# Handbook for Scratch

For Primary School Students



Prashanth Purushotham
ITPROACADEMY.CO.IN

# About Me

**Prashanth Purushotham**

**Head of IT Infrastructure & Automation**
**Support@ITProAcademy.co.in** | **www.ITProAcademy.co.in**

With over **24 years of extensive experience in the IT industry**, I have dedicated my career to mastering and implementing innovative solutions across diverse domains, including **infrastructure, virtualization, converged systems, cloud technologies, cybersecurity**, and **automation**. As a **Microsoft Certified Trainer (MCT)** for the past **20 years**, I have been empowering professionals and students to harness cutting-edge technologies effectively.

Currently serving as the **Head of IT Infrastructure & Automation** for a UK-based IT company, I specialize in performance monitoring for both infrastructure and applications, business continuity planning (BCP), disaster recovery (DR), and solution design. My expertise spans a wide range of platforms, such as **Microsoft Windows, Linux, VMware, Hyper-V**, and **networking**. I excel in **migrating workloads to public, private, and hybrid cloud environments** while leveraging the latest advancements in **business process automation**.

I have obtained numerous certifications across multiple domains, including **Azure Solutions Architect Expert**, **Azure AI Engineer Associate**, and **Cybersecurity Architect Expert**, among others. These accolades reflect my commitment to continuous learning and staying at the forefront of technology trends. I have also been recognized for my contributions to the **Dell EMC Proven Professional Program**, **Oracle Cloud Foundations**, and **ISC² certifications**.

Through ITProAcademy, my vision is to bridge the gap between academic learning and industry demands by equipping engineering students with the knowledge and skills needed to excel in the dynamic world of IT. This guide is part of that mission, offering a comprehensive learning resource for **engineering students** as part of the **Connect Program** conducted with colleges. It is my endeavour to inspire the next generation of IT professionals and empower them to thrive in an ever-evolving technological landscape.

# Contents

# Scratch Learning Material for Primary School Students

Scratch learning material for primary school students can make coding fun and engaging while aligning with their level of comprehension and creativity. Here's a step-by-step outline of learning material and activities.

## 1. Introduction to Scratch

### Objective

Understand what Scratch is, how it works, and why it's a fun and creative way to learn coding.

### What is Coding?

Coding is like giving instructions to a computer to make it do what you want. Just like you follow steps to solve a math problem, computers follow steps written in code to complete tasks.

But coding doesn't have to be complicated! Scratch is a simple, colourful platform where you can build games, animations, and stories by putting together puzzle-like blocks instead of writing lines of text.

### Why is Scratch Fun?

- **Interactive Learning**: Scratch lets you create projects that move, talk, and even play music.
- **Build What You Love**: Make your own characters, design games, and tell stories.
- **Share and Explore**: Share your creations with friends and explore other cool projects on Scratch.

### Overview of the Scratch Interface

1. **The Stage**
    - The area where your project comes to life.
    - You'll see your characters (called sprites) move, talk, and interact here.
2. **Sprites**
    - These are the actors in your projects.
    - Sprites can be anything—a person, animal, car, or even a ball.
3. **Blocks Palette**

- This is where you find all the coding blocks.
- Blocks are organized into categories like Motion, Looks, Sound, Events, and more.
- Each block has a specific function, like moving your sprite or playing a sound.

4. **Script Area**
   - The space where you connect blocks to build your program.
   - Drag blocks from the palette to this area to create your code.

## Importance of Logic and Creativity in Coding

1. **Logic**
   - Coding teaches you to think step by step.
   - Example: If you want a sprite to jump, you'll need to write instructions like "Move up" and "Move down."
2. **Creativity**
   - Scratch encourages you to dream big!
   - You can design your own characters, create unique stories, and build games that reflect your imagination.

---

### Fun Fact

Scratch was created by MIT (Massachusetts Institute of Technology) to help kids learn coding in a fun and easy way. Millions of students around the world use Scratch to explore their ideas.

---

## Quick Activity

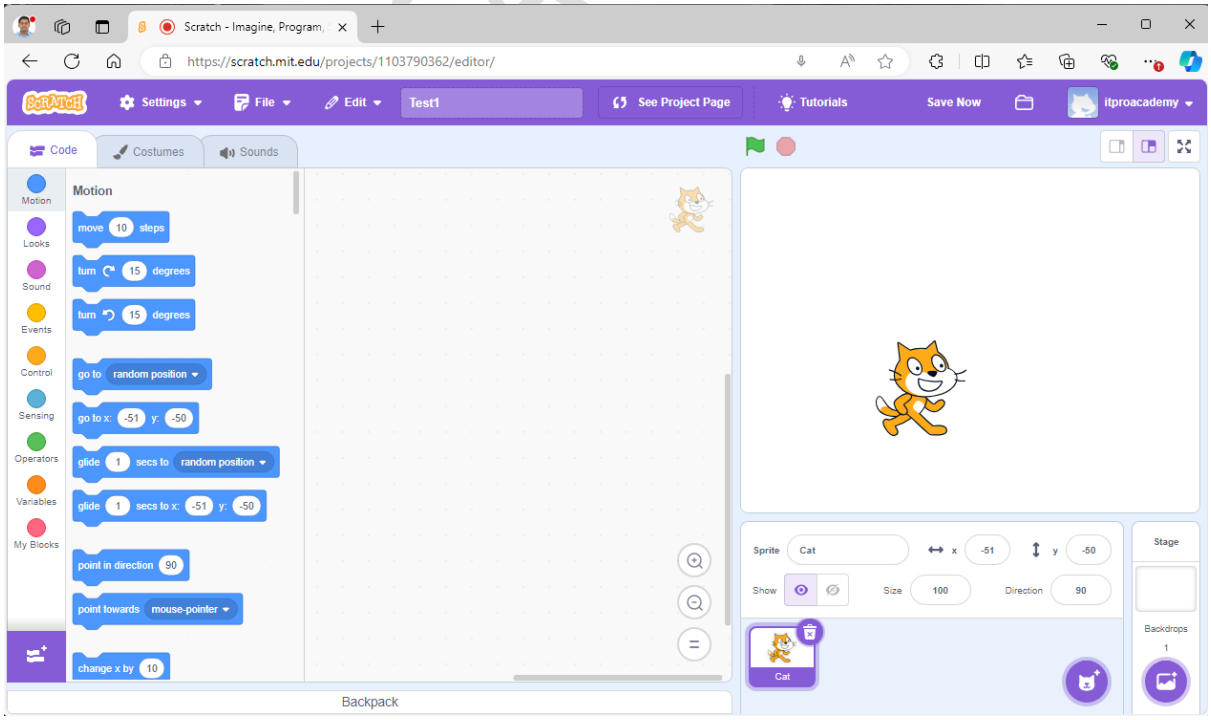
- **Task**: Open Scratch, drag the block **"When green flag clicked"**, and add the block **"Say Hello!"**.
- **What Happens?**: Your sprite will greet you on the stage!.

# 2. Getting Started with Scratch

## How to Create an Account and Save Projects

- **Why Create an Account?**
    - Save your projects and access them anytime from any device.
    - Share your creations with friends and explore projects by others.
- **Steps to Create an Account**:
    1. Open scratch.mit.edu.
    2. Click on **"Join Scratch"** at the top-right corner.
    3. Fill in your details: choose a username, create a password, and enter your birth month/year.
    4. Confirm your email address with the help of your parents or teacher.
- **Saving a Project**:
    1. Click on **“File”** in the top-left corner.
    2. Select **“Save now”** to save your project online.
    3. You can also download your project by selecting **“Save to your computer”**.

## Exploring the Interface by Dragging and Dropping Blocks




- Open a new project in Scratch to explore.

- Click on “File -> New” (top left side of the screen).

**Try This**:

- Drag the **“Move 10 steps”** block from the Motion category and drop it into the script area.
- Click the block to see your sprite move on the stage.

**Explore More Blocks**:

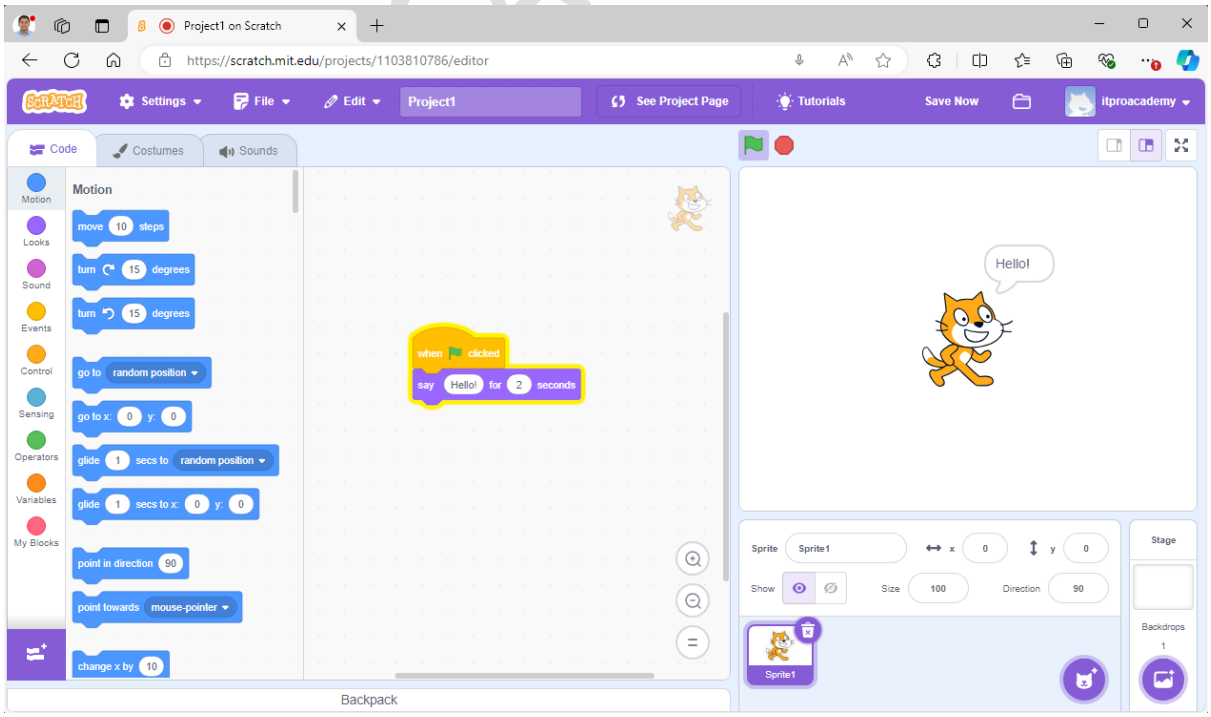
**Looks**: Make the sprite talk or change its costume.

**Sound**: Play a sound or a song.

**Events**: Start actions by clicking a flag or a sprite.

## Create Your First Program: "Hello, World!"

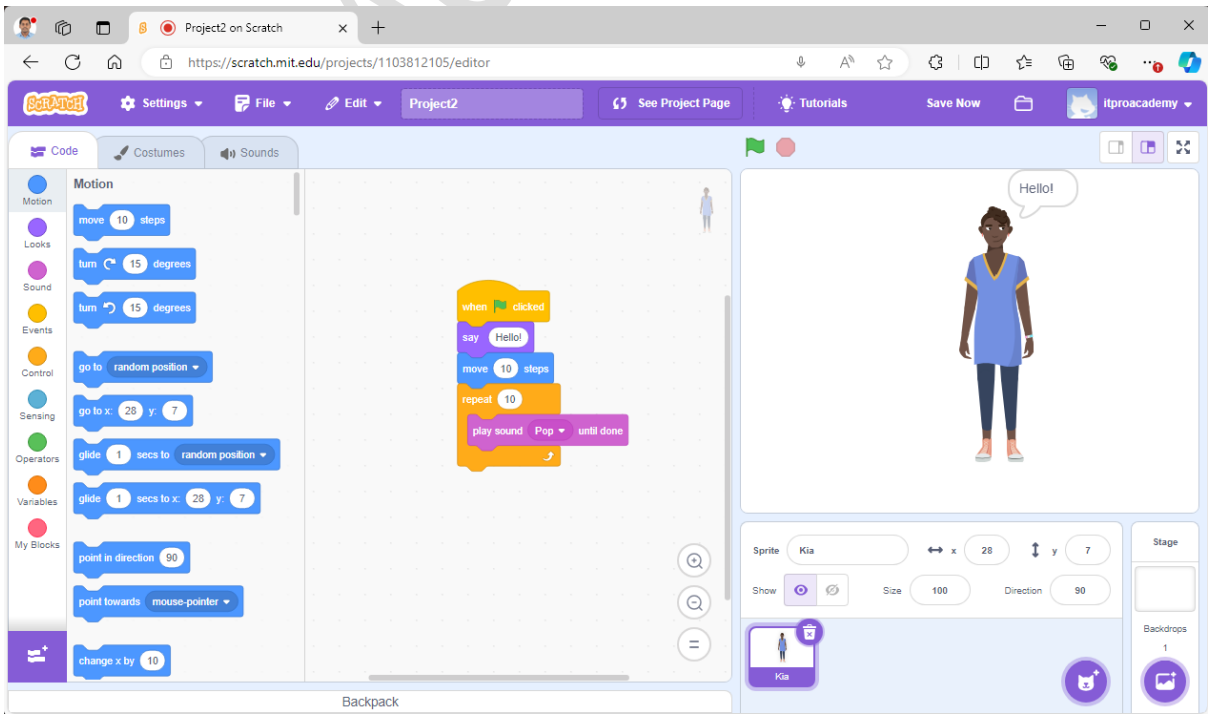
- **Steps**:
  - Click on “File -> New” (top left side of the screen).
  - Drag the **“When green flag clicked”** block from the Events category.
  - Drag the **“Say Hello for 2 seconds”** block from the Looks category.
  - Snap the blocks together in the script area.
  - Click the green flag above the stage.
- **What Happens?**
  - Your sprite will display the message "Hello!" on the stage.



## Activity

Animate a Character Introducing Themselves with a Speech Bubble

1. **Steps**:
    - Add a sprite:
        1. Click the **"Choose a Sprite"** button and select a character (below right side of the screen).
    - Create an introduction:
        1. Drag **“When green flag clicked”** block from Events category.
        2. Attach a **“Say Hello!”** block from Looks category. Change the text to "Hi, I am [Sprite Name]!".
    - Add movement:
        1. Drag **“Move 10 steps”** block from Motion category.
        2. Use a **“Repeat 10 times”** block from the Control category to make the sprite move repeatedly.
2. **Optional**:
    - Add sound by dragging the **“Play sound”** block from the Sound category.
    - Change the backdrop to make it more creative (below right side of the screen).



## Challenge:

Customize the character’s message and movements. Make the sprite wave or bounce as it speaks!

# 3. Basic Concepts in Scratch

## 1. Events

Events are how you tell Scratch when to start or respond to something.

- **Examples**:
  - **"When green flag clicked"**: Runs your code when the green flag above the stage is clicked.
  - **"When sprite clicked"**: Runs code when you click on a sprite on the stage.
- **Try This:**
  1. Drag **"When sprite clicked"** from the Events category.
  2. Attach **"Say Hello!"** from the Looks category.
  3. Click the sprite to see what happens.

## 2. Motion

Motion blocks make your sprites move around the stage.

- **Examples**:
  - **"Move 10 steps"**: Moves the sprite forward by 10 steps.
  - **"Turn 15 degrees"**: Rotates the sprite clockwise by 15 degrees.
- **Try This**:
  1. Drag **"When green flag clicked"** from Events.
  2. Attach **"Move 10 steps"** and **"Turn 15 degrees"** from Motion.
  3. Click the green flag to see the sprite move and turn.

## 3. Looks

Looks blocks change how your sprite appears.

- **Examples:**
  - **"Say Hello!":** Displays a speech bubble with text.
  - **"Switch costume to [costume name]":** Changes the sprite's outfit.
- **Try This:**
  1. Select a sprite with multiple costumes (e.g., the cat sprite).
  2. Drag "Switch costume to next costume" from Looks.
  3. Attach it to "When sprite clicked" and click the sprite to see the costume change.

## 4. Sounds

Sound blocks add audio to your project, like effects or music.

- **Examples:**

- **“Play sound [meow]”:** Plays a sound.
- **“Start sound [meow]”:** Starts playing a sound but doesn’t wait for it to finish.

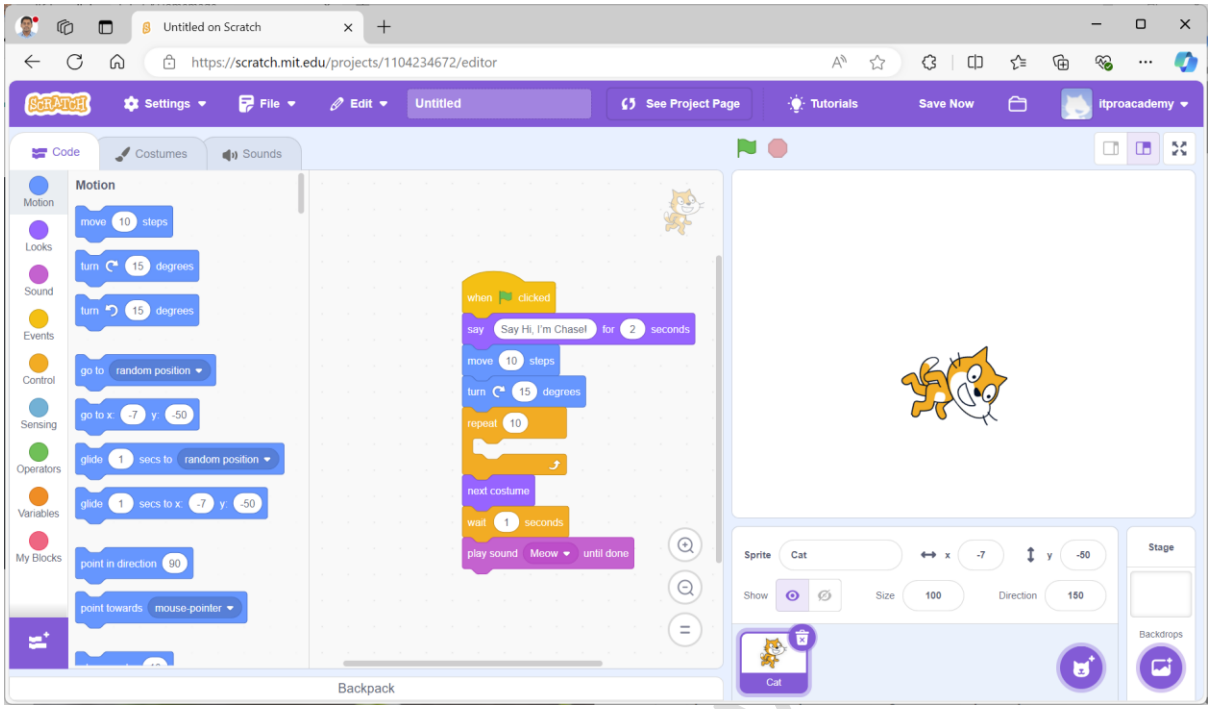
- **Try This:**
  1. Drag “When green flag clicked” from Events.
  2. Attach “Play sound meow” from the Sound category.
  3. Click the green flag to hear the sound.

## Activity

Create an Interactive Project Where a Sprite Introduces Itself, Moves, and Changes Costumes

### Steps

1. **Set Up the Sprite**:
   - Choose a sprite with multiple costumes (e.g., the Scratch cat).
2. **Add the Introduction**:
   - Drag **“When green flag clicked”** from Events.
   - Attach **“Say Hi, I’m [Sprite Name]! for 2 seconds”** from Looks.
3. **Make the Sprite Move**:
   - Attach **“Move 10 steps”** and **“Turn 15 degrees”** from Motion.
   - Use a **“Repeat 10”** block to make the movement happen multiple times.
4. **Change Costumes**:
   - Attach **“Next costume”** from Looks.
   - Use a **“Wait 1 second”** block to pause between costume changes.
5. **Add Sound**:
   - Attach **“Play sound meow”** from Sound.
6. **Final Code**:
   - The sprite introduces itself, moves in a pattern, changes costumes, and plays a sound.




## Challenge

- Add a backdrop to make the project more colourful.
- Use the **“When sprite clicked”** event to make the sprite do something new, like jumping or spinning.

# 4. Simple Animations

## Objective

Learn how to create animations in Scratch by combining motion blocks and costume changes. Animations bring your projects to life, making them fun and dynamic!

## Activity 1

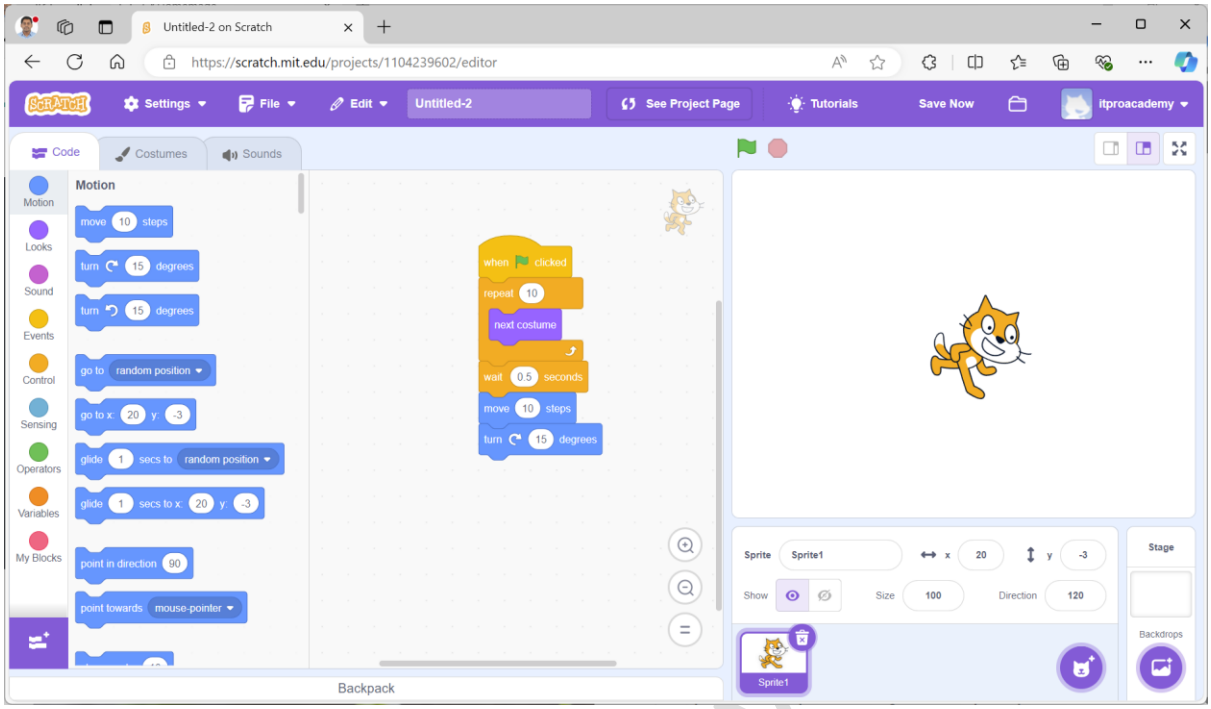
Make a Sprite Dance Using Costume Changes and Motion Blocks

### Steps to Create a Dancing Sprite

1. **Choose Your Sprite**:
    - Pick a sprite with multiple costumes, such as the Scratch cat.
    - Each costume represents a different dance pose.
2. **Set Up Costume Changes**:
    - Drag **"When green flag clicked"** from the Events category.
    - Attach a **"Repeat 10"** block from the Control category.
    - Inside the Repeat block, add **"Next costume"** from the Looks category.
    - Add a **"Wait 0.5 seconds"** block to pause between costume changes.
3. **Add Movement**:
    - Add **"Move 10 steps"** and **"Turn 15 degrees"** blocks from Motion after each costume change to simulate dancing.
4. **Final Code**:
    - The sprite will move, turn, and change costumes repeatedly, creating a dance animation.

### Enhancements:

- Play music using **"Play sound [music loop]"** from Sound.
- Add a backdrop to create a dance floor.

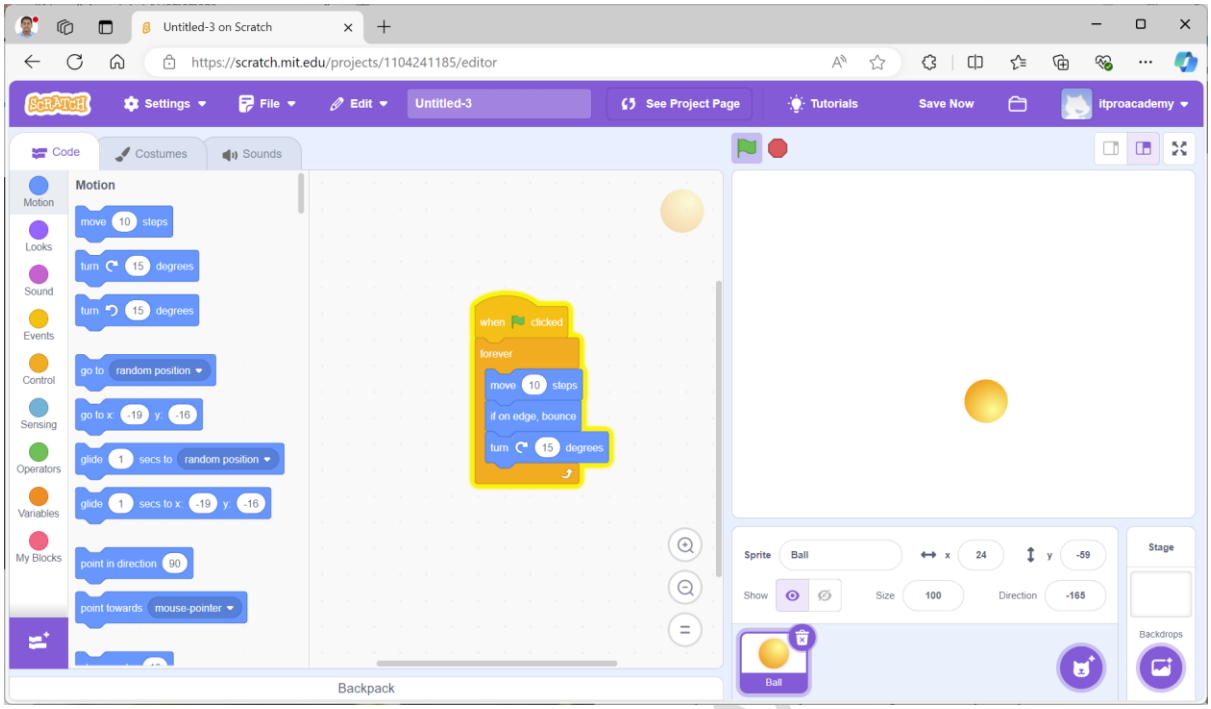



## Activity 2

Animate a Ball Bouncing Across the Screen

### Steps to Animate a Bouncing Ball

1. **Choose Your Sprite**:
    - Select a ball sprite from the library.
2. **Set Up the Bounce Animation**:
    - Drag **"When green flag clicked"** from Events.
    - Attach a **"Forever"** block from Control.
    - Inside the Forever block, add:
        - **"Move 10 steps"** from Motion.
        - **"If on edge, bounce"** from Motion to make the ball bounce off the edges of the stage.
3. **Enhance the Animation**:
    - To make the ball spin as it bounces, add **"Turn 15 degrees"** inside the Forever block.
4. **Adjust the Speed**:
    - Experiment with the number of steps in the **"Move"** block to control the ball's speed.




## Challenge

1. Can you make the ball change colors as it bounces?
    - Hint: Use the **“Change color effect by 25”** block from the Looks category.
2. Add a second ball sprite and animate it differently.

# 5. Storytelling with Scratch

## Objective

Learn how to combine different Scratch blocks to create and tell an engaging story. Use backdrops, sprites, and animations to bring your imagination to life.

### 1. Using Backdrops to Set Scenes

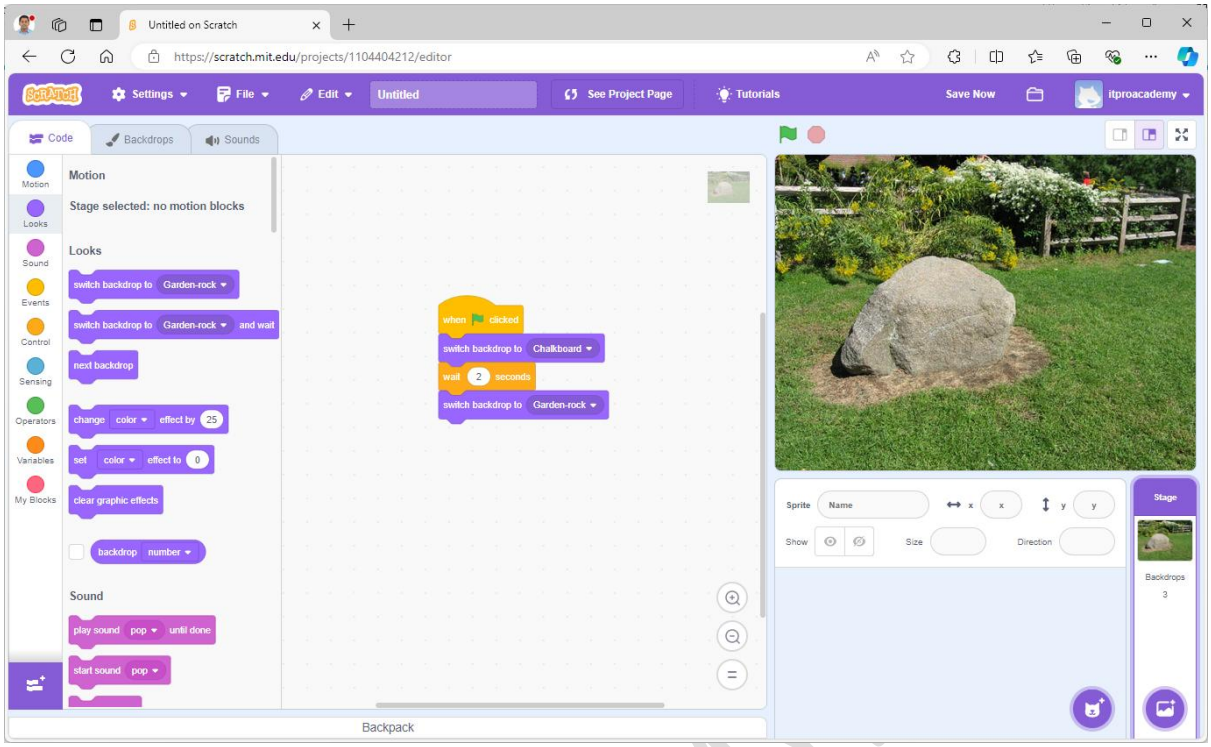
Backdrops are like the setting of a stage in a play. They help create the mood and context for your story.

- **Steps to Add a Backdrop**:
    1. Click the **"Choose a Backdrop"** button below the stage.
    2. Select a backdrop from the Scratch library, such as a forest, classroom, or beach.
- **Tip**: Use the **“Switch backdrop to [name]”** block from the Looks category to change backdrops during the story.

### 2. Switching Between Backdrops

You can change backdrops to show different scenes in your story.

- **Example**:
    - Scene 1: A classroom where the characters introduce themselves.
    - Scene 2: A park where they play together.
- **Steps to Switch Backdrops**:
    1. Add multiple backdrops to your project.
    2. Drag the **“When green flag clicked”** block from Events.
    3. Attach **“Switch backdrop to [first backdrop]”** from Looks.
    4. Use **“Wait [seconds]”** from Control to show the scene for a specific time.
    5. Attach **“Switch backdrop to [second backdrop]”** for the next scene.




## Activity

Create a Two-Scene Story Where Characters Interact

### Story Outline:

1. **Scene 1 (Introduction)**:
    - The first character greets the audience in a classroom chalkboard.
    - The second character enters and invites them to the park.
2. **Scene 2 (At the Park)**:
    - Both characters play together, then say goodbye to the audience.

### Steps to Create the Story:

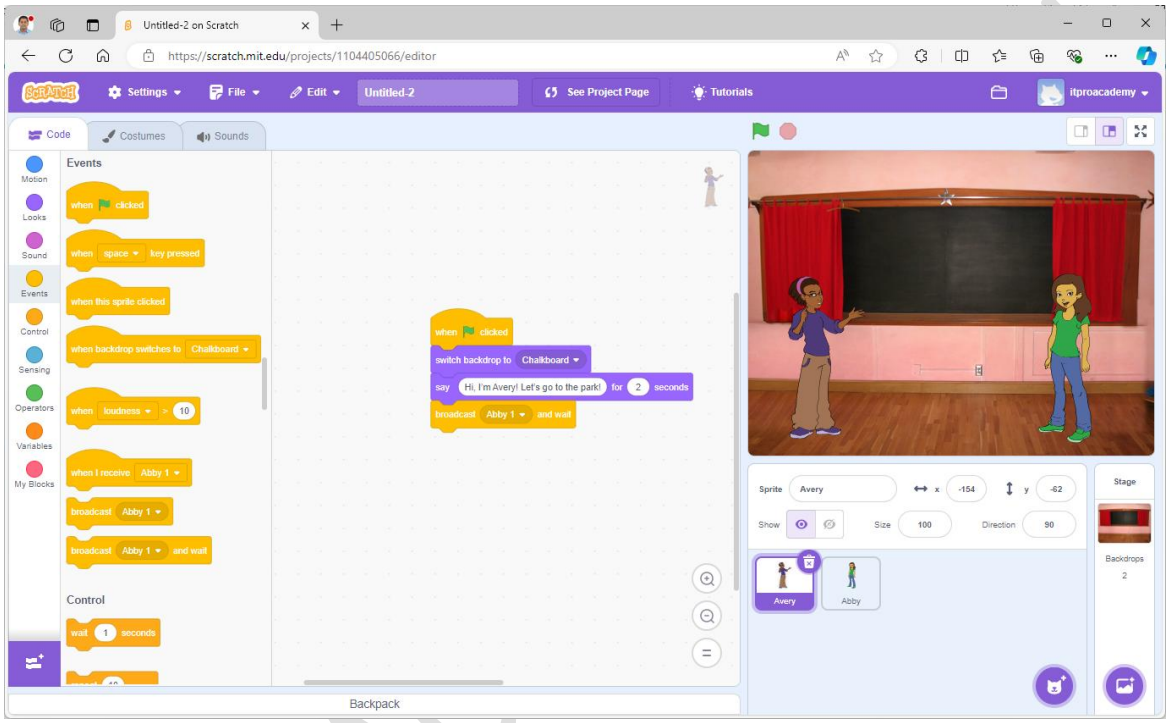
1. **Set Up Scene 1**:
    - Add a Chalkboard backdrop, by clicking on backdrop at the right-side bottom of the screen.
    - Choose two sprites (Avery & Abby) for the characters, by clicking on sprite at the right-side bottom of the screen.
    - Click on sprite Avery on right side of the screen.
    - Drag **“When green flag clicked”** from Events.
    - Drag **“switch backdrop to chalkboard”** from Looks.

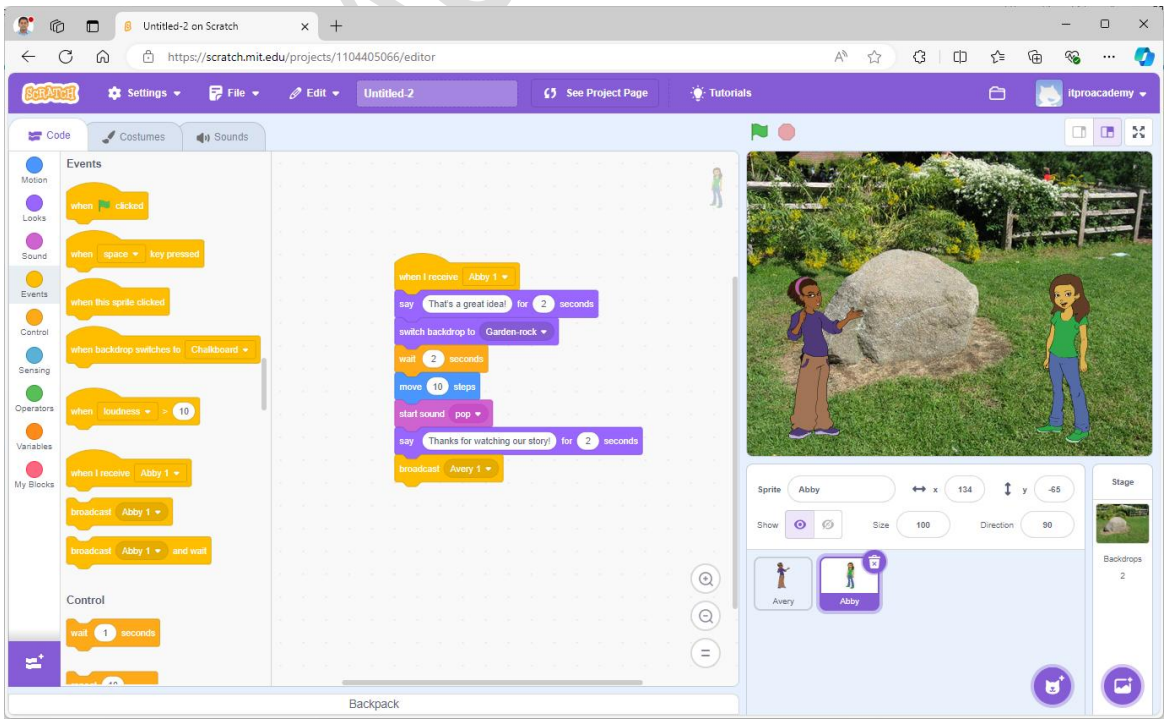
- Use speech bubbles:
  - First sprite (Avery): **“Hi, I’m Avery! Let’s go to the park!”.**
  - Drag “Say message for 2 seconds” (type the message after say **“Hi, I’m Avery! Let’s go to the park!”.**
  - Drag **“Broadcast Abby 1and wait”** (click on the down arrow key after broadcast, select New message and type “Abby 1”) from Events.
  - Click on sprite Abby on right side of the screen.
  - Drag **“When I receive Abby 1”** from Events.
  - Second sprite (Abby): **“That’s a great idea!”.**
  - Drag “Say message for 2 seconds” (type the message after say **“That’s a great idea!”.**

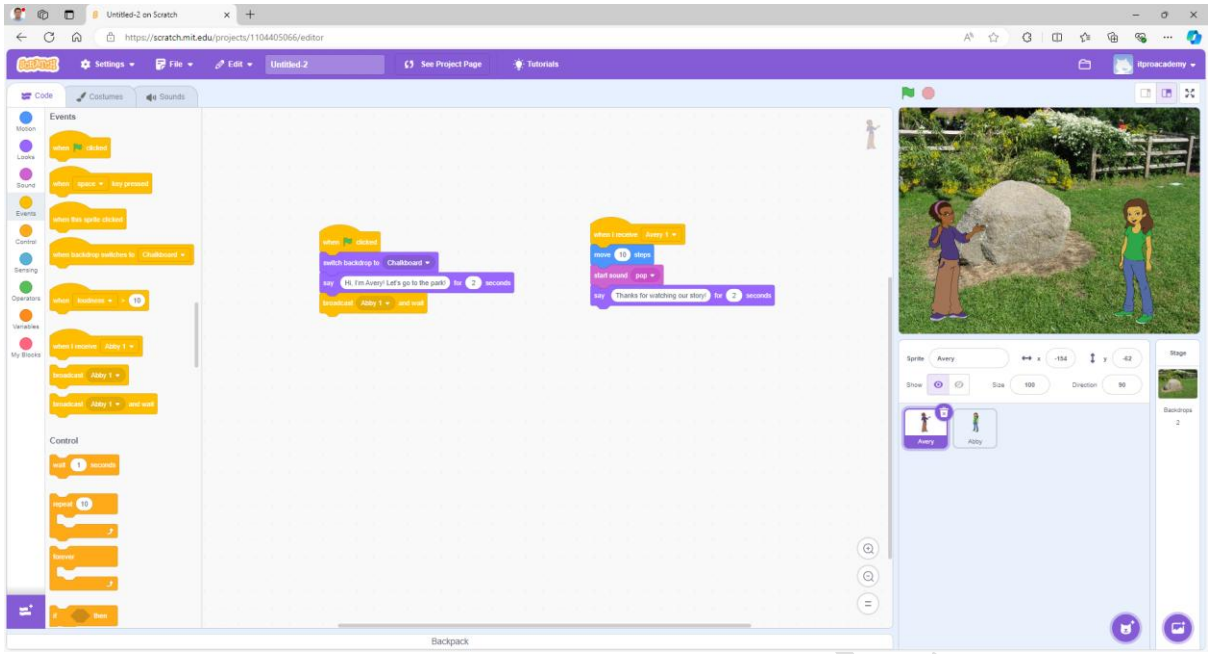
2. **Transition to Scene 2**:
   - Add a Garden-rock backdrop, by clicking on backdrop at the right-side bottom of the screen.
   - Use the **“Switch backdrop to Garden-rock”** block from Looks (select “**Garden-rock**” from the drop down after switch backdrop to).
   - Add a **“Wait 2 seconds”** block from Control to allow the audience to see the transition.
3. **Animate the Characters**:
   - In Scene 2, use motion blocks to make the characters move and play.
   - On sprite Abby, drag **“move 10 steps”** block from Looks.
   - Drag **“start sound pop”** from block sound”.
   - Add a final message: **“Thanks for watching our story!”.**
   - Drag “Say message for 2 seconds” (type the message after say **“Thanks for watching our story!”.**
   - Drag **“Broadcast Avery 1and wait”** (click on the down arrow key after broadcast, select new message and type “Avery 1”) from Events.
   - Click on sprite Avery on right side of the screen.
   - Drag **“When I receive Avery 1”** from Events.
   - Drag **“move 10 steps”** block from Looks.

- Drag **“start sound pop”** from block sound”.
- Add a final message: **“Thanks for watching our story!”.**
- Drag “Say message for 2 seconds” (type the message after say **“Thanks for watching our story!”.**

4. **Add Sound Effects (Optional)**:
   - Use **“Play sound [fun sound]”** to make the story lively.






## Challenge

1. Add more scenes to extend the story.
2. Use the **“Broadcast [message]”** block from Events to trigger actions between characters, like one sprite reacting to another.
3. Add music that matches the mood of each scene.

# 6. Introduction to Loops

**1. What Are Loops?**

Loops are coding blocks that repeat a set of instructions multiple times or forever. They are useful for actions that need to happen repeatedly without rewriting the same code.

**2. Types of Loops in Scratch**

1. **Repeat Loop**:
    - Repeats a set of actions for a specific number of times.
    - Example: Make a sprite move forward 10 times.
    - **Block**:
        - **"Repeat [number]"**: Add blocks inside this loop, and Scratch will repeat those actions the number of times specified.
2. **Forever Loop**:
    - Repeats a set of actions endlessly until you stop the program.
    - Example: A sprite spinning continuously.
    - **Block**:
        - **"Forever"**: Add blocks inside this loop, and the actions will repeat forever.

## Activity

Build a Project Where a Sprite Repeats a Simple Movement

## Objective

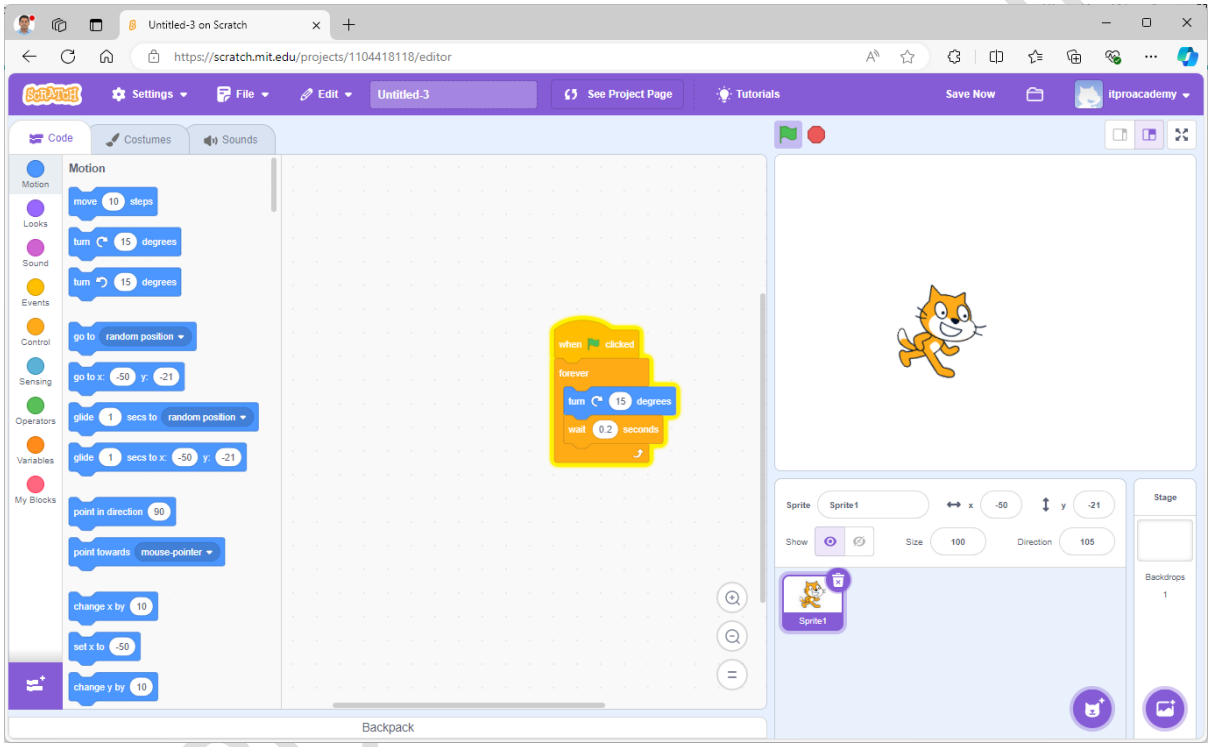
Use loops to make a sprite spin continuously or perform a repeated action.

*Steps to Create the Project*

1. **Choose Your Sprite**:
    - Select a sprite from the library, like the Scratch cat.
2. **Set Up the Movement**:
    - Drag **"When green flag clicked"** from Events.
    - Attach a **"Forever"** loop from Control.
3. **Inside the Loop**:
    - Add **"Turn 15 degrees"** from Motion.

- Add a **“Wait 0.2 seconds”** block from Control to control the spinning speed.

4. **Enhance the Movement**:
   - To make the sprite move in a circle:
     - Inside the Forever loop, add **“Move 10 steps”** and **“Turn 15 degrees”** together.
5. **Test Your Project**:
   - Click the green flag to watch your sprite spin or move in a circle.




---

## Experiment with the Repeat Loop

- Replace the **“Forever”** loop with a **“Repeat 10”** loop to make the sprite stop after 10 spins.
- Observe the difference and discuss which loop is better for specific tasks.

---

## Challenge

1. Use both **Repeat** and **Forever** loops in the same project.
   - Example: The sprite spins forever but says “Hello!” 5 times.

2. Add a backdrop to make the spinning sprite part of a story or game.
3. Change the size of the sprite in each loop cycle to create a zooming effect.

---

# 7. Introduction to Conditions

**1. What Are Conditions?**

Conditions are "if-then" statements that make decisions in a program. They allow a sprite to perform an action only when something specific happens.

- **Example**:
    - **If** the spacebar is pressed, **then** the sprite jumps.
    - **If** a sprite touches another sprite, **then** it says "Hello!".

**2. Key Scratch Blocks for Conditions**

1. **If-Then Block**:
    - Found in the Control category.
    - Executes the blocks inside only if the condition is true.
    - **Structure**:
        - **"If [condition] then"**: Place a condition (e.g., "key pressed") in the hexagonal slot.
2. **Sensing Blocks**:
    - Detect events like a keypress, mouse click, or collision.
    - Common examples:
        - **"Key [arrow] pressed?"**: Checks if a specific key is pressed.
        - **"Touching [sprite]?"**: Checks if the sprite touches another sprite or the edge of the stage.

## Activity

Create a Mini-Game Where a Character Moves When Arrow Keys Are Pressed

## Objective

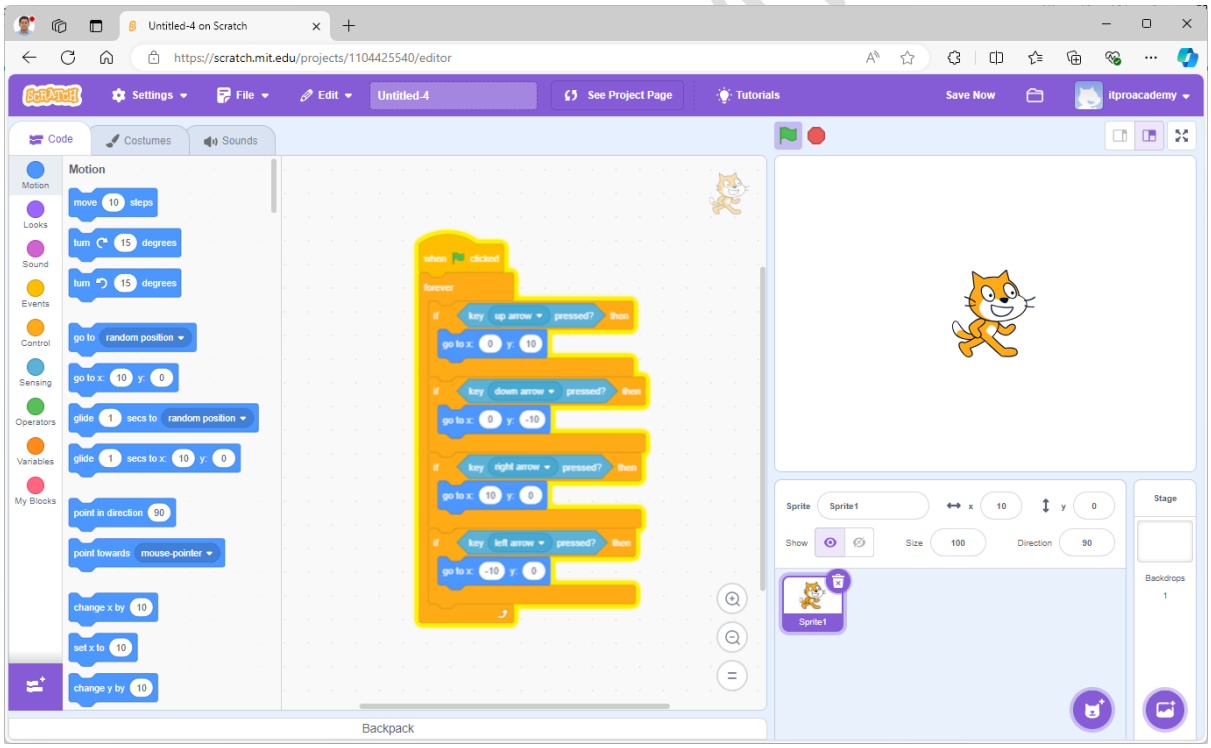
Use conditions to control a sprite's movement based on arrow key inputs.

*Steps to Create the Mini-Game*

1. **Choose Your Sprite**:
    - Select a sprite like the Scratch cat or any other character.
2. **Set Up Movement with Arrow Keys**:
    - Drag **"When green flag clicked"** from Events.

- Attach a **“Forever”** loop from Control.
- Inside the loop, use multiple **“If-then”** blocks to check for keypresses.

3. **Move the Sprite**:
   - Use the **“Key [arrow] pressed?”** block from Sensing in each **“If-then”** block.
   - Add movement commands from Motion:
     - **Move up**: **“Change y by 10”** (for the up arrow).
     - **Move down**: **“Change y by -10”** (for the down arrow).
     - **Move right**: **“Change x by 10”** (for the right arrow).
     - **Move left**: **“Change x by -10”** (for the left arrow).
4. **Test Your Mini-Game**:
   - Click the green flag and use the arrow keys to move your sprite.




## Enhancements

1. **Add Boundaries**:
   - Prevent the sprite from leaving the stage using conditions like **“If touching edge”** and stopping the movement.

2. **Add Obstacles**:
    - Use another sprite as an obstacle.
    - Add a condition: **“If touching [obstacle] then say ‘Ouch!’”.**
3. **Add a Goal**:
    - Create a target sprite and add: **“If touching [target] then say ‘You win!’”.**

---

## Challenge

1. Add sounds when the sprite moves or touches another sprite.
    - Example: Play a "whoosh" sound for movement and a "ding" sound when the sprite reaches the goal.
2. Add a score counter using variables to track points whenever the sprite reaches the target.

---

# 8. Fun with Variables

**1. What Are Variables?**

- A variable is a way to store information in your program.
- Think of it as a container that holds numbers, words, or other values that can change as the program runs.
- **Examples in Real Life**:
    - A score in a game.
    - The number of times a button is clicked.

**2. How to Create a Variable in Scratch**

1. Click on the **“Variables”** category in the Blocks Palette.
2. Select **“Make a Variable”** and name it (e.g., "Counter").
3. Choose whether the variable is for all sprites or just one sprite.

- Once created, the variable appears as a block that you can use to:
    - **Set** its value.
    - **Change** its value by adding or subtracting.
    - **Show** or **hide** it on the stage.

## Activity

Create a Counter That Increases Every Time a Sprite Is Clicked

## Objective

Learn how to use a variable to count clicks on a sprite.

*Steps to Create the Project*

1. **Choose Your Sprite**:
    - Select a sprite, such as the Scratch cat.
2. **Create the Variable**:
    - Go to the Variables category and create a variable named **"Counter"**.
    - Click on **“Make a Variable”**, enter the name **“Counter”** and select **“For this sprite only”** and click Ok.
    - The variable will appear on the stage.

3. **Set Up the Code**:
    - Drag **"When green flag clicked"** from Events.
    - Attach **"Set [Counter] to 0"** from Variables to reset the counter at the start.
4. **Add the Click Event**:
    - Drag **"When this sprite clicked"** from Events.
    - Attach **"Change [Counter] by 1"** from Variables.
    - Optionally, add a sound: **"Play sound [pop]"** from Sound.
5. **Test Your Counter**:
    - Click the sprite and watch the counter increase on the stage.

---

## Enhancements

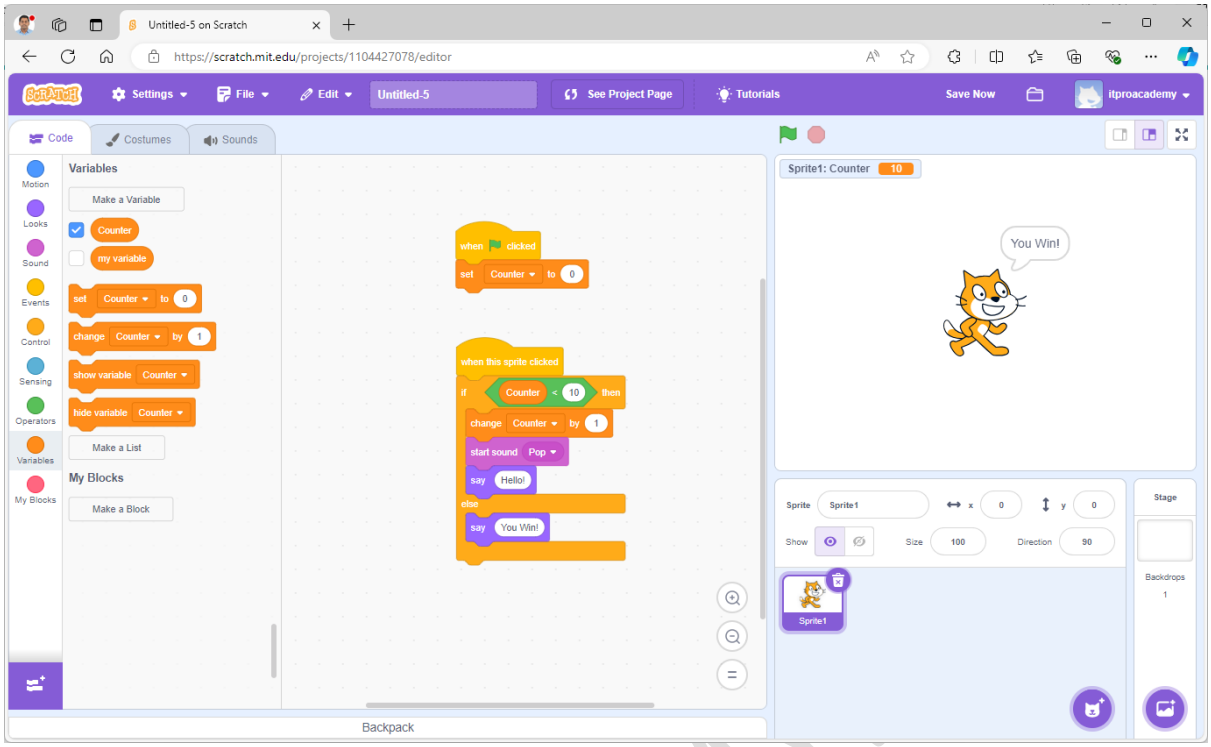
1. **Display a Message**:
    - Add a **"Say [message] for 2 seconds"** block to make the sprite say how many times it's been clicked.
2. **Add Animation**:
    - Use motion blocks to make the sprite jump or spin each time it's clicked.
3. **Set a Goal**:
    - Add a condition: **"If [Counter] = 10, then say 'You win!'"**

---

## Challenge

1. Add another sprite with its own counter.
    - Compare the counters to see which sprite gets clicked more.
2. Use the **"Reset"** button:
    - Create a reset button sprite.
    - Program it to set the counter back to 0 when clicked.

---



# 9. Simple Games

## Objective

Apply the concepts you've learned so far to create fun and interactive games. In this section, we will make two simple games: a **maze game** and a **clicker game**. You'll use what you know about **motion**, **variables**, **conditions**, and **events** to bring your games to life!

## Projects

### Maze Game

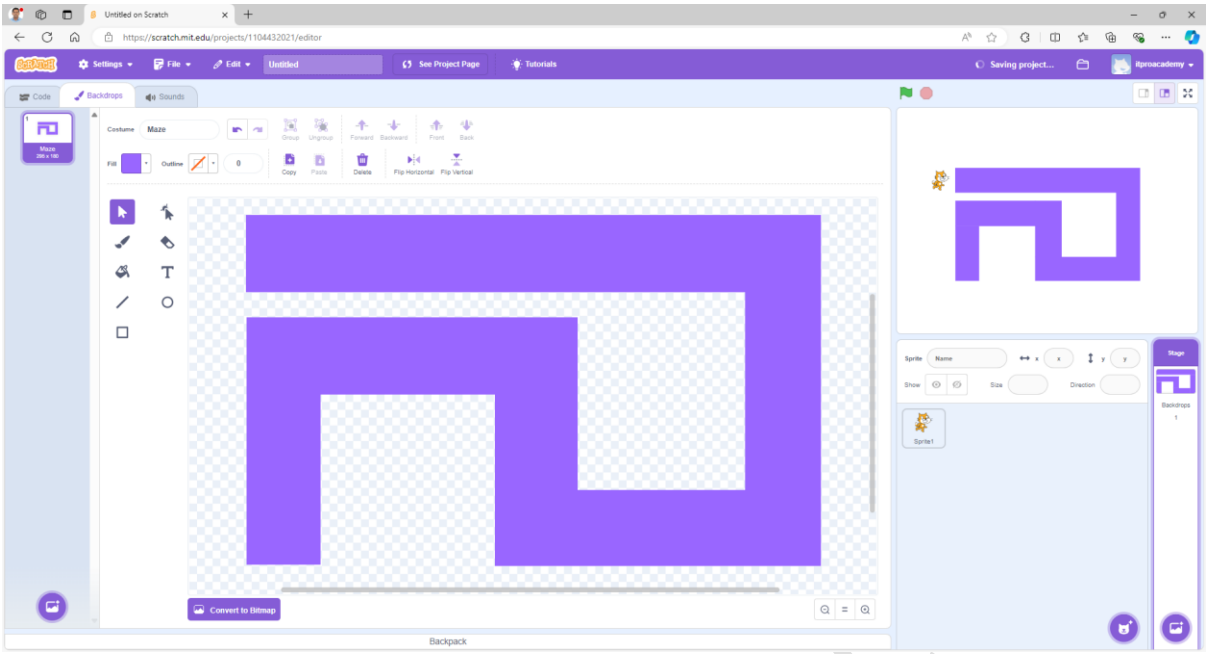
Navigate a Sprite Through a Maze Using Arrow Keys

#### *Objective*

Use the arrow keys to move a sprite through a maze while avoiding walls and reaching a goal.

#### Steps to Create the Maze Game:

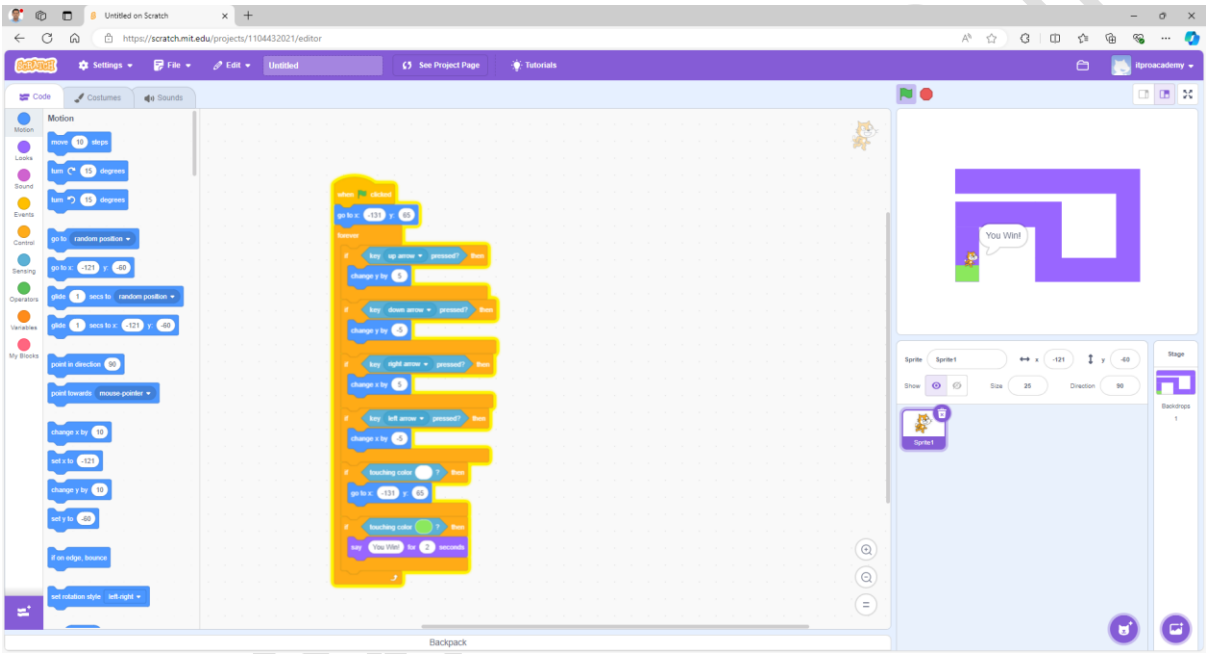
1. **Set Up the Maze**:
    - Choose a backdrop or draw your own maze using the Scratch backdrop editor.
    - Add walls and paths to create the maze design.
    - Select the Backdrops, at the right-side bottom of the screen.
    - Go to Backdrops, at the left side top of the screen, use the tools and draw your maze.
    - Use rectangle, go to fill and reduce the size of the border.
    - Use select (arrow) for selecting and adjusting the size of the rectangle (just like shown in the picture below).



2. **Create the Player Sprite**:
    - Select or draw a sprite to be your player (e.g., Scratch cat).
    - Reduce the size of the sprite to 30.
    - Place the sprite at the starting position of the maze.
3. **Set Up Movement**:
    - Use the **"When green flag clicked"** block to start the game.
    - Inside the **Forever** loop, add the **"If [key pressed] then"** blocks from the **Sensing** category.
        - **Up arrow**: **"Change y by 10".**
        - **Down arrow**: **"Change y by -10".**
        - **Right arrow**: **"Change x by 10".**
        - **Left arrow**: **"Change x by -10".**
4. **Avoid the Walls**:
    - Add an **If-Then** block to check if the sprite is touching a wall:
        - **"If touching [color] then".**
        - Inside the **If-Then**, add **"Go to [start position]"** to reset the sprite to the starting point.
5. **Set the Goal**:

- Create a goal sprite (e.g., a flag).
- Add a condition:
  - **“If touching [goal sprite] then”.**
  - **“Say ‘You Win!’ for 2 seconds”.**
  - Optionally, add a score to keep track of time or steps taken.

6. **Test Your Game**:
   - Click the green flag and try navigating the sprite through the maze using the arrow keys.




## Clicker Game

Click on a Moving Sprite to Score Points

### *Objective*

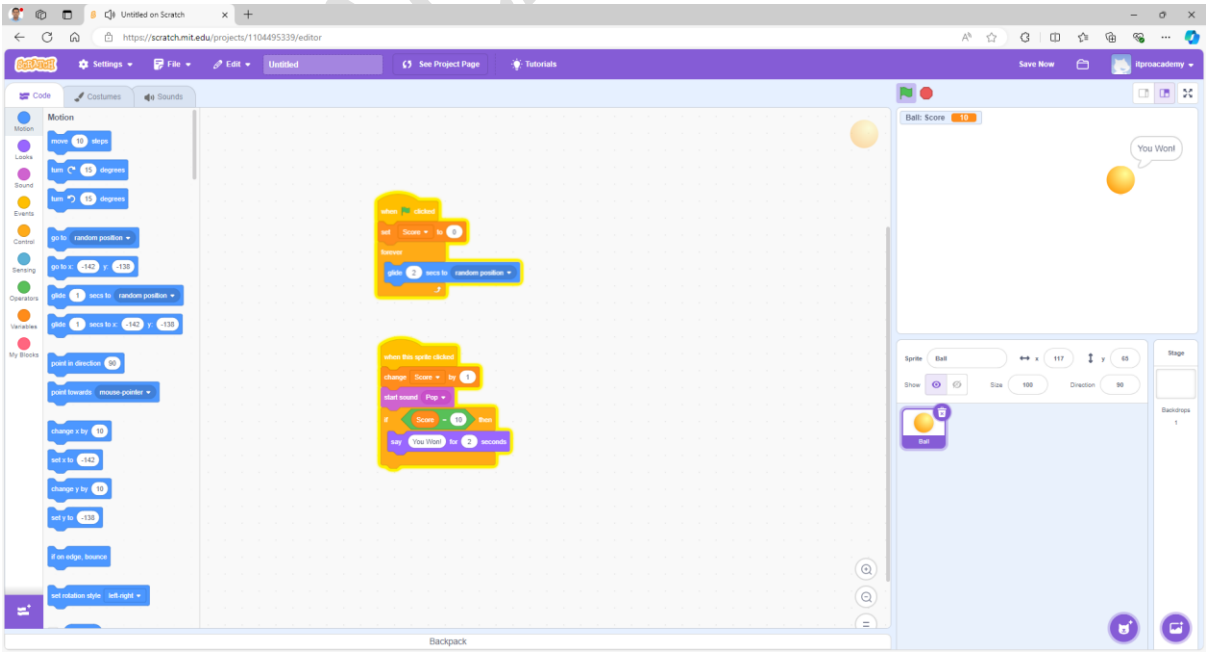
Click on a moving sprite to score points and keep track of your score.

### Steps to Create the Clicker Game

1. **Create the Player Sprite**:
   - Choose or draw a sprite to be clicked on (e.g., a bouncing ball or a character).
   - Make the sprite move randomly around the stage.
2. **Add a Score Variable**:
   - Create a variable named **“Score”**.

- Set the initial score to 0 using the **“Set [Score] to 0”** block.

3. **Make the Sprite Move**:
   - Use the **“Glide”** block to make the sprite move randomly.
   - Example: **“Glide 2 secs to x: [random number] y: [random number]”**.
   - Place this inside a **Forever** loop for continuous movement.
4. **Set Up the Clicking Mechanism**:
   - Use the **“When this sprite clicked”** block.
   - Inside, add **“Change [Score] by 1”** to increase the score each time the sprite is clicked.
   - Optionally, add a sound when the sprite is clicked (e.g., **“Play sound [pop]”**).
5. **End the Game**:
   - After a certain number of clicks, display a message:
     - **“If [Score] = [10], then say ‘You Won!’ for 2 seconds”**.
   - You can also add a timer to make the game time bound.
6. **Test Your Game**:
   - Click the sprite as it moves around and watch your score increase.



## Enhancements for Both Games

1. **Add Multiple Levels (Maze Game)**:
    - As the player reaches the goal, increase the difficulty by creating a new maze or adding more obstacles.
2. **Add Timer (Clicker Game)**:
    - Set a timer and stop the game after a certain time. Display the final score at the end.
3. **Add Sound Effects**:
    - Add sounds to make the game more engaging. Use sounds for when the sprite moves, when clicked, or when the game ends.

## Challenge

1. **Maze Game Challenge**:
    - Create a timer that counts down from 30 seconds. The player must reach the goal before time runs out.
2. **Clicker Game Challenge**:
    - Add obstacles to the clicker game. If the player clicks on the wrong sprite, their score goes down.

# 10. Creative Projects

**1. Encourage Free Exploration and Creativity**

In this section, we encourage you to unleash your creativity by using everything you've learned to design and build projects that are fun and unique. Whether it's creating animations, games, or simple tools, Scratch allows you to explore a wide variety of ideas. Don't be afraid to experiment and think outside the box!

## Examples of Creative Projects

### 1. Design a Greeting Card with Animations and Sounds

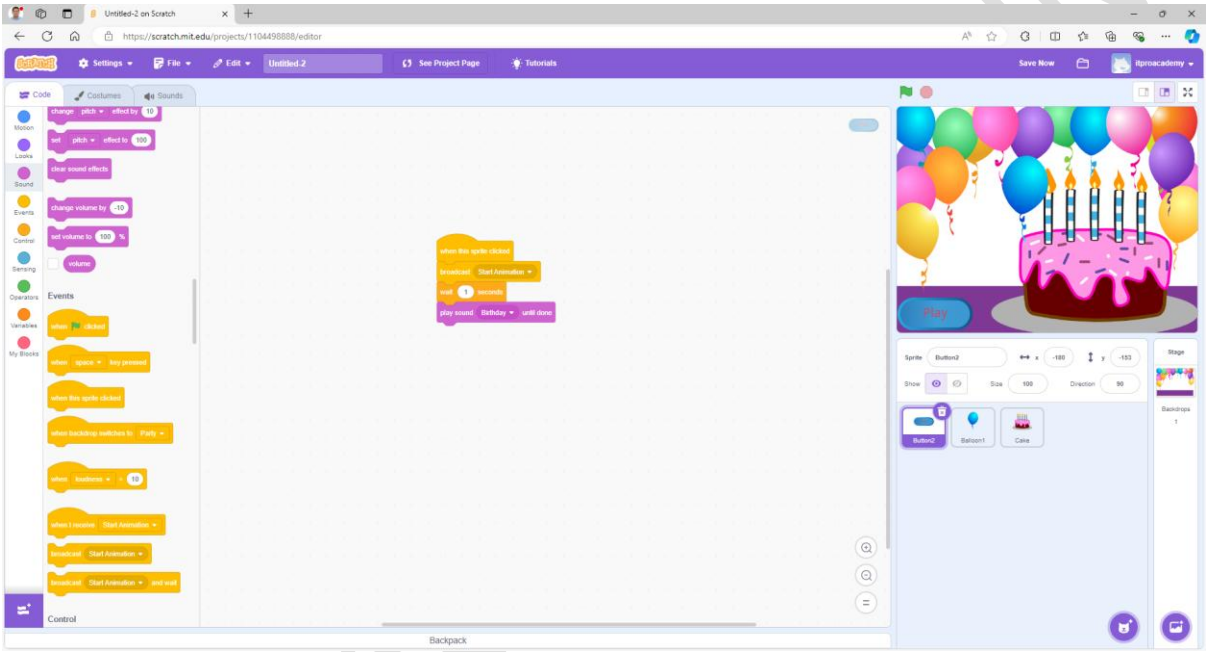
*Objective*

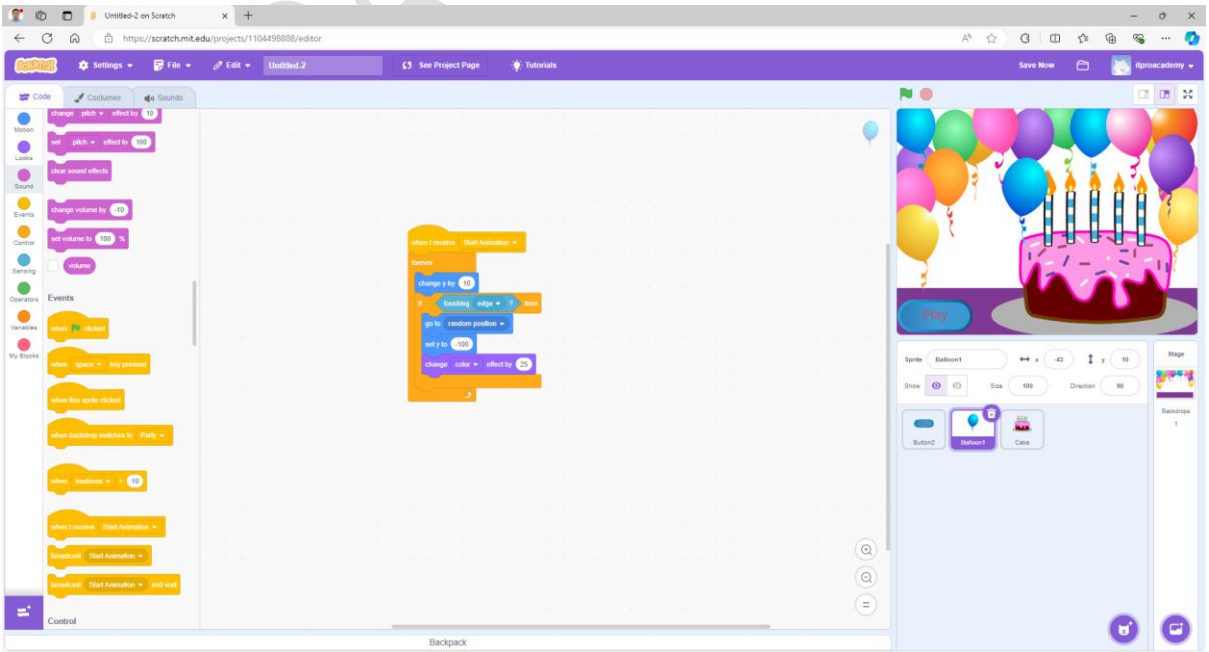
Combine animations, sounds, and text to create a personalized greeting card for a special occasion, such as a birthday or festival.

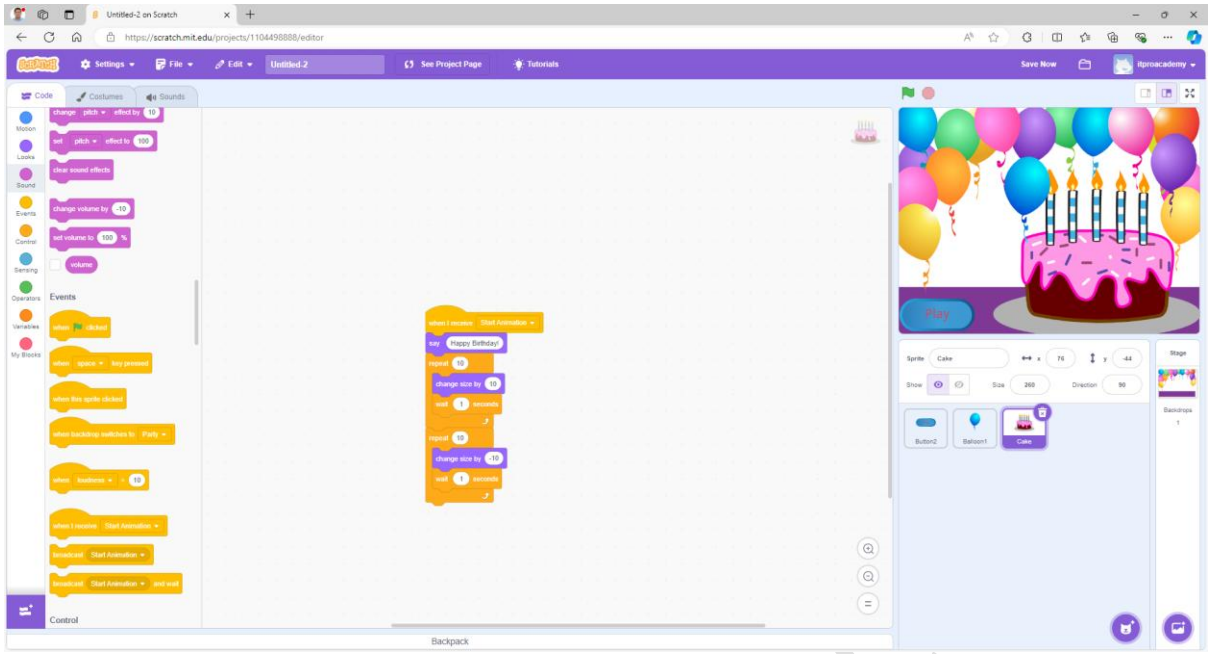
#### Steps to Create a Greeting Card:

1. **Choose Your Theme**:
    - Decide the occasion for your greeting card (e.g., birthday, festival, or congratulations).
    - Pick a backdrop that fits the theme. You can use Scratch's backdrops or create your own.
2. **Add Text**:
    - Use the **"Say [text] for [time]"** block to make your sprite say something like "Happy Birthday!".
    - Customize the text by changing the font, size, and color to match the theme.
3. **Add Animations**:
    - Use motion blocks (e.g., **"Glide"**, **"Move"**, or **"Turn"**) to animate your sprite.
    - Create a sequence of movements or have the sprite spin, jump, or dance for fun.
4. **Add Sounds**:
    - Add a festive sound, like music or a sound effect, using the **"Play sound [sound] until done"** block.
    - You can also add a voice message or sound effects when the sprite says something.

5. **Personalize the Card**:
    - You can include more sprites that represent people or objects (e.g., balloons for a birthday).
    - Use **“Change size by”** or **“Change color effect by”** to make the sprites stand out in different ways.
6. **Test Your Greeting Card**:
    - Click the green flag to watch the greeting card come to life. Play with different designs and animations to make your card even more unique.







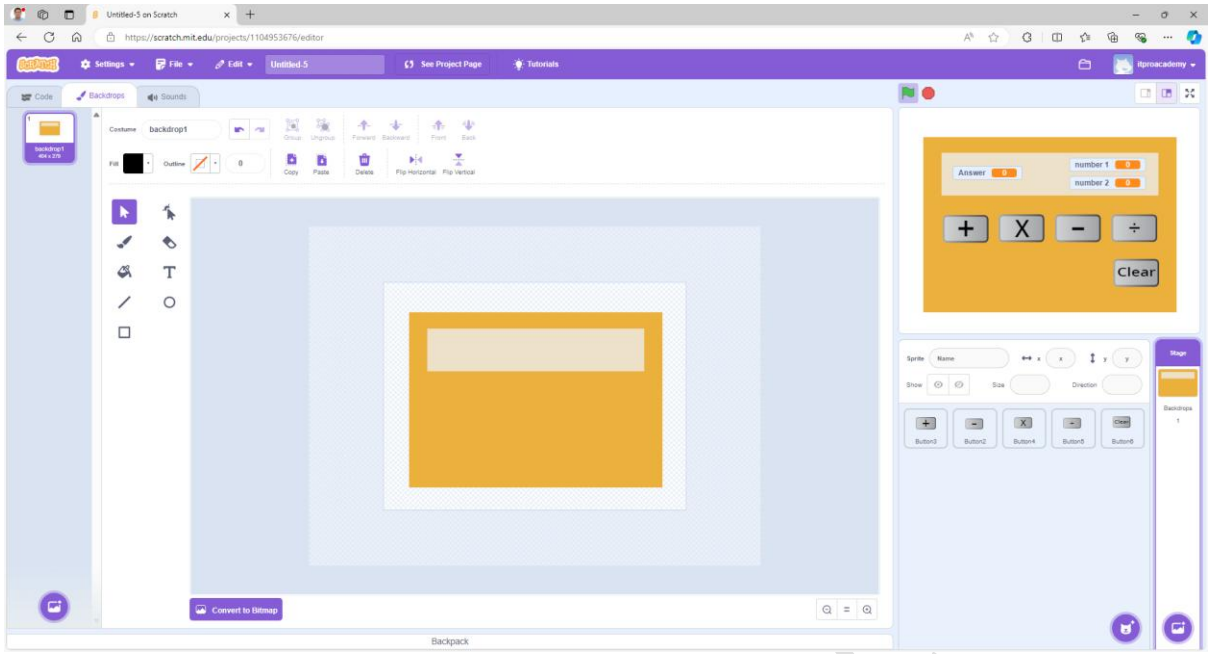
## 2. Make a Simple Calculator Using Scratch

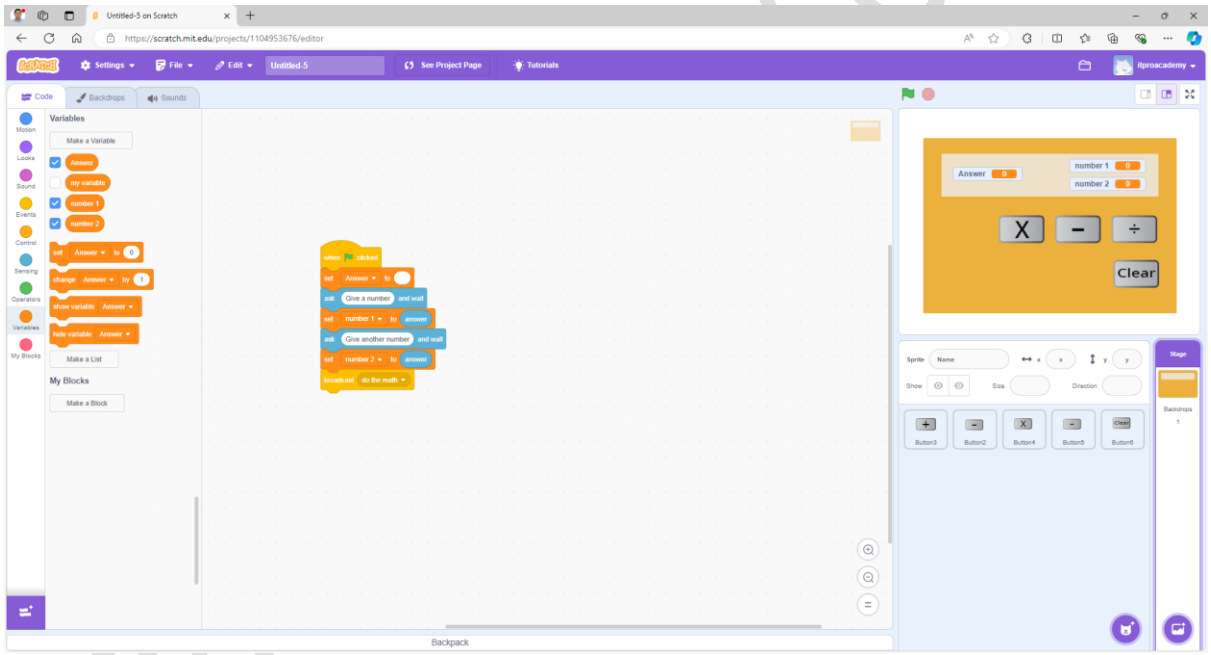
*Objective*

Create a calculator that can add, subtract, multiply, or divide two numbers using Scratch.

### Steps to Create the Calculator:

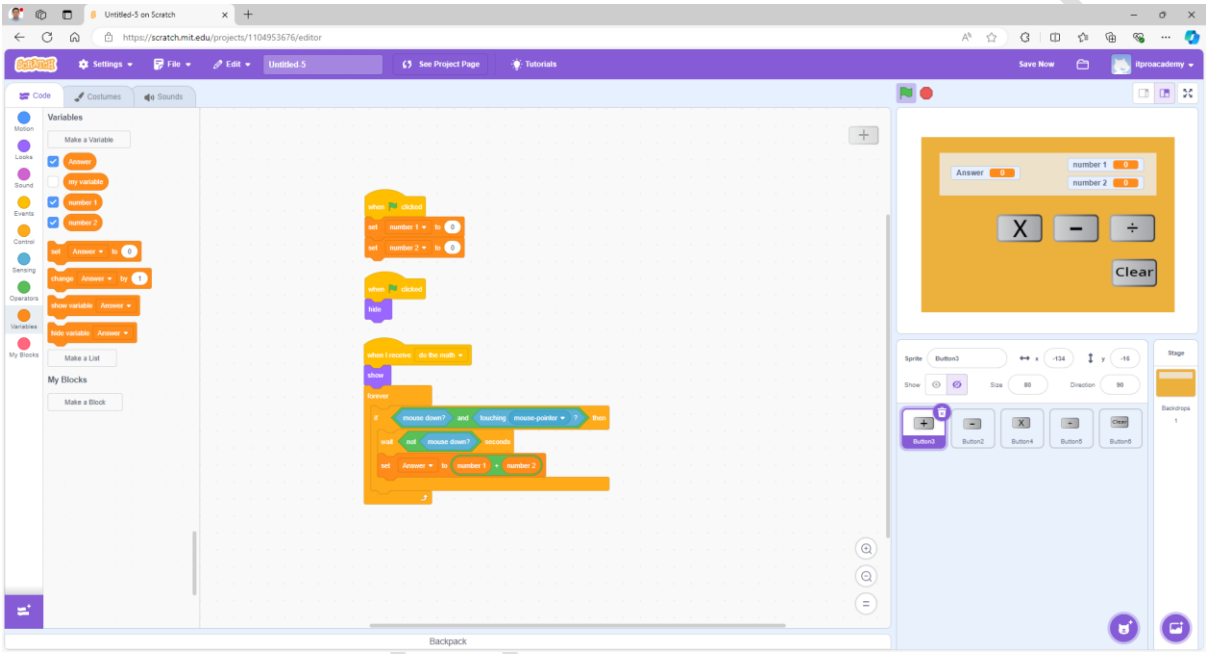
1. **Design the Layout**:
    - Create a backdrop with calculator, using rectangle object.
    - Choose or create a sprite that will act as the "calculator operator buttons". This sprite will show the operations.
    - Add mathematical operations (+, –, *, ÷). Each button will be its own sprite.
    - Write the code to set the variable **Answer** to **0**.
    - Ask user to enter the value for the variable **Number 1**.
    - Ask user to enter the value for the variable **Number 2**.
    - Store the values in the respective variables.

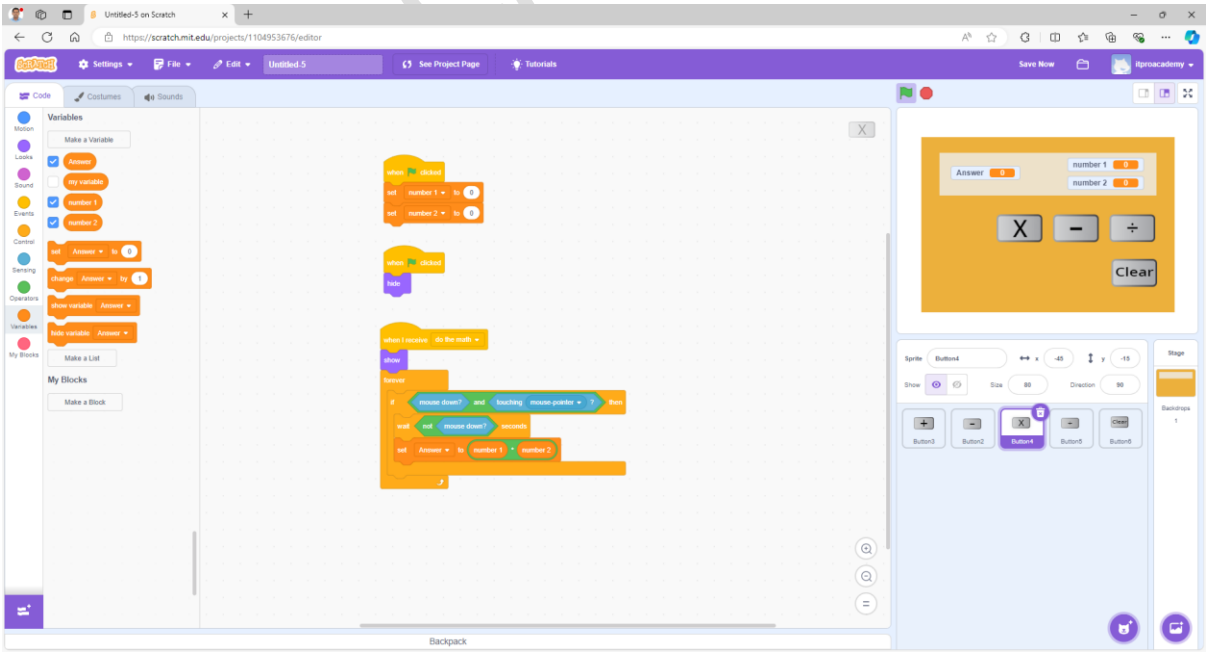


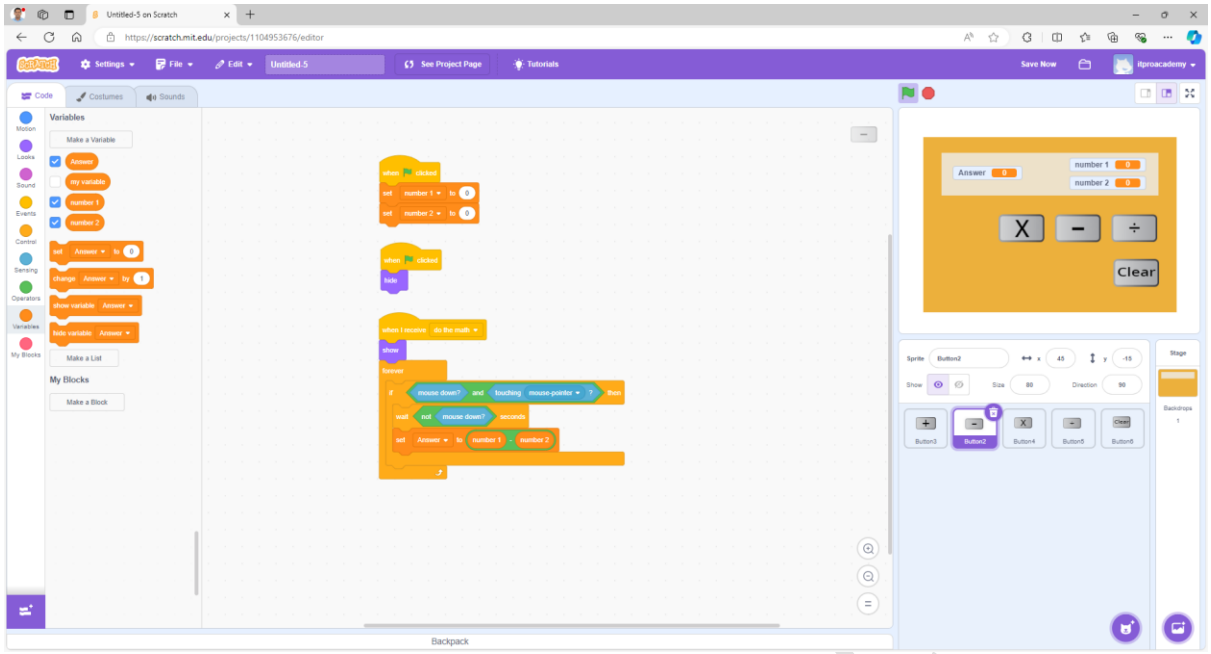



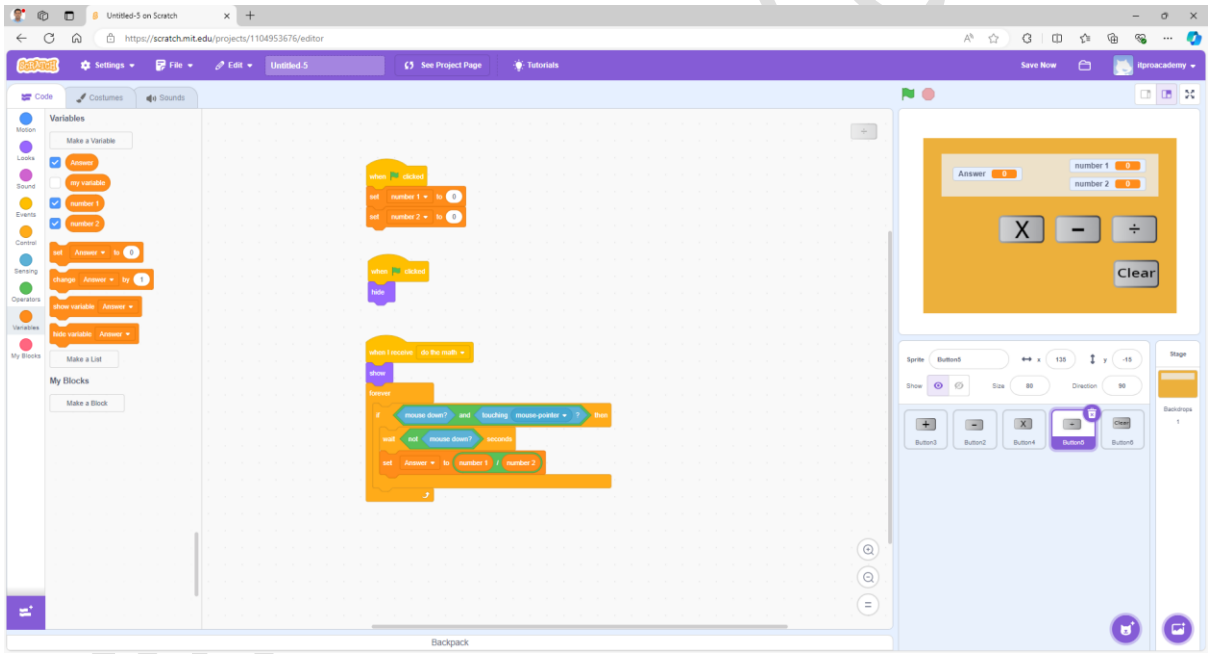

2. **Set Up Variables**:
    - Create variables for the two numbers and the answer (e.g., **"Number 1"**, **"Number 2"**, **"Answer"**).
    - Use these variables to store the numbers entered and the result of the operation.
3. **Set Up the Operations**:
    - For each operation button (+, –, *, ÷), use the **“When this sprite clicked”** block to perform the calculation.

- Example: For addition, use a block like **“Set [Answer] to [Number 1] + [Number 2]”**.
- Before you enter the number for **Number 1** or **Number 2** variable, it should be set to **0**.
- Operation sprits (buttons) should be visible only after you enter value for the variable **Number 1** and **Number 2**.
- Each operation sprites (buttons) should have the same settings.

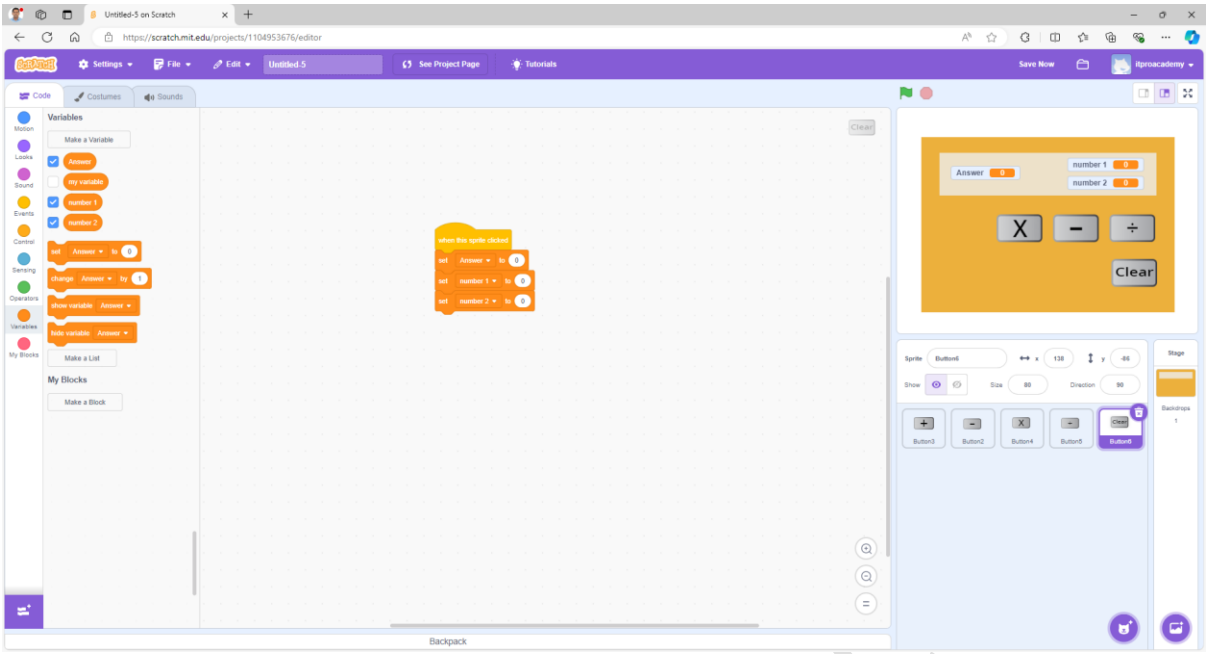


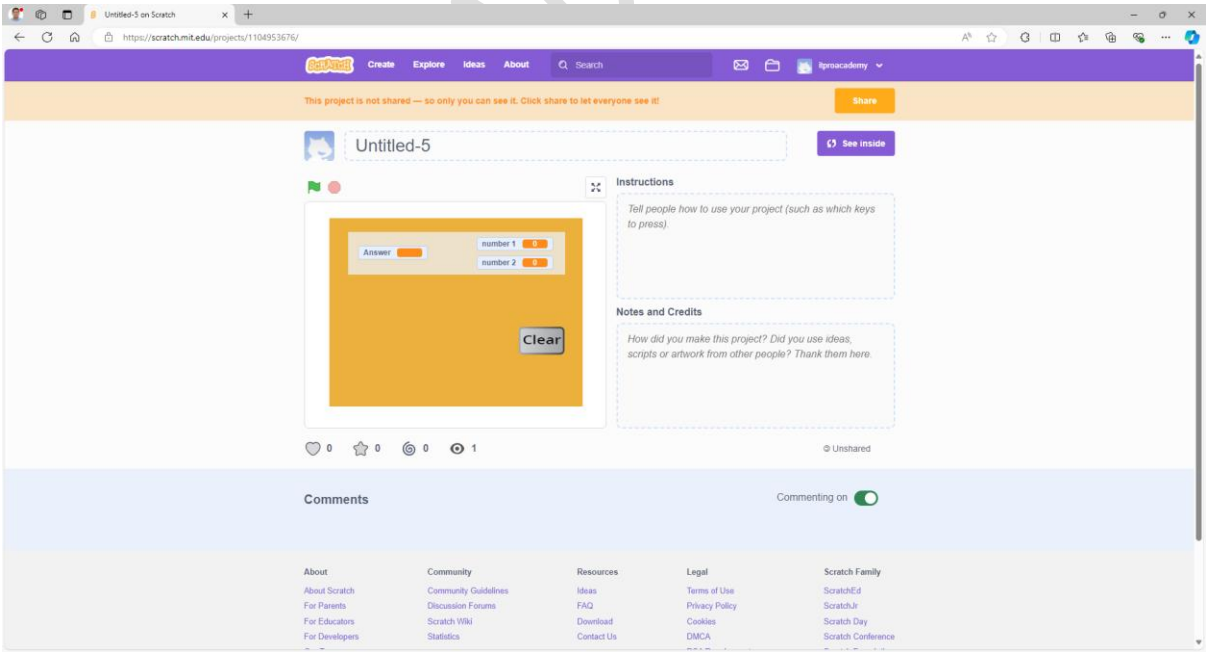


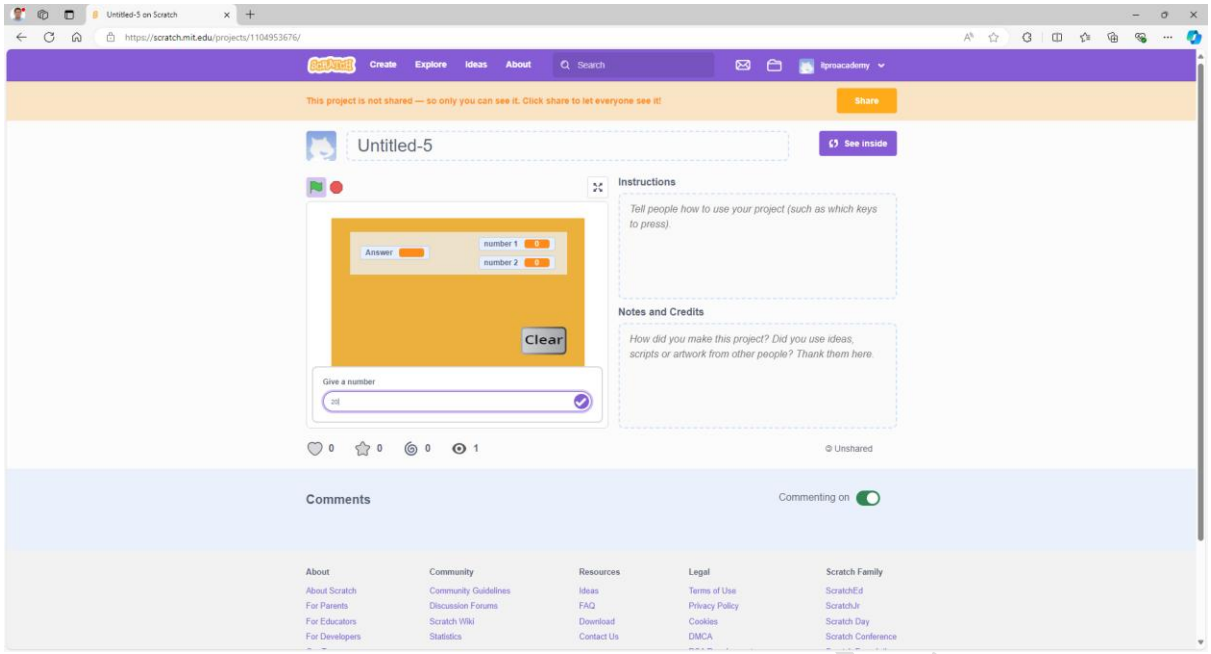



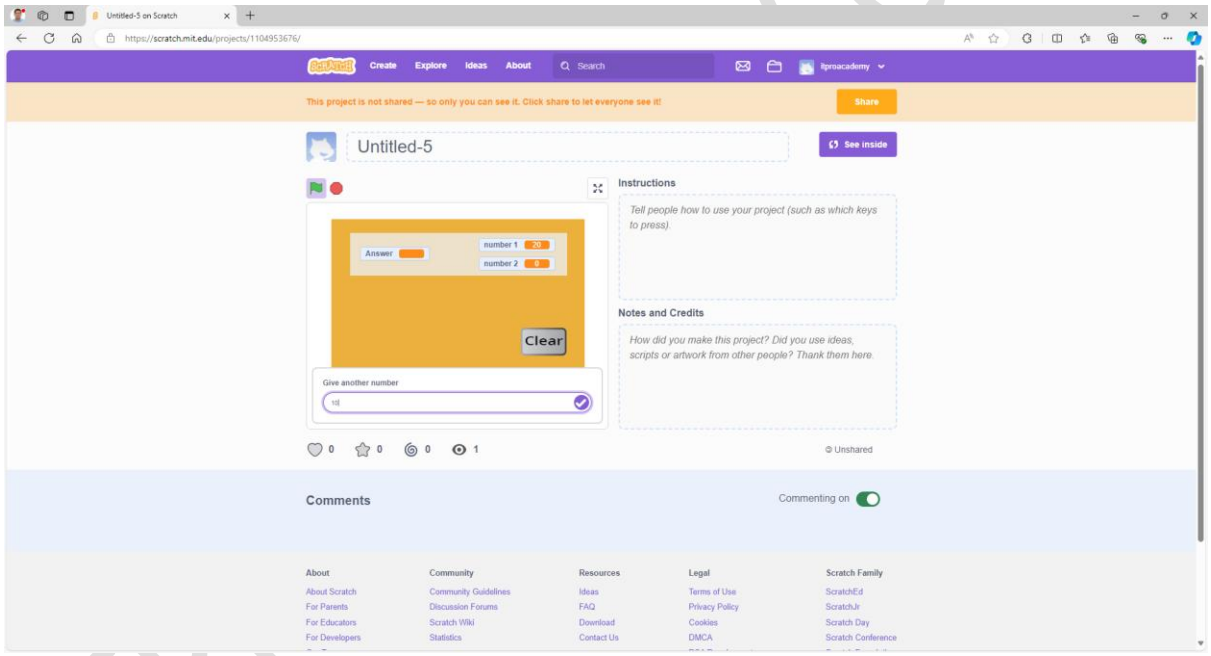

4. **Clear the Screen**:
   - Add a **"Clear"** button to reset the calculator after each calculation.
   - When clicked **"Clear"** sprite, it should set the value **0** for **Number 1**, **Number 2** and **Answer**.

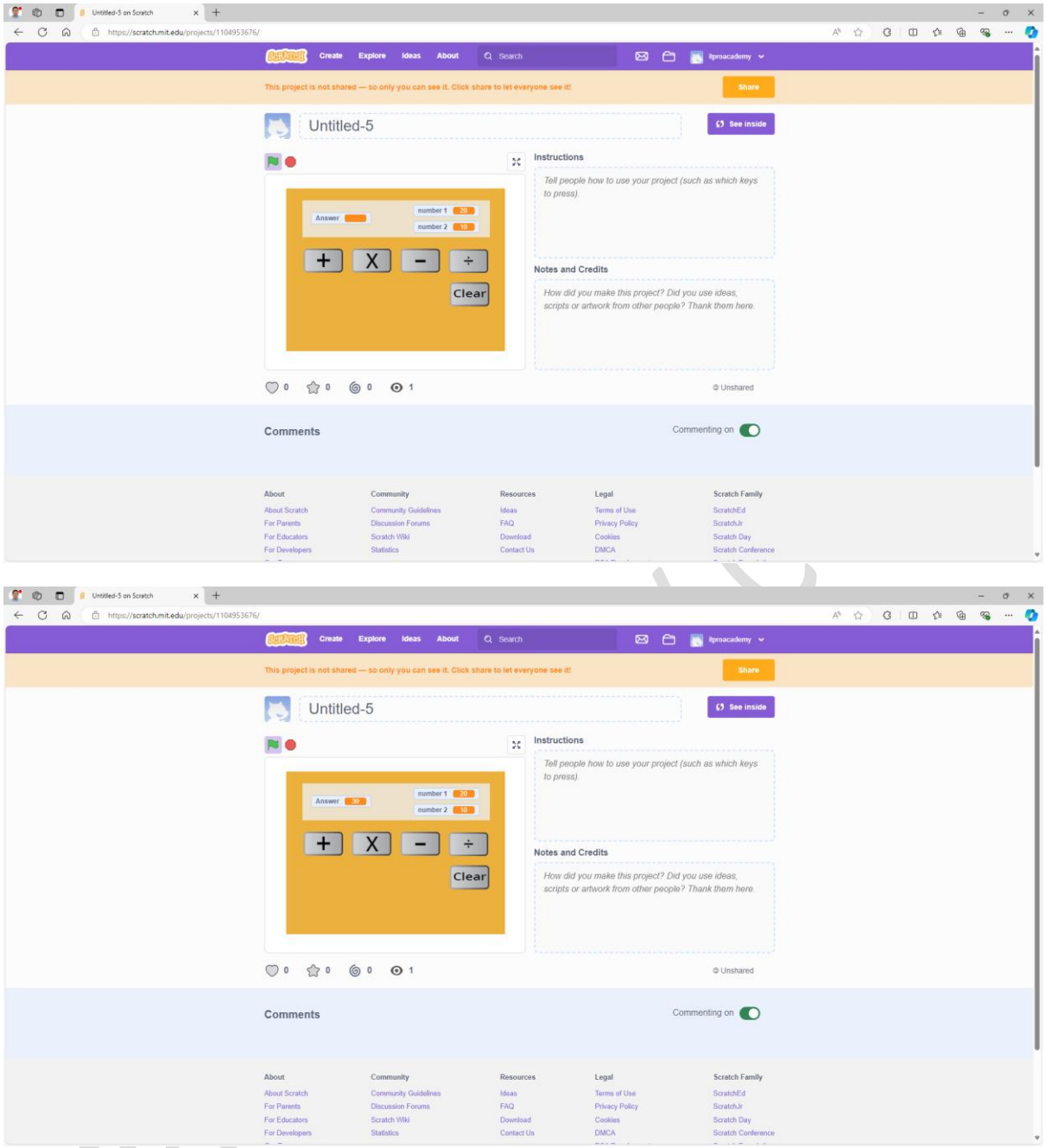



5. **Test the Calculator**:
    - Try clicking the number and operation buttons and see if the result shows correctly.
    - Check if the calculator performs addition, subtraction, multiplication, and division accurately.




Untitled-5 on Scratch
https://scratch.mit.edu/projects/1104953676/
Create
Explore
Ideas
About
Search
ltproacademy
This project is not shared — so only you can see it. Click share to let everyone see it!
Share
Untitled-5
See inside
Instructions
Tell people how to use your project (such as which keys to press).
Answer
number 1
number 2
Clear
Give a number
Notes and Credits
How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.
0
0
0
1
Unshared
Comments
Commenting on
About
About Scratch
For Parents
For Educators
For Developers
Community
Community Guidelines
Discussion Forums
Scratch Wiki
Statistics
Resources
Ideas
FAQ
Download
Contact Us
Legal
Terms of Use
Privacy Policy
Cookies
DMCA
Scratch Family
ScratchEd
ScratchJr
Scratch Day
Scratch Conference


Untitled-5 on Scratch
https://scratch.mit.edu/projects/1104953676/
Create
Explore
Ideas
About
Search
ltproacademy
This project is not shared — so only you can see it. Click share to let everyone see it!
Share
Untitled-5
See inside
Instructions
Tell people how to use your project (such as which keys to press).
Answer
number 1
20
number 2
0
Clear
Give another number
Notes and Credits
How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.
0
0
0
1
Unshared
Comments
Commenting on
About
About Scratch
For Parents
For Educators
For Developers
Community
Community Guidelines
Discussion Forums
Scratch Wiki
Statistics
Resources
Ideas
FAQ
Download
Contact Us
Legal
Terms of Use
Privacy Policy
Cookies
DMCA
Scratch Family
ScratchEd
ScratchJr
Scratch Day
Scratch Conference







## Enhancements for Both Projects

1. **Add Visual Effects (Greeting Card)**:
    - Use **"Change color effect"** or **"Change size"** blocks to make your sprites change as the greeting card animation plays.
    - Make the card more dynamic by adding more sprites or changing the backdrop during the animation.
2. **Add Sound Effects (Calculator)**:

- Use **“Play sound”** when buttons are pressed or when the calculation is completed. This will make the calculator feel more interactive.

3. **Add Multiple Pages (Greeting Card)**:
    - You can make the greeting card have multiple scenes. Each backdrop change can represent a different message or event, such as "Have a great day!" after the initial greeting.

---

## Challenge

1. **Greeting Card Challenge**:
    - Design a greeting card that reacts to user interaction. For example, let the card change colors when clicked or animate when the mouse pointer is over it.
2. **Calculator Challenge**:
    - Add more advanced features to the calculator, such as allowing the user to input decimal numbers, perform more complex calculations, or even add a "Clear All" button.

---

# 11. Assessment and Showcasing

## Assessment

The goal of this section is to evaluate the students' understanding of Scratch programming concepts such as loops, conditions, and variables by recreating a project. This activity will help you identify how well the students have grasped these concepts and give them a chance to demonstrate their skills.

### Assessment Activity

Recreate a Simple Project

#### *Objective*

Students will recreate a basic project using loops, conditions, and variables to demonstrate their learning.

**Steps for Assessment**:

1. **Choose a Simple Project**:
   Select a simple project for the students to recreate. It could be:
   - A **maze game** where the sprite moves through the maze using arrow keys.
   - A **clicker game** where the score increases each time the sprite is clicked.
   - A **greeting card** project with animations and sounds.
2. **Provide Guidelines**:
   - Students should include **loops** (e.g., **Repeat** or **Forever** loops) in their project to control repeated actions like movement or animation.
   - They should use **conditions** (e.g., **If-Then** blocks) to create interactions, such as detecting key presses or sprite collisions.
   - **Variables** should be used to store values like scores, time, or positions (e.g., a score counter for a game).
3. **Project Creation**:
   Allow students to work on the project. Encourage them to:
   - Plan out their project before coding.
   - Experiment with different types of blocks, including loops, conditions, and variables.
   - Test and refine their project until it works as expected.

4. **Evaluate**:
   Assess each student's project based on:
   - **Correct use of loops**: Are the loops used to repeat actions effectively?
   - **Appropriate use of conditions**: Does the project have if-then statements for decision-making?
   - **Effective use of variables**: Are variables used for storing and updating values, such as score or time?

## Showcasing

Giving students the opportunity to present their projects helps them build confidence and allows for peer learning. It also gives them a chance to showcase their creativity and problem-solving skills.

### Showcasing Activity

Presenting Projects to the Class

*Objective*

Students will present their completed Scratch projects to the class, explain their code, and share their creative process.

#### Steps for Showcasing:

1. **Prepare for the Presentation**:
   - Students should be prepared to explain how their project works, including:
     - The purpose of the project.
     - How they used loops, conditions, and variables.
     - Any challenges they faced and how they solved them.
2. **Present the Projects**:
   - Each student takes turns presenting their project to the class. Encourage them to:
     - Show the project in action (e.g., playing the game or interacting with the animation).
     - Talk about the coding blocks they used.

- Share any special features or creative touches they added to their project.

3. **Provide Feedback and Encouragement**:
   - After each presentation, provide constructive feedback:
     - Highlight what was done well (e.g., “Great use of loops to animate the sprite!”).
     - Offer suggestions for improvement (e.g., “You could add more levels to make the game more challenging.”).
   - Encourage students to ask questions or suggest improvements to their peers' projects.
4. **Celebrate Creativity**:
   - Recognize the unique aspects of each student’s project. Celebrate their creativity and effort.
   - Create a positive environment where students feel proud of their work and are motivated to continue learning and experimenting.

# Resources for Students

Learning Scratch is fun and exciting! Here are some great resources to help you become a Scratch programming expert.

## Scratch Tutorials:

1. **Built-in Tutorials in Scratch**
   Scratch offers several interactive tutorials that help you get started with coding. You can find them easily by clicking on the "Tutorials" tab on the Scratch website or in the Scratch app. These tutorials guide you step by step to create projects like animations, games, and stories.

2. **Online Resources:**

   - **Scratch Website**
     The official Scratch website is the best place to start. You can explore projects created by others, get inspiration, and even share your own creations with the Scratch community.

   - **Interactive Coding Exercises on Platforms like Tynker**
     Tynker provides fun, game-based learning for students to learn coding in a way that's similar to Scratch. It has interactive exercises for beginners that help reinforce Scratch concepts.

## Books:

1. **"Scratch Programming for Kids" by Raj, Nithin**
   This book is a great beginner-friendly guide that introduces the concepts of Scratch programming to kids. It provides simple explanations and fun exercises to help you get started with Scratch.

2. **"Learn to Program with Scratch" by Majed Marji**
   This book helps kids understand basic programming concepts in an easy-to-follow way. It focuses on creating interactive stories and games using Scratch.

# Teacher/Parent Tips

Supporting students while they learn Scratch is key to their success. Here are some tips for teachers and parents:

**1. Encourage Exploration**

- Let students explore the Scratch interface and play with different blocks and features. The more they explore, the more they will understand how things work.

Scratch is designed to be user-friendly, so students can experiment without fear of making mistakes.

**2. Focus on Creativity**

- Scratch is all about creativity. Encourage students to design projects based on topics they enjoy, like making games about animals, creating a story about their favourite character, or designing a dancing robot. Creativity makes learning more engaging and fun.

**3. Build Together**

- Instead of just instructing students to complete tasks, try building projects together. Collaborative projects give students a chance to ask questions, discuss ideas, and share their own unique ways of solving problems. Work together on projects like simple games or animations to foster teamwork and problem-solving skills.

**Would You Like Specific Lesson Plans or Downloadable Material?**

If you'd like, I can provide tailored lesson plans or downloadable material to help you in your Scratch learning journey. Just let me know what you need, and I can help make the learning experience even better!

Please reach out to us on our website https://ITProAcademy.co.in

By using these resources, students can confidently start coding with Scratch and unlock the potential to create their own amazing projects! Happy coding!

# Questions with Answers

Here are 10 questions with answers based on the content of the Scratch Learning Material for Primary school student's document:

---

**1. What is Scratch, and why is it fun?**

**Answer:** Scratch is a visual programming language designed to make coding simple and engaging by using colourful, puzzle-like blocks. It is fun because it allows users to create interactive games, animations, and stories while encouraging creativity and logic.

---

**2. What are sprites in Scratch?**

**Answer:** Sprites are the characters or objects in Scratch projects. They can be people, animals, or any item that performs actions on the stage.

---

**3. How can you make a sprite move in Scratch?**

**Answer:** To make a sprite move, you can use the **"Move [x] steps"** block from the Motion category. Drag it to the script area and click it to see the sprite move.

---

**4. What is the purpose of the 'Events' blocks in Scratch?**

**Answer:** Event blocks are used to trigger actions in Scratch. For example, the **"When green flag clicked"** block starts a script when the green flag is clicked.

---

**5. What is a loop, and how is it used in Scratch?**

**Answer:** A loop is a block that repeats a set of actions multiple times. In Scratch, the **"Repeat [x]"** or **"Forever"** blocks are used to repeat actions like moving or playing sounds.

---

**6. What are backdrops, and how do they enhance a Scratch project?**

**Answer:** Backdrops are the backgrounds of the stage in Scratch projects. They help set the scene or mood for animations and stories by creating an immersive environment.

---

**7. How can you use conditions in Scratch to make decisions?**

**Answer:** Conditions in Scratch are implemented using the **"If [condition] then"** block. For example, you can make a sprite jump if the spacebar is pressed by checking the condition **"Key [space] pressed?"**.

---

**8. How can variables be used in Scratch?**

**Answer:** Variables in Scratch store information such as numbers or text. They can be used to track scores in a game, count clicks, or store player names.

---

**9. What steps are involved in creating a simple animation in Scratch?**

**Answer:**

1. Choose a sprite with multiple costumes.
2. Use the **"Next costume"** block to switch costumes.
3. Add motion blocks like **"Move 10 steps"** or **"Turn [x] degrees."**
4. Use a loop to repeat these actions.

---

**10. How can you create a simple game in Scratch?**

**Answer:**

1. Design the game layout (e.g., a maze or clicker game).
2. Use motion blocks to move the sprite.
3. Implement conditions to detect collisions or interactions.
4. Add variables to track scores or time.
5. Test and refine the game.

# About ITProAcademy.co.in

**Empowering Learners in the Digital Age**


ITProAcademy.com is a premier online platform dedicated to equipping individuals with the technical knowledge and skills required to thrive in the ever-evolving world of IT and cloud technologies. Established with a mission to bridge the gap between aspirations and achievements, ITProAcademy offers cutting-edge training programs tailored for both students and professionals.

With a strong focus on certification-based learning, ITProAcademy.com specializes in areas such as Microsoft Azure, AWS, Google Cloud, Artificial Intelligence, AIOps, Automation, Security, and more. The platform provides a comprehensive suite of services, including certification exam preparation, deep-dive webinars, and real-world scenario-based training.

**Key Highlights of ITProAcademy.co.in:**

- **Programming Languages Made Fun and Easy for Kids & Students:**
  Our interactive courses in **HTML, C++, Java, and Python** make learning to code fun and engaging. We inspire young learners to explore their creativity, develop problem-solving skills, and build a strong coding foundation, all while enjoying the journey. Join us to turn curiosity into coding skills and imagination into innovation!
- **Expert-Led Training:** Courses are designed and delivered by seasoned industry professionals with hands-on experience in cloud technologies, infrastructure automation, and cybersecurity.
- **Certifications for Growth:** Dedicated programs for Microsoft certifications, cloud computing, AI, and other emerging technologies to help learners stay ahead in their careers.
- **Innovative Learning Approach:** ITProAcademy integrates theoretical concepts with practical demonstrations, ensuring learners gain both knowledge and applicable skills.
- **Custom Learning Tracks:** Whether you are a student looking to begin your IT journey or a professional seeking to upgrade skills, ITProAcademy offers customized learning tracks to meet individual needs.

**Our Commitment** at ITProAcademy.co.in, we believe in empowering learners to excel in the digital landscape. Our aim is to make advanced IT education accessible, affordable, and impactful for individuals and businesses alike.

Join us at ITProAcademy.com and take the next step towards mastering IT technologies and achieving your career goals!

## Disclaimer

This document, *Handbook for Scratch for Primary School Students*, is exclusively designed for **primary school students** as part of the *Programming Languages Made Fun and Easy for Kids & Students* initiative. The content is intended solely for **learning and practice purposes** and aims to make programming engaging and accessible for young learners.

All material presented in this handbook is of a general nature and has been referenced from multiple sources, including the **Internet** and **ChatGPT**. While every effort has been made to ensure the accuracy and relevance of the content, it should be used as a supplementary guide to enhance students' understanding of Scratch programming.

The authors and contributors to this handbook disclaim any responsibility for errors, omissions, or the outcomes of any actions taken based on the information provided. This material should be used under the guidance of educators or guardians to support the students' learning journey.

## Contact Us

**Website Link (ITProAcademy.co.in):** https://itproacademy.co.in/

**Contact Email Address:** support@itproacademy.co.in

**Services & Programs Link:** https://itproacademy.co.in/services-%26-programs

**Contact Us Link:** https://itproacademy.co.in/contact-us