urbancode>

Training & Solutions

SCRATCH COURSE

About Course

Bring your child's coding dreams to reality. With Urbancode's Scratch course, train your kid to be Harvard or Google certified. Learn coding through experienced instructors, making it fun and accessible.

Primary Aspect

- 24 hrs Instructor-Led Training
- 10+ Projects & Exercises
- 9 Assessments
- 1 Final Project
- 2 Months Duration

Course Overview:

What is scratch programming?

Scratch programming is a block-based programming language. It helps kids to understand the fundamentals of coding by doing various interactive tasks. Students can create their own stories, animations, games and more programs on this platform. It enables children to improve their critical thinking skills.

Who is this course for?

This course is made for kids aged 7 to 16 years. Students with no prior coding experience, as well as those who know the basics of block-based programming languages, may join this course.

What will you learn in this course?

You will learn the fundamentals of coding, such as sequence, logic, events, variables, functions, and so on.

- Students will learn to create digital storytelling, animation, game development, and much more.

Course Curriculum

Module 1: Introduction

- Introduction to Scratch
- Sprite, Backdrop, and Sounds
- Character animation
- Loops and Sequences

Module 2: Story

- Narrating stories
- Text-to-Speech extension
- Parallelism

Module 3: Art & Animation

- Pen tool
- Advanced animation
- Music extension
- Function (Abstraction)

Module 4: Game & Application Development

- Basic game development
- Advanced game development
- Video sensing
- Translate extension
- Function with parameter
- Operators
- List

Computational Concepts

Concept	Description
Sequence	Identifying series of steps for a task
Loops	Running the same sequence multiple times
Parallelism	Making things happen at the same time
Events	An action being a root cause for another action
Conditionals	Taking decisions based on conditions

Computational Practices

Practice	Description
Being iterative and incremental	Iterating on development, testing periodically, and continuing the development process incrementally.
Testing and debugging	Making sure that the code works; find and fix errors
Reusing and remixing	Amend the existing code/project
Abstracting and modularizing	Combining smaller objects to build a bigger project

Certification from

HARVARD UNIVERSITY

Google