

**CS4HS**  
**ECS : Scratch Programming**  
**Summer 2017**

***Chuck Arbutina***  
***arbuticg@buffalostate.edu***

# **Unit 4 : Introduction to Programming**

- ECS pedagogical approach overview
- Getting Started with Scratch
- Dialogues between sprites
- Methods of moving sprites
- Event-driven programming
- Broadcasting through role play
- Variables

# Unit 4 : Intro. to Programming (Con't)

- Conditionals
  - The if block
- And, Or and randomness
- Creating a timer
- Games
  - Monkey Game
  - Pinball Game

# ECS Lesson Strategies

- Journaling
- KWL chart (Know, Want to Learn, Learned)
- Collaborative work (we will work in pairs)
- Charting
- Mapping

# Journal Entry

How do you think programs like Microsoft Word, Internet Explorer and Windows are made ?

# KWL Chart

| Know | Want to learn | Learned |
|------|---------------|---------|
|      |               |         |

# What is Scratch ?

- “Scratch is a programming language and online community where you can create your own interactive stories, games, and animations -- and share your creations with others around the world.

(<https://scratch.mit.edu/>)

# What is Scratch ? (con't.)

- Young people learn to think creatively, reason systematically, and work collaboratively when designing and programming Scratch projects.



# Where to access Scratch

- **Scratch is a free program that can accessed online**
- **Scratch can also be downloaded**

**Do either at :** <https://scratch.mit.edu/>

# How to access Scratch online

➤ Create an account :



Create stories, games, and animations  
Share with others around the world



A creative learning community with **23,508,508** projects shared

# Animations

## Two kinds of animations:

### ➤ **Movie**

- Passive user watches the animation

### ➤ **Interactive**

- Active user clicks on mouse, types a key on keyboard ...
- Actions of user are called **events**

# A simple story

- 🌐 A dancer would like to show off his skills by jumping up and dancing in mid-air. When he lands back on the ground he asks “Whoa, how was that ?”
- Nouns are characters/objects (sprites)
- Verbs are actions/methods (scripts)

# Step 1: Designing a digital story

- Determine the characters or objects needed in the story
- How are objects going to interact ?
- Determine the correct sequence of actions the objects must perform

# Step 2 : Writing an Algorithm

Dancer moves up in mid-air

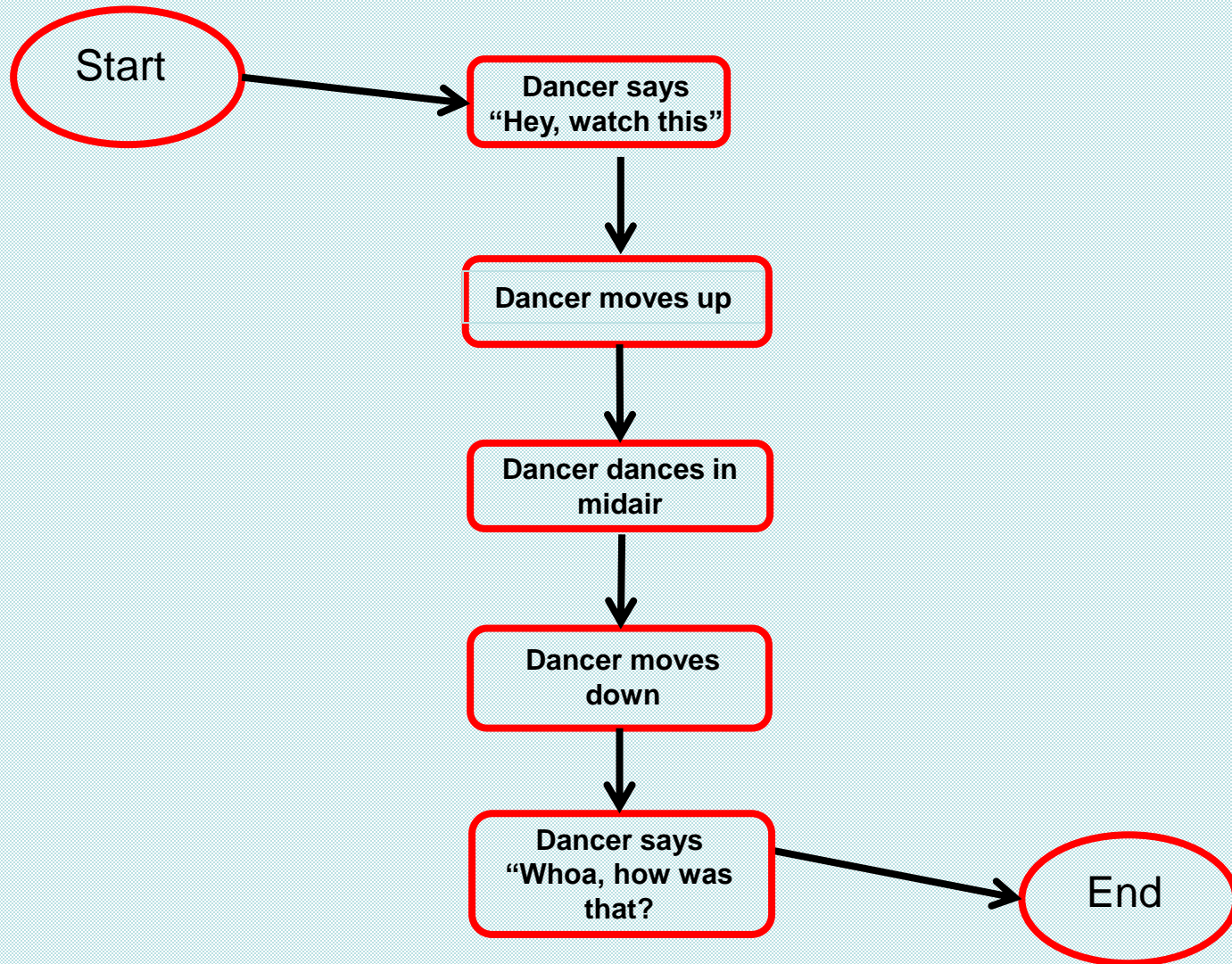
Dancer does a dance

Dancer moves down to ground

Dancer says “ Whoa,how was that ?”

This is also called pseudocode

# Flowchart



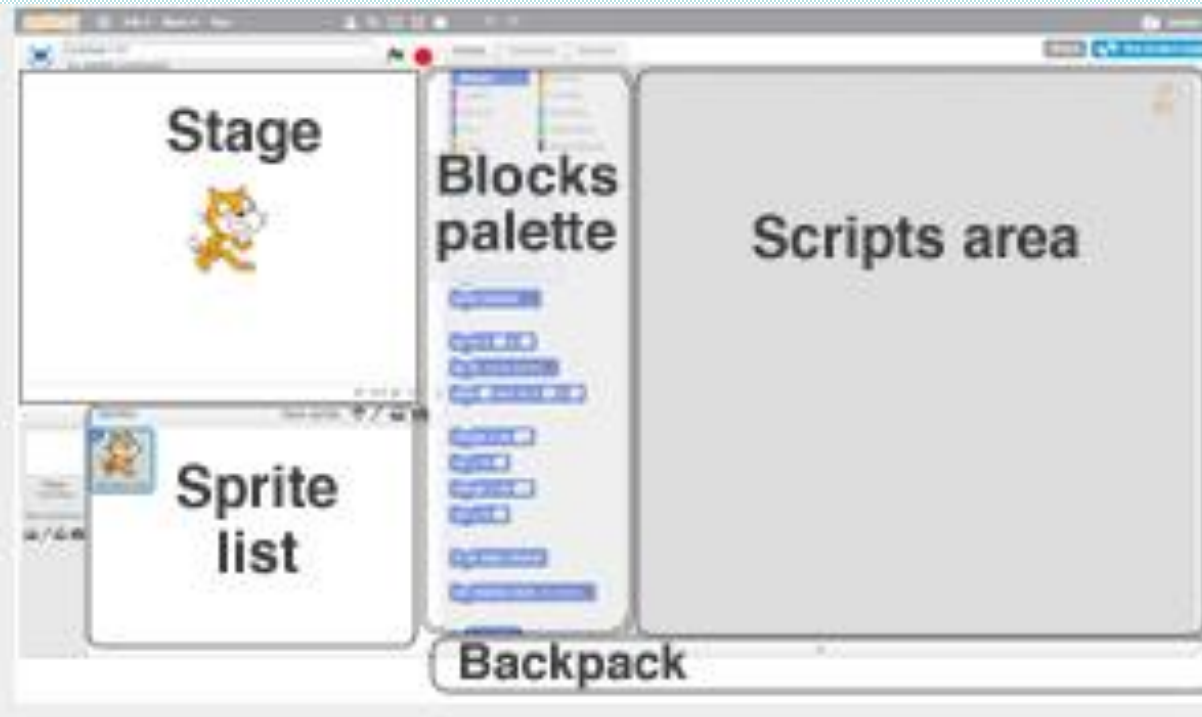
# Step 3 : Translate algorithm

- The algorithm or pseudocode or flowchart needs to be translated into a Scratch program by using the Scratch script blocks.
- The script blocks are dragged into the “Scripts area”.



# The Scratch Project Editor

**Main areas of the Project editor :**



# The Scratch Project Editor

The screenshot displays the Scratch Project Editor interface. At the top, the Scratch logo is on the left, followed by a menu bar with 'File', 'Edit', 'Tips', and 'About'. A toolbar contains icons for user, undo, redo, and help. The main workspace shows a project titled 'Project 1' by 'arbuticg (unshared)' with a character sprite. The right panel features a block palette with categories like Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators, and More Blocks. The script area contains a sequence of blocks: 'when green flag clicked', 'switch costume to champ99-b', 'say Hey, watch this. for 1.5 secs', 'go to x: -12 y: -23', 'change y by 75', a 'repeat 6' loop containing 'next costume' and 'wait 0.2 secs', 'change y by -75', 'next costume', and 'say Whoa, how was that? for 4 secs'. The bottom left shows the 'Sprites' panel with 'Champ99' as the current sprite. The bottom right shows the 'x: 2 y: 61' coordinates.

Scratch

File Edit Tips About

Project 1  
by arbuticg (unshared)

Scripts Costumes Sounds

Motion Looks Sound Pen Data Events Control Sensing Operators More Blocks

when green flag clicked

switch costume to champ99-b

say Hey, watch this. for 1.5 secs

go to x: -12 y: -23

change y by 75

repeat 6

next costume

wait 0.2 secs

change y by -75

next costume

say Whoa, how was that? for 4 secs

x: 2 y: 61

Sprites

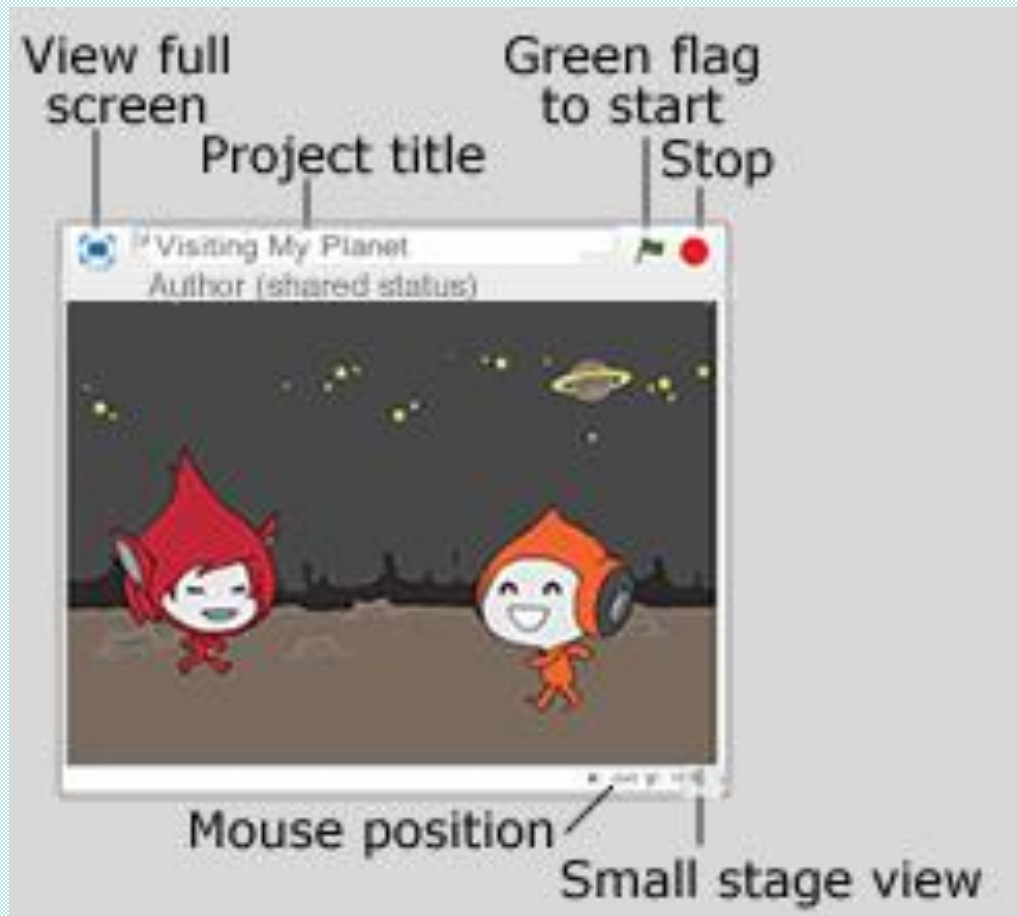
New sprite:

Stage 1 backdrop

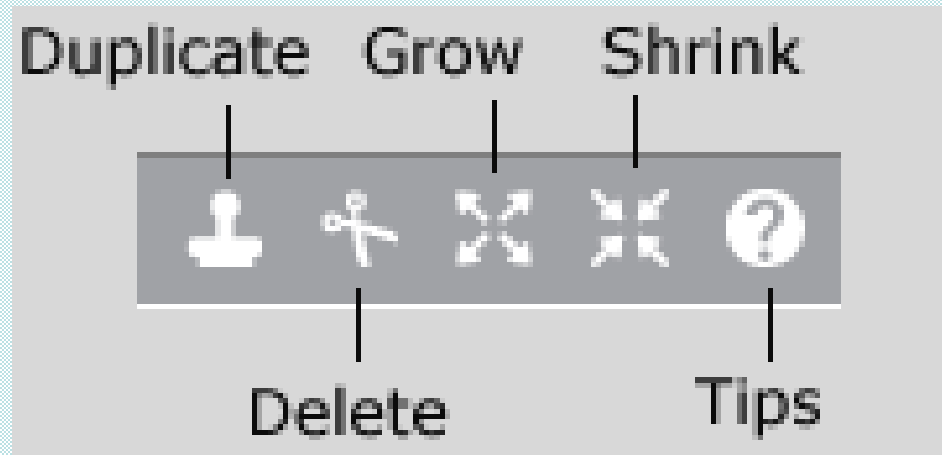
New backdrop:

Champ99

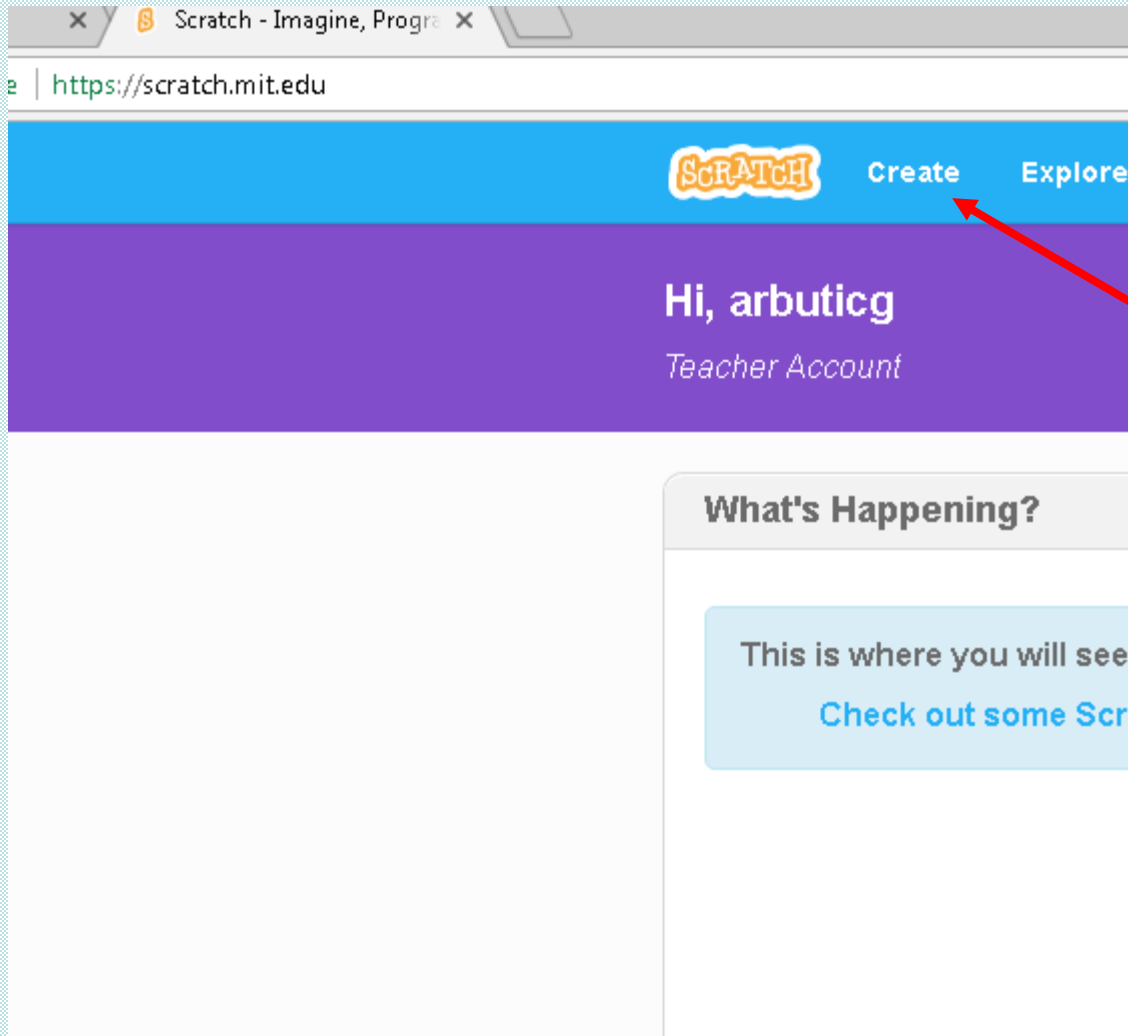
# The Stage Area



# The Curser Tools

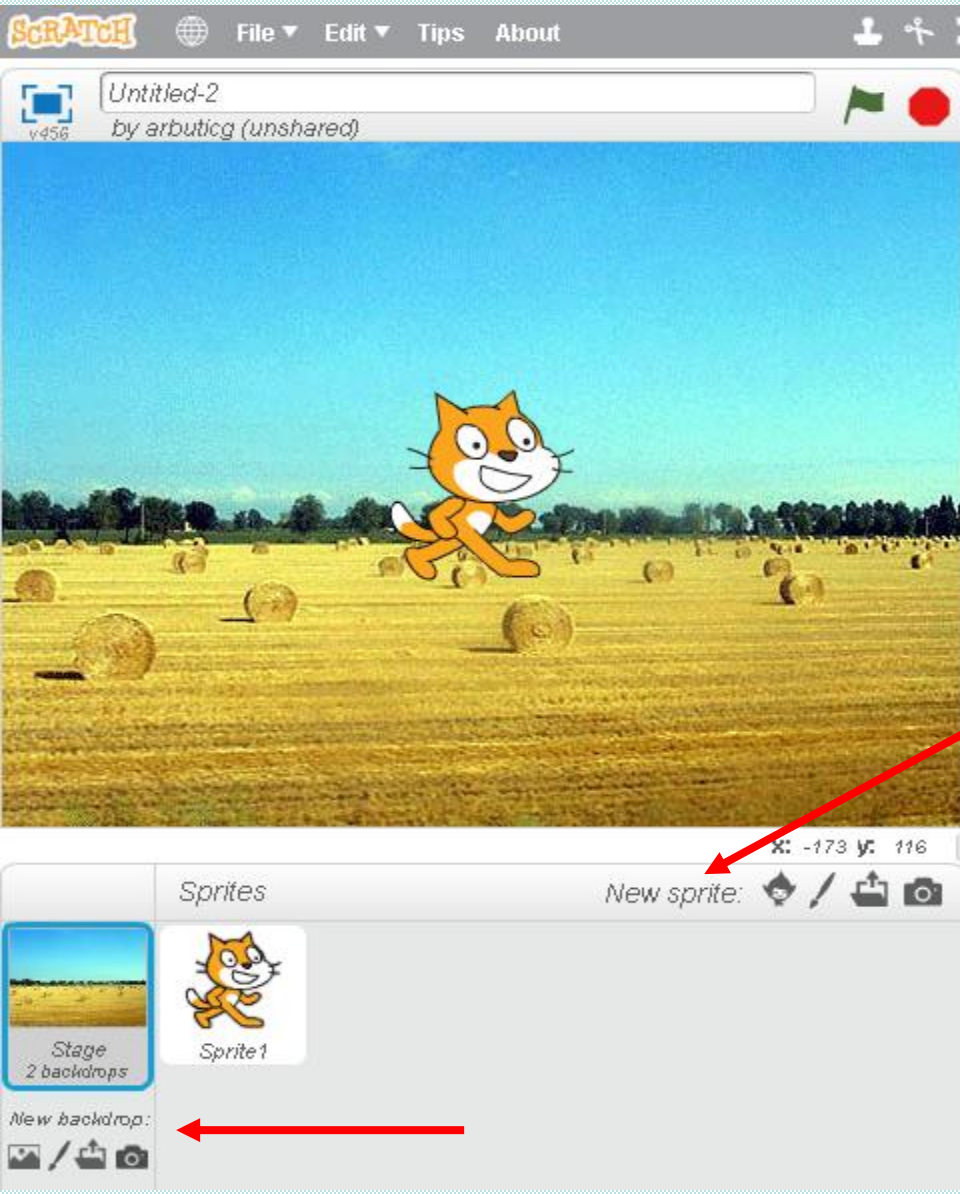


# Getting Started



Chose **Create**

# Adding sprites and changing backdrops



# Sprites

- A “**sprite**” is :
  - An object in **Scratch** which performs actions controlled by scripts
  
- How is a sprite unique?
  - has a name
  - has properties:
    - costumes, width, height, color, location, size
  - can perform scripts :
    - “scripts” blocks determine how sprites interact with each other and the backdrop

# Demo: Saving a project

➤ Writing and testing an animation is an intense load on the computing system – a crash can occur.

## ➤ **Best solution:**

 **save your project every 15 minutes**

 **(Or at least every half hour)**

 **also save to a backup system**

**(for example, a thumb drive)**



# Another story

Create an underwater scene where a fish and a crab see a scuba diver swim by. The creatures don't know what to make of the human and the fish says "Hmm...that's a funny looking fish". The crab agrees and says "For sure, goofy looking". As the scuba diver leaves, the fish is relieved and says "Good, it's gone".

# Algorithm/ pseudocode:

- Scubadiver starts swimming across the screen
- Fish says “Hmm...that’s a funny looking fish”
- Crab says “For sure, goofy looking”
- Scubadiver swims off screen
- Fish says “Good, it’s gone”

# Create initial scene

- Three sprites – Fish1, Crab and Diver2 from the Underwater Theme Library
- Use mouse to position scubadiver on left and the fish and crab along the bottom.
- Choose an appropriate backdrop from the Underwater Backdrop library

# Dialogues

Note that in the underwater example, the fish and crab were saying things to each other. This is known as a 'dialogue'

Why did the scripts contain `wait _ secs` blocks ?

# Methods to Move Sprites

Under the Motion script :

➤ Move block



➤ 'Go to' block



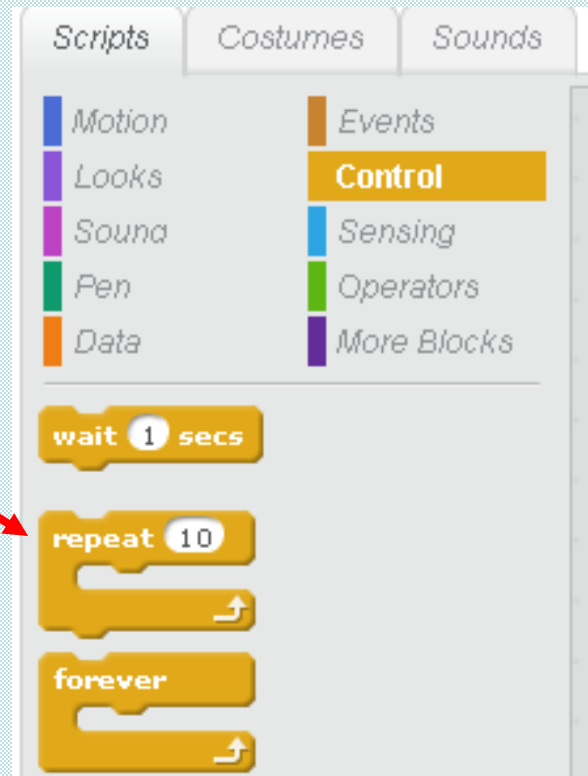
➤ Glide block



# Repetition

- In many kinds of animations, especially simulations and games, some actions happen again and again.

- A repeat block is a way to write repetitive code



# Repetition

Script from baseball.sp :

```
when clicked
  forever
    next costume
```

```
when clicked
  point in direction 90
  go to x: 0 y: -147
  glide 1 secs to x: 145 y: 1
  repeat 10
    next costume
  point in direction -90
  glide 1 secs to x: 0 y: 155
  repeat 10
    next costume
  glide 1 secs to x: -150 y: 0
  repeat 10
    next costume
  point in direction 90
  glide 1 secs to x: 0 y: -150
```

# Hands-On Assignment

- ❖ Do assignment 1



# KWL Chart


| Know | Want to learn | Learned |
|------|---------------|---------|
|      |               |         |

# Hands-On Assignment

- ❖ Do assignment 2

# Concepts seen in first projects

- Script **blocks** may have **arguments**

 Examples: for the **say** block, the arguments we used are what is said and for how long :



A Scratch 'say' block with a purple background. The text 'say' is on the left, followed by a white input field containing 'Watch us dance !', then 'for', a white input field containing '2', and finally 'secs'.

**Question** : What argument did the **move** block use ?

- Scripts for multiple sprites can begin with :



A Scratch 'when clicked' block with an orange background. The text 'when' is on the left, followed by a green flag icon, and 'clicked' is on the right.

# Testing

- 🌐 An important step in creating a program is to run it – to be sure it does what you expect it to do.
- 🌐 We recommend that you use an **incremental development** process:
  - 💡 drag a few blocks and then run it
  - 💡 drag a few more blocks and run it
  - 💡 drag a few more blocks and run it...
- 👤 This process allows you to find any problems and fix them as you go along.

# Event Based Programming

- **Event based programming** is programming in which the code is based on events such as the mouse moving or a key being pressed.
- Scripts are triggered when the event occurs

# Event Blocks



# Hands-On Assignment

- ❖ Do assignment 3

# Broadcasting

- **Broadcasts** are similar to events which trigger specific scripts.
- Sent with Broadcast () and Broadcast () And Wait, and are received by the hat block When I Receive ().
- Useful in games and animations.



# Broadcasting Example 1



x: -223 y: 180

Sprites

New sprite:



Stage  
2 backdrops



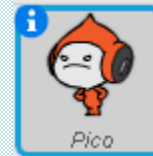
Pico



Prince

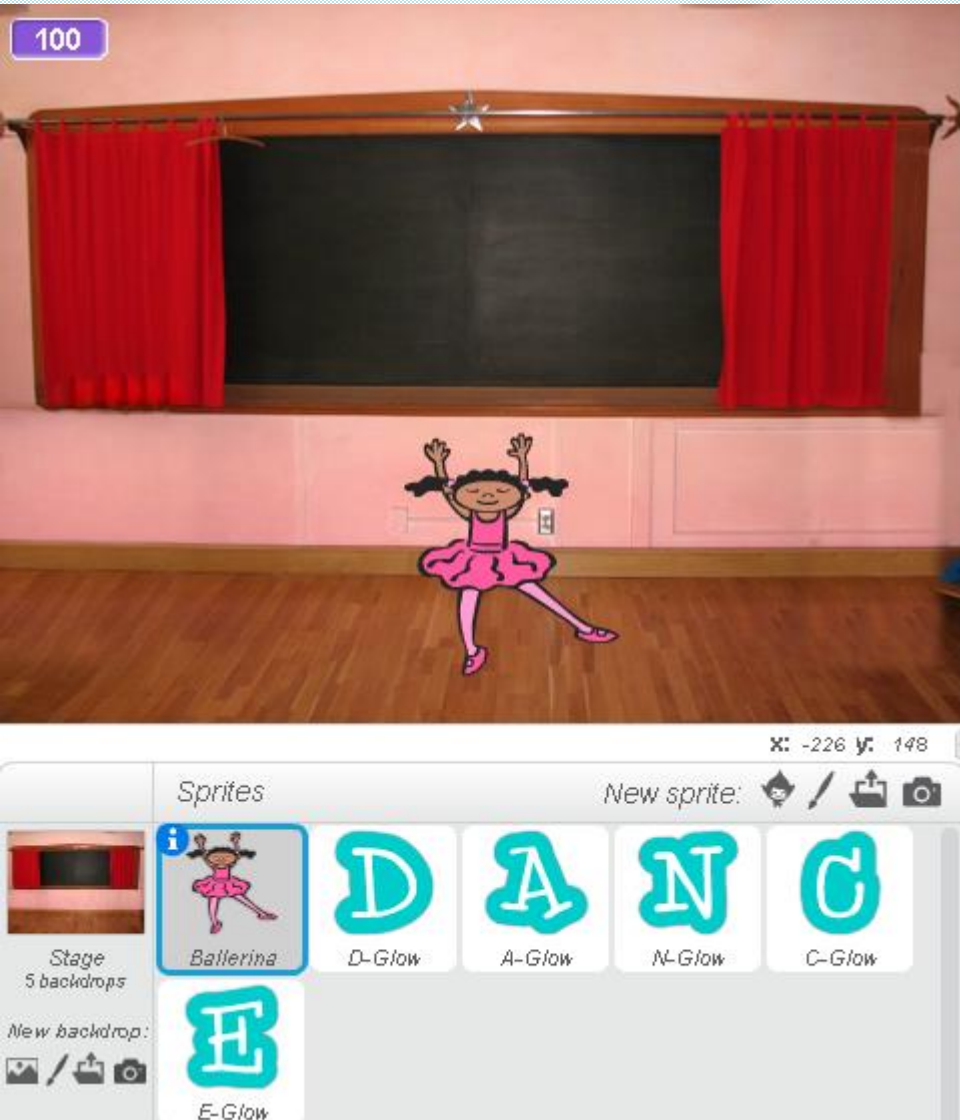
```
when green flag clicked
  switch costume to prince
  broadcast message1
  go to x: -94 y: -10
  think I am the greatest! for 2 secs
  repeat 5
    wait 0.4 secs
    move 10 steps
    switch costume to prince2
    wait 0.4 secs
    move 10 steps
    switch costume to prince
  wait 1 secs
  broadcast ha
```

```
when I receive ha
  wait 1 secs
  say Hello! Who are you? for 2 secs
  wait 0.5 secs
  broadcast intro and wait
```



Pico

# Broadcasting Example 2



```
when green flag clicked
switch backdrop to previous backdrop
go to x: -165 y: -95
repeat (30)
  move 10 steps
  next costume
  wait 1 secs
say Time for my show! for 2 secs
broadcast show
go to x: -163 y: -79
repeat (30)
  move 10 steps
  next costume
  wait 0.5 secs
say Thank You!!!! for 2 secs
```

# Broadcasting Example 2 (con't)

The Scripts area is shown with the 'Scripts' tab selected. The left sidebar lists categories: Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators, and More Blocks. Below the sidebar, it says 'Stage selected: No motion blocks'. Two scripts are visible in the workspace:

- Script 1: **when green flag clicked** followed by **switch backdrop to chalkboard3**.
- Script 2: **when I receive show** followed by **switch backdrop to stage2 and wait**.

The Backdrops area is shown with the 'Backdrops' tab selected. It displays a 'New backdrop:' section with a search bar containing 'stage2'. Below this, there are three backdrop thumbnails:

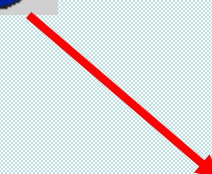
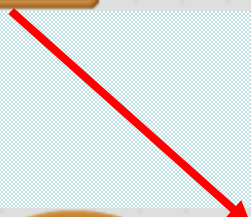
- Thumbnail 1: A white square labeled 'backdrop1 480x360'.
- Thumbnail 2: A white square labeled 'chalkboard 480x360'.
- Thumbnail 3: A green stage with a wooden floor, labeled 'stage2 480x360'. This thumbnail is highlighted with a blue border and a red arrow points from the 'stage2' block in the script above to it.

To the right of the thumbnails is a toolbar with various drawing tools like a pencil, eraser, fill, and text.

# Broadcasting Example 3

```
wait 1 secs
play note 55 for 0.5 beats
say I could broadcast "Turn orange", for 4 secs
say Ready? for 2 secs
broadcast Turn orange
```

```
when I receive Turn orange
switch costume to orange ball
broadcast Show turn orange script
```



# Hands-On Assignment

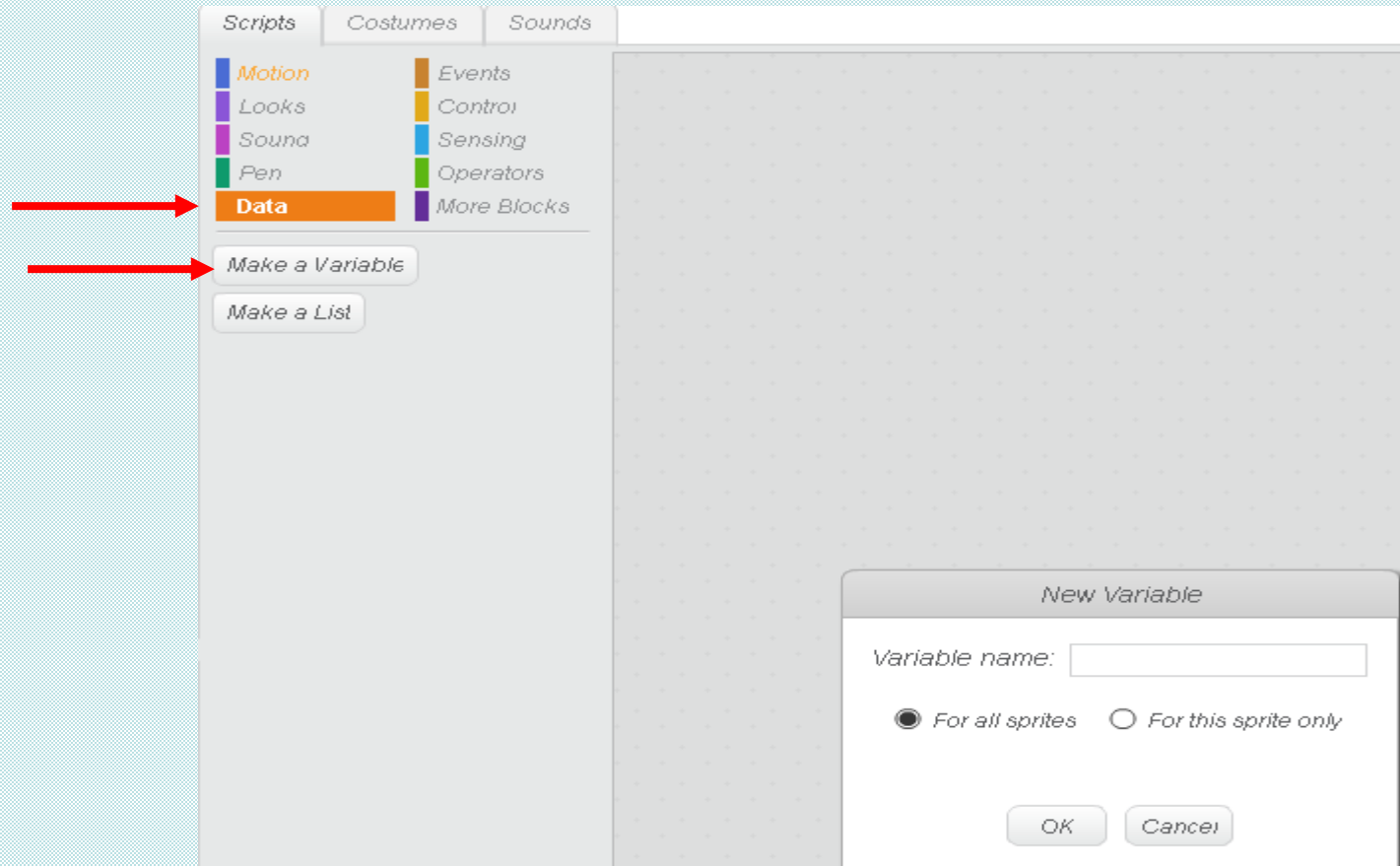
- ❖ Do assignment 4

# Variables

- A variable is a changeable value recorded in Scratch's memory.
- Variables can only hold one value at a time, unlike lists
- These values can be either numbers or strings — any text.

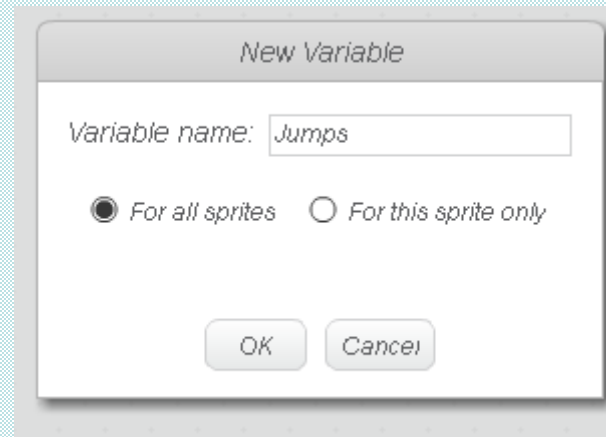
# Creating a Variable in Scratch

- Go to the Data Block and click on Make a Variable :

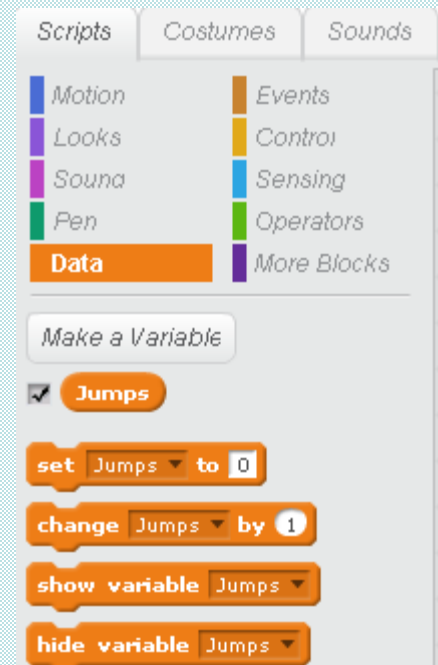


# Creating a Variable in Scratch

- Provide a name for the variable :



- Click ok and see :





# Variable Example

The screenshot displays the Scratch IDE interface. At the top, the menu bar includes 'Scratch', 'File', 'Edit', 'Tips', and 'About'. The main workspace shows a project titled 'variable example.sb' by 'arbuticg (unshared)'. A variable named 'Good Nutrition Points' is defined with a value of 0. The stage contains a green instruction 'Click on food to eat it.' and four food sprites: a bunch of bananas, a fruit platter, a donut, and a cake. The 'Scripts' panel on the right shows a 'when this sprite clicked' event block followed by a 'change Good Nutrition Points by 2' block. The 'Sprites' panel at the bottom shows the 'Fruit Platter' sprite selected.

Scratch v456  
variable example.sb  
by arbuticg (unshared)

Good Nutrition Points 0

Click on food to eat it.

when this sprite clicked  
change Good Nutrition Points by 2

Scripts  
Motion  
Looks  
Sound  
Pen  
Data  
Events  
Control  
Sensing  
Operators  
More Blocks

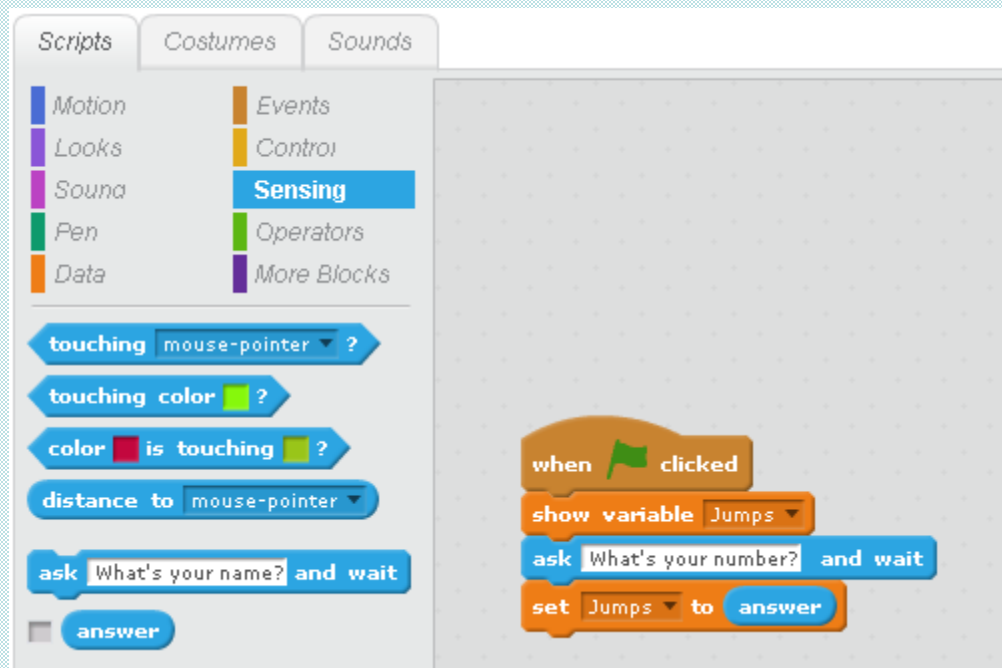
Make a Variable  
Good Nutrition Points  
set Good Nutrition Points to 0  
change Good Nutrition Points by  
show variable Good Nutrition Points  
hide variable Good Nutrition Points

Make a List

Sprites  
New sprite: [ ] [ ] [ ] [ ]  
Stage 1 backdrop  
Sprite2  
Sprite4  
Fruit Platter  
Donut  
Cake

# User input

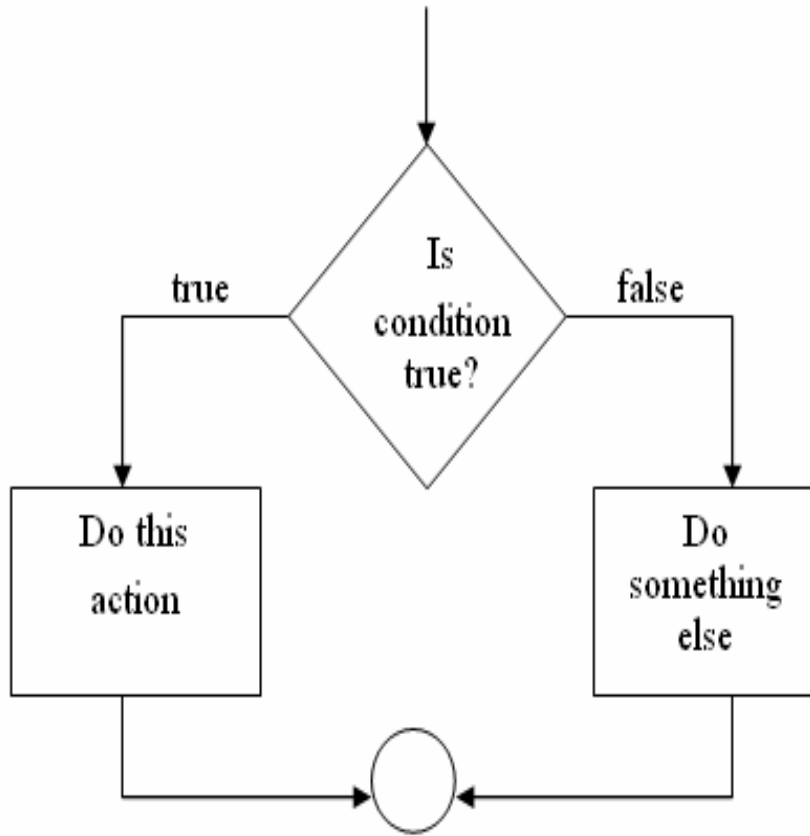
- Use of the **answer** and **ask and wait** blocks :



# Thinking About More Advanced Projects

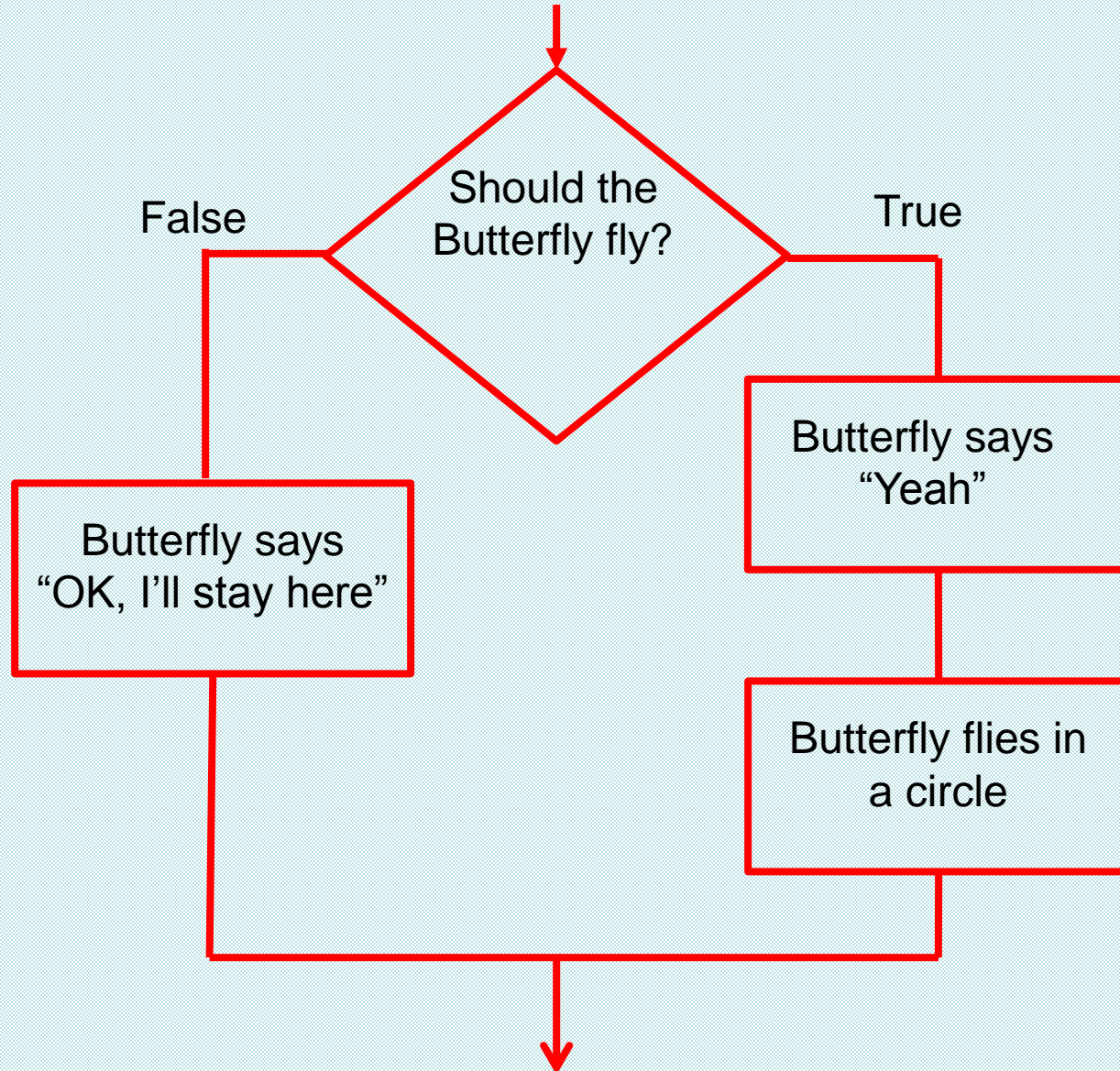
- To build more advanced projects, you will need to write code that involves **conditionals**
- We control the execution of the animation with the **if/else** structure and logical expressions

# If/Else





In Scratch, a logical expression is used as the condition in an ***If/Else*** control structure.

# If/Else



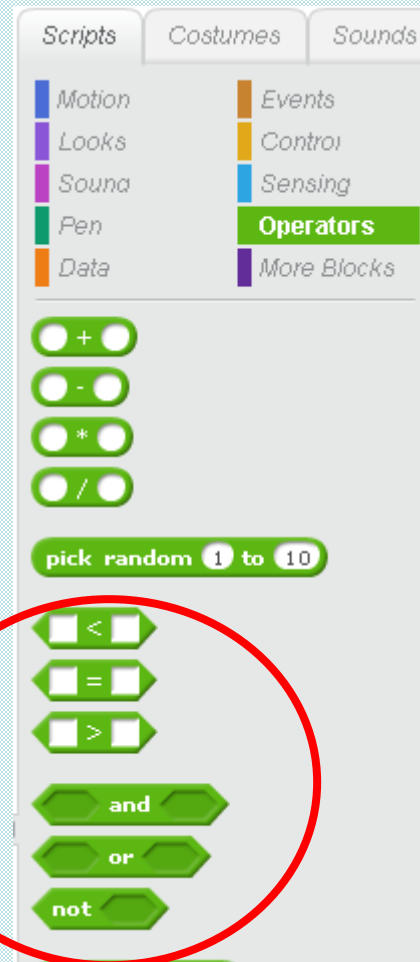
# Logical Expressions

- A **decision** is made based on current **conditions**
- A condition is checked in a **logical expression** that evaluates to *true* or *false* (Boolean) value
  - car on road  *true*
  - car over finish line  *false*

# Logical Expressions (Con't)

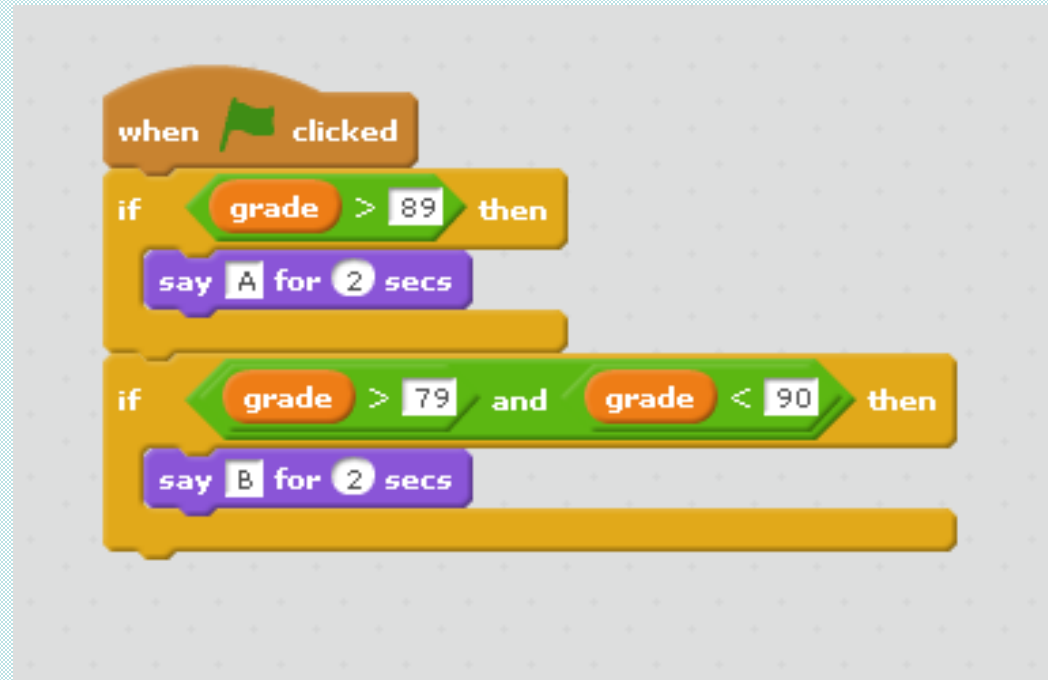
- Logical expressions can be more complex
- Is your grade greater than 85 ?
- Are you a girl AND wearing blue ?
- Are you a boy OR wearing blue ?

# Logical Expressions (Con't)





# Logical Expressions Example



# Hands-On Assignment

- ❖ Do assignment 5

# Randomness

➤ **Randomness** can be defined as the lack of pattern or predictability in events

➤ Random numbers can be generated in Scratch using :



➤ It is found under



# Randomness Example

Sets the variable die 1 to a random number between 1 and 6



```
when green flag clicked
  set die 1 to pick random 1 to 6
  if die 1 = 1 then
    switch costume to costume1
  if die 1 = 2 then
    switch costume to costume2
  if die 1 = 3 then
    switch costume to costume3
  if die 1 = 4 then
    switch costume to costume4
  if die 1 = 5 then
    switch costume to costume5
  if die 1 = 6 then
    switch costume to costume6
```

The image shows a Scratch script starting with a 'when green flag clicked' block. The second block is 'set die 1 to pick random 1 to 6'. A red arrow points from the text 'Sets the variable die 1 to a random number between 1 and 6' to this 'set' block. Below the 'set' block are six 'if' blocks, each with a condition 'die 1 = [number]' and a 'switch costume to [costumeX]' block. The numbers 1 through 6 are in the 'if' conditions, and the costume names are costume1 through costume6.

# Hands-On Assignment

- 🌐 Do assignment 6

# The Timer (block)

- The **timer** block is found under **Sensing**
- The block starts at 0 when Scratch is launched and increases gradually; every second it will have increased by 1
- Usually used with the **reset timer** block at beginning of project so it holds the correct time

# Timer Example

The screenshot displays the Scratch code editor interface. The stage area shows two sprites: a cat and a yellow spiky character. The Scripts panel is active, showing the following code blocks:

- when green flag clicked
- reset timer
- wait until touching Gobo ?
- say join Seconds to Gobo: timer for 4 secs
- when green flag clicked
- forever loop containing:
  - go to x: mouse x y: mouse y

The Sprites panel at the bottom shows the cat and the yellow spiky character. The coordinates are X: -78 Y: 172.

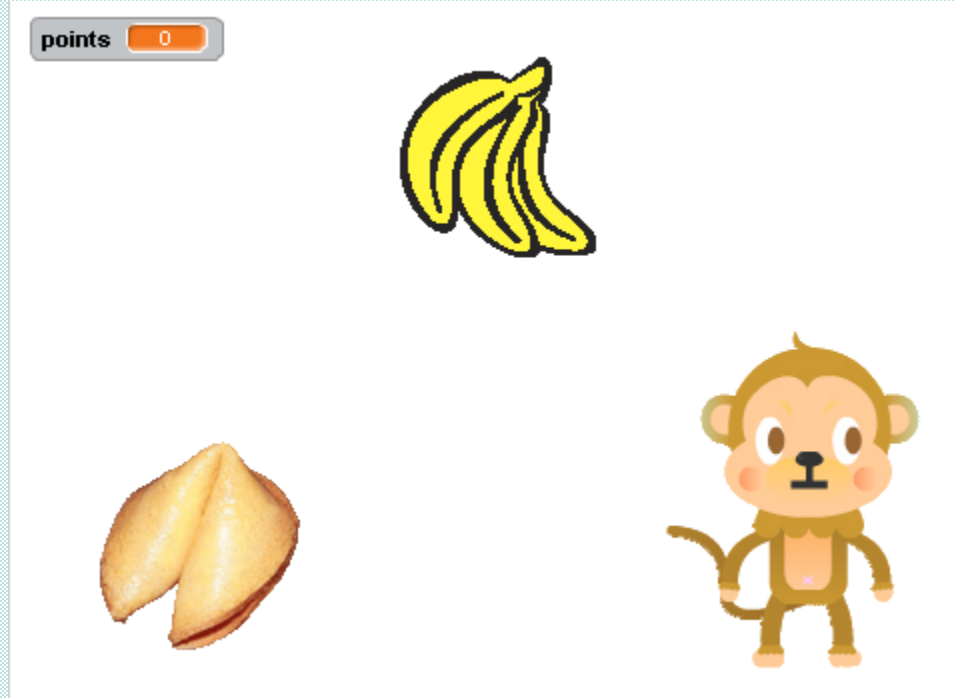
# Hands-On Assignment

- 🌐 Do assignment 7 (Timer Project)



# Monkey Game

- The objective is to move the monkey using the arrow keys until it touches a piece of food



# Monkey Game (Con't)

Part of Monkey script :



```
when green flag clicked
  set points to 0
  show
  when up arrow key pressed
    point in direction 0
    move 10 steps
    broadcast moved
  when down arrow key pressed
    point in direction 180
    move 10 steps
    broadcast moved
```

The script for the Monkey character starts with a 'when green flag clicked' event. It sets a 'points' variable to 0 and shows the character. It then has two key press events: one for the 'up arrow' key that points the character in direction 0 and moves it 10 steps, and another for the 'down arrow' key that points the character in direction 180 and moves it 10 steps. Both key press events broadcast a 'moved' message.

Bananas script :

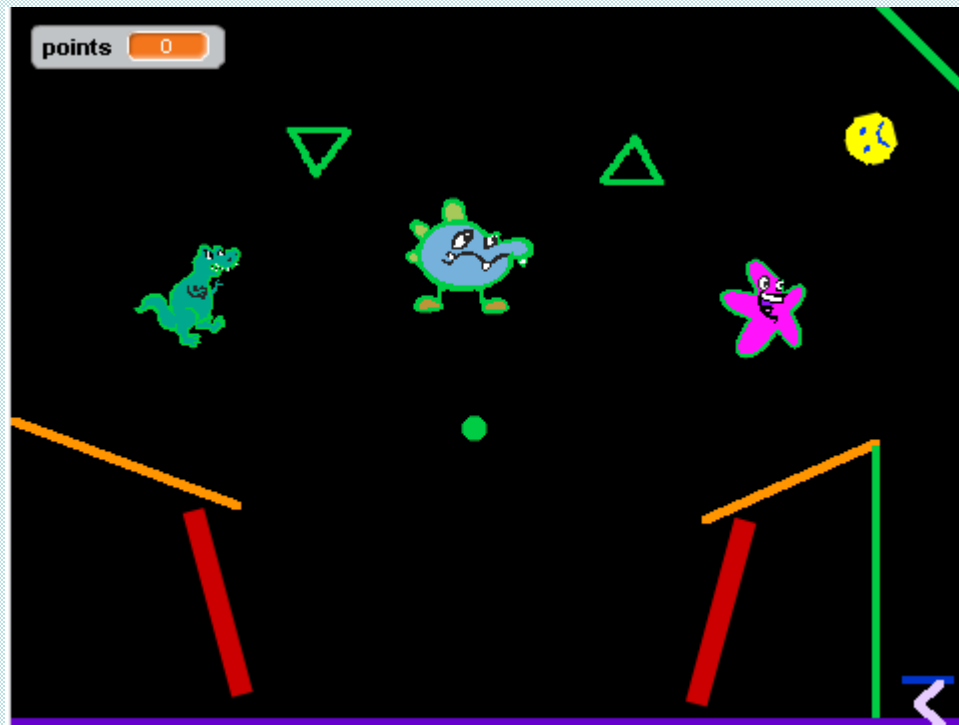


```
when I receive moved
  if touching Sprite1 ? then
    set x to pick random -240 to 240
    set y to pick random -180 to 180
    change points by 1
    if points > 9 then
      broadcast win
```

The script for the Bananas character starts with a 'when I receive moved' event. It checks if the character is touching 'Sprite1'. If so, it sets the x and y coordinates to random values within the specified ranges (-240 to 240 for x, -180 to 180 for y), increases the 'points' variable by 1, and checks if the points are greater than 9. If they are, it broadcasts a 'win' message.

# Pinball Game

- Points are achieved when the yellow ball bounces off one of the sprites in the upper part of the screen.



# Team Project

- 🌍 Working with your partner, develop a scratch project. In addition, prepare a lesson plan which can be used to teach the concepts illustrated in the project.
- 🌍 This will be presented to the CS4HS attendees on Friday