comiCS Cheat Sheet

- programming -

FIRST it is important to know WHY we use comics.

What makes programming difficult?

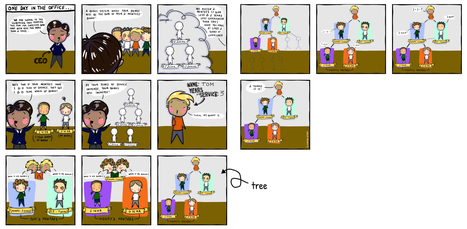
Programming is difficult because programming concepts, languages, and procedures are generally abstract:

- 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**.

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)

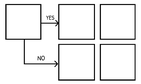
```
function how_many_in_front (num):
    if num == 0:
        return 0;
    else:
        return 1 + how_many_in_front (num - 1);

num(4);
```

PROCEDURES

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

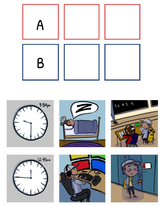
BRANCHING (e.g., if-else, case logic)

if else



if else if else

switch
case a:
case b:



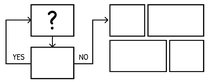
REPETITION (loop)

Counted Loop (e.g., for)

ITERATION <= N

ITERATION > N

Conditional Loop (e.g., while)



USE CASES

We descrobe various use cases for inspiration.

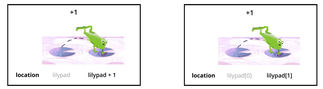
Introduce Concept INSTRUCTION

(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

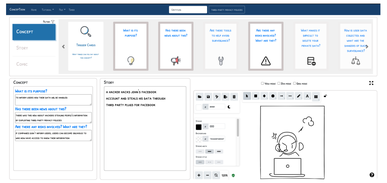


2. **Scaffold**
   - show comic
   - abstract to English
   - abstract to code

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

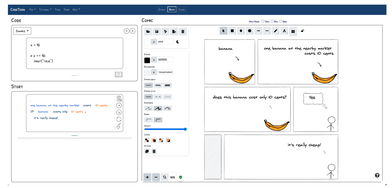
```
lilypad = ["purple", "purple", "purple", "rock", "red"]

for position in lilipad:
    if position == "rock":
        frog.confused()

    frog.jump();
```

Create Comics from Concept ACTIVITY



ConcepToon 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



CodeToon is another comic authoring tool that allows you to create comics. The difference between ConcepToon and CodeToon is that CodeToon can automatically generate comics from code, whereas you have to manually create comics with ConcepToon..

OTHERS

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

CONCEPTS

Help students make sense of underlying ideas and properties.

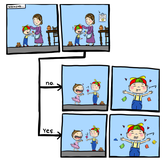
CONSTRUCTS	DATA STRUCTURES	ALGORITHMS	PROBLEM SOLVING TECHNIQUES
variable	array	selection sort	greedy
boolean	linked list	insertion sort	divide & conquer
condition	queue	merge sort	recursion
counted loop	stack	bubble sort	
conditional loop	tree	linear search	
function	graph	binary search	
	dictionary		

CONSTRUCTS

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

VARIABLE

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



BOOLEAN / CONDITION

- 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

LANGUAGES

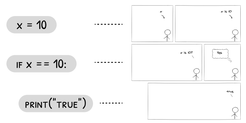
Help students make sense of code semantics, syntax, and conventions.

Semantics

You can turn each line of code into a row of comics and illustrate what the line of code means semantically

"line-by-line mapping"

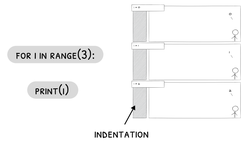
Example



Syntax

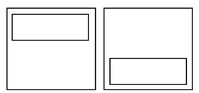
Describe syntax by showing it with comics (+ visual cues)

e.g., indentation



Conventions

Describe conventions using comics



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)

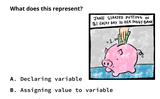
Code execution refers to execution step(s) of a program.

- **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)

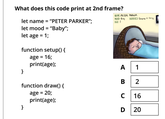
Code structure refers to the organization that the lines of code has according to their purposes. The action contents are those that visualize the execution steps and code expressions, whereas the scene and setting contents are those that provide context for the story.

Review Concept, Code, Procedure ASSESSMENT



"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...
MMM, CAPN CRUNCH IS THE BEST CEREAL! I'LL EAT IT EVERY DAY FOR 100 DAYS!
DAY 1
DAY 100
DAY 54
THAT'S ENOUGH!
THAT'S ENOUGH!
THAT'S ENOUGH!