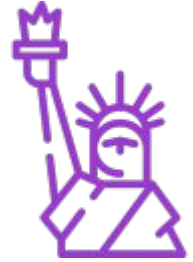


**Модуль 3. Урок 2.**  
**Модули.**

алгоритмика

**Повторим**

# Сегодня на занятии:



- Модули — создавай да применяй!
- Уже созданные интересные модули и их возможности.

# **Демонстрация**

**(функция шифрования с прошлого урока)**

**Как использовать  
функцию в другой  
программе, не  
копируя код?**

# Модуль —

это файл, в котором сохранена программа.

# Что необходимо, чтобы создать модуль:

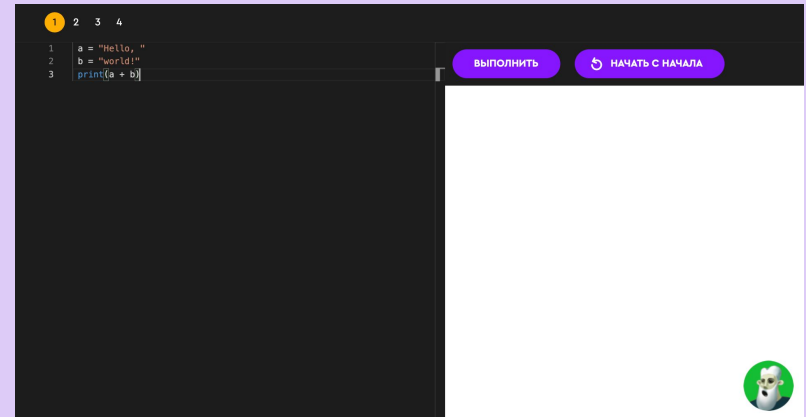
- 1) Сохранить в файл программу, в которой записаны необходимые нам функции.
- 2) Подключить этот файл в нужной программе.

# Заходим на платформу





# Создание модулей

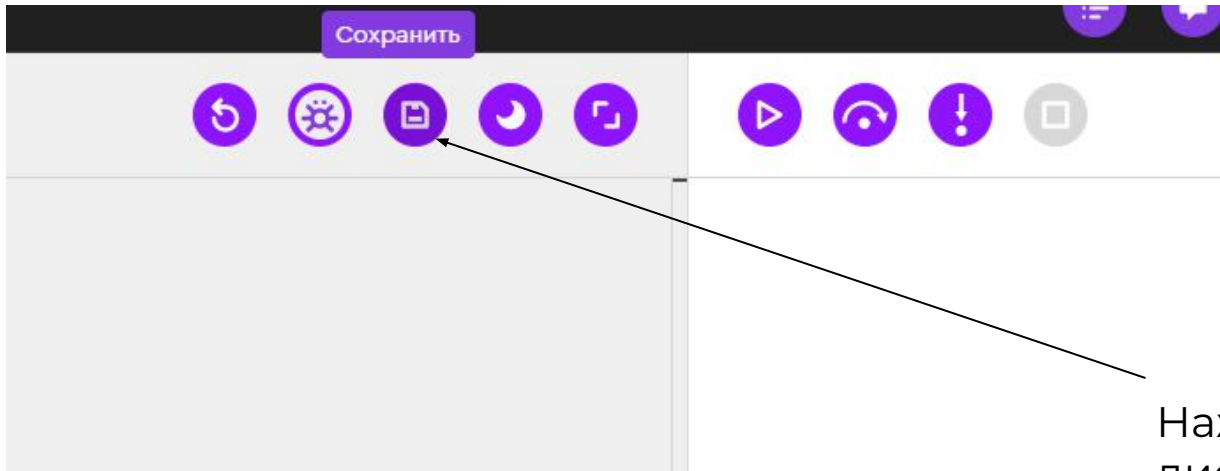


**Задание на платформе**

# Задание №1

**(функция шифрования с прошлого урока)**

# Сохранение программы в модуль



Нажимаем на значок дискеты (сохранить)

# Сохранение программы в модуль

Введи имя модуля

OK Отмена

Вводим имя модуля

# Внимание!!!

**Имя** — это важная часть, так как именно с помощью него впоследствии мы будем подключать и использовать модуль в других программах. Поэтому имя должно быть однозначно понятным.

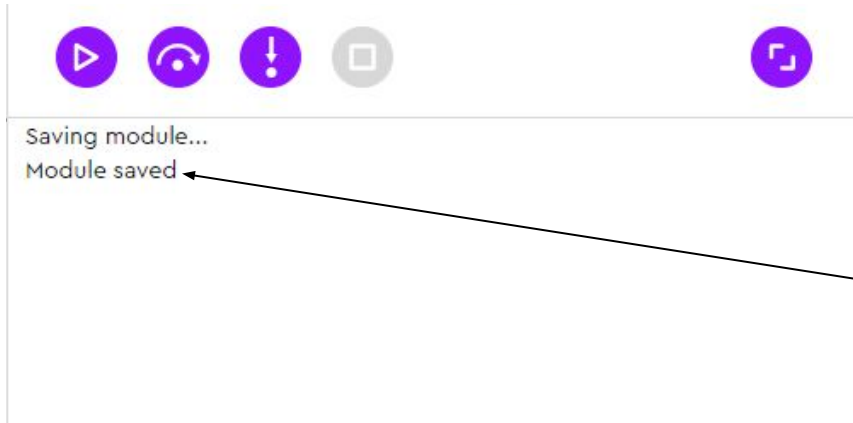
# Сохранение программы в модуль

Введи имя модуля

**OK** Отмена

После того, как мы записали имя модуля, нажимаем кнопку «OK»

# Сохранение программы в модуль



Надпись, которая гласит, что модуль успешно сохранён

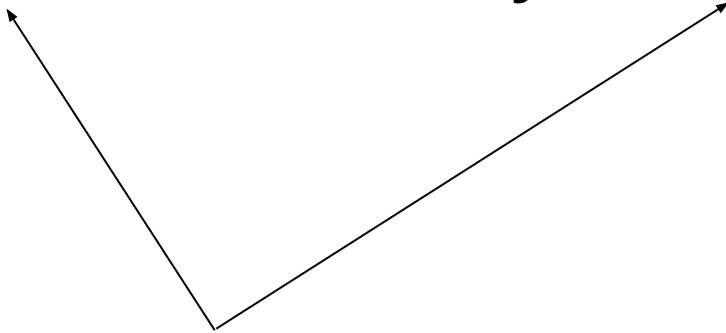
# **Задание №2**

**(функция без модуля)**



# Конструкция для подключения модуля

`from` имя модуля `import` имя функции



Ключевые слова

# Подключение encoder модуля в программу

```
1  from encoder import flip_out
2  from encoder import cut_out
```

Ключевые слова

Имя модуля

Имя функции

**Теория**

# Подключения модуля целиком

`from` ИМЯ модуля `import` \*



Если необходимо подключить все имеющиеся функции, то есть весь модуль целиком, то после слова `import` вместо имени конкретной функции ставится символ `*`.

# Итог первой половины урока



алгоритмика

**Давайте отдохнём!**

алгоритмика

# Интересные модули в Python

алгоритмика

# Модуль turtle

# **Демонстрация**

**(модуль черепашки)**



# Функции модуля turtle (часть 1)

Имя функции	Принцип работы
<code>forward(100)</code>	Движение вперёд на заданное в скобках количество пикселей.
<code>left(90)</code>	Поворот налево на заданное в скобках количество градусов.
<code>right(90)</code>	Поворот направо на заданное в скобках количество градусов.
<code>penup()</code>	Поднять перо, чтобы черепашка не оставляла след.
<code>pendown()</code>	Опустить перо, чтобы черепашка оставляла след.
<code>circle(10)</code>	Нарисовать окружность, радиус которой указан в скобках.

# Функции модуля turtle (часть 2)

<code>width(5)</code>	Установить ширину линии в соответствии с цифрой, указанной в скобках.
<code>color("green")</code>	Установить цвет линии, указанный в скобках в двойных кавычках.
<code>goto(x, y)</code>	Переместить черепашку в координаты x, y, указанные в скобках
<code>begin_fill()</code>	Команда для закрашивания замкнутой фигуры в цвет, указанный в функции <code>color()</code> . Ставится перед началом рисования фигуры.
<code>end_fill()</code>	Команда для закрашивания замкнутой фигуры в цвет, указанный в функции <code>color()</code> . Ставится в конце рисования фигуры.
<code>exitonclick()</code>	Команда, не дающая окну свернуться и пропасть рисунку.

# Демонстрация

**(квадрат)**

# Демонстрация

**(круг)**

# Демонстрация

(светофор)

алгоритмика

# Модуль `time`

# Функции модуля `time`

Имя функции	Принцип работы
<code>ctime</code> ([сек])	Преобразует время, выраженное в секундах с начала эпохи в строку вида "Thu Sep 15 13:33:01 2019".
<code>sleep</code> (сек)	Приостановить выполнение программы на заданное количество секунд.
<code>strptime</code> (строка, [формат])	Разбор строки, представляющей время в соответствии с форматом. Возвращаемое значение <code>struct_time</code> . Формат по умолчанию: "%a %b %d %H:%M:%S %Y".
<code>time</code> ()	Время, выраженное в секундах с начала эпохи.

# Демонстрация

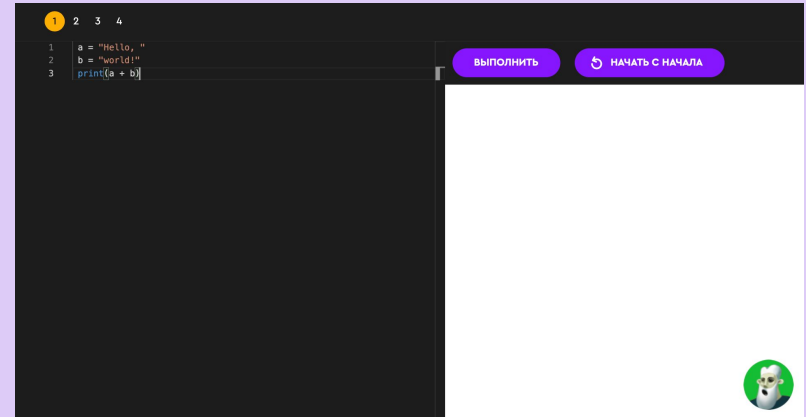
**(таймер)**



# Заходим на платформу



# Интересные модули в Python



The image shows a screenshot of a Python code editor interface. The editor has a dark background and contains the following code:

```
1 a = "Hello, "  
2 b = "world!"  
3 print(a + b)
```

At the top of the editor, there are four numbered tabs (1, 2, 3, 4) and a yellow circle icon. Below the code, there are two buttons: "ВЫПОЛНИТЬ" (Execute) and "НАЧАТЬ С НАЧАЛА" (Run from Start). The right side of the editor is currently blank, suggesting the output of the code execution is not yet visible. In the bottom right corner, there is a small circular icon of a white dog's head.

**Задание на платформе**

алгоритмика

**Как прошло занятие?**

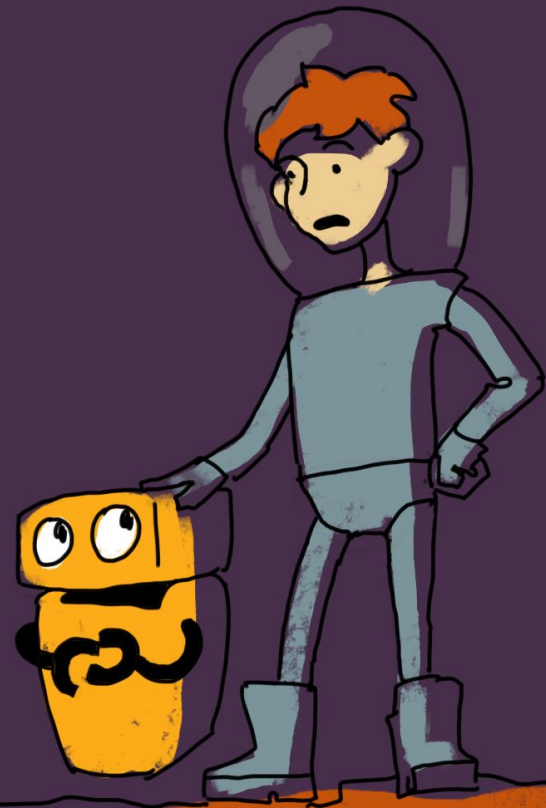
# Проверь себя

- Что такое модуль?
- Куда нажать, чтобы сохранить модуль?
- Каким должно быть имя модуля?
- Как подключить конкретную функцию из модуля и весь модуль?
- Какие модули вы сегодня узнали?

# На следующем занятии:

- Lvl up — изучаем классы!
- Научимся делить мир на объекты

**До встречи!**



# Модуль `math` —

модуль, предоставляющий функционал для работы с числами.

# Функции модуля math

Имя функции	Принцип работы
<b>ceil</b> (X)	Округление до ближайшего большего числа.
<b>fabs</b> (X)	Модуль X.
<b>factorial</b> (X)	Факториал числа X.
<b>floor</b> (X)	Округление вниз.
<b>pi</b>	Возвращает число пи



# Модуль `random` —

модуль, предоставляющий функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

# Функции модуля random

Имя функции	Принцип работы
<b>randint</b> (A, B)	Случайное целое число N, $A \leq N \leq B$ .
<b>random</b> ()	Случайное число от 0 до 1.
<b>randrange</b> (start, stop, step)	Возвращает случайно выбранное число из последовательности, где start – наименьшее число, stop - наибольшее число, step – размер шага). Например, команда <code>print(randrange(5, 10, 2))</code> будет случайно выдавать одно из следующих чисел: 5, 7, 9.
<b>choice</b> (последовательность)	Случайный элемент непустой последовательности (например, списка).

# Этапы игры «Угадай число»

## Правила игры

Рассказ о  
правилах игры.

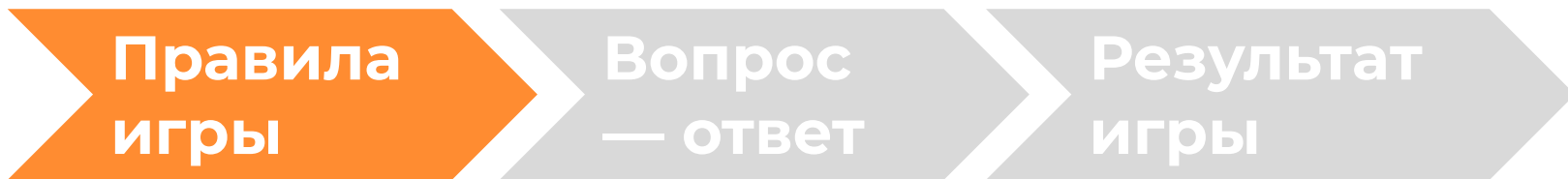
## Вопрос — ответ

Игра: генерация  
случайного числа,  
ввод числа  
пользователем,  
программа  
отвечает  
«больше» или  
«меньше».

## Результат игры

Программа  
говорит, что  
число угадано и  
поздравляет.

# Этапы игры «Угадай число»



Из чего состоит этот этап?

# Этапы игры «Угадай число»

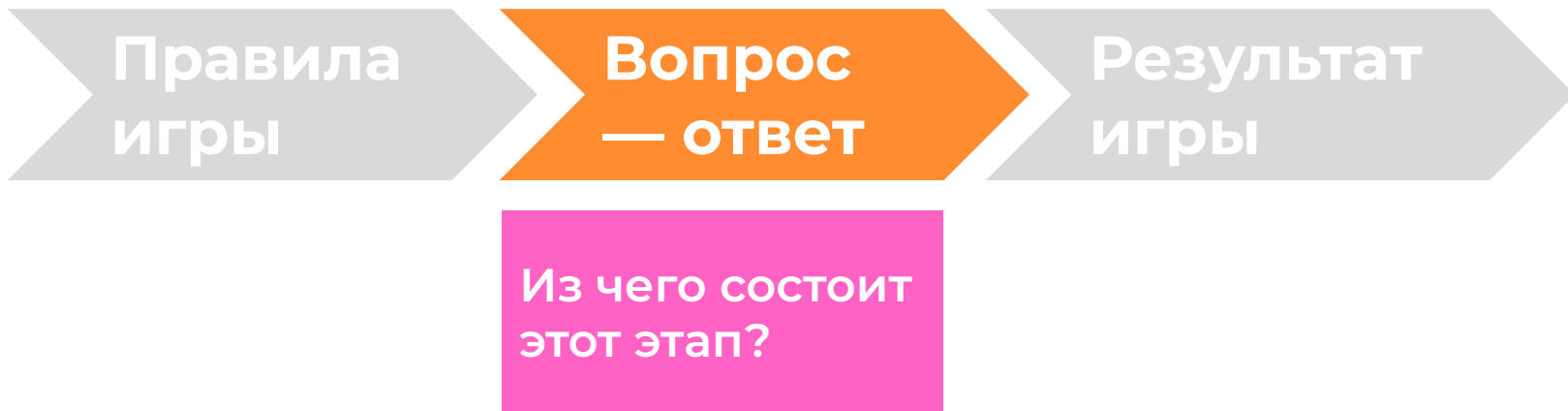
Правила  
игры

Вопрос  
— ответ

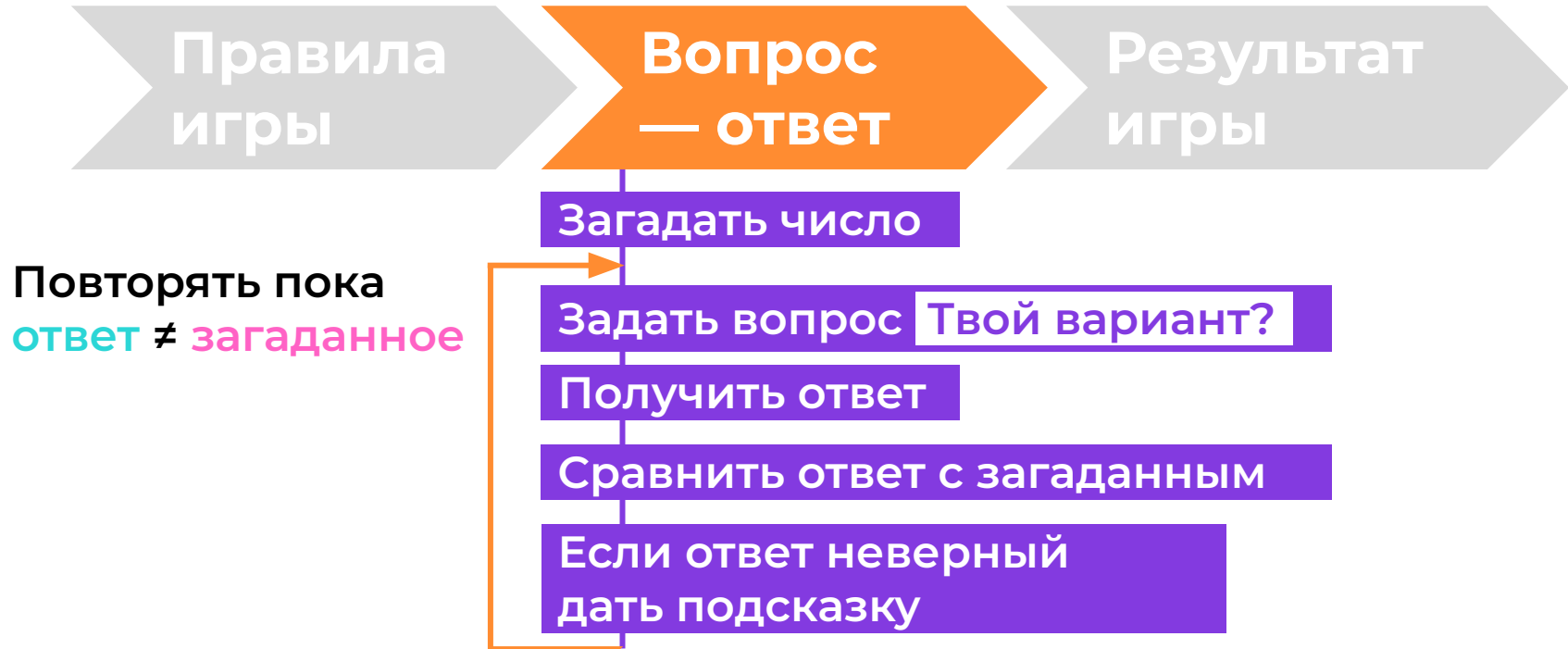
Результат  
игры

```
print("Я загадал число от 1  
до 100. Угадай его!")
```

# Этапы игры «Угадай число»



# Этапы игры «Угадай число»



# Этапы игры «Угадай число»

Правила  
игры

Вопрос  
— ответ

Результат  
игры

Из чего состоит  
этот этап?



# Этапы игры «Угадай число»

Правила  
игры

Вопрос  
— ответ

Результат  
игры

```
print("Ты угадал! Я загадал  
" + str(a))
```