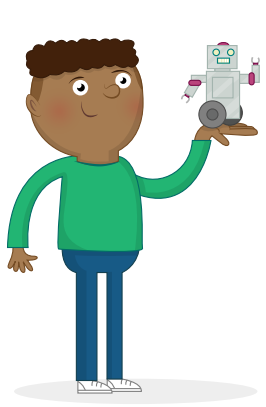
raspberrypi.org

	Creator	Builder	Developer	Maker
				
Design	Design basic 2D and 3D assets.	Combine 2D and 3D assets in the assembly of a project	Use multiple designed assets in completed products and models	Design multiple and integrating assets for use in complex finished projects and models
Programming	Use basic programming constructs to create simple programs	Combine programming constructs to solve a problem	Apply abstraction and decomposition to solve more complex problems	Apply higher-order programming techniques to solve real-world problems
Physical Computing	Use basic digital, analogue, and electromechanical components	Combine inputs and/or outputs to create projects or solve a problem	Process input data to monitor or react to the environment.	Create automated systems to solve complex real-world problems
Manufacture	Use basic materials and tools to create project prototypes	Use manufacturing techniques and tools to create prototype projects	Use manufacturing techniques and tools to create a completed product	Independently use fabrication systems to produce complex finished projects
Community and Sharing	Engage and share with the digital making community	Collaborate on digital making projects with other community members	Support others in the design and construction of their digital making projects	Support others in the design and construction of their digital making projects

Creator

Design

Design basic 2D and 3D assets.

Learners can use basic design principles to produce simple graphics, video, circuit schematics, formatted web pages, cutting sheets, and 3D model components, utilising CAD and graphics software, or markup languages.

Example Outcomes
- Can use CAD tools to create simple 2D designs for projects
- Can use a markup language to create and style text and images for a web page
- Can use graphical applications to manipulate bitmap and vector images

Example Projects
- Create a simple web page containing styled text and images
- Design a simple circuit using standard symbols
- Custom-design a polystyrene case to carry a payload into the upper atmosphere using a HAB

Programming

Use basic programming constructs to create simple programs

Learners are able to demonstrate awareness of simple programming concepts, such as sequencing, repetition, variables, and selection by writing short programs, probably in isolation.

Example Outcomes
- Can use simple control flow statements
- Can use variables and simple data structures
- Can use a variety of logical, arithmetic, and comparison operators

Example Projects
- Create a simple animation
- Create a simple game of scissors, paper, stone that can be played against the computer
- Write a program that can produce simple patterns and pictures using 'turtle'-style graphics

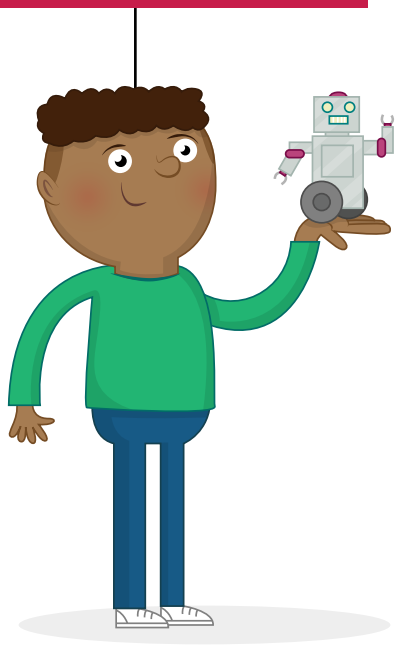
Physical Computing

Use basic digital, analogue, and electromechanical components

Learners can write programs that using active and passive electronic components. They can receive data from input components and control output components, using a computer or microcontroller.

Example Outcomes
- Can design and construct simple prototype circuits, using components connected directly to GPIO pins
- Can recognise and use polar components, such as LEDs and capacitors
- Can write programs to read data from digital and analogue inputs, and control digital and analogue outputs

Example Projects
- Create a simple traffic light system using LEDs and a buzzer
- Use motors to create a buggy that can drive a predefined route
- Measure the temperature in a room with a temperature sensor



Manufacture

Use basic materials and tools to create project prototypes

Learners can use tools to cut, measure, and join materials, and use bonding agents where appropriate, to produce simple parts for prototype projects. They can produce prototype circuits, following simple circuit schematics.

Example Outcomes
- Can create simple circuits, using a breadboard and jumper leads or crocodile clips
- Can use and manipulate upcycled materials for use in prototypes
- Can repurpose commercial products for use in project prototypes

Example Projects
- Use a breadboard and jumper cables to make a simple traffic light circuit using multiple components
- Use upcycled and modified materials as a housing for a simple weather station
- Use plastic bottles and drinking straws to make a model of the ISS

Community and Sharing

Engage and share with the digital making community

Learners interact with other members of the community, and share their creations. They may attend events and meetups, or interact with other community members online.

Example Outcomes
- Share a digital creation with the wider community
- Participate in an event within the wider community
- Engage in an online discussion about a digital making project

Example Projects
- Use safe online sharing platforms to showcase their digital creations
- Attend a Raspberry Jam and listen to some talks or attend a workshop
- Visit their local makerspace and learn from others in the community

rpf.io/learn

Builder

Design

Combine 2D and 3D assets in the assembly of a project

Learners can design 2D and 3D assets for use in projects, using CAD and graphics software, or markup languages. They can independently produce simple graphics, video, circuit schematics, formatted web pages, cutting sheets, and 3D model components for use in a project.

Example Outcomes

- Can use simple video editing tools to manipulate video footage
- Can use HTML, CSS, and JavaScript to produce browser-based content
- Can use CAD tools to produce component parts for 3D structures

Example Projects

- Create a sprite with an animated walk cycle for use in a computer game
- Create a website to showcase a digital making project
- Create a 3D model of a physical make to demonstrate its planned appearance and dimensions

Programming

Combine programming constructs to solve a problem

Learners understand simple programming concepts, and can combine concepts to produce projects that satisfy predefined outcomes or project briefs.

Example Outcomes

- Can create subroutines/procedures/functions in their programs
- Can programmatically read and manipulate a data structure
- Can use standard communication protocols and APIs to transfer data between computers and applications

Example Projects

- Use an API to discover the number of people in space and output this to a user
- Create a program that uses a weather API to display the current weather anywhere in the world
- Use simple communication protocols to create a game controller using two devices

Physical Computing

Combine inputs and/or outputs to create projects or solve a problem

Learners can write programs that use active and passive electronic components in combination with one another. They can receive data from input components and perform limited processing of that data, and control multiple output components, using a computer or microcontroller.

Example Outcomes

- Can use sensors to trigger real-world outputs, such as cameras and electromechanical devices
- Can use combinations of input and output components to transfer data between systems
- Use transistors and relays to power components requiring higher voltages

Example Projects

- Use an LDR and an LED to create an automated night light
- Use a PIR sensor to trigger video capture from a camera
- Use motors and a distance sensor to control a robot that can avoid obstacles

Manufacture

Use manufacturing techniques and tools to create prototype projects

Learners can choose and manipulate appropriate materials for use in their finished prototypes. They can use a variety of techniques and tools to accurately cut, measure, and manipulate materials. They can construct circuits by soldering to prototyping boards.

Example Outcomes

- Can create soldered circuits using stripboards or protoboards
- Can use tools to manipulate materials and create component parts for a prototype
- Can use a multimeter to debug circuits

Example Projects

- Solder a circuit using a breadboard or stripboard to produce a digital sound meter
- Hand-stitch or use a sewing machine to add components to a jacket, to make a mode detector
- Use a craft knife to cut laminated cardboard and create a 3D housing for a pet feeder

Community and Sharing

Collaborate on digital making projects with other community members

Learners collaborate on projects with other members of the community. This could be face-to-face or online, in either formal or informal settings.

Example Outcomes

- Collaborate in a digital make with other people
- Participate in a digital make event with others in the community
- Make improvements to or find bugs in a version-controlled project

Example Projects

- Create a project with another member of their Code Club, CoderDojo, or other organisation
- Team up with some friends and go to a hackathon
- Find someone from the community that has made a project similar to theirs, and ask them for help in their make

Developer

Design

Use multiple designed assets in completed products and models

Learners can design assets in a variety of media, and combine those assets into finished products. They can independently produce graphics, video, circuit schematics, interactive web pages, cutting sheets, and 3D models that meet a design brief.

Example Outcomes

- Can use CAD software to design circuits and PCBs
- Can use CAD software to design 2D components to be laser cut and perfectly assembled
- Can use CAD software to design 3D models and components to be 3D printed and assembled

Example Projects

- Create a 2D design for the components to a simple box that can be used in a laser cutter
- Create a 3D body for a robot that can be used in a 3D printer
- Produce a design for a PCB that can be etched or sent for fabrication

Programming

Apply abstraction and decomposition to solve more complex problems

Learners are able to work on larger problems. They are able to simplify and break up larger problems, making use of abstraction and decomposition, as well as more complex reusable data structures.

Example Outcomes

- Can decompose a large problem into parts and design algorithms to solve them
- Can recognise similar problems, and apply generic solutions and abstractions
- Can effectively combine functionality from multiple libraries or APIs and refer to documentation

Example Projects

- Create a program that uses a weather and messaging API to send updates on tomorrow's weather
- Create a maze game, where a player needs to navigate through a random maze
- Create a program that allows two people to send encrypted messages to each other

Physical Computing

Process input data to monitor or react to the environment

Learners can design and write programs that are able to process data from external sources and control output devices that react to, or alter, their environment, using a computer or microcontroller. They can design and write programs that are able to monitor various aspects of an environment.

Example Outcomes

- Combine and process data from multiple sources to build a real-time system
- Use, and extrapolate values from, input data to make predictions about a system
- Use output devices to react to live data

Example Projects

- Use LDRs, distance sensors, and motors to create a robot that can follow a line and avoid obstacles
- Use a variety of sensors to create a weather monitoring station that uploads data to a database
- Use a camera and servos to motion track a moving object in a room

Manufacture

Use manufacturing techniques and tools to create a completed product

Learners can use industry-standard prototyping and manufacturing techniques, such as 3D printing, laser cutting, and PCB etching, to produce and assemble components to be used in their finished projects.

Example Outcomes

- Use cutting tools and joining/bonding techniques to create a finished product
- Solder components to PCBs and test circuits using a multimeter
- Can manufacture products that integrate with existing systems

Example Projects

- Expose and etch a PCB for a simple traffic light circuit and then solder components to it, testing the circuit with a multimeter
- Use MDF and wood glue to produce a concealed housing for a burglar alarm
- 3D print a decorative ornament that contains flashing LEDs

Community and Sharing

Support others in the design and construction of their digital making projects

Learners offer support and assistance to other members of the community. They create content that other community members can follow to create projects that interest them.

Example Outcomes

- Assist others in a digital make, either online or in person
- Produce learning materials to guide others in a digital make
- Produce and maintain code that is shared through an online version control platform

Example Projects

- Create a YouTube tutorial for their project, showing others how it can be done
- Join their local makerspace and help others with their digital makes
- Give a talk at a developer conference, a Raspberry Jam, or a maker meetup

Design

Design multiple and integrating assets for use in complex finished projects and models

Learners can design a range of assets in a variety of media, to be used in projects and models that solve real-world problems. They can analyse a problem and then produce graphics, video, circuit schematics, interactive web pages, cutting sheets, and 3D models that assist in solving that problem.

Example Outcomes

- Use PCB design tools to create PCBs that conform to appropriate dimensions and component positioning
- Use CAD tools to design housings and component parts that integrate into a finished product
- Use graphics and video editing tools to produce complete audiovisual products

Example Projects

- Produce a detailed character sheet for use in a graphical computer game
- Produce a 3D model of a character that can be imported into a game engine
- Produce a dynamic website that displays data harvested from an API

Manufacture

Independently use fabrication systems to produce complex finished projects

Learners can appropriately combine a variety of industry-standard manufacturing techniques to independently produce and assemble components for a finished project.

Example Outcomes

- Use a 2D cutter (laser, jigsaw, water, ultrasonic) to produce 2D components for assembly
- Use a 3D printer/router to produce a completed product or components for an assembly
- Use exposure and etching techniques to produce custom PCBs

Example Projects

- Use a laser cutter to produce 2D acrylic sides for a housing for a model traffic light, including holes for projected components like LEDs and buttons, and integrated space for a PCB
- Print a 3D housing for a Pi Zero Game Boy clone
- Produce a custom PCB for an automated pet feeder

Programming

Apply higher-order programming techniques to solve complex real-world problems

Learners are able to make use of a variety of programming paradigms, and combine data from various systems to solve complex, real-world problems.

Example Outcomes

- Can use a variety of programming paradigms in their programs, choosing techniques appropriate to the problem being tackled
- Can implement and use a variety of more complex data structures in their programs, such as trees, graphs, and sets
- Can integrate user interface constructs such as GUIs into their solutions

Example Projects

- Create an importable module that interacts with the Minecraft
- API, allowing the easy construction and manipulation of large structures.
- Create an automated maze generator for producing differently sized mazes that are suitable for displaying on an LED matrix Create a web-based front end for a HAB balloon, where sensor data is displayed in real time when received via radio

Maker

Physical Computing

Create automated systems to solve complex real-world problems

Learners can analyse a problem and design and create automated systems that monitor, react to, or influence an environment, in order to solve a real-world problem.

Example Outcomes

- Process multiple data and/or input sources, and use them to control multiple and interconnected output devices
- Use online data sources and inputs to control output devices
- Publish data from sensors to an online platform and display the data graphically.

Example Projects

- Create a system that monitors and controls the environment of living things such as fish or plants
- Create an early warning system that monitors local weather conditions and warns people of impending weather
- Produce a mechanical arm that can be monitored and controlled over the internet

Community and Sharing

Educate others in the skills and ethos of digital making

Learners take on an educator or mentor role to assist in the learning of other community members. They may take part in workshops and seminars, or inspire others through educator-focused volunteering.

Example Outcomes

- Regularly assist others in their digital making experiences
- Educate or assist large groups of individuals in their digital makes
- Regularly publish learning materials for use by the community

Example Projects

- Write tailored resources that meet the needs of a particular audience, e.g. children attending a Code Club
- Become a Code Club or Pioneers volunteer
- Run a workshop at a Raspberry Jam or a conference

Computational Thinking

One of the aims of the Raspberry Pi Foundation is to help people to learn about Computer Science and how to make things with computers. We believe that learning how to create, control, and contrive technology will help people shape an increasingly digital world and prepare them for work of the future.

Computational thinking is at the heart of the learning that we advocate. It is the thought process that underpins computing and digital making: formulating a problem and expressing its solution in such a way that a computer can effectively carry it out. CT covers a broad range of knowledge and skills including, but not limited to:

- **Logical reasoning**
- **Algorithmic thinking**
- **Pattern recognition**
- **Abstraction**
- **Decomposition**
- **Debugging**
- **Problem Solving**

By progressing through our curriculum, learners will develop computational thinking skills and put them into practice.

Glossary

A

Abstraction Representing a concept by hiding unnecessary details, revealing only the details that are important.

Active sensor Sends a signal and then detects changes in the environment based on the data returned. Usually requires an external power source.

Algorithm A series of precise instructions to solve a problem.

Analogue signal A continuous variable signal.

API Application Programming Interface. A set of functions used when building software.

Assets Items needed to create a project, such as images or 3D-printed parts.

B

Bonding agent Adhesive for sticking components together (e.g. glue).

Breadboard A reusable circuit board for prototyping electronic circuits.

C

CAD Computer-Aided Design. Allows precise 2D and 3D drawings and models of real-world objects to be designed on a computer.

Circuit A path along which current flows.

Computational thinking A set of mental skills that help to define a problem in such a way that a computer can solve it.

Control flow The order in which a computer executes instructions.

D

Data structure A format for storing and organising data (e.g. an array).

Decomposition Breaking a problem down into smaller, more manageable parts.

Digital signal A signal that is either on or off.

Digital making Creative projects requiring technical skill and understanding.

Glossary

G

GPIO General-Purpose Input/Output. Pins connected to the Raspberry Pi that can be controlled with code.

I

Input Data entered into a program (e.g. a mouse click).

M

Markup language A language specifying the presentation and style of text.

Microcontroller A small computer using a single chip.

O

Output Data sent out of a program (e.g. sound from a speaker).

P

Passive sensor Detects changes in the environment without actively sending out a signal (e.g. a PIR sensor).

PCB Printed Circuit Board. A board connecting electronic components.

Programming paradigm A style or way of thinking about programming (e.g. procedural, object-oriented).

Protoboard Allows you to add components to a prototype circuit by soldering or cutting strips.

R

Repetition Executing the same code multiple times (while/for loops).

S

Schematic A diagram or plan representing a circuit.

Selection Choosing which code to execute based on a condition (if/else).

Sequencing Executing lines of code one after another.

Subroutine A named set of instructions in a program (also referred to as a function/procedure).

U

Upcycled Using materials which otherwise would be thrown away, in order to make something of value.

V

Variable A named area in memory where data is stored.