

REFERENCE GUIDE

FIRST

it is important to know WHY we use comics.

What makes programming difficult?

Programming is difficult because programming concepts, languages, and procedures are generally abstract:

- Programming concepts are often described using unfamiliar terms but how they relate to what we already know (e.g., real-life equivalents) is not always explained.
- Programming languages (i.e., code) require learning seemingly arbitrary rules (syntax & conventions) and semantics.
- Programming procedures requires students to understand how computer programs are executed and what happens (e.g., in memory) in each step, but they are often omitted or presented as abstractions (e.g., loop), obscuring the steps for novice learners before they can learn to step through the procedure on their own.

Use comics to complement the abstract nature of programming concepts, languages, and procedures.

CONCEPTS

Help students make sense of underlying ideas and properties.

CONSTRUCTS	DATA STRUCTURES	ALGORITHMS	PROBLEM SOLVING TECHNIQUES
variable	array	selection sort	greedy
boolean	linked list	insertion sort	divide & conquer
condition	queue	merge sort	recursion
counted loop	stack	bubble sort	
conditional loop	tree	linear search	
function	graph	binary search	
dictionary			

CONSTRUCTS

- Identify main idea(s)
- Find relevant metaphor, analogy, story
- Illustrate in comics



- Idea: variables store values
- Metaphor: piggy bank
- Analogy: piggy bank store money
- Story: Jane put \$1 to piggy bank

- Idea: either true or false
- Metaphor: birthday or not-birthday
- Analogy: if birthday, extra candy
- Story: if birthday, get 2 extra candies and share 2 candies with sister instead of 1 candy

DATA STRUCTURES

• Explain data structure (left)

• Use data structure in the story (below)

ALGORITHMS & PROBLEM SOLVING TECHNIQUES

- Show steps of the algorithm (e.g., left - merge sort)
- Show execution steps to make the process (e.g., right - recursion)
- Show code and comic side-by-side (e.g., right)

PROCEDURES

- Visualize procedures to help students learn to trace (visualize) them
- Control flow in computing procedures: sequence, branching, repetition

BRANCHING (e.g., if-else, case logic)

REPETITION (loop)

COUNTED Loop (e.g., for)

CONDITIONAL Loop (e.g., while)

MAPPING PATTERNS

Code (line) - Comic (panel)

- Code line refers to Line(s) of code.
- 1-to-1: 1 panel (or 1 row of panels) illustrates 1 line of code
 - 1-to-many: 1 panel (or 1 row of panels) summarize many lines of code
 - many-to-1: multiple panels are used to illustrate 1 line of code

Code (execution) - Comic (panel)

- Code execution refers to execution steps of a program.
- 1-to-1: 1 panel is allocated for 1 execution step
 - 1-to-many: a few (f) panels are used to summarize many (m) execution steps (where f < m)
 - many-to-1: multiple panels are used to illustrate 1 execution step

Code (structure) - Comic (panel layout)

- Code structure refers to the organization that the lines of code has according to their purposes. The action contents are those that visualize the execution steps and code expressions, whereas the scene and setting contents are those that provide context for the story.

USE CASES

We describe various use cases for inspiration.

Introduce Concept

INSTRUCTION

(1) show comic (e.g., left)
(2) point to the idea you want students to take away from the comic

e.g., "Look at how Spiderman's age, mood, hobbies change over time; variables are like that, too."

Introduce Code & Procedure

INSTRUCTION

1. Overlay
• show code in comic

2. Scaffold
• show comic
• abstract to English
• abstract to code

3. Juxtapose
• place code and comic side-by-side

Review Concept, Code, Procedure

ASSESSMENT

What does this represent?
A. Declaring variable
B. Assigning value to variable
C. Creating constant
D. None of the above

What is the index of the last element?
A. 0
B. 3
C. 4
D. 5
E. 6

What does this code print at 2nd frame?
let name = "PETER PARKER";
let mood = "Baby";
function setup() {
 age = 18;
 print(name);
}
function draw() {
 age = 20;
 print(age);
}

"Do they remember that the comic was used to explain variable assignment?"
"Do they remember that index begins at 0 and not 1?"
"can they predict what the value is in the 2nd iteration?"

This is an activity teachers can assign during the class. This activity proceeds with a teacher showing a comic to students in the class and then asking them to write code that corresponds to the comic. In the above example,

Write Code from Comics

ACTIVITY

THIS IS HOW IT ALL STARTED...
MOM, CAP'N CRUNCH IS THE BEST CEREAL! I CAN EAT IT EVERY DAY FOR 100 DAYS!!!!!!
DAY 1: YUM, YUM!
DAY 2: YUM, YUM!
DAY 50: YUM, YUM!
DAY 100: YUM, YUM!

#instructor's code
for (let day = 0; day < 100; day++) {
 eat(captain_crunch);
}

#student's submission
let interest;
for (let i = 0; i < 100; i++) {
 interest = 1;
}

This is an activity teachers can assign during the class. This activity proceeds with a teacher showing a comic to students in the class and then asking them to write code that corresponds to the comic. In the above example, the comic on the above left was shown to students. Students submitted their code using an online platform, Socrative, allowing the teacher to see the submissions in real-time and share with students what others submitted. On the right are examples of the teacher's code and a student's code. This activity is interesting in that different code can be submitted for the same comic. It provides an example for the teacher to highlight that there is no one right way to write code but that there are many different ways to write it. Because it is challenging to mark whether one is correct, this activity should just be an exercise, not quiz.

Create Comics from Concept

ACTIVITY

ConceptToon is a comic authoring tool for creating comics from concept. The tool guides users through the 3-stage design process, which consists of concept, story, and comic. After you teach programming concept(s), you can have students create comics about a given concept or concept of their choice.

Create Comics from Code

activity

OTHERS

- include comics in slides to complement your instructions
- introduce comic that represent computational concepts and procedures and then introduce corresponding code

RESOURCES & TOOLS

- Download comics at:
Coding Strips <https://codingstrip.github.io>
- Download cheat sheet at:
Reference Guide <https://codingstrip.github.io/use>
- Download design patterns at:
Design Pattern <https://codingstrip.github.io/design>
- Create comics at:
ConceptToon <https://codingstrip.github.io/create>
- Create comics at:
CodeToon <https://codetoon.uwaterloo.ca>

PUBLICATIONS

- Suh, Sangho, Martinet Lee, Gracie Xia, and Edith Law. "Coding strip: A pedagogical tool for teaching and learning programming concepts through comics." In 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1-10. IEEE, 2020.
- Suh, Sangho, Celine Latulipe, Ken Jen Lee, Bernadette Cheng, and Edith Law. "Using comics to introduce and reinforce programming concepts in cs1." In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, pp. 369-375. 2021.
- Suh, Sangho, Sydney Lamore, Edith Law, and Leah Zhang-Kennedy. "PrivacyToon: Concept-driven Storytelling with Creativity Support for Privacy Concepts." In Designing Interactive Systems Conference, 2022.
- Sangho Suh. 2022. Coding Strip: A Tool for Supporting Interplay within Abstraction Ladder for Computational Thinking. Ph.D. Dissertation, University of Waterloo. <https://hdl.handle.net/10012/18318>
- Sangho Suh, Jian Zhao, and Edith Law. CodeToon: Story Ideation, Auto Comic Generation, and Structure Mapping for Code-Driven Storytelling. In Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, 2022.

By Sangho Suh

Twitter: @sangho_suh

Email: sh31659@gmail.com