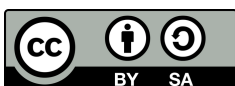# PIAF : développer la Pensée Informatique et Algorithmique dans l'enseignement Fondamental

Referential of competencies
Annex 2 : glossary

Glossary

Term	Definition	Examples
Action	something to be done, also called *instruction* in CS. NB: One can distinguish atomic actions (that will not be split into several steps) from more complex actions. What is considered "atomic" vs. "non atomic" depends on the language at hand (i.e. given set of actions, comparable to the existing vocabulary).	Atomic: - The sentence "Turn right" - The picture of a right arrow (—>) - "Put the hat in the box" - "Put your cap on" - 1 + 2 - etc. Non atomic: - Go to the mountain - Get ready for school - Get dressed

Algorithm	Precise description of the sequence of actions to achieve a goal.
	
	The solution path to a problem is developed first (algorithm), and only in a second step potentially realized as a computer program
		- A precise cooking recipe
		- Take a number
		  Add 3 to this number
		  Divide the result by 2
		  Print the new result

Combine	Given two sequences of actions that each partially achieve a goal, chain them together while applying required modifications	Make a game with two levels from two simple games (second game played after first one is won)
Compare	Apply a given criterion to two objects - From two objects, being able to tell whether they are	Given two animals, which one is the biggest? Which one has more legs? Which number is the smallest?

	identical with respect to this criterion. - When these objects can further be ordered, find the minimum (resp. maximum) considering this criterion.	Are both animals white?
Complete	Add one or more actions to an incomplete sequence to achieve a goal.	Add an action to complete a recipe. A character has already moved from point A to point B, add actions to go to point C.
Compose	See *Combine*	
Condition	A situation which holds or not, also called *If statement* in CS.	**If** the box is full **If** I don't touch the wall → The if-statement can be true or false
Constant	**An object whose value is known and does not change**	A person's name (of type string) A house's street number (of type integer)
Criterion	A property that will be used to compare objects / sequences of actions	Color of an animal, number of "move" actions performed when executing an algorithm/program
Decompose	- Split a goal into several subgoals - Split a complex action into an equivalent sequence of atomic actions	To find the biggest number out of 4 number, you first split the 4 numbers into 2 sets of 2, and compute the biggest number of each set.
Define	Choose appropriate actions and order them within a sequence so that the goal is achieved	How to add three numbers (first add the first two and then the third number)
Evaluate	Give a value to an object or to a sequence of actions according to some criterion	The number of lines of an algorithm, the result of some mathematical calculation
Extend	Given an algorithm that reaches a goal, add the necessary actions to achieve related but more complex goal	The algorithm makes the cat dance. Extend it to an algorithm that makes a cat and a turtle dance synchronously. There is already an algorithm to draw a

		triangle; extend it to draw a hexagon.
Formal	See *representation*	
Function	A named sequence of actions that possibly uses input values (also called *parameters* in CS) and computes and returns (i.e. stores) a result (number, image, word,…) which can thus be reused later on.	Compute the biggest of three integers.

Change the background color of an image (returns the new image) |
| Goal | Task to be done using an algorithm. | Create a memory game.
Have the cat move slowly towards the top of the mountain |
Identifier	A name (which can refer either to some object or some sequence of actions).	Let us call x the name of the tallest person in a group ("x" is an identifier referring to a variable value)
Improve	Modify the algorithm to maximize/minimize a given criterion	Given a program that needs 25 steps to take a robot from A to B, change it to have less than 22 steps
Informal	See *representation*	
Input	The values needed by a sequence of actions to achieve a goal	Order a meal at the restaurant (input = the menu)
Compute the average mark of a class at an exam (input = the individual marks)
Instruction	See *Action*	
Iteratively	By repeating actions.	Repeat n times "hello"

Achieve a subgoal, test the program, fix bugs, extend the program, test it again, fix remaining bugs, go on until no more bugs are found and the main goal is achieved |
| Modify | Adapt / change a sequence of actions to meet some requirements (new goal, new criterion, …) | Given an algorithm computing the maximum of three numbers, modify it to compute the minimum of the three numbers. |
| Name | A sequence of symbols, also called *identifier* in CS. | Toto
Foo
Bar |

	NB: a name can be used to refer to objects or sequences of actions.	Lulu54 Make_chocolate_cake
Object	Typically a noun phrase, which refers to a value which can either be *constant* or change over time (also called *variable* in CS)	- 12.5, - "Platypus" (here quotes are used to refer to the word) - Platypus (no quotes here mean it is a name / identifier)
Outcome	Can refer either to a value returned by a sequence of actions (also called *result* in CS), or to a modification of a setting once a sequence of actions has been performed (see *function* and *procedure*).	- New position of the cat - "Hello" printed on the screen - A dance performed by the cat on the screen - The next birthday in the class - The number of kids in the class that wear glasses
Parameter	An information that is required to achieve a given goal	Identifying the tallest person in a group requires to know the size of each of these persons (list of sizes = parameter).
Predict	To foresee the outcome of a sequence of actions (without seeing it running)	This sequence of actions draws a square: "Repeat four times: go forward, turn right 90°" What does the following sequence draw? "Repeat three times: go forward, turn right 120°
Procedure	A named sequence of actions that possibly uses input values (also called *parameters* in CS) but **does not return** (i.e.; does not store) any result. The outcome of the procedure is a modification of a setting (it cannot be reused by another sequence of actions).	Write the name of the cat on a paper. Erase the paper on the table. Fold a piece of paper in two.
Program	Translation of an algorithm into a programming language	Add 3 to 2 => 2 + 3 the robot moves if it hits an obstacle it goes back 5 steps and it turns right 45° else it continues and never stops =>

		to move.robot goforward 10 if captor = 1 [goback 5 right 45] else move.robot
Representation	Formal representation: - Expressed in a precisely and completely defined representation Informal representation: - Expressed in everyday language (intuitive, not precisely defined) → more flexible and ambiguous representation	Formal representation: - numbers - QR codes Informal representation: - vague description in e.g. French
Result	A value returned by a function.	Max(2,3) returns the result 3

Sequence of actions	Several actions executed in order.
	
	NB: algorithms, programs, functions and procedures are particular types of sequences of actions.
		def f(x):
		  y= x+3
		  y = y /2
		  return y
		
		To make a ham sandwich,
		- I take two slices of bread
		- I put butter on a slice,
		- I put ham on a slice
		- I add the salad
		- I put the second slice on the whole.

Test	Run a program/algorithm using several input values to check whether it reaches the expected goal. NB: in some methodological frameworks, tests are designed prior to algorithms.	In a maze game, move the cat into the wall to make sure that it cannot walk through it. Test the function max with input values (3,6), (1,12) and (5,5).
Translate	Convert a value or sequence of actions from one representation to another	Translate a pseudo-code into an actual scratch program Convert an arabic number into a roman equivalent one
Value	In computer science, a value can be a number, a word, a letter, an image,... It is an information whose interpretation depends on the	The value 6 can be an age, a size (in feet), the number of legs of an insect...

	context.	
Variable	Object whose value can change The variable has a name (also called: *identifier*)	x (in x = 3) E.g., a variable for the age of someone can be called "age", "ageInYears", or "banana". The value here would be a number (e.g., 24) and this number can change.