

- ✔ Text in a program is referred to as a *string*. For example, “la-de-da” is a string of text. The string is enclosed by double quotes.
- ✔ A C language function starts with the function type, such as `int`, and then the function name and parentheses, as in `main()`. Then come a set of curly braces, `{` and `}`. Everything between the `{` and `}` is part of the function.
- ✔ The C language is composed of keywords that appear in statements. The statements end in semicolons, just as sentences in English end in periods. (Don’t frazzle your wires over memorizing this right yet.)

The C Language Itself — the Keywords

The C language is really rather brief. C has only 32 *keywords*. If only French were that easy! Table 3-1 shows the keywords that make up the C language.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Not bad, eh? But these aren’t all the words you use when writing programs in C. Other words or instructions are called *functions*. These include jewels like `printf()` and several dozen other common functions that assist the basic C language keywords in creating programs.

Beyond keywords, programming languages (like human languages) also involve grammar, or properly sticking together the words so that understandable ideas are conveyed. This concept is completely beyond the grasp of the modern legal community.