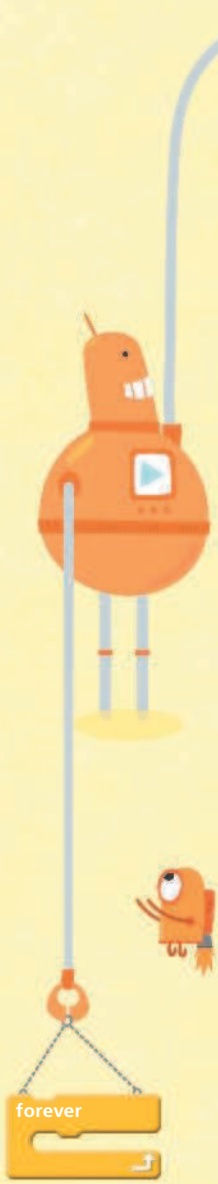
Пособие по программированию для обучения с нуля

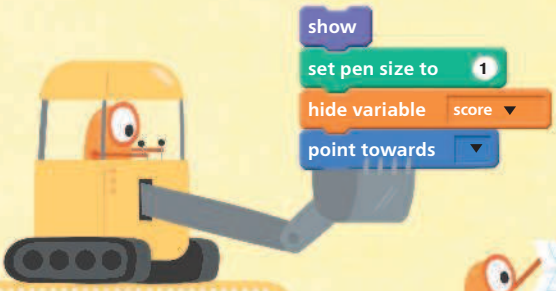
SCRATCH

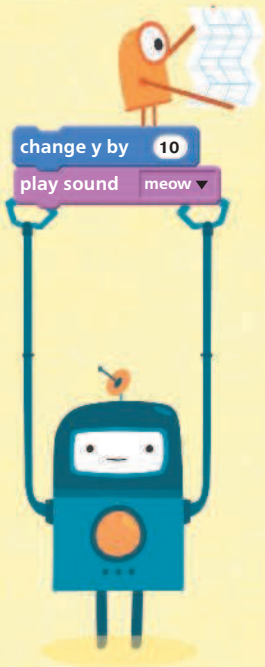
для юных программистов

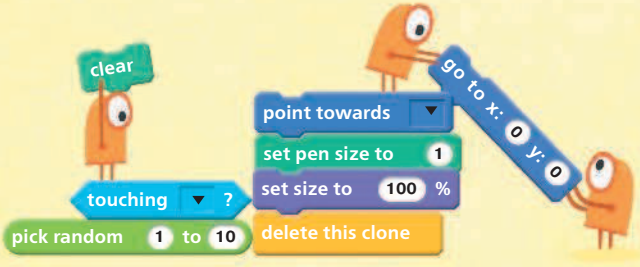
Аванта

Оглавление









Что такое программирование?

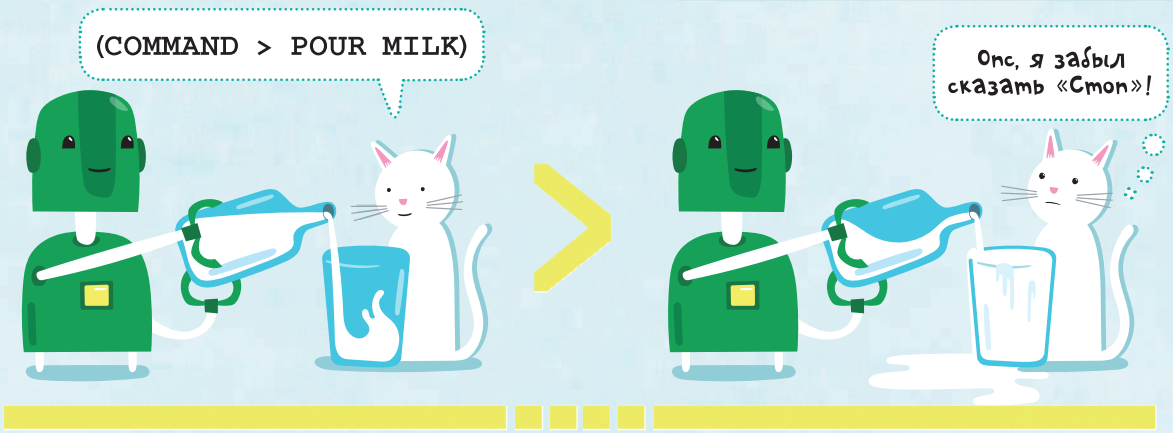
Программирование — это написание инструкций для компьютера, а окончательная версия таких инструкций называется программой. Когда вы научитесь программировать, сможете создавать собственные программы.

Быть понятным

Для того чтобы программа работала, ее нужно написать понятным компьютеру образом. Это означает, что все инструкции следует разбить на простые и ясные шаги, не говоря уже о том, что они должны быть написаны на определенном языке программирования.

ВНИМАНИЕ!

Компьютер четко следует инструкциям. Он не может мыслить сам, так что все действия должны быть ясно описаны и ничего не должно быть оставлено без внимания.

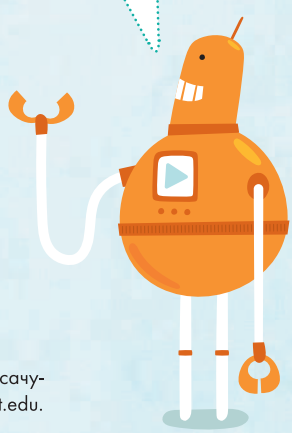


Язык программирования

Язык программирования во многом схож с обычным языком человеческого общения. Только он имеет ограниченный набор слов и точные правила, как писать этими словами программы.

Существует множество языков программирования, адаптированных под разные типы задач. Для многих юных читателей первым языком программирования может стать язык Scratch («Скретч»), созданный специально для учебных целей.

Язык Scratch прекрасно подходит для создания игр и анимации, а также позволяет получить основные навыки написания программ.

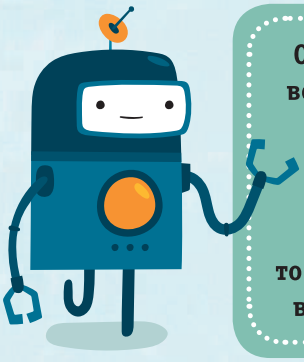


SCRATCH

Язык Scratch разработан Lifelong Kindergarten Group при MIT (Массачусетский технологический институт) Media Lab. См. http://scratch.mit.edu.

Почему мы выбрали Scratch?

Язык Scratch разработан таким образом, чтобы писать программы на нем было не сложнее, чем играть в конструктор. Вставляя друг в друга готовые к использованию блоки кода, вы, по сути, уже пишете программу.



Обратите внимание на такие вот текстовые вставки. В этих вставках на зеленом фоне объясняются различные КЛЮЧЕВЫЕ ПОНЯТИЯ. Если текст на синем фоне, то это СОВЕТ по использованию возможностей языка Scratch.

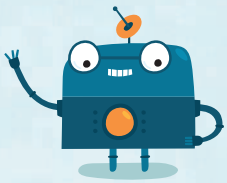
Немного об этой книге

Эта книга покажет вам, как использовать бóльшую часть инструментов языка Scratch для создания анимированных историй и игр. Также вы получите множество советов, как создавать собственные программы. Все примеры разбиты на короткие шаги, которым легко следовать при обучении.

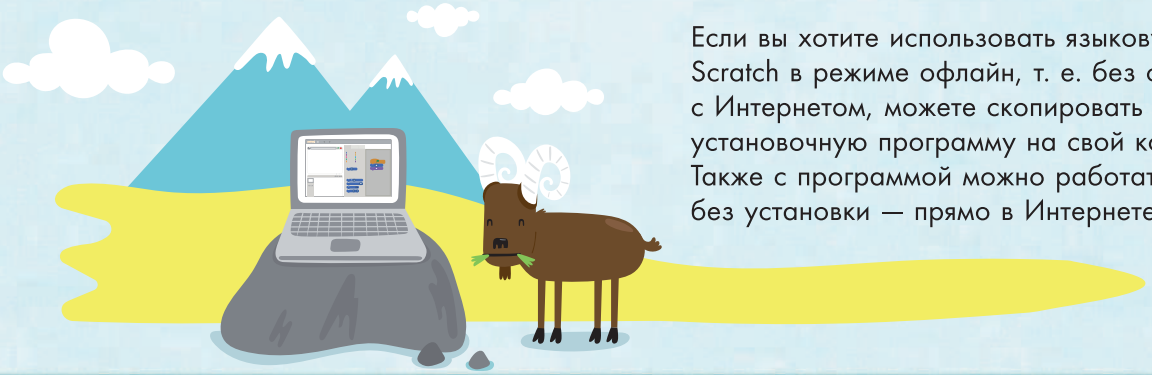
Приступаем

Чтобы начать использовать язык Scratch, проще всего посетить вебсайт этого обучающего проекта. Все, что вам нужно, — это компьютер — тот, который с клавиатурой (планшет не подойдет), — и выход в Интернет.

Посетите ресурс **https://scratch.mit.edu**. Для использования программы необходимо создать учетную запись.



Scratch есть и на русском языке, но лучше привыкать работать с английским интерфейсом, ведь более серьезные программы для программирования имеют только английский интерфейс.



Если вы хотите использовать языковую среду Scratch в режиме офлайн, т. е. без связи с Интернетом, можете скопировать установочную программу на свой компьютер. Также с программой можно работать без установки — прямо в Интернете.

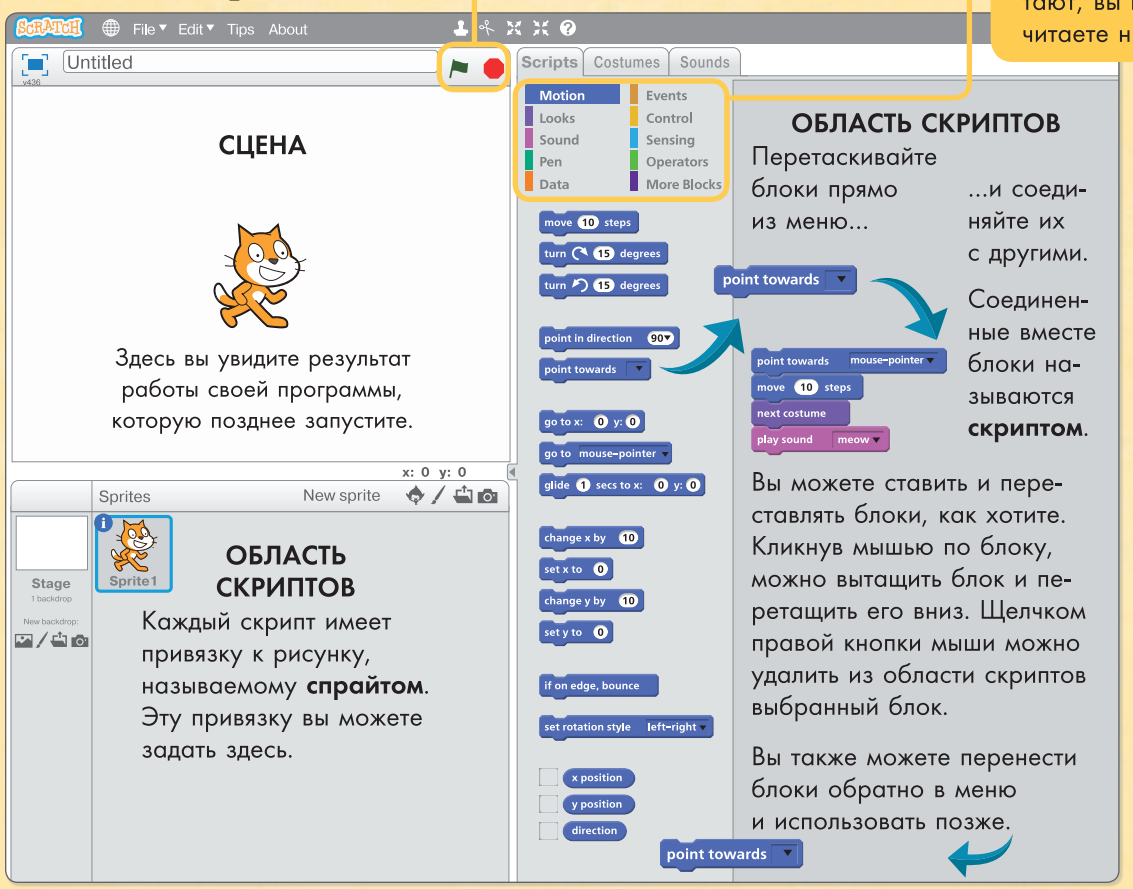
НАЧИНАЕМ С ЯЗЫКА SCRATCH

Когда вы запустите на компьютере Scratch, увидите такую картинку.

Для начала работы кликните на синей плашке по надписи Create («Создать»).

Эти кнопки — **зеленый флаг** и **красный шарик** — используются для запуска и остановки программы.

Под этими пунктами скрываются **блоки меню**, а о том, как они работают, вы прочитаете ниже.

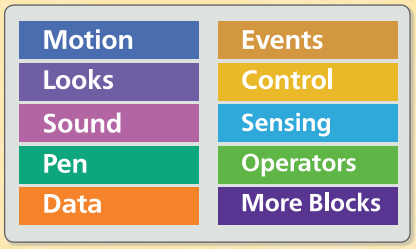


Блоки меню

В меню каждого из **пунктов основного меню** вы увидите множество различных по форме и цвету блоков. Например:

- **Motion** («Движение») — блоки меню (синие), заставляющие спрайты двигаться.
- **Looks** («Внешний вид») — блоки меню (фиолетовые), изменяющие внешний вид спрайтов.
- **Control** («Управление») — блоки меню (золотистые), контролирующие исполнение скриптов.

Кликните мышью по пункту меню, и вы увидите доступные блоки (полный список этих блоков представлен начиная со стр. 82).

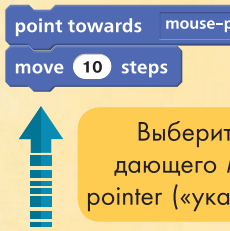


Вот все десять пунктов меню.

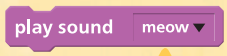
Первые шаги

1 Для начала перетащите первые два блока (например, из меню **Motion**) в **область скриптов** и научите кота ходить...

Затем кликните мышью по пункту меню **Sound** («Звук») и добавьте блок **play sound** («проиграть звук»).



Выберите из выпадающего меню mouse-pointer («указатель мыши»).



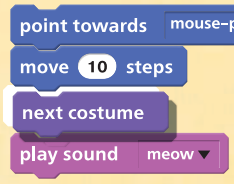
Выберите из выпадающего меню meow («мяу»).

2 Чтобы запустить скрипт, просто кликните по нему мышью. Для эксперимента кликните по скрипту несколько раз и посмотрите, что будет.

Код запущенного скрипта подсветится, а кот начнет перемещаться по экрану и мяукать. (Если кот уйдет слишком далеко вбок, вы можете перетащить его обратно мышью.)

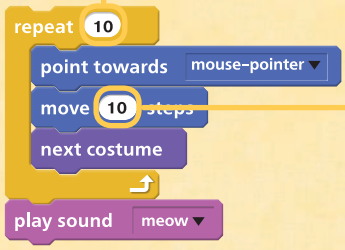
3 Но кот не будет выглядеть так, словно он ходит. Для этого его лапы должны двигаться...

Кликните мышью по пункту меню **Looks** и добавьте блок **next costume** («следующий костюм»). Такое переключение на следующую картинку, или костюм, того же спрайта создаст иллюзию того, что кот двигает лапами. Кликните по этому скрипту несколько раз.



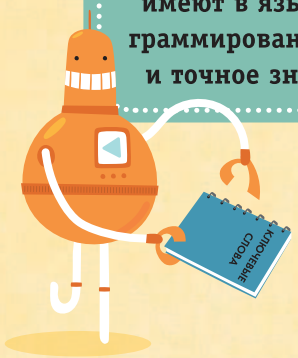
4 Кот действительно будет двигать лапами, но только тогда, когда вы кликаете по скрипту мышью. Чтобы он делал это без вашего участия, нужно обратиться к меню **Control** и выбрать там блок **repeat** («повторить»). Этот блок сделает так, что все инструкции внутри него будут повторяться.

В этом окошечке вы можете указать цифрой, сколько раз будет повторяться действие.



КЛЮЧЕВЫЕ СЛОВА

Слова инструкций, такие как ДВИЖЕНИЕ или ПРОИГРЫВАНИЕ, иногда называют КЛЮЧЕВЫМИ СЛОВАМИ, поскольку они имеют в языке программирования ясное и точное значение.

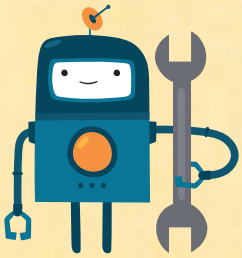


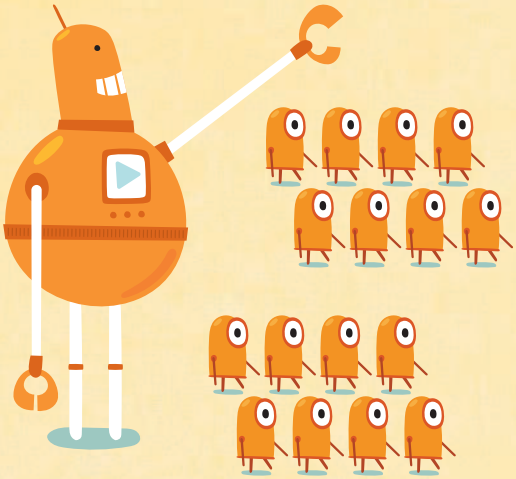
Поздравляем, только что вы написали свою первую программу!

ЦИКЛЫ

Блоки, запускающие ЦИКЛЫ (циклические повторения), очень широко используются в программировании, поскольку они позволяют сделать программу намного короче. С этими командами программы пишутся гораздо быстрее.

Переверните страницу, и вы увидите, как превратить этот скрипт в простую игру «Кошки-мышки».





Первые проекты

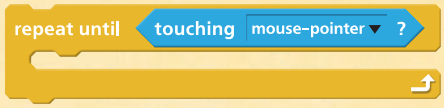
КОШКИ-МЫШКИ

В этой простой игре нужно не дать коту догнать мышку, удерживая ее в одном шаге от него. Если кот коснется указателя мыши, он скажет «Поймал!» и игра закончится.

1 Эта игра требует новой команды организации цикла — блока **repeat until** («повторять, пока не») из меню **Control**.

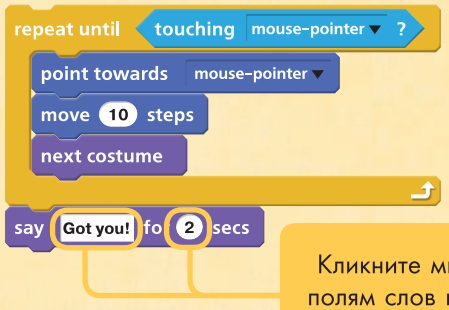
И не забудьте ромбовидный блок из меню **Sensing** («Сенсоры»).

2 Вставьте условие в соответствующее окошко команды повторения (блок повторения при этом увеличится и вместит окошко условия). Затем кликните по черному треугольнику и выберите из выпадающего меню mouse-pointer.

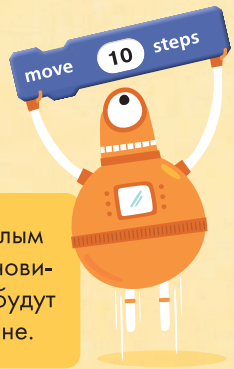


Этот цикл будет выполняться снова и снова до тех пор, пока кот не коснется указателя мыши.

3 Теперь возвратитесь к скрипту с предыдущей страницы и кликните мышью по первому блоку в цикле. Затем перетащите этот блок вместе со всеми присоединенными снизу блоками в свой новый цикл.



Кликните мышью по белым полям слов кота и установите, сколько секунд они будут высвечиваться на экране.

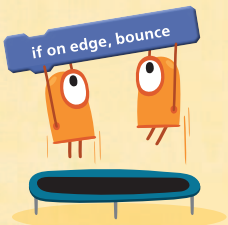


4 Закончите программу блоком **say** («сказать») из меню **Looks**.

Тестирование скрипта

5 Кликните мышью по скрипту и поводите указателем мыши по экрану. Кот должен следовать за «мышкой» (указателем мыши), пока не поймает ее. Запустите скрипт несколько раз.

Если кот достигнет края **сцены**, он начнет мерцать (его изображение станет неустойчивым). Однако вы можете исправить эту ошибку, поставив в начало цикла блок **if on edge, bounce** («если на краю, отскочить») из меню **Motion**.



ВЕТВЛЕНИЯ

Такие условные конструкции ветвления, как IF («если») и REPEAT UNTIL, позволяют сделать так, чтобы компьютер по-разному реагировал на выполнение условия (в данном случае — где находится кот). Именно поэтому эти команды называют и командами ветвления, и условными конструкциями.

Если ваша программа работает, возьмите с полочки пирожок.