

# 14 УРОК В SCRATCH

Обновление	Просмотров	Рейтинг
5 февраля 2015	10	• • 0

## И обобщим основные приемы...

### 1. Переменные

Переменная – это ячейка памяти, в которую программа может записывать значения: числа, строки и использовать (считывать) их позднее для вычисления новых значений.

В Scratch переменные можно создавать и использовать с помощью блоков в группе **Переменные**. При создании переменной задается имя, которое используется в командах для ссылки на ее текущее значение. Кроме того, при создании задается, является ли переменная доступной всем спрайтам (глобальная переменная) или только одному конкретному спрайту (локальная переменная). Для присваивания и изменений значений переменных используются блоки **поставить переменная в выражение** и **изменить переменная на выражение**. Значение переменной можно отобразить на сцене с помощью монитора. Значение можно изменять и вручную с помощью слайдера (рычажка).

Посмотрите примеры: [Korvpall](#) и [Ideaal](#)

### 2. Списки

Список в Scratch представляет собой упорядоченный набор ячеек памяти (элементов). На элементы можно ссылаться с помощью имени массива и индексов. В Scratch можно создавать и использовать списки с помощью блоков группы **Переменные**. С помощью блоков этой группы можно добавлять элементы в конец списка или в середину, заменять, удалять и ссылаться на элементы. Посмотрите [пример](#)

### 3. Ввод/вывод

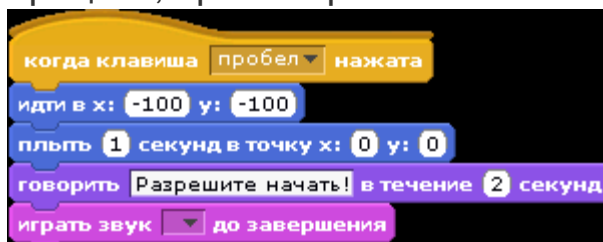
При решении задач часто необходимо задать программе исходные данные и отобразить результаты. В первом случае говорят о вводе (считывании) данных, во втором – о выводе данных (печати, отображении). В Scratch для ввода данных

можно использовать мониторы данных с рычажком (только для чисел) и блок **Спросить**, который дает возможность вводить числа и тексты в режиме диалога. Результаты переменных и/или списков можно выводить с помощью мониторов или блока **Говорить**.



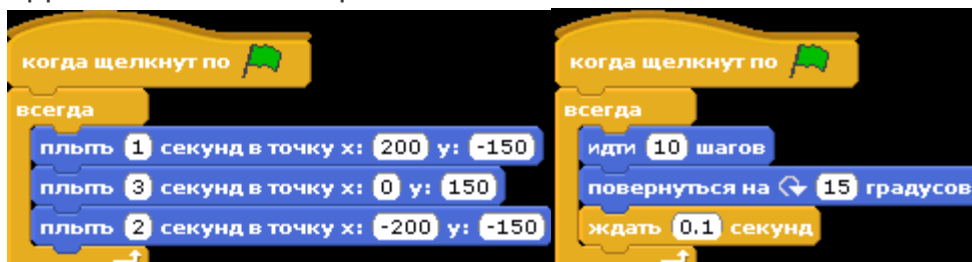
## 4. Последовательность и параллельность

При создании программы следует учитывать, что действия, задаваемые блоками, выполняются в определенном порядке. Простейшим случаем является последовательный процесс, при котором блоки выполняются подряд сверху вниз.



Два или несколько скриптов (процессов) можно выполнять одновременно, т.е. параллельно. Параллельное выполнение можно задавать несколькими способами. Например все скрипты, первый блок в которых это блок с зеленым флажком, запускаются одновременно и выполняются параллельно, если щелкнуть по зеленому флажку.

Параллельно выполняются также скрипты, начинающиеся блоком **Когда я получу сообщение**, в которых принимается одинаковое сообщение.

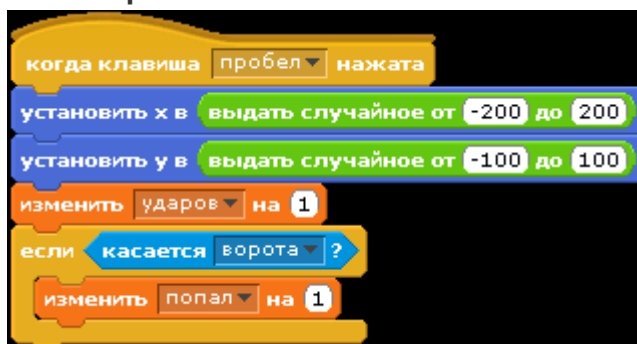


## 5. События

Объекты могут реагировать на определенные события: нажатие на какую-либо клавишу, щелчок мышью по объекту, прикосновение к другому объекту или краю сцены и т.п. В скриптах можно предусмотреть реакции на определенные события. Основным средством для обработки события являются так называемые блоки заголовков: **Когда клавиша ... нажата** и **Когда щелкнут по спрайту**.

Внутри скриптов часто используется блок **касается {спрайт | край | курсор}**? Если данный спрайт касается другого спрайта, края сцены или указателя мыши (курсора), то блок возвращает значение истина.

В принципе нажатие на зеленый флажок и запуск скриптов с помощью блоков **передать** и **когда я получу сообщение** также являются событиями.



## 6. Взаимодействие между спрайтами

Если программа состоит из нескольких скриптов, то часто возникает необходимость в координировании и синхронизации их работы. Один спрайт может обратиться к другим, те в свою очередь к следующим и т.д. Для организации работы скриптов используются

блоки: **передать сообщение**; **передать сообщение и ждать**; **когда я получу сообщение**.

Представленная рядом программа состоит из четырех скриптов. Скрипт с зеленым флажком является главным. С него начинается работа программы и он запускает остальные. Если пользователь вводит букву "д", то запускается скрипт **Читай**, и после окончания им работы запускается скрипт **Вычисли**. Если введена отличная от д буква, то сразу запускается скрипт **Вычисли**.

