

Teacher & Homeschool Parent's Guide

Everything you need to run Galaxy Defender with one child, a homeschool group, a classroom, or a holiday club. ~30 minutes to read end-to-end.

What this guide covers

- Who the course is genuinely for (and who it's not)
- How the 11 lessons are structured
- Three suggested pacing plans (weekly · weekend · holiday-intensive)
- How to do it WITH a kid vs let them self-pace
- Common stumbling blocks per lesson + what to say
- Assessment ideas without making it feel like school
- Frequently asked questions
- How to get help when you're stuck

1. Who Galaxy Defender is for

Galaxy Defender teaches real Python programming through a game kids actually want to build. It's been designed for **ages 7 and up**, but the experience shifts as the age changes:

Ages 7–9 (with adult support)

Sit with them for the first 2–3 lessons. Read the explainers aloud. Help with typing — typos cause 80% of beginner frustration. From Lesson 3 onwards most confident readers can drive solo with light check-ins.

Ages 10–13 (the sweet spot)

This is the target zone. Most kids in this range can run the course independently after a quick 5-minute kick-off. Check in at the end of each lesson, ask them to show you what they built, and watch the pride.

Ages 14–16 (binge-mode)

Older teens often finish in a weekend or two of focused work. Encourage them to go straight to the capstone (Lesson 11) and design something original.

Who it's NOT for: kids who want a video-only learning experience. Galaxy Defender is hands-on — every lesson involves typing code and clicking Run. If your kid won't type, they won't finish. (Counter-tip: pair them with a sibling who WILL type.)

2. How the course is structured

Eleven self-paced lessons. Lessons 0 + 1 are **free**; lessons 2–11 require the course password. Every lesson follows the same predictable rhythm — once your kid knows the pattern, they can navigate by themselves.

Inside every lesson

- **Quick Recap (30 sec)** — 2 collapsible questions reviewing the previous lesson
- **Rocket Ladder** — visual showing where this stage fits in the bigger plan
- **The big idea** — one new concept explained with a story-style analogy
- **Code recipe** — numbered steps for what to type
- **The code so far** — the cumulative Galaxy Defender game grows with each lesson
- **Trinket editor** — in-browser Python, no install needed
- **Mini-challenges** — 3–5 small tweaks to make it your own
- **Handouts** — Code Library PDF + the big-picture infographics

The 11 lessons at a glance

#	Stage	Core idea	Watch for
0	Big Picture	Every game = Import → Setup → Body → End	Skipping straight to Lesson 1
1	Imports	Libraries = pre-written toolboxes	Typos in module names
2	Setup Screen	pygame.init + a 800x600 window	Forgetting pygame.display.flip
3	Colours	(R, G, B) + named constants	Mixing up the colour name and value
4	Player	pygame.Rect + arrow-key movement	Wrong window size in boundary check
5	Bullets	Lists + KEYDOWN events	Indentation inside the event loop
6	Enemies	for-loops + recycling pattern	Off-screen enemies never resetting
7	Score & Font	pygame.font + str(score)	Forgetting str() around the number
8	Game States	MENU / PLAYING / GAME_OVER	Code in wrong elif branch
9	Helpers	def + the global keyword	Forgetting 'global' on score/lives
10	Game Loop	6 phases x 60 FPS	Putting flip BEFORE the drawing
11	Capstone	Design YOUR own game	Trying to build too much at once

3. Three pacing plans (pick what fits your week)

■ ■ Weekly (6–8 weeks · 1–2 hrs/week)

Best for: term-time homeschool, after-school clubs, working parents. Do one lesson per week. Friday afternoons + weekend morning are popular slots. Recap from the previous week takes 30 seconds (we built this in).

■ ■ Weekend Sprint (2 weekends · 4 hrs each)

Best for: motivated 10–14 year olds and parents who want a focused project. Weekend 1: Lessons 0–5 (you'll have the game playing by Sunday lunch). Weekend 2: Lessons 6–10 + Capstone start. Demo at Sunday dinner.

■ ■ Holiday Intensive (5 weekday mornings)

Best for: summer holiday, half-term, holiday clubs, summer camps. 2-hour morning sessions for 5 days:

- **Day 1:** Lessons 0–2 (foundation + first window)
- **Day 2:** Lessons 3–5 (colours + player + bullets)
- **Day 3:** Lessons 6–8 (enemies + score + states)
- **Day 4:** Lessons 9–10 (helpers + the polished game)
- **Day 5:** Capstone — kids design and ship their own game, demo at the end

Pro tip: never end a session mid-lesson if you can help it. Each lesson ends with a runnable game — that's where the dopamine hit happens. Finish the lesson, then pause; *never* pause inside the middle of one.

4. Do it WITH them or let them self-pace?

Both work. Here's how to choose:

Do it WITH them when:

- They're ages 7–9 OR new to typing on a laptop
- It's your first "do something together" tech project
- You're a homeschool parent and this is part of the curriculum
- You want to learn alongside them (no shame — most parents do)

Sit on the same side of the screen. Read the explanations out loud. Let them do the typing even when it's slow — typing IS the learning. Resist the urge to fix their typos; let them find them. That's where the muscle is built.

Let them self-pace when:

- They're 10+ and a confident reader
- They've shown enthusiasm for the demo on the landing page
- You can check in for 5 minutes at the end of each lesson

At the end of each lesson, ask three questions: (1) *What new thing did you build?* (2) *Can you show me?* (3) *What broke and how did you fix it?* Those three questions are 80% of good teaching — and you don't need to know Python to ask them.

5. Common stumbling blocks (and what to say)

"It says SyntaxError"

99% of the time: a missing colon, a missing parenthesis, or wrong indentation. Ask: *"What does the line number say? Read the line above and the line itself out loud — does anything feel off?"*

Don't fix it for them. Have them find it.

"Nothing happens when I press Run"

Usually they forgot `pygame.display.flip()` or the while running: loop is missing. Have them compare line-by-line with the lesson's "code so far" block.

Side-by-side compare is the most powerful debugging tool.

"I want it to be way more complicated"

Great instinct — channel it into the mini-challenges below the code, or save it for Lesson 11's capstone. Don't let them rabbit-hole on Lesson 4.

Capture their idea on paper for later — they'll feel heard.

"This is boring" (rare but happens)

Skip ahead to the Trinket editor and let them play the example game first. Once they've seen the goal, the steps feel more meaningful.

Outcome before process. Always works.

"I broke it and can't fix it"

Show them how to copy the lesson's "code so far" block back in — it's a reset button. They lose their tweaks but the game works again.

Make resetting feel normal, not failing.

6. Assessment without it feeling like school

Ditch quizzes. The course has built-in Quick Recap questions at the start of each lesson — those handle pure recall. For deeper assessment, use these:

Show-and-tell at the end of each lesson

Ask the kid to **play their game** and explain what each part does. Can they point at a line of code and tell you what it does? That's the proof.

The remix challenge

After Lesson 7, give them a 10-minute remix: "Change three things and show me a new screenshot." Different colours, different speeds, different sprites — all valid. Creative tweaks prove they understand the levers.

The capstone is the real assessment

Lesson 11 asks them to design and ship an original game. The Game Design Document is the spec. The shipped game is the proof. Mark it on 4 dimensions: **(1) runs without crashing, (2) original design, (3) win + lose conditions clear, (4) polish** (sprites, colours, sound).

The Galaxy Defender Pilot Certificate (also included) is a nice ceremonial moment. Print it on thicker paper. Frame it. Take a photo. The ritual matters as much as the test.

7. FAQ

Q. Do we need to install Python?

A. Nope. Everything runs in the browser via Trinket. Any laptop with a modern browser works (Chromebook, Mac, Windows, Linux).

Q. Will my kid actually finish?

A. If they get through Lessons 0+1 (free), they almost always finish. Those two lessons are the deciding-to-buy moment AND the deciding-to-finish moment.

Q. My kid finishes a lesson too fast — what now?

A. Open the Code Library PDF for that lesson. Each one has 5–6 bonus mini-games using the same idea — that's where the depth lives.

Q. Can siblings share one purchase?

A. Yes — the password is for your household. We just ask that you don't share publicly. Each kid can save their own progress (different browser profiles or different laptops).

Q. What if my kid is 7 and a reluctant reader?

A. Sit with them. Read the explainers aloud. Type for them on the harder lessons (1, 5, 6) — they'll watch you and start helping. By Lesson 4 most kids want to take the keyboard back.

Q. Are there refunds?

A. Yes — try the free Lessons 0+1 first. If you've paid and the course isn't a fit within 14 days, email me and I'll refund you, no questions asked.

Q. Can I use this in my classroom?

A. Absolutely. Print the Code Libraries and the Big Picture / Galaxy Map / Rocket Ladder infographics. Each lesson is roughly one 45–60 minute class slot. The Capstone (Lesson 11) is a 2-week project.

Q. What comes after Galaxy Defender?

A. Lesson 11 (the Capstone) ends with a "Where to go next" section pointing to real local pygame, pygame-zero, the arcade library, game jams, and GitHub Pages. They'll be ready for any of those.

8. Where to get help

Inside the course

- Every lesson HTML page has a printable handout option (use Ctrl+P / Cmd+P)
- Code Library PDFs ship with 5–6 worked examples per lesson
- The Sprite Bonus PDF helps with visual polish
- The Capstone Starter Pack has a paste-and-go skeleton + 4 alternative sprite designs

Direct support

Email nextgencoders@jestechsolutions.co.uk any time. Reply within 24 hours. Send a screenshot of the error or a link to the kid's Trinket — we'll usually have a fix back the same day.

If you're using this in a classroom or club

Drop me an email — happy to set up a single bulk-licence (instead of one password per household) and a teacher-only Slack channel for live support during your sessions.

Thank you for choosing Galaxy Defender. The most important thing I've learned running this course: **kids don't need perfect teachers — they need an adult who's curious WITH them.** If you're willing to say "I don't know, let's find out together," you've already got the most important teaching skill. The course handles the rest. ■

— Goodness Wariboko
Director, NextGen Coders Lab