# Why a containerized toolchain

A **container** is a self-contained bundle of software that runs the same way on every computer, whether that computer runs Windows, macOS or Linux, so you don't have to install each program separately. You can think of it as one pre-built package that holds an entire toolchain, ready to run. Strictly speaking, the package you download is called an **image**, and a container is a copy of that image running on your computer. You download the image once, and after that you can remove a container and start a new one from the same image without downloading anything, as you'll do in Lab 03.

Open-source ASIC flows use dozens of tools: schematic capture, SPICE, layout, DRC/LVS, synthesis, and place and route (P&R). Installing each of them natively on Windows, macOS and Linux is fragile, because tool versions drift from one machine to the next and the PDKs on different machines disagree, until sorting out "works on my machine" becomes the course.

## What the team standardizes

The team uses the IIC-OSIC-TOOLS Docker image, which IIC-JKU maintains. The team's `workspace` repo wraps that image with:

- launch scripts for noVNC (a Linux desktop in your browser) and, optionally, X11
- a pinned SKY130 PDK (`pdk/volare.lock`)
- shared shell helpers (`common/.designinit`)
- a starter folder for each course, under `modules/`

Apart from git, the only thing you install yourself is Docker. Everything else runs inside one container, and the `workspace` folder on your laptop appears inside that container at `/foss/designs`. noVNC shows the container's desktop in Chrome, Firefox or Edge, so you can open terminals, XSchem, Magic and the other tools without installing any EDA package on your own operating system. The workbench desktop explains how that desktop starts.

## What IC101 doesn't cover

- It isn't a theory-heavy introduction to semiconductors; later courses cover that.
- It isn't a substitute for the ECE curriculum.
- It doesn't give each course its own Docker image, because one workspace serves every tool-heavy course.

Next: What Docker needs.