



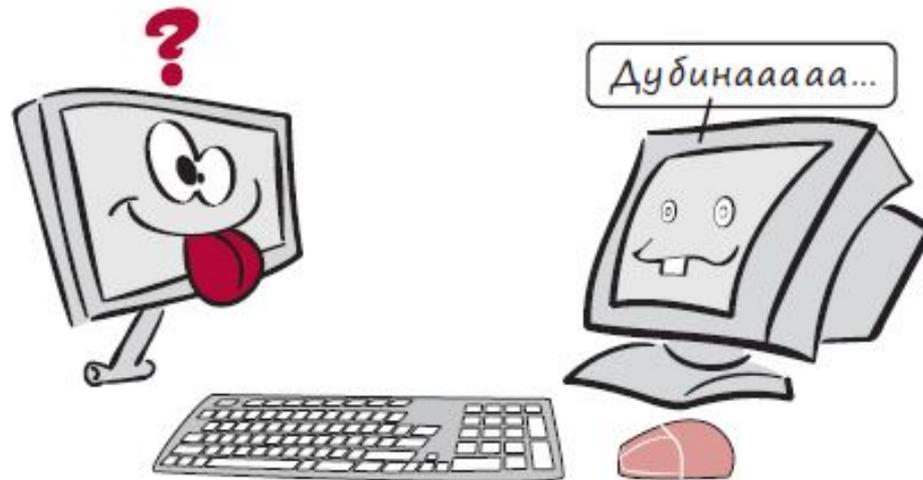
# PYTHON. ОЛИМПИАДНОЕ ПРОГРАММИРОВАНИЕ

Часть 1. Основы программирования (Hello World)

# ПРЕДИСЛОВИЕ

## ЧТО ТАКОЕ ПРОГРАММИРОВАНИЕ

Говоря по-простому, программирование заставляет компьютер выполнять некие действия. Компьютеры — это глупые машины, не имеющие никакого представления о том, как делается что бы то ни было. Вы должны им все объяснить, не упустив при этом ни одной детали.



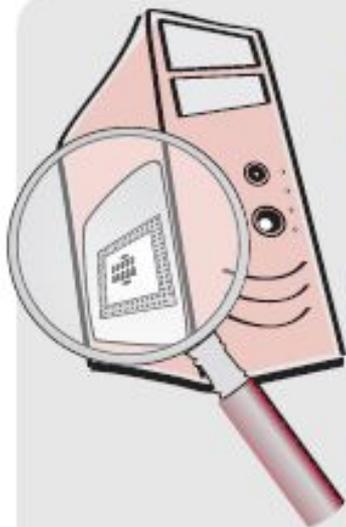
Дав компьютеру правильные инструкции, вы сможете делать множество потрясающих вещей.



## НОВЫЕ СЛОВА

*Инструкцией* называется базовая команда, направленная на выполнение одного конкретного действия.

## ЧТО ПРОИСХОДИТ ВНУТРИ



Компьютеры «думают» при помощи множества электрических контуров. На базовом уровне эти контуры представляют собой переключатели, которые могут находиться в двух положениях: включено или выключено. Инженеры и специалисты по информатике используют для обозначения этих положений значения 1 и 0. Все эти единицы и нули представляют собой код, называемый двоичным. Этот термин появился потому, что мы имеем дело с «двумя состояниями». Эти состояния – включено и выключено, или 1 и 0. Вы знаете, что двоичная цифра равна биту?



# PYTHON — ЯЗЫК ДЛЯ НАС И ДЛЯ КОМПЬЮТЕРА



Python - один из многих языков программирования.  
Позволяет приручить ваш компьютер, научить его  
слушаться ваших инструкций.



## ПРИМЕР ЭЛЕМЕНТАРНОЙ ПРОГРАММЫ:

- $a = 4$
- $b = 2$
- $c = a + b$
- `print(c)`

Вывод программы:

**6**





# ДУМАЙ КАК ПРОГРАММИСТ

## Запомни операторы:

- + служит для сложения двух операндов;
- для вычитания...;
- \* для умножения...;
- / для деления.

## Запомни функцию:

print(a) - выводит значение параметра a.



## ПРИМЕР ЭЛЕМЕНТАРНОЙ ПРОГРАММЫ:

- `a=4`
- `b=2`
- `c=a+b`
- `print("Результат вычислений равен:")`
- `print(c)`

### Вывод программы:

Результат вычислений равен:

6



## ПРИМЕР ЭЛЕМЕНТАРНОЙ ПРОГРАММЫ:

- `a=4`
- `b=2`
- `c=a+b`
- `print("Результат вычислений равен", c)`

### Вывод программы:

Результат вычислений равен 6



## ПРИМЕР ЭЛЕМЕНТАРНОЙ ПРОГРАММЫ:

- `a=int(input())`
- `b=2`
- `c=a+b`
- `print("Результат вычислений равен", c)`

### Вывод программы:

Результат вычислений равен 6



# ДУМАЙ КАК ПРОГРАММИСТ



Запомни функцию:

**input()** - позволяет пользователю ввести строку.

Например `a=input()`

присвоит переменной `a` значение введенной пользователем строки.



## ПРИМЕР ЭЛЕМЕНТАРНОЙ ПРОГРАММЫ:

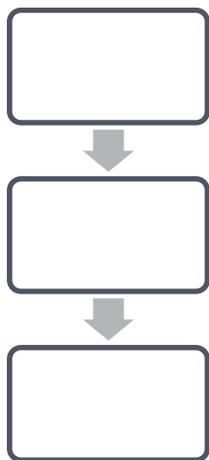
- `print("Введи любое число, а я его увеличу на два: ")`
- `a=int(input())`
- `b=2`
- `c=a+b`
- `print("Результат вычислений равен", c)`

### Вывод программы:

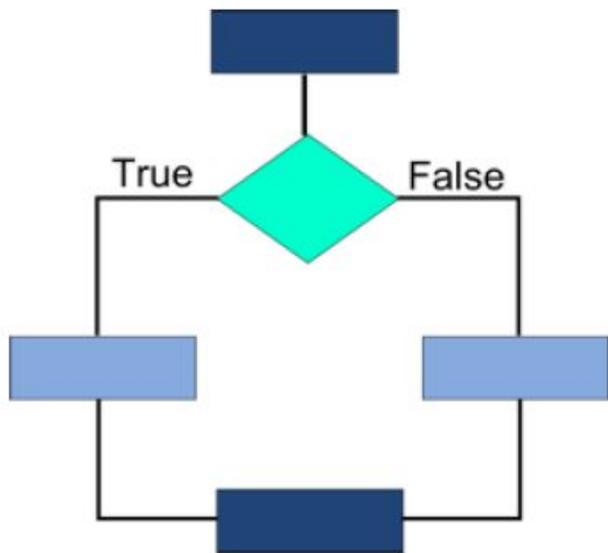
Результат вычислений равен 6



## ЛИНЕЙНЫЕ И НЕЛИНЕЙНЫЕ АЛГОРИТМЫ



**Линейный алгоритм**, это такая последовательность операций, которая не меняется при любых исходных данных.

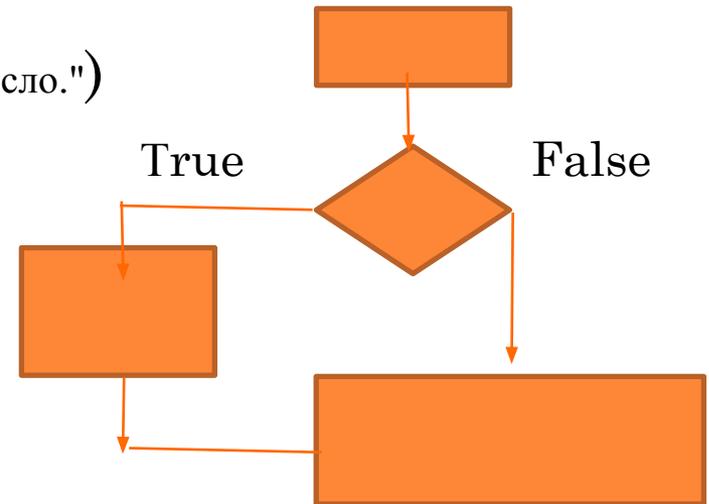


Кроме того существуют **нелинейные алгоритмы**.



## ПРИМЕР ПРОГРАММЫ, РЕАЛИЗУЮЩЕЙ ЭЛЕМЕНТАРНЫЙ НЕЛИНЕЙНЫЙ АЛГОРИТМ:

- `a=int(input("Введите целое число от 1 до 20 : "))`
- `if a<20:`
- `print("ПОЗДРАВЛЯЮ! Вы ввели правильное число.")`



### Вывод программы:

**ЗАРАНЕЕ НЕ ИЗВЕСТЕН.** Поведение программы зависит от исходных данных введенных пользователем



# ЖУРАВЛИКИ

(Время: 1 сек. Память: 16 Мб Сложность: 7%)

Петя, Катя и Сережа делают из бумаги журавликов. Вместе они сделали  $S$  журавликов. Сколько журавликов сделал каждый ребенок, если известно, что Петя и Сережа сделали одинаковое количество журавликов, а Катя сделала в два раза больше журавликов, чем Петя и Сережа вместе?

## Входные данные

В единственной строке введенной пользователем записано одно натуральное число  $S$  – общее количество сделанных журавликов ( $S < 10^6$ ).

## Выходные данные

В единственную выведенную строку нужно вывести три числа, разделенных пробелами – количество журавликов, которые сделал каждый ребенок (Петя, Катя и Сережа).

## Примеры

№	INPUT.TXT	OUTPUT.TXT
1	6	1 4 1
2	24	4 16 4
3	60	10 40 10



# "ЧЕРЕПАШЬЯ ГРАФИКА"-1

- `import turtle`
  - `turtle.reset()`
  - `turtle.down()`
  - `turtle.forward(35)`
  - `turtle.left(55)`
  - `turtle.up()`
  - `turtle.forward(30)`
  
  - `turtle.goto(100,150)`
  - `turtle.down()`
  - `turtle.color("red")`
  - `turtle.circle(100)`
  - `turtle.mainloop()`
- `import turtle` #подключить модуль turtle
  - **Команды для перемещения курсора:**  
`forward(n)` #вперед на n пикселей  
`left(n)` #влево на n градусов  
`right(n)` #вправо на n градусов  
`circle(r)` #начертить окружность радиуса r, с центром слева от курсора, если  $r > 0$ , справа, если  $r < 0$
  - `goto(x,y)` #переместить курсор в точку с координатами (x,y)
  - **Команды рисования**  
`down()` #опустить курсор для рисования  
`up()` #поднять курсор  
`color(s)` #где s #цвет рисования курсора  
пример: "red", "blue"  
`begin_fill(),end_fill()` #рисует закрашенные области (начало и конец рисунка)
  - **Дополнительно:**  
`reset()` #очищается экран, возвращает курсор  
`clear()` #очистить экран
- `mainloop()` #задержка окна



ДАВАЙТЕ ИГРАТЬ И КОДИТЬ !

