University of Cyprus

Training programme on "Computational thinking and coding skills"

Overall aims of the training programme: Create a training methodology and resources for youth workers to learn and subsequently teach computational thinking and basic coding skills, in a non-formal education setting.

Characteristics of the training programme:

Hands-on: the participants are actively engaged in making, doing and testing to learn the lesson

Working in Groups: the participants feel a stronger motivation to do well, to support their team and vice versa (team will support them).

Result-oriented: Having a concrete outcome provides a goal for students and a sense of accomplishment when the learning activity is complete

Fun: If the experience is also enjoyable and can even be "gamified", the learning experience is better

Social impact: makes learners feel that they can contribute to their community

General overview:

This methodology was designed in a modularised way, including 5 thematic areas (1 per training day) divided into 3-4 activities per thematic (day).

Thematic Areas:

Day 1: Algorithms and Problem-solving skills
Day 2: Prototyping, UIs, Design Principles
Day 3: Basic Coding Principles, Visual Coding (using MIT App Inventor)
Day 4: Introduction to Programming Languages – Web development (JS, HTML, CSS)
Day 5: Debugging, Testing, User Evaluation.

Activity Types used in the programme:

Brainstorming activities
Roleplay activities
Group work / Exploring in groups
Group interactions
Testing - Tryouts
Using handouts
Presentations from the participants
Demonstration by the facilitator
Debriefing questions
Hands-on activities

Different objectives are targeted through the various activities, all related to the main topic of the training.

Activity Objectives:

Ice breaking
Coding-skills
Computational thinking
Team building
Problem-solving skills
Critical thinking
UI design
Evaluation

Methodology

Day 1 Thematic "All is Algorithms" (Algorithms & Problem-Solving)

Activity 0. 'Who we are.' [Completed]

A. Using the custom-made online board in **padlet.com** (https://padlet.com/vanezievangelia/who-we-are-d4fbfq9ya6jv144l), all participants get to introduce themselves.
Information to be asked:
- Name, photo
- A few words about your educational background and current job
- Which emoji do you use the most?
- If you would be an electronic device, which one would you be and why?
- Would you prefer to be a chicken or a cow?

Type: Presentations from the participants
Objectives: Ice breaking. Everybody gets to know each other in a fun way.
Duration: 30 minutes

Step-by-step implementation:

1. **Introduction to the workshop:** The board will be shared some days before, with an email explaining to all what they need to do and that they need to prepare their Padlet post before joining the training. All participants need to join and prepare the introduction for themselves.
2. **Main part of the Activity:** At the projector, the whole Padlet board is shared in front of the room. Then one by one, each participant is called (based on the Padlet posts order), and they have 1-2 minutes to quickly present themselves (from their seating space) according to what they have written in their Padlet posts.
3. **Evaluation:** no evaluation for this activity.

Tips for the facilitator: You need to create an account at padlet.com to create the board. You can log in with Google, Apple, or Microsoft or sign up with an email. Select the free plan. Click on "Let's go". Click on "Recipes -> View all". Select "Education". Select "About us". In the pop-up window, click "Create". OR Go to the original board and select "Remake" to copy it to your own Padlet.

Options for adaptation:

- If participants are not in the labour market (e.g. teenagers), change the question about their current job. Make it about current hobbies, interests, sports, or desired studies/dream job.
- If time is limited, keep only 1-2 questions. In these, include one of the fun ones.

- If, for some reason, using Padlet is not possible, do only an oral presentation answering the same questions, where each learner stands in front and introduces themselves.

B. Statements game adapted to programming/coding experiences.

Type: Presentations from the participants, Group interaction
Objectives: Ice breaking. Learners and teachers can see the diversity in participants' knowledge, experience, and interests around the topics of the training programme (coding and computational thinking).
Duration: 15 minutes

Step-by-step implementation:

1. **Introduction to the workshop:** The activity is explained to the participants.
2. **Main part of the Activity:** All the participants should stand in a horizontal line. The facilitator reads a list of statements (e.g. “This is my first experience with coding”), link to list: folder “Thematic Area 1 Activity 0 (Statements.docx)”, and participants move forward by one step for every statement that is valid for them. After each statement, participants can elaborate more on why they took a step forward.
3. **Evaluation:** no evaluation for this activity.

Tips for the facilitator:

- The facilitator should take a look at the list of statements before the activity to be prepared.
- A set of cards (folder “Thematic Area 1 Activity 0 Cards” // Editable version: click here) will be available for the facilitator, one for each statement, to assist in discussing the topic or answering questions.

Options for adaptation:

- If time is limited, fewer statements should be read
- If there is only a small number of participants, then this can be more interesting with discussion after statement. On the contrary, if there are many participants, discussion should be avoided for time management purposes, and to keep responses quick.

Activity 1. 'This is a cooking Class', a.k.a. How algorithms and problem-solving work. [Completed]

Type: Demonstration by the facilitator.
Objectives: Understand what an algorithm is. Explain computational thinking skills (problem-solving: identifying the problem, decomposing the problem, designing the solution).
Duration: 45 minutes

Step-by-step implementation:

1. **Introduction to the workshop:** The Facilitator explains that this activity will explain the main essence of what algorithms are and how problem-solving works.
2. **Main part of the Activity: (a)** Explain how algorithms are step-by-step instructions to solve a problem (very simple explanation, the algorithm card from activity 0 can be used: click here). **(b)** Then, algorithms will be shown with the example of "making a ham & cheese sandwich", via a live demo. The facilitator says: "I've asked 2 different individuals to write down with numbered instructions the recipe for making a ham and cheese sandwich. Let's run all the different instructions and see if their algorithms work" – presenting instructions with errors, leading to wrong procedures and results (check tips for the facilitator for more instructions on this).
   (c) The session will conclude with a short presentation (folder "Thematic Area 1 Activity 1 2. Presentation (CodeCivics Computational Thinking.pdf)" / Editable version click here) on computational thinking methods for algorithm creation and problem-solving.
3. **Evaluation:** A Slido or Kahoot mini quiz, with participants scanning the code with their mobile phones and responding to a few multiple-choice questions (Each facilitator needs to set-up their own slido or kahoot. Click here for the questions). At the end, the winner of the quiz is presented automatically.

Tips for the facilitator:

- The original set of instructions for the recipe collected from 2 people are here with instructions on how to do the demo to facilitate showing how different algorithms work or don't work based on the original collected recipes: folder "Thematic Area 1 Activity 1 1. Recipes Recipes collected by UCY".
- Notes to help with the concluding presentation (specifically for slide 4) can be found here: folder "Thematic Area 1 Activity 1 2. Presentation (Notes for slide 4.docx)".

Options for adaptation: If you want to collect recipe instructions in your language, or from different people than the original, or even change the food, please feel free to do so. Make sure that the recipe leaves room for differences and faults in instructions to occur.

The template for asking people for recipes is in folder "Thematic Area 1 Activity 1 -> 1. Recipes (CODECIVICS - Recipe Template.pdf)" for printing directly as it is or here for editing. Please copy before editing if you will be changing the recipe.

Activity 2. 'Scavenger hunt: The Algorithm Quest' [Completed]

Participants in groups will follow a series of algorithmic instructions, solve logic puzzles, and decode challenges to find the treasure, putting algorithm design and problem-solving techniques into practice. Challenges are complex and require decomposition and design/follow algorithms to solve them.

Type: Hands-on activity, Group work
Objectives: Creating, debugging, and understanding algorithms. Team building activity. Critical thinking, teamwork, computational thinking, and multi-step problem-solving
Duration: 1 hour and 30 minutes
Step-by-step implementation:

1. **Introduction to the workshop:** Explain the rules and the whole concept and instructions. Participants are separated into four teams. Hand out the card with the first stop for each team (each team visits different stations in different order).
2. **Main part of the Activity:** Each team has to visit four stations and solve the challenges they will be given there. The first team to finish with all four stations wins. Complete detailed instructions and material can be found here: "Thematic Area 1 – Activity 2 – Scavenger Hunt: The Algorithmic Quest.docx"
3. **Evaluation:** A winning team is announced. Debriefing and reflection at the end of the activity (as described in the document).

Tips for the facilitator: This activity required a lot of preparation to set-up all stations and material. Additional help is needed to facilitate all stations.

Options for adaptation: Should work for fewer participants (e.g. forming only 1 team and in this case maybe set up a time limit to add a challenge as there will be no winning team), up to many participants (bigger or medium sized teams). Should work for less duration (e.g. 30 minutes), by selecting only a subset of the challenges (stations). Should work for limited space: e.g. a classroom, in which all stations are set or for larger spaces (set up the stations in different buildings).

Activity 3. 'A picture is worth a thousand words': Visualising Algorithms [Completed]

Participants should be by now already familiar with algorithms, and now they will learn how to visualise them by using Flowcharts. This activity is the first touch with programming principles like conditions and selection, iterations, sequential execution, etc.

Type: Group work, Hands-on activity, Using handouts
Objectives: Problem-solving, Computational thinking, Planning for complex problems/algorithms, Visually describe the steps of an algorithm

Duration: 1 hour and 30 minutes

Step-by-step implementation

1. **Introduction to the workshop:** As a first step, a demo will be done by the facilitator to explain flowcharts using a set of 9 numbered cards (folder "Thematic Area 1 → Activity 3 → 1. Cards" / Editable version click here). The set of cards can also be printed and distributed to the groups as handouts. Then, the facilitator will further demonstrate the usage of flowcharts via an example (folder "Thematic Area 1 → Activity 3 → 2. Example" / Editable version click here).
2. **Main part of the Activity:**
   (a) (45 minutes) The participants divided into 5 groups will be asked to design a flowchart for a given algorithm (list of algorithms: folder "Thematic Area 1 → Activity 3 → 3. Algorithms for teams / Solution editable versions click here") on a big piece of paper as a team. Flowchart posters will be exhibited in the room.
   (b) (45 minutes) Then, all participants will be asked to use a digital tool (draw.io) to individually create a variation of the flowchart diagram done in the group by adding an extra level of detail and optimisation (variation ideas here: (folder "Thematic Area 1 → Activity 3 → 4. Algorithms Variations").

3. Evaluation: The facilitator visits the posters' exhibition before the individual work and explains if there are any errors in the flowcharts (using the solutions provided along with the algorithms). Then, during the individual work, the facilitator goes around and checks the flowcharts and provides help. At the end, the facilitator collects the final flowcharts per team to check and give any comments, if necessary, at the beginning of the next day.

Tips for the facilitator:

- The facilitator is expected to study the cards and supporting material, to get to know and understand what will be demonstrated and be able to respond to learners' questions.
- Cards and examples are offered in both pdf and image formats, for the facilitator to select what is most suitable for them.
- No preparation of extra material is expected for the theoretical part (demo).
- For the practical part a pc/laptop is needed for each participant and a big piece of paper for each group with some markers. Also tape, to exhibit the flowchart posters on the walls.
- Suggested solutions for the team flowcharts can be found along with the algorithms on the 2nd page of the pdf or as separate image files.

Options for adaptation:

- In case there is not enough time, the activity can be adjusted to include only the group flowchart on paper and, if time permits, the group flowchart in the tool (excluding individual flowcharts).
- In case no laptops are available, both group and individual flowcharts can be done with pen and paper.
- In case of a limited number of devices available, participants can work in groups in the digital tool as well.
- The list of algorithms can be adapted accordingly if there is something more suitable for the specific participants' audience.

Day 2 Thematic "Prototyping is key" (Prototyping, UIs, Design Principles)

Activity 1: A pictionary game with low-fidelity prototypes. [Completed]

Type: Group interaction, Group work, Brainstorming activity, Hands-on activity
Objectives: Learning that each person can perceive designs differently. Develop low-fidelity prototypes before any presentations or explanations (flipped classroom approach). UI development.
Duration: 1 hour and 30 minutes
Step-by-step implementation:

1. **Introduction to the workshop:** The facilitator presents the learners with the rules of the game. Teams are created, and game cards (folder "Thematic Area 2 □ Activity 1 □ Game cards" // Editable version click here) are distributed.
2. **Main part of the Activity:** The game is played with cards describing different UIs (e.g. a registration form, an e-shop cart, or a landing page). Learners play in teams. In each round, the timer starts, and one member of the team makes a design. The team members try to guess what kind of website the drawing is presenting. After the time is out, the members of the team work together to fix any issues and prepare a nicer version of each drawing (prototype) before moving on to the next round as follows: first, the team discuss the recognised issues and brainstorms ideas to fix the design, then they take some decisions and proceed to make the final design.
3. **Evaluation:** At the end of the activity, each team submits a portfolio of before and after designs to the facilitator. The facilitator explains, "...what you have been doing today so far is called low-fidelity prototyping..." to conclude the activity.

Tips for the facilitator: All 15 cards need to be printed for all the teams to use for the game. Drawing markers and A4-sized paper will be needed.

Options for adaptation: If printing is not possible (even though recommended), cards can be sent to the learners in a digital format. This can be played with 1 team or more. If time is limited, permit fewer rounds (fewer designs).

Activity 2: 'You have a visitor': Mentors short presentations on UI design topics [Completed]

Type: Presentations by experts
Objectives: UI development
Duration: 1 hour and 30 minutes

Step-by-step implementation:

1. **Introduction to the workshop:** The facilitator explains how the session will develop and introduces each one of the mentors to the learners.
2. **Main part of the Activity:** 5 different mentors visit the audience of learners and deliver a quick but engaging presentation for 15 minutes each on an area related to designing UIs, as follows:
    a. Design using Basic design usability principles: the 10 heuristics of Nielsen (Presentation can be found in folder "Thematic Area 2 → Activity 2 → 1. Presentations (1. CodeCivics Nielsen Heuristics.pdf)" // Editable version click here)
    b. Design using basic colouring schemes (Presentation can be found in folder "Thematic Area 2 → Activity 2 → 1. Presentations (2. CodeCivics Colors.pdf)" // Editable version click here)
    c. Design for your audience (discuss about user personas creation) (Presentation can be found in folder "Thematic Area 2 → Activity 2 → 1. Presentations (3. CodeCivics User Persona.pdf)" // Editable version click here)
    d. Design for accessibility (Presentation can be found in folder "Thematic Area 2 → Activity 2 → 1. Presentations (4. CodeCivics Accessibility.pdf)" // Editable version click here)
    e. Design for responsiveness (Presentation can be found in folder "Thematic Area 2 → Activity 2 → 1. Presentations (5. CodeCivics Responsiveness.pdf)" // Editable version click here)

3. **Evaluation:** A Slido or Kahoot mini quiz, with participants scanning the code with their mobile phones and responding to a few multiple-choice questions. Each facilitator needs to set-up their own slido or kahoot. Questions in

folder “Thematic Area 2 □ Activity 2 (Evaluation Questions.docx)”. At the end, the winner of the quiz is presented automatically.

Tips for the facilitator: You need to plan early for this activity and invite some people to act in the role of experts. You can either find individuals that are indeed experts on each specific field and they can do their own presentation, or you can find people that are relevant to the general topic, and provide them with the presentations.

Options for adaptation: There are 5 topics and presentations offered. You can adjust based on the time and visiting guests you can find or the interests of your participants. You can also present some of the topics yourselves (the facilitator) if needed.

Activity 3: ‘Hi-Fi Art competition’ [Completed]

Type: Presentations by experts, Hands-on activity, Group Work
Objectives: UI development
Duration: 3 hours

Step-by-step implementation:

1. **Introduction to the workshop**: (5 minutes) The development of the session is explained.
2. **Main part of the Activity:** (a) (5 minutes) Prototyping will be explained in brief by the facilitator, mentioning low-fidelity that was included in activity 1 and High-fidelity prototypes that will be practiced in this activity (Presentation in folder: “Thematic Area 2 □ Activity 3 (CodeCivics Prototyping.pdf)” // Editable version click here). (b) (20 minutes) Proto.io (https://proto.io/) or Figma (https://www.figma.com/) is presented as the digital tool to be used. (c) (2 hours) Groups (same groups as in activity 1) will now design detailed high-fidelity prototypes in figma/proto.io for an app (ideas in folder: “Thematic Area 2 □ Activity 3 □ Hi-Fidelity Ideas” // Editable version click here) following all the principles taught in activity 2. No two teams should select the same idea.
3. **Evaluation:** (30 minutes) Groups present their prototypes and explain their design choices. Peers vote for the most creative or innovative prototype.

Tips for the facilitator: Study the slides for (a) beforehand to be ready to present them. Keep the same groups as in the first activity. These groups have already worked together on low-fidelity prototyping, so they can continue with low-fi.
For (b) the facilitator should study and be prepared to present the tool to be used (proto.io or figma) and show how it is used (only one of the two tools will be shown

and used - decision is up to the facilitator). Demo videos are provided for the preparation of the facilitator:

- Figma: Playlist Click here
- Proto.io: Video 1 Click here, Video 2 Click here

For more information, facilitators and learners can visit the official YouTube channels:

- Figma: Click here
- Proto.io: Click here

Another option is to invite an expert for this demo.

Options for adaptation:

- Another tool can be selected, based on the available expertise.
- Ideas for apps are provided, however if the teams prefer to design their own idea, or to adapt and customize the given ideas, this is ok.
- If time is limited, the screens to be designed can be limited, for example only design landing pages (explicit instructions should be given).

Day 3 Thematic "Coding 101" (Basic Coding Principles, Visual Coding)

Activity 1: 'Basic Ingredients' [Completed]

Type: Demonstration, presentation by the facilitator
Objectives: Coding skills
Duration: 1 hour and 30 minutes

Step-by-step implementation:

1. **Introduction to the workshop**: An explanation for this session is given to the learners.
2. **Main part of the Activity:** Teach programming principles independently of any language (concept introduction), including (1) basic programming structures: sequential execution, conditions, iterations; (2) operators: relational, boolean, logical; (3) data types, variables, constants, collections (array, lists sets); (4) functions. The facilitator gives a presentation with small interactive activities to be done by the learners. The material can be found in folder: "Thematic Area 3 □ Activity 1 (CodeCivics Coding 101.pdf)" // Editable version click here
   They already have the feeling of these, because of the flowcharts.
3. **Evaluation:** Correct execution of the activities.

Tips for the facilitator: Get familiar with the presentation and basic principles beforehand. Prepare for the activities included in the presentation: (a) for the functions activity, print the exercise sheets with the problems and space for solutions to hand out to the participants (sheets can be found in folder "Thematic Area 3 □ Activity 1 □ Functions Exercise Sheets") // Editable version click here; (b) for the data types activity the facilitator should organise a kahoot or slido game (find the question in folder "Thematic Area 3 □ Activity 1 □ Data Types Exercise

(Questions.docx)") and share the qr code with the learners to join. The activity is developed there.

Options for adaptation: If the time is limited you can exclude Logical Operations and Collections. Adjust the activities accordingly.

Activity 2: 'Can you code without code?' An intro to Visual coding [Completed]

Participants are now familiar with the basic ingredients of programming, and they are going to have the first touch using the MIT App Inventor, a visual programming environment for building functional mobile apps - suitable for any level of programming experience (even none). Everything is done with drag and drop of blocks. In this activity they have the opportunity to watch a live demo of MIT App Inventor and in the next activity to use it to build an app.

Type: Demo from the facilitator
Objective: Coding skills, computational thinking, problem solving skills
Duration: 1 hour and 30 minutes

Step-by-step implementation:

1. **Introduction to the workshop**: The facilitator explains how there are several apps that learners can use for 'no code' coding. Shows 2-3 examples live (cards with description, link, and resources for each in folder: "Thematic Area 3 □ Activity 2 □ 1. Apps Cards" // Editable version click here).
2. **Main part of the Activity:** The facilitator guides the learners on how to use the MIT App Inventor to develop mobile applications. The facilitator demonstrates the interface building and logic building panels via examples.
3. **Evaluation:** No evaluation is needed.

Tips for the facilitator:

- The facilitator should study the manual provided for using the MIT app inventor (folder: "Thematic Area 3 □ Activity 2 □ 2. MIT App Inventor (CodeCivics MIT App Inventor How To.docx"))
- Examples of MIT app inventor apps can be found in folder: "Thematic Area 3 □ Activity 2 □ 2. MIT App Inventor (CodeCivics MIT App Inventor Examples.pdf)" // Editable version click here
- External resources: https://www.youtube.com/watch?v=L-6Y4eEJ2Ho

Options for adaptation: If the facilitator prefers to demonstrate and use another tool, with the same basic principles (visual programming), this is ok. However, the same tool should be used in both activities 2 and 3.

Activity 3: 'Building a mobile app' [Completed]

After the live demo on how to use the MIT App,

Type: Hands-on activity, Group work, Handouts distribution
Objective: Coding skills, computational thinking, problem solving skills
Duration: 3 hours

Step-by-step implementation:

1. **Introduction to the workshop**: (30 minutes) Learners are divided into 5 teams. The activity plan is explained. One mentor (junior developer) is assigned to each team. The software engineering process to be followed is explained (Poster in folder: "Thematic Area 3 □ Activity 3 (CodeCivics Development Process.pdf)" // Editable version click here)
2. **Main part of the Activity:** (2 hours)The teams will be creating their own apps (to address community challenges??) using the MIT app inventor, working with their mentors. (a) Initially teams and mentors select a topic from the list in folder: "Thematic Area 3 □ Activity 3 □ Topics" // Editable version click here. The facilitator will distribute printed handouts for noting down the app specification - what will the app functionality be (template provided in folder: "Thematic Area 3 □ Activity 3 (CodeCivics App Specifications Template.docx)"), which the teams will complete with their mentors. The facilitator will be having small meet-ups with each team to discuss their ideas, review the requirements handout, and agree if they can develop in this limited time or not. This procedure simulates software engineering processes in real environments. The facilitator will help them adjust the requirements if needed. (b) Teams start developing their apps. They request for help whenever needed. (c) Teams test (intuitively) their apps and do any adjustments needed.
3. **Evaluation:** The mentors and the senior developers give qualitative comments to the learners for the whole duration. (30 minutes) All teams present to the learners' audience their apps and receive comments.

Tips for the facilitator:

- The manual guides or external resources for using the MIT app inventor from the previous activity can be used.
- Encourage teams to start small.
- Focus on usability and UI design over feature-completeness.
- Support creativity by letting teams personalize colors, icons, and layout.

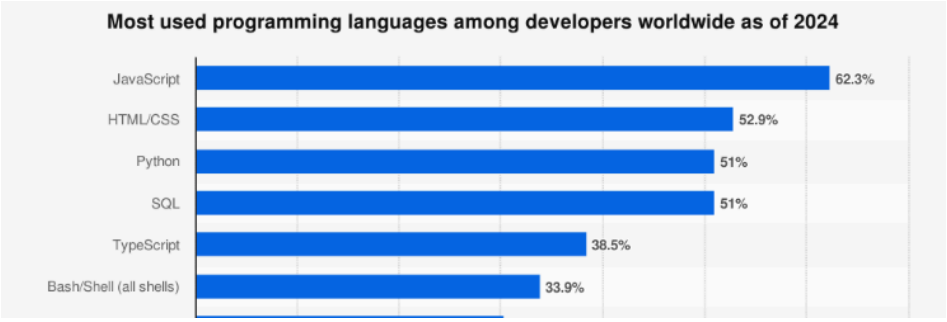
Options for adaptation: Another tool can be used instead, however the same tool should be explained in activity 2. If many mentors are not available, maybe each can supervise more than one team. If no mentors are available this can be skipped. If there are not so many learners, less teams can be created.

Day 4 Thematic Introduction to Programming Languages – Web development

Activity 1: 'Mirror, Mirror on the wall, what's the most popular language of them all?' [Completed]

Type: Presentation by the facilitator
Objectives: Coding skills
Duration: 1 hour and 30 minutes

JavaScript (JS) is the world's most popular programming language, followed by HTML(markup language) combined with CSS (styling sheets).



Principles taught on the previous day will take formal form in this activity with JS. HTML & CSS basics accompany this activity to give a complete web development experience.

Step-by-step implementation:

1. **Introduction to the workshop**: The activity plan is explained. Learners are asked to open the following website: https://codepen.io/pen, create a free account and log-in. The learners are provided with the material for this activity. Web development and the role-connection of the three languages is explained with one card (in folder: "Thematic Area 4 Activity 1 (CodeCivics Web Dev.pdf)" // Editable version click here)
2. **Main part of the Activity:**
    a. The facilitator demonstrates HTML, and CSS through an infographics sheet for each one. The infographics include the most basics of each language, and resources for more. HTML infographics in folder: "Thematic Area 4 Activity 1 HTML (CodeCivics HTML.pdf)" // Editable version click here, CSS infographics "Thematic Area 4 Activity 1 CSS (CodeCivics CSS.pdf)" // Editable version click here. After each one is presented, the facilitator shows the respective coding examples (HTML examples in folder: "Thematic Area 4 Activity 1 HTML Examples" and CSS examples in folder "Thematic Area 4 Activity 1 CSS Examples") and the learners follow the steps in codepen. Finally, the facilitator explains how they work together with JS.

b. The facilitator gives a brief presentation (in folder "Thematic Area 4 □ Activity 1 □ JS" // Editable version click here) on JS syntax, including the basic programming concepts taught on the previous day and some other important features for interactivity (e.g., selecting and modifying an element in the DOM). Small examples are shown so that the learners can try JS in codepen (JS examples in folder: "Thematic Area 4 □ Activity 1 □ JS □ Examples").

3. **Evaluation:** Evaluation will be done in the next activities.

Tips for the facilitator: Get familiar with the infographics, the JS pdf, the examples and codepen (or another editor) before developing this activity with learners. This activity requires deep engagement of both the learners and the facilitator.

Options for adaptation: Other editors can be used instead of codepen.

Activity 2: 'Challenge Yourself!' [Completed]
This follows as the practical part of the previous activity.

Type: Hands-on activity
Objectives: Coding skills
Duration: 1 hour and 30 minutes

Step-by-step implementation:

1. **Introduction to the workshop**: The activity plan is explained. In this activity, the 3 codepen boxes are replaced by 3 individual files - one for each box/language.
2. **Main part of the Activity:** Two coding exercises, including all three languages combined, will be distributed to the learners, which with the instructions of the facilitator, they will try to complete. The facilitator will explain the aim of each exercise (5 minutes at the beginning), learners will then work individually for a small amount of time (25 minutes), and then the facilitator will demonstrate the solution, discuss and respond to questions. (15 minutes).
3. **Evaluation:** Points for each exercise/challenge they solve. At the end, a winner is announced if possible.

Tips for the facilitator: You can find *exercise 1* instructions, files, and solution here in folder "Thematic Area 4 □ Activity 2 □ Exercise 1", and exercise 2 instructions, files, and solution here in folder "Thematic Area 4 □ Activity 2 □ Exercise 2".

Options for adaptation: Different exercises can be used if the facilitator prefers. The same basic knowledge should be included.

Activity 3: 'Weather forecast: Today it will be raining Websites' [Completed]
Type: Hands-on activity
Objectives: Coding skills
Duration: 3 hours

Step-by-step implementation:

1. **Introduction to the workshop**: A template is distributed, and learners walk through the template first with the facilitator. Facilitators also provide them with plenty of online resources to search for solutions e.g. w3 schools, and show them how to use these resources.
2. **Main part of the Activity:** Learners will work individually creating their own web apps/websites based on an assigned category (a list of categories will be provided). They will all work on the same basic template to prepare something completely different. Working on their own PCs. Facilitators go around and discuss with everyone, helping them debug, find solutions, etc.
3. **Evaluation:** Websites will be evaluated during the next day.

Tips for the facilitator: The facilitator should be someone familiar with HTML, CSS, and JS, in order to help learners with any issues. A second facilitator can help in this activity. Instructions, categories, and template can be found in folder "Thematic Area 4 Activity 3".

Options for adaptation: A different template can be distributed, as well as different categories and ideas. Learners can implement their own ideas as well. Learners can work in teams instead of individually (next day activities should be adapted accordingly if so).

Day 5 Thematic "Testing, User Evaluation"

Activity 1: 'I am ready to receive feedback!': Metrics for user evaluation [EV]

Type of activity: Presentation from the facilitator, hands-on activity
Objectives: UI Design, Evaluation
Duration: 45 minutes

Step-by-step implementation:

1. **Introduction to the workshop**: The facilitator explains what this session is about.
2. **Main part of the Activity:** The facilitator presents the different types of metrics there are to evaluate software (presented in folder "Thematic Area 5 Activity 1 (Metrics.docx)") and explains that to evaluate software we use typical and established evaluation questionnaires (three types of questionnaires are presented in folder "Thematic Area 5 Activity 1

(Established Evaluation Questionnaires.docx)"). Then, learners prepare their own questionnaires (using google forms or similar) for evaluating their websites,selecting the questions and methods they believe are more suitable for evaluating their websites. Selecting questions for all three presented methods is allowed, if they have a reason to do that. Questionnaires should not be too long, as users will not be able to respond timely. These questionnaires should be completed, as they will be used in the next activity.

3. **Evaluation:** In the next activity.

Tips for the facilitator: The facilitator should be familiar with google forms or similar, in case the learners require help. The facilitator should be familiar with the three types of questionnaires beforehand.
Options for adaptation: If time is limited, 1 established questionnaire should be selected, presented, and used by all.

Activity 2: 'Speed Evaluating' [CL]

Learners will evaluate the products of each other Part A. They will be wearing tags as "programmer" or "user". Link https://www.canva.com/design/DAGjfCeu7FI/ruBuIiNne_ZCKB0cE6uM5Q/edit?utm_content=DAGjfCeu7FI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Type of activity: role-play activity
Objectives: UI Design, Evaluation, to learn how evaluation works and feels from both sides (programmer and user)
Duration: 3 hours

Step-by-step implementation:

1. **Introduction to the workshop**: Stations (laptops) with the produced websites are set up and ready for testing (each learner can set up their laptop with their website and questionnaire). Numbers are given to the stations, and all learners get their schedules (the order in which they will visit the stations, which is their initial station etc). Rules are explained. All learners get to wear the tag "users".
2. **Main part of the Activity:** All learners will test & evaluate all products, like speed-dating (changing from table to table, viewing the product, responding to the questionnaire, moving to the next). The slots will be timed. This way, all programmers will gather a lot of results in their questionnaires (those created during activity 2).
3. **Evaluation:** Combined with next step

Tips for the facilitator: Tags need to be printed beforehand. The order of visits for each learner needs to be prepared and printed before the activity. Please set up everything for the activity to take place, as it will be time-consuming.

Options for adaptation: If there is no time to develop this activity, then learners just send their links and questionnaires to each other and ask for evaluation.

Activity 3: 'Do they like it? Do they not?' [Completed]

Learners will evaluate the products of each other Part B. They will be wearing tags as "programmer" or "user".

Type of activity: role-play activity, presentations by participants, debriefing questions
Objectives: UI Design, Evaluation, to learn how evaluation works and feels from both sides (programmer and user), to learn how to analyse and interpret results
Duration: 45 minutes

Step-by-step implementation:

4. **Introduction to the workshop**: Learners now wear their "Programmers" tag. An introduction on how to analyse their graphs is given (by using charts or tables, produced by google forms, or that can be produced in excel).
5. **Main part of the Activity:** Programmers analyse their results and then present to the audience their graphs. During the presentation, they respond to the following questions: "What did you learn?", "What would you do differently?", "What could have gone better?", "Did you achieve your objective?". Every time someone is presenting, they wear their "Programmers" tag, but everybody else is wearing their "Users" tags.
6. **Evaluation:** Peers ("users") provide feedback and vote for the best website.

Tips for the facilitator: The facilitator should do the administration of who presents, ask the questions, and help in facilitating the whole procedure.

Options for adaptation: To make it more fun and interactive, they could answer with emojis or receive answers and reactions from the audience as well.

Agenda for the LTT in Cyprus:
folder "Cyprus Training CodeCivicsTraining_Agenda.pdf"

References:

[1] Coding for Young People Erasmus+ Project
https://erasmus-plus.ec.europa.eu/projects/search/details/2015-1-ES02-KA205-006026

[2] CODING AND YOUTH: AN INNOVATIVE PROGRAMME IN THE DIGITAL ERA Erasmus+ Project
https://erasmus-plus.ec.europa.eu/projects/search/details/2016-2-CY02-KA205-000819

[3] FUTURE SKILLS Erasmus+ Project
https://erasmus-plus.ec.europa.eu/projects/search/details/2021-2-AT01-KA210-YOU-000048298