WEB/CWEB

Julian Lang

Outline

Literate Programming

- WEB was created by Donald E. Knuth.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.
- In literate programming a program should be like a literaric work.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.
- In literate programming a program should be like a literaric work.
- A program should be written to be easily understandable for humans.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.
- In literate programming a program should be like a literaric work.
- A program should be written to be easily understandable for humans.
- The order of a program should be in the order of the programmers thoughts.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.
- In literate programming a program should be like a literaric work.
- A program should be written to be easily understandable for humans.
- The order of a program should be in the order of the programmers thoughts.
- Documentation and code of a program in one file.

Figure : Donald Knuth

Literate Programming

- WEB was created by Donald E. Knuth.
- WEB was the first literate programming language.
- In literate programming a program should be like a literaric work.
- A program should be written to be easily understandable for humans.
- The order of a program should be in the order of the programmers thoughts.
- Documentation and code of a program in one file.

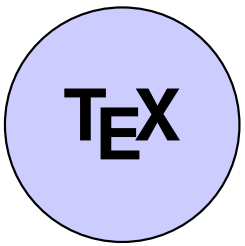
Figure : Donald Knuth

Donald E. Knuth:

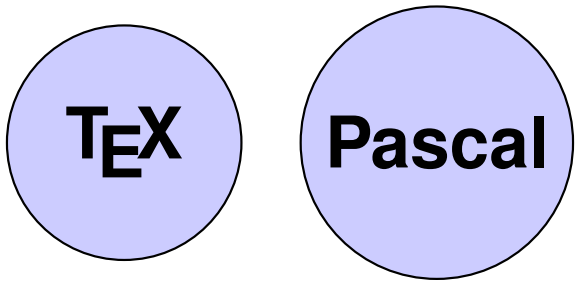
"I am enormously biased about Literate Programming, which is surely the greatest thing since sliced bread."

Design of WEB

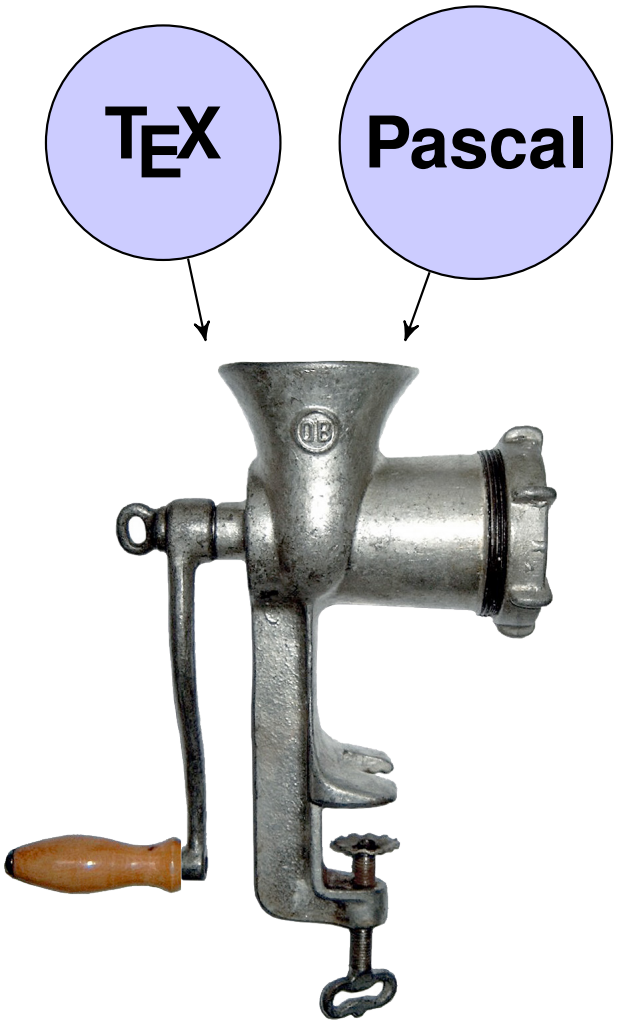
Design of WEB



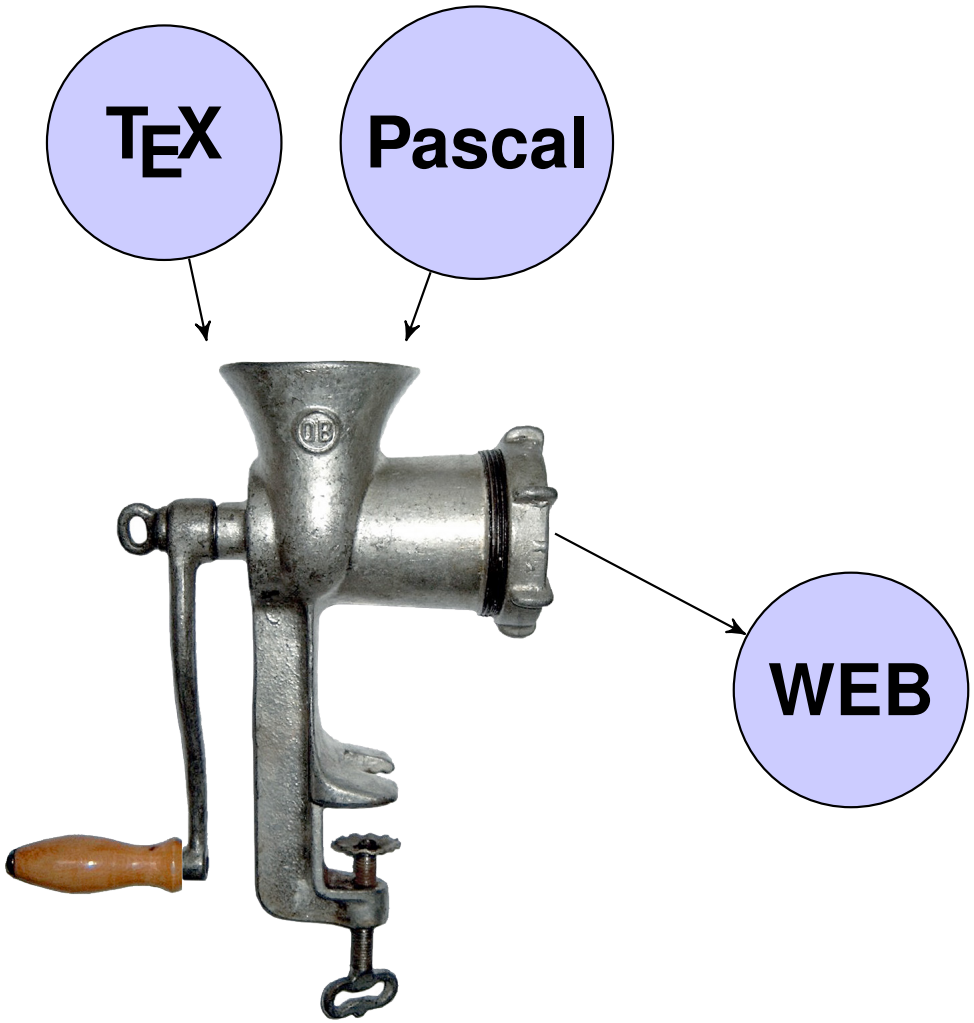
Design of WEB



Design of WEB



Design of WEB



What is WEB?

- The name WEB described good a complex program out of many interconnected pieces.

What is WEB?

- The name WEB described good a complex program out of many interconnected pieces.
- WEB uses Pascal to write the code and TEX to document it.

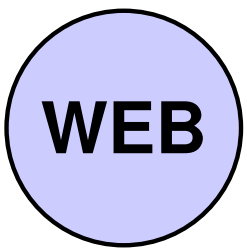
What is WEB?

- The name WEB described good a complex program out of many interconnected pieces.
- WEB uses Pascal to write the code and TEX to document it.
- WEB was invented to help people to write better programs and to get a better documentation.

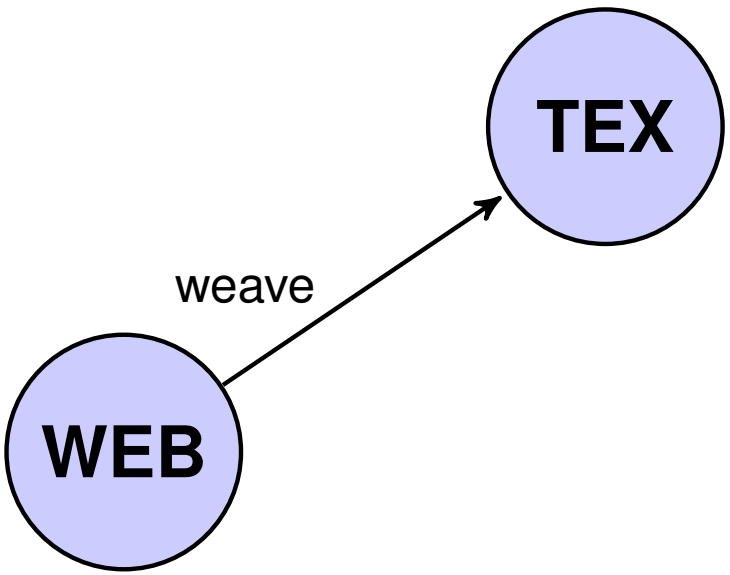
What is WEB?

- The name WEB described good a complex program out of many interconnected pieces.
- WEB uses Pascal to write the code and TEX to document it.
- WEB was invented to help people to write better programs and to get a better documentation.
- The famous programs TEX and Metafont are written in WEB.

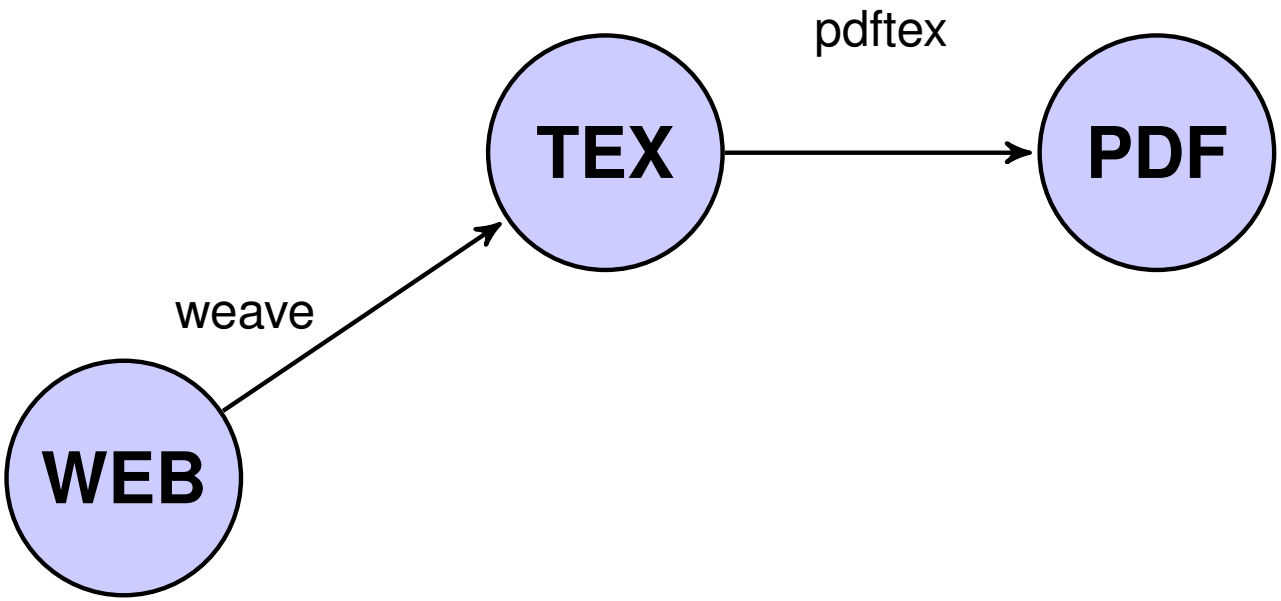
The compilation of a WEB file



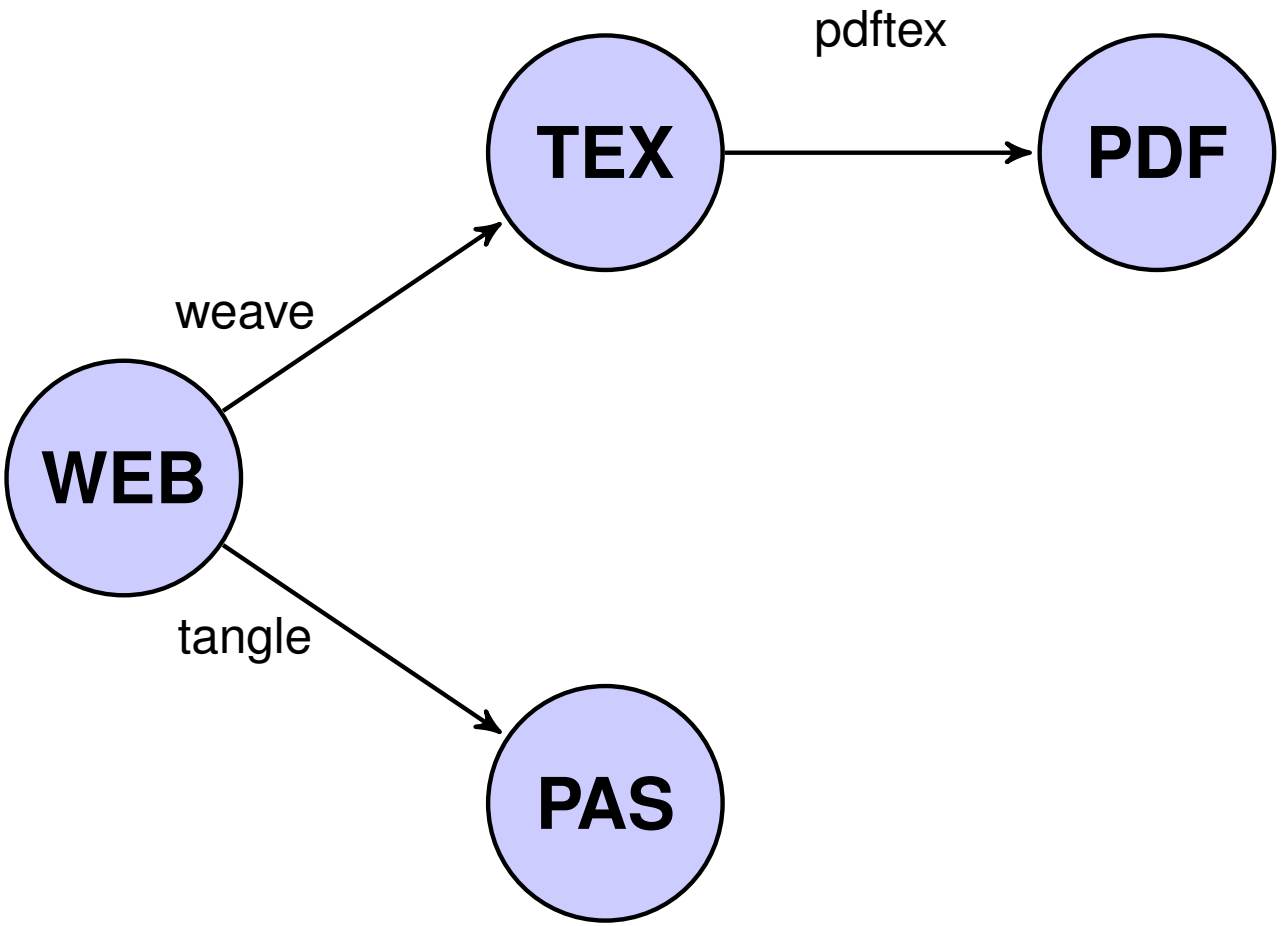
The compilation of a WEB file



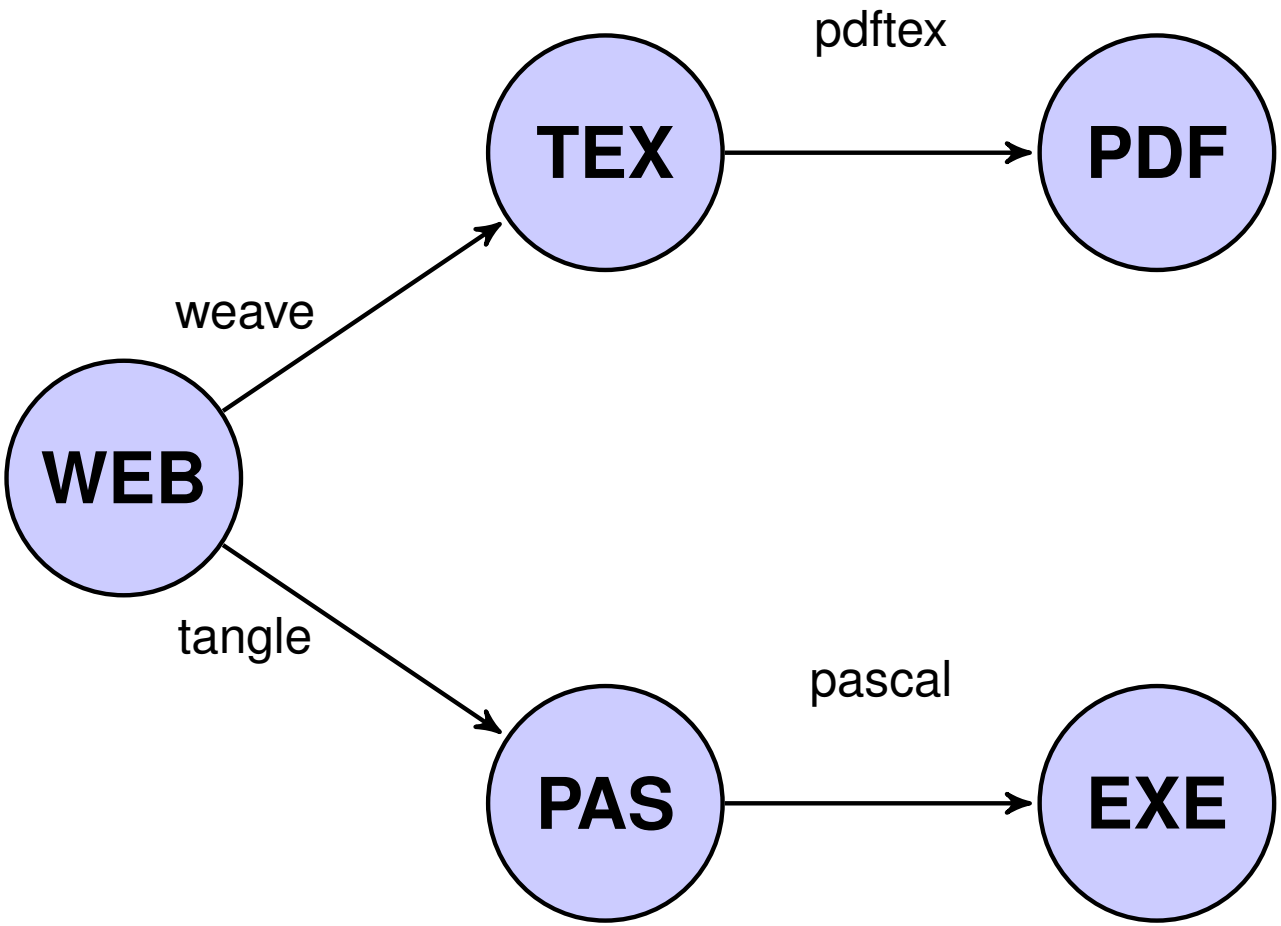
The compilation of a WEB file



The compilation of a WEB file

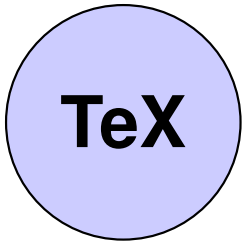


The compilation of a WEB file

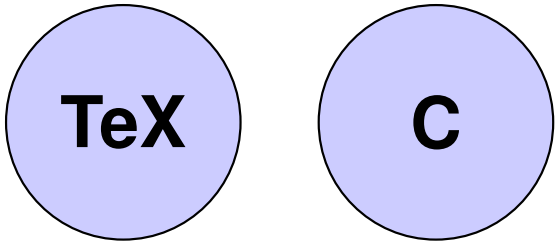


What is CWEB

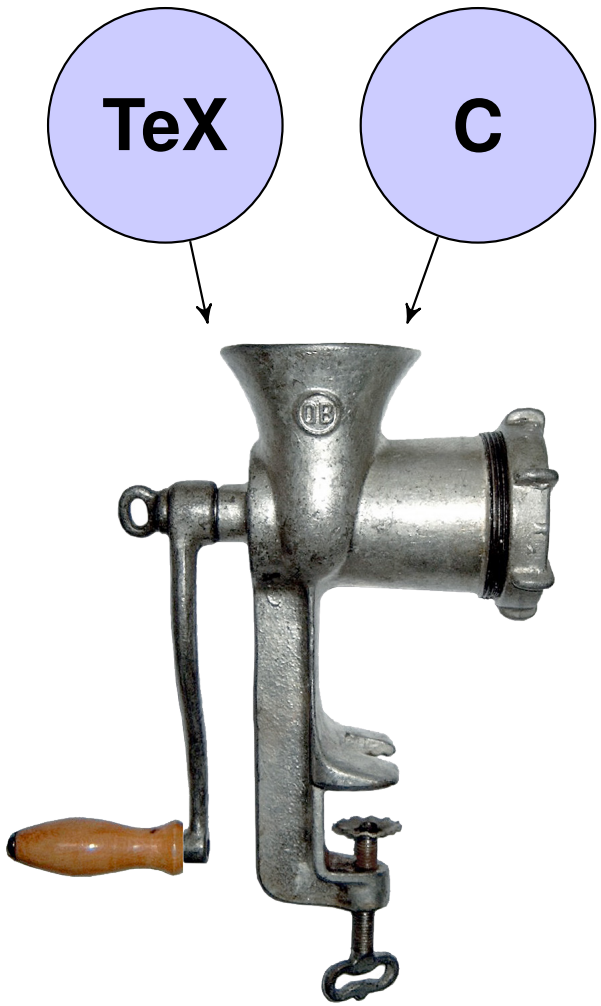
What is CWEB



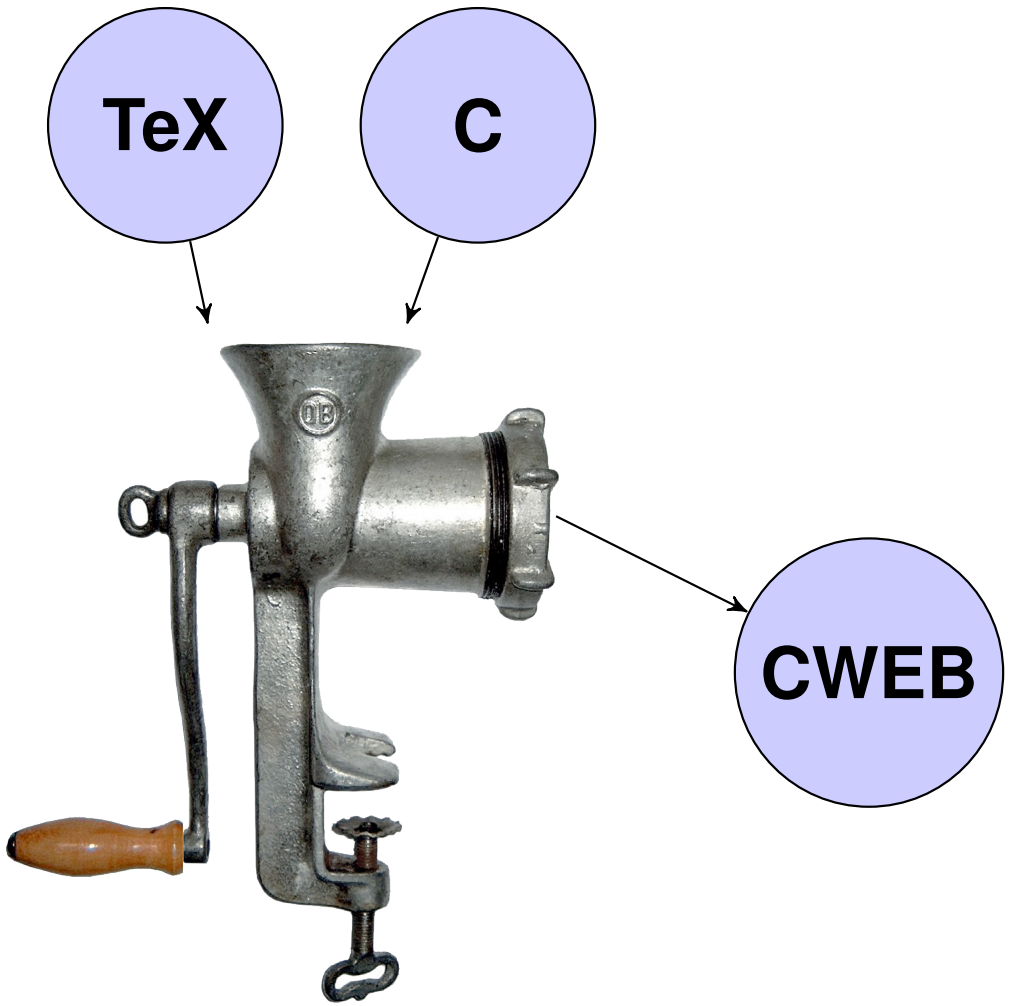
What is CWEB



What is CWEB

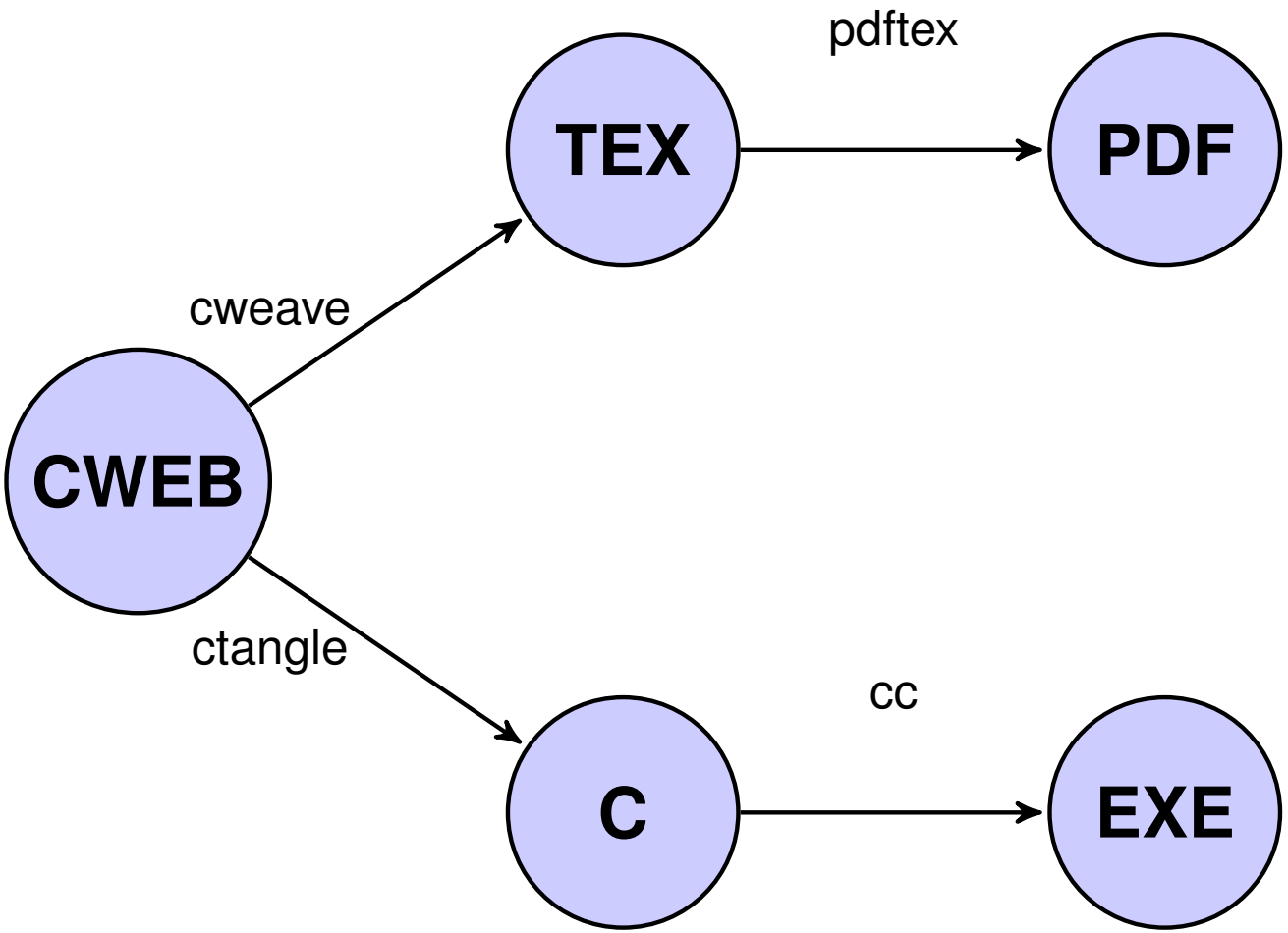


What is CWEB



CWEB

The compilation of a CWEB file



CWEB

- CWEB was created by Donald E. Knuth and Silvio Levy.

CWEB

- CWEB was created by Donald E. Knuth and Silvio Levy.
- CWEB uses C instead of Pascal as the programming language.

CWEB

- CWEB was created by Donald E. Knuth and Silvio Levy.
- CWEB uses C instead of Pascal as the programming language.
- It has almost the same syntax as WEB.

CWEB

- CWEB was created by Donald E. Knuth and Silvio Levy.
- CWEB uses C instead of Pascal as the programming language.
- It has almost the same syntax as WEB.
- Is used more frequently than WEB.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.
  - A code part where the program can be coded.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.
  - A code part where the program can be coded.
- The parts of a section must be exactly in this order.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.
  - A code part where the program can be coded.
- The parts of a section must be exactly in this order.
- An unnamed section with the char sequence “@␣”.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.
  - A code part where the program can be coded.
- The parts of a section must be exactly in this order.
- An unnamed section with the char sequence “@␣”.
- A named section begins with the char sequence “@*Name.”.

Sections

- A CWEB program consists of sections where each section consist of 3 parts:
  - A TEX part where the content of the section can be described.
  - A middle part which consists of special macros.
  - A code part where the program can be coded.
- The parts of a section must be exactly in this order.
- An unnamed section with the char sequence "@␣".
- A named section begins with the char sequence "@*Name.".

Example of a section

```
@
This is a example section
@c
int main();
```

C part

- An unnamed C part can be started with the sequence “@c”

C part

- An unnamed C part can be started with the sequence “@c”
- A named C part can be started with “@<Part Name@>=”

C part

- An unnamed C part can be started with the sequence “@c”
- A named C part can be started with “@<Part Name@>=”
- In a C part it is possible to include another named C part.

C part

- An unnamed C part can be started with the sequence “@c”
- A named C part can be started with “@<Part Name@>=”
- In a C part it is possible to include another named C part.

Example

```
@
This is the declaration of the main function
@c
int main(int argc,char** argv){
@<Main...@>;
}
@
This is the code of the main function
@<Main function Code@>=
printf("Hello World");
```

A example of a CWEB documentation

§2 HELLO CWEB OUTPUT 1

1. This is the declaration of the main function

```
int main(int argc, char **argv)
{
  ⟨ Main function Code 2 ⟩;
}
```

2. This is the code of the main function

⟨ Main function Code 2 ⟩ ≡

```
printf("Hello World");
```

This code is used in section 1.

argc: 1.
argv: 1.
main: 1.
printf: 2.

Figure : The document output of a CWEB file

In Limbo section

- The code which is written before the first section starts is called “in Limbo”
- This part will be copied in plaintex to the .tex file that is produced by cweave.
- It can be used to specify special TEX options like the page layout or function definitions.

In Limbo section

- The code which is written before the first section starts is called “in Limbo”
- This part will be copied in plaintex to the .tex file that is produced by cweave.
- It can be used to specify special TEX options like the page layout or function definitions.

The middle part

- The formatting of the keywords can be changed.
- In WEB this part is used to define macros.
- In CWEB the macro function was dropped because C already supports macros.

Formatting

- In the TEX part there is a special mode to easily format C code.
- This code can simply be written between “|”.
- For example “|int x = 2;|”.
- In a C comment also TEX commands can be used.

Formatting

- In the TEX part there is a special mode to easily format C code.
- This code can simply be written between “|”.
- For example “|int x = 2;|”.
- In a C comment also TEX commands can be used.

§1 TEST CWEB OUTPUT 1

1. A integer with a value: **int** $x = 2$;
A C function: **extern void** *function*( );
A String: `"Hello this is a Test String"`

```
int math_func( )
{
  calc( );      /* Calculates ∛((n+1)/3) */
}
```

calc: 1.
function: 1.
math_func: 1.
x: 1.

Figure : Formatting

- WEB and CWEB where never widely used but many newer programming languages got influenced by these languages

- WEB and CWEB where never widely used but many newer programming languages got influenced by these languages
  - Haskell has a own mode for literate programming.

- WEB and CWEB where never widely used but many newer programming languages got influenced by these languages
  - Haskell has a own mode for literate programming.
- Many similar languages or forks of WEB were created.

- WEB and CWEB where never widely used but many newer programming languages got influenced by these languages
  - Haskell has a own mode for literate programming.
- Many similar languages or forks of WEB were created.
  - noweb
  - nuweb
  - Funnelweb
  - pyWeb
  - Lp4all
  - ...

99 Bottles of Beer

99 Bottles of Beer

```
\def\title{99 Bottles of Beer}
@
In this part the output of the song is created.
It is simply a output of the text with a specific number of beers.

@<Text of the Song@>=
printf("%d bottles of beer on the wall, %d bottles of beer.\n",bottles,bottles);
printf("Take one down pass it around,%d bottles of beer on the wall.\n\n",bottles-1);

@
In this part the loop of the program is defined. It is a simple for loop
where the number of bottles is initialized and then decrased until
it reaches 1.
@<Loop of the function@>=
int bottles = 99;
for(;bottles>0;bottles--)
{
        @<Text...@>;
}
```

99 Bottles of Beer

99 Bottles of Beer cont.

```
@
The main function of the 99 bottles of beer code which contains the
main structure of the program.
@c
@<Includes@>;
int main(int argc,char** argv)
{
        @<Loop...@>;
        @<Finish of the Song@>;
}

@
This is the finish of the song which is executed when the loop ends.
@<Finish of the Song@>=
printf("1 bottle of beer on the wall, 1 bottle of beer.\n");
printf("Take one down and pass it around, no more bottles of beer on the wall.\n\n");
printf("No more bottles of beer on the wall, no more bottles of beer.\n");
printf("Go to the store and buy some more, 99 bottles of beer on the wall.\n");

@
The includes which are needed to use the |printf| function
@<Includes@>=
#include<stdio.h>
#include<stdlib.h>
```

Thank you for your attention.