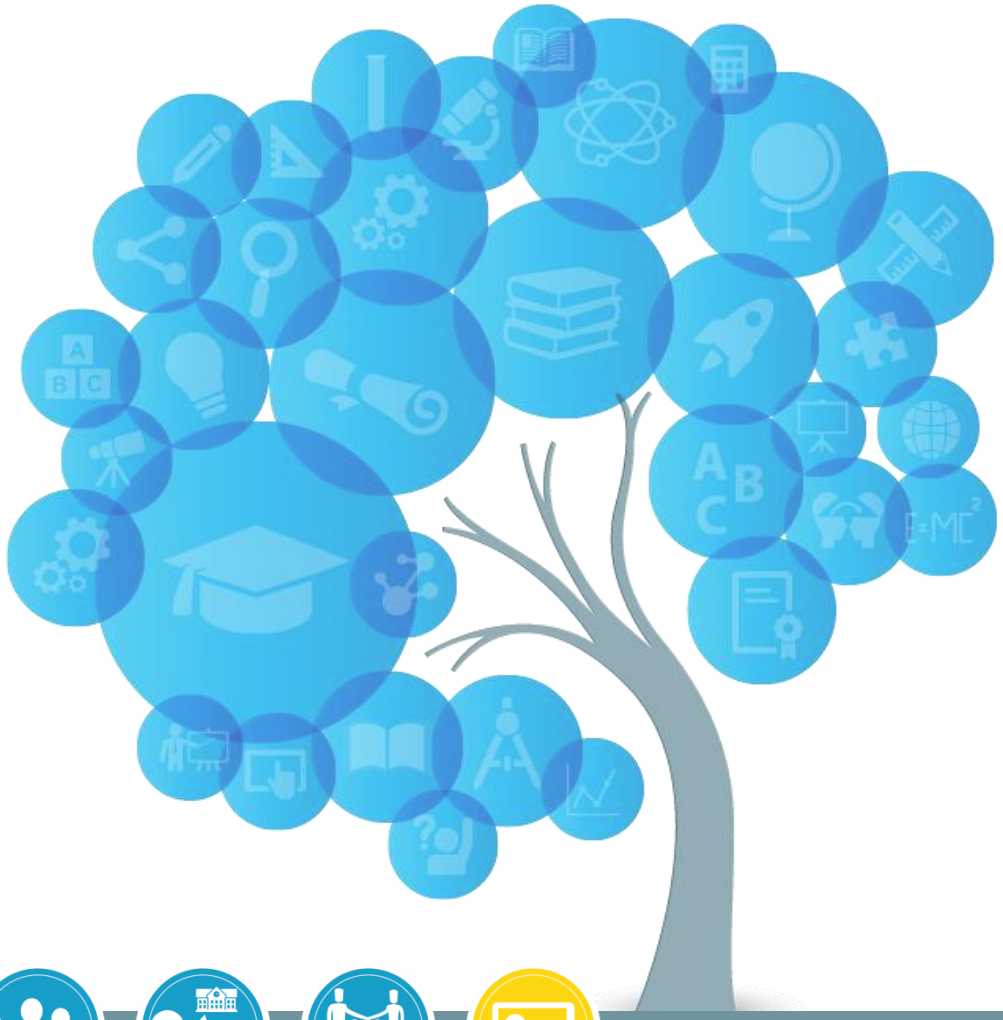
European Commission



Coding and computational thinking on the curriculum

Key messages of PLA#2
Helsinki, September 2016

Produced by the ET 2020 Working Group on Digital Skills and Competences

Education and Training

The second Peer Learning Activity (PLA) of the ET2020 Working Group on Digital Skills and Competences took place on 28-29 September 2016 in Helsinki.

The PLA focused on Coding and Computational Thinking (CT) in formal education; taking an in-depth look at Finland's experience and the new core curriculum where coding and robotics have been introduced for students age 6 to 16 years.

The two-day meeting included case studies from Finland, Poland and Estonia, expert and project presentations, as well as visits to three pilot schools to meet with school leaders, teachers and students.

Background[1]

Computational Thinking is understood as shorthand for "thinking like a computer scientist", i.e. using concepts of computer science to formulate and solve problems. In the past decade Computational Thinking has increasingly gained attention in the educational field for its potential to teach logical thinking, problem-solving and digital competence.

Despite the widespread interest in developing Computational Thinking at all levels of education and many public and private initiatives, the successful integration of CT in school curricula is still facing questions and challenges: Is CT a skill that benefits all living in an increasingly digital world? ? What features characterise CT instruction? How does the concept of CT and CT instruction relate to programming/coding, computer science and digital literacy? How should CT be assessed? How can teachers be prepared to successfully integrate CT into their teaching practice?

The discussions at this PLA touched on all these questions. Below we have captured some of the key findings and messages. These are not meant to represent the views of individual members or consensus of the group, but to reflect our overall discussions.

Summary

- Computational thinking (CT) can be loosely defined as the ability to use the concepts of computer science to formulate and solve problems. Definitions and terms vary between and within countries. Coding is generally understood as a

[1] A new study from the Joint Research Centre provided the context and background information for the meeting. The upcoming publication maps the field and state-of-play of CT in formal education in 13 countries; for details see: https://ec.europa.eu/jrc/en/computational-thinking

tool to teach CT, but CT entails a wider range of abilities (e.g. problem analysis, algorithmic thinking).

- Some countries have a long-standing tradition of CT on curriculum; many more are in the progress of integrating or planning to do so. Currently the integration is mostly on secondary level, but there is increasing trend to introduce it in primary education.
- Coding/computational thinking and digital competence in general is in some countries taught as a separate subject; an increasing number of countries integrate the themes with other subjects (e.g. mathematics, sciences …). Combining both seems to be optimal.
- Supporting peer-to-peer training of teachers can be key in spreading good practice; communities of practice can support sharing across schools.
- Coding and CT are themes where non-formal learning can play a key role and the boundaries between formal and non-formal education are blurred.
- New approaches to assessment are needed to assess CT learning, particularly if it is integrated across subjects.

Clear and consistent definitions are needed

Coding/programming certainly supports the teaching of Computational Thinking, but CT encompasses a wider range of abilities. Computational thinking usually involves the core concepts of abstraction, algorithm, automation, decomposition, debugging and generalisation. It can be understood as directly linked to and as a component of 'digital competence.'

Achieving a common definition of CT might not be possible or necessary at European level, but more clarity on terms and definitions would help discussions. At present some countries refer to CT; others, for example, prefer algorithmic thinking or favour a wide definition of programming.

Media (and industry) interest in coding can lead to misinformation on curriculum reform and educational priorities; and also influence policy priorities. Estonia, for example, was reported in international media to have made coding a compulsory element of primary education; in fact it is not compulsory but a large number of schools have indeed taken it up.

Coding and computational thinking are increasingly present on curricula across Europe

Young people need to understand our networked world and at the same time benefit from being able to use abstraction and decomposition when tackling complex tasks. Computational thinking can help students with both these aspects.

Computational thinking skills can be taught jointly with other competences such as communication, social competences and the ability to work in teams. Coding can make deeper computer science concepts concrete and can be combined with elements such as creativity, team work and problem solving. In this sense it can be a useful tool for the teaching of CT.

In a number of countries, including Finland, CT and coding are integrated across different subjects such as maths or music. This can motivate students and provide authentic learning experiences but can be a challenge for teachers.

A participatory approach including in-school and the wider community (parents, local industry) can help successful curriculum reform to include CT. Defining a vision and clear objectives for the integration of CT in compulsory education is crucial.

A key issue for education systems is to adapt to changing environments, workplaces and societies so that students develop skills that they need now and in their future lives.

The assessment and evaluation of CT can be a particular challenge but is vital to ensure effective integration into school practices, especially in the context of transversal skills. New tools and criteria to help teachers in assessing CT skills, in particular in a cross-curricular approach, need to be defined.

Training and support for teachers is vital; peer-to-peer training can be a solution

Teachers need a good understanding of what CT is and how to teach it. Introducing CT requires new training, possibly at large scale, as CT does so so far not often feature in teachers' initial training. Support services for teachers that provide concrete advice and examples can support teachers to use coding and computational thinking in class.

In delivering CT education, teachers' knowledge and starting point should be considered; e.g. the similar focus on logical steps between mathematics and CT can be exploited.

Teachers need examples and good practices relevant for their specific teaching context in order to confidently approach CT.[2] Teaching CT may require new pedagogical approaches that put students at the centre of the learning process.

[2] This was in particular a lesson of the TACCLE 3 project (http://www.taccle3.eu)

Ambitious changes to the Finnish curriculum on computational thinking are facilitated by strong professionalism of teachers. As a consequence of a participatory approach which encouraged bottom up input, many teachers felt their voices were heard in the curriculum reform; the new curriculum recognises and formalises practices that are already happening in many Finnish schools.

While the majority of teachers are keen to learn, the introduction of compulsory top-down training can cause resistance. The Finnish and Estonian examples show the benefits of peer-to-peer learning.[3]

Policy makers should support and encourage networks of teachers and peer-to-peer training. European platforms can support the sharing of good practices across borders. Easily accessible digital training offers can motivate teachers to upskill; this can take place with or without government support.[4]

The role of school leadership

School-leaders and "frontrunner" teachers can provide leadership, encouragement and guidance in technology-rich learning environments.

Education and training leaders should understand the potential and role of computational thinking and coding so that teaching staff are supported and encouraged to integrate these concepts in their teaching.

Leadership at schools is crucial when including CT in compulsory education.

Developing CT/coding on the curriculum is, according to one speaker at the event, like all areas of digital skills education, it is "'10% technology and 90% about the people".

Good quality learning materials for coding/CT are crucial

Good learning materials and tools and guides for delivery of CT are crucial. Naturally the teaching of CT must be adapted for young children; e.g. children learn through games and tasks that can be digital (e.g. visual programming), but could also be paper-based or physical ("unplugged" teaching).

Developing a comprehensive and forward-looking curriculum to engage students across compulsory education is a challenge, in particular as technologies continue to

[3] E.g the Innokas network in Finland support such peer-to-peer learning through an 'innovative school network', training courses and a coding and robotics roadshow.
[4] A MOOC for teachers was developed by volunteers Koodiaapinen. 2700 participants took part in the first year with a high completation rate of 30% 500 graduates.

change. Core questions include when and how to start coding/programming in the early years, and at what stage to switch from visual to more textual programming.

To keep children engaged, careful planning and choices of pedagogical approaches and quality learning materials that help enhancing students' understanding of core concepts related to CT sequentially over several school years are necessary; an approach that will need to go much beyond offering a few hours of coding.

Gender aspects need to be considered

Closing the gender gap in relation with technology and CT still remains crucial. Across the EU just 21% of computer science graduates are female; of the graduates only a minority stay in tech careers; this choice is influenced by early experiences such as in school, parental influence and a wider lack of female role models in this field. [5]

The choice of tools used in formal education (e.g. a robot or programming environments like Scratch) influences which students might be engaged and motivated and such preferences might also vary by gender, social background and age. It is important whether and how technology is contextualised and the use case personalised to individual or groups of learneres. Introducing some compulsory elements of coding/CT in formal education could be considered, as optional courses/modules are usually taken up by those students that are already interested, but there should remain a minimum of flexibility.

Non-formal and informal learning play a crucial role

A strong message to come out of the PLA is the importance of cooperation between non-formal and formal learning. Boundaries between formal and non-formal education are shifting. In Finland, non-formal teaching and learning of CT and coding often takes place on school premises outside school hours. Validation and recognition of this non-formal training can be an important motivator.

Grassroots, non-formal learning for students and for teacher training have changed perceptions, teaching practices and helped to integrate CT across subjects.

Such learning opportunities can include MOOCs and in-person courses on programming, and voluntary in-school or afternoon code clubs. Formal education should make use of these opportunities, e.g in Finland coding clubs are understood as a way of complementing formal education and aim to reflect the national curriculum.

[5] e.g. https://ec.europa.eu/research/swafs/pdf/pub_gender_equality/she_figures_2015-final.pdf, p. 5

Moreover, teachers can refer their particularly gifted or interested students to such coding clubs.

Teachers can also benefit from and participate in such non-formal activities, as for instance many coding clubs are run by teacher volunteers. Training materials for and dedicated training for teachers from such clubs can help teachers to integrate CT and coding in their daily practice. More cooperation and synergies between formal and non-formal education should be encouraged.

Open questions/future work

Some of the complex questions and misinformation on CT/coding are due to unclear terms and definitions.

The ongoing revision of the key competences framework by the European Commission could help to clarify terms. The role of CT and its relationship with the digital competence should be clarified. Coding can contribute to CT or wider digital competence education but the scope and whether an ability to code should be a learning aim is open to debate. Key competences are often used as a reference point for curriculum design in the Member States.

Funding programmes such as Erasmus+ programme could further support innovation, good practice and peer-to-peer learning of teachers in coding and CT. Funding should focus on forward-looking new approaches on teaching digital competences and on the teaching with digital technologies. Results should be made available in a clear and accessible form.

Exchange and training across Europe on the integration of CT and coding, both for initial teacher education and continuous professional development, could help to spread best practice and could go hand-in-hand with efforts to translate or adapt tested teaching materials. Further exchanges between policy makers can be extremely valuable, particularly on the area of assessment and successful models of teacher training.

Gender aspects of coding and CT, as well as wider equity and inclusion could be fields for further research, in particular considering developments such as 'Bring Your Own Device' and the growing use of online and smartphone apps. Anecdotal evidence points to the potential of coding/CT to engage less engaged low-achieving students and students with special needs in learning, as developing own projects using coding/CT allows students to work at their own pace on different outputs according to their ability. Sharing their own project can give them a great sense of achievement.