SWI-Prolog format/2	Accepts	Alternatively accepts	Place holder	Variation	Printing in current locale	Description
Skipping an argument	Anything		~i			Skip argument
Text printing	Atom	Numbers, Strings (but not CharList, CodeList)	~a			Prints an atom
	String	Atoms, CharList, CodeList (but not Numbers)	~s			Prints a string
			~n			Newline
			~N			Newline, but suppressed if newline printed just before (not properly implemented yet)
			~~			Prints the ~
Character printing	Integer (interpreted as unicode char point)		~c			Prints the corresponding unicode character
Integer printing	Integer		~d	~D	~:d	Prints integer interpreted as a decimal number; With "D", the number is printed in english local formatting (using a comma thousands separator)
			~*X*d	~*X*D	~*X*:d	Prints integer interpreted as a decimal number * 10^-X (i.e. the decimal dot, printed a ".", will appear X positions from the right) With "D", the number is printed in english local formatting (using a comma thousands separator)
			~I	~*X*I		Print with underscore digit grouping, by default in groups of 3 X can be used to set the group width.
			~*X*r		~*X*:r	Radix numeric conversion (e.g. for printing hexadecimal: ~16r) X is an integer between 2 and 36 ":" may activate locale-specific grouping of the output digits
Float printing	Float	Integer Non-Integer Rational	~e	~E	~:e ~:E	Print a float in exponential notation, either with small "e" or capital "E". The precision is 6 decimal digits.
			~*X*e	~*X*E	~*X*:e ~*X*:E	As above, but the precision is given by X.
			~f		~:f	Float in non-exponential notation
			~*X*f		~*X*:f	Float in non-exponential notation, but the precision is given by X.
			~g	~G	~:g ~:G	Selects ~e/~E or ~f/~F depending on which yields the shortest string
			~*X*g	~*X*G	~:*X*g ~:*X*G	As above, but the precision is given by X.
Punting the problem to other predicates	Any term		~k			Call **write_canonical/1** with the next argument. "Write Term on the current output stream using standard parenthesised prefix notation (i.e., ignoring operator declarations)."
			~p			Call **print/1** with the next argument. "Print a term for debugging purposes" As the format of "debugging purposes" may change, this is not well defined.
			~q			Call **writeq/1** with the next argument. "Write Term to the current output, using brackets and operators where appropriate. Atoms that need quotes are quoted." (serialization, term can be read back using read/1)
			~w			Call **write/1** with the next argument. "Write Term to the current output, using brackets and operators where appropriate."
	Any term + Option list		~W			Call **write_term/2** with the next 2 arguments, the 2nd argument being options. "The predicate write_term/2 is the generic form of all Prolog term-write predicates."
Calling a goal	Term that can be called as a goal		~@			Call the argument as a goal; output generated on "current_output" by that call is inserted (as string)
Whitespace distribution			~t	~*X*t		Tabstop filler. Wherever it appears, it grabs part of the space divided up between all the ~t found before the next tabstop. By default, 0x20 is used for filling, but for ~Xt, X gives the (decimal) character code (unicode code point) of the character to be used instead.
			~|	~*X*|		Tabstop setter. Either set the tabstop at the current position or at absolute position X (i.e. after the character at position X counted from 0).
				~*X*+		Set tab stop relative to previous stop.