



Creative Learning Philosophy

Scratch pioneered block-based programming, enabling young people to learn to code creatively and interactively. Creating Scratch projects fosters the **development of computational and creative thinking skills** critical for future success: learners identify problems, break them into smaller parts, debug them, and iterate on solutions.

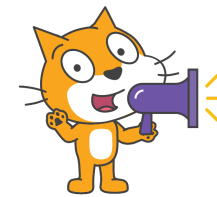
The Scratch Foundation creative learning philosophy is aligned with a **constructionist learning model**: We believe that learning is most impactful when making something, whether unplugged or using digital tools. We encourage focus on activities that center student-led inquiry, hands-on building, testing, iterating, collaboration, and presenting ideas through authentic dissemination to peers and/or the greater community.

At the Scratch Foundation, we design products and programs that support young people developing their:

- **Thinking skills**, which include tinkering; problem solving; hypothesizing, testing, and iterating; perspective-taking and reflecting; and transferring new skills to other aspects of learning and life.
- **Voice**, by which we mean the development of personal and external self-expression; and the ability to communicate with others appropriately in digital and public spaces.
- **Identity**, by which we mean the pursuit and development of personal interests; participation in affinity groups; a growing sense of self-awareness and acceptance; and an increase in self-efficacy over time.



THINKING SKILLS



VOICE



IDENTITY

What is Creative Learning?

You're engaged in creative learning when you develop ideas or things that are meaningful to you or to others you care about.

Creative Coding refers to creative learning experiences that involve coding and/or computing.

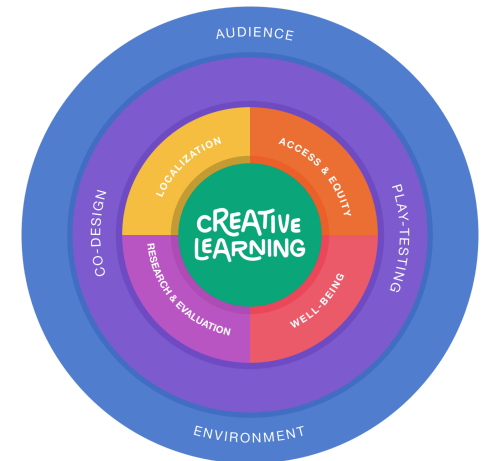
As facilitators, we want to support **playful learning and tinkering mindset values** so that participants can:

- Engage playfully in **projects** that are meaningful to them and elicit joy
- Collaborate with **peers** to experiment, share, and celebrate ideas
- Develop a mindset that is **comfortable with the discomfort of getting stuck**
- Develop a mindset that thinks critically about **strategies for getting unstuck**

Designing for Creative Learning

When we adapt/remix or develop a creative coding activity/lesson, we look to:

- Design for tinkering and learning through play
- Design spaces for learners to bring themselves and see themselves in the lesson
- Allow multiple pathways to foster a wide diversity of projects support different experience levels
- Create opportunities that naturally guide learners through the creative learning spiral



Recommended Reading:

- [“Computational Fluency \(excerpt from Lifelong Kindergarten\)”](#) by Michael Resnick [blog post]

Designing for Creative Learning: The 4Ps + 1

When designing activities for creative learning, we like to keep in mind the 4Ps + 1:

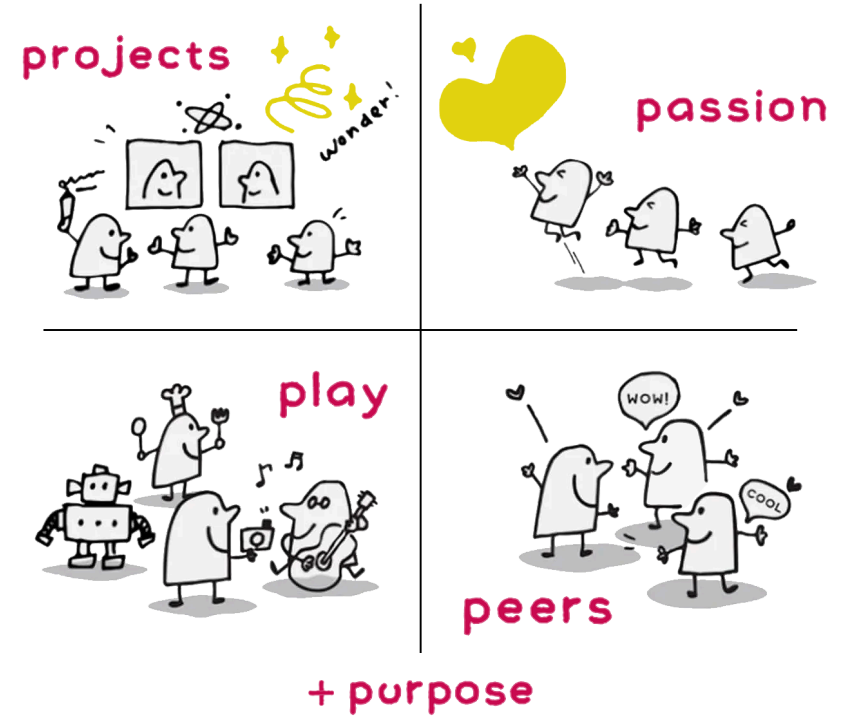
Projects includes computational tinkering and nurturing progressive improvement, through debugging and iterating.

Passion includes bringing yourself into Scratch and creating projects meaningful to you.

Peers includes social and emotional learning, remixing, collaborating, and sharing.

Play includes experimentation, taking risks, and open-ended exploration.

Purpose is an additional P that focuses on designing with intention and problem solving.



Recommended Reading:

- “[Coding at a Crossroads](#)” by Michael Resnick and Natalie Rusk [short PDF article, or [3:57 minute video](#)]
- “[Give P’s a Chance: Projects, Peers, Passion, and Play](#)” by Michael Resnick [short chapter excerpt]

Designing for Creative Learning: The House Model

Low Floor with Ramps

Make it easy for anyone to get started; ramps symbolize modifications for accessibility

Wide Walls with Frames of Interest

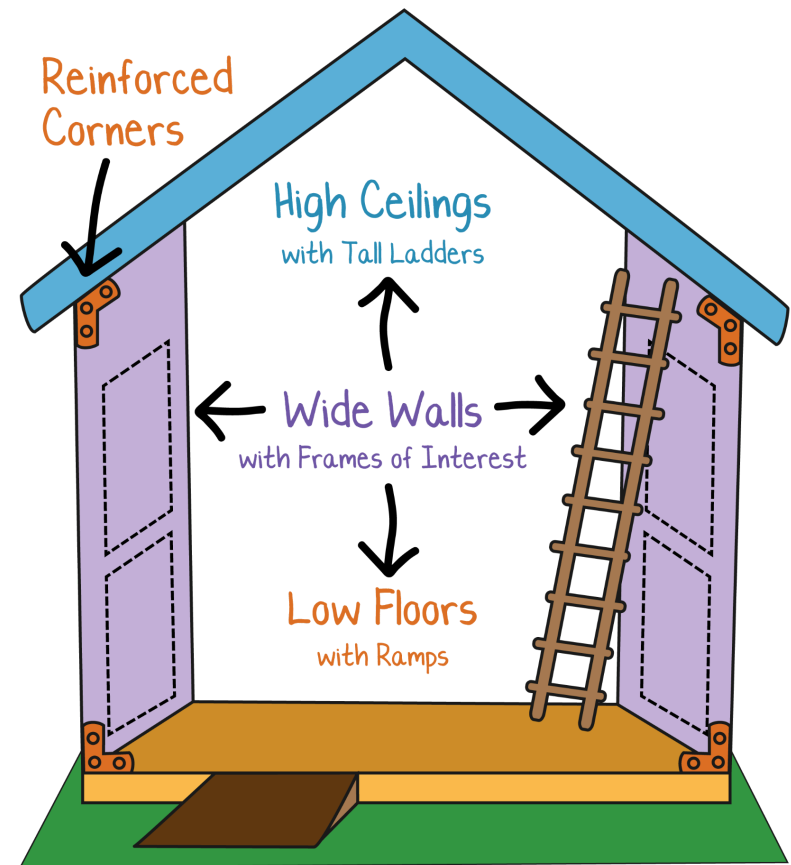
Appeal to a wide range of interests and passions and support students' specific interests

High Ceilings with Tall Ladders

Provide scaffolding for an extensible experience

Reinforced Corners

At intersections of floors, walls, and ceilings, consider what additional or unique supports or resources may be needed



Remix of image originally created by Alper, Hourcade, & Gilutz, 2012, which was adapted from Resnick & Silverman, 2005.

Recommended Reading:

- [“Designing for Wide Walls”](#) by Michael Resnick [blog post]
- [“Adding Reinforced Corners: Designing Interactive Technologies for Children with Disabilities”](#) by Meryl Alper, Juan Pablo Hourcade, and Shuli Gilutz [article]

Facilitating Creative Learning: Creative Learning Spiral

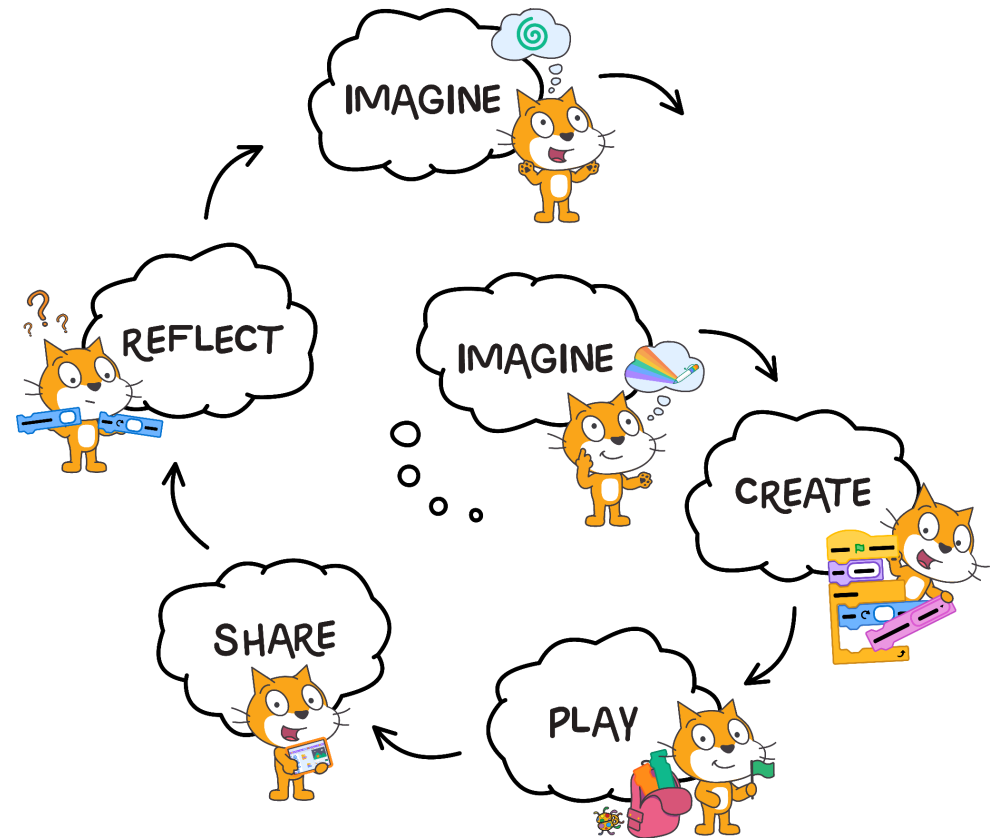
The “**Imagine, Create, Play, Share**” framework centers a collective process of learning.

It reminds us that we’re all in this **together**.

The role of the creative learning facilitator is to help **guide the creative thinking process**.

Keep this spiral in mind when guiding learners through the **creative coding process**.

In his blog post “[Ten Tips for Cultivating Creativity \(excerpt from Lifelong Kindergarten\)](#),” Mitch Resnick (Professor of Learning Research at the MIT Media Lab, director of Lifelong Kindergarten research group, and founder of the Scratch project) shares, “Although it’s certainly true that children are naturally curious and inquisitive, they need support to develop their creative capacities and reach their full creative potential....As children move through the [creative learning spiral], they get new ideas and continue to the next iteration of the spiral, with another cycle of imagining, creating, playing, sharing, and reflecting.”



Recommended Reading:

- “[Lifelong Kindergarten Cultivating Creativity through Projects, Passion, Peers, and Play](#)” by Mitch Resnick [book]

Facilitating Creative Learning: Tips

The following facilitation tips have been collected from partners at the [Lifelong Kindergarten Group at MIT](#), [Family Creative Learning](#), and [Harvard's Creative Coding Lab](#), in addition to our experience leading workshops around the world. Use these tips to help guide participants.

Build trust and relationships

Learning is a social process. Get to know your learners and help them get to know you. Learning new things requires learners to be open and vulnerable. Being around people that they know and trust can facilitate the learning process.

Surface their interests

It can sometimes take time for people to know what they want to do. Create an environment that is open to many interests. Ask questions like “What do you like to do?” Celebrate the variety of projects in progress!

Hold the tools as a last resort

It's tempting to grab the mouse, but try describing the steps rather than doing it for learners. If you have to navigate the tools, let them try for themselves after you show them and guide them along.

Be a connector

Connect learners with similar interests to each other and to relevant resources in the workshop.

Put yourself in their (unique) shoes

Empathize with learners to better understand their motivations and actions. Consider what creates a joyful, safe environment for the group.

Authentic enthusiasm

Sometimes learners, especially beginners, can feel unsure about their projects. Encouragement can help them feel more confident about next steps.

Ask questions vs giving answers

It may be tempting to give answers to questions right away, but if possible, ask questions instead so that learners can arrive at their own answers. See the next page for prompt ideas.

Encourage experimentation

Gently encourage participants to move out of their comfort zones to try new activities and concepts. Remixing other people's projects is a great way to explore new ideas!

Mistakes and failures are welcome

Get excited when something goes wrong! Rather than avoiding mistakes, encourage learners to be open to them. As you support them through their work, help them focus on the process.

Use technical words cautiously

Be aware of the words you use. Avoid technical jargon. If you have to use it, make explaining the jargon a learning opportunity. Focus energy, time, and attention on supporting the creative process, not solving technical problems.

Facilitating Creative Learning: Start with Exploration

The blog post “[Start with Exploration, Not Explanation](#)” by Natalie Rusk, a research scientist who was one of the creators of the Scratch programming language, offers this advice:

In some settings, Scratch is introduced with a long explanation of the interface and blocks, while students listen passively. This explanatory approach is based on the assumption that students need to be taught how to use Scratch before getting started.

In fact, our group at the MIT Media Lab designed and developed Scratch for learning through exploring, experimenting, and tinkering. Scratch coding blocks are designed like LEGO bricks....children learn by putting blocks together and taking them apart. They also pick up new ideas by seeing what others are making and collaborating to make something together.

Once students see how Scratch blocks snap together, they can start creating and notice what happens. Just like with physical building blocks, children learn to code with Scratch by experimenting and revising as they make projects. As the educational pioneer Seymour Papert emphasized: Using comes before understanding. People learn a tool or concept first by using it, and then their understanding develops over time through noticing and reflecting on their experience.

An interesting finding from our research is that young people who have developed broad creative, computational, and collaboration skills with Scratch usually first learned by “playing” or “messing around” with it, trying things out and seeing what worked. This playful approach helped them build their confidence in their ability to learn and problem solve.

So, how do we introduce Scratch so that beginners feel encouraged to dive in and start creating? When we offer Scratch workshops, we often show a couple simple but inspiring examples, then give a quick demo. We’ll briefly show how to snap together blocks to make something happen and then encourage getting started.

Recommended Reading:

- “[An Exploration in the Space of Mathematics Educations](#)” by Seymour Papert [paper, reposted]



Facilitating Creative Learning: Multiple Pathways

The blog post “[There’s More Than One Way to Code a Cat](#)” by Natalie Rusk, offers this advice:

“How do I make my character jump?” a student asks while coding a project in Scratch. Before responding, I find it’s helpful to ask what they have in mind, so they can think aloud about the process. Talking out their idea is often enough to help them to figure out what to do next. It’s interesting how often students will come up with a way to code that is different than one I might have suggested, but ends up working the way they want.

If I do suggest example code to look at (such as from another student’s project or from a Scratch coding card), I feel it’s important to acknowledge that there are multiple possible approaches, saying something like, “Here’s one way to do it, see if it does what you want...”

As coding is brought into more and more classrooms around the world, it’s sometimes introduced by asking students to solve a coding problem with a single right answer or with one predetermined path to a goal....Focusing on a single pathway not only limits the creative potential of coding, it also limits who becomes interested in coding and decides to learn more....I was fortunate to be introduced to a broad and inviting view of computer programming by Sherry Turkle and Seymour Papert. As they wrote in the early 1990s, “The computer is an expressive medium that different people can make their own in their own way....The diversity of approaches to programming suggests that equal access to even the most basic elements of computation requires accepting the validity of multiple ways of knowing and thinking, an epistemological pluralism.”

[We] refer to these ideas with the easy-to-recall phrase: “Many paths, many styles.”...We recognize that creating a learning environment that supports many paths and many styles can be challenging, particularly at the beginning and especially with the numerous constraints and demands often faced in schools. Yet each day we see how opening up possibilities provides motivation and meaning for more students to continue learning, creating, and sharing their ideas and interests with others.

Recommended Reading:

- “[Epistemological Pluralism and the Revaluation of the Concrete](#)” by Sherry Turkle and Seymour Papert [paper]

Facilitating Creative Learning: Debugging and Reflection

Getting stuck is a natural part of the creative learning process. Try these prompts to support participants in deepening their critical thinking, self efficacy, and modeling collaborative problem solving to help them “get unstuck” when faced with program bugs or challenging issues.*

“I love it! What is it?”

“Let’s test it out.”

“Can you explain what your program does?”

“Walk me through your code. What does it say?”

“What do you want your program to do?”

“Can you tell me more about that?”

“What new things did you try out?”

“Which category do you think would be helpful?”

“I don’t know, but let’s find out together.”

“I don’t know, but let’s look around the room.”

If using Scratch Coding cards, start with the front and ask:

“How might we do this?” before flipping it over.

“What are your next steps for this project?”

Debugging

- “Ask Three Before Me,” ask three peers before asking a facilitator
- See resources under [Debugging](#) [Scratch Foundation Learning Library]

Reflection Ideas

Cultivate a community of facilitators: help participants reflect on their own practice.

- [Scratch Design Journal](#) [Scratch Foundation Learning Library]: individual written or illustrative reflection on process
- **Turn and Talk:** peer to peer conversations
- **Gallery Walk:** [Reflection Sharing Sheets](#) [Scratch Foundation Learning Library], the group walks around the room and provides feedback for several peers

* Inspired by the [Creative Computing Lab “Getting Unstuck Curriculum.”](#)

Recommended Resources:

- [Getting Started with Scratch](#) [Scratch Foundation Learning Library]
- [Scratch Design Journal](#) [Scratch Foundation Learning Library]
- [Debugging](#) [Scratch Foundation Learning Library]
- [Reflection Sharing Sheets](#) [Scratch Foundation Learning Library]
- The [Indicators of Playful Learning](#) framework developed by Pedagogy of Play at Harvard Project Zero, a helpful resource for defining your own localized indicators or observable criteria aligned with your definition of creative learning

Additional Recommended Reading/Watching:

- [“Ten Sparks that Lit the Flame of Scratch”](#) by Michael Resnick [blog post]
- [“The Seeds That Seymour Sowed”](#) by Michael Resnick [blog post]
- [“Situating Constructionism”](#) by Seymour Papert and Idit Harel [paper]
- [“Mindstorms: Children, Computers, and Powerful Ideas”](#) by Seymour Papert [book]
- [“Scratch in Practice: How Do You Scaffold Peer Learning?”](#) by Natalie Rusk [blog post]
- [Learning Creative Learning Playlist](#) by the Lifelong Kindergarten Group at MIT Media Lab [a set of 6 short videos]
- [Minute with Mitch: Many Paths, Many Styles](#) [Scratch Foundation Learning Library]
- [Minute with Mitch: Curriculum Connections](#) [Scratch Foundation Learning Library]
- [Minute with Mitch: Peer Learning with Scratch](#) [Scratch Foundation Learning Library]
- [Minute with Mitch: Projects With A Purpose](#) [Scratch Foundation Learning Library]
- [“In the age of AI, we need a human-centered society more than ever”](#) by Michael Resnick [blog post]
- [About Universal Design for Learning \(UDL\)](#) by CAST [webpage]