



CoderDojo



CoderDojo champion/mentor training: facilitators guide

Welcome to CoderDojo

Welcome to this facilitator's guide! Firstly, I'd like to take a moment to thank you for your part in growing the CoderDojo movement. CoderDojo is built on the work of volunteers. Without your continued dedication and passion, we would not be able to grow and sustain this fantastic movement that brings joy to thousands of young people all across the world.

In this guide we aim to provide you with resources, advice, tips, and much more, so that you can introduce CoderDojo to more people in your area. You will be able to teach the necessary skills to create the next wave of champions and mentors, who can help young people in your area to push the boundaries of technology.

This training is designed to give new volunteers the confidence and skills to get involved in CoderDojo. Upon completion of the training session, your attendees will have gained confidence in basic levels of Scratch, HTML/CSS, and working with young people. After the training they can learn more in their own time or with the young people in their Dojo.

We'd love to hear from about how the training course was received. Please get in touch with us to share any feedback via info@coderdojo.org.

Again, I'd like to thank you for taking the time to bring these skills to those in your local area and wish you the best of luck with your training sessions.

**Giustina Mizzoni,
Executive Director, The CoderDojo Foundation.**



**CoderDojo
champion/mentor
training: facilitator's
guide**

Introduction

The purpose of this guide is to ensure that you feel confident to facilitate your own CoderDojo champion or mentor training session.

CoderDojo depends on volunteers. Sometimes getting involved can be intimidating, particularly for people who have limited or no experience of coding or of working with young people. This training course aims to increase the confidence of new and potential volunteers for your Dojo.

These techniques, exercises, and activities are tried and tested. We've collected feedback from participants from a number of different backgrounds and levels of skill, and integrated their suggestions to bring you the best possible training materials.

This guide is intended for current community members who would like to introduce CoderDojo to others who may be interested in helping young people in their communities have fun with

technology. The training is particularly aimed at the parents/guardians of young people attending your Dojo who may already be accompanying their children to sessions. We encourage you to invite parents, teachers, tech workers, students, and any other members of your community who might like to get involved. A full training session will take approximately 5-6 hours to complete, but we encourage you to deliver it in multiple parts if that's easier.

This guide should be used in conjunction with our mentor/champion training slide deck, which can be located at dojo.soy/TrainingSlideDeck. Feel free to make a copy of these slides and make amendments or additions so that your training is tailored for your target audience. If you would like to leave any feedback to help us improve this guide in the future, please do so at dojo.soy/TrainingGuideFeedback.

Let's jump straight in!



Contents

Welcome to CoderDojo	2
Introduction	4
What's required to run a training session?	8
Mentor training pre-survey	9
Welcome	10
Course aims and objectives	11
Icebreaker	12
Concerns and expectations	14
Why volunteer? (video)	16
The CoderDojo movement	17
What is CoderDojo?	18
History of CoderDojo	18
CoderDojo philosophy	19
Unplugged activities	21
What is coding?	24
Where do you find computer code?	26
Coding and young people	30
CoderDojo lingo	31
Scenarios	34



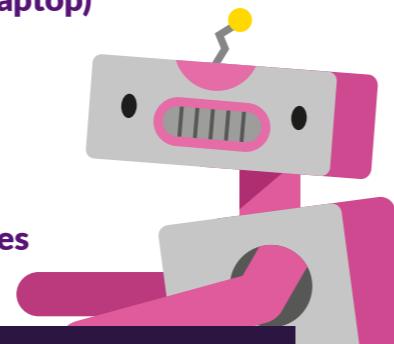
Scratch	35
Optional hardware activity	37
Making websites	39
Educational content	42
The CoderDojo community	43
Starting a Dojo	46
Feedback	50



What's required to run a training session?

Before you run your training session, it's best to check that your venue is equipped with the basics needed to run a successful session. These include:

- **A room that fits the number of attendees you expect**
- **Tables and chairs**
- **Projector/large screen (and a cable to connect your type of laptop)**
- **Speakers (ideally linked to the projector through HDMI)**
- **Whiteboard or flipchart**
- **Post-it notes and pens or felt-tip markers**
- **WiFi**
- **Laptop or desktop computers, one per group of 2–3 attendees**



Depending on your plans, they might also include:

- **Hardware station (Raspberry Pi Sense HAT or micro:bit)**
- **Tea/coffee**
- **Lunch**

Before you begin

Before the training, update the following slides with relevant information:

- **Slide 1:** Pre-training survey (add a link to your own online survey, if you have one; alternatively, delete the link and use the pen-and-paper survey overleaf)
- **Slide 3:** Facilitator's introduction
- **Slide 4:** Agenda
- **Slide 37:** Our global community (updated stats can be found at coderdojo.com)
- **Slide 46:** Post-training survey (add a link to your own survey, or delete the link and use the pen-and-paper survey on page 51 of this guide)



CoderDojo mentor pre-training survey

1

Please rate your knowledge of coding.

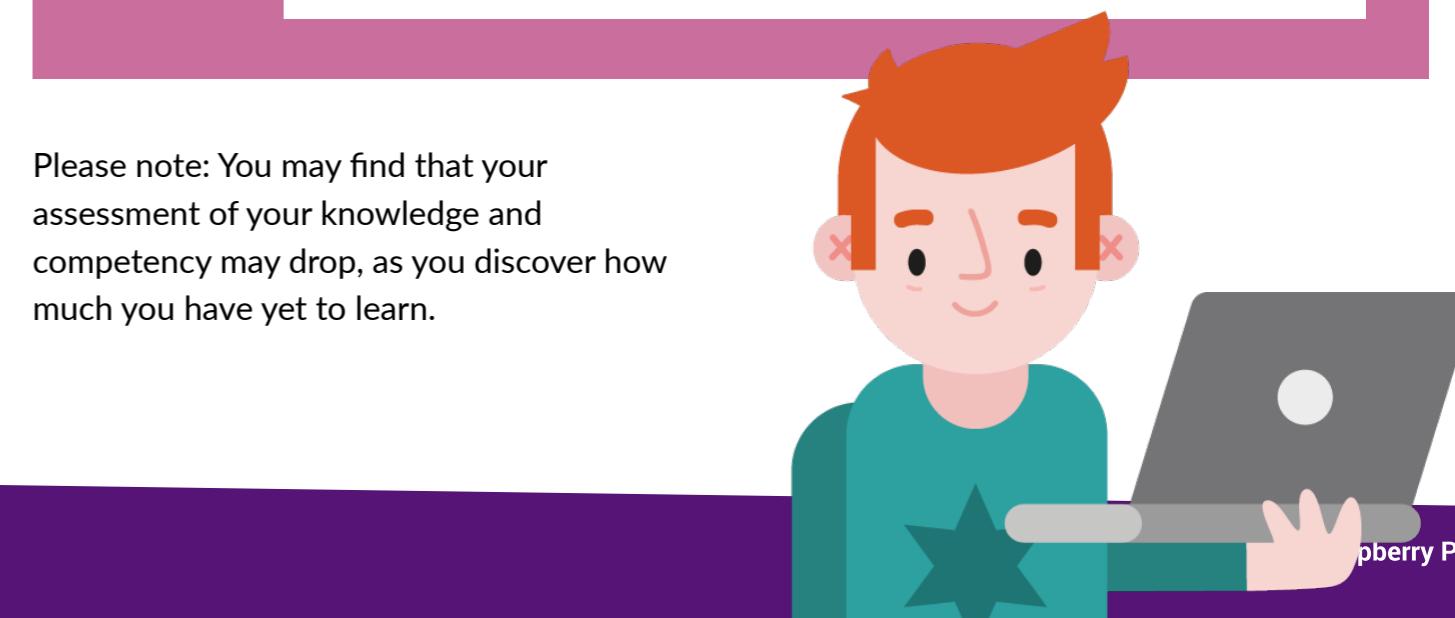
- 1 2 3 4 5 6 7 8 9 10

2

How confident do you feel to start a Dojo as a champion?

3

How confident are you to mentor at a Dojo?



Please note: You may find that your assessment of your knowledge and competency may drop, as you discover how much you have yet to learn.

Welcome (15 mins)

First, introduce yourself by name and talk a little about your background with CoderDojo.

For example:

'Good morning! My name's Sam and I've been mentoring at CoderDojos for 18 months. I'm really excited to work with you all today on getting started as volunteers in the CoderDojo movement. I've had a great time over the months and I'm sure you will too!'

Talk briefly about why you're all here, and outline the schedule.

For example:

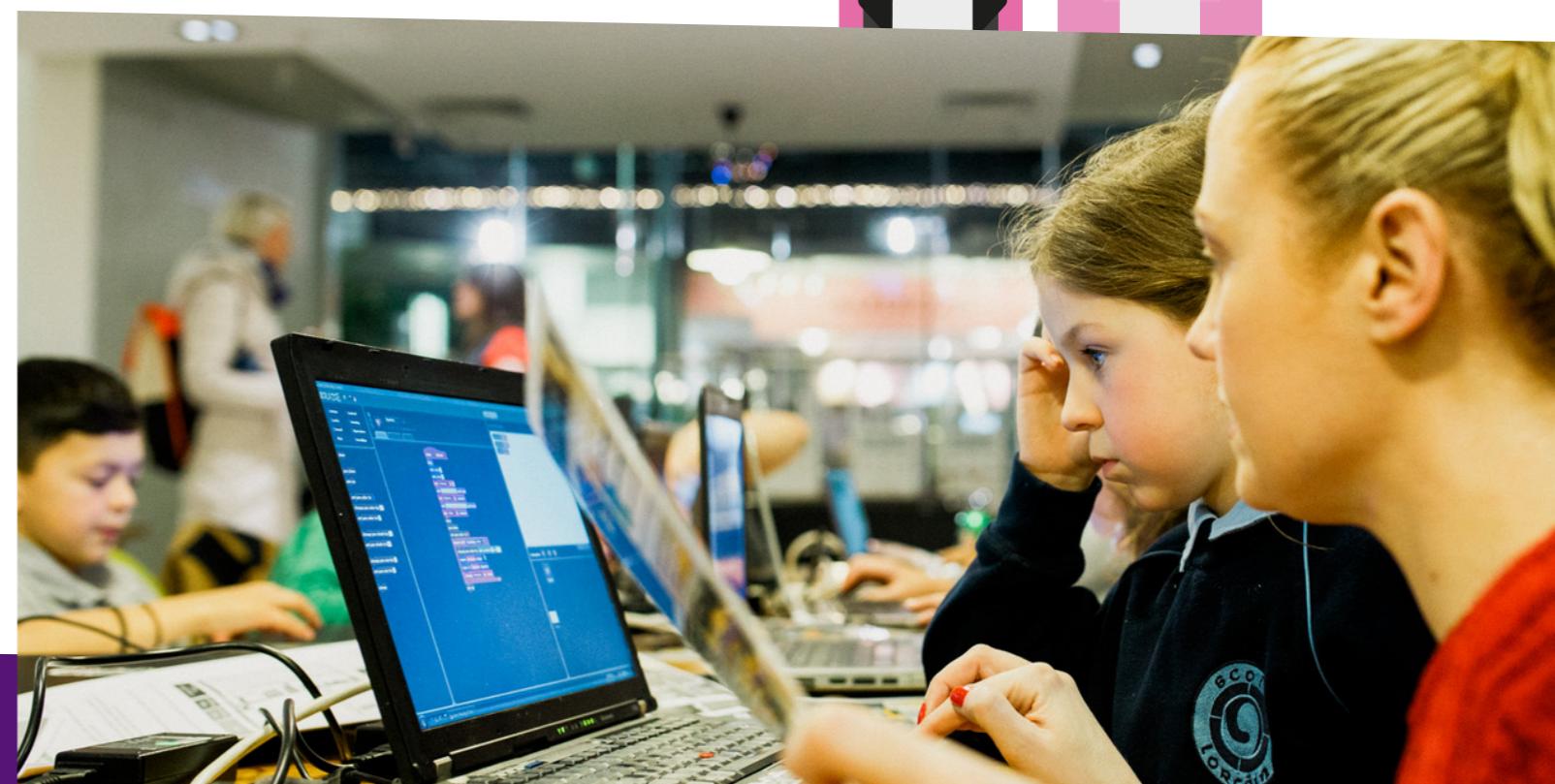
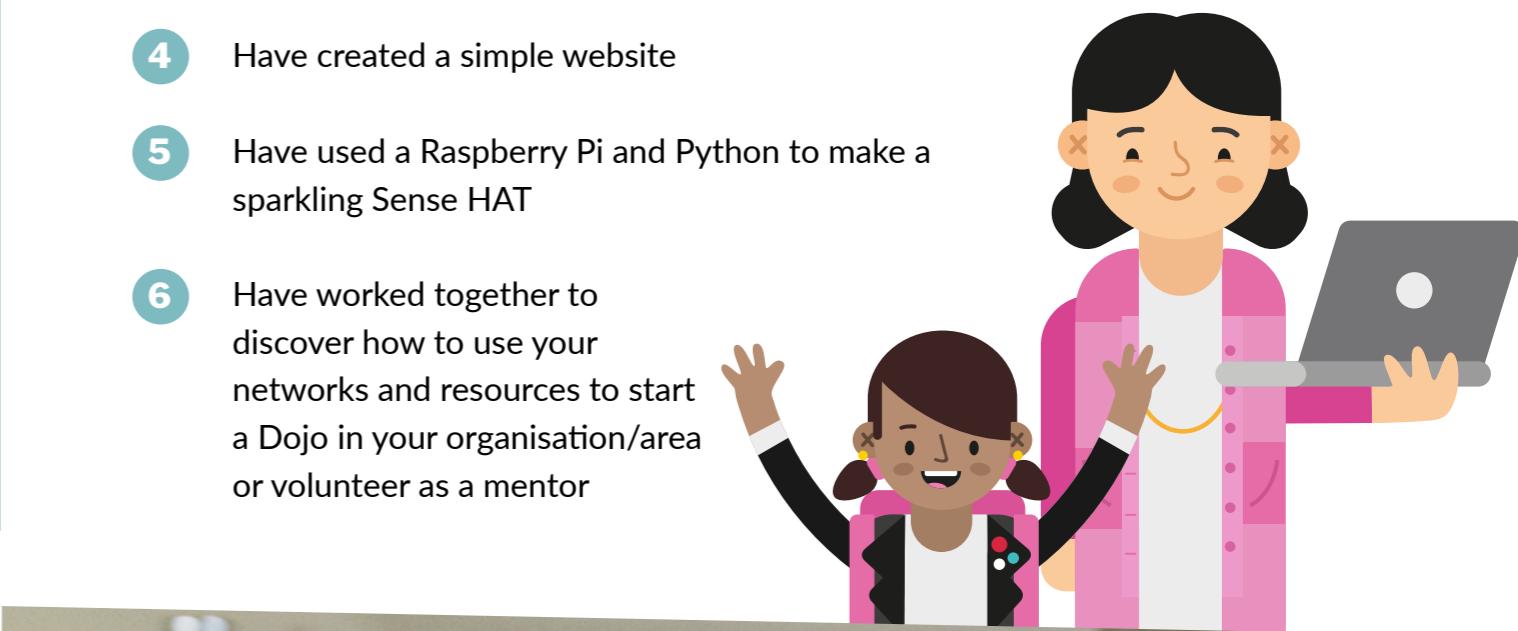
'Today we're going to introduce you to the basics of coding and the CoderDojo movement. We'll get to know each other and then move on to working on some code together. By the end of this session, you'll all have written your first computer programs!'



Course aims and objectives

Next, informally introduce the aims and objectives of the course. E.g. 'Today we are hoping that by the end of this workshop you will be able to do all these things!'

- 1 Have a greater insight into CoderDojo, the Foundation, its ethos, and how we work
- 2 Have worked together to create solutions for solving challenges that commonly occur in Dojos
- 3 Have worked through CoderDojo sushi cards in groups to complete a Scratch activity
- 4 Have created a simple website
- 5 Have used a Raspberry Pi and Python to make a sparkling Sense HAT
- 6 Have worked together to discover how to use your networks and resources to start a Dojo in your organisation/area or volunteer as a mentor



Icebreaker

In this activity, participants will introduce themselves and answer the following questions.

- Why do you want to volunteer with CoderDojo?
- What is your existing level of experience with computers?
- Do you have any experience working with young people?
- Tell one unusual fact about yourself.

Aim: To give participants a chance to get to know one another in a relaxed way, in order to create a comfortable atmosphere for engaging in training and linking together to create resources for Dojos.

What you need: Room set up in a participatory way, e.g. in a circle or three sides of a square.

Attendees at the workshop may not know each other, or may not know each other well. Since they're going to be working together to develop their mentoring skills, you should fix that! This exercise should only take a few minutes and will help everyone get to know each other.

Introduce the aim of the exercise

For example:

'You're all here because you want to become CoderDojo mentors. Our mentors are part of our team and part of the CoderDojo community, so we're going to get to know each other!'



Explain how it works

For example 'there are four questions on the slide. We're going to go around the room in turn and listen to everyone answer them.'

Note that for large groups (more than 20 participants) you may want to consider having them break into groups for this exercise, to save time.



Model an answer

Give your own answer to the questions.

Make sure you have a good interesting fact prepared!

For example:

'My name is Daragh and I want to be a CoderDojo mentor because I loved attending my local Dojo as a teenager. It inspired me to pursue IT education and I want to give young people today that opportunity. I'm a moderately experienced computer programmer and I have a few years experience, on and off, working with young people at Dojos, but none before that. I live 5 minutes away from Jedward's family home in Lucan, Dublin but have never seen them in person, weird..!'

Start them off

Pick someone to start the introductions.

Proceed around the room in some order until everyone has introduced themselves.

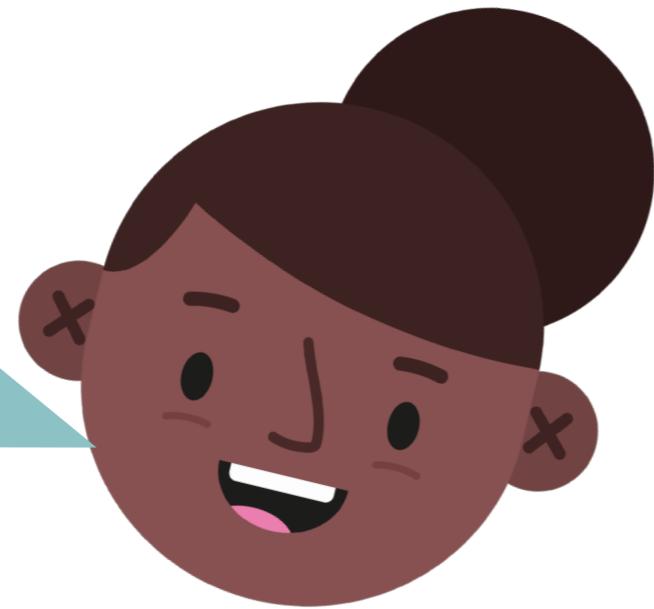
Discussion points

Afterwards invite answers to these questions in a group discussion

- 1 Did anyone find out anything about someone that surprised you?
- 2 Did you find anything in common with another participant? Keep in mind through the training that you can support each other by sharing resources.

Concerns and expectations (10 mins)

What are your concerns around becoming a CoderDojo mentor?



What do you expect to get out of this training course?

Aim: To gain an understanding of the reasons why participants are attending your workshop and what their concerns around joining the CoderDojo Community are. Ideally, you will be able to address most of these concerns over the course of the workshop.

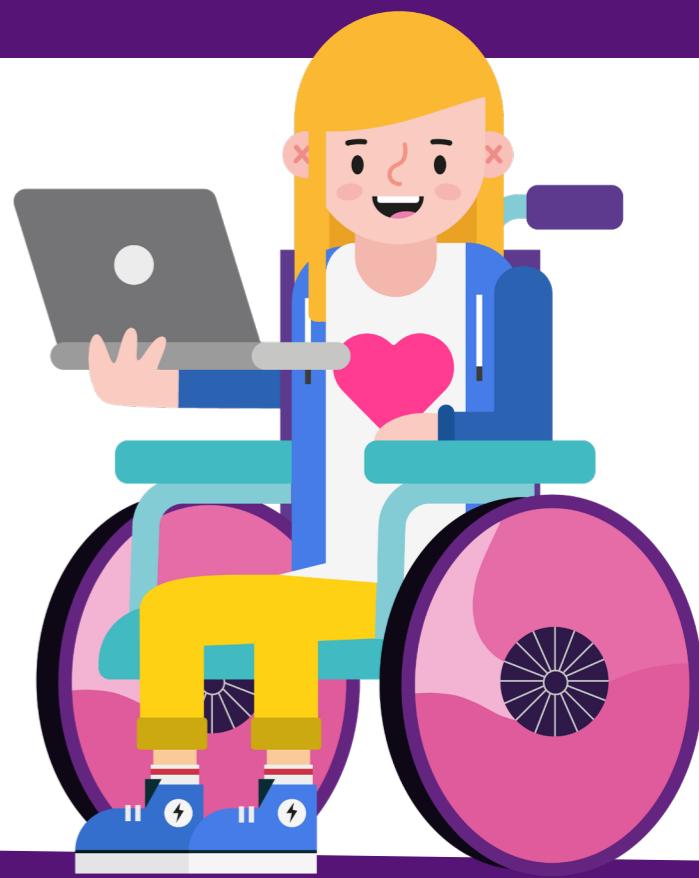
What you need: Post-it notes and felt-tip markers for participants. A flipchart and felt-tip marker for you.



- Ask participants to write answers on Post-it notes or pieces of paper, one per person. Ask for at least one concern and one expectation from each participant.
- If you aren't getting a response, suggest some concerns or expectations you originally had before becoming a mentor.
- Ask everybody to stick their Post-it notes on the flipchart or whiteboard, then look through them and group together similar concerns and expectations.
- If an expectation or concern seems likely to be shared by a few people, consider asking for a show of hands as to who shares it.
- Read each answer or group of answers (if they are similar) and inform the group if these will be addressed in the session.
- Label an area as the 'parking lot'. This will give you a space to put expectations that won't be addressed on the training day, but at the lunch break, you can write where to find the answers, or where people can learn more.

Discussion points

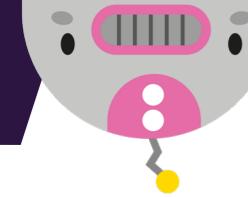
- Are there any areas that we haven't addressed?
- Are you all ready to start the course?



Why volunteer?



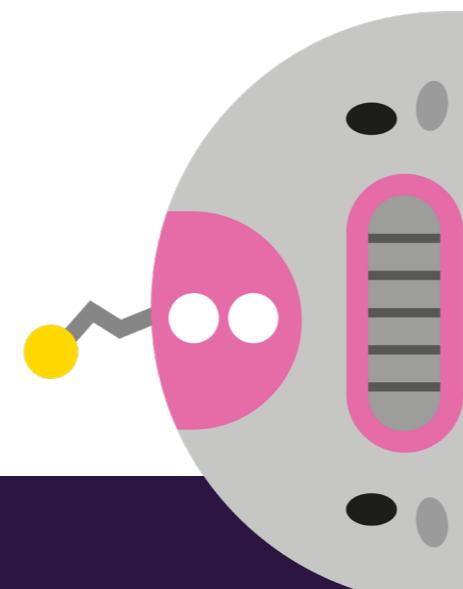
The CoderDojo movement



What is CoderDojo?

Introduce CoderDojo to the group and emphasise the two points on the slide

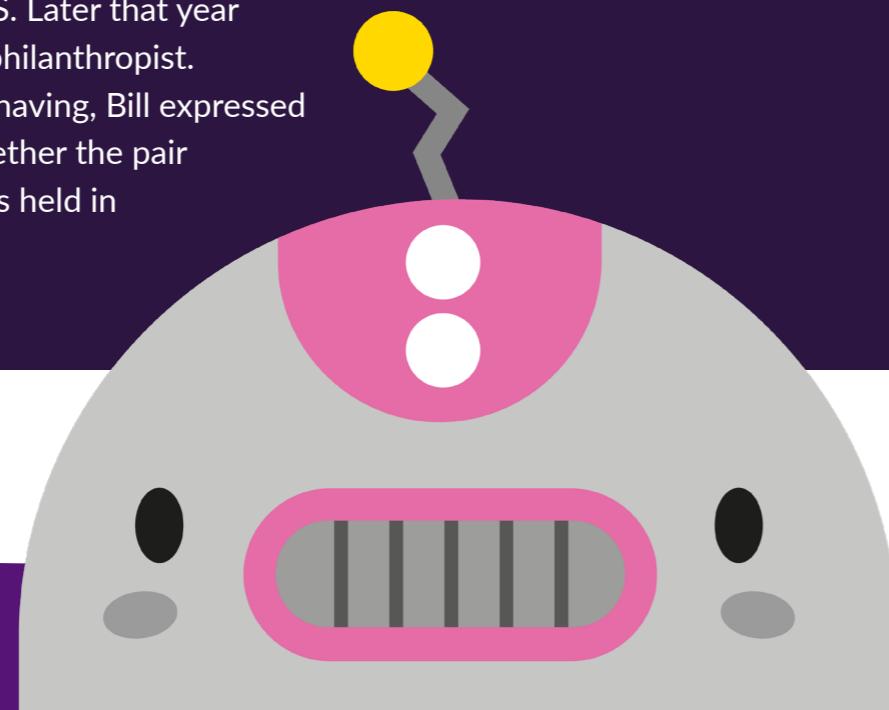
- CoderDojo is a global movement of free, open coding clubs (Dojos) for young people (aged 7–17)
- CoderDojo's mission is to give young people around the world the opportunity to learn to code in a relaxed and fun environment



History of CoderDojo

Introduce the history of CoderDojo

Tell the story of how James came up with the idea. In early 2011, when James Whelton was an 18 year old coder in Cork, he received some publicity after hacking the iPod Nano. As a result, some of the younger students at his school expressed an interest in learning how to code. He set up a computer club in his school where he started teaching students basic HTML and CSS. Later that year he met Bill Liao, an entrepreneur and philanthropist. Seeing the positive impact James was having, Bill expressed an interest in growing the project. Together the pair founded CoderDojo. The first Dojo was held in Cork on 23 July 2011.



CoderDojo philosophy

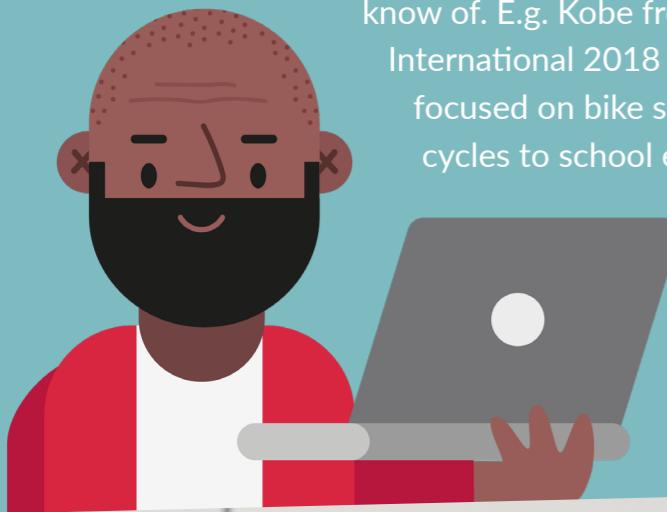
A Dojo is a space for learning. Let your participants know this as it will help set the scene for how Dojos are formatted.

Go through each of the six principles of the CoderDojo philosophy, giving a description of each.

- 1 **One rule: be cool!**: Try to convey that we avoid scolding Ninjas (Dojo attendees) for misbehaviour, instead taking the approach of simply saying 'be cool!' In short it means that we all must behave well towards one another. Feel free to quote CoderDojo co-founder Bill Liao: 'Helping, sharing, supporting, encouraging, cooperating, and being kind are all very cool. Bullying, lying, time-wasting, and disrupting the Dojo are all considered uncool.'
- 2 **Open source**: All resources and materials are available free online, including projects, guides and template letters for getting a venue or mentors. We encourage everybody involved in the community to create and share learning resources so that all Ninjas can access a wide range of projects in a variety of languages.
- 3 **Inclusive and free**: Dojos have to be free and all young people under the age of 18 are welcome to attend CoderDojo regardless of their ability, need, background, culture, religion, gender, sexual orientation or economic circumstances.
- 4 **Collaboration and teamwork**: At CoderDojo we encourage youth mentoring in Dojos. Introduce the 'Ask three, then me' motto here. Explain how young people should look to other sources including fellow Ninjas and online forums before asking a mentor, and detail how this approach often starts young people on the journey to youth mentoring as they get used to educating each other. Working together in a team always makes projects easier and more enjoyable.
- 5 **Informal and fun**: Starting a Dojo with an unplugged activity creates a fun environment from the beginning. This mixed with allowing the Ninjas to roam (within reason) around the room and to discuss problems they're having, or the cool things



- 6 Changemaking:** Explain that Ninjas are actively encouraged to create projects that tackle issues that are important to them. Give some examples that you know of. E.g. Kobe from Belgium attended Coolest Projects International 2018 with his project 'Light up my helmet'. His project focused on bike safety and he chose this as he is an avid cyclist and cycles to school every day. He created a bike helmet with LED strips and paired it with a controller attached to his handlebars. The controller is used to turn the indicator lights on and off.



they have created, with other Ninjas really creates an informal atmosphere that is different from that found in a school.

Unplugged activities (15 mins)



Aim: To introduce the concept of unplugged activities and demonstrate their value by participating in an activity and discussing what happens in the activity.

What you need: For this exercise you will need to print off some instruction cards. They can be found here: dojo.soy/TrainingCards. They contain a 'control' card, a foreign word (we use Klingon), and some instructions to make a noise (beep, whoosh, etc) or perform an action (jump, wave or sleep).

Start by explaining what unplugged activities are: coding and coding-related activities that don't actually require a computer. Lots of these are computational thinking tasks where Ninjas can learn to break down tasks and design solutions to problems just like programmers do. Say that these unplugged activities can be used as fun icebreakers at the start of a term or when new Ninjas or mentors join. They can also be used regularly as a warm-up in your Dojo to get people settled in. After you have given an introduction to unplugged activities, say that the entire group and extra facilitators (if any) will be participating in an unplugged activity called 'Human computer'.



Human computer

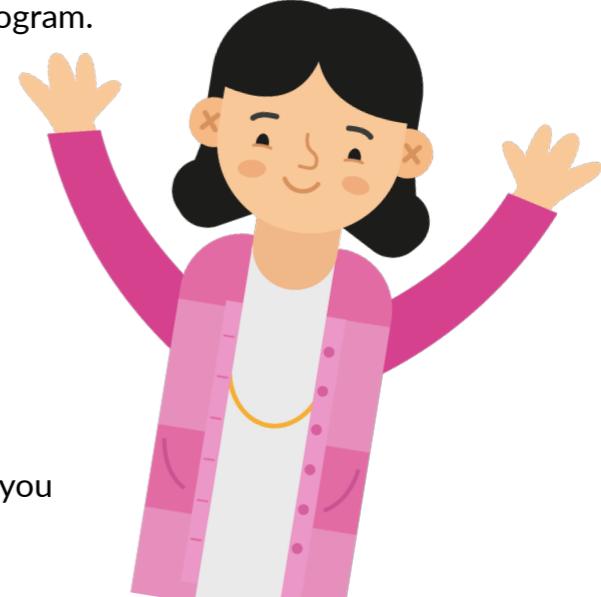
- 1 Ask all participants to line up.
- 2 While giving one instruction card to each participant, explain that a computer program consists of lines of code and each person is going to be one line of code as represented by their instruction. Programs run sequentially, so each line must finish before the next can start.
- 3 To start the activity, hand the control card to the first person in the line. Announce that the person with the control card must perform their instruction, and only when they have finished pass the control card to the person beside them, thus running our human computer! When all the instructions have finished, generate a round of applause and say that it was good.



- 4 Mention that computers are really really fast at carrying out their instructions. Ask the group to go again but much faster. Again, celebrate your speedy program with a round of applause.
- 5 Now tell the group that you are going to swap out some instructions just as a programmer may make changes (perhaps swap out some sleep instructions to increase speed).
- 6 Once the group has run the program with some straightforward swaps, select somebody in the third quarter of the line, swap their instruction for the foreign word, and ask the group to run the program again. They will probably run into difficulty when the control card reaches the person with the foreign word! You are introducing a bug here, something our computer cannot understand.
- 7 Now start the program again and cruelly wait for your unlucky bug's turn to run. Don't let them skip past your bug.
- 8 Now query the group as to what has just happened and why. Explain that when a computer program comes across something that it doesn't understand, it can't continue. The programmer must debug their program.

Discussion points

- 1 Does anyone know what happened here?
- 2 How do you think this relates to actual coding?
- 3 Could you do this as part of your Dojo?
- 4 Have you come across any other activities that you could use like this?



Notes: You can also tell the story of how the term computer bug came about. The terms 'bug' and 'debugging' are popularly attributed to US Navy Admiral Grace Hopper in the 1940s. While she was working with one of the first computers, a moth flew into a relay and stopped the machine's operation. The system then needed to be 'debugged'. They also taped the moth into the computer log entry. Pictures of this entry can easily be found online!

What is coding? (20 mins)



Aim: To gain an insight into the knowledge that the participants already have about computer programming, and to help them share that knowledge with each other, by asking them to define coding and piece together a definition with the group.

What you need: Post-it notes, felt-tip markers, a flip chart and a felt-tip marker for you.

Ask participants to write a definition of what they thinking coding is on Post-it notes or pieces of paper. If the group is small, go through each person's answer, reminding them that there are no right or wrong answers. If the group is larger, considering ask the group to stick their Post-it notes on a whiteboard/flipchart. This allows you to easily group similar answers together when you review them.

Review the answers, either by asking participants to call out their ideas, or by grouping them together on the flipchart. You should see some or all of the following keywords pop up in the answers. Discuss these ideas with the group.

Instructions: The idea that a program consists of a set of instructions to solve a problem.

Coding is the act of writing these instructions.

Languages: These instructions are written in special languages, called programming languages. These are designed so programmers can translate the solutions they come up with into code that the computer can understand.

Numbers/binary/ones and zeros: All languages eventually turn our code into a series of ones and zeros. These are used to represent electricity flow being on or off in our computer parts.

Programming: 'Coding' is also called 'programming' or writing 'computer programs'.

Make things/games/apps: Creating digital solutions to problems is the outcome of computer programming. This is often done by creating games, apps, websites and much more. Before you move on, present the slide with the 'make some tea' sample answer.



Discussion points

- 1 Does anyone have any other understanding of coding?
- 2 Do you have any further questions?
- 3 Is there anything that you do in your work/life that works in a similar way to coding?



Where do you find computer code?

Aim: This exercise gets the participants to really think hard about how much of the world around us is affected by technology and computer programs.

What you need: Post-it notes, paper, and pens or markers to write down ideas.

Making a phone call

Driving a car

Switching on lights

Going for a walk

Sending an email

Drawing a picture

Eating lunch

Taking notes

Playing video games

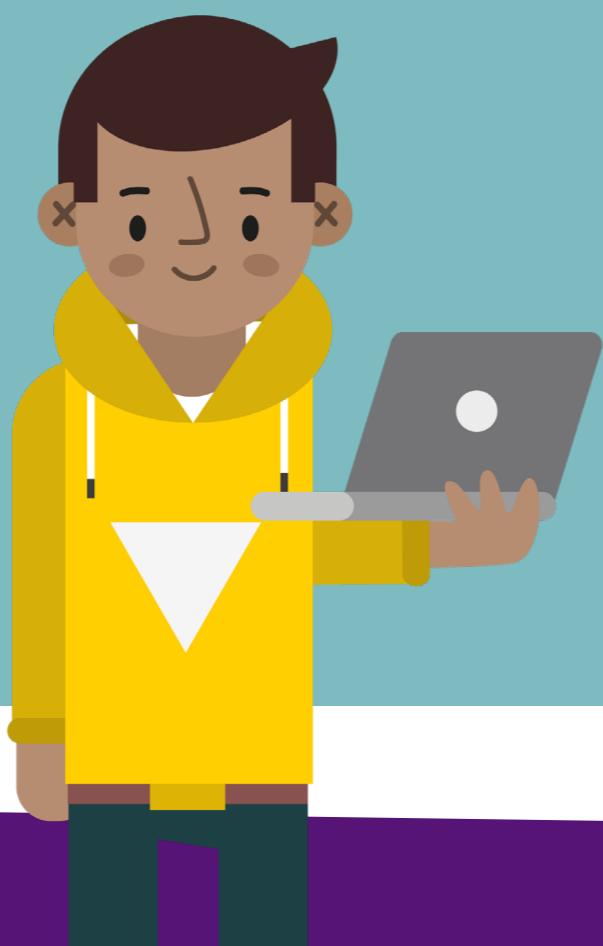
Reading a book

Taking a photo

Listening to music

Break the participants into groups of three or four and ask them to categorise the activities on the slide into four categories: those which **definitely**, **probably**, **might** and **don't** involve code. Before they start, say that as we are using the words 'probably' and 'might', answers are often down to the individual's opinion.

Ask which activities definitely involve code, hear peoples answers, and then reveal the answers you chose (or we chose, if you're using our slide deck). Then do the same for the other three categories in turn. Engaging in short debates about these is a great way for everyone to look at each scenario from different angles. Below are the reasons why we chose



Definitely

- **Phone call:** All mobile phones are computers, even back to 1980s models. Furthermore, the phone network itself is highly computerised.
- **Email:** You need a computer and software to send or retrieve email.
- **Video games:** Made with code.

Making a phone call

Driving a car

Switching on lights

Going for a walk

Sending an email

Drawing a picture

Eating lunch

Taking notes

Playing video games

Reading a book

Taking a photo

Listening to music



Probably

- **Driving a car:** Almost all modern (last 30 years) cars contain at least some sort of computer, running some amount of code. The newer the car the more obvious it will be, but cars back as far as the 1970s have used computers to make it easier to operate them. A modern car contains dozens of computer chips.
- **Taking a photo:** Almost all cameras these days are digital, recording photographs as computer data that can be read again later to display the image.
- **Listening to music:** Likewise, most music is stored in digital files (MP3, AAC, etc.) and requires code to play back. Even CD players were a form of digital storage, played back by a simple computer. The last popular music format that didn't use code was the cassette tape.

Making a phone call

Driving a car

Switching on lights

Going for a walk

Sending an email

Drawing a picture

Eating lunch

Taking notes

Playing video games

Reading a book

Taking a photo

Listening to music

Might

These all involve a somewhat needlessly technical, or unusually advanced, approach to things: walking with a pedometer, smart home lighting, drawing on a tablet rather than paper, using a Kindle to read, etc. However, it's possible to do all of them with tech baked into the process.

Don't

Physically eating lunch doesn't involve code but it might be worth mentioning that ordering food from an app would, as would cooking with most new ovens or microwaves.



Discussion points

It is really important to facilitate this activity in a way that allows people to have a discussion. Try to ask questions that will create discussion rather than telling participants the answers. E.g. 'Does anyone disagree with any of these? Why?'

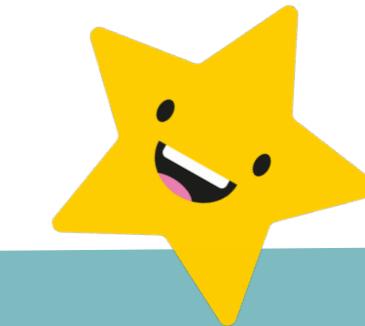
Tip: Try to avoid saying 'no' to people's answers and dismissing their opinions. In every session carried out, people had valid examples of where coding is evident.





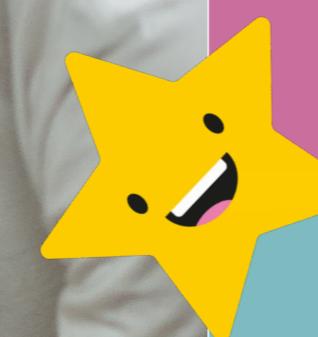
Coding and young people (30 mins)

CoderDojo lingo



CoderDojo

An international movement of free coding clubs for young people.



Dojo

An individual CoderDojo club. For example, 'the Midway Dojo runs every second Saturday'.



Ninja

Young people aged 7 to 17 attending and participating at a Dojo.

Champion

A Dojo organiser, either the original founder of the Dojo or someone who has since taken on the role.

Mentor

Volunteers who support, guide, and encourage the Ninjas as they develop their skills.

Have you heard any other CoderDojo terms you'd like to know the meaning of?



These slides introduce terms used in the CoderDojo community. It can be very beneficial if you have a story about the difficulties you faced with this language when you first joined the community. For each term, ask the group if they know what it means before revealing the definition. Finally ask if there are any other terms they'd like to know the meaning of and tell them, if you can. If you need help understanding any terms you or your participants have come across, please contact us at info@coderdojo.com.



Recap: CoderDojo philosophy

Announce that everybody will be splitting into groups for the next activity (aim for 3–6 groups). While they are moving around, remind them briefly of CoderDojo philosophy.

- 1 One rule: be cool!**
- 2 Open source**
- 3 Inclusive and free**
- 4 Collaboration and teamwork**
- 5 Informal and fun**
- 6 Changemaking**



Scenarios



Aim: To work together to create solutions for solving challenges that commonly occur in Dojos.

What you need: Post-it notes or paper and pens or felt-tip markers to write down ideas.

You will also need to download and print the scenario cards, which can be found at:
dojo.soy/TrainingScenarios.

- 1 Give each group one or two scenario cards.
- 2 Ask them to discuss the scenario in their group and come up with solutions to deal with each situation.
- 3 After five or ten minutes, ask each group to read through their scenarios in any order and then describe how they would approach the situation.
- 4 When they have finished, open the scenario up to the wider group or add your own ideas!



Discussion points

- 1 Do you think any of these are real life scenarios a Dojo would face?
- 2 Are there any other scenarios that you would like to discuss?



Scratch (45 mins)



Scratch is a block-based programming language created at MIT. It was designed to be accessible to beginners, who can write their first program in minutes.



In this section you will introduce the Scratch programming language to the group. Use the Scratch introduction slide to give a brief overview of Scratch. Cover the following points:

- **It's perfect for beginner programmers**
- **You can get your first program up and running in just minutes**
- **Due to its being block-based, many typing errors are eliminated**
- **It's great for learning by experiment (you can add things in to see what happens)**
- **It's very creative, allowing users to draw their own programmable assets to create games and digital stories.**



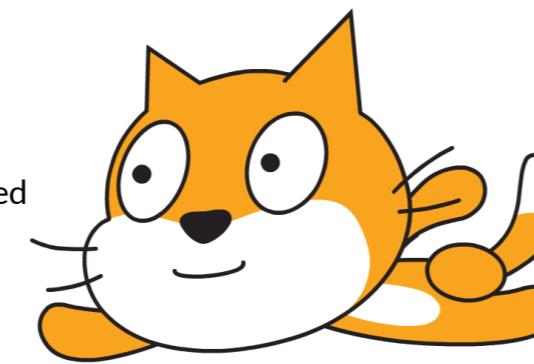


Activity: Make hippos fly!

Break the class into groups of two or three, and ensure each group has a computer or laptop. Hand out physical copies of the Scratch Hippo project if you have them; if not they can open the project in another tab or window. It can be found at dojo.soy/miniscratch. Then let the group work on the project for 30 minutes. Make sure no-one has problems getting started; remember it might be their first time using Scratch! Treat this like a Dojo! Once everybody is up and running, move around keeping an eye on everyone's progress. Encourage teams to help each other if they get stuck.

Scratch challenges

After ten minutes, move onto the next slide. Here we've listed some extra challenges for groups that finish early. Also encourage groups to work on their own personal creations. At the end, you can demonstrate a show-and-tell session where the groups can show off their crazy creations!

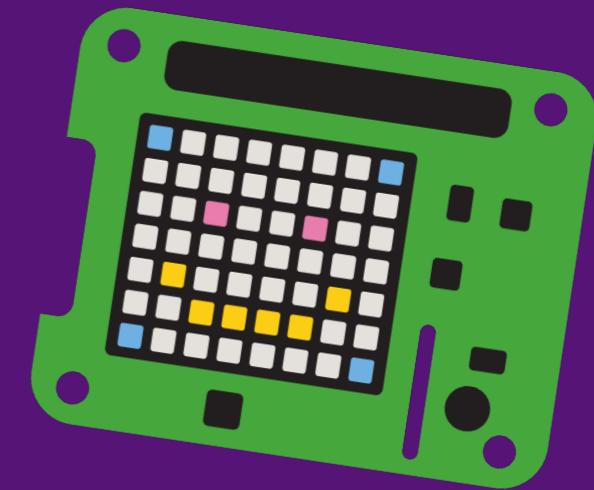


Optional hardware activity: Introduction to Raspberry Pi and the Sense HAT

The Raspberry Pi is a tiny low-cost computer designed to help people understand and learn about computer hardware and programming.

The Sense HAT is a collection of LEDs and sensors that can be attached to a Raspberry Pi to add a variety of inputs and a sparkly set of multi-coloured lights to play with.

- Shiny lights, joystick, and button
- Detect, humidity, temperature, and rotation
- There's one in space — astro-pi.org
- Python-based Sense HAT projects can use the free simulated Pi and Sense HAT at trinket.io/sense-hat



This section is designed to overcome attendees' fear of the unknown when it comes to some pieces of hardware. Ideally you will have the physical hardware available to use, but it's not essential.

You can also use trinket.io/sense-hat to let participants write code designed for the Raspberry Pi Sense HAT, and it will run on a 3D emulator.



Check out projects.raspberrypi.org for lots of options for projects using these pieces of hardware. An easy project to introduce the Sense HAT can be found at rpf.io/sparkles. Introduce the piece of hardware (and pass it around if you have one) and then, as in the Scratch activity, break into groups and allow people to try a project. Again, treat it like a Dojo and offer assistance where needed!



Making websites

Websites are a good starting point for slightly older Ninjas (twelve years and up) and a good progression from Scratch after anywhere from 6 to 18 months.

Web languages introduction

In this section you will introduce three languages used in web development to the group. Use the 'Making websites' slide to give a brief overview of creating websites. Cover the following points:

- It's a great next step for people who are proficient in Scratch or looking to try something different
- The same tools, technologies and languages are used by professional web developers
- You can see quick, cool-looking graphical results almost instantly when making changes
- Lots of room to grow as Ninjas' skills and ambitions develop



Web languages

Explain the purpose of each of the three core web development languages (HTML, CSS and JavaScript):

- HTML is like the building blocks of a website. Everything that you want to appear on a website goes here, e.g. text, images, links, buttons, etc.
- CSS controls how everything looks: the colour, size or font of the text; the position of the images; whether or not your links have lines under them. These are only a few examples of the power CSS has over the appearance of a website.
- JavaScript is used to enable users to interact with the page. It can be used to manipulate HTML elements on a web page without reloading it. It is also used to allow browsers to communicate with servers. Examples of its use include dynamic information (information that continuously changes or updates), search bars, and button functionality.

Then show them some websites that further illustrate the points. Here are some examples I use:

- The first ever website (contains only HTML):
info.cern.ch/hypertext/WWW/TheProject.html
- **CSS Zen Garden** (www.csszengarden.com) showcases the power of CSS. This website has multiple different CSS stylesheets that you can swap between. The HTML remains the same, the only thing that changes is the CSS style file.
- **Web Developer browser extension:** chrispederick.com/work/web-developer. I use this to disable JavaScript. You can then demonstrate what happens to popular websites when this happens. As I'm living in Ireland, I use www.rte.ie. This is our national broadcaster, and without JavaScript, the search bar ceases to work and the live weather disappears too. Try to use a website that your audience will relate to.



Activity: Make a zoo website!

Break everyone into groups of two or three and ensure each group has a computer. Hand out physical copies of the 'Make a zoo website!' project if you have them; if not, they can open the project in another tab or window. It can be found at dojo.soy/minicollaboration. Then let the group work on the project for 30 minutes. Make sure no-one has any problems getting started. Remember it might be their first time using Trinket or writing code. Treat this like a Dojo! Once everybody is up and running, move around keeping an eye on everyone's progress. Encourage teams to help each other if they get stuck.

Note: www.trinket.io requires users to have their own account to upload their own pictures

Web challenges

After ten minutes, move onto the next slide. Here we've listed some questions for groups that finish early. Also encourage groups to work on their own personal creations. At the end, you can host another show-and-tell session where the groups can show off their fantastic websites and reveal their answers to the extra questions, if they have any.





Educational content (5 mins)

Describe the benefits of using pre-written content in a Dojo.

- You do not have to spend time creating resources
- Encounter fewer issues. Pre written content will have been heavily tested and support is readily available if you do encounter problems
- Vast array of choice available so that the Ninjas can choose content that interests them

The CoderDojo Foundation and the Raspberry Pi Foundation have lots of resources available: coderdojo.com/resources and projects.raspberrypi.org.

These resources are available in many different languages too! Mention there are many projects available online for free around the web.



The CoderDojo community (20 mins)

Our global community

In this section you can introduce the wider CoderDojo community. As of October 2018, there are more than 1900 Dojos spread across over 100 countries. 58,000 young people are being creative with technology with the help of 12,000 volunteers. You can find up-to-date numbers at coderdojo.com.



Stay in touch with the community

Detail the many ways that community members stay in touch with each other and with the CoderDojo Foundation. Our community platforms are the best place to hear about news, events, resources and more. Details are available at dojo.soy/chat.

Mentor meetups

Describe mentor meetups. Group of local mentors and volunteers get together to discuss and share ideas, successes, events and plans for the future. They often bring tea and biscuits or beer and pizza!

DojoCon

Talk about the CoderDojo community conferences, DojoCons. They're always full of interesting talks on methods of mentoring, new technologies, new resources and much more. The CoderDojo Foundation is available to help all CoderDojo communities to organise and run regional DojoCon events in their area. More information can be found in our recent blogpost dojo.soy/HowToRunADojoCon.

Coolest Projects

Talk about Coolest Projects, the annual showcase event where Ninjas from CoderDojo, Code Club and Raspberry Jams get together to show off the projects they have been working on. There are a lot of regional events around the world, including in the USA, UK and more. Then there is the huge international event that occurs in Ireland and is usually held in May. As with DojoCon, local communities can run a regional Coolest Projects event as well as entering the International event in Dublin. See coolestprojects.org for more details!





Starting a Dojo (20 mins)

In this section you will tell participants what is needed to start their own Dojo. Go through the ingredients they need to get started, get people networking within the group, and describe the process of verifying a Dojo.

Ingredients of a Dojo

Here you can go through everything needed when starting up a new Dojo.

- **Register as a champion** on Zen, the platform used on coderdojo.com.
- **Find a suitable venue.** This must be a public venue. Great examples include libraries, technology hubs, community centres, offices, computer labs in schools, and universities.
- **Build your team.** Try to find a mix of technical mentors and non-technical mentors who have experience with children. Ask parents of Ninjas attending if they'd like to get involved. Ask companies to enquire if any of their staff would like to mentor at your Dojo. Make an announcement at a university to get students involved.
- **Plan your first Dojo.** It's easier to grow a Dojo once it's up and running. You don't need 20 Ninjas to start a Dojo; many start with as few as three or four young people. If they each bring a friend or two, you'll quickly have a Dojo filled with excited and eager innovators. Just pick a date that suits you and dive straight in. Look at a beginner project in Scratch or HTML yourself so you can stay one step ahead of the Ninjas. **Don't fear questions**, CoderDojo is all about learning together. If you can't answer any questions try using Google or our community forums, or get in contact with us at info@coderdojo.com.
- **Promote your Dojo.** Create an identity for your Dojo. You could do this by creating social media pages. Try to get an ad (for free!) in the local newspaper. Put up posters in high traffic areas. Try to get it circulating in school/club newsletters. Visit dojo.soy/style to get our logos, colours and fonts. Feel free to change it up a little bit and make your Dojo unique. Look at some of the Dojo twitter accounts from around the world for inspiration.

Build your Dojo

Aim: To use connections and skills in the group's network to find all the ingredients of a Dojo. Then use these to create a plan to start a new Dojo.

What you need: Flipchart or paper and markers to write ideas.



- 1 Break the participants into groups. Try to get them together with people who are from different organisations/backgrounds.
- 2 Give each group an A3 flipchart page (or paper), and some markers. Ask them to write the ingredients of a Dojo on the page and try to fill in where they would find them.
- 3 If they have an ingredient already, ask them to put a gigantic tick symbol beside it.
- 4 Give each group ten minutes to figure out where they would get each ingredient, or a plan of where to find them. Note: Tell people to remember their networks and skills. CoderDojo is a community that depends on volunteers harnessing their local resources.
- 5 Give each group three minutes to present their 'Start a Dojo' plan.
- 6 Invite people from the rest of the group to support their ideas, ask questions, and offer their resources.

Discussion points

- 1 Did any of the groups miss any of the ingredients?
- 2 Are there any ideas from other participants for how to find them?
- 3 Are there any other networks that could help, such as libraries, colleges, parents, or other clubs?



Verify your Dojo

Finally run them through the verification process on coderdojo.com. You might like to do a live demo of this if it is likely that there will be a prospective Dojo created in the group. Emphasise that you do not need to fill in every detail in your application in one go. For example, you can add a venue, then come back a week later and fill in details about your first Dojo session or your team of mentors.



Feedback

Finally, as with all workshops, it is invaluable to collect some feedback so that you can improve your training sessions. You can do this the old fashioned way on paper, or use an online survey tool like SurveyMonkey to collect your feedback. An example feedback form can be found overleaf.

Self reflection

It is also very important to examine your own performance after facilitating a session like this. Ask yourself questions like the following after each one to keep improving

- **How did you do? Rate yourself 1–5.**
- **What was the aim of this training? Did you achieve your aims?**
- **How can you use this training in the future?**



If you need any more help please visit our helpdesk at dojo.soy/help, check out our forums at dojo.soy/sd-forums or email us at info@coderdojo.org. If you use this guide to deliver your own training sessions, please let us know how we can improve it by completing our short survey here: dojo.soy/TrainingGuideFeedback. Best of luck with your champion/mentor training sessions!

CoderDojo mentor training feedback form

Please rate your knowledge of coding after the training.

1

1 2 3 4 5 6 7 8 9 10

2

How confident do you now feel to start a Dojo as a champion?

3

How confident are you to mentor at a Dojo now?



4

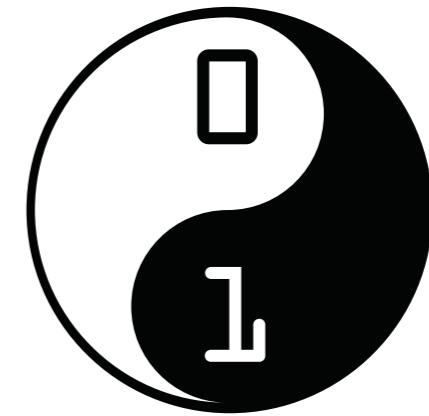
Do you have any other feedback about the CoderDojo champion training?

5

One thing I would change about the training is...

6

In the training, I would have liked to learn more about...



CoderDojo



CoderDojo

Find us on social media:



/CoderDojo

@coderdojo

@CoderDojo