

Ветвление

Тимофеева Т.В., педагог ДО ККДП

Ветвление. Сравнение чисел, переменных

Истина или ложь?

Компьютеры принимают решения о том, что делать дальше, задавая вопросы и анализируя ответы «Да» или «нет». Истина или ложь? Вопросы, на которые есть лишь два варианта ответа, называют булевыми (логическими) выражениями.

Сравнение чисел. Числа можно сравнивать с помощью блока «=» из секции Operators («Операторы») палитры блоков.



Этот блок дает один из двух вариантов ответа: «истина», если числа равны, и «ложь», если не равны

Ветвление. Сравнение чисел, переменных

Используй блок «=» внутри блока say («сказать»), чтобы увидеть ответ true («истина») или false («ложь») в баллоне над спрайтом.

Если числа отличаются, в баллоне над спрайтом появится слово false («ложь»).



Числа равны, так что
в баллоне появится
true («истина»)

Эти числа не равны,
поэтому в баллоне
будет false («ложь»)

Ветвление. Сравнение чисел, переменных

Сравнение переменных. Внутри блоков сравнения можно использовать переменные: зачем сравнивать фиксированные числа, если результат всегда один и тот же?



Создаем переменную. Кликни по кнопке Data («Данные») в палитре блоков и создай переменную с именем age («возраст»). Задай ей значение 10 (кликни по блоку, чтобы убедиться, что значение поменялось). Перетаскивай эту переменную в блоки сравнения.

Это знак «равно»: блок спрашивает, правда ли, что в age число 7. Ответ будет «ложь», так как в age число 10



Это знак «больше»: блок спрашивает, правда ли, что age больше, чем 11. Ответ будет «ложь», ведь 10 не больше, чем 11.

Это знак «меньше»: блок спрашивает, правда ли, что age меньше, чем 18. Ответ будет «истина», ведь 10 меньше 18.

Ветвление. Сравнение чисел, переменных.

Сравниваем числа

Зеленые блоки сравнения находятся в секции Operators («Операторы»). Можно выяснить, являются ли числа равными или одно больше другого.

Сравнение слов.

Блок «=» подходит не только для чисел можно узнать, совпадают ли две строки. Разницу между заглавными и строчными буквами этот блок не учитывает.



Создаем переменную. Чтобы поупражняться в сравнении строк, создай новую переменную с именем name («имя») и задай ей значение Kate.

Ветвление. Сравнение чисел, переменных.

«Не»

Блок `not` («не») пригодится, когда надо поменять значение булева выражения на противоположное. Например, проще проверить `age` на не равенство 10, чем на равенство каждому из остальных чисел.



Создаем переменную. Чтобы поупражняться в сравнении строк, создай новую переменную с именем `name` («имя») и задай ей значение `Kate`.

Без блока `not`. Числа 10 и 7 различны, так что ответ будет «ЛОЖЬ».

С блоком `not`. Если добавить блок `not`, результат изменится. Поскольку 10 не равно 7, теперь ответ будет «истина».

Ветвление. Сравнение чисел, переменных.

Объединение вопросов

Чтобы задавать более сложные вопросы, состоящие из нескольких простых, блоки сравнения можно объединять.



Ответ будет «истина», если ответ на любой из двух вложенных вопросов – «истина»

Ответ будет «истина», только если ответ на каждый из вложенных вопросов – «истина»

Пример. Верхний блок проверяет что age меньше 18 или больше 65. Нижний блок проверяет, что age равна 11, 12, 13 или 14.



Блоки сравнения. Блоки or («или») и and («и») нужны, чтобы по-разному объединять булевы выражения.

Ветвление. Сравнение чисел, переменных.

Сведения об истинности или ложности можно использовать, чтобы объяснить компьютеру, что делать дальше. Он выполнит разные действия в зависимости от того, был ли ответ «истина» или «ложь».

Принятие решений. Блоки `if` («если») используют булевы выражения, чтобы решить, что делать дальше. Если поместить внутрь блока `if` другие блоки, они будут выполняться, лишь когда булево выражение в заголовке `if` дает истину.

Ветвление. Сравнение чисел, переменных.

Перетащи сюда
булево выражение



Если выражение ложно,
выполнятся блоки из
набора else («иначе»)

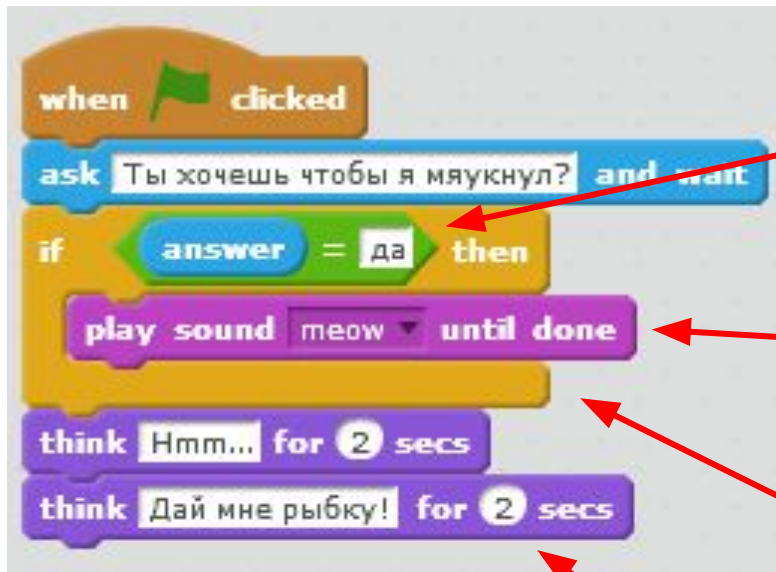
Блок if-then. Если булево
выражение истинно, блоки
внутри if-then («если, то»)
выполняются.

Блок if-then-else. «Если, то,
иначе»: если булево выражение
истинно, выполнится первый
набор блоков, иначе - второй
набор.

Использование блока if-then. Блок if-then позволяет на основе значения булева выражения решить, нужно ли выполнять часть скрипта. Чтобы попробовать этот скрипт, добавь его к спрайту кота.

Ветвление. Сравнение чисел, переменных.

Мяукающий кот. Эта программа провернет значение булева выражения и запускает блоки внутри if-then, если это значение истинно. Кот мяукает, только если его об этом попросить.



Перетащи блок «=» из секции Operators («Операторы») в заголовок блока if-then («если, то»). Затем помести в «=» переменную answer («Ответ») из секции Sensing («Сенсоры»)

Эти блоки находятся внутри if-then («если, то» - кот мяукает, только если на вопрос был дан ответ «Да»)

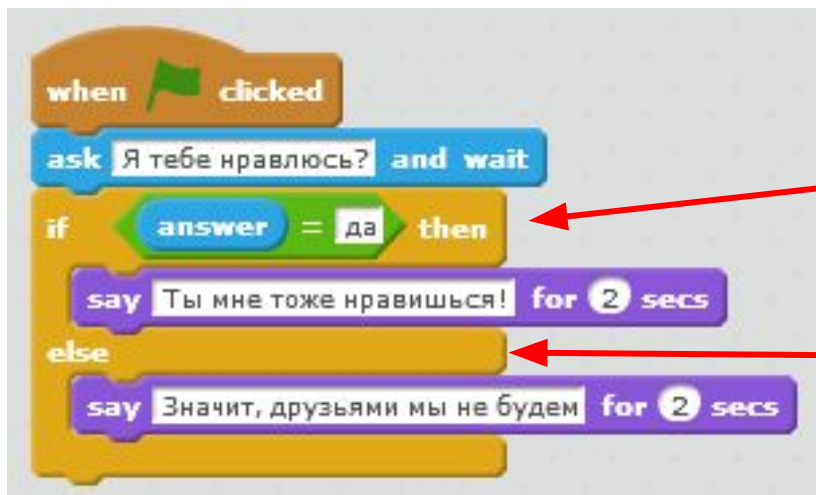
Конец блока if-then («если, то»)

Эти блоки think («думать»)- за пределами блока if-then («если, то»), а значит, они будут выполнены в любом случае.

Ветвление. Сравнение чисел, переменных.

Часто нужно, чтобы программа сделала что-то одно, если условие выполняется, и что-то другое, если нет. Блок if-then-else («если, то, иначе») задает два варианта исполнения программы, которые называют ветвями. В зависимости от ответа на булево выражение запустится лишь одна из ветвей.

Ветвление программы. У этой программы две ветви: если ответ был yes («да»), запустится одна ветвь, иначе – другая.



Если ответить «Да», запустится эта ветвь

Если ответить что угодно, кроме «Да», запустится эта ветвь

Управление с клавиатуры.

В Scratch у блоков булевых выражений заостренные края, однако в лунки другой формы их тоже можно перетаскивать.



Блоки Sensing. Эти блоки проверяют, не нажата ли кнопка мышки.

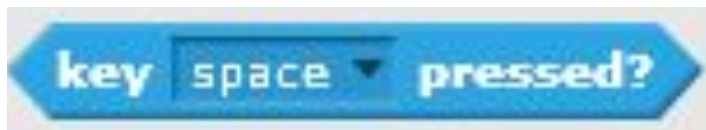


Блоки Control. В некоторых блоках Control («Управление») есть лунки заостренной формы для булевых выражений.

Блоки из секции Sensing («Сенсоры») рассказывают скрипту, что происходит с компьютером. Они могут считывать нажатия клавиш и позволяют спрайтам реагировать на столкновения.

Управление с клавиатуры.

Управление с клавиатуры. С помощью блоков Sensing («Сенсоры») и блока if-then можно двигать спрайт по экрану, управляя им с клавиатуры. В меню блока key pressed? («мышка нажата?») есть большинство клавиш клавиатуры, и спрайт сможет на них реагировать. Также можно выполнять действия при нажатии кнопок мышки.



Этот блок проверяет, была ли нажата клавиша. Какая именно, можно выбрать из меню.



Этот блок проверяет, была ли нажата кнопка мышки

Блоки Sensing. Используя эти блоки с блоком if-then, можно узнать, была ли нажата кнопка мышки или клавиша на клавиатуре.

Управление с клавиатуры.

Клавиатура подходит для точного управления спрайтами - как раз то, что нужно для игр



Благодаря тому, что все заключено в блок forever («всегда»), скрипт постоянно проверяет нажатия клавиш.

Проверяет, нажата ли стрелка вверх, и, если нажата, двигает спрайт вверх.

Управление с клавиатуры.

Столкновения спрайтов. Бывает важно знать, когда спрайты соприкасаются один с другим - например, в играх. Используйте блоки Sensing («Сенсоры»), чтобы реагировать на столкновения спрайтов или на касание спрайтом области заданного цвета.




Этот блок определяет, что спрайт столкнулся с другим спрайтом.

Этот блок определяет, что спрайт столкнулся с областью заданного цвета.

Использование блоков Sensing. Сделаем из нашего управляемого кота настоящую игру.

Управление с клавиатуры.

В этом скрипте блоки Sensing («Сенсоры») используются, чтобы управлять взаимоотношениями слона и кота. Чем ближе кот подходит к слону, тем больше становится слон. Если кот прикоснется к слону, слон сменит костюм, затрубит в хобот и убежит в другое место. К какому спрайту писать данный скрипт?



The image shows a Scratch script starting with a 'when clicked' event block. It enters a 'forever' loop. Inside the loop, the first block is 'set size to 200 - distance to Sprite1 %', where 'Sprite1' is selected from a dropdown. Below this is an 'if touching Sprite1?' block. If true, it executes three blocks: 'switch costume to elephant-b', 'play sound pop until done', and 'switch costume to elephant-a'. Finally, it goes to a 'go to x: pick random -240 to 240 y: pick random -180 to 180' block. Red arrows point from text annotations to these specific blocks.

Цикл forever («всегда») непрерывно проверяет позиции спрайтов и обновляет размер и координаты слона

Проверяет расстояние от кота до слона

Чем кот дальше, тем меньше слон

Если спрайты столкнутся, запустятся блоки внутри блока if-then

Этот блок выбирает для слона случайные координаты