LeetCamp Creators: 4-Week Teaching Guide

Group: Creators (Grades 4 to 6)
Taught in this camp: HTML, CSS and a little JavaScript. (Advanced Scratch appears on the camp site as part of the Creators toolkit, but we do **not** teach it here. The time goes into building the website together.)
Schedule: Monday to Friday, 4 weeks = **20 days** (19 build days plus a play day and party on Day 20), 9:30am to 1:00pm each day
Final product: a personal showcase website with an interactive quiz, shown to classmates on Day 19 and kept for next year's Tech Showcase.

How we teach: build it together

The whole class builds the **same project at the same time**. The teacher live-codes on the projector; learners code along on their own laptops, then make it their own with their own topic, colours and words.

Stage	Who codes	What it looks like
I do	Teacher	Teacher types one small piece on the projector and explains why.
We do	Everyone	Learners type the same piece together, teacher circulating.
You do	Learner	Learners repeat it for their own topic, colours and content.

Repeat in small loops of 10 to 15 minutes. Never talk for more than 15 minutes in a row.

The daily rhythm (same every day)

About 3 hours of teaching time per day.

Time	Slot	What happens
9:30 – 9:45	**Warm-up**	Arrive, open projects, 2-minute recap: "What did we do yesterday? What's today's goal?"
9:45 – 10:45	**Block 1: Build together**	Teacher demos (I do / we do), learners follow, then personalise.
10:45 – 11:10	**Break (25 min)**	Snack, water, move around. Screens off.
11:10 – 12:30	**Block 2: Build and test**	Longer personalising time with the teacher circulating; a new small feature.
12:30 – 1:00	**Test, Debug, Show**	Partner testing, fix bugs, "show your neighbour", quick class share.

> If the break runs to 30 minutes, trim Block 2 and protect the last half hour. That is where learners practise debugging and presenting.

The project: "My Topic Showcase"

Each learner picks **one topic they love** (Kenya's national parks, football, space, animals, music…) and builds a two-page website:

1. **Home page:** a header, an intro, a "cool facts" card row, an **interactive quiz**, and a footer, all built from `<div>` containers.
2. **Who I am page:** the learner writes about themselves, in their own words.

The reference site in this folder (`index.html`, `about.html`, `styles.css`) is the **teacher demo**. Show it on Day 1 so learners see the finish line, and reuse its patterns (navbar, 3-card grid, footer).

The 3-step loop (repeat every week)

Step	Meaning in class
Plan and Build	Sketch on paper, then build in small chunks.
Test and Debug	Test, find the bug, fix, test again, until it works.
Deploy and Showcase	Swap with a partner, demo to the class. Week 4: the class (Day 19).

Every Friday is a Test, Debug and Showcase day.

WEEK 1: Plan and build the structure (HTML)

Weekly goal: a complete, plain home page with all content in containers.

Day 1 (Mon): Welcome, topic and plan

- **Warm-up:** icebreaker. Each learner says their name and one thing they love.
- **Block 1:** demo the finished showcase site: "What do you notice?" Introduce the 3-step loop and the three tools: HTML = skeleton, CSS = style, JS = behaviour. Learners choose a topic and complete the **planning sheet**: topic, audience, 5 cool facts, 3 quiz questions, a boxes-and-labels sketch of the page.
- **Block 2:** set up the editor (VS Code or similar). Create the folder `my-showcase/` and `index.html`. Type the basic template together and open it in the browser:
  ```html
  ```

My Topic

```

- **Test, Debug, Show:** save, refresh, show a neighbour. Teach: *save, then refresh, every time.*
- **Checkpoint:** planning sheet + a page showing the topic as a heading.

## Day 2 (Tue): Text and lists

- **Block 1:** headings `h1`–`h3`, paragraphs `p`, lists `ul/li`. Together, write the intro and the 5 cool facts.
- **Block 2:** learners write their own intro and facts. Add comments (`<!-- like this -->`) to label parts of the code.
- **Test, Debug, Show:** typical bug: a missing closing tag. Teach indentation to spot it.
- **Checkpoint:** heading, intro paragraph, list of facts.

## Day 3 (Wed): Containers (`<div>` and `<section>`)

- **Block 1:** big idea: **everything lives in a box.** Draw the page as boxes, then turn each box into a `<div>` or `<section>`.
- **Block 2:** build `header`, a "facts" section, a "quiz" section (empty for now), and a `footer`. Give each a **class** (`class="facts"`, `class="quiz"`). Look at the demo site's source as a model.
- **Test, Debug, Show:** can a partner point to each box on the sketch and find it in the code?
- **Checkpoint:** page divided into named containers.

## Day 4 (Thu): Links, buttons and a second page

- **Block 1:** links (`<a href="about.html">`) and file paths, and buttons. Quick side lesson for your own topic page: `<img src="..." alt="...">` and why `alt` matters. (The demo site has no pictures, so the day-by-day snapshots do not either.)
- **Block 2:** create `about.html` with the quiz buttons that will come alive in Week 3, and link the two pages from the navbar. Optional: add a picture or two to your own topic page (own drawings or free images; check licences).
- **Test, Debug, Show:** broken link? Check the path and file name together.
- **Checkpoint:** a link, buttons, two linked pages.

## Day 5 (Fri): Test, debug and showcase

- **Block 1:** partner **page review**: can someone who didn't build it read the page and understand it? Checklist: closing tags, working links, images show, alt text.
- **Block 2:** fix issues, tidy indentation, add the content you ran out of time for.
- **Final slot:** Friday mini-showcase: each learner shows their page for 1 minute. Tease next week: "now we make it beautiful."
- **Deliverable:** complete, unstyled home page with two linked pages.

---
```

WEEK 2: Make it beautiful (CSS)

Weekly goal: a styled, well-laid-out page that looks like the learner's own brand.

Day 6 (Mon): First styles

- **Warm-up:** show a page with and without CSS, side by side.
- **Block 1:** create `styles.css` and link it: `<link rel="stylesheet" href="styles.css" />`. Pattern: `selector { property: value; }`. Style `body` (background, text colour, font) and headings.
- **Block 2:** learners pick a **colour palette** (3 to 4 colours) on paper, then apply it. Style links and buttons.
- **Test, Debug, Show:** no change appearing? Check the link, the folder and the semicolons.
- **Checkpoint:** colours and fonts applied.

Day 7 (Tue): The box model and spacing

- **Block 1:** every container has **content, padding, border, margin**. Draw it on the board; use **Inspect** on the demo site to see it live.
- **Block 2:** style each container: `padding`, `border`, `border-radius`, `background`. Centre the page with `.container { max-width: 1000px; margin: 0 auto; }`.
- **Test, Debug, Show:** play with values live in Inspect, then copy the good ones back to the file.
- **Checkpoint:** containers look like neat cards.

Day 8 (Wed): Layout with Flexbox

- **Block 1:** `display: flex; gap: 16px;` puts boxes in a row. Build the navbar: title on the left, links on the right (`justify-content: space-between`).
- **Block 2:** make the footer and the quiz area layout using flex. Align and centre things.
- **Test, Debug, Show:** break the layout on purpose and fix it: a great way to learn.
- **Checkpoint:** a working navbar and footer.

Day 9 (Thu): Layout with Grid and cards

- **Block 1:** `display: grid; grid-template-columns: repeat(3, 1fr);`. Turn the cool facts into a **row of 3 cards**, like "How we do it" on the demo site.
- **Block 2:** polish the cards: gradient heading (`background-clip: text`), hover effect (`.card:hover { transform: translateY(-4px); }`).
- **Test, Debug, Show:** compare each learner's cards with the demo; what's different?
- **Checkpoint:** a styled 3-card row.

Day 10 (Fri): Test, debug and showcase

- **Block 1:** design review in pairs: "one thing I love, one thing to improve". Check colours are readable (enough contrast).

- **Block 2:** apply feedback. Style `about.html` with the same `styles.css`.
- **Final slot:** Friday showcase of styled sites.
- **Deliverable:** styled two-page site.

WEEK 3: Responsive and JavaScript

Weekly goal: a responsive site with a working interactive quiz.

Day 11 (Mon): Responsive and final CSS polish

- **Block 1:** shrink the browser window; where does it break? Add one media query together: `css @media (max-width: 700px) { .cards { grid-template-columns: 1fr; } }`
- **Block 2:** polish: spacing, shadows, a hero gradient, a styled quiz box ready for tomorrow.
- **Test, Debug, Show:** test on a phone-sized window and on the big screen.
- **Checkpoint:** page looks good at two widths.

Day 12 (Tue): JavaScript, first steps

- **Warm-up:** "Our page looks great. Now let's make it react."
- **Block 1:** what JS is, where it goes (`<script>` at the bottom of `body`), the **Console** (Inspect → Console) and `console.log("Hello")` . Then **variables** (`let score = 0;`) and a first function:
  ```html
  <button onclick="alert('Hello!')">Click me</button>
  ```

- **Block 2:** learners make a button that shows a fun fact about their topic.
- **Test, Debug, Show:** bug hunt: typical errors are quote marks, brackets and misspelled function names. Read the Console message together.
- **Checkpoint:** a button that reacts when clicked.

Day 13 (Wed): Build the quiz together

- **Block 1:** `if / else` and one quiz question. Build it together:
  ```html
  <button onclick="alert('Correct!')">Tsavo</button>
  ```

- **Block 2:** learners write their own question and 3 answer buttons about their topic.
- **Test, Debug, Show:** partners try each other's quiz; does every answer behave correctly?
- **Checkpoint:** one working quiz question.

Day 14 (Thu): Show results on the page (no more `alert`)

- **Block 1:** change the page from JS: `document.getElementById("result").textContent = "Correct!";` . Add `<p id="result"></p>` in the quiz box and show the message there.
- **Block 2: stretch (optional):** add a **score** variable that goes up on a correct answer and displays on the page, or a second question. The demo site's quiz shows just the message.

- **Test, Debug, Show:** does the score ever go wrong (e.g. double-counting)? Fix it together.
- **Checkpoint:** quiz with on-page feedback.

Day 15 (Fri): Test, debug and showcase

- **Block 1:** swap laptops and play each other's quiz for 10 minutes. Feedback card: 1 compliment, 1 bug, 1 idea.
- **Block 2:** fix bugs, polish text and styles.
- **Final slot:** Friday showcase.
- **Deliverable:** responsive home page with a working quiz.

Stretch for fast finishers: a light/dark theme button using `classList.toggle`, a "next question" button, or a fact that appears when a card is clicked. Keep JS small: a quiz that works is a success.

WEEK 4: Finish, deploy and showcase

Weekly goal: the "Who I am" page, a final test, a live website and a confident showcase among the learners. **Day 18 is the last build day**; Day 19 is the written quiz and class showcase; Day 20 is play day and the end-of-camp party, so learners have time to rest and study for exams.

Day 16 (Mon): "Who I am" and finishing touches

- **Block 1:** open `about.html`. Learners replace every `[ ... ]` with their own words. Rules: **first name only**, no surname, address, phone number or school name; photos only with parent permission. The "Try my quiz" box already contains a quiz; learners change the question, the answers and the correct answer to match their topic.
- **Block 2:** pick **one** finishing touch from the stretch list or the demo site (the tabbed tools showcase, a favicon, hover effects, a closing "Thanks for visiting!" message). Then a content check: all facts true, spelling, images with `alt` text, the About page linked from the navbar on both pages.
- **Test, Debug, Show:** partners proofread each other's writing (spelling, kindness, privacy), then walk through each other's site as a visitor would.
- **Checkpoint:** About page written and linked, one extra feature added, a site the learner is proud of.

Day 17 (Tue): Final test, debug and deploy

- **Block 1:** the **bug hunt**: learner tests every link, button, image and quiz answer, then a partner tests again with fresh eyes. Fix, test, fix.
- **Block 2: deploy**: the teacher publishes each folder to a free host (for example GitHub Pages or Netlify Drop) so every project has a real web address. Keep accounts under the teacher or school, not the children. Open each live link on a different device.
- **Test, Debug, Show:** the live site works exactly like the local one?
- **Checkpoint:** every learner has a live URL.

Ready-to-show checklist

- [ ] All links work (home, how, who I am)
- [ ] Looks good on a narrow window too
- [ ] Quiz works for every answer and shows a message
- [ ] About page is in the learner's own words
- [ ] No private information on any page

Day 18 (Wed): Rehearsal and final polish (last build day)

- **Block 1:** write a 2-minute talk using three prompts: *What I built · What was hardest (and how I fixed it) · What I'm proudest of.* Practise in pairs.
- **Block 2:** last bug fixes only (no new features). Re-test the live link on another device. Back up each folder on a USB drive.
- **Test, Debug, Show:** each learner gives their talk to a partner and gets one compliment and one idea.

- **Deliverable:** a finished project and a rehearsed talk. **Building stops today.**

Day 19 (Thu): Written quiz and class showcase

- **Warm-up:** quiet start; pencils out, laptops closed.
- **Block 1: written quiz** (about 20 minutes, 20 marks) on HTML, CSS, JavaScript and debugging. Most questions come from the LeetCamp site the class built (its navbar, 3-card grid, CSS variables and quiz function); three are general. Print `camp/pdf/day-19-written-quiz.pdf` (one copy per learner); the teacher marks with `camp/pdf/day-19-answer-key.pdf`. The HTML versions sit beside them in `camp/`.
- **Block 2: class showcase** (learners among themselves). Sites open on laptops; each learner presents their 2-minute talk, and classmates give one compliment and one idea. Finish with the badges.
- **After:** save every project folder. These become the exhibits for next year's Tech Showcase.

> **Parents and teachers:** the showcase this week is among the learners. If you still want visitors, the simplest place is the Day 20 party.

Day 20 (Fri): Play day and end-of-camp party

No coding project today. See `camp/day-20-party.html` for the full plan.

- **Warm-up:** music, name cards, "my favourite moment of camp".
- **Block 1: Quiz Battle.** Everyone plays every classmate's quiz; keep a class leaderboard.
- **Block 2:** games and free play (unplugged coding games, Scratch free play, free-build time).
- **Final slot: party.** Certificates, fun awards for every learner (Bug Squasher, Best Design, Most Creative Topic, Helpful Hand), snacks, class photo (only with consent).
- **Before leaving:** back up every project folder to USB and hand out take-home links.

Assessment (simple and encouraging)

Skill	Beginning	Developing	Secure	Wow
HTML structure	Headings and text	Divs with classes	Clean sections and working links	Multi-page, tidy, commented code
CSS	Colours	Cards with padding	Grid/flex layout	Responsive + hover effects
JavaScript	Button shows a message	Quiz with if/else	Result shown on the page	Extra features, e.g. a score or theme toggle
Debugging	Asks for help	Finds bugs with hints	Fixes independently	Helps others debug
Presenting	Shows the screen	Explains what they made	Explains how	Explains a bug they fixed

Materials checklist

- Laptops/Chromebooks with a modern browser and a code editor (VS Code or similar)
- Printed planning sheets, feedback cards and sticky notes
- Teacher demo site (this folder) on the projector
- Parent permission slips for photos and for online publishing
- USB drives for backups; party invitations for parents and teachers (optional, Day 20)

Troubleshooting

- **Page looks unstyled:** check the `<link>` to `styles.css` and that both files are in the same folder.
- **Changes don't show:** save the file, then refresh the browser.
- **Button does nothing:** open the Console; it names the line with the error (usually a typo or missing bracket).
- **Tabs on the demo site don't switch:** they use CSS only; the radio inputs' `id`s and the labels' `for` values must match exactly.
- **A learner falls behind:** each day's checkpoint is the minimum. Give a ready-made starter file so they can rejoin the next day.