What is a programming language?

- Here are separable concepts for defining and learning a language:
  - syntax: how do you write the various parts of the language?
  - semantics: what do programs mean? (One way to answer: what are the evaluation rules?)
  - paradigm: what is the primary approach to structuring code?
  - idioms: how do you typically use the language to express computations?
  - libraries: does the language provide “standard” facilities such as file-access, hashtables, etc.? How?
  - tools: what is available for manipulating programs in the language? (e.g., compiler, debugger, REP-loop)

OCaml basics

- A program is a sequence of *bindings*
- One kind of binding is a *variable binding*

  `let x = e ;;` (semicolons optional in a file)

- A program is evaluated by evaluating the bindings in order
- A *variable binding* is evaluated by:
  - Evaluating the expression in the environment created by the previous bindings. This produces a *value*.
  - Extending the (top-level) environment to bind the variable to the value.
- Examples of values: 13, 4.8, true, “hello”, [3; 4; 5], (8, 8.2)