LaTeX and Friends Algorithms

http://csweb.ucc.ie/~dongen/LAF/LAF.html

M. R. C. van Dongen

UCC

Loading the Package

- Use `algorithm2e` [Fiorio 2004] for pseudo code algorithms.
- Choosing right option saves time/space.
- An important option is `algo2e`.
  - Renames `algorithm` to `algorithm2e`.

LaTeX Usage

```
\usepackage[algo2e]{algorithm2e}
```

- Several options affect the appearance of the algorithms.

`noline` Omits drawing of vertical lines.
`lined` Vertical line marks the block. Keeps end keywords.
`vlined` Vertical “bent” line marks the block.

Effect of Using the Different Options

LaTeX Output

if ⟨*cond*⟩ **then**
    ⟨stuff⟩
end

LaTeX Output

if ⟨*cond*⟩ **then**
  | ⟨stuff⟩
end

LaTeX Output

if ⟨*cond*⟩ **then**
  ⌊ ⟨stuff⟩

Basic Environments

`algorithm` Typesets its body as an algorithm.

`algorithm*` Typesets body as algorithm in 2-column document.

`procedure` Typesets its body as a procedure.

- Caption lists Procedure ⟨`name`⟩.
- `\caption` starts with ⟨`name`⟩`(` ⟨`arguments`⟩ `)`.

`procedure*` Typesets body as procedure in 2-column document.

`function` Typesets its body as a function.

`function*` Typesets body as function in 2-column document.

The algorithm2e Package

LaTeX Usage

```
\begin{algorithm2e}[H]
\KwIn{Integers $a \geq 0$ and $b \geq 0$}
\KwOut{\textsc{Gcd} of $a$ and $b$}
\While{$b \neq 0$}{
    $r \leftarrow a \bmod b$\;
    $a \leftarrow b$\;
    $b \leftarrow r$\;
}
\caption{Euclidean Algorithm}
\end{algorithm2e}
```

LaTeX Output

Input: Integers $a \geq 0$ and $b \geq 0$
Output: Gcd of a and b
while $b \neq 0$ **do**
    $r \leftarrow a \bmod b$;
    $a \leftarrow b$;
    $b \leftarrow r$;

Algorithm 1: Euclidean Algorithm

Describing Input and Output

```
\KwIn{⟨input⟩}
```
⟨In Label⟩: ⟨input⟩

```
\KwOut{⟨output⟩}
```
⟨Out Label⟩: ⟨output⟩

```
\KwData{⟨data⟩}
```
⟨Data Label⟩: ⟨data⟩

```
\KwResult{⟨output⟩}
```
⟨Result Label⟩: ⟨output⟩

```
\KwRet{⟨return value⟩}
```
⟨Ret Label⟩: ⟨return value⟩

Conditional Statements

LaTeX Input

```
\If{⟨condition⟩}
   {⟨clause⟩}
```

LaTeX Output

if ⟨*condition*⟩ **then**
  ⟨clause⟩

Conditional Statements

LaTeX Input

```
\uIf{⟨condition⟩}
    {⟨clause⟩}
```

LaTeX Output

if ⟨*condition*⟩ **then**
  | ⟨clause⟩

Conditional Statements

LaTeX Input

```
\ElseIf{⟨condition⟩}
       {⟨clause⟩}
```

LaTeX Output

else if ⟨*condition*⟩ **then**
  ⟨clause⟩

Example

LaTeX Input

```
\begin{algorithm2e}[tbp]
\uIf{$a < 0$}{
    \tcp{$a < 0$}
} \uElseIf{$a = 0$}{
    \tcp{$a = 0$}
} \lElse\eIf{$a = 1$}{
    \tcp{$a = 1$}
} {
    \tcp{$a > 1$}
}
\end{algorithm2e}
```

LaTeX Output

if $a < 0$ **then**
  // $a < 0$
else if $a = 0$ **then**
  // $a = 0$
else if $a = 1$ **then**
  // $a = 1$
else
  // $a > 1$

The Switch Statement

LaTeX Input

```
\Switch{⟨value⟩}
       {⟨cases⟩}
```

LaTeX Output

switch ⟨*value*⟩ **do**
  ⟨cases⟩

The Switch Statement (Continued)

LaTeX Input

```
\uCase{⟨condition⟩}
      {⟨statements⟩}
```

LaTeX Output

case ⟨*condition*⟩
  | ⟨statements⟩

The Switch Statement (Continued)

LaTeX Input

```
\Other{⟨statements⟩}
```

LaTeX Output

otherwise
  ⟨statements⟩

Example

LaTeX Input

```
\begin{algorithm2e}[tbp]
\Switch{order}{
    \uCase{bloody mary}{
        Add tomato juice\;
        Add vodka\;
        break\;
    }
    \uCase{hot whiskey}{
        Add whiskey\;
        Add hot water\;
        Add lemon and cloves\;
        Add sugar or honey to taste\;
        break\;
    }
    \Other{Serve water\;}
}
\end{algorithm2e}
```

LaTeX Output

switch *order* **do**
- **case** *bloody mary*
  - Add tomato juice;
  - Add vodka;
  - break;
- **case** *hot whiskey*
  - Add whiskey;
  - Add hot water;
  - Add lemon and cloves;
  - Add sugar or honey to taste;
  - break;
- **otherwise**
  - Serve water;

Iterative Statements

LaTeX Input

```
\For{⟨condition⟩}
    {⟨body⟩}
```

LaTeX Output

for ⟨*condition*⟩ **do**
  ⟨body⟩

Iterative Statements (Continued)

LaTeX Input

```
\ForEach{⟨condition⟩} {⟨body⟩}
```

LaTeX Output

foreach ⟨*condition*⟩ **do**
  ⟨body⟩

Iterative Statements (Continued)

LaTeX Input

```
\While{⟨condition⟩} {⟨body⟩}
```

LaTeX Output

while ⟨*condition*⟩ **do**
  ⟨body⟩

Comments

LaTeX Input

```
\tcp{⟨line one⟩}\\
    {⟨line two⟩}
```

LaTeX Output

```
// ⟨line one⟩
// ⟨line two⟩
```

Comments (Continued)

LaTeX Input

```
⟨statement⟩
  \tcp*{⟨comment⟩}
```

LaTeX Output

⟨statement⟩ ; // ⟨comment⟩

Comments (Continued)

LaTeX Input

```
\If(\tcp*[h]{⟨comment⟩})
   {⟨condition⟩}
   {⟨statement⟩}
```

LaTeX Output

if ⟨*condition*⟩ **then** // ⟨comment⟩
  ⟨statement⟩

The listings Package

- Typeset listings with listings [Heinz, and Moses 2007].
- Support for several languages:
  - ANSI C, and ANSI C++,
  - Eiffel,
  - HTML,
  - Java,
  - PHP,
  - Python,
  - LaTeX, and
  - XML.
- Supports different styles for keywords and identifiers.
- Two methods for specifying a listing:

  environment For specifying a listing.
  command For creating a listing from a file.
- Provides command to specify new defaults.
- Result is typeset in place or as a float.
- Provides command for typesetting list of listings.

First Example

LaTeX Input

```
\begin{lstlisting}[language=Java
                   ,gobble=3
                   ,numbers=left
                   ,firstline=2
                   ,lastline=4
                   ,firstnumber=2
                   ,caption=Hello World.
                   ,label=example]
public class Greetings {
    public static void main( String[] args ) {
        System.out.println( "Hello world!" );
    }
}
\end{lstlisting}
```

First Example (Continued)

LaTeX Output

```
2 ␣public static void main( String[] args ) {
3 ␣    System.out.println( "Hello world!" );
4 ␣}
```

Listing 1. Hello world.

Specifying Defaults

LaTeX Usage

```
\lstset{language=Java%
        ,keywordstyle=\bfseries\ttfamily%
        ,stringstyle=\ttfamily%
        ,identifierstyle=\ttfamily\itshape%
        ,showspaces=false%
        ,showstringspaces=true%
        ,numbers=left%
        ,float%
        ,floatplacement=tbp%
        ,captionpos=b}
```

Bibliography


Fiorio, Christophe [14th Dec. 2004]. *algorithm2e.sty—package for algorithms.* Version 3.3.


Heinz, Carsten, and Brooks Moses [22nd Feb. 2007]. *The Listings Package.* Version 1.4.

Acronyms and Abbreviations

AMS	American Mathematical Society
API	Application Programming Interface
APL	A Programming Language
CTAN	Comprehensive TEX Archive Network
CD	Compact Disk
FAQ	Frequently Asked Question
GUI	Graphical User Interface
IDE	Integrated Development Environment
ISBN	International Standard Book Number
OS	Operating System
SI	Système International d'Unités/International System of Units
TUG	TEX Users Group
URL	Uniform Resource Locator
WYSIWYG	What You See Is What You Get

About this Document

- This document was created with `pdflatex`.
- The LaTeX document class is `beamer`.