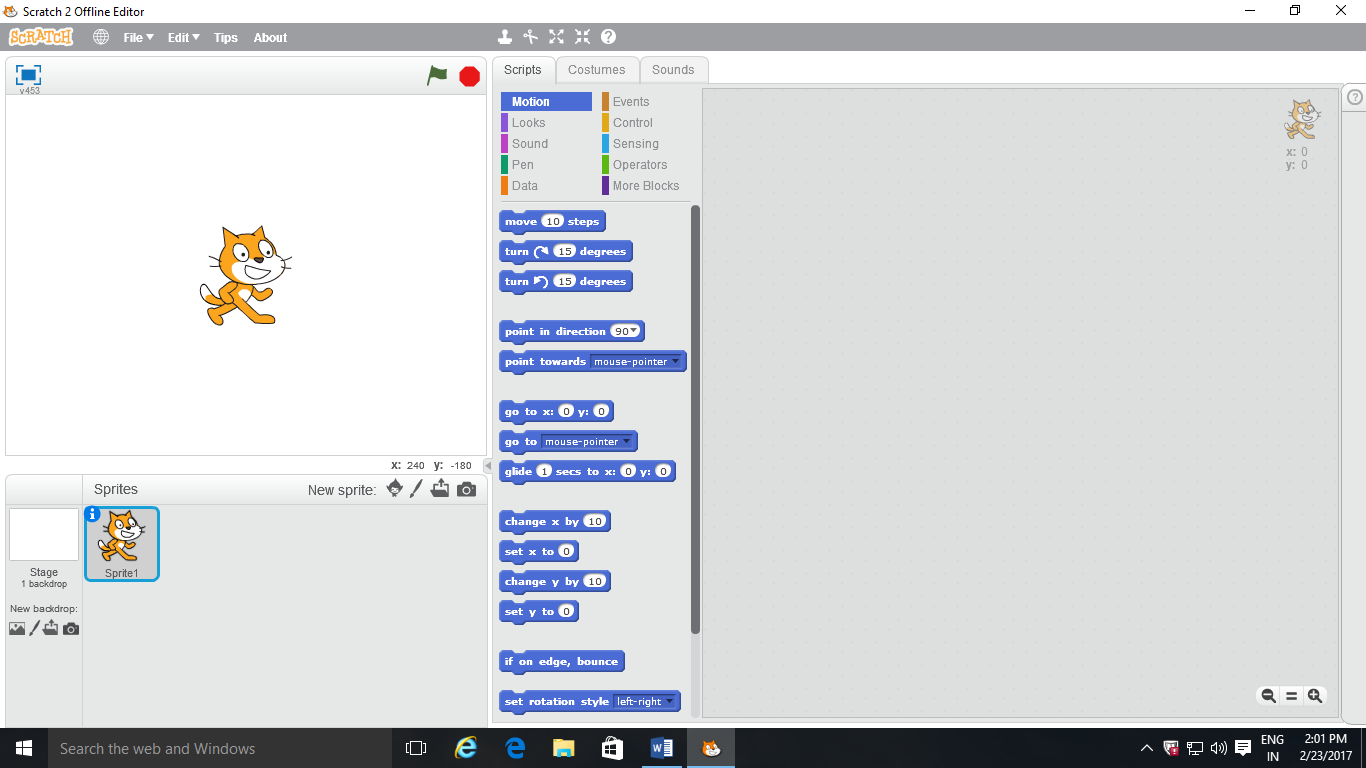
Scratch

What is scratch?

Scratch is a free visual programming language developed by the MIT Media Lab. Scratch is used by students, scholars, teachers, and parents to easily create animations, games, etc. It provides a stepping stone to the more advanced world of computer programming.



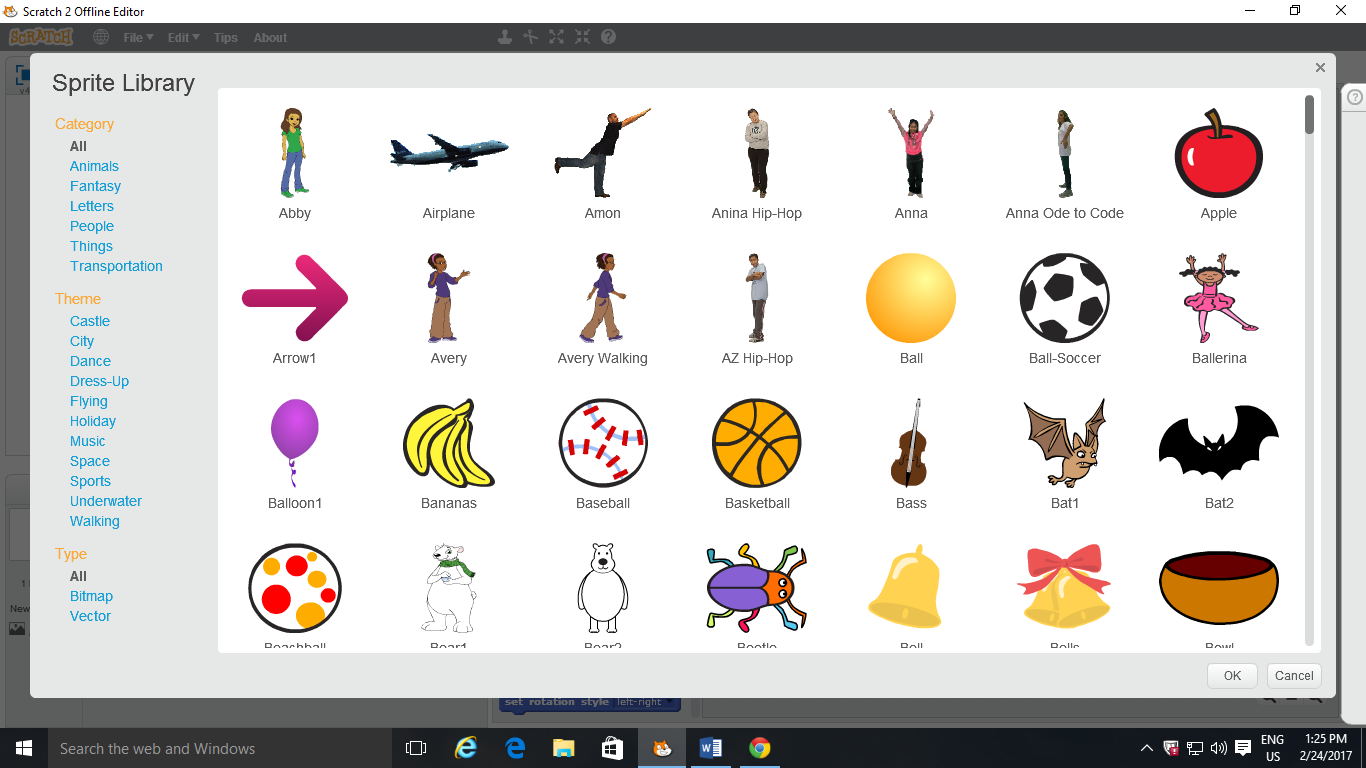
What is blocks?

Blocks are used to create code in Scratch. The blocks connect to each other like a jigsaw puzzle - this prevents syntax errors. There are eight categories of blocks: Control, Motion, Looks, Sound, Pen, Sensing, Operators and Variables.

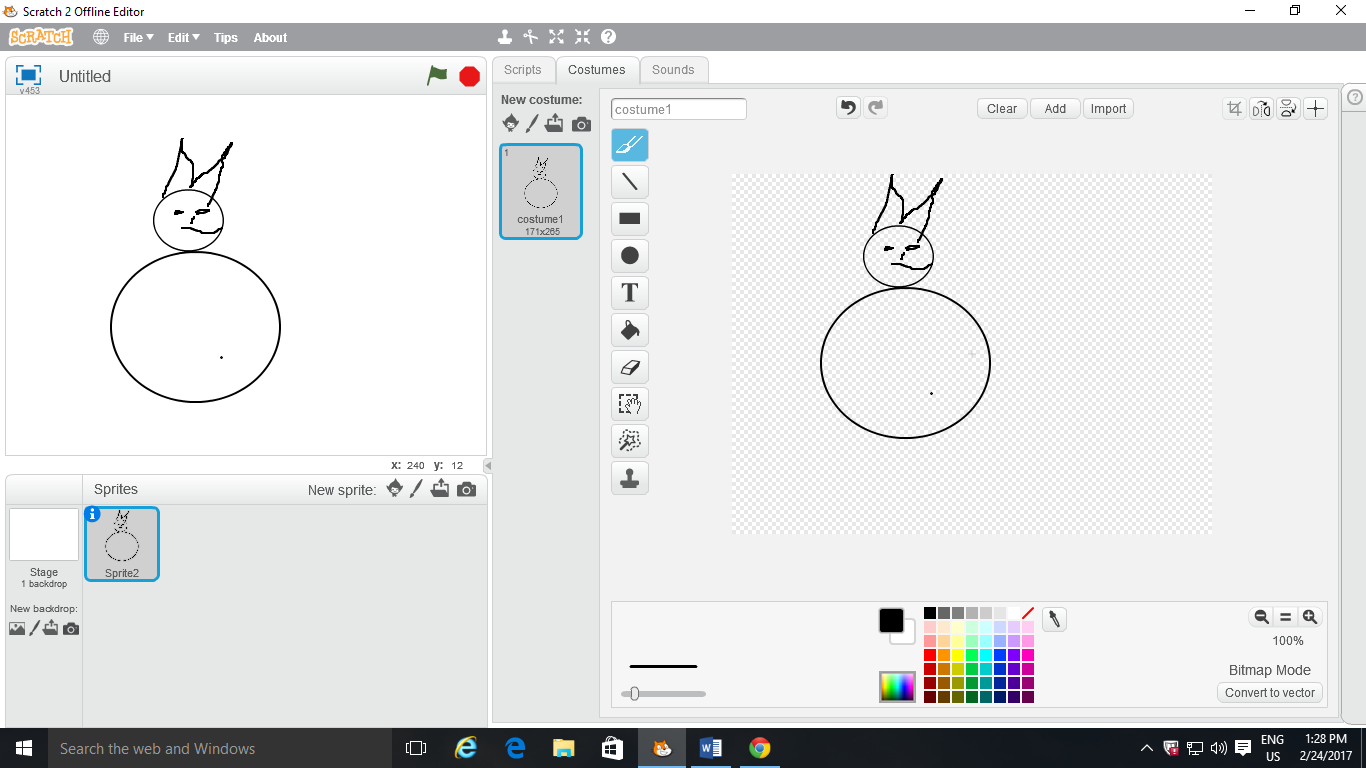
What is sprites?

 Sprite — An object in Scratch which performs functions controlled by scripts. In sprites there are different options like choose sprite from library, paint a new sprite, upload sprite from file, new sprite from camera.

**Choose sprite from library,**



**Paint a new sprite,**

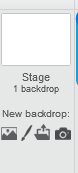


In **Upload sprite from file**, we can upload many images from your file which we need.

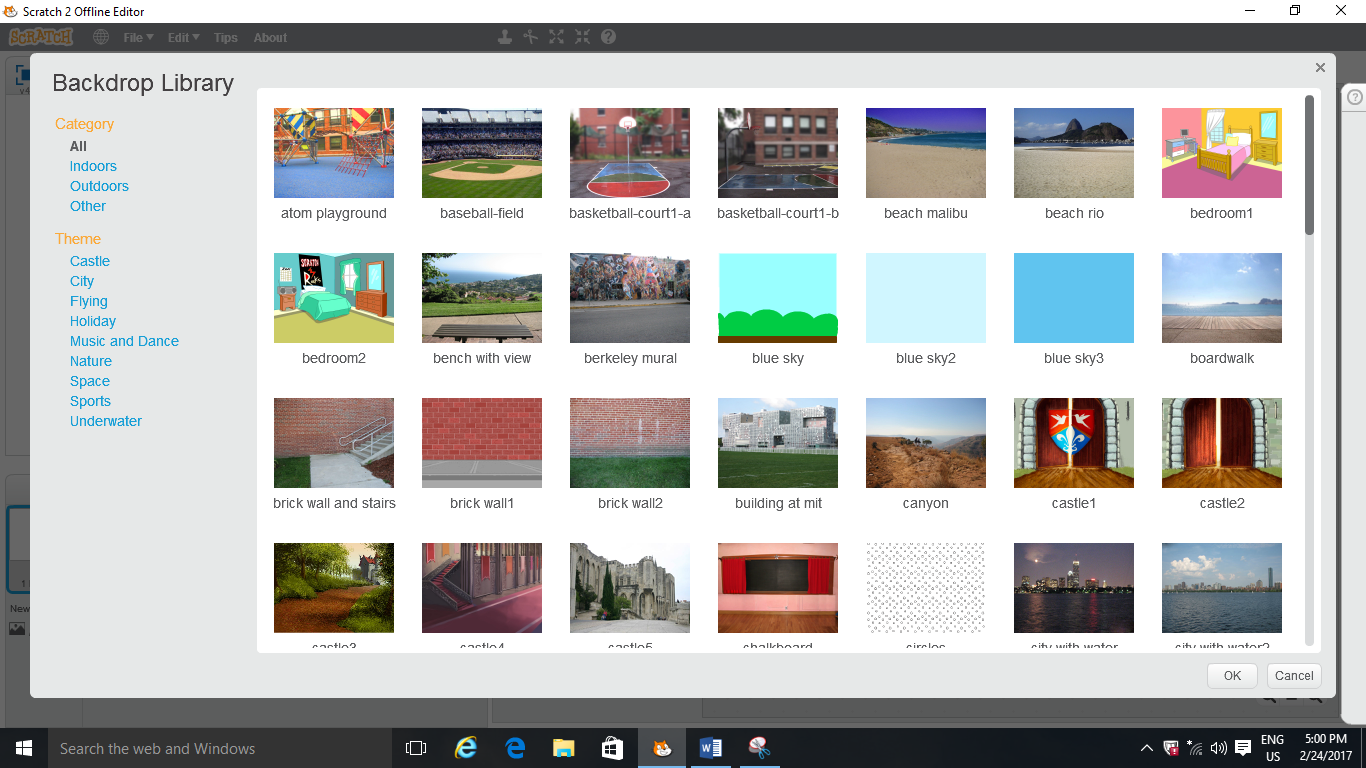
In **new sprite from camera**, we can make a sprite move with a camera on Scratch

What is stage backdrop?

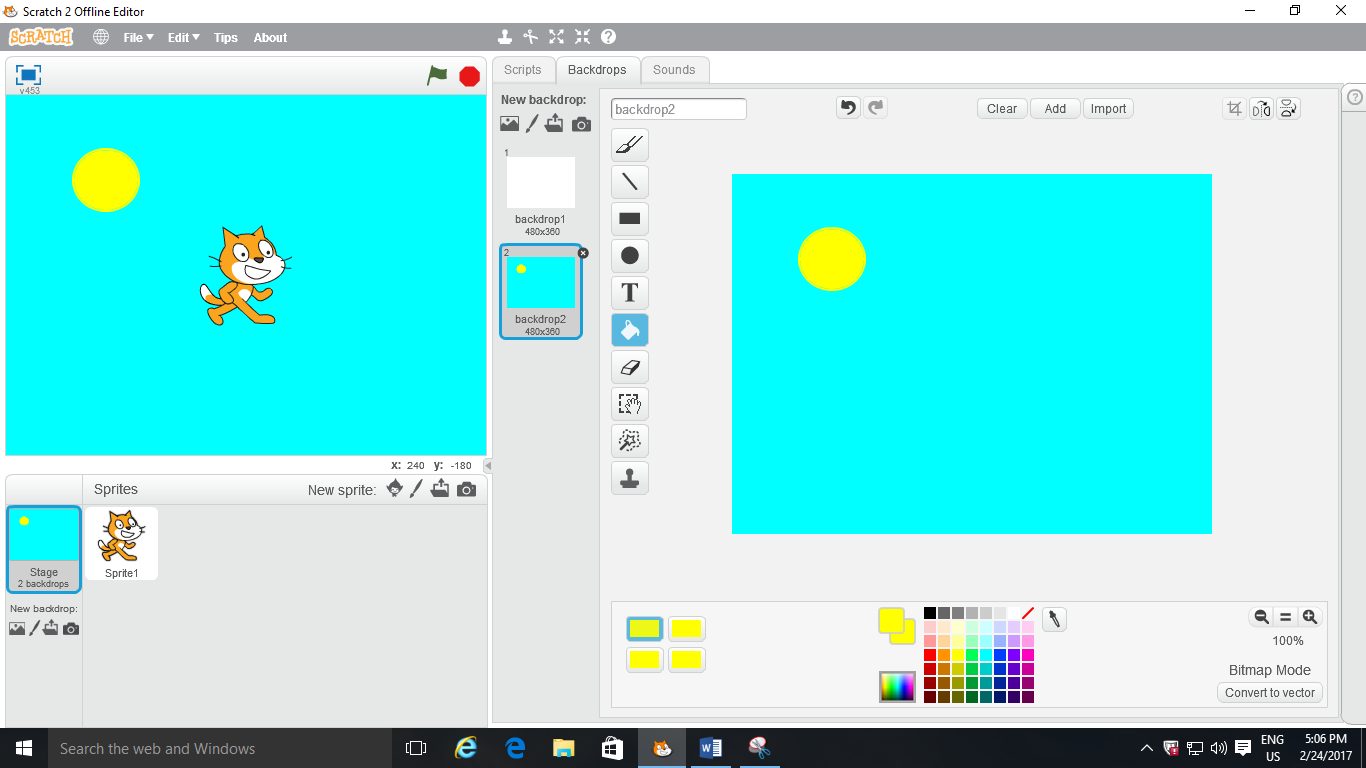
Stage backdrop is an option in which we can change the background. In stage backdrop there are different option like choose backdrop from library, paint a new backdrop, upload a backdrop from file ;and new backdrop from camera.



**Choose backdrop from library.**



**Paint a new backdrop**

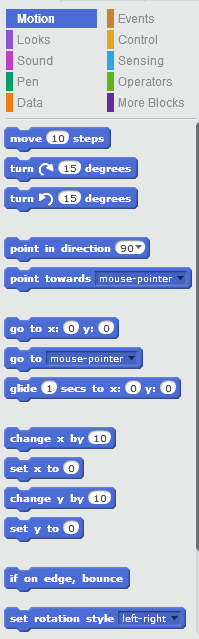


**In upload a backdrop from file,** we can upload images from file.

**New backdrop from camera**, we can make a backdrop move with a camera on Scratch.

**Scripts**

The sprite will move in the current direction. Type in how far you want it to move.





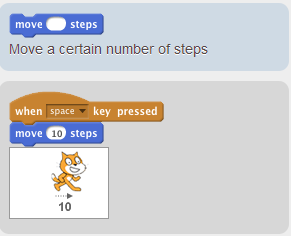
**Motions**

**Move ten steps**



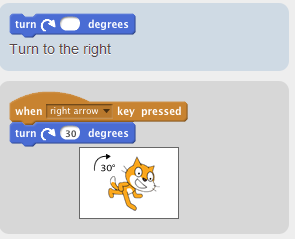
In move ten steps we can move a sprite. If you type in a negative number (such as -10), the sprite will go in the opposite direction.

How to move a sprite with Move ten steps?



**Turn a sprite right**

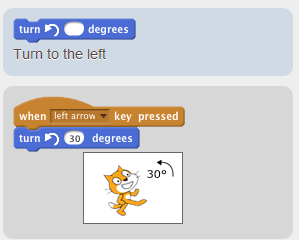


 If you type in a negative number, the sprite will go in the opposite direction.

**Turn a sprite left**



If you type in a negative number, the sprite will go in the opposite direction.



**Point in direction**

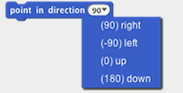


Set the direction of the current sprite.

**How to do?**

\*Set the direction of the current sprite.

\* Click to choose from the menu.



\*Or, type in a number.



**Point towards mouse pointer**:

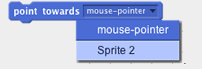


Points sprite towards mouse-pointer or another sprite.



point towards changes the direction of the current sprite.

You can choose from the menu:



The menu includes any other sprites in the project.

**Go to x: 0 y: 0**

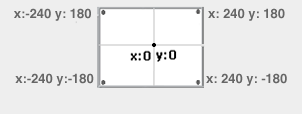


Go to this position on the Stage.



Type in a number for x and y to tell the sprite where to go to on the Stage.

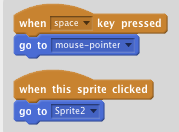
Here are the x y coordinates of the Stage:



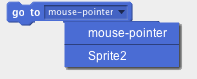
**Go to ( )**



Go to the location of the mouse-pointer or of another sprite.



Choose from the menu:

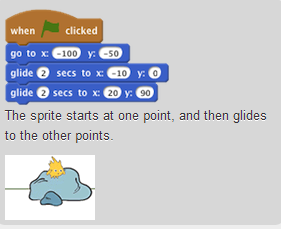


Then, click the block.

The current sprite will jump to the x y location of the mouse-pointer or of the sprite you choose.

**Glide\_secs to x:\_y:\_**





Glide over time to this x y position.

Use glide to move a sprite smoothly to an x y position on the Stage.

You can see x and y update in the Motion blocks palette when you drag a sprite to another point.

**Change x by\_**



Change the x position by this amount.



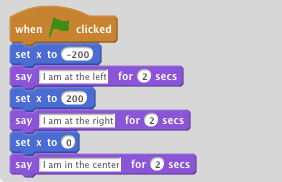
A positive number shifts the sprite to the right.

A negative number shifts the sprite to the left.

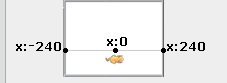
**Set x to\_**



Set the x position of a sprite.



Set x places a sprite across the Stage (horizontal position).



**Change y by**



Changes sprite's y-position by specified amount.



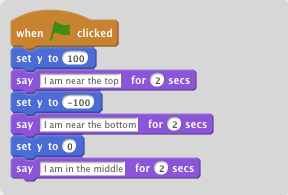
Change y by a positive number to move the sprite up.

Change y by a negative number to move the sprite down.

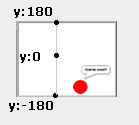
**Set y to**



Set the y position of a sprite.

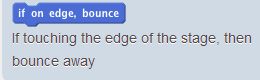


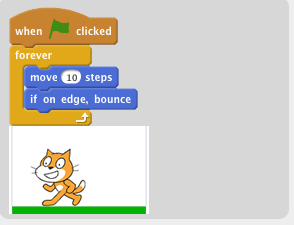
Set y places a sprite up or down (vertical position).



**If on edge, bounce**







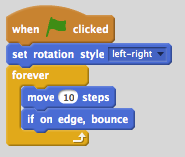
The sprite will bounce at an angle if it is touching the side, top, or bottom of the stage.

**Set rotation style**



Sets the sprite's rotation style

Select "left-right" from the drop-down menu to make the sprite only rotate horizontally.



Select "all around" from the drop-down menu to make the sprite flip vertically.



Select "don't rotate" from the drop-down menu to make the sprite face only one direction.

**X position**



Reports the x position of a sprite.



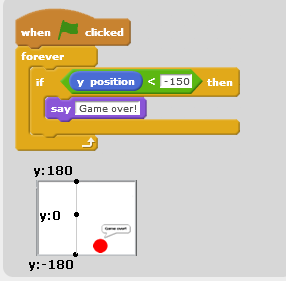
To display the sprite's x position on the Stage, click the checkbox next to the block.



**Y position**



Reports the y position of a sprite.



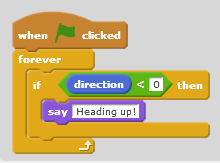
To see the sprite's y position on the stage, click the checkbox next to the block:



**Direction**



Reports the current direction of a sprite.



direction tells which way the sprite is heading.

To see a sprite's direction, click:

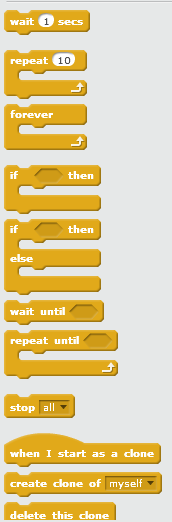


You can drag the blue line to change direction.

To display a sprite's direction on the Stage, click the checkbox in the palette:



**Control**



**Wait( ) secs**



Waits specified number of seconds, then continues with next block.



**Repeat ( )**



Runs the blocks inside a specified number of times.



In repeat, the sprite will repeat the action that, how many numbers you put.

**Forever**



Runs the blocks inside over and over.



In forever, the sprite repeat the action forever.

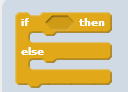
**If then( )**



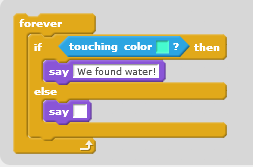
If condition is true, runs the blocks inside.



**If then( ) else**



If condition is true, runs the blocks inside the if portion; if not, runs the blocks inside the else portion.



**Wait until**



Waits until condition is true, then runs the blocks below.



This whole stack will just run once. If you want to keep checking, put the whole stack inside a forever block.

**Repeat until**



Repeat blocks that follow until condition is true.

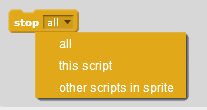
Checks to see if condition is false; if so, runs blocks inside and checks condition again. If condition is true, goes on to the blocks that follow.



**Stop**



Stops all scripts in all sprites



Same as clicking the stop button at the top of the screen.

**When I start as a clone**



Tells a clone what to do once it is created.

The script is triggered as soon as the clone is created. 

Clones also respond to all "when..." blocks triggered after they are created.

**Create clone of**



Choose which sprite to clone from the drop-down menu.

Use  to tell the clone what to do once it's created.

Sample Script:



Troubleshooting:

If you can't see the clone, move it so the original sprite doesn't cover it. The clone appears in the same location as the sprite initially.

Make sure you've chosen the sprite you want to clone from the menu in the

block.

Note: The clone only lasts while the project is running.

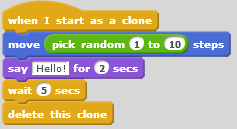
**Delete this clone**

Deletes the current clone

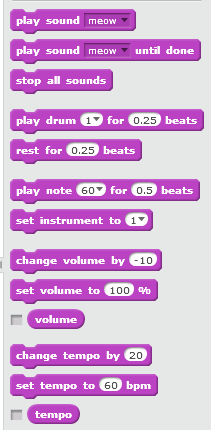
Place this block in a script after the clone is done its actions.

All clones are automatically deleted when the program is stopped.

Sample Script:



**Sound**

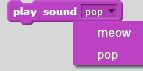


**Play sound**

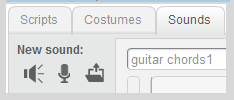




Choose a sound from the menu:



To add a sound, click the Sounds tab:



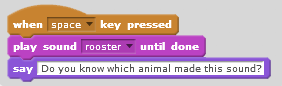
 starts playing a sound and continues to the next block immediately.

 waits until the sound is finished playing before continuing to the next block.

**Play sound until done**



Plays a sound and waits until the sound is finished.



starts playing a sound and continues to the next block immediately.

waits until the sound is finished playing before continuing to the next block.

**Stop all sounds**





Stop other sounds before playing a new sound.

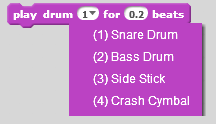
**Play drum ( ) for( ) beats**



Plays a drum sound for a specified number of beats.



Choose a drum sound from the list



Or, type in a number from 1 to 22



Note: the length of a beat can be set with



**Rest for ( ) beats**



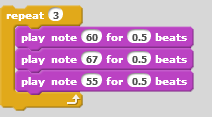
Rests (plays nothing) for specified number of beats.



**Play note( ) for ( ) beats**



Plays a musical note for specified number of beats.



Type in a number from 0 to 127 (60 is middle C)



Use higher numbers for higher pitches

The length of a beat can be set with



**Set instrument to ( )**

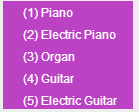
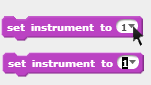


Sets the type of instrument that the sprite uses for play note blocks



Each sprite has its own instrument.

Click to choose from the menu. Or, type in a number from 1 to 21.



**Change volume by ( )**



Changes sprite's sound volume by specified amount



Volume ranges from 0 to 100, with 100 as the default volume.

You can set the volume level for each sprite. To play two sounds at once with different volumes, use two sprites.

**Set volume to ( ) %**



Sets sprite's sound volume to specified value



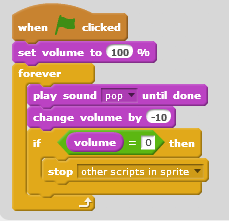
You can set the volume level for each sprite.

To play two sounds at once with different volumes, use two sprites.

**Volume**



Reports sprite's sound volume.

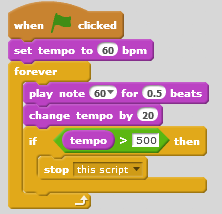


You can set the volume level for each sprite. To play two sounds at once with different volumes, use two sprites.

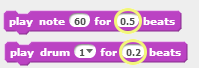
**Change tempo by ( )**



Changes sprite's tempo by specified amount



Tempo is the speed (bpm = beats per minute) at which Scratch notes and drums play. The larger the tempo value, the faster the notes and drums will play.



**Set tempo to ( )**



Sets sprite's tempo to specified value in beats per minute.



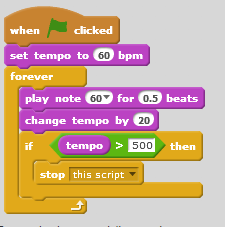
Tempo is the speed (bpm = beats per minute) at which Scratch notes and drums play. The larger the tempo value, the faster the notes and drums will play.



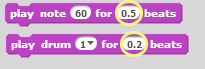
**Tempo**



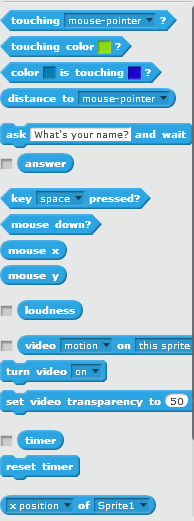
Reports sprite's tempo in beats per minute



Tempo is the speed (bpm = beats per minute) at which Scratch notes and drums play. The larger the tempo value, the faster the notes and drums will play.



**Sensing**





**Touching ( )?**



Reports true if sprite is touching specified sprite, edge, or mouse-pointer.



You can use the block in three different ways. Select from the pull-down menu to choose. You can check if the sprite is:



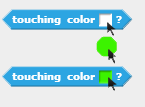
Touching color ( )?



Reports true if sprite is touching specified color.



To choose a color, get the eye dropper by clicking in the square. Use the eye dropper to click on the color you want. The color will appear in the square.



**Color ( ) is touching ( )?**

Reports true if first color is touching second color

The first color is within the sprite, while the second is from the background or another sprite.



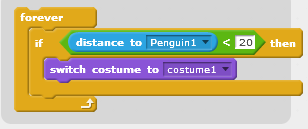
To choose a color, get the eye dropper by clicking in the square. Use the eye dropper to click on the color you want. The color will appear in the square.



**Distance to**



Reports distance from the specified sprite or mouse-pointer



Select from the pull-down menu to choose

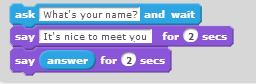


**Ask ( ) and wait**



Asks a question on the screen and stores keyboard input in the

After receiving an answer, causes the program to wait until the Enter key is pressed or check mark is clicked.



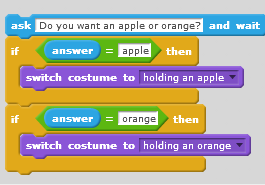


asks a question and stores the keyboard input in . The question appears in a voice balloon on the screen. The program waits as the user types in a response, until the Enter key is pressed or the check mark is clicked.

**Answer**



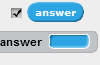
Reports keyboard input from the most recent use of



 asks a question and stores the keyboard input in . The answer is shared by all sprites.

If you want to save the current answer, you can store it in a variable or list. For example, 

To view the value of answer, click the checkbox next to the answer block.



**Key pressed**



Reports true if specified key is pressed



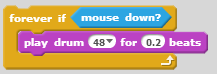
Use  instead of 

when you want to keep, for example, the space key pressed continuously.

**Mouse down?**



Reports true if mouse button is pressed.

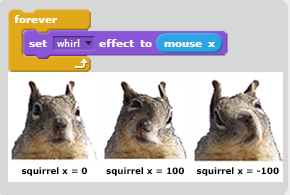


reports true if the mouse button is clicked anywhere on the screen.

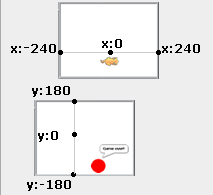
**Mouse x**



Reports the x-position of mouse-pointer.



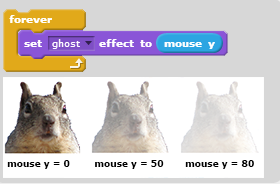
You can move the mouse around to see the position of the mouse pointer.



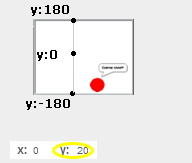
**Mouse y**



Reports the x-position of mouse-pointer



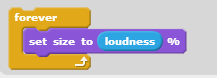
You can move the mouse around to see the position of the mouse pointer.



**Loudness**



Reports the volume (from 1 to 100) of sounds detected by the computer microphone.



To view the value of size, click the checkbox next to the loudness block.



Note: Your computer must have a working microphone in order for this block to work.

**Video ( ) on ( )**



Senses how much motion or direction is currently in the video image.

How to Use It:

Requires webcam.

Use to check the amount of motion in the video under the current sprite.

Use  to check the amount of motion over the entire video image.

Use to check the direction of motion in the video under the current sprite.

Use  to check the direction of motion in the video image.

Sample Script:

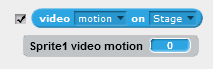


Troubleshooting:

Make sure the video camera is on.

Adjust the % of motion or direction to get it to respond more or less.

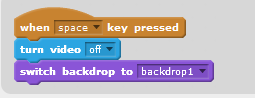
Click the checkbox to see the current value:



**Turn video**



Turns the video camera on

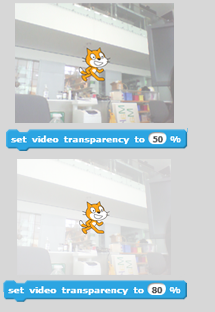


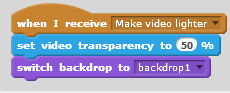
**Set video transparency to ( ) %**



Sets the video transparency

Enter a number from 0-100. Higher numbers make the video more transparent (lighter), while lower numbers make it less transparent (darker)





**Timer**



Reports the value of the timer in seconds.



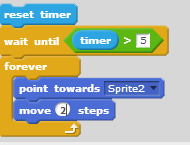
To see the value of timer, click the checkbox next to the block. The timer is always running.



**Reset timer**

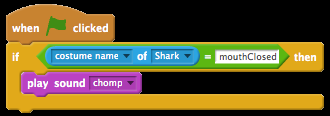


Sets the timer to zero



**( ) of ( )**



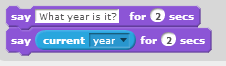


Choose different attributes from the drop-down menu and combine with other blocks, such as operators.

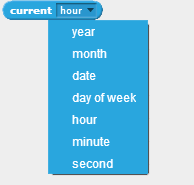
**Current**



Reports the current time



You can select from the menu to indicate which item you want.



To view the value of the current time, click the checkbox next to the current time.



**Day since 2000**

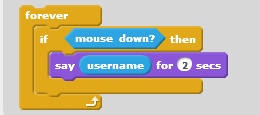


Reports the number of days since 2000

**User name**



Reports username of the viewer



This block will show the username of the user currently viewing the project.

If you want to save the current username, you can store it in a variable or list. For example,

.

**More blocks**

**Make a block**



Creates a custom block

Click "Make a Block" to get this dialog box.



Click inside the block to edit its name.

When you click OK, the new block will appear in : 

A define block will appear in Scripts. Use define to tell the custom block what to do:



You can create a custom block with inputs, for example

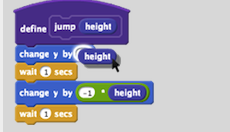


Click "Make a Block" to see this dialog box



Click Options to add inputs. For example, add a number input:



In the define block, inputs will appear as rounded, light purple blocks. To use them, drag copies of the input blocks into other blocks within the define script.  


Note: The input blocks can only be used in the define script.

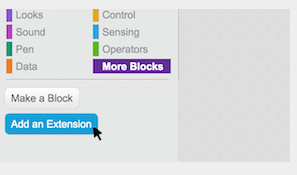
**Add a extension**



Add blocks that extend what you can do in Scratch

Extensions allow you to connect to hardware devices or web services.

In any project, click the More Blocks category and choose Add an Extension.

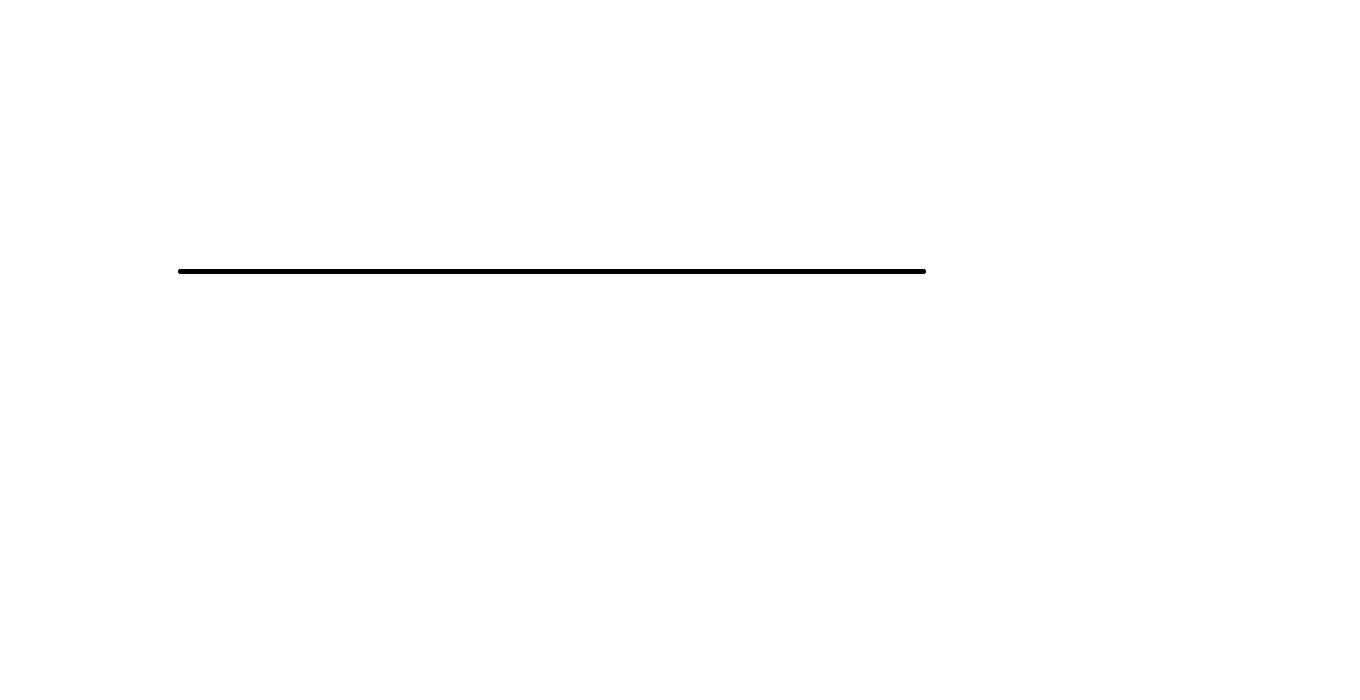


In the future, more extensions will be added to Scratch. To try out experimental extensions, visit the website scratchx.org

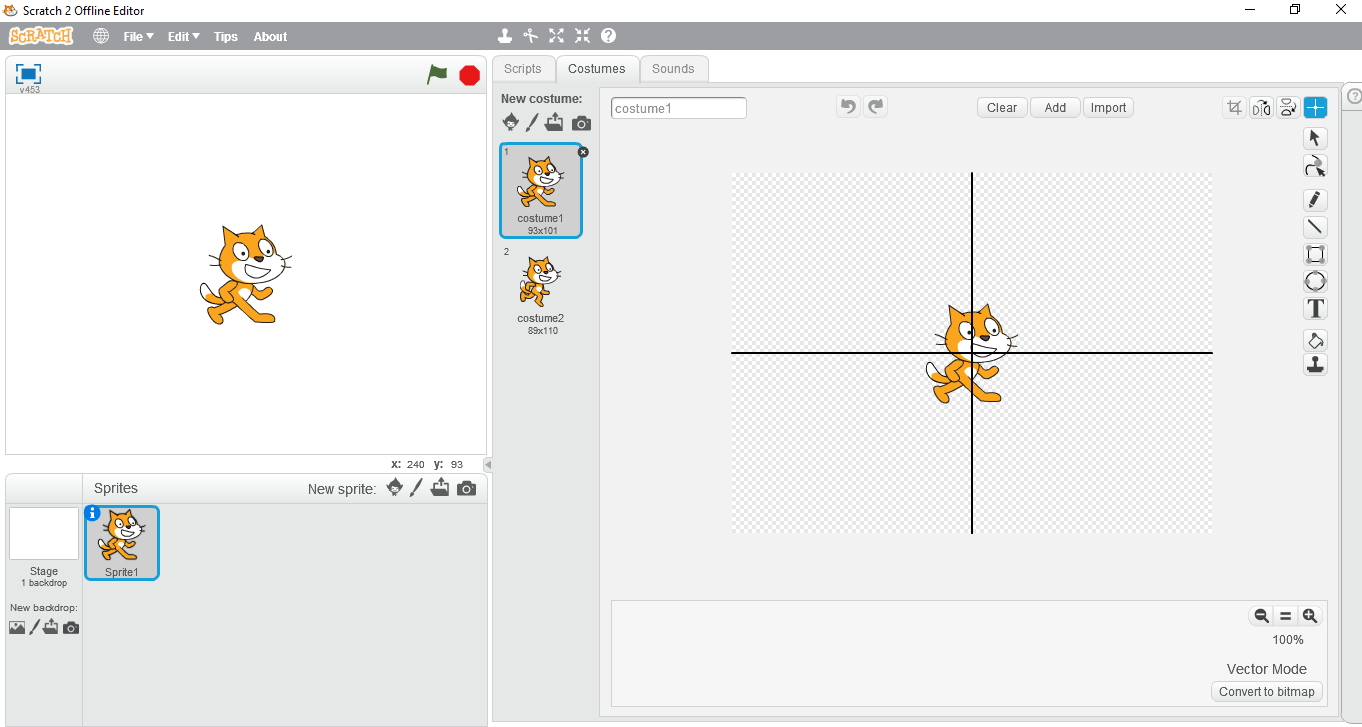
Hardware Devices

To use LEGO WeDo 2.0, see LEGO WeDo 2.0 Setup.

TO use LEGO WeDo 1.0 or a PicoBoard, see LEGO WeDo 1.0 or Picoboard Setup.



**Costumes**



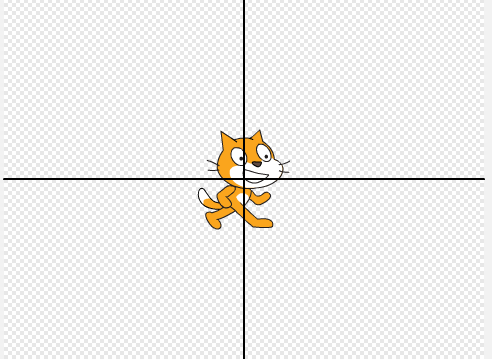
**What is costumes?**

A costume is one out of possibly many "frames" or alternate appearances of a sprite. Sprites can change their look to any of its costumes. They can be named, edited, created, and deleted, but every sprite must have at least one costume. One of the most common uses of costumes is to make an animation for a game or other project. In costumes there are different options. Lets see!!!!!!!!!!!!!!!!!

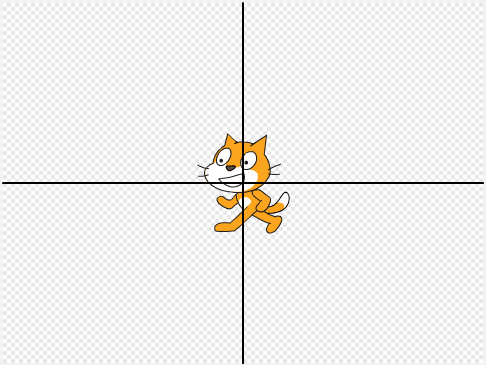


\*Flip left to right



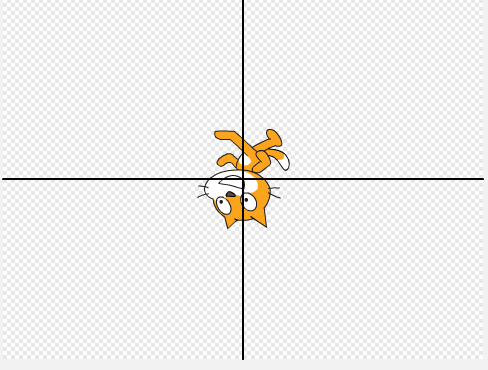


After clicking this option the sprite to right.

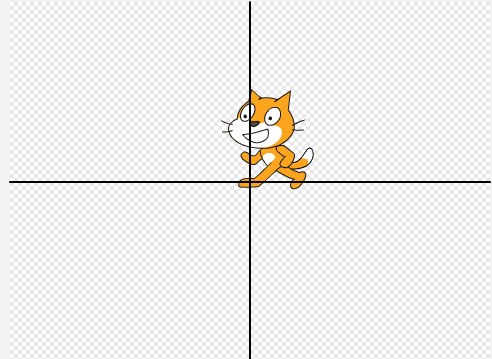


\*Flip up-down

After clicking this option the sprite will turn down.



Set costume center



In set costumes center, the plus point will