

Другие блоки

Рассмотрим более подробно *Другие блоки* редактора Scratch. Блоки понадобятся при создании 3 уровня игры.

Другие блоки выполняют в Scratch роль подпрограмм или функций, которые необходимы для оптимизации скрипта проекта (уменьшения количества блоков в скрипте). Также команды данной группы необходимы для программирования некоторых устройств, например LEGO WeDo.

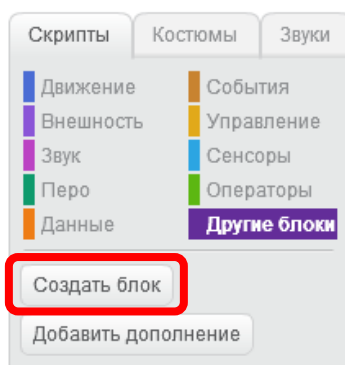
Другие блоки по своим свойствам очень похожи на блоки группы *Данные* — здесь также создаются команды. Но главное отличие состоит в том, что в *Данных* создаются переменные, а в *Других блоках* – блоки, в состав которых могут входить любые команды (в т.ч. для программирования LEGO-устройств), переменные и значения Scratch. Команды, значения и переменные программируют блок на определенные действия. Команды (блоки), созданные при помощи *Других блоков*, доступны в разделе *Другие блоки*.

Вывод: другие блоки при помощи других команд выполняют определенные задачи, а реализуют это выполнение при помощи своих команд.

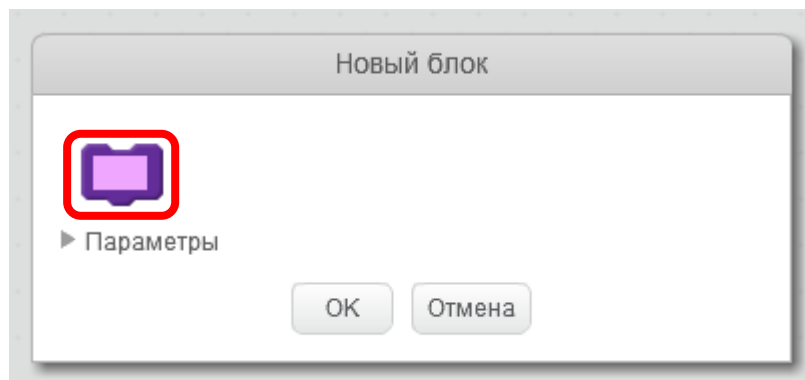
Создание блока

Раздел *Другие блоки* состоит из двух команд: *Создать блок* и *Добавить дополнение*.

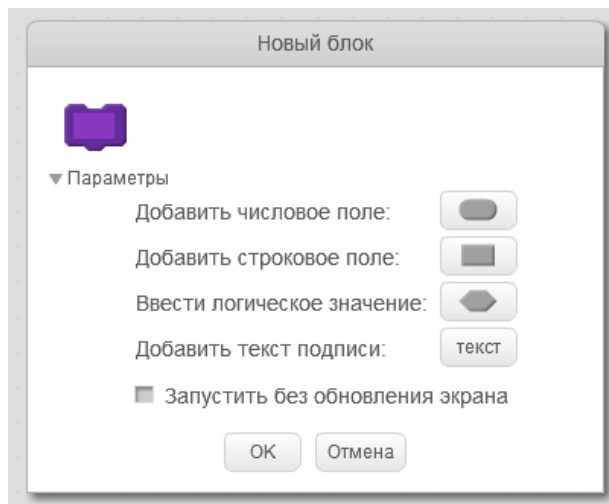
Чтобы создать новый блок, щелкните *Создать блок*.



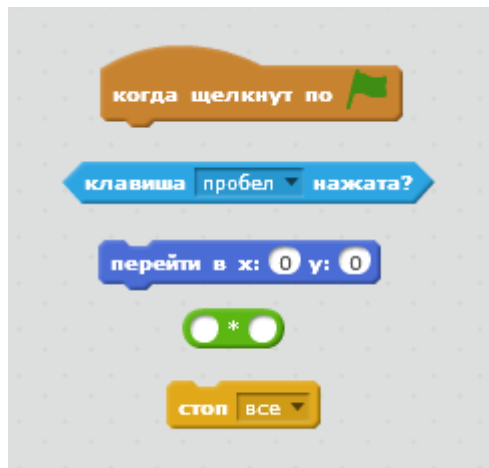
В диалоге *Новый блок* в фиолетовое поле введите название блока.



Доступны также параметры блока.

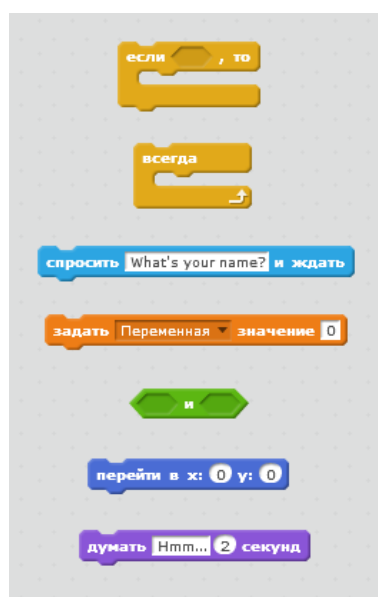


В среде программирования Scratch все блоки имеют свою «puzzle форму», которая намного облегчает программирование в Scratch.



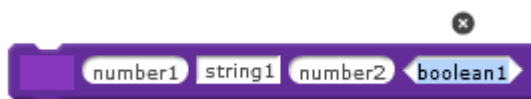
Благодаря «puzzle форме», пользователь понимает, какие блоки можно между собой соединить, а какие нельзя, и вскоре добивается правильного алгоритма действий.

Стоит отметить, что «puzzle формы» есть внутри блока. Они позволяют понять, какие команды и значения можно вставить внутрь блока.



Параметры *Других блоков* идентичны внутренним «puzzle формам». При создании нового *Другого блока* возможно задать «внутренние puzzle формы» (параметры).

Чтобы создать «внутреннюю puzzle форму», щелкните по необходимому параметру. В одном блоке может быть несколько параметров.

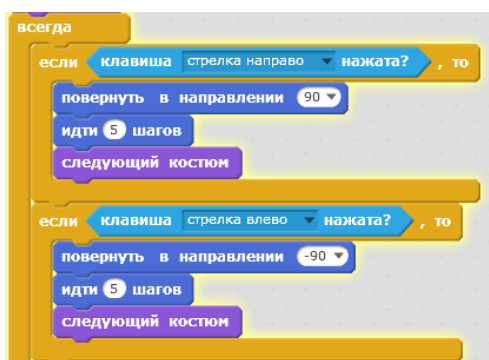


Чтобы удалить параметр, щелкните по «внутренней puzzle форме» и нажмите .

Параметры необходимы для того, чтобы придать блоку наибольшую функциональность, а также передать значения команд блока другим командам.

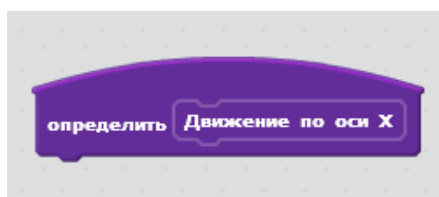
Рассмотрим пример использования *Другого блока* в скрипте.

В 1 уровне для *Марио1* были заданы команды управления движением (направо и влево).

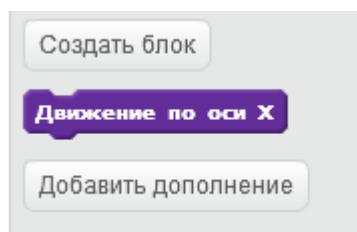


При помощи *Других блоков* можно упростить (оптимизировать) данный скрипт.

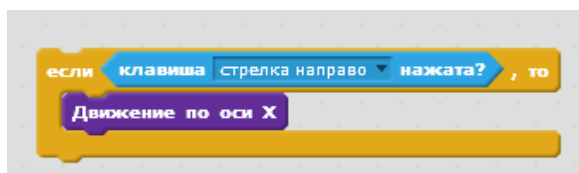
Создайте новый блок и назовите его, например, *Движение по оси X*. Никаких дополнительных параметров задавать пока не нужно. Щелкните **ОК**. Блок появится в области скриптов.



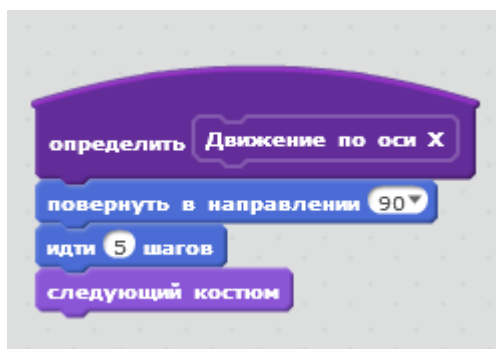
Команда самого блока отобразится среди других команд в *Других блоках*.



Данная команда должна заменить скрипт движения персонажа направо. Уберите, но не удаляйте, команды *Повернуть в направлении...*, *Идти... шагов* и *Следующий костюм*, а вместо них поставьте в условие *Если..., то* блок *Движение по оси X*.

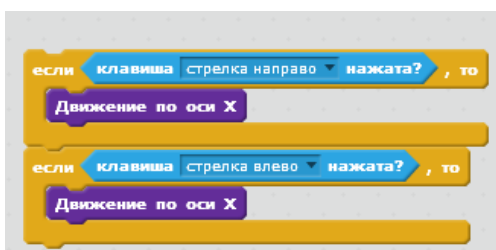


Вставленная в условие команда должна выполнять определенную функцию, заданную блоком. Запрограммируйте блок *Движение по оси X* с помощью команд. Т.к. блок создавался для оптимизации движения спрайта, следовательно, в него должны входить те блоки, которые он и его команда заменяют в скрипте. Прикрепите к блоку *Движение по оси X* команды *Повернуть в направлении...*, *Иди... шагов* и *Следующий костюм*.



Теперь при нажатии стрелки направо Марио будет двигаться в нужном направлении.

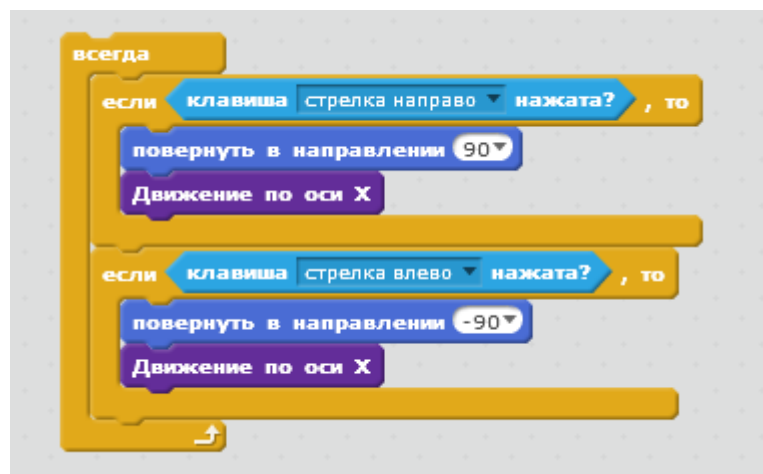
Блок был назван *Движение по оси X*, следовательно, команда предусматривалась для движения направо и влево и должна подходить для другого условия. Если поставить команду блока в условие движения влево, то при нажатии соответствующей клавиши Марио будет двигаться вправо.



Это происходит потому, что блок *Движение по оси X* запрограммирован на поворот в правую сторону и на дальнейшее движение. Необходимо задать поворот влево. Есть несколько вариантов. Например, можно задать направление движения через переменную, которая при нажатии на клавишу *Стрелка влево* (или *Стрелка направо*) будет принимать определенные значения, которые будут передаваться в команду *Повернуть в направлении*.



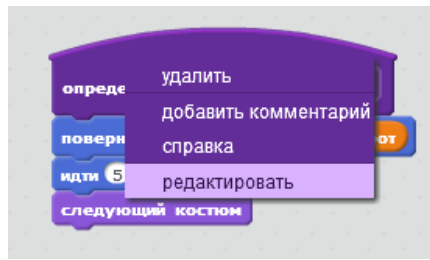
Также возможен другой вариант. Уберите команды *Повернуть в направлении...* из блока *Определить движение по оси X* и верните обратно в условие.



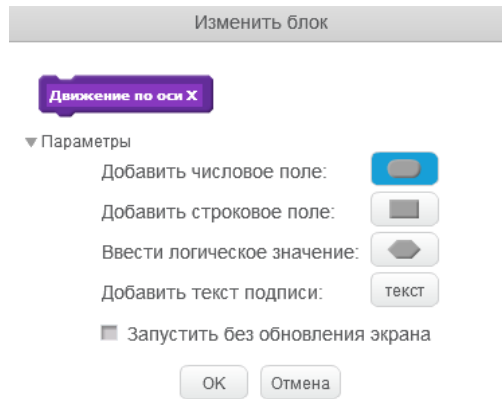
В выше указанных примерах не происходит оптимизация скрипта с помощью *Других блоков*, а, наоборот, происходит его увеличение.

Если еще раз посмотреть на примеры решений и подумать над тем, как задать движение влево и направо в одном блоке, то можно прийти к следующему заключению. Движение влево и направо являются компенсирующими друг друга движениями, а значит, не могут быть в одном блоке. Следовательно, необходимо какое-то значение, которое будет при нажатии «стрелок» задавать направление команде в блоке *Движение по оси X*. Блоку не нужно изначально задавать определенное направление направо или влево. Блоку необходимо лишь знать значение направления, которое будет передаваться команде направления в блоке, при этом не создавая лишних строк или переменных (команды *Задать... значение...* и *Изменить... на...* внесут в скрипт проекта лишние строки и переменные).

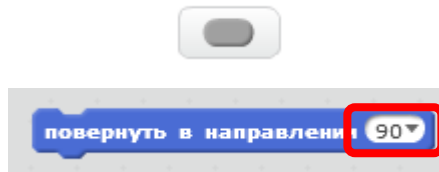
Чтобы оптимизировать скрипт, добавьте определенный *Параметр* («внутреннюю puzzle форму») для блока *Движение по оси X*. В области скриптов в контекстном меню блока выберите *Редактировать*.



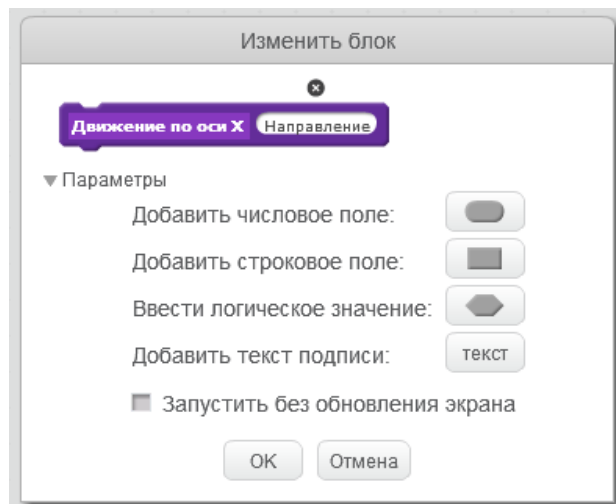
В диалоге *Изменить блок* щелкните *Параметры – Добавить числовое поле*.



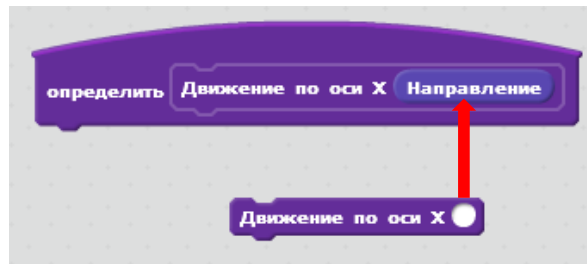
Новый параметр будет отвечать за направление движения спрайта, т.е. «работать напрямую» с командой *Повернуть в направлении...* Параметр должен совпадать с «внутренней puzzle формой» блока для дальнейшей интеграции.



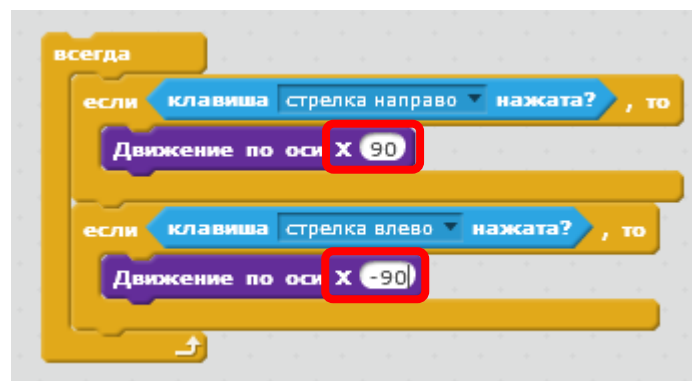
После выбора параметра введите его название (например, *Направление*) и нажмите *OK*.



В области скриптов в блоке *Определить движение по оси X* отобразится новый параметр *Направление*. В самом скрипте *Марио1* в блоке *Движение по оси X* появится поле, куда вводится значение параметра *Направление* (значение, которое интегрируется в это поле, будет передано параметру *Направление*).



Чтобы параметр *Направление* принимал при нажатии «стрелок» нужное значение, необходимо сначала задать эти значения. В данном случае значения задаются в блоках *Движение по оси X*, которые находятся в скрипте движения. Для движения направо спрайту необходимо повернуть на 90 градусов, следовательно, в пустом поле блока введите значение *90*. Для движения влево введите *-90*.



Блок *Повернуть в направлении...* пока «не знает», в какую сторону поворачивать спрайт, потому что параметр *Направление* не передает ему значения.

В блоке *Определить движение по оси X* зафиксируйте курсором параметр *Направление* и перетащите его в команду *Повернуть в направлении...*

