AEMASE
AFRICAN EUROPEAN MEDITERRANEAN ACADEMIES FOR SCIENCE EDUCATION

CODING AND COMPUTATIONAL THINKING IN THE SCHOOL CURRICULUM

Marco Giordano - marco.giordano@gmail.com
Conservatories of L'Aquila and Roma
Teacher trainer

WHAT ARE CODING AND COMPUTATIONAL THINKING ABOUT?

CODING IS...

… defining a set of instructions in a defined programming language, to make a machine perform a certain task.

Prejudices:

- coding is difficult
- coding is for *nerds* and insiders
- coding if you want to find a job as a pro developer
- coding is for boys

COMPUTATIONAL THINKING IS…

*… **NOT** thinking like a computer (a computer doesn't actually think…), but rather thinking like a **computer scientist** when he approaches problems or designs systems that must be implemented on an automatic machine.*

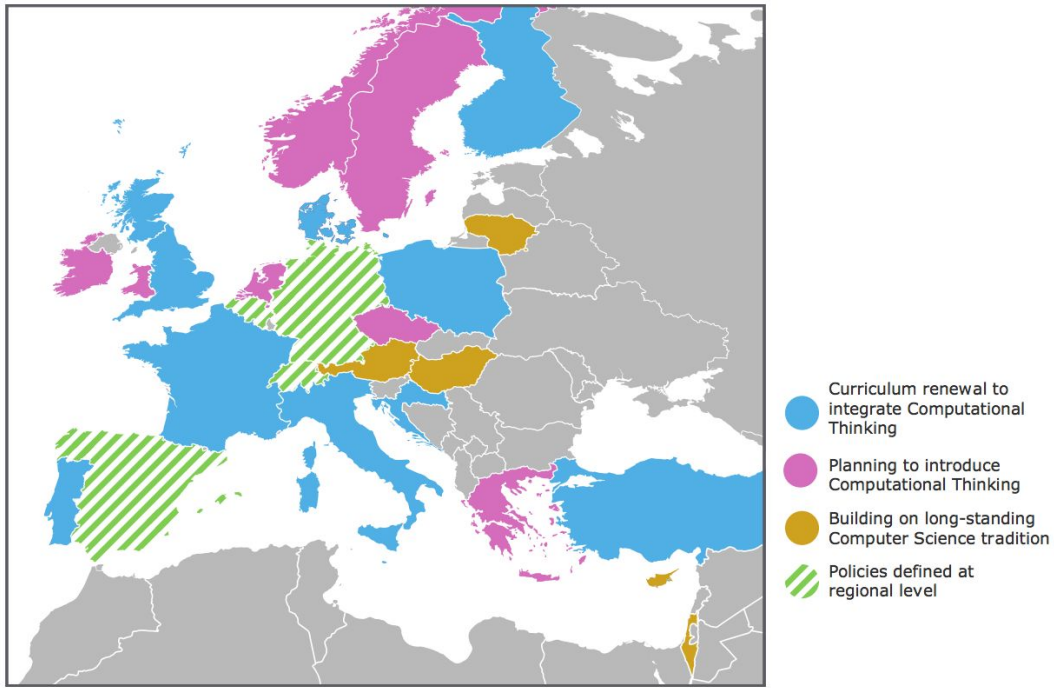
Concepts and skills:
data collection and analysis, data representation, problem decomposition, abstraction, pattern generalization and recognition, algorithms, automation, simulation, test, debug, parallelization.

WHY CODING AND COMPUTATIONAL THINKING...

...are so important for educational innovation?

- they are a fantastic gym for logical thinking
- they foster passion and strengthen motivation
- they are *learning by doing* in many fields of knowledge
- you can practice them with few infrastructures
- there is a plenty of initiatives and communities in the world
- they develop a conscious digital culture
- they are (also) a crucial investment for professional development
- …

Coding in EU educational systems



Source: 2016 - EU JRC Science for Policy report

OTHER INITIATIVES

Coderdojo: https://coderdojo.com/

in 2016: 65 countries WW, 1100 clubs, 48000 participants

EU Code Week: http://codeweek.eu/

in 2016: 50 EU countries, more than 980000 participants

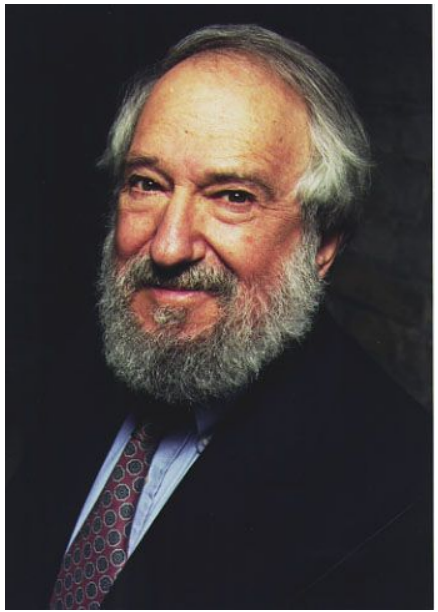
Africa Code Week: http://africacodeweek.org/

in 2016: 30 African countries, 430000 participants

Seymour papert (1928-2016)

mathematician, computer scientist, AI pioneer, pedagogist, Piaget's disciple, MIT Media Lab founder

theorist of **constructionism**



"thinking about thinking transforms children into epistemologists,

an experience that not many adults do."

Seymour Papert's 8 Big Ideas about Constructionism

Seymour Papert's 8 Big Ideas - 1

Learning by doing

We all learn better when learning is part of doing something we find really interesting. We learn best of all when we use what we learn to make something we really want.

Seymour Papert's 8 Big Ideas - 2

Technology as building material

If you can use technology to make things you can make a lot more interesting things. And you can learn a lot more by making them. This is especially true of digital technology: computers of all sorts including the computer-controlled Lego in our Lab.

Seymour Papert's 8 Big Ideas - 3

Hard fun

We learn best and we work best if we enjoy what we are doing. But fun and enjoying doesn't mean "easy." **The best fun is hard fun**. Our sports heroes work very hard at getting better at their sports. The most successful carpenter enjoys doing carpentry. The successful businessman enjoys working hard at making deals.

Seymour Papert's 8 Big Ideas - 4

Learning to learn

Many students get the idea that "the only way to learn is by being taught." This is what makes them fail in school and in life. Nobody can teach you everything you need to know. **You have to take charge of your own learning.**

Seymour Papert's 8 Big Ideas - 5

Taking time – the proper time for the job

Many students at school get used to being told every five minutes or every hour: do this, then do that, now do the next thing. If someone isn't telling them what to do they get bored. Life is not like that. **To do anything important you have to learn to manage time for yourself.** This is the hardest lesson for many of our students.

Seymour Papert's 8 Big Ideas - 6

You can't get it right without getting it wrong

Nothing important works the first time. The only way to get it right is to look carefully at what happened when it went wrong. To succeed you need the freedom to goof on the way

Seymour Papert's 8 Big Ideas - 7

Do onto ourselves what we do onto our students

We are learning all the time. We have a lot of experience of other similar projects but each one is different. We do not have a preconceived idea of exactly how this will work out. We enjoy what we are doing but we expect it to be hard. We expect to take the time we need to get this right. Every difficulty we run into is an opportunity to learn. **The best lesson we can give our students is to let them see us struggle to learn.**

Seymour Papert's 8 Big Ideas - 8

We are entering a digital world...

...where knowing about digital technology is **as important as reading and writing**. So learning about computers is essential for our students' futures BUT the most important purpose is using them NOW to learn about everything else.

CONCLUSIONS

- Computational Thinking is a key-competence in all the fields of learning and in STEM disciplines primarily. It can be practiced through coding, even “unplugged”.

- Constructionism is a pedagogical framework which is perfectly compatible with Inquiry Based Science Education

- To engage teachers and motivate them, I simply apply with them the same educational model I’m proposing: *learning by doing* holds not only between teachers and children...


when clicked
say Thank you for your kind attention!
forever
Enjoy Learning!!

BASIC REFERENCE

Papert S. (1980). Mindstorms: Children, Computers and powerful ideas. Basic Books. ISBN 978-0465046744

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice; EUR 28295 EN; doi:10.2791/792158

Wing J. (2006). Computational Thinking. Communications of the ACM, Vol. 49 n. 3.

Lodi M. (2013). Imparare il pensiero computazionale, imparare a programmare. Ph.D dissertation, Università di Bologna “Alma Mater Studiorum”.

Papert S. (1996). An exploration in the space of mathematics education. International Journal of Computers for Mathematical Learning, Vol. 1 n.1.