

«Адаптация перевода Системы Наглядного  
Анимационного Программирования для  
внедрения её в школьную программу обучения  
начальных классов»



ХСнап!



ХСнап!  
Справочник

- Основной идеей проекта является применение Снап! в образовательных и практических целях в качестве системы для разработки и замены Flash-уроков из приложений к учебникам младших классов.
- Справочник с начальными сведениями о работе с системой Снап!, который был сделан в виде сайта, может помочь начинающим пользователям и учителям.
- Конечная цель — формат, когда дети 4–5 класса могут помогать учителям делать обучающие материалы для более младших классов.



- Полное прекращение поддержки Flash от Adobe является очень весомой причиной поиска альтернатив
- Снап! может стать полноценной заменой. Обучающие программы не уступают в интерактивности и превосходят в удобстве создания.



λСнап!





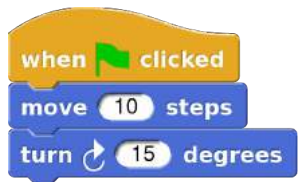
- Проблема Логомиров и Перволого - они работают на пиктограммах.
- Все алгоритмы в Снап! строятся из слов, записанных в блоки, что позволяет приблизиться к синтаксису и структуре написания кода на продвинутых языках.

```
при нажатии на   
очистить всё  
перейти в точку x: 75 y: -50  
указывать в направлении 90  
непрерывно  
  повторять 3  
    ждать 1 сек.  
    опустить перо  
    повернуть  на 90 градусов  
    передвинуть на 50 шагов  
  передвинуть на 150 шагов
```

```
при нажатии на   
передвинуть на 10 шагов  
повернуть  на 15 градусов
```



- Scratch обладает лицензией MIT
- Я не хотел, чтобы мои изменения в будущем могли перестать быть свободными.



- Требовалось перевести текст.
- В основном весь текст находится в блоках с командами и в разных меню.
- Часто с выходом обновлений добавляются новые возможности, блоки и, соответственно, новый текст.



**if on edge, bounce**

**forever** keep doing this:  
**move 10 steps** move 10 steps  
**if on edge, bounce** if at the edge of stage, then turn around

**if on edge, bounce** checks to see if this sprite has reached the edge--and if it has, it turns away from the edge. (And then is ready for the next move.)

**список** Передает список с вложенными элементами

Используй стрелки для увеличения количества ячеек для ввода

Элемент списка может быть чем угодно: текстом, цифрой, блоком, скриптом и даже другим списком

Создать пустой список:

Ты можешь положить список в ячейку других блоков:

Также в ячейку специальных блоков, которые предназначены только для списков:

**если на границе, то развернуться**

**навсегда** бесконечно повторять это:  
**передвинуть на 10 шагов** переместиться на 10 шагов вперед  
**если на границе, то развернуться** развернуть Спрайт на границе сцены

**если на границе, то развернуться** Этот блок проверяет, коснулся ли Спрайт границы сцены. Если да, то он развернется в обратную сторону и продолжит выполнять действия

**list** Reports a newly created list with the given items.

Use the arrows to change the number of inputs.

The list items can be anything: text, numbers, blocks, scripts, and even other lists!

Make an empty list.

You can put a list in a rectangular input slot:

and also in a lists-only input slot:

- Переводу подлежали вспомогательные картинки.
- На данный момент переведены не все картинки, но те, которые уже переведены, находятся на сервере, и я постепенно добавляю туда новые.

Безымянный

Движение Управление  
Внешность Сенсоры  
Звук Операторы  
Перо Переменные

Спрайт  
✓ двойной

Скрипты Костюмы Звук

передвинуть на 10 шагов  
повернуть на 15 градусов  
повернуть на 15 градусов  
указывать в направлении 90  
указывать на курсор мыши  
перейти в точку x: 0 y: 0  
перейти в точку случайная позиция  
скользить 1 сек x x 0 y y 0  
изменить x на 10  
установить x 0  
изменить y на 10  
установить y 0  
на границе развернуться  
координаты x  
координаты y  
направление  
Новый блок

Сцена



# ХС Нап!


Справочник

- В основном этот справочник предназначен для пользователей, которые впервые пользуются Снап!

Снап! Что такое Снап! Руководство Возможности Уроки Загрузка

## Что такое Снап!


**Снап!** - система наглядного анимационного программирования, ориентированная на обучение пользователей, которые являются начинающими в сфере программирования. Нажав на первую кнопку в меню, вы перейдете в Снап!. Благодаря русскоязычной адаптации и русскоязычному сайту, Снап! стал более доступным для русскоязычных пользователей. Главное преимущество Снап! - простота и понятность. Все программы строятся из блоков с кодами. Нужно лишь присоединить их друг к другу. С помощью Снап! можно делать многое, начиная от простых математических программ и заканчивая настоящими играми. На англоязычном веб-сайте программы Снап! ( [http://www.snaps.ru](#) ) есть огромное количество уже готовых работ, которые можно скачать абсолютно бесплатно. Для ознакомления с интерфейсом нажмите на "Руководство"



## Возможности

### Создание блока

В Снап! имеется возможность создать свой собственный блок. Он может выполнять роль программы, которая обрабатывает введенные данные. Собственный блок содержит в себе другие блоки, которые в сумме создают код. Таким образом один блок может делать почти что угодно. Создать собственный блок можно нажатием на "+", либо нажатием на "новый блок". После нажатия на кнопку у вас откроется меню, в котором потребуется выбрать вид блока и его название. Блок "Команда" встраивается непосредственно в код. У этого блока есть характерные выпуклости и вогнутости похожие на пазл. К этому блоку можно присоединить другие блоки.

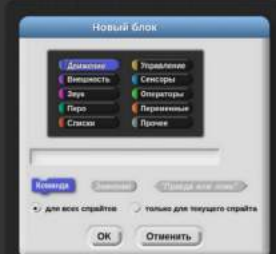


- 1. Поле с блоками.** Все блоки с кодами находятся здесь. Для удобства поле поделено на тематические секторы. Также присутствуют кнопка поиска блока и кнопка создания своего собственного блока. Также нажав правой кнопкой мыши на блок в этом поле, вы можете открыть справку (справки не все на русском языке) по блоку. В справке находится информация о блоке: что это за блок, что он делает и будут приведены примеры его использования.
- 2. Рабочее поле.** Рабочее поле поделено на три секции. В секции "Код" будет находиться весь выстроенный вами код. Чтобы переместить блок в секцию "Код", перетащи его из поля с блоками. Весь выстроенный вами код будет выполняться от лица Спрайта. В секции "Костюмы" находятся все добавленные костюмы для Спрайта. В секции "Звуки" находятся все добавленные звуки для Спрайта. Добавлять звуки и костюмы можно с помощью переноса файлов с компьютера, с помощью встроенных редакторов или с помощью встроенных библиотек. В верхней части поля находится информация о Спрайте: каким костюмом он обладает, каким именем и каким свойством передвижения.
- 3. Поле Вывода.** Это поле можно условно поделить на две секции. В секции "Сцена" находятся все результаты ваших действий: изменение фона Сцены, добавление Спрайта и вывод вашего кода. Также к дополнению есть секция "Настройка", которая позволяет управлять фоном Сцены и Спрайтами на ней.
- 4. Верхнее поле.** В верхнем поле находятся (слева на право): кнопка дополнительной информации, меню файлов, меню "Облако", меню настроек, имя проекта, кнопка отображения шагов выполнения, кнопка отображения/увеличения/уменьшения сцены, кнопка полноэкранного режима, кнопка начала (если стоит специальный блок) выполнения кода, кнопка паузы выполнения кода Спрайта и кнопка остановки выполнения кода Спрайта.

## Возможности

### Создание блока

В Снап! имеется возможность создать свой собственный блок. Он может выполнять роль программы, которая обрабатывает введенные данные. Собственный блок содержит в себе другие блоки, которые в сумме создают код. Таким образом один блок может делать почти что угодно. Создать собственный блок можно нажатием на "+", либо нажатием на "новый блок". После нажатия на кнопку у вас откроется меню, в котором потребуется выбрать вид блока и его название. Блок "Команда" встраивается непосредственно в код. У этого блока есть характерные выпуклости и вогнутости похожие на пазл. К этому блоку можно присоединить другие блоки. Блок "Значение" обладает какими-либо данными. Это могут числа, строки или списки. Он имеет закругленную форму. Его можно встроить только в ячейки других блоков. Блок "Правда или ложь" имеет булево значение. Если внутри этого блока выполняется условие, то он будет иметь значение "правда". Блок обладает формой с острыми концами по краям. Его можно встроить только в ячейки других блоков.



### Урок по теме "Присваивание"

Первая и основная тема базового программирования - "Присваивание". Присваивание это оператор, позволяющий присвоить к переменной, которая стоит перед оператором, значение, которое стоит после оператора. Если вам нужно более подробное объяснение этой темы, нажмите на стрелку после абзаца. В этой главе будет написано про план, с помощью которого можно будет провести урок по теме "Присваивание".

В разных частях программы нередко попадают одни и те данные — строчки текста или числа. Чтобы каждый раз не писать огромные числа или строки, в языках программирования добавили константы. С помощью константы любому элементу данных — числу или строке — вы можете назначить удобное имя, а затем подставлять это имя вместо самих данных. Прежде, чем применить символическую константу, её надо объявить, то есть дать ей имя и значение. Чтобы объявить константу, перейдите в раздел переменные и нажмите на кнопку "объявить переменную". Как уже было сказано ранее, значением переменной может являться как число, так и строка.

Урок можно начать с объяснения - "Что такое Присваивание?". Далее, приступите к простому заданию. Задание будет заключаться в написании программы, которая будет приветствовать пользователя. [Здесь](#) можно скачать пример задания. Для закрепления пройденного материала предложите ученикам придумать собственное подобное задание и выполнить его самостоятельно. Далее, приступите вместе с учениками к выполнению усложненного задания, которое будет выполнять математическое сложение двух введенных переменных. [Здесь](#) можно скачать задание пример данного задания. После завершения работы, задайте ученикам домашнее задание. Примером может быть совмещение двух ранее сделанных работ - сначала компьютер приветствует пользователя, а потом спрашивает числа, которые будет складывать. Готовый код можно скачать [здесь](#). Если осталось время от урока, приступите к новой теме под названием "Оператор ветвления".

### Урок по теме "Циклы"

Циклы - последняя из трёх базовых тем программирования. Циклы позволяют многократно выполнять одни и те же задачи с помощью одной конструкции. Если вам нужно более подробное объяснение этой темы, нажмите на стрелку после абзаца. В этой главе будет написано про план, с помощью которого можно будет провести урок по теме "Циклы".

Урок можно начать с объяснения: "Что такое циклы, и для чего они нужны". В качестве пояснения выполните задание на построение несложной геометрической фигуры. Пример можно скачать [здесь](#). Для закрепления пройденного предложите ученикам выполнить самостоятельно построение простой фигуры. После выполненной самостоятельной работы, переходите к выполнению несложного задания, которое будет проверять знание таблицы умножения, и будет начинать цикл заново, если ответ на вопрос неправильный. [Здесь](#) можно скачать пример задания. После выполненной самостоятельной работы задайте ученикам домашнее задание. Примером может служить комбинация уже ранее сделанных заданий - при каждом неправильном ответе, алгоритм будет задавать вопрос заново. При правильном ответе алгоритм будет переходить на следующий вопрос и повторять ранее упомянутую конструкцию. Пример можно скачать [здесь](#).

### Урок по теме "Оператор ветвления"

Ещё одна из трёх базовых тем программирования - "Оператор ветвления". Ветвление обеспечивает выполнение команд при соблюдении или не соблюдении кого-либо условия. Если вам нужно более подробное объяснение этой темы, нажмите на стрелку после абзаца. В Снап! этот оператор находится в разделе "Управление". В этой главе будет написано про план, с помощью которого можно будет провести урок по теме "Ветвление".

Урок можно начать с объяснения - "Что такое оператор ветвления и где он используется". Далее, можно выполнить небольшое задание на знание простых математических вычислений, которое будет на практике демонстрировать как работает оператор ветвления. Готовое задание можно скачать [здесь](#). Далее, для закрепления пройденного, предложите ученикам придумать и выполнить своё собственное задание. После выполненной самостоятельной работы, переходите к заданию, которое является немного усложненной копией первой задачи - в нем добавлено еще одно условие после верного ответа на первый вопрос. [Здесь](#) можно скачать готовое задание. После завершения основной части урока, задайте ученикам домашнее задание. Примером может быть задание, которое совмещает алгоритмы ранее сделанных работ - после верного и неверного ответа будет следовать еще один вопрос. Задание можно скачать [здесь](#). Если осталось время от урока, переходите к новой теме под названием "Циклы".

Снап!

Что такое Снап!

Руководство

Возможности

Уроки

Загрузка

Загрузить файл: Обзор...  X

Отправить файл

Снап! - система наглядного обучения пользователей. Нажав на первую кнопку адаптации и русскоязычные пользователи. Главные строятся из блоков с которыми можно делать многое, настоящие игры. На ( скачать абсолютно бесплатно "Руководство"

Чтобы запустить урок:

1. Нажмите на кнопку выбора файла.
2. Добавьте файл двойным нажатием по нему (файл должен обязательно содержать расширение ".xml").
3. Нажмите кнопку "Отправить файл".

После отправки файла откроется новая вкладка с запущенным уроком. Для запуска урока повторно не нужно заходить на сайт и загружать его. В самом Снап! перейдите в меню файлов --> Открыть...--> Проекты. Нажмите на строку поиска два раза и введите название своего файла. После того как файл найден, кликните на него дважды и он откроется. Для запуска нажмите на кнопку с зеленым флажком. Для увеличения нажмите на кнопку рядом.



Создание блока

В Снап! имеется возможность создать свой собственный блок. Он может выполнять роль программы, которая обрабатывает введенные данные. Собственный блок содержит в себе другие блоки, которые в сумме создают код. Таким образом один блок может делать почти что угодно. Создать собственный блок можно нажатием на "+", либо нажатием на "новый блок". После нажатия на кнопку у вас откроется меню, в котором потребуется выбрать вид блока и его название. Блок "Команда" встраивается непосредственно в код. У этого блока есть характерные выпуклости и вогнутости похожие на пазл. К этому блоку можно присоединить другие блоки.



*λСнап!*



*λСнап!*  
Справочник

