PRESENTATION.md — 15-Minute Stand-Up Demo Notes for Teachers

Audience: Intro CS/high-school students or bootcamp learners Format: No slides. Live terminal + editor. Hand out printed STUDENTS.md. Goal: Show what Basil is, run a couple examples, and briefly demystify the Rust project behind it. Last updated: 2025-10-06

---

0) What to prepare (5–10 minutes before class)

- Print enough copies of STUDENTS.md (one per student or per pair).
- Open this repository in your editor (RustRover/VS Code/etc.).
- Open a terminal in the project root.
- Ensure Rust and Cargo are installed and builds succeed:
  - Windows PowerShell:
    - cargo --version
    - rustc --version
    - cargo run -q -p basilc -- run examples\hello.basil
- Optional: Keep README.md open in a second panel for reference.

Tip: If a demo example fails, switch to the known-good ones:

- examples\hello.basil (Hello World)
- examples\input.basil (reads input)
- examples\bigtest.basil (kitchen-sink of features)

---

1) Agenda at a glance (timeboxed)

- 0:00–1:00 — Welcome, what you'll get today
- 1:00–3:00 — What is Basil? Why BASIC-style?
- 3:00–6:00 — Run Hello World; tiny language tour
- 6:00–9:00 — Under the hood: the Rust project (lexer → parser → compiler → VM)
- 9:00–12:00 — Web templating (CLI vs CGI), caching, error messages
- 12:00–14:00 — Where to learn more (repo, docs, examples)
- 14:00–15:00 — Q&A

---

2) Scripted walkthrough (with talk-track and commands)

2.1 Welcome (0:00–1:00)

Talk-track:

- "You're getting a printed copy of STUDENTS.md — a gentle, hands-on overview of how Basil runs your code."
- "For a deeper dive, everything is in this GitHub repository: README.md for overview, TECHNICAL.md for internals, plus examples."
- "We'll do a quick 15-minute tour: run a Basil program, peek inside the Rust engine, and see how it can also render web pages."

Hold up STUDENTS.md and point to the sections on the compiler/VM mental model.

2.2 What is Basil? (1:00–3:00)

Talk-track:

- "Basil is a tiny, modern BASIC-style language. It's simple to read and perfect for teaching and scripting."
- "It compiles your Basil source to bytecode, then runs it on a small virtual machine (VM)."
- "It's written in Rust, open-source, cross-platform, and designed to be AI-friendly — clear, consistent, and easy to refactor."
- "You can run Basil from the command line, or embed it in web pages via CGI-style templating."

Point out these docs later in the repo: BASIL.md (language), README.md (overview), TECHNICAL.md (deep internals), WEB_PAGES.md/WASM.md (advanced).

2.3 Run Hello World (3:00–6:00)

In terminal (Windows PowerShell):

- cargo run -q -p basilc -- run examples\hello.basil

Explain as it runs:

- The basilc CLI reads the .basil file.
- Lexer → Parser → Compiler → Bytecode → VM executes.
- The first run caches compiled bytecode next to your source as a .basilx file so future runs can be instant.

Show another quick example (optional):

- cargo run -q -p basilc -- run examples\input.basil
- cargo run -q -p basilc -- lex examples\hello.basil # shows tokens (debug)

Teacher cues:

- If input is requested, type a short response to show interactive behavior.
- If anyone asks about speed: caching and the lightweight VM keep startup fast.

2.4 How the Rust project is organized (6:00–9:00)

Talk-track:

- "This repo is a Rust workspace. The big pieces are:"
  - basilc — the CLI (entry point for run/test/lex and CGI mode)
  - basilcore — the language engine in crates: lexer, parser, compiler, bytecode, vm
  - basil-objects — optional object/class helpers and examples
- "The pipeline you just used was: source → tokens (lexer) → AST (parser) → bytecode (compiler) → executed by a VM."
- "When we say 'bytecode', think 'small instructions for a tiny imaginary computer'."

Teacher cues:

- Briefly click through basilc\src\main.rs and the basilcore{lexer,parser,compiler,vm}\src directories so students see real code exists without diving deep.

2.5 Web templating and CGI (9:00–12:00)

Talk-track:

- "Basil can also render web pages by mixing HTML and Basil in one file — similar to classic PHP."
- "The CLI detects template markers and precompiles them to Basil before compiling to bytecode."
- "In server environments, a CGI handler can invoke basilc to render pages on demand."

Optional local demo ideas (pick one):

- Show examples\cgi.basil in the editor; point out the mix of HTML and Basil tags.
- For advanced audiences, show how environment variables (REQUEST_METHOD, SCRIPT_FILENAME) are used (see README.md and WASM.md for commands).

2.6 Where to go next (12:00–14:00)

Point students to:

- This GitHub repository for a deeper dive (read it online later; links and examples are all here).
- STUDENTS.md they're holding now for the big-picture mental model.
- README.md for a guided tour, installation and example commands.
- BASIL.md for language overview and idioms.
- TECHNICAL.md for internals (AST, bytecode format, VM), plus error handling and caching details.
- WEB_PAGES.md and WASM.md for web/CGI and WASI/wasmtime usage.
- The examples\ folder for runnable programs (hello, input, classes, arrays, loops, and a big combined demo in bigtest.basil).

2.7 Q&A (14:00–15:00)

Suggested prompts if the room is quiet:

- "What parts of BASIC feel familiar or surprising?"
- "Which is clearer — the STUDENTS.md mental model or reading code?"
- "Do you want to see how an error is reported? (We can break a line and re-run.)"

---

3) Quick reference (CLI)

- Show help:
  - cargo run -p basilc --
  - or run with no args to see commands (init/run/test/lex, plus fun aliases like sprout/chop).
- Run a program:
  - cargo run -q -p basilc -- run examples\hello.basil
- Token dump (debug):
  - cargo run -q -p basilc -- lex examples\hello.basil
- Test mode (auto-mock input):
  - cargo run -q -p basilc -- test examples\input.basil

Notes:

- Windows paths use backslashes (examples\hello.basil). On macOS/Linux, use forward slashes (examples/hello.basil).
- First run writes a .basilx cache next to your .basil file when possible.

---

4) Troubleshooting in front of a class

- Build fails? Run: cargo clean; then cargo build -p basilc.
- Example path wrong? Tab-complete to confirm the file exists in examples.
- Terminal weirdness? Close and reopen a fresh PowerShell.
- CGI demo too advanced for the room? Just scroll through examples\cgi.basil and describe it.

5) One-liner summary to close

"Basil is a small, readable BASIC-style language implemented in Rust. You write simple Basil, it becomes bytecode, and a tiny VM runs it. You can script from the command line or render web pages. Grab STUDENTS.md for the mental model, and explore the GitHub repo later for all the details and examples."