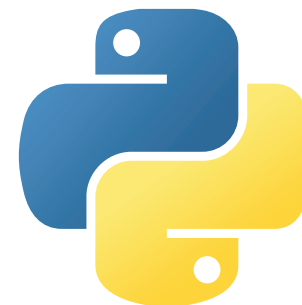
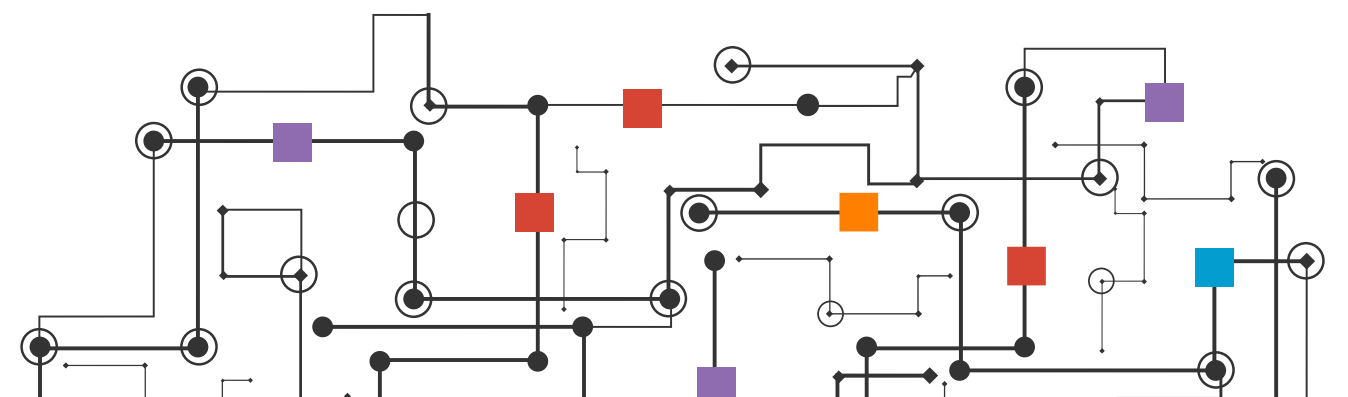
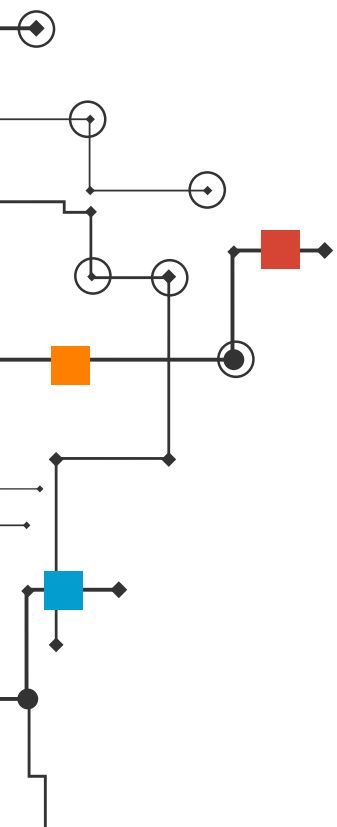


Основы программирования на языке Python



Основные типы данных	2	Инструкция ветвления	7
Основные числовые операции	3	Циклы	8
Основные логические операции	4	Функции	10
Основные функции	5	Списки	11



Основные типы данных

Название	Описание	Допустимые значения
int	Целочисленный	Любые целые числа
float	Вещественный	Любые числа с точностью до 15 значимых цифр
str	Строковый	Упорядоченные последовательности символов
bool	Логический	True, False

В Python динамическая типизация

Это означает, что в ходе исполнения программы тип данных, содержащихся в переменной, может меняться.

Преобразование типов

возможно с использованием функции, имя которой совпадает с названием конечного типа данных.

Основные числовые операции

Название	Пример	Результат
Присваивание	$a = b$	Переменная a примет значение b
Числовые		
Сложение	$a + b$	Сумма значений a и b
Вычитание	$a - b$	Разность значений a и b
Умножение	$a * b$	Произведение значений a и b
Деление	a / b	Частное значений a от b
Возведение в степень	$a ** b$	a в степени b
Целочисленное деление	$a // b$	Целая часть от частного значений a от b
Остаток от деления	$a \% b$	Остаток от деления a на b

Основные логические операции

a = 3, b = 5

a = True, b = False

Операции сравнения

Название	Пример	Результат
Больше	a > b	False
Меньше	a < b	True
Больше или равно	a >= b	False
Меньше или равно	a <= b	True
Равно	a == b	False
Не равно	a != b	True

Логические операции

Название	Пример	Результат
И	a and b	False
Или	a or b	True
Не	not a	False

Преобразование типа

	int (a)	1
	int (b)	0

Основные функции

Название	Пример	Результат
Ввода-вывода		
Ввод	<code>input ()</code>	Возвращает текст, введённый с клавиатуры
Вывод	<code>print (x)</code>	Выводит на экран значение x
Встроенные числовые		
Модуль	<code>abs (-3)</code>	3
Целая часть	<code>int (4.61)</code>	4
Округление до целых	<code>round (4.61)</code>	5

Основные функции

Название	Пример	Результат
Из модуля <code>math</code>		
Квадратный корень	<code>sqrt (25)</code>	5
Экспонента	<code>exp (8)</code>	e^8
Синус угла	<code>sin (a)</code>	Синус угла a , заданного в радианах
Косинус угла	<code>cos (a)</code>	Косинус угла a , заданного в радианах
Натуральный логарифм	<code>log (x)</code>	Натуральный логарифм t
Из модуля <code>random</code>		
Модуль	<code>random ()</code>	Случайное число на промежутке [0; 1)
Целая часть	<code>randint (a, b)</code>	Случайное целое число на промежутке [a; b]

Инструкция ветвления

Блоков **elif** может быть несколько

Схематичный текст

Пример кода

Описание

```
if <условие 1>:  
    <инструкция 1>  
    ...  
    < инструкция n>
```

```
if a = b:  
    a = a ** 2  
    b = b - 7
```

Условие 1 проверяется в первую очередь, инструкции выполняются, если оно истинно

```
elif <условие 2>:  
    <инструкция n + 1>  
    ...  
    < инструкция m>
```

```
elif a > b:  
    a = b + 5
```

Условие 2 проверяется, если ложно условие 1, и инструкции выполняются, если условие 2 истинно

```
else:  
    <инструкция m + 1>  
    ...  
    < инструкция k>
```

```
else:  
    b = a + 5
```

Инструкции, которые выполняются, если все условия оказались ложными

Циклы в Python

С заданным условием продолжения работы

Схематичный текст

```
while <условие продолжения работы>:  
    <инструкция 1>  
    ...  
    <инструкция n>
```

Пример кода

```
while a > b:  
    a = a - 1
```

С заданным условием окончания работы

Схематичный текст

```
while True:  
    <инструкции 1 - n>  
    if <условие окончания работы>:  
        break
```

Пример кода

```
while True:  
    a = a - 1  
    if a <= b:  
        break
```


Циклы в Python

Цикл с параметром

Схематичный текст

```
for <параметр> in <набор значений>:  
    <инструкция 1>  
    ...  
    <инструкция n>
```

Пример кода

```
for i in range(0, 4, 1):  
    a = a - 1
```

Функция `range`

возвращает последовательность из элементов целочисленной арифметической прогрессии.

Пример вызова

```
range(b, e, s)
```

Результат

последовательность целых чисел на промежутке **[b; e)** с шагом приращения **s**.

Функции

Схематичный текст

Описание

```
def <имя функции> (<входные параметры>):  
    <инструкция 1>  
    ...  
    <инструкция n>  
    return <выходные параметры>
```

Вызов в программе

В составе инструкции
<имя функции> (<значения входных параметров>)

Пример кода

Функция вычисления суммы цифр целого положительного числа.

Описание

```
def sumDigits (num):  
    s = 0  
    while num > 0:  
        s = num % 10  
        num = num // 10  
    return s
```

Вызов в программе

```
s_d = sumDigits (456)
```

Списки

Создание списка

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

```
A = [i for i in range(1, 5, 1)]
```

```
A = list(range(1, 5, 1))
```

Склейка списков

```
A = [1, 2, 3] + [4, 5, 6]
```

Выделение части списка

```
A[n:m],
```

где n и m – номера начального и конечного срезов промежутка.

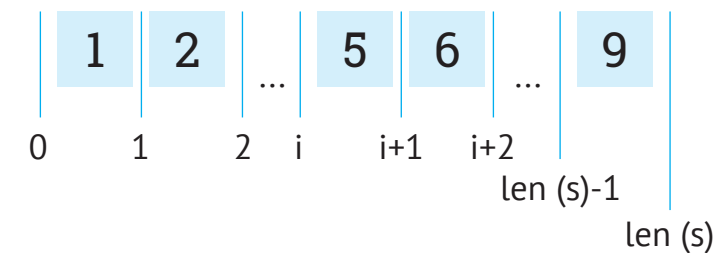
Длина списка

```
len(A)
```

Сумма элементов

```
sum(A)
```

Номера срезов списка



Списки

Перебор элементов списка без изменения их значений

```
for x in A:  
    <инструкции>
```

Перебор индексов элементов списка (значения элементов можно изменять)

```
for i in range(0, len(A), 1):  
    <инструкции>
```

Проверка присутствия элемента в списке

```
x in A
```

Сортировка списка по возрастанию

```
sorted(A)
```

Сортировка списка по убыванию

```
sorted(A, reverse = True)
```

Определение индекса элемента списка

```
A.index(x)
```