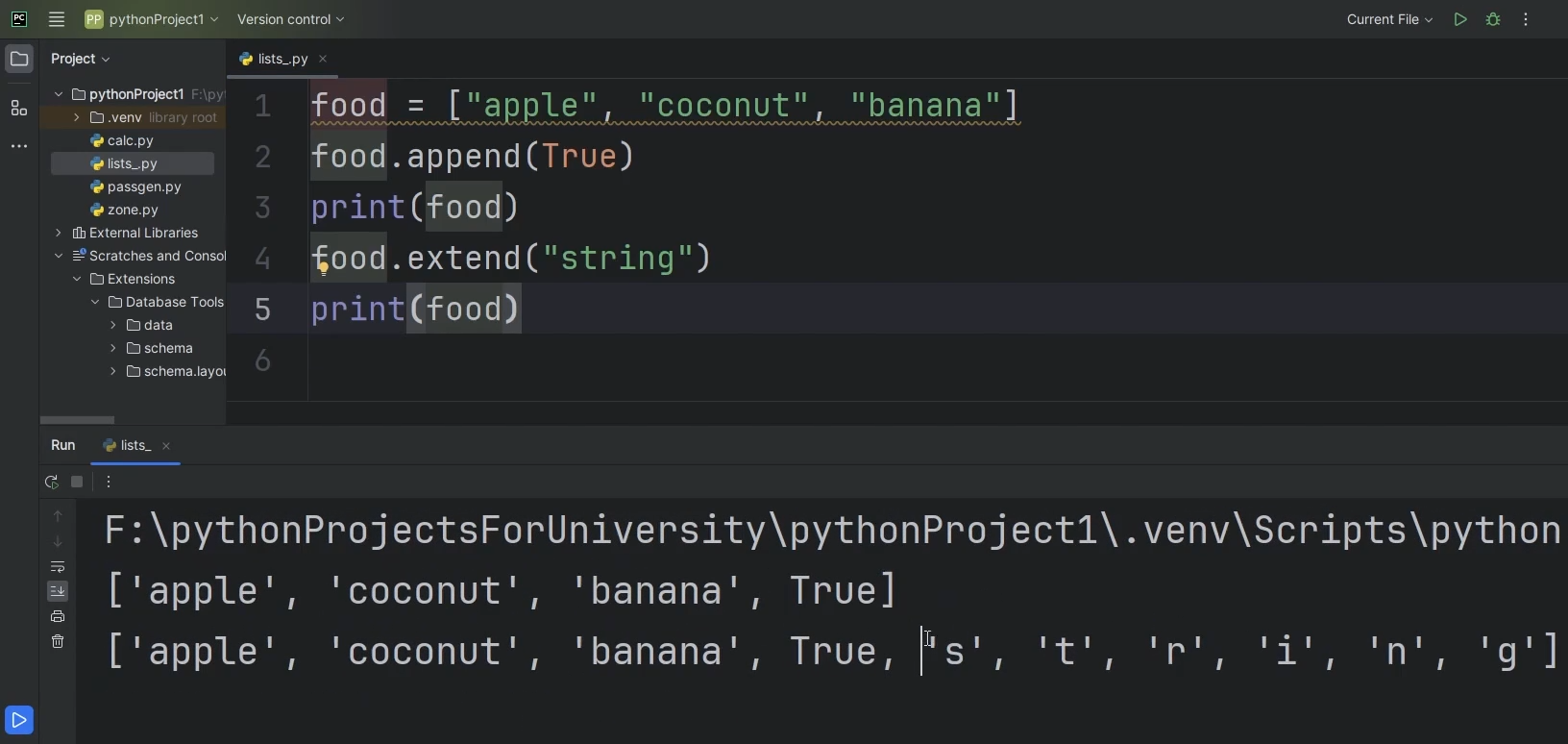
Первый метод – **append**, он добавляет новый элемент в конец списка. Для примера добавим с его помощью элемент «True», это продемонстрирует нам, что в списке могут находиться объекты разных типов. Если выведем наш список, увидим, что новый элемент успешно добавился в конец (Рис.5).



(Рис.5)

Таким образом мы видим, что элементы списка могут хранить в себе совершенно разные значения. Неважно будь то числа, строки и т. д.

Второй метод добавления элементов – **extend**, например добавим с его помощью строку «string» (Рис.6). В этом случае, как можно заметить, компьютер перебирает все символы, которые мы ему передали и добавит их в список по отдельности.



(Рис.6)

А вот если мы добавим внутрь круглых скобок квадратные, то есть попытаемся внести в наш список другой список, то компьютер их объединит, например (Рис.7).



(Рис.7)

Здесь важно подметить, что строка, которую мы внесли в этот раз, не была разобрана на символы. А все потому, что в этом случае она сама является элементом последовательности, которую мы присоединяем. Если вносить строку отдельно, то она считывается, как последовательность элементов, которые нужно присоединить.

Если мы захотим что-то убрать из списка, мы можем воспользоваться еще одним методом – **remove**. К примеру, уберем из списка элемент яблоко (Рис.8). Выведем наш список на экран и увидим, что теперь первый элемент в списке у нас кокос.

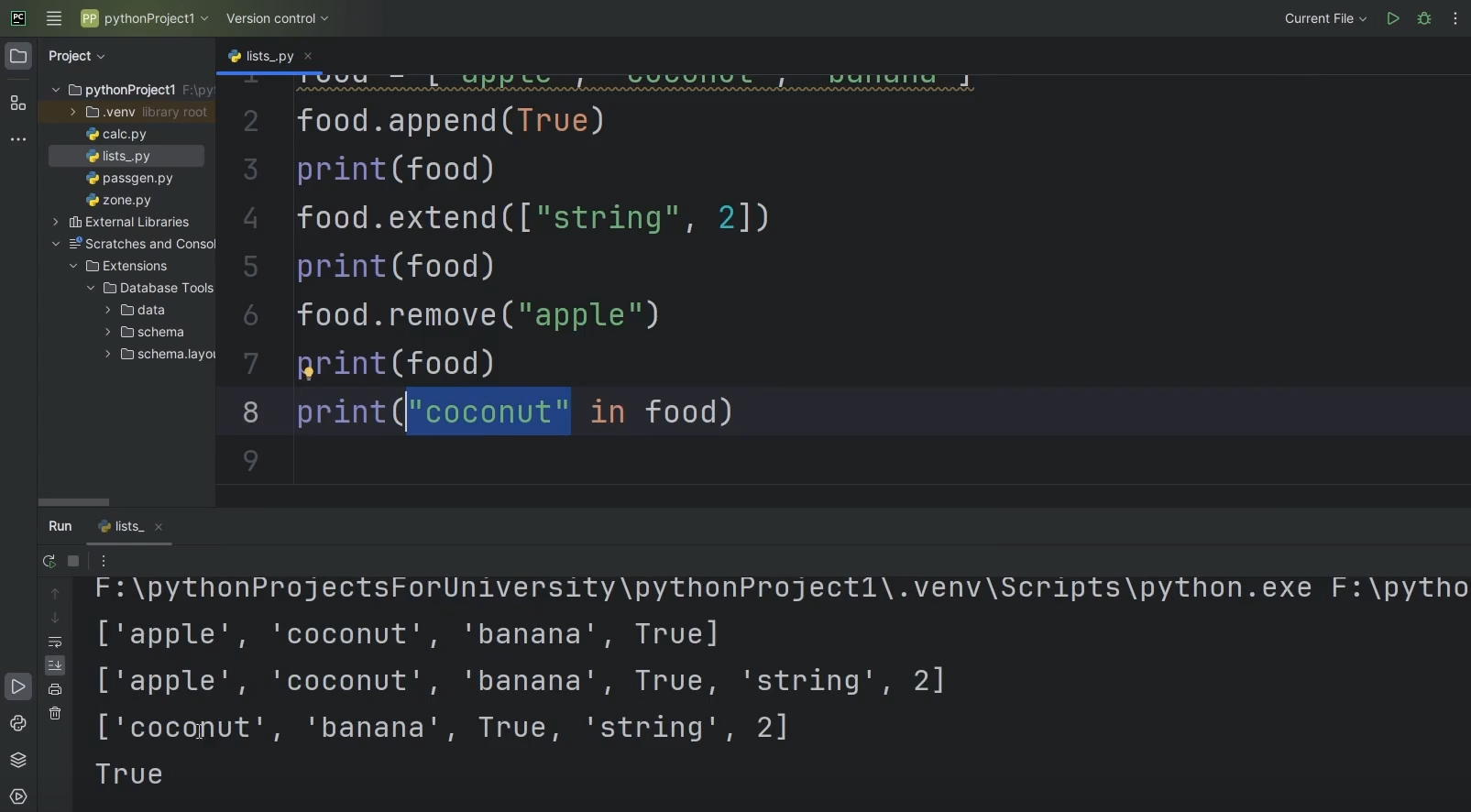


(Рис.8)

Методов существует еще множество, но о них мы сможем поговорить позже. Пока стоит запомнить самые основные и часто встречающиеся методы, с которыми мы сегодня познакомились.

### Команды для работы со списками

Помимо методов в работе со списками мы можем использовать и команды. Например, чтобы проверить есть ли конкретный элемент в нашем списке. Используем здесь команду **in**, которая поможет определить, является ли кокос элементом нашего списка «food» (Рис.9). В итоге мы видим ответ «True», который подтверждает, что это так.



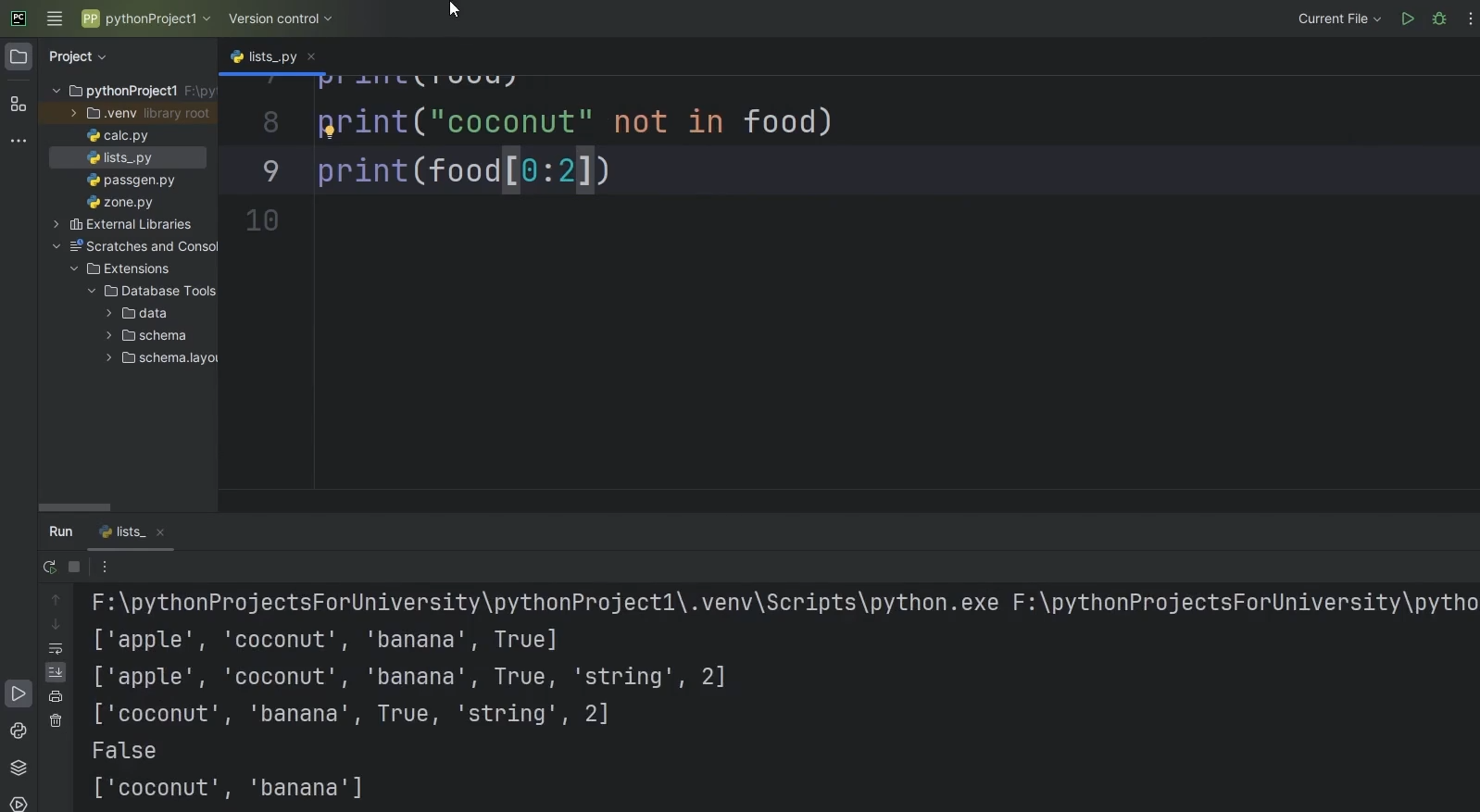
(Рис.9)

Так мы можем проверять есть ли какой-то элемент в списке или нет. Но мы можем проверить и обратное. То есть отсутствует ли некий элемент в списке. Для этого используется команда **not in**. Если зададим тот же запрос, то тогда уже ответ будет «False», поскольку кокос есть в списке (Рис.10).



(Рис.10)

В заключение поговорим о срезах, подобных тем, что есть у строк. Со списками можно проделывать то же самое. Допустим, выведем элементы от 0 до 2 и получим кокос и банан (Рис.11). Как мы знаем, последний элемент, в данном случае с индексом 2, в расчет не берется.



(Рис.11)

Ну и также мы можем задать шаг, к примеру 2 (Рис.12). Тогда ответ будет просто кокос.



(Рис.12)

### Интересные ссылки

<https://dzen.ru/a/Y1PCrL7TgDWtMzHU?sid=111922473914352640> - История IT. Создание Pascal и патент на компьютерную мышь