composing programs

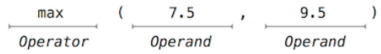
part 1

- programs must be written for people to read, and only incidentally for machines to execute.

When one describes a language, one must pay close attention to the means that the language provides for combining simple ideas to form more complex ideas. Every powerful language has three such mechanisms :

- **primitive expressions and statements**, which represent the simplest building blocks that the language provides,
- **means of combination**, by which complex elements are built from simpler ones, and
- **means of abstraction**, by which compound elements can be named and manipulated as units
- **Test incrementally**: Every well-written program is composed of small, modular components that can be tested individually. Try out everything you write as soon as possible to identify problems early and gain confidence in your components.
- **Isolate errors**: An error in the output of a statement can typically be attributed to a particular modular component. When trying to diagnose a problem, trace the error to the smallest fragment of code you can before trying to correct it.
- **Check your assumptions**: Interpreters do carry out your instructions to the letter — no more and no less. Their output is unexpected when the behavior of some code does not match what the programmer believes (or assumes) that behavior to be. Know your assumptions, then focus your debugging effort on verifying that your assumptions actually hold.
- **Consult others**: You are not alone! If you don't understand an error message, ask a friend, instructor, or search engine. If you have isolated an error, but can't figure out how to correct it, ask someone else to take a look. A lot of valuable programming knowledge is shared in the process of group problem solving.
- Programs serve to communicate those ideas among the members of a programming community. Thus, programs must be written for people to read, and only incidentally for machines to execute.
- **primitive expressions and statements**, which represent the simplest building blocks that the language provides

- **means of combination**, by which compound elements are built from simpler ones, and
- **means of abstraction**, by which compound elements can be named and manipulated as units.



The operator specifies a *function*. When this call expression is evaluated, we say that the function `max` is *called* with *arguments* 7.5 and 9.5, and *returns* a *value* of 9.5.

- An important role for you as a programmer is to structure expressions so that they remain interpretable by yourself, your programming partners, and other people who may read your expressions in the future.
- python has a long list of modules which in them have a list of methods that one can use