



Программирование в Scratch

(для начинающих)

О.Н. Буртаева

Введение

Среда программирования Scratch позволяет детям создавать собственные интерактивные проекты: игры, мультфильмы, обучающие системы, комиксы и т.д.

Scratch - это новая среда программирования, которая создавалась специально для того, чтобы подростки 8 – 16 лет использовали его самостоятельно. Эта технологическая среда позволяет им выразить себя в компьютерном творчестве.

В основе Scratch традиции языка Лого, а также языка программирования, использующегося в конструкторах Лего. В среде Скретч пользователь из отдельных блоков собирает свою программу. Простая форма позволяет даже самым маленьким приобщаться к программированию, превращая обучение в увлекательную игру. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования настолько прост и доступен, что Scratch рассматривается в качестве средства обучения не только старших, но и младших школьников.

Когда ученики создают проекты в Скретч, они осваивают множество навыков XXI века:

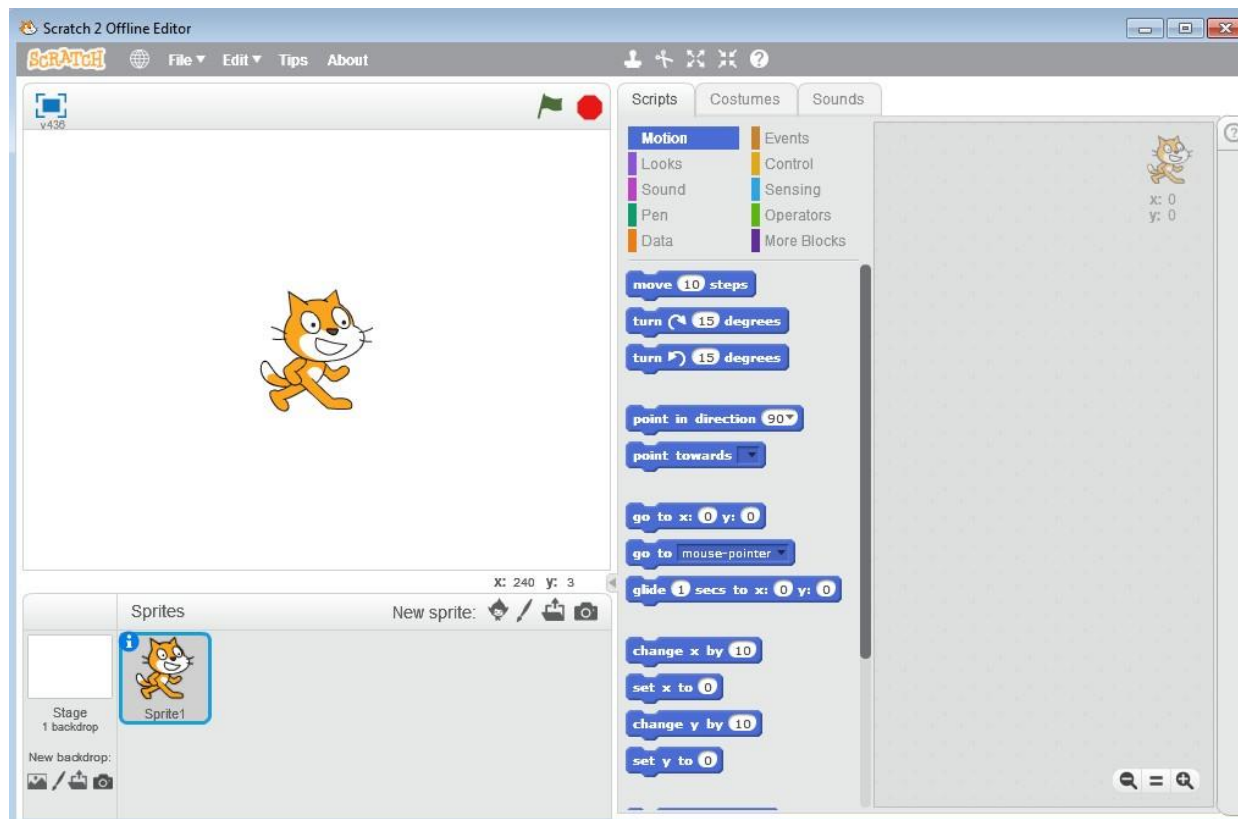
- творческое мышление,
- предметное общение,
- системный анализ,
- беглое использование технологий,
- эффективное взаимодействие,
- проектирование,
- постоянное обучение.

Одним из принципиальных достоинств данной среды является то, что она является свободно распространяемым программным продуктом, таким образом, любое образовательное учреждение может скачать программу из Интернет, и приступить к непосредственному изучению и работе в новой среде программирования.

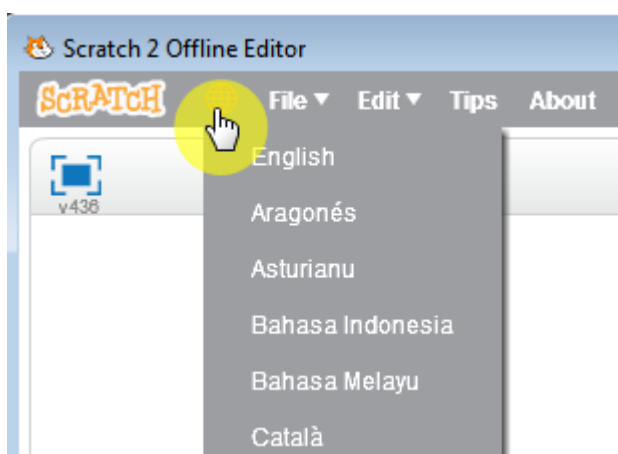
Знакомство с интерфейсом Scratch 2.

Обзор интерфейса.

При первом запуске программы Scratch, вы увидите, что ее интерфейс на английском языке.



Переключите интерфейс на русский язык. Для этого нажмите на глобус. Откроется список доступных языков, осталось найти в этом списке русский.



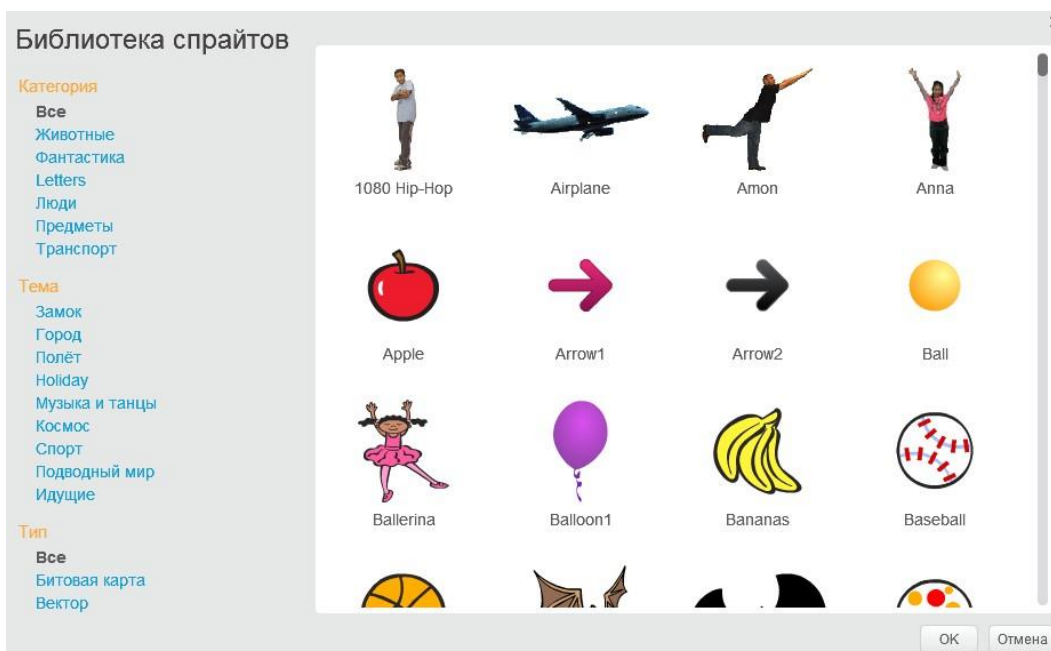
Редактор Scratch состоит из нескольких крупных частей: сцена, список спрайтов, палитра блоков и область скриптов. Спрайты – это объекты, которыми можно управлять с помощью блоков. Обычно это игровые персонажи и герои мультфильмов. При запуске редактора в проекте всегда есть один персонаж – рыжий Кот, он же является символом Скретч.

Дважды кликните на синем блоке **идти 10 шагов** – Кот переместится вправо. Для того, чтобы сделать полноценную программу, блоки нужно переместить в область скриптов и соединить друг с другом.



Список спрайтов.

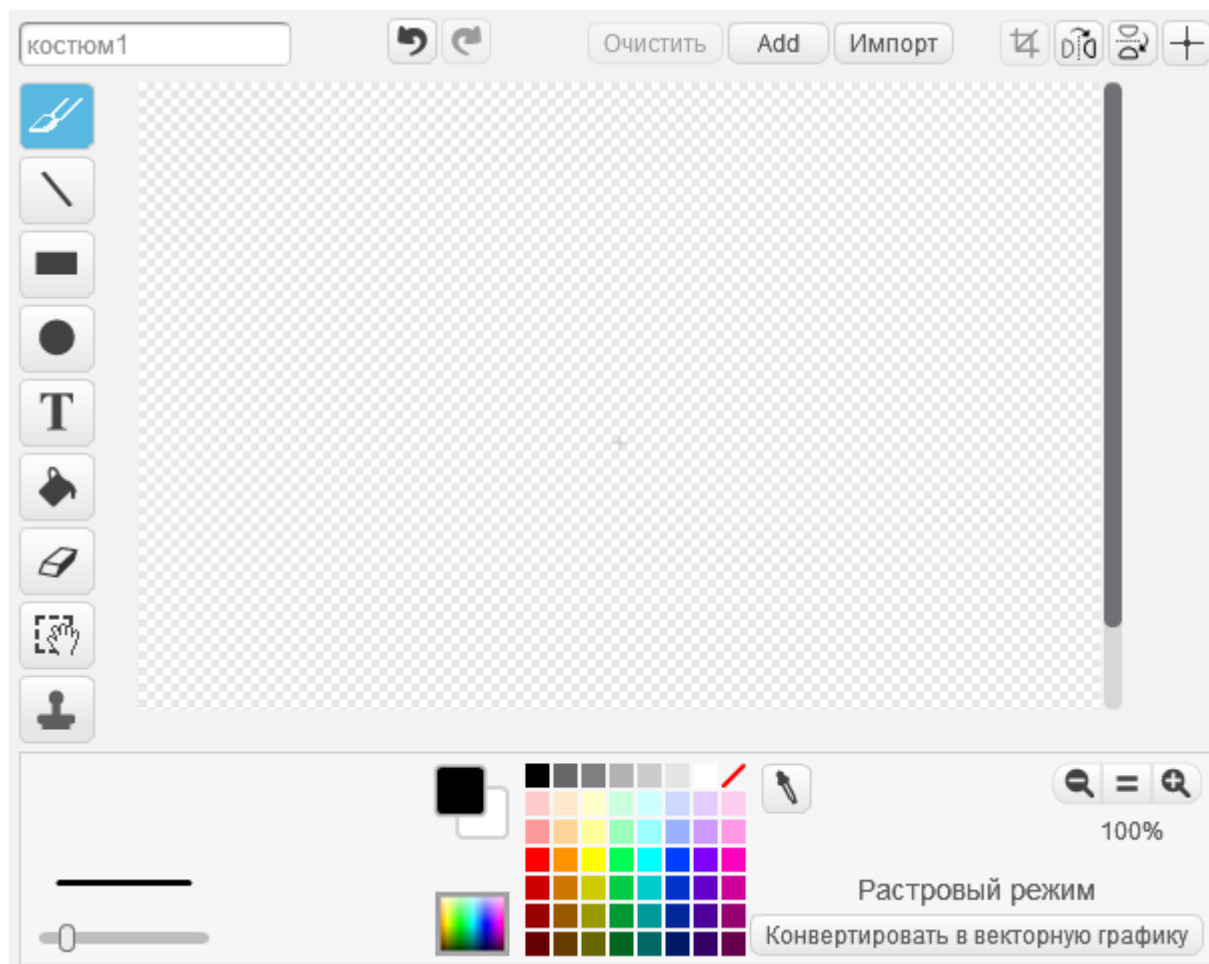
Рассмотрим подробнее область списка спрайтов. В правом верхнем углу области списка спрайтов есть четыре кнопочки добавления новых спрайтов. Первая – добавление спрайта из библиотеки спрайтов. Она содержит огромное количество персонажей.



Выбрав кого-нибудь в библиотеке спрайтов, вы увидите, что новый спрайт появился в списке.



Следующая кнопка позволяет нарисовать новый спрайт самостоятельно во встроенном графическом редакторе.



Графический редактор имеет целый ряд инструментов для рисования. Позднее мы более подробно познакомимся с этим редактором, а пока нарисуйте какую-нибудь закорючку.



Новый спрайт появится в списке.



Следующая кнопка позволяет импортировать спрайт из файла с картинкой.



Выберите любой рисунок хранящийся у Вас на компьютере. Новый спрайт появится в списке. Его изображение можно отредактировать в графическом редакторе.

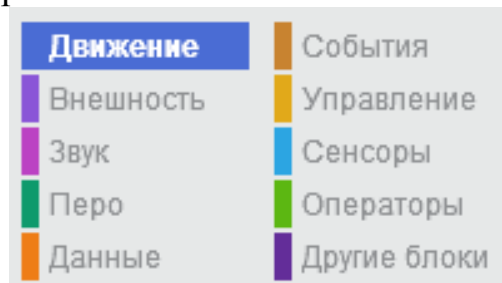


Четвертая кнопка позволяет импортировать изображение с вебкамеры, подключенной к компьютеру, т.е. сделать фотографию и преобразовать ее в спрайт.



Скрипты









При создании скрипта (программы) используется палитра блоков, которая занимает среднюю часть экрана. В ее верхней части располагается 8 разноцветных ссылок, которые выбирают нужную группу команд. Команды выбранной группы отображаются в нижней части окна.



Блок «Движение»

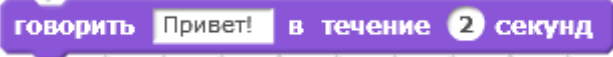

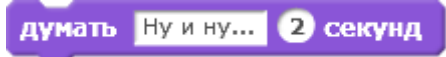






Синие блоки – это блоки движения. Они заставляют спрайт перемещаться по сцене.

	Спрайт переместиться на 10 шагов в текущем направлении.
	Повернуть спрайт на 15 градусов по часовой стрелке по отношению к текущему направлению спрайта.
	Повернуть спрайт на 15 градусов против часовой стрелки по отношению к текущему направлению спрайта.
	Повернуть спрайт в указанном направлении.
	Повернуться по направлению к одному из спрайтов. Для выбора спрайта следует нажать на чёрный треугольничек, и выбрать из выпадающего списка.
 	Переместить спрайт в точку экрана с заданными координатами X и Y. В точке с заданными координатами окажется центр спрайта. Применение этого блока равносильно применению одновременно двух блоков «Установить X в» и «Установить Y в».
	Переместить спрайт в указатель мышки, либо в центр указанного, с помощью выпадающего списка, спрайта. При этом центр спрайта окажется на самом кончике мышки.
	Плавно переместить спрайт в точку с заданными координатами. Время перемещения задаётся в секундах. Для

	<p>быстрого перемещения можно использовать десятичные дроби. Например, 0.5 секунд – это половина секунды, 0.25 секунд – это четверть секунды, 0.1 секунд – это одна десятая секунды, это очень быстро.</p>
	Изменить координату X спрайта на указанную величину. При этом спрайт сместиться вправо. При изменении на отрицательную величину, спрайт сместиться влево.
	Установить новое значение координаты X спрайта.
	Изменить координату Y спрайта на указанную величину. При этом спрайт сместиться вверх. При изменении на отрицательную величину, спрайт сместиться вниз.
	Установить новое значение координаты X спрайта.
	Очень важный блок! Позволяет спрайту отразиться от границы окна и продолжить движение в новом направлении.
	Изменяет стиль вращения спрайта. Всего есть три типа вращения: вправо-влево, не вращать, кругом.
	Это не управляющий блок, а блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение координаты X спрайта. Этот блок, как и все овальные блоки, можно использовать в операторах наравне с числами и переменными
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение координаты Y спрайта.
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущее направление спрайта в градусах.

Блок «Внешность»

Фиолетовые блоки – блоки внешности. Они позволяют изменять внешний вид спрайтов, позволяют им «разговаривать», а также прятаться и появляться.

	Спрайт будет говорить слово «Привет!» в течение двух секунд. При этом выполнение скрипта будет приостановлено.
	Спрайт постоянно будет говорить слово «Привет!» без остановки выполнения скрипта. Сказанная фраза будет постоянно сопровождать спрайт, пока он не скажет другую фразу.
	Спрайт будет думать «Ну и ну...» в течение двух секунд. При этом выполнение скрипта не будет приостановлено.
	Спрайт постоянно будет думать «Ну и ну...» без остановки выполнения скрипта. Эта мысль будет постоянно сопровождать спрайт, пока он не подумает о чем-нибудь другом.
	Спрайт станет виден. Если, при выполнении этого блока, спрайт не виден, то, возможно, он скрыт другим спрайтом. В этом случае может помочь блок «Перейти в верхний слой». Также следует проверить координаты спрайта, может быть, он находится за границей экрана.
	Спрайт станет невидимым.
	Спрайт изменит свой текущий костюм на костюм, выбранный из выпадающего списка.
	Спрайт изменит свой текущий костюм на следующий костюм.
	Текущий фон сцены изменится на фон, выбранный из выпадающего списка.

	Значение эффекта «цвет» изменится на указанную величину. В этом примере значение эффекта увеличится на 25.
	Значение эффекта «цвет» изменится до указанного значения. В этом примере значение эффекта станет равно нулю.
	Спрайт примет исходный вид, все графические эффекты будут убраны.
	Изменить размер спрайта на указанное количество процентов. В данном случае на 10%.
	Установить размер спрайта в процентах от исходного размера. В данном случае спрайт примет исходный размер.
	Спрайт перейдёт в самый верхний слой.
	Спрайт перейдёт на один слой назад.
	Это не управляющий блок, а блок получения информации из программы. Он возвращает из памяти работающей программы цифровое значение текущего номера костюма.
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущее имя фона.
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущий размер спрайта.
Дополнительные блоки для сцены	
	Сцена сменит фон на выбранный из выпадающего списка.
	Фон сцены изменится на следующий.
	Блок получения информации из программы. Он возвращает из памяти работающей программы цифровой

	номер текущего фона.
--	----------------------

Блок «Звук»












Малиновые блоки – это блоки звуков.

	Играть звук «рор», выбранный из библиотеки звуков, без остановки выполнения скрипта.
	Играть звук «рор», выбранный из библиотеки звуков до завершения. При этом выполнение скрипта будет приостановлено.
	Тишина! Остановить все звуки.
	Первому барабану играть одну четверть такта. В выпадающем списке можно выбрать один из восемнадцати барабанов.
	Подождать одну четверть такта.
	Играть ноту №60 (среднее До) полтакта.
	Выбрать один из двадцати музыкальных инструментов с помощью выпадающего списка.
	Изменить громкость на указанное количество процентов. В данном случае сделать тише на 10%.
	Установить требуемый уровень громкости в процентах от системного уровня.
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущий уровень громкости в процентах.
	Изменить темп на 20 ударов в минуту.
	Установить темп 60 ударов в минуту.
	Блок получения информации из программы. Он возвращает из памяти работающей программы текущий темп.

Блок «Перо»

Блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение темпа в ударах в минуту.

Г. Зелёные блоки рисования пером.

	Очищает всё отпечатанное и нарисованное перьями на сцене.
	Отпечатывает изображение спрайта на сцене. Не на фоне, а на сцене. Фоны можно изменять, а отпечаток будет поверх всех фонов.
	Опустить перо. После этой команды все перемещения пера будут сопровождаться линией.
	Поднять перо. После этой команды перемещения пера не будут оставлять следов на сцене.
	Установить цвет пера. Выбор цвета можно осуществить при помощи курсора.
	Изменить цвет пера на 10. Принцип изменения цвета такой же, как и в блоке изменения графического эффекта «цвет».
	Установить цвет пера в соответствии с таблицей цветов.
	Изменить оттенок пера на 10%.
	Установить оттенок пера в 50%.
	Изменить размер пера: увеличить на 1 пункт.
	Установить размер пера равным 1.

Блок «Данные»

Блоки переменных не видны до тех пор, пока вы не создадите переменную с помощью кнопки «создать переменную».



Это сама переменная. При использовании в программе этот миниблок возвращает из памяти работающей программы текущее значение переменной «а». Если около переменной поставить маленькую галочку,



то её значение будет показано на экране.



Установить значение переменной «а» в определённое значение. Значение может быть, как числовым, так и текстовым. С помощью выпадающего списка можно выбрать любую переменную.



Изменить значение переменной «а» на единицу. С помощью выпадающего списка можно выбрать любую переменную. Изменять можно только числовую переменную, при попытке изменить текстовую переменную будет ошибка «NaN».



Переменная будет показана на экране, и около неё появится галочка.



Переменная скроется из вида, а маленькая галочка будет снята.

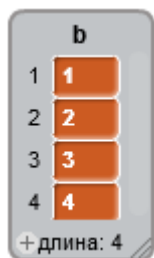


Блоки списков не видны до тех пор, пока вы не создадите список с помощью кнопки «создать список».

Создать список

b

Это сам список. Если около него поставить маленькую галочку, то на экране появится список.



добавить слово к b

Добавить ещё один элемент списка к списку «b». Элемент списка может быть числовым или текстовым.

удалить 1 из b

Удалить элемент под номером 1 из списка «b».

вставить слово в 1 из b

Вставить значение в элемент списка «b» под номером 1, при этом все последующие элементы сдвинутся. Вставляемое значение может быть либо числовым, либо текстовым.

заменить элемент 1 в b на слово

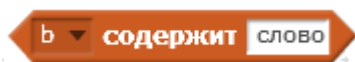
Заменить значение в элементе списка «b» под номером 1, при этом все последующие элементы останутся на своих местах. Вставляемое значение может быть либо числовым, либо текстовым.

элемент 1 из b

Блок возвращает из памяти работающей программы текущее значение элемента списка «b» под номером 1.

длина списка b

Блок возвращает из памяти работающей программы длину списка «b», то есть количество элементов списка «b».



Логическое значение. Принимает значение «истина», если список «b» содержит элемент со значением «слово».




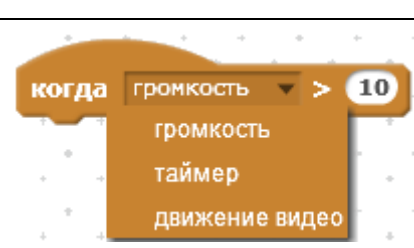

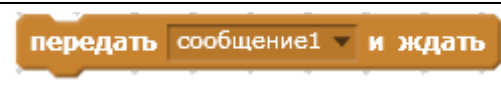
Список будет показан на экране, и около него появится галочка.



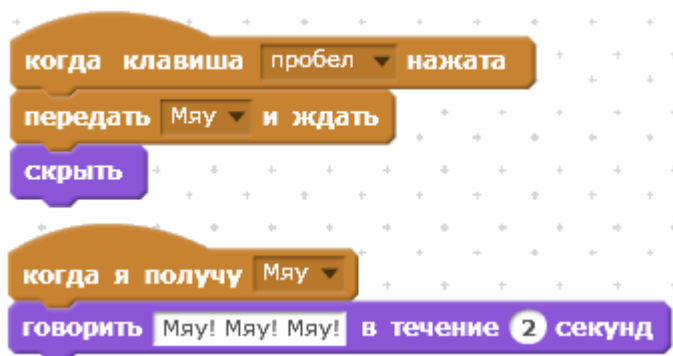
Список скроется из вида, а маленькая галочка будет снята. Е. Блоки событий.

Блок «События»

Следующие блоки – коричневые блоки событий. Эти блоки имеют «шапочку», поэтому они всегда располагаются в начале скрипта. Они позволяют Вашей программе реагировать на различные события, такие как нажатие клавиш на клавиатуре, клик мышкой и т.п.

	Блок начала выполнения программы. Выполняется, если нажат зелёный флажок.
	Блок управления с клавиатуры. Выполняется, если нажата клавиша, выбранная из выпадающего списка.
	Блок управления мышкой. Выполняется, если мышкой кликнуть на спрайт.
	Блок управления скриптом в зависимости от фона. Выполняется, если произошло изменение фона.
	Блок управления скриптом в зависимости от уровня громкости, либо от значения таймера, либо от движения видео. Выполняется при соответствующем изменении параметра, выбранного из выпадающего списка.
	Блок управления скриптом с помощью сообщений. Выполняется при получении соответствующего сообщения, выбранного из выпадающего списка.
	Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта не приостанавливается.
	Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта приостанавливается, и он ждёт, пока другие спрайты выполнят все скрипты блока «когда я получу сообщение» до конца. Затем выполнение текущего скрипта продолжается.

Чтобы лучше понять, как работает этот блок, потренируемся на кошках. Сделайте Коту следующие два скрипта.


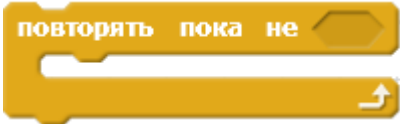
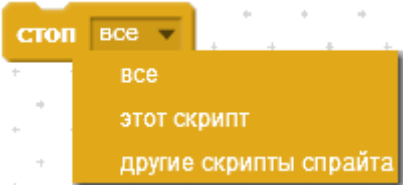





При нажатии на пробел, первый скрипт передаст сообщение «Мяу», и будет ждать, пока не закончится выполнение второго скрипта. Второй скрипт выполняется ровно две секунды. После того, как Кот наорётся, управление снова будет передано первому скрипту, и он скроет Кота с глаз долой, чтобы не шумел по ночам.

Блок «Управление»

Желтые блоки – блоки управления. Это очень важные блоки, они позволяют организовать циклическое выполнение программы, позволяют создавать ветвление и т.п.








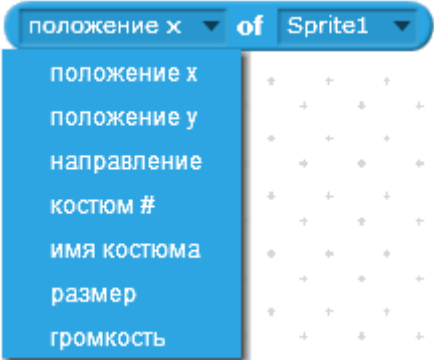
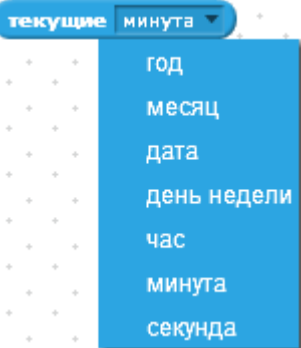

	Выполнение скрипта будет приостановлено на указанное количество секунд.
	Блок циклического повтора. Указанное количество раз выполняет блоки, которые находятся у него внутри.
	Блок постоянного циклического выполнения. Постоянно выполняет блоки, которые находятся у него внутри.
	Блок условия «если». Если условие в заголовке блока истинно, то будут выполнены блоки, которые находятся у него внутри.
	Блок условия «если-иначе». Если условие в заголовке блока истинно, то будут выполнены блоки, которые находятся после слова «если», в противном случае, будут выполнены блоки, которые находятся после слова «иначе».

	«иначе».
	Блок ожидания выполнения условия. Выполнение скрипта приостанавливается до тех пор, пока условие не станет истинным. Если условие сразу было истинным, то задержки выполнения скрипта не происходит.
	Блок цикла с условием. Содержимое блока будет выполняться до тех пор, пока условие в заголовке не станет истинным. То есть цикл работает до тех пор, пока условие в заголовке ложно, как только оно станет истинным, выполнение цикла завершится, и продолжится выполнение скрипта.
	Скрипт остановки выполнения. Из выпадающего списка можно выбрать, что остановить. Все скрипты программы, текущий скрипт, в котором использован этот блок или другие скрипты этого спрайта.
	Блок управления клонами спрайта. Этот блок выполняется при создании клона спрайта.
	Блок создания клонов. Из выпадающего списка можно выбрать, клон какого спрайта необходимо создать, если выбрать «себя самого», то будет создан клон текущего спрайта.
	Блок удаляет клон спрайта. Удаление клонов необходимо для того, чтобы их не становилось слишком много. При большом количестве клонов программа может зависнуть.

Блок «Сенсоры»

Голубые блоки – блоки сенсоров. Они используются совместно с желтыми блоками условий. Также эти блоки позволяют организовать ввод информации в компьютер.





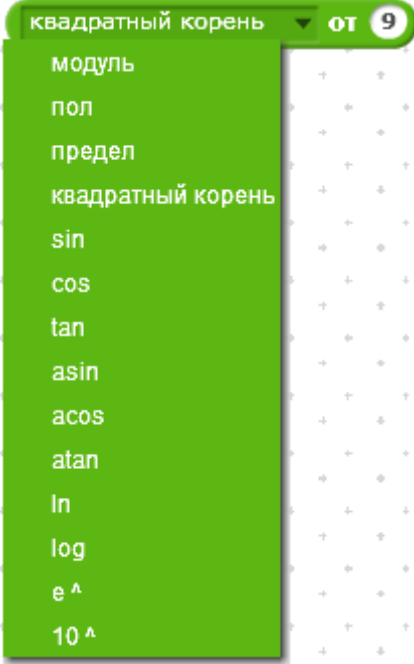
	Сенсор касания спрайтов. Из выпадающего списка можно выбрать любой спрайт, а также курсор мышки и край экрана. Сенсор принимает значение «истина», если текущий спрайт касается выбранного.
	Сенсор касания цвета. Цвет можно выбрать с помощью курсора мышки, который становится похож на руку. Сенсор принимает значение «истина», если текущий спрайт касается выбранного цвета.
	Сенсор касания одного цвета другим. Цвета можно выбрать с помощью курсора мышки, который становится похож на руку. Сенсор принимает значение «истина», если два выбранных цвета соприкасаются.
	Блок вычисления расстояния от текущего спрайта, до спрайта, выбранного из выпадающего списка или до указателя мышки. Возвращает числовое значение расстояния.
	Блок запроса на ввод информации. Выводит на экран вопрос, и строку ввода информации. После ввода ответа нажмите Enter, или кликните мышкой на синий кружочек с галочкой.
	Блок ввода информации. Возвращает значение, введенное пользователем в строке ввода.
	Сенсор нажатия клавиш клавиатуры. Сенсор принимает значение «истина», если нажата клавиша, выбранная из выпадающего списка.
	Сенсор нажатия мышки. Сенсор принимает значение «истина», если мышка нажата.
	Блок возвращает из памяти работающей программы текущую координату X курсора мышки.

	Блок возвращает из памяти работающей программы текущую координату Y курсора мышки.
	Блок возвращает из памяти работающей программы текущее значение уровня громкости.
	Блок возвращает из памяти работающей программы количество движений над видеоизображением.
	Блок включения, выключения и поворота видеоизображения.
	Блок изменения прозрачности видеоизображения.
	Блок возвращает из памяти работающей программы текущее значение таймера в секундах, то есть время с момента последнего перезапуска таймера.
	Блок перезапуска таймера. Обнуляет значение таймера.
	Блок возвращения из памяти работающей программы различных параметров спрайта, выбранного из выпадающего списка.
	Блок возвращения из памяти компьютера текущих значений даты и времени. Все данные возвращаются в числовом формате.
	Блок возвращения из памяти компьютера количества дней, прошедших с начала века.

Блок «Операторы»

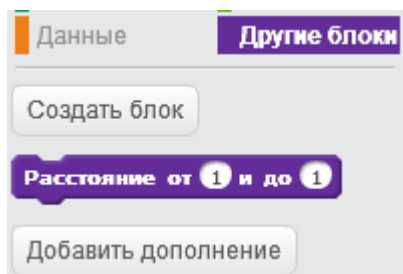
Зеленые блоки – операторы. Используются совместно с сенсорами, блоками условий и с переменными. Самый нижний зеленый блок содержит в себе множество математических функций.

	Оператор сложения двух чисел.
	Оператор вычитания двух чисел
	Оператор умножения двух чисел.
	Оператор деления двух чисел.
	Оператор генерации случайных чисел. Возвращает случайное целое число из заданного диапазона.
	Логическое сравнение двух чисел. Возвращает значение «истина», если первое число меньше второго.
	Логическое сравнение двух чисел. Возвращает значение «истина», если числа равны.
	Логическое сравнение двух чисел. Возвращает значение «истина», если первое число больше второго.
	Логическая операция И. Возвращает значение «истина», если оба условия истинны.
	Логическая операция ИЛИ. Возвращает значение «истина», если хотя бы одно из условий истинно.
	Логическое НЕ. Возвращает значение «истина», если условие ложно.
	Блок объединения двух текстовых строк. Возвращает объединённую строку.

	<p>Блок получения символа из строки текста. Возвращает символ с указанным номером из строки текста.</p>
	<p>Блок определения длины строки. Возвращает числовое значение длины текстовой строки.</p>
	<p>Оператор остаток от деления. Возвращает остаток от деления первого числа на второе.</p>
	<p>Оператор округления. Возвращает число без дробной части.</p>
	<p>Оператор математических функций. Возвращает числовое значение математической функции, выбранной из выпадающего списка.</p>

Блок «Другие блоки»

Темно-фиолетовые блоки – это самодельные блоки, которые выполняют в Скретч роль подпрограмм. Вот, например, блок, вычисляющий расстояние от первой точки на прямой до следующей.



Кнопка «Добавить дополнения» позволяет Скретч взаимодействовать с устройствами, которые подключаются к компьютеру через USB порт.

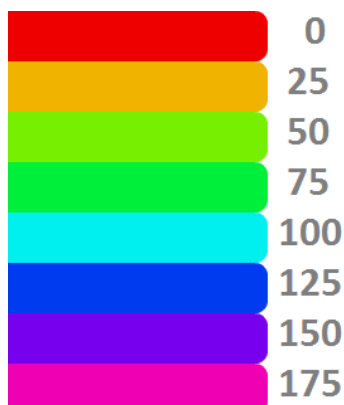


Если подключить PicoBoard, то появятся черные блоки.

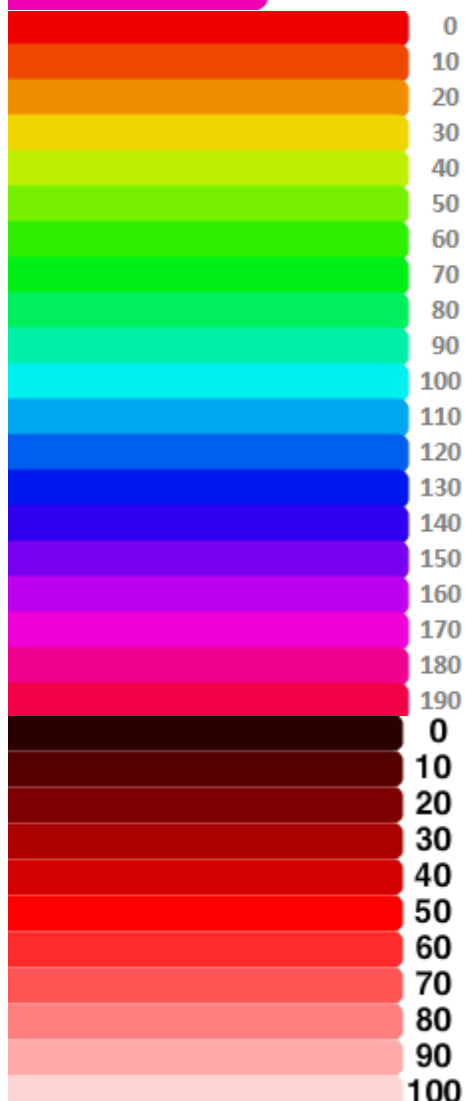
Пронумерованные цвета и оттенки.

В Scratch каждый цвет имеет свой номер. Оттенки цвета также имеют номера. Это сделано для более удобной работы с цветом. Числовые значения цветов используются в блоках «изменить цвет пера» и «установить цвет пера».

Числовые значения оттенков используются в блоках «изменить тень пера» и «установить тень пера». Обратите внимание, из-за ошибки перевода на русский язык оттенок пера назван тенью, возможно, это скоро будет исправлено.



Началом отсчёта, цветом с номером ноль, как и в радуге, является красный цвет. На следующем рисунке приведена таблица соответствия цветов и их числовых обозначений с шагом 25.



Более подробная таблица соответствия цветов и их числовых обозначений с шагом 10.

Все оттенки цветов также пронумерованы от нуля, самого тёмного оттенка, до ста, самого светлого оттенка. Ниже приведена таблица оттенков. Нормальный красный цвет получается при значении оттенка в 50.

Графические эффекты.

В Scratch 2.0 есть семь графических эффектов, которые могут быть применены к спрайтам и фонам. Графические эффекты применяются с использованием блоков «Изменить эффект на», и «Установить эффект в значение».

Эффект «цвет»

изменить цвет эффект на 25



Это циклический эффект. Цвет спрайта изменяется при изменении значения эффекта от 0 до 200. При значении больше 200, например, при 201, спрайт выглядит так же, как и при 1. Например, если у нас есть спрайт красного цвета, то при значении эффекта «цвет» равного 0, спрайт будет красным. При значении эффекта 25, спрайт станет оранжевым. При значении эффекта 50, спрайт станет жёлтым и так далее, пока, при достижении значения эффекта 200, спрайт снова не станет красным. Направление изменения цвета спрайта показано на рисунке.



Эффект «рыбий глаз»

изменить рыбий глаз эффект на 25



При применении этого эффекта спрайт как будто надувается изнутри. Самые интересные изменения происходят со спрайтом при изменении значения эффекта «рыбий глаз» от 0 до 200.

Эффект «завихрение»

изменить завихрение эффект на 100



При применении этого эффекта спрайт как будто закручивается изнутри. Края спрайта остаются почти без изменений, а центр вращается против часовой стрелки. Самые интересные изменения происходят со спрайтом при изменении значения эффекта «завихрение» от 0 до 500.

<p>Эффект «укрупнение пикселей»</p> <p>изменить укрупнение пикселей эффект на 25</p> 	<p>Пиксели – это точки на экране из которых состоит изображение спрайта. При применении эффекта «укрупнение пикселей», точки изображения начинают увеличиваться в размерах и сливаться, образуя всё более крупные квадратики. В результате, при значении эффекта равном 500, остаётся один или два цветных квадратики.</p>
<p>Эффект «мозаика»</p> <p>изменить мозаика эффект на 5</p> 	<p>Этот эффект превращает спрайт в мозаику. Изменения происходят не плавно, а скачкообразно. Например, при значении эффекта от 0 до 4, со спрайтом ничего не происходит. При значении 5 спрайт превращается в мозаику из четырёх элементов. При дальнейшем изменении значения от 5 до 14 спрайт по-прежнему состоит из четырёх элементов, а при достижении 15 количество котиков скачкообразно увеличивается до девяти.</p>
<p>Эффект «яркость»</p> <p>изменить яркость эффект на 25</p> 	<p>При применении этого эффекта спрайт становится ярче. При значении эффекта больше 100 никакие изменения не происходят, потому что спрайт достиг максимальной яркости. На рисунке показан Кот с яркостью 100. Все линии чёрного цвета стали белыми.</p>
<p>Эффект «призрак»</p> <p>изменить призрак эффект на 50</p> 	<p>Этот эффект можно было бы назвать «прозрачность», он задаёт величину прозрачности спрайта. При значении эффекта «призрак» равного 0, прозрачности нет совсем, а при значении 100, прозрачность максимальна, то есть спрайт вообще невидим.</p>