

алгоритмика

Международная школа
программирования для детей

Модуль 1. Урок 2.

Переменные

Во время урока откройте презентацию в режиме Просмотр (ctrl+enter).

Так ученики увидят только свои слайды. В презентации есть методические слайды, которые нужны только преподавателю. Они отмечены иконкой «глаз».



Слайд-инструкция



**Что мы делали на
прошлом уроке?**

Обсуждение

Мы узнали:

- язык программирования,
- алгоритм,
- программа,
- правила синтаксиса в Python,
- функции,
- строки,
- функция `print()`.



Что такое «язык программирования»?

Обсуждение

Язык программирования —

**язык для общения с машинами,
в котором каждая команда имеет только одно значение.**

Повторение

A large, stylized white question mark is centered on a yellow background. The question mark is composed of a circular dot on the left and a curved tail on the right. The text is overlaid on the question mark.

Что такое «алгоритм»?

Обсуждение

Алгоритм —

последовательность действий
для достижения цели.

Важен порядок выполнения
команд в алгоритме.



Повторение



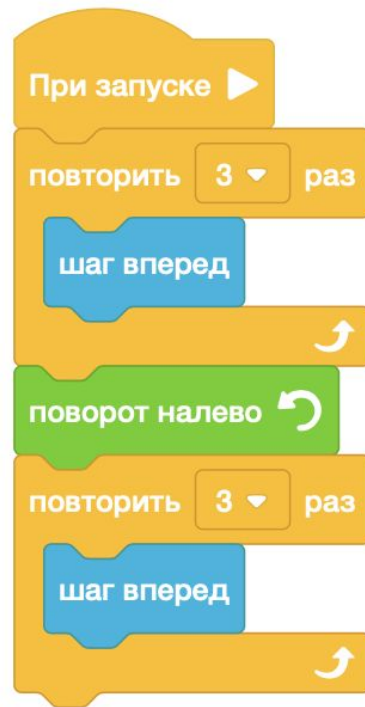
Что такое «программа»?

Обсуждение

Программа —

алгоритм, записанный на языке программирования.

Программа составляется из команд, которые есть в языке программирования.



Повторение

Где правильный алгоритм?

Обсуждение

Отметь правильно составленные алгоритмы

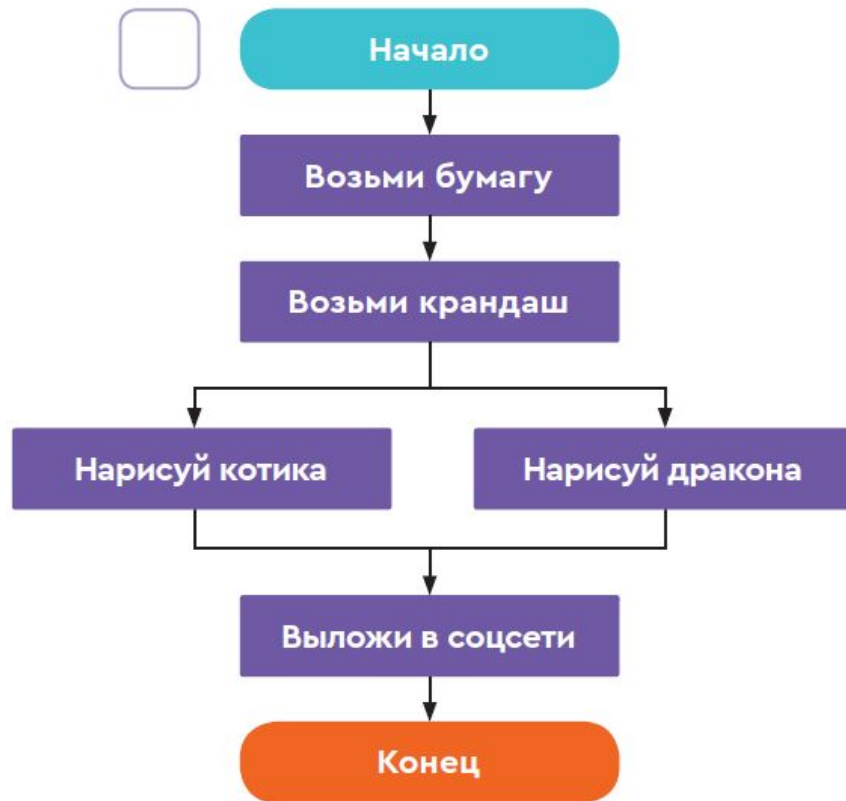


Домашнее задание

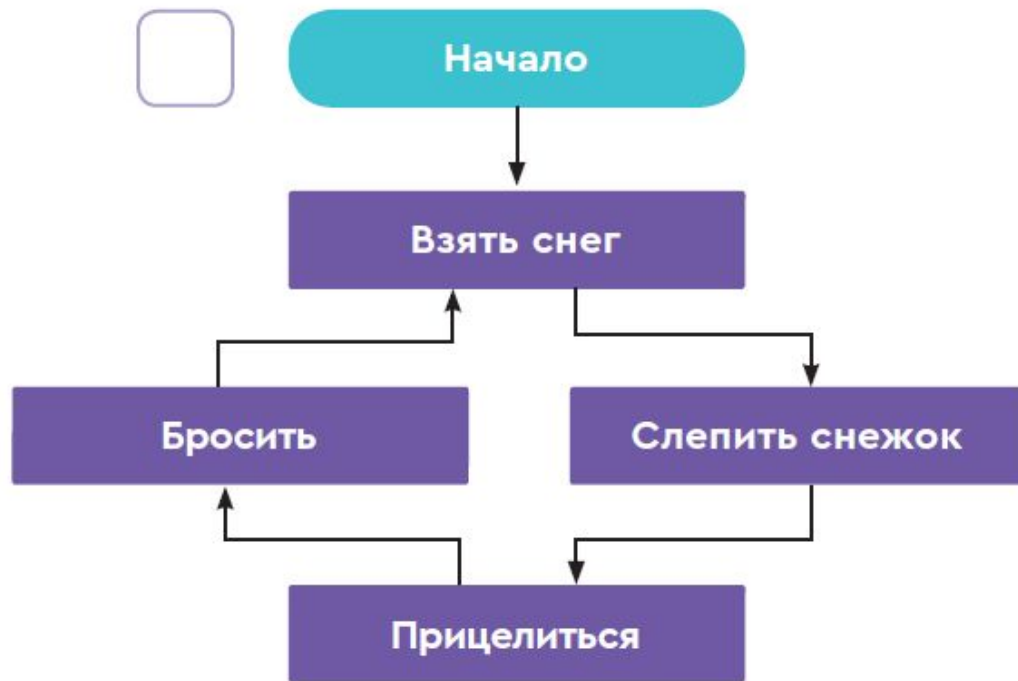
Отметь правильно составленные алгоритмы



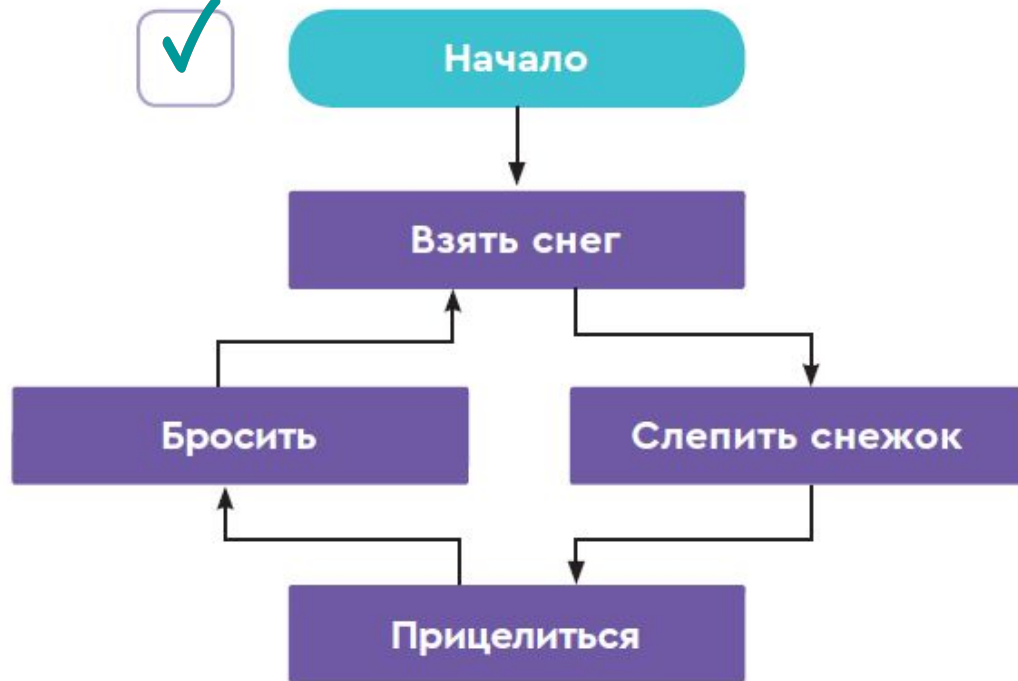
Домашнее задание



Домашнее задание



Домашнее задание



Домашнее задание



Читай книгу

Сделай уроки

Программируй

Иди гулять

Играй

Домашнее задание

Какие существуют правила синтаксиса в Python?

Обсуждение

Правила синтаксиса в Python:

- правило порядка
- правило начала
- аккуратность в буквах и знаках



Правила синтаксиса Python.

Правило порядка

Команды будут выполняться по очереди, если они написаны ровно друг под другом:

- или Python начнёт искать специальные указания, в каком порядке их выполнять;
- без них Python выдаст ошибку и не будет выполнять программу вовсе.

Повторение

Правила синтаксиса Python.

Правило начала

Первая команда должна быть написана в самом начале строки.

Команда пишется на одной строчке.
Переход на следующую строку означает, что команда закончилась и началась другая.

Повторение

Правила синтаксиса Python.

Аккуратность в буквах и знаках

В Python нельзя путать строчные буквы и заглавные буквы (маленькие и большие)

Случайная ошибка всего в одном символе (даже один лишний пробел) может сломать программу

Повторение

Что такое функция?


Обсуждение

Запись команды функции

`имя_функции` (аргумент1, аргумент2, ...)




Имя алгоритма



Данные,
как выполнять алгоритм сейчас.
Параметры разделяются через
запятые

Функция может быть без аргументов

имя_функции ()



Скобки указывают,
что это функция

**Что делает функция
print()?**

Функция print()

```
print ("Привет, хакеры!")
```



Текст пишется в кавычках,
чтобы Python не применял к
строке правила

A large, white, stylized question mark is centered on a solid yellow background. The question mark is composed of a circular dot on the left and a curved hook on the right. The text "В чем ошибка?" is written in bold black font across the center of the question mark.

В чем ошибка?

Обсуждение

Укажи строки, в которых нарушено правило порядка

```
1 import tkinter
2 w = Tk()
3     cnv = Canvas(root,width=640,height=480)
4     cnv.pack()
```

Укажи строки, в которых нарушено правило порядка

```
1 import tkinter
2 w = Tk()
3     cnv = Canvas(root,width=640,height=480)
4     cnv.pack()
```

3

Укажи строки, в которых нарушено правило порядка

```
1     age = 10
2     myname = "Peter"
3     yourname = input("What is your name?")
4     print("Hello")
5     print(yourname)
6     print("My name is")
7     print(myname)
8     print("I am")
9     print(age)
```


Укажи строки, в которых нарушено правило порядка

```
1   age = 10
2   myname = "Peter"
3   yourname = input("What is your name?")
4   print("Hello")
5   print(yourname)
6   print("My name is")
7   print(myname)
8   print("I am")
9   print(age)
```

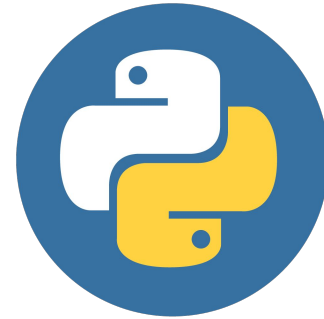
2, 5

Переменные

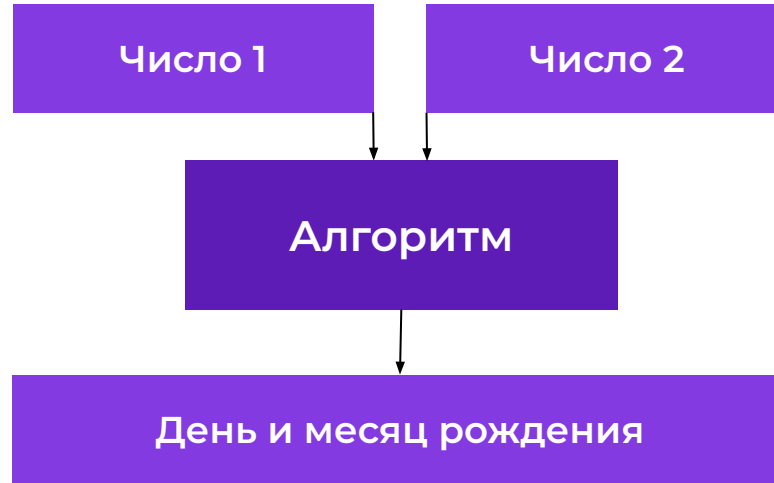
Для чего нужны?

Как создаются?

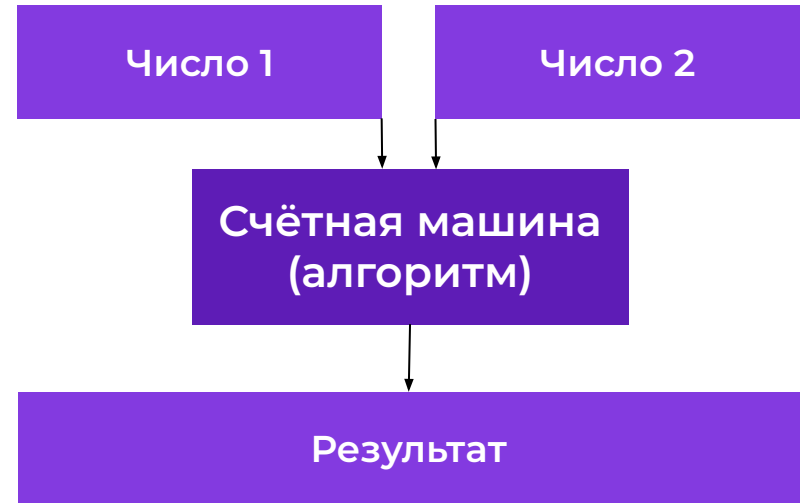
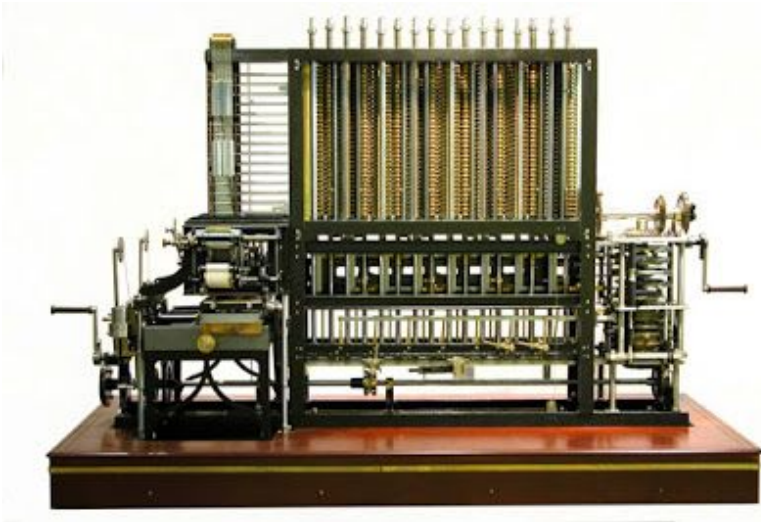
Как используются?



Фокус: угадаю день рождения



Счётная машина Ч. Бэббиджа

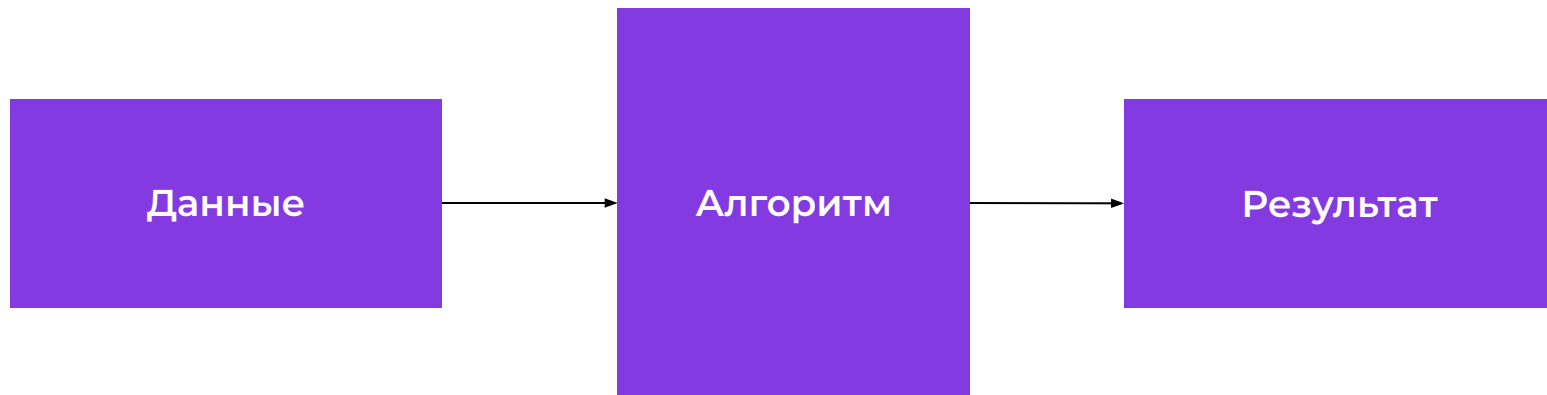


Сортировка багажа



*В программе всегда записан алгоритм.
В алгоритме порядок действий одинаковый.*

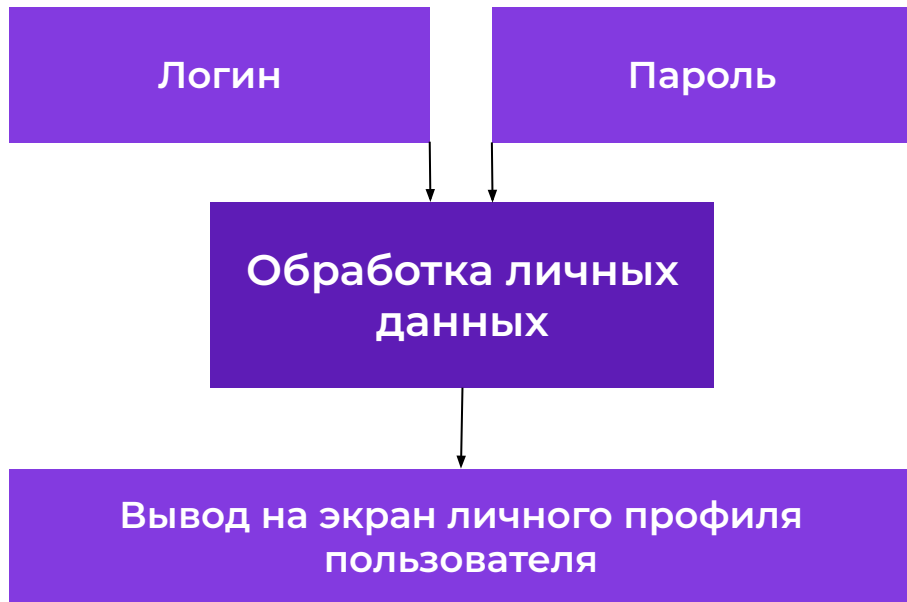
**Как же добиться
разных результатов?**



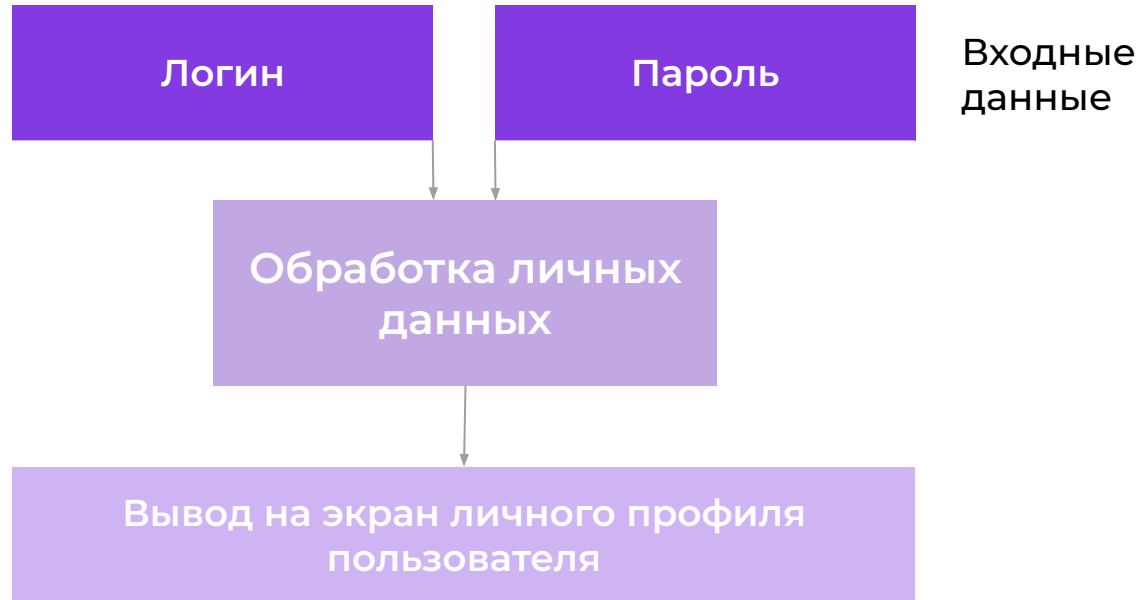
Данные могут меняться.

Вход на сайт

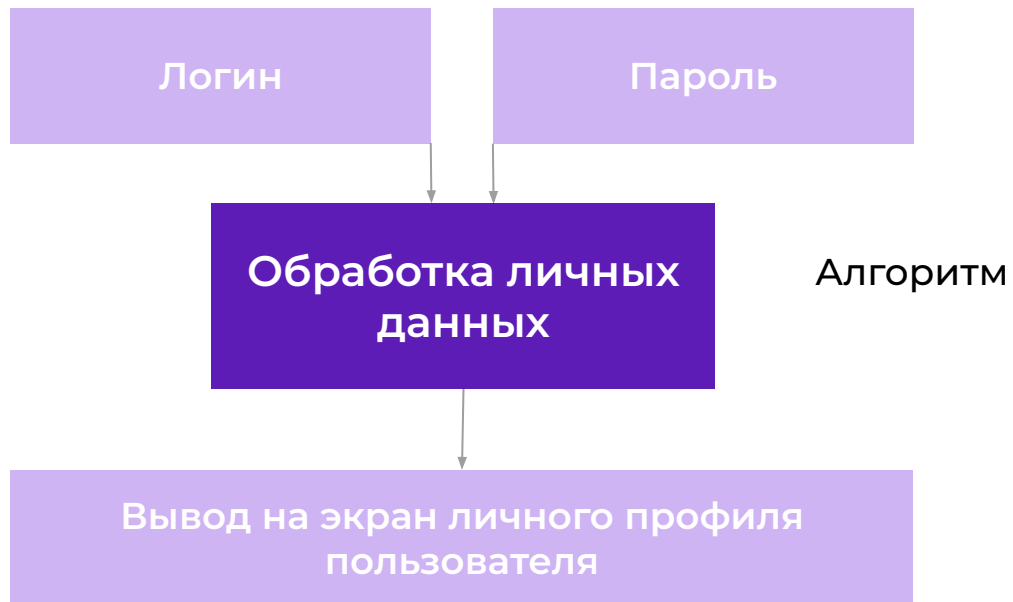
Зайдите
на платформу
со своими логином
и паролем.



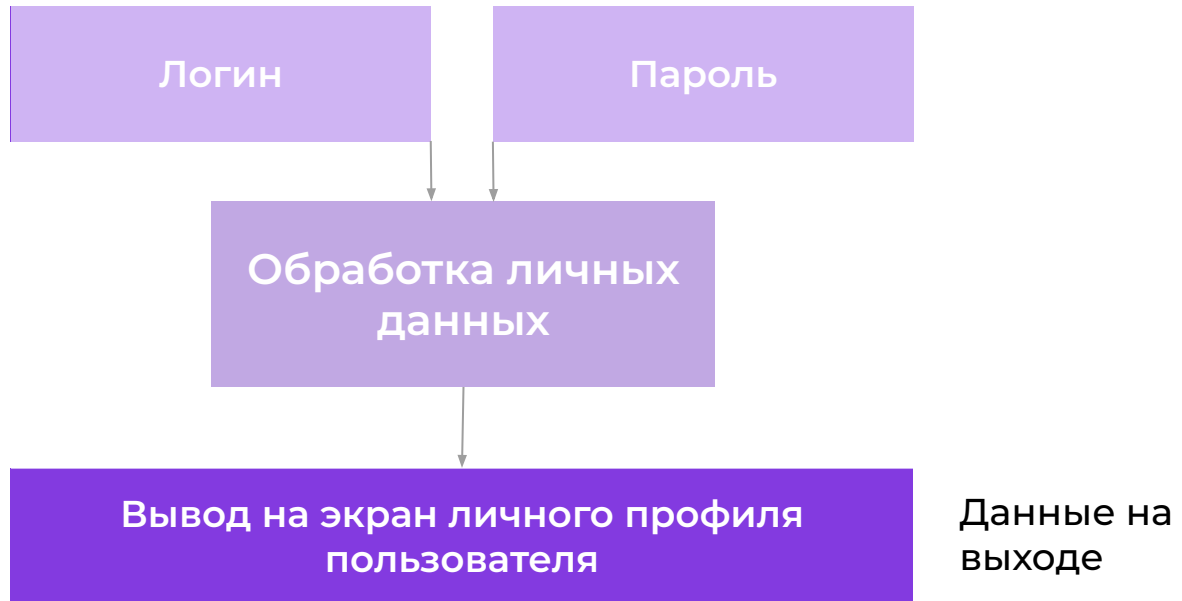
Вход на сайт



Вход на сайт



Вход на сайт



Переменные

нужны для работы с данными,
которые могут **меняться**
в вашей программе.

Выражения с числами и переменными

$2-1$

$a-b$

До того как использовать переменную, нужно:

- создать переменную,
- присвоить значение переменной.

Использование переменных

```
a = 2
```

```
b = 1
```

```
print(a-b)
```

Использование переменных

```
a = 2
```

```
b = 1
```

```
print(a-b)
```



Установка значения:
создать переменные
и присвоить им значения

Использование переменных

```
a = 2
```

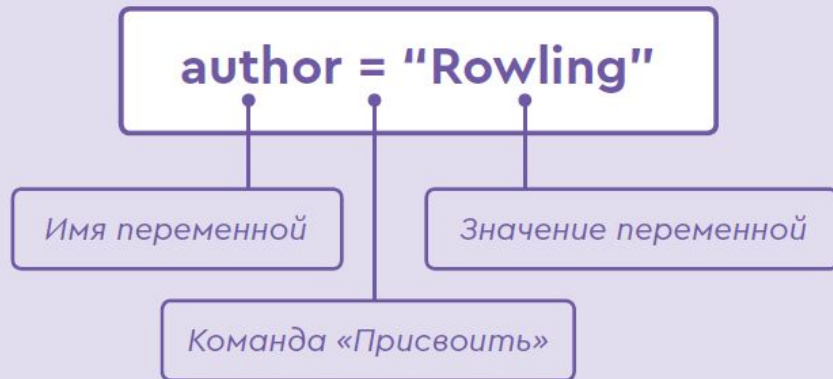
```
b = 1
```

```
print(a-b)
```



Алгоритм,
использующий переменные

Оператор присваивания



Важно: значение строки всегда указывается в кавычках.

Резюме

Алгоритмы работают с **данными**.

Резюме

Любая команда алгоритма может применяться к **данным, значение которых** программисту **неизвестно**, оно будет известно на момент работы этой команды.

Резюме

Алгоритм может оперировать не только **значениями**, а ещё и **именами данных**.

Переменная

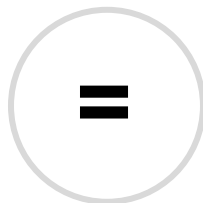
**это элемент данных,
имеющий своё имя.**

Резюме

Чтобы использовать переменную, нужно её **создать** (придумать имя) и **задать начальное значение**.

Резюме

Оператор присваивания



задает значение переменной.

Теория

```
name = "Robert"
```

```
hi = "Hello"
```

```
s = name + ", " + hi
```

```
print(s)
```

```
name = "Jon"
```

```
s = hi + ", " + name
```

```
print(s)
```


Резюме

Оператор присваивания
создаёт новую переменную,
если встречает новое имя.

```
name = "Robert"  
hi = "Hello"  
s = name + ", " + hi  
print(s)  
name = "Jon"  
s = hi + ", " + name  
print(s)
```

Резюме

Использовать переменные в вычислении выражений **можно только** в том случае, **если** эти переменные уже **созданы** и **имеют** какие-то **значения**.

```
name = "Robert"
```

```
hi = "Hello"
```

```
s = name + ", " + hi
```

```
print(s)
```

```
name = "Jon"
```

```
s = hi + ", " + name
```

```
print(s)
```

Пример 1

```
print("Привет!")
```

```
print("Привет!")
```

```
print("Привет!")
```

Пример 1

```
h = "Привет!"
```

```
print(h)
```

```
print(h)
```

```
print(h)
```

Пример 1

```
h = "Привет!"
```

```
print(h)
```

```
print(h)
```

```
print(h)
```



Установка значения

Пример 1

```
h = "Привет!"
```

```
print(h)
```

```
print(h)
```

```
print(h))
```



Алгоритм, использующий
переменные

Пример 2

```
a = 1
```

```
b = 2
```

```
c = a + b
```

```
print(c)
```



Установка значения



Алгоритм, использующий
переменные

Пример 2

```
a = 1
```

```
b = 2
```

```
c = a + b
```

```
print(c)
```



Выведет 3

Пример 3

```
a = "Hello, "
```

```
b = "world!"
```

```
c = a + b
```

```
print(c)
```



Установка значения



Алгоритм, использующий
переменные

Пример 3

```
a = "Hello, "
```

```
b = "world!"
```

```
c = a + b
```

```
print(c)
```



Выведет "Hello, world!"

Пример 4

$$a = 1$$

$$b = 2$$

$$c = a + b$$

$$d = b - a$$

Пример 4

a = 1

b = 2

c = a + b

d = b - a



Ничего не выведет

Пример 5

```
a = 1
```

```
b = 2
```

```
print(a * b)
```

```
print(b / a)
```

Пример 5

```
a = 1
```

```
b = 2
```

```
print(a * b)
```

```
print(b / a)
```



Операторы умножения и деления

Переменные могут менять значения

```
a = 1
```

```
b = 2
```

Установка значения

```
a = "Hello, "
```

```
b = "world!"
```

```
c = a + b
```

```
print(c)
```

Алгоритм

```
c = a + b
```

```
print(c)
```

Имена переменных

name

point1

code

airport

my_sum

login

author7

new_store

point2

password

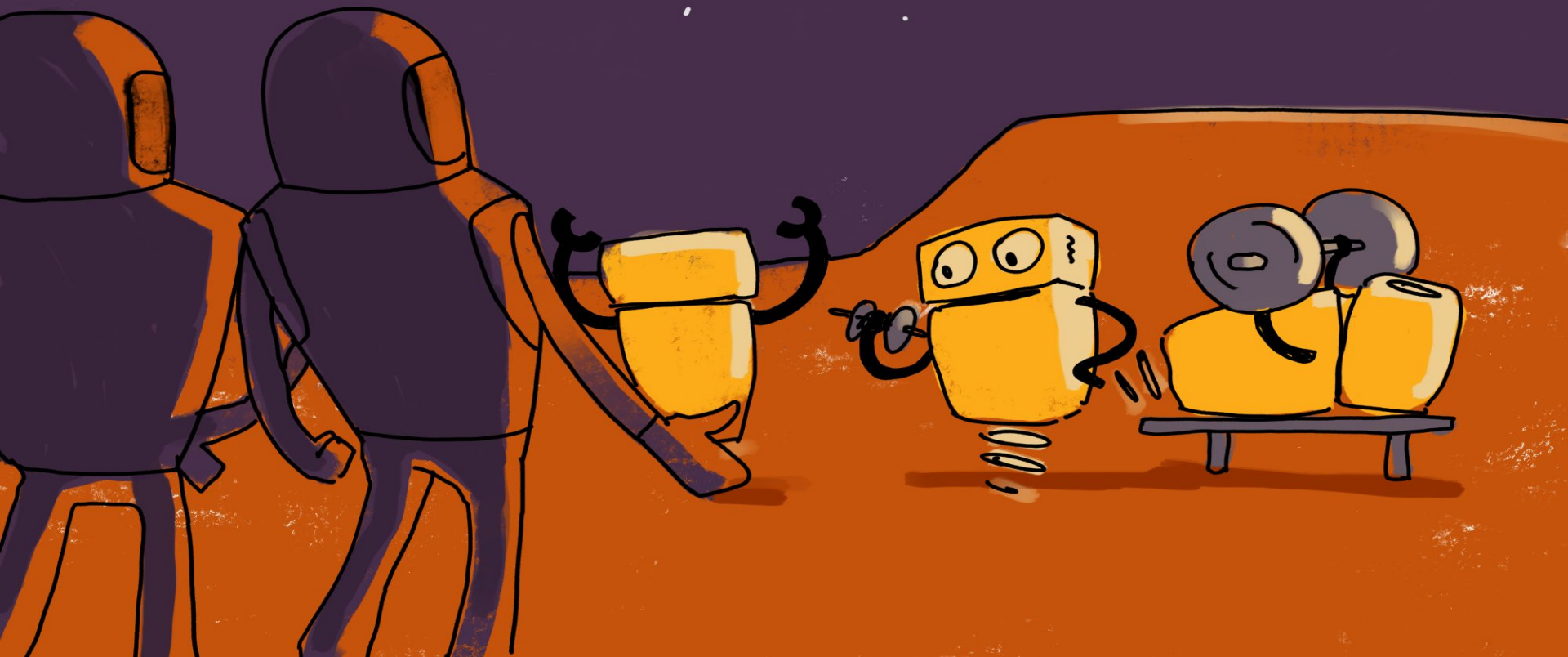
Платформа

**Переменные:
тренируемся печатать.**

**Переменные: первые
программы.**

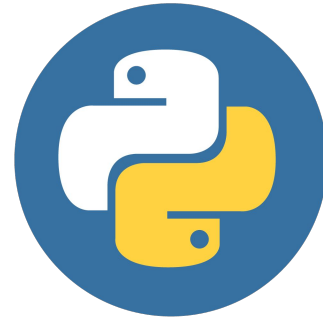


Время для разминки



Функция `input()`

Способ ввода данных



Введение

Программа «Повтор строки»

**Как менять вводимую
информацию,
не изменяя саму
программу?**

**Предложить
человеку вводить
эти строки!**

Функция `input()`

```
input("Как тебя зовут?")
```

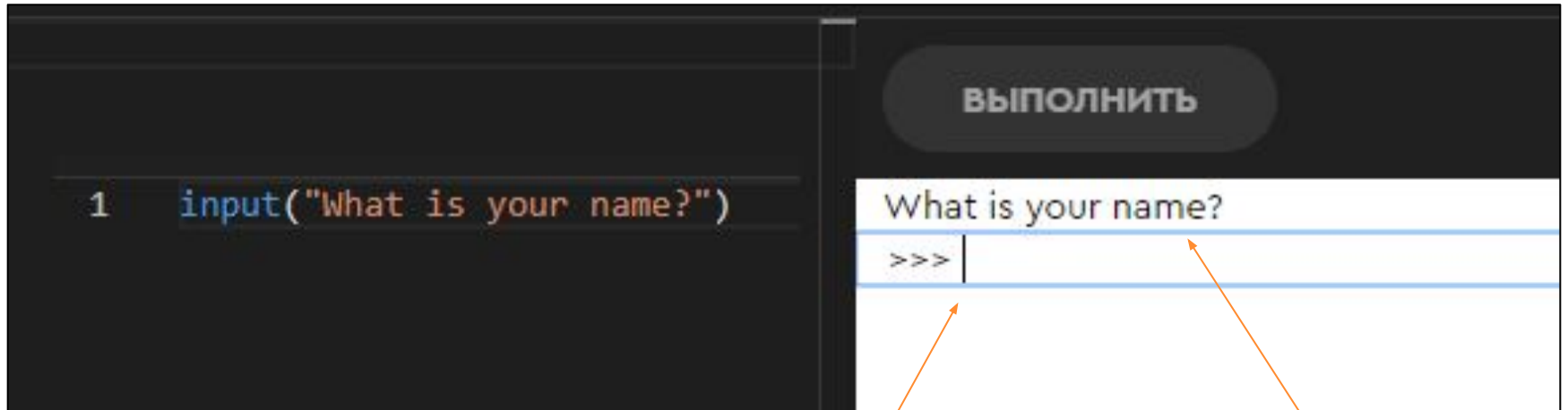
Функция ввода данных.



Параметр.



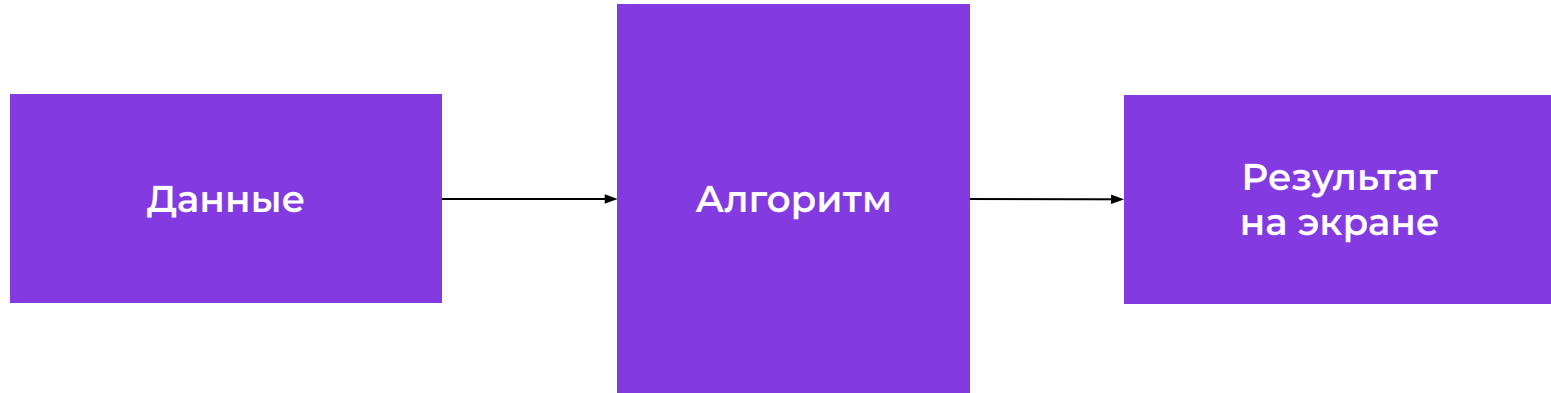
Функция `input()`



Курсор для ввода данных.

Параметр — это строка для вывода на экран.

Результат работы алгоритма



Пример

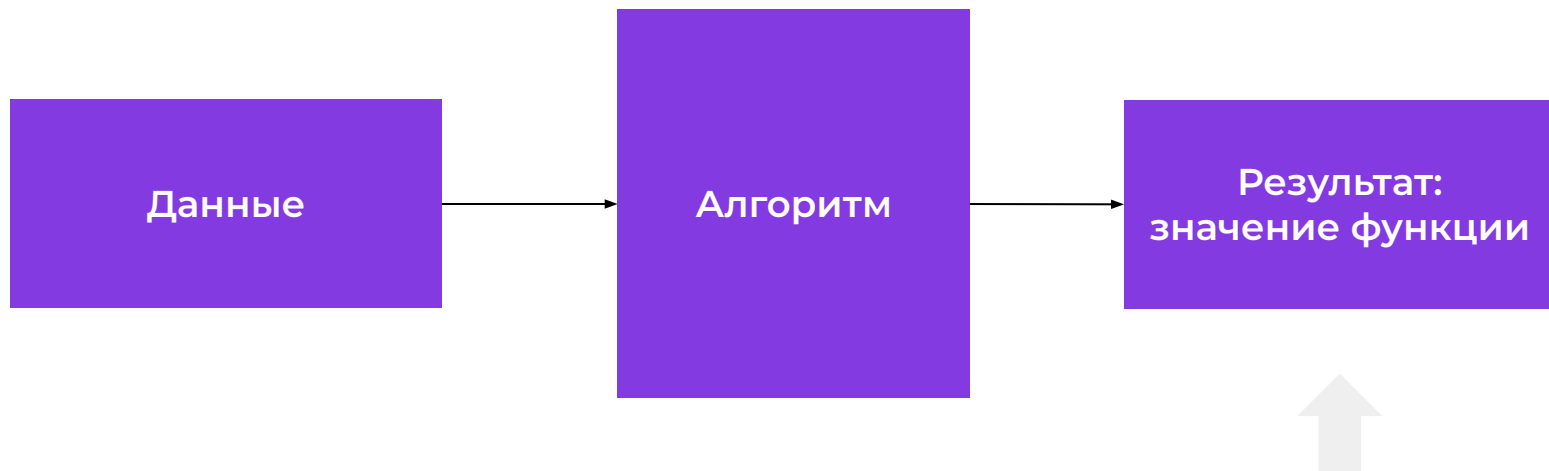
Шифр(25, 12)

результат: число 2762

Фокус(2762)

результат: число 2512

Результат работы алгоритма



Например, подсчёт числа для нашего фокуса. Результат сохранится в памяти, и программа может его использовать дальше.

Используем значение функции:

`shifr(25, 12) - 250`

`n = shifr(6,6)`

Использование функции `input()`

*Переменная для
хранения ответа.*

```
name = input("Как тебя зовут?")
```

Функция ввода данных.

Параметр.

**Как написать
программу, которая
запросит строчку у
пользователя и
напечатает её три раза?**

Пример 1

```
s = input("Что повторять?")  
print(s)  
print(s)  
print(s)
```

**Как написать
программу, которая
запросит у
пользователя три
строки и напечатает их?**

Обсуждение

Пример 2

```
s1 = input("1?")
```

```
s2 = input("2?")
```

```
s3 = input("3?")
```

```
print(s1)
```

```
print(s2)
```

```
print(s3)
```



Как написать программу, которая ведёт диалог?

Обсуждение

Пример 3

```
music = input("Какую музыку ты любишь?")  
print("Ого, мне тоже нравится слушать " +  
music + "!")  
print("У нас похожие вкусы!")
```

Пример 4

```
1 s = input("В каком городе ты живешь?")
2 answer = "О, " + s + "!"
3 print(answer)
```

В каком городе ты живешь?

>>>

```
1 s = input("В каком городе ты живешь?")
2 answer = "О, " + s + "!"
3 print(answer)
```

В каком городе ты живешь?

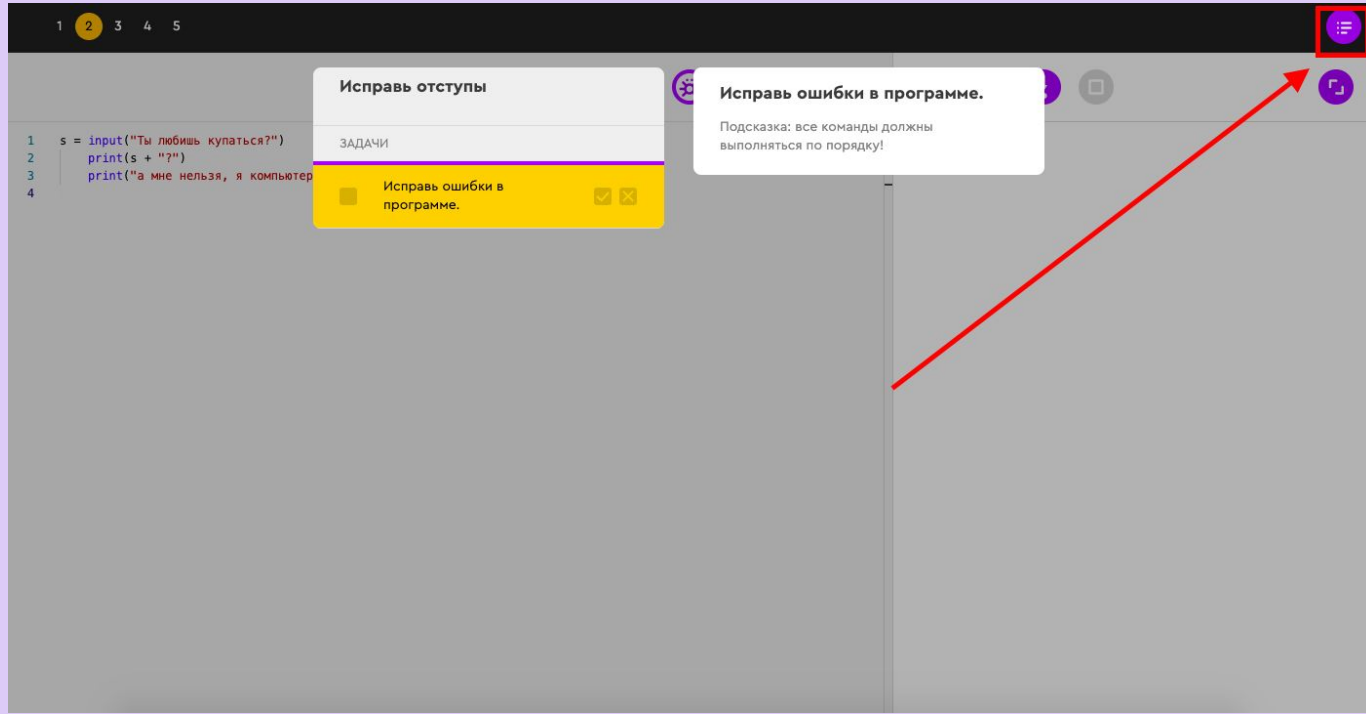
>>> Баку

О, Баку!

Заходим на платформу



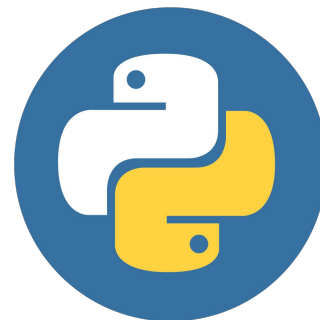
В некоторых уровнях есть подсказки!



Задание на платформе

Какие диалоги у вас получились?

Покажите свои программы друг другу.



Поиграйте с программой вашего соседа,
отвечая на её вопросы.

Введение

**Чем отличается цель от
мечты?**

Обсуждение

**Когда есть очевидная
планка,
мы всегда способны на большее!**

Цели — это планки:

**они позволяют реализовывать
наши мечты.**

Видео про постановку целей. Вариант 1.



Видео про постановку целей. Вариант 2.



Видео про постановку целей. Вариант 3.



SMART

S (*Specific*)
— Конкретно.

M (*Measurable*)
— Измеримо.

A (*Achievable*)
— Достижимо.

R (*Relevant*)
— Релевантно (важно конкретно для вас).

T (*Time*)
— Время.



**Что можно сделать
дома?**

Выполни дома задания 1 и 2 из рабочей тетради (стр 7-9).

ЗАДАНИЕ № 1.

Что напечатают эти программы?

```
name = "Amir"
print("Hello")
print(name)
```

```
name = "Amir"
print("Hello")
print("name")
```

ЗАДАНИЕ № 2.

Программа должна поздороваться с человеком, назвав его по имени. Напиши, какие ошибки допущены в предложенных вариантах

```
a = input("Как тебя зовут?")
b = "hello"
c = print(a)
```


```
a = input("Как тебя зовут?")
b = hello
print(b)
print(a)
```

Домашнее задание

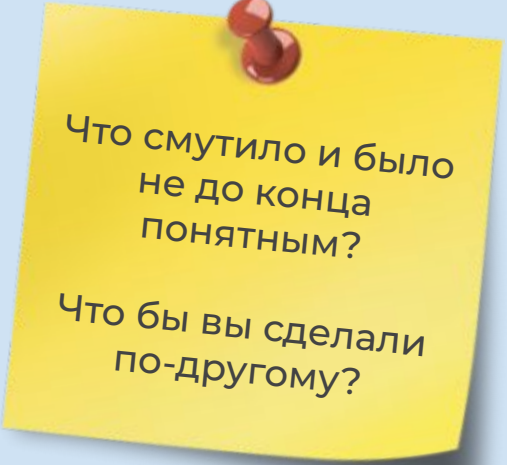
Итог урока



Как прошел урок

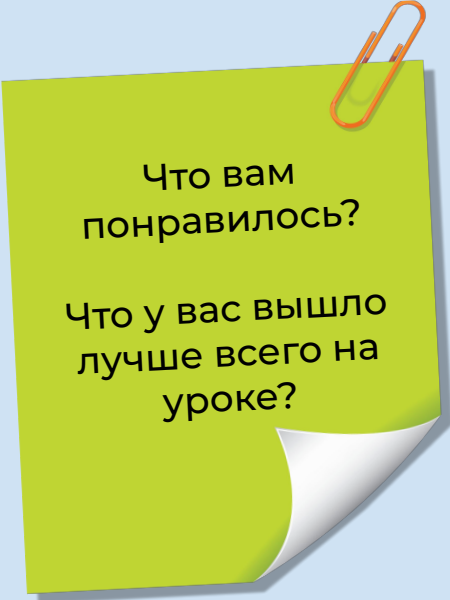


Что было сложным?



Что смутило и было не до конца понятным?

Что бы вы сделали по-другому?



Что вам понравилось?

Что у вас вышло лучше всего на уроке?

Что мы узнали на уроке?

- Что такое переменная?
- Как задается значение переменной?
- Когда создается новая переменная?
- Что такое значение функции?
- Какое значение возвращает функция `input()`?
- Какой параметр передается функции `input()`?

**До встречи на
следующем уроке**

Спасибо!

