Table 1: Predefined languages. Note that some definitions are preliminary, for example HTML and XML. Each underlined dialect is default dialect

ABAP (R/2 4.3, R/2 5.0, R/3 3.1, R/3 4.6C, R/3 6.10)	
ACSL	Ada (83, 95)
Algol (60, 68)	Assembler (x86masm)
Basic (Visual)	C (ANSI, Objective, Sharp)
C++ (ANSI, GNU, ISO, Visual)	Caml (light, Objective)
Clean	Cobol (1974, 1985, ibm)
Comal 80	csh
Delphi	Eiffel
Elan	Euphoria
Fortran (77, 90, 95)	Haskell
HTML	IDL (empty, CORBA)
Java	ksh
Lisp (empty, Auto)	Logo
make (empty, gnu)	Mathematica (1.0, 3.0)
Matlab	Mercury
Miranda	ML
Modula-2	NASTRAN
Oberon-2	OCL (decorative, OMG)
Octave	Pascal (Borland6, Standard, XSC)
Perl	PHP
PL/I	POV
Prolog	Python
R	S (empty, PLUS)
SAS	SHELXL
Simula (67, CII, DEC, IBM)	SQL
tcl (empty, tk)	
TeX (AlLaTeX, common, LaTeX, plain, primitive)	
VBScript	VHDL (empty, AMS)
VRML (97)	XML

2.5 Special characters

Tabulators You might get unexpected output if your sources contain tabulators. The package assumes tabulator stops at columns 9, 17, 25, 33, and so on. This is predefined via `tabsize=8`. If you change the eight to the number n, you will get tabulator stops at columns $n+1, 2n+1, 3n+1$, and so on.

```
123456789
  { one tabulator }
    { two tabs }
123   { 123 + two tabs }
```

```
\lstset{tabsize=2}
\begin{lstlisting}
123456789
    { one tabulator }
        { two tabs }
123     { 123 + two tabs }
\end{lstlisting}
```

For better illustration, the left-hand side uses `tabsize=2` but the verbatim code `tabsize=4`. Note that `\lstset` modifies the values for all following listings in the same environment or group. This is no problem here since the examples are typeset inside minipages. If you want to change settings for a single listing, use the optional argument.