1.1 The Python programming language

The [ ] language you will learn is Python. Python is an example of a **high-level language**; other high-level languages you might have heard of are C, C++, Perl, and Java.

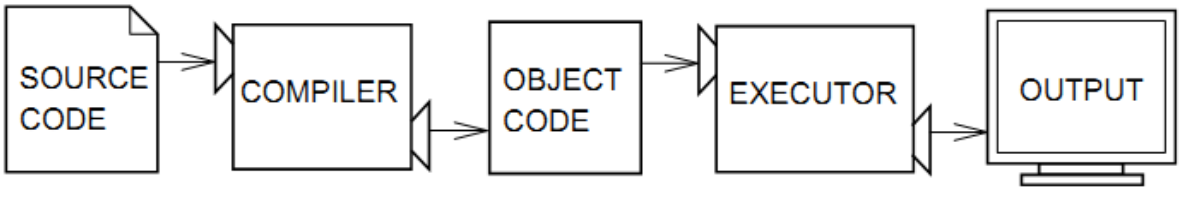
There are also **low-level** [ ], sometimes referred to as "machine languages" or "assembly languages." Loosely speaking, computers can only execute programs written in low-level languages. So programs written in a high-level language have to be [ ] before they can run. This extra processing takes some time, which is a small disadvantage of high-level languages.

The advantages are enormous. First, it is much [ ] to program in a high-level language. Programs written in a high-level language take less time to write, they are shorter and easier to read, they are more likely to be correct, and are easier to maintain. Second, high-level languages are **portable**, meaning that they can run on [ ] kinds of computers with few or no modifications. Low-level programs can run on only one kind of computer and have to be rewritten to run on another.

Due to these advantages, almost all programs are written in high-level languages. Low-level languages are used only for a few specialized applications.

Two kinds of programs [ ] high-level languages into low-level languages: **interpreters** and **compilers**. An interpreter reads a high-level program and executes it, meaning that it does what the program says. It processes the program a little at a time, alternately [ ] ines and performing computations.

A compiler reads the program and [ ] it completely before the program starts running. In this context, the high-level program is called the **source code**, and the translated program is called the **object code** or the **executable**. Once a program is compiled, you can [ ] it repeatedly without further translation.



Python is considered an [ ] language because Python programs are executed by an interpreter. There are two ways to use the interpreter: **interactive mode** and **script mode**. In interactive mode, you type Python programs and the interpreter prints the result:

```
>>> 1 + 1
2
```

executes - translates - reading - different - processed - languages - easier - programming - interpreter - process