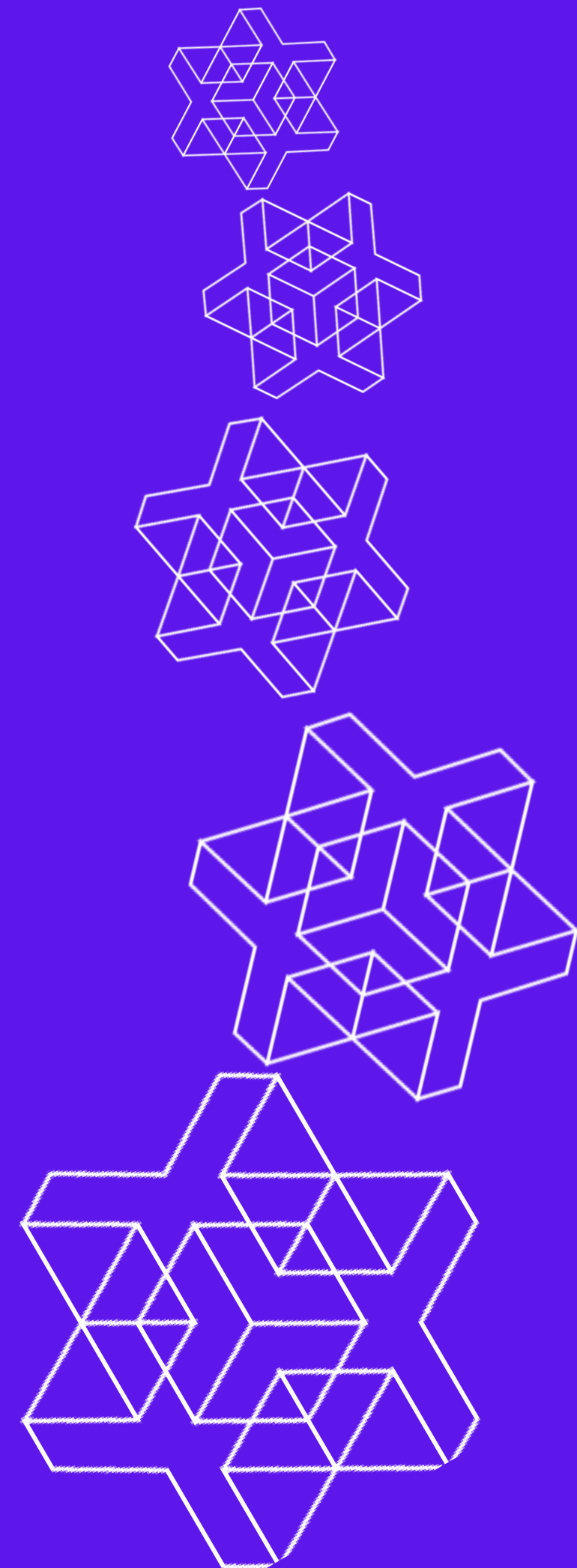
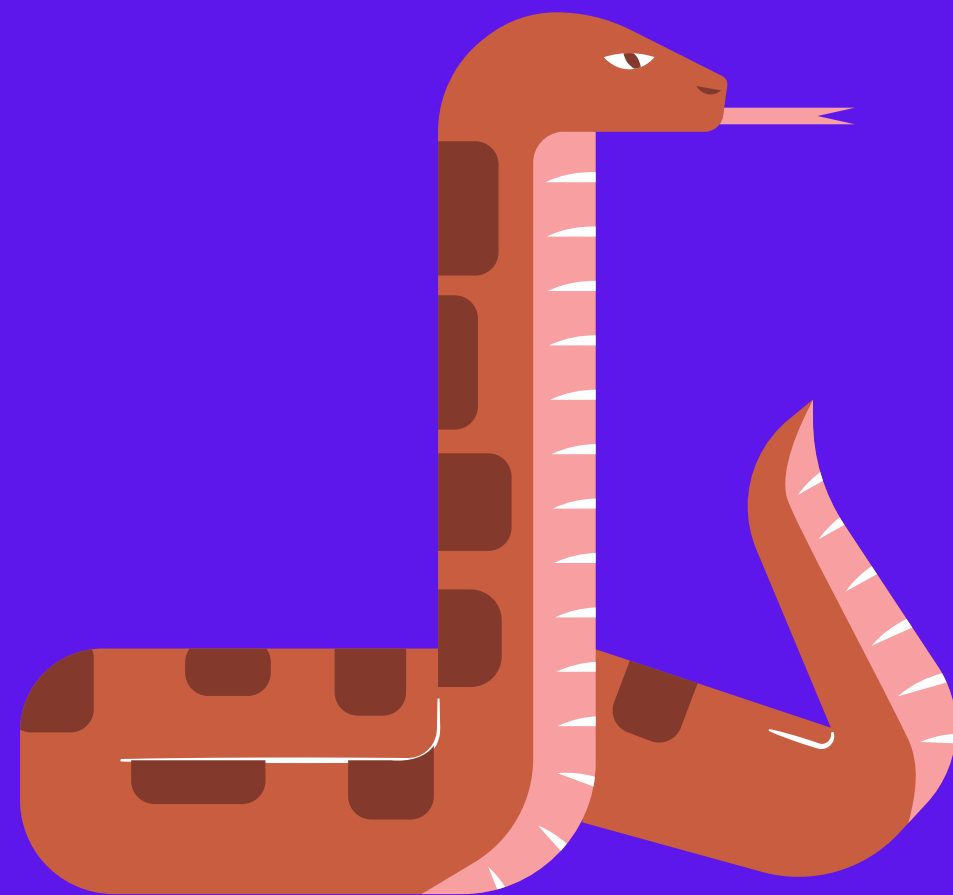



Урок 7

СПИСКИ В PYTHON





ЦЕЛИ СЕГОДНЯШНЕГО ЗАНЯТИЯ

1. Узнаем, что такое списки и как их создавать

2.



ЧТО МЫ УЖЕ УМЕЕМ

Что нужно сделать	Что можно использовать
Запросить ввод количества билетов	????
Вывести на экран фразу	????
Запрограммировать случайный выбор	????
Запрограммировать выбор	????
Заставить программу сработать ровно 5 раз	????



ЧТО МЫ УЖЕ УМЕЕМ

Что нужно сделать	Что можно использовать
Запросить ввод количества билетов	<code>input()</code>
Вывести на экран фразу	<code>print()</code>
Запрограммировать случайный выбор	<code>import random</code>
Запрограммировать выбор	<code>if ... else</code>
Заставить программу сработать ровно 5 раз	<code>for i in range(5):</code>



ЧТО ТАКОЕ СПИСОК

Список - последовательность из элементов

- элементов может быть любое количество
- элементами могут быть числа, строки, другие списки или словари
- элементы оформляются в []
- элементы отделяются друг от друга запятыми

```
spisok_pokupok = ['banan', 'moloko', 'bulochka']
```

```
numbers = [1, 2, 3, 4, 5]
```



КАК СОЗДАТЬ СПИСОК

Можно создать:

- пустой список

```
numbers = []
```

- список с элементами

```
numbers = [1, 2, 3, 4, 5]
```



КАК ВЫЗВАТЬ ЭЛЕМЕНТ СПИСКА

У каждого элемента есть свой индекс.

Индекс - это местоположение элемента в списке или его порядковый номер.

Нумерация индексов в списке всегда начинается с 0:

z =	[3,	7,	4,	2]
index	0	1	2	3

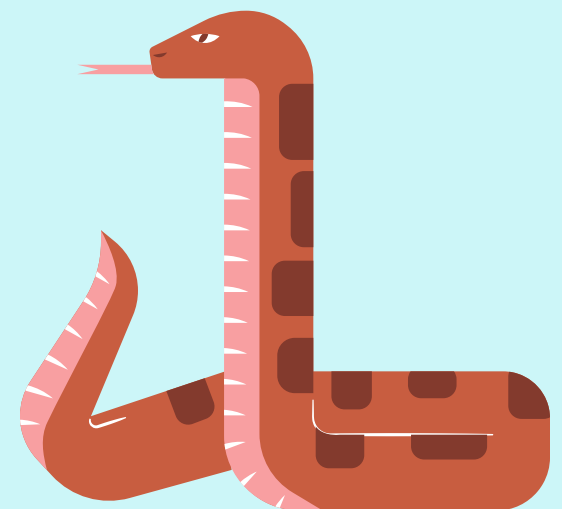
Чтобы вызвать элемент списка, нужно знать его индекс:

Пишем так: **имя_списка[индекс]**

```
z = [3, 7, 4, 2]
```

```
z[0]
```

```
3
```



КАК ВЫЗВАТЬ ВСЕ ЭЛЕМЕНТЫ СПИСКА

Для того чтобы вызвать все элементы списка по очереди, мы будем использовать цикл FOR:

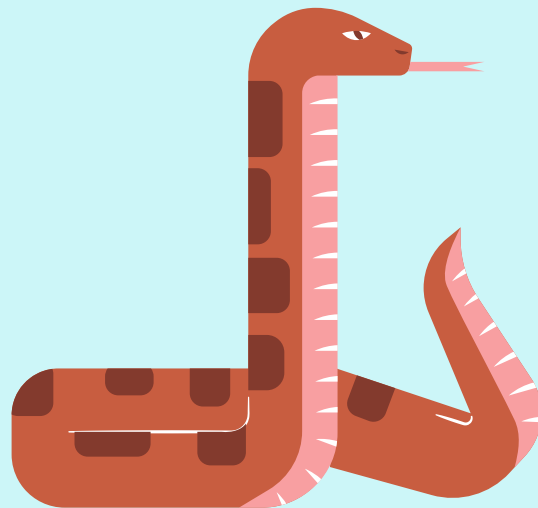
для каждого *i* в списке *z*:

i - временная переменная, которая будет по очереди принимать значение каждого элемента из списка

```
z = [3, 7, 4, 2]
```

```
for i in z:  
    print(i)
```

```
3  
7  
4  
2
```



КАК ДОБАВИТЬ/УДАЛИТЬ ЭЛЕМЕНТ

чтобы добавить элемент в конец списка
используй метод `append()`:

```
z = [3, 7, 4, 2]
```

```
z.append(99)
```

```
print(z)
```

```
[3, 7, 4, 2, 99]
```

чтобы удалить элемент из списка
используй метод `remove()`:

```
z = [3, 7, 4, 2]
```

```
z.remove(7)
```

```
print(z)
```

```
[3, 4, 2]
```



ПОЛЕЗНЫЕ КОМАНДЫ

sum() - находим сумму элементов списка.

Пример:

```
numbers = [1, 2, 3, 4]
print(sum(numbers)) # 10
```

min() и **max()** - находим минимальный или максимальный элементы.

Пример:

```
numbers = [1, 2, 3, 4]
print(min(numbers)) # 1
print(max(numbers)) # 4
```

sorted() - сортируем список.

Пример:

```
numbers = [3, 1, 4, 2]
print(sorted(numbers)) # [1, 2, 3, 4]
```

len() - находим длину (количество элементов) списка.

Пример:

```
s = [1, 22, 34, 4, 54]
print(len(s)) # 5
```

ЭТИ КОМАНДЫ
ЕЩЕ НЕ РАЗ
ПОНАДОБЯТСЯ!



ВНИМАНИЕ, У НАС НОВЫЙ ЗАКАЗ

К нам обратился интернет-магазин “Бозон”, им нужно создать способ подсчета общей суммы заказа.

Нам нужно написать код, который возьмет все покупки и выведет общую сумму заказа.



КАК ДОЛЖНО ВЫГЛЯДЕТЬ РЕШЕНИЕ

Так как наш код должен работать для любых заказов, то нам лучше всего написать его в виде функции.

Наша функция должна получать список покупок, а в результате должна подсчитывать общую сумму.

Пример:

на вход подаем список из цен товаров: [120, 35, 670]

в результате получим: $120 + 35 + 670 = 825$



КОД РЕШЕНИЯ

```
spisok_pokupok = [120, 35, 670]  
result = sum(spisok_pokupok)  
  
print(result)
```

825



ЗАДАЧИ

**Вот и все! А сейчас, пожалуйста,
выполните похожее задание
в тренажере!**

