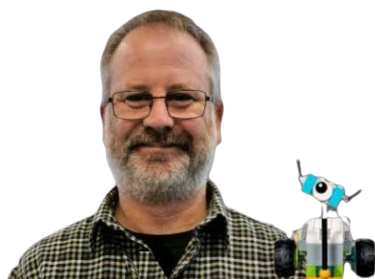


# EdTech Insight #8 Sept '24

## SCRATCHJR SPECIAL EDITION

### About Us

Support-IT has been providing professional IT support since early 2003. We pride ourselves on delivering IT Support that is fast, reliable and cost-effective for our clients. Michael the General Manager is a hands-on leader who has always ensured our educational clients are well looked after. We started our first school in 2004, [Bayswater School](#) and they are still with us today. I started working with Michael when I was the digital technology teacher at Coatesville School, in 2006. We often had long chats about how different schools used digital technology and how they could benefit from expert guidance. Long story short, in 2018 I joined Michael's team as the Educational Specialist and soon after we became accredited Ministry of Education PLD Providers. Since then we have helped many schools develop their digital technology and we continue to grow our schools across New Zealand. We have a team of excellent [facilitators](#) (see below) and despite the changing goalposts of this current government, we are still out there supporting schools. This month's EdTech Insight put a spotlight on one of my all-time favourite junior educational apps ScratchJr. We look back to 2014 when ScratchJr first started, and today, we see the latest developments in teaching and learning with ScratchJr.



# Celebrating 10 Years of ScratchJr



“With over 50 million users worldwide, ScratchJr is a free programming environment developed for children ages 5-7. With ScratchJr, children can learn the powerful ideas of computer science and literacy through programming and customising their own animations, stories, and games.” - DevTech Research Group

ScratchJr, a visual programming language, has revolutionised early childhood education by making coding accessible and engaging. The brainchild of a collaborative effort, the app has its roots in the DevTech Research Group at Boston College, led by the pioneering Professor Marina Umaschi Bers.

Recognising the potential of coding to foster creativity, problem-solving, and critical thinking, Bers and her team embarked on a journey to create a simplified version of the popular Scratch programming language. With funding from the National Science Foundation and additional support from a successful Kickstarter campaign, ScratchJr was born.



The initial release was launched in July 2014 for iPad; an Android version was released in March 2015 and a Chromebook app followed in March 2016. There is also a version called PBS Kids ScratchJr, which was released in partnership with PBS Kids in 2015. This version is not available in New Zealand unfortunately as it has lots of new fun characters. An app called Codeable Crafts was launched in 2015 by a Japanese company Benesse Corporation. It's available for iPad, Android Tablet and iMac. [Codeable Crafts Video](#) It uses the same coding blocks but has a different spin on costumes and backgrounds. It's fun but a little more clunky and glitchy compared to ScratchJr. An open source project developed a version for Windows and Mac desktop. It has been released on GitHub, follow this link to download: [Scratch JR for Desktop | Open Source Community Port](#). This version is handy for teachers modelling using their laptops. My older students like to use this version instead of the iPad version.

Professor Bers and the DevTech Research Group continue to be at the forefront of research into how ScratchJr impacts children's learning and development. Their work has shown that the app

can significantly enhance children's computational thinking, literacy, and mathematical skills while fostering creativity and collaboration.

By making coding accessible and enjoyable for young children, ScratchJr has opened up a world of possibilities for future generations of innovators and problem-solvers.

To help support teachers get started with ScratchJr I have created a website with many helpful resources - [Let's Create with ScratchJr](#).

ScratchJr is not only for 5-7 years olds. There are two ways I have used ScratchJr with upper primary-aged students. Firstly, ScratchJr can be used to create more sophisticated projects like recreating classic games such as Frogger see example: [2 Minutes of Tech - Project 3](#). This project is challenging for all ages and can be a good test of creativity and coding skills. There are many other games like Pacman or Angry Birds that could be recreated. The second way I have used ScratchJr with upper primary is by asking older students to be coding buddies with younger ones. ScratchJr buddy classes. This has been



very popular with older and younger learners. I like to model a lesson with the older students first and make sure they are familiar with ScratchJr. I have found that giving the older students the opportunity to teach gives them more confidence in their own ability to code and create projects.

The [ScratchJr starter cards](#) can be used as a starting point building a base knowledge. With the ideas from the starter cards teachers can develop projects that integrate with their current class curriculum. For example - [Space Rescue Project](#).

*Photo: Meghan our facilitator taking a class - Space Rescue Project at Edmonton School, Te Atatu South.*

There are a number of resources that can be purchased to support learning and teaching with

ScratchJr including [ScratchJr Coding Cards](#). Written by the ScratchJr co-creator, Prof. Marina Umaschi Bers, and Dr. Amanda Sullivan, the exercises in ScratchJr Coding Cards will encourage kids to develop coding skills as well as foundational concepts for literacy, math, planning, and problem-solving, all while having fun. The cards are created using the pedagogical approach developed by Prof. Bers to teach coding in a playful way to young children.

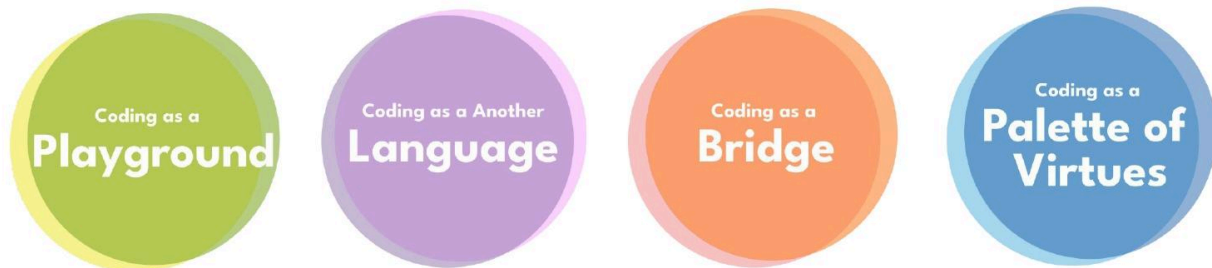


# Coding as Another Language

A new approach to teaching coding

Coding as Another Language is a fresh and inspiring way to look at teaching coding to young children. The way we view technology can make all the difference. For a long time I have considered coding through the lens of STEM or STEAM. Prof. Marina Umaschi Bers argues strongly that this approach is limiting. We need to think of teaching coding as much more. When I reflect on my junior play-based programme I can see the wisdom in approaching coding as another language or literacy. The philosophy of CAL is grounded in what DevTech calls the four pillars of pedagogy. This was explained by the DevTech team in a recent webinar celebrating ScratchJr's 10th Birthday.

View @ [ScratchJr Virtual Birthday Recording](#)



## 4 pillars of pedagogy





The DevTech group proposes we need to think about coding as a playground. A playground where children can make mistakes, invent, socialise, try things, imagination, and create. It is a place of autonomy where children can explore in persistent ways. Contrast this to a playpen. A playpen is safer but its potential for learning is limited. There is less social interaction and less opportunities for discovery. Most educational software is designed like a playpen. Marina Bers' says that learning through coding, in this case using ScratchJr, opens up many possibilities just like a physical playground does. If young children are learning digitally it should be like a playground not a playpen.



We used to see coding as STEM or STEAM. “We are against that”, says Bers. “Why limit the power of coding, which is a new literacy, to 4 or 5 letters?” Instead, if we position the teaching and learning of programming as a new way of thinking and expressing ourselves, we are in the domain of language. Mastering a symbolic system of representation with communicative and expressive functions, opens up many opportunities. Learning to code becomes a creative, expressive activity, to produce something meaningful and shareable, and not only a problem-solving skill set. The CAL pedagogy promotes the exploration of the similarities and differences between natural and artificial languages for the creation process, their syntax and grammar, and their potential to empower individuals. When coding is taught as a language, and not only as STEM, the human dimension comes into play. To learn more about this pedagogical approach, you can read: Bers, M. U. (2019). Coding as another language: a pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4), 499-528.

The Coding As Another Language for ScratchJr (CAL – ScratchJr) curriculum introduces powerful ideas from computer science, in conversation with literacy in a playful, structured, and developmentally appropriate way. The Coding as Another Language (CAL) approach, developed by Prof. Marina Umaschi Bers and members of her DevTech Research Group, understands the learning of computer science as a new literacy for the 21st century that supports young learners in developing new ways of thinking about themselves and the world. A CAL curriculum has been developed for New Entrants, Year 1, Year 2 and Year 3.

[ScratchJr Curricula – Coding as Another Language](#) and [KIBO Curricula – Coding as Another Language](#).

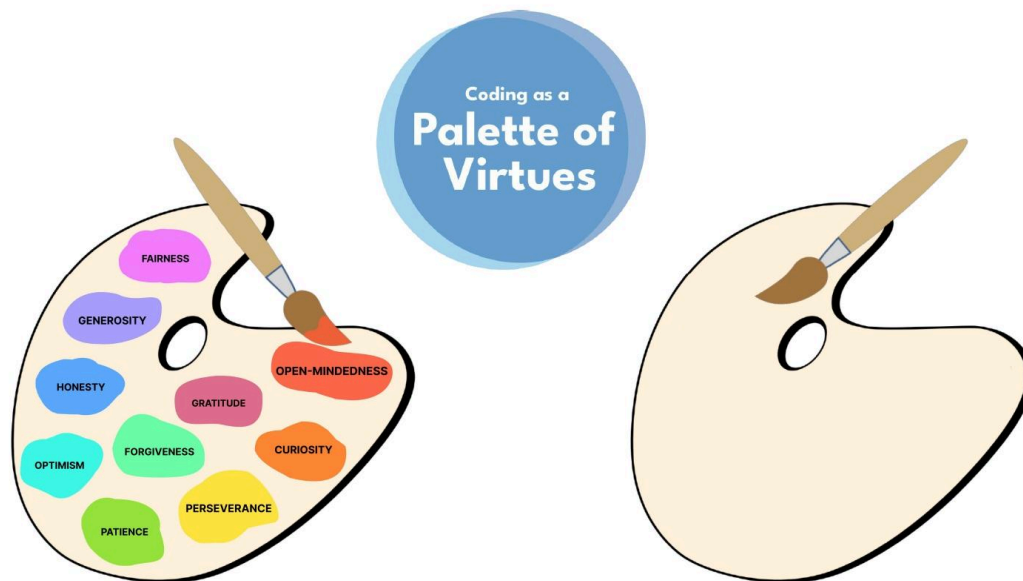
Example of Year 2 Curriculum: CAL (ScratchJr) for Year 2 has 24 lessons, designed for a total of 18 hours, but can be adapted to any learning setting. Students will learn computer science , develop problem solving and computational thinking in the context of creating their own projects. To strengthen the literacy connection, the curriculum is integrated with two books “Ada Byron Lovelace & the Thinking Machine” and “Where the Wild Things Are.”

Read more: [ScratchJr 1st – Coding as Another Language](#)



Coding as a creative activity where people can come together and connect. Coding as an activity that brings together different cultures, different age groups and helps people play, problem solve and create together. This is something we can do in our schools by using buddy classes to code together to work on shared projects. On a global scale I can recall being part of the [World Museum Project](#) that started in 2010 this project continued for a numbers of years and brought together many teachers and students from around the globe. This was using Scratch as a means of connecting through shared projects. The metaphor of coding as a bridge promotes dialogue and meaningful encounters with others and situates coding as socio emotional

learning, and not only a cognitive activity.



The metaphor of a palette of virtues recalls the painter's palette described in Marina Bers' [Beyond Coding: How to Teach Human Values through Programming](#).

"When teaching Computer Science or any subject, what values do you prioritize? Every human endeavor involves ethical considerations, choices, and consequences. The CAL pedagogy offers a unique opportunity to integrate values into the coding experience.

Imagine a painter's palette: filled with vibrant colors, ready to be mixed and matched to create stunning works of art. In a similar way, the coding playground is a dynamic space for exploring and expressing values.

Our palette of virtues starts with ten essential qualities, but it's adaptable. As students delve deeper into creative programming, they can discover and add new values to their toolkit. Coding can be a powerful pathway for character development, fostering socio-emotional intelligence and ethical awareness.

Read more: [Palette of Virtues in Practice – DevTech Research Group](#)

DevTech is currently reaching out to educators around the world to learn and take part in the development of CAL. I have started a free course with DevTech to learn more about CAL and hope to bring the four pillars into my current programme. There are opportunities to connect with DevTech and be part of this exciting research. [Get Involved – ScratchJr – Coding as Another Language](#) If you are interested in CAL I would love to hear from you.

**James Robson, Digital Facilitator and Lead Educational Specialist Support IT.**  
Contact me by email: [james@support-it-pld.co.nz](mailto:james@support-it-pld.co.nz)

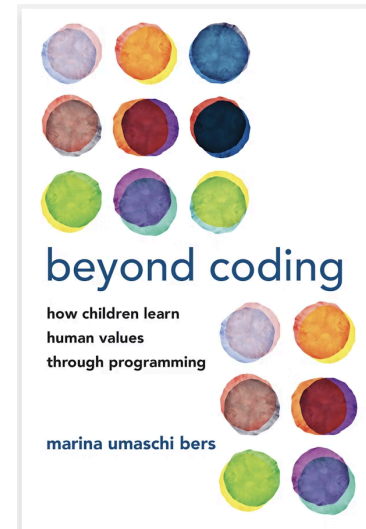
# Book Review #1

## **Beyond Coding** - How Children Learn Human Values through Programming.

Marina Umaschi Bers' *Beyond Coding* offers a groundbreaking perspective on the role of coding in education. Challenging the traditional view of coding as solely a technical skill, Bers argues that it is a powerful tool for fostering human values, critical thinking, and creativity in young learners.

### Key Themes:

- **Coding as a New Literacy:** Bers positions coding as a new form of literacy, similar to reading and writing. This perspective emphasizes its potential to empower children to express themselves, engage with the world, and develop essential skills.
- **Human Values and Character Development:** The book explores how coding can be used to cultivate empathy, respect, collaboration, and other important character traits. Bers demonstrates how coding activities can provide opportunities for moral education and social-emotional learning.
- **Computational Thinking:** Bers highlights the importance of computational thinking skills, such as problem-solving, critical thinking, and creativity, which are essential for success in the digital age.
- **Inclusive Education:** The book emphasizes the importance of creating inclusive coding environments that are accessible to all students, regardless of their background or abilities.



### Practical Implications:

- **Curriculum Integration:** Bers offers practical strategies for integrating coding into various subject areas, such as math, science, language arts, and social studies.
- **Classroom Activities:** The book provides concrete examples of coding activities that can be used to foster creativity, collaboration, and critical thinking.
- **Teacher Professional Development:** Bers offers guidance for educators who are new to coding, providing resources and support for professional development.

### Theoretical Underpinnings:

- **Constructivism:** Bers' approach aligns with constructivist learning theories, emphasizing active learning and student-centered approaches.



- **Sociocultural Theory:** The book highlights the role of social interaction and cultural context in learning, emphasizing the importance of creating collaborative and inclusive learning environments.
- **Developmental Psychology:** Bers considers the cognitive and social-emotional development of young children, providing age-appropriate coding activities.

In summary, *Beyond Coding* is a valuable resource for educators seeking to enrich their students' learning experiences. It offers a compelling vision of how coding can be used as a catalyst for personal and social growth.

To purchase [Beyond Coding](#)

Read a [Sample](#)

## Assessing Young Coders:

The ScratchJr Creative Coding Rubric



From the Scratch Conference July 2024:

### **Scratch Conference 2024 Workshop - Creative Coding Rubric for Early Childhood**

The ScratchJr Creative Coding Rubric, developed by the DevTech Research Group, offers educators a valuable tool for assessing young children's coding skills. This rubric goes beyond traditional coding assessments, focusing on the creative and design aspects of coding projects.

### Key Components of the Rubric

The rubric evaluates projects based on several key criteria:

- **Program Syntax, Length, and Amount:** Measures the complexity and quantity of programs created.
- **Block Variety:** Assesses the range of block types used.
- **Coordination:** Evaluate how well different blocks work together.
- **Repeat and Number Parameters:** Looks at the use of loops and numerical values.
- **Visual Customization:** Examines the creativity and customization of characters and backgrounds.
- **Narrative and Cohesion:** Assesses the story's clarity, continuity, and connection to the project.
- **Programmed Customization:** Evaluates the use of programming to customize characters and objects.

**Note:** The rubric has a wide set of criteria. It is not expected that all projects will score highly against all criteria. This will depend on the nature of the project. To gain an accurate picture of a student's ability the rubric would be best applied over a series of projects.

## The "Coding as a Playground" Approach

The rubric is grounded in the "Coding as a Playground" framework, which emphasizes exploration, risk-taking, and collaboration. This approach encourages children to experiment and learn at their own pace.

## Assessing Creative Expression and Technical Skills

The ScratchJr Creative Coding Rubric provides a holistic assessment of children's coding abilities. It not only evaluates their technical skills but also their creative expression and ability to tell stories through their projects. This approach recognizes that coding is a form of creative expression and that children can learn valuable skills while having fun.

## Using the Rubric Effectively

To use the rubric effectively, educators should:

- **Understand the rubric's criteria:** Familiarize themselves with the specific categories and subcategories of the rubric.
- **Provide clear instructions:** Explain the expectations and guidelines for the coding projects.
- **Offer guidance and support:** Provide support and feedback to children as they work on their projects.
- **Use the rubric iteratively:** Use the rubric to assess multiple projects over time to track children's progress.

## Benefits of the Rubric

The ScratchJr Creative Coding Rubric offers several benefits for educators:

- **Provides a holistic assessment:** Evaluates both technical skills and creative expression.
- **Supports a playful and exploratory approach:** Aligns with the "Coding as a Playground" framework.
- **Helps identify areas for improvement:** Provides insights into areas where children may need additional support.
- **Encourages creativity and innovation:** Promotes the development of creative coding skills.

By using the ScratchJr Creative Coding Rubric, educators can gain valuable insights into children's coding abilities and support their development as creative and confident young coders.

Below are the rubrics taken from the webinar. If you would like to no more about the ScratchJr Rubric Project or get involved go to: [ScratchJr Project Rubric – DevTech Research Group](#)

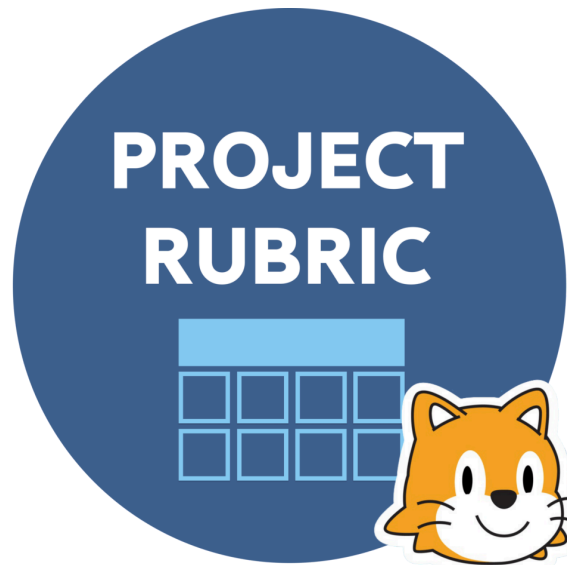
		0	1	2	3	4
Syntax Complexity	<b>Program Syntax, Length, and Amount</b>  <i>"Program" means any block attached to a start block</i>  <i>"Correctly" means the block made a change to the project</i>	(0 points) No <b>programs</b> in project	(1 point) One <b>program</b> in project	(2 points) 2-3 <b>programs</b> in project (can be across one or multiple characters)  <b>OR</b>  One <b>program</b> with <b>multiple start blocks</b> after the start block, used <b>correctly</b>	(3 points) 4-5 <b>programs</b> in project (can be across one or multiple characters)  <b>OR</b>  2-3 <b>programs</b> with <b>multiple start blocks</b> after the start block, used <b>correctly</b>	(4 points) 6 or more <b>programs</b> in project (can be across one or multiple characters)  <b>OR</b>  4 or more <b>programs</b> with <b>multiple blocks</b> after start block, used <b>correctly</b>
	<b>Variety of Blocks</b>  <i>Categories:</i> Yellow/Start Blue/Motion Purple/Looks Green/Sound Orange/Control Red/End	(0 points) 0-1 distinct blocks used	(1 point) 2 color categories of blocks used correctly  <b>OR</b>  Within 1 color category, at least 2 distinct blocks of that color are used  *plain red counts as one for Variety of Blocks	(2 points) 3 color categories used correctly  <b>OR</b>  Within 1 color category, at least 3 distinct blocks of that color are used	(3 points) 4 or more color categories used correctly  <b>OR</b>  Within 2 color categories, at least 2 distinct blocks of each color are used	(4 points) All color categories used correctly  <b>OR</b>  Within 3 or more color categories, at least 2 distinct blocks of each color are used

Coordination	<b>Coordination</b>	(0 points) No start blocks used correctly	(1 point) Only one character with a <b>program</b> (per page, if multiple pages)  <b>OR</b>  Green Flag is used in any program	(2 points) 2-3 characters with <b>programs</b> on the same page  <b>OR</b>  Correct usage of wait, speed up or slow down, or stop in at least one character  <b>OR</b>  Used go-to page with a green flag program on the next page	(3 points) At least one character has 2 or more <b>programs</b> in their programming area  <b>OR</b>  Correct usage of start on/bump in one character to coordinate with other characters in that page.	(4 points) At least one character has 2 or more <b>programs</b> in their programming area, utilizing at least one start block other than the Green Flag  <b>OR</b>  Correct usage of start on/send message in more than one character to coordinate the program.
	<b>Repeat and Number Parameters</b>	(0 points) No correct usage of repeat loop block or repeat forever block, and no parameters changed  <b>OR</b>  Used multiple identical blocks next to each other to repeat actions (e.g. 5 move forward blocks to move forward 5 times)	(1 point) One or more parameters are not default	(2 points) Used repeat forever  <b>OR</b>  Used repeat loop with one block inside correctly	(3 points) Used repeat loop block with more than one block inside correctly	(4 points) Used nested loop correctly (nested repeat means a repeat loop block within another repeat loop block or repeat loop block with repeat forever block used <b>correctly</b> *)  <b>OR</b>  Used at least a single block with negative/zero number parameter  <i>*To use a repeat forever and repeat loop together correctly, there must be at least one block outside of the repeat loop, or consecutive repeat loops within the repeat forever otherwise the repeat loop would not make a change to the project</i>

Project Cohesion	<b>Narrative and Cohesion</b>  <i>Theme = continuity across an element of the project. Elements of the project include programs, characters, pages, or visual customization</i>	(0 points) Project has no clear theme	(1 point) At least one element of the project conveys a theme. Some elements may not connect to the central theme.	(2 points) The elements of the project relate to a central theme, but there is no clear sequence of events.	(3 points) The entire project has a clear sequence of events, with at least one element connecting the events by a theme.	(4 points) The entire project has a clear sequence of events, with at least one element connecting the events by a theme  <b>AND</b>  All pages and character programs run when the project is played (meaning all pages are connected by go-to page, no go-to page blocks are used to skip over other characters programs etc)
	<b>Programmed Customization</b>  <i>Blocks do not have to be in a program to be counted for programmed customization, but they do have to make a change to the project when tapped in the programming area</i>	(0 points) No purple or green blocks used	(1 point) 1 purple block  <b>OR</b>  Pop block	(2 points) 2 types of purple blocks were used correctly in one or more characters  <b>OR</b>  Record block used. Could hear a sound or voice but it did not make a clear connection to the project  <b>OR</b>  Text entry that isn't default in say block (anything other than "hi"), but has no clear relation to project	(3 points) 3 types of purple blocks that were used correctly in one or more characters  <b>OR</b>  The recording was not accidental. Could hear and understand a narration or sound effect  <b>OR</b>  Non-default text entry in Say block and has a clear relation to the project	(4 points) 4 or more types of purple blocks were used correctly in more than one character  <b>OR</b>  Multiple recording blocks, with a clear relation to one another  <b>OR</b>  Non-default text entries in multiple Say blocks, with a clear relation to one another



Visual Customization	Visual Customization	(0 points) Nothing done to the aesthetic of the character or background (meaning a white blank background)	(1 point) Changed the background <b>OR</b> Added 1-2 characters (other than default Cat) with or without code	(2 points) Used the "Color Fill" function in the paint editor on a character or background <b>OR</b> Added 3 or more characters (other than default Cat) with or without code	(3 points) Used one of the following functions from the paint editor to customize a character or background: Free Draw Create Shapes Text tool Copy/Stamp Camera Reshape <b>OR</b> 4 or more characters and/or backgrounds customized with any tool including "paint fill" from the paint editor	(4 points) Used 2 or more of the following functions from the paint editor to customize a single character or a single background: Free Draw Create Shapes Text tool Copy/Stamp Camera Reshape <b>OR</b> Used one of the following functions from the paint editor to customize 4 or more characters and/or backgrounds: Free Draw Create Shapes Text tool Copy/Stamp Camera Reshape



# Book Review #2

## **Coding as a Playground**, Second Edition

Marina Umaschi Bers' [\*Coding as a Playground\*](#) is a groundbreaking resource that challenges traditional perceptions of computer science education and offers a compelling vision for incorporating coding into the early childhood curriculum.

The book convincingly argues that coding can be a powerful tool for fostering holistic child development. Bers masterfully demonstrates how coding can be integrated into various curricular areas, promoting literacy, math, science, engineering, and the arts through a project-based approach. Her emphasis on coding as a new literacy is particularly insightful, positioning it as a fundamental skill for the 21st century.



One of the book's strengths lies in its focus on the social-emotional benefits of coding. Bers writes about coding as a palette of virtues. The projects we choose and the people we work with reflect our character and values. She also highlights how coding can enhance problem-solving, collaboration, and perseverance, essential skills for young learners. By emphasizing a playful and exploratory approach, she provides educators with practical strategies to create engaging coding experiences that are both fun and educational.

The second edition's inclusion of new case studies, vignettes, and projects is invaluable for educators seeking concrete examples of coding in action. The book offers a wealth of practical guidance and inspiration, making it a valuable resource for teachers who may feel overwhelmed by the prospect of incorporating coding into their classrooms.

*Coding as a Playground* is a transformative resource that empowers educators to view coding as a catalyst for child development. By shifting the focus from producing programmers to cultivating creativity, critical thinking and problem-solving. Bers has created a blueprint for a more inclusive and engaging early childhood education.

*Coding as a Playground* is a must-read for educators seeking to harness the power of coding to create rich and meaningful learning experiences for young children. Bers' expertise and passion for the subject shine through, making this book an essential addition to any early childhood educator's professional library.

Currently, the DevTech Research Group is developing assessment strategies. The ScratchJr Creative Coding Rubric provides specific examples of how to measure coding-related skills and knowledge. Read about this - **Assessing Young Coders: The ScratchJr Creative Coding Rubric**. This work compliments and strengthens the pedagogical philosophy outlined in *Coding as a Playground*.

# Googley Tips for Googley Teachers



## New Google Classroom Groups

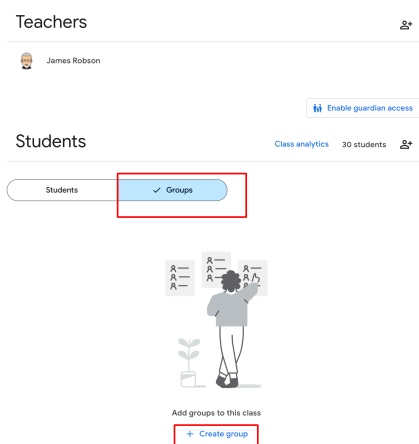
**New Feature:** Google Classroom now offers a **Groups** feature that allows teachers to create subgroups within their main class. This feature is very easy to use and super helpful for teachers.

### How to Use Groups:

1. **Access the Groups Feature:** Log into your Google Classroom course and navigate to the "People" tab.
2. **Create a Group:** Click the "Create Group" button at the bottom of the page.
3. **Name and Assign Students:** Give your group a name and assign students to it. You can create up to 50 groups with a maximum of 100 students per group. Students can be part of multiple groups.
4. **Use Groups for Assignments:** When creating a new assignment, go to the "Students" section on the right side. Instead of selecting individual students, choose the group you want to assign the assignment to.

### Important Notes:

- Students cannot see which group they are in. Only teachers and co-teachers can view group membership.
- To use the Groups feature, you need a premium Google Classroom license (Teaching and Learning or Education Plus). If you don't have a premium license, you can use a workaround involving manually selecting students for groups.



Source:  Managing small groups with Google Classroom (NEW for 2024) by John R. Sowash

## Next Month

- Hear from our new Schools
- Scratch - Learning with Peers, Projects, Play and Passion.
- Reporting back - Scratch Conference 2024



- Halloween Projects



- More Googley Tips for Googley Teachers