LaTeX Webinar

"A Practical Guide to Typesetting"

by: Mustafa Youldash

LaTeX Webinar

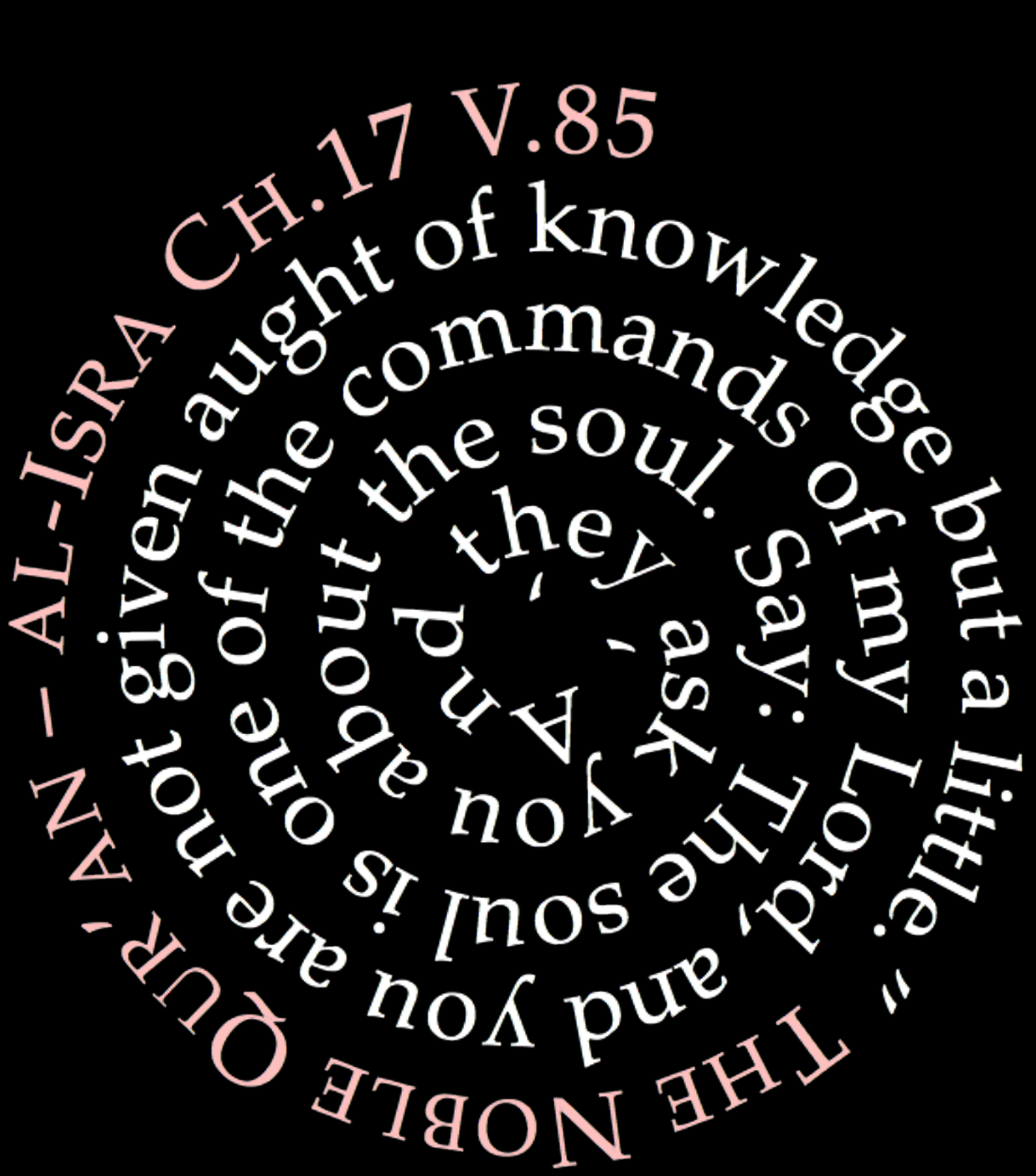
"A Practical Guide to Typesetting"

special thanks: Omar Mayan

LaTeX

Webinar

Mustafa Youldash
mmyouldash@uqu.edu.sa



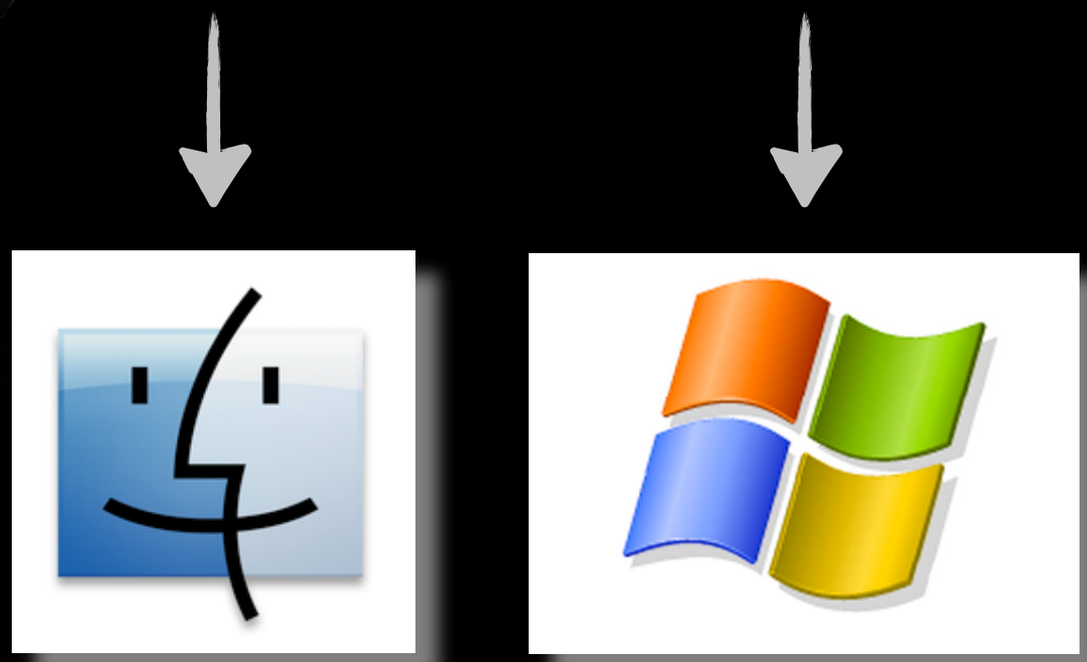
ويسألونك عن الروح قل الروح من أمر ربى وما أوتيتم من العلم إلا قليلا

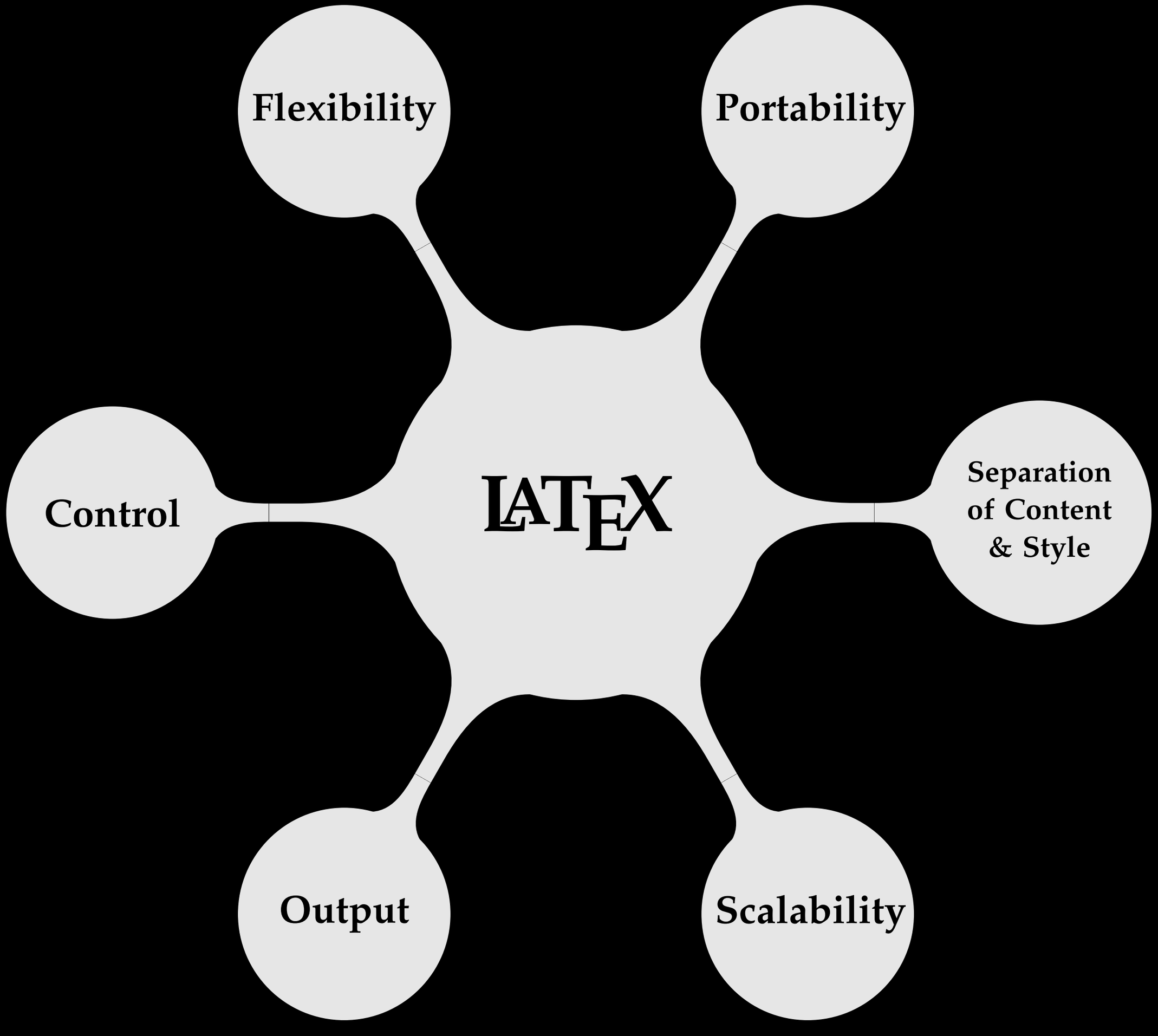
Let's
do this!

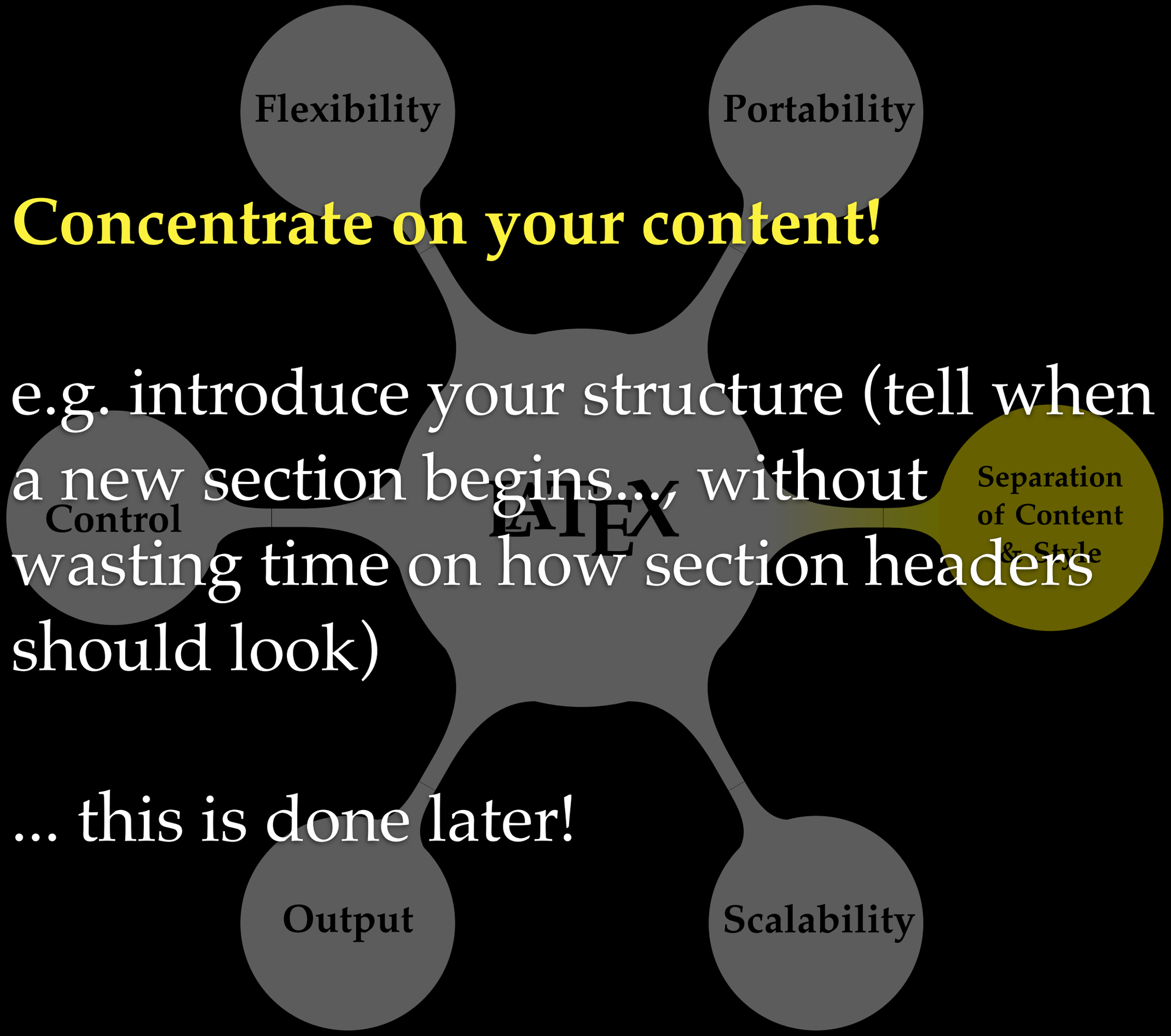
- Why LaTeX?
- Prerequisites
- Writing and Typesetting LaTeX Files
- Bibliographies
- Further to LaTeX...

Why?

A free, multilingual, open source typesetting system, used for creating beautiful-looking documents, books, images, ...



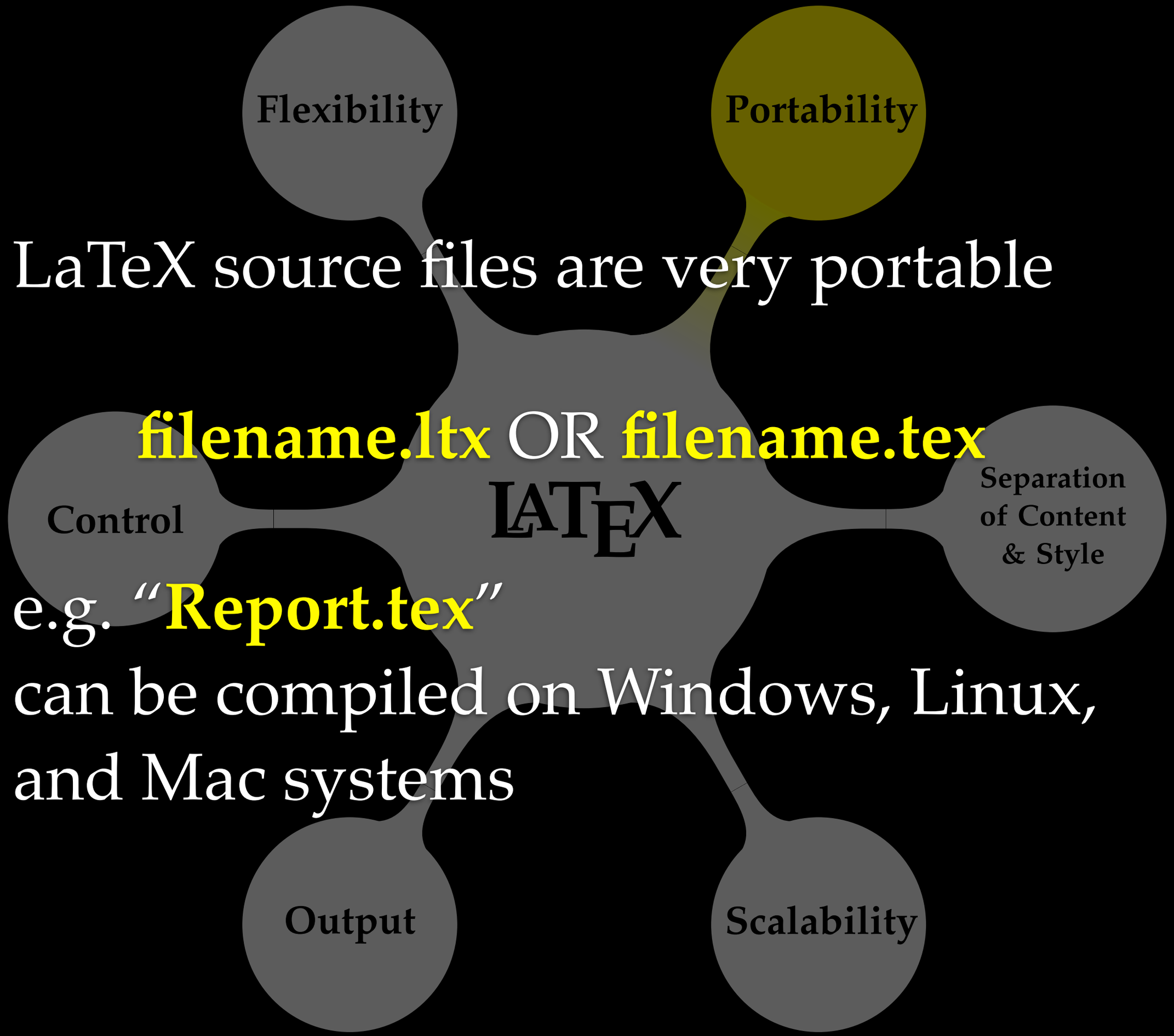

Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability



Concentrate on your content!

e.g. introduce your structure (tell when a new section begins...) without wasting time on how section headers should look)

... this is done later!


Flexibility
Portability
LaTeX source files are very portable
filename.ltx OR filename.tex
LATEX
Control
Separation of Content & Style
e.g. "Report.tex"
can be compiled on Windows, Linux, and Mac systems
Output
Scalability

Flexibility

Portability

You can get it to do just about anything you can think of!

Control

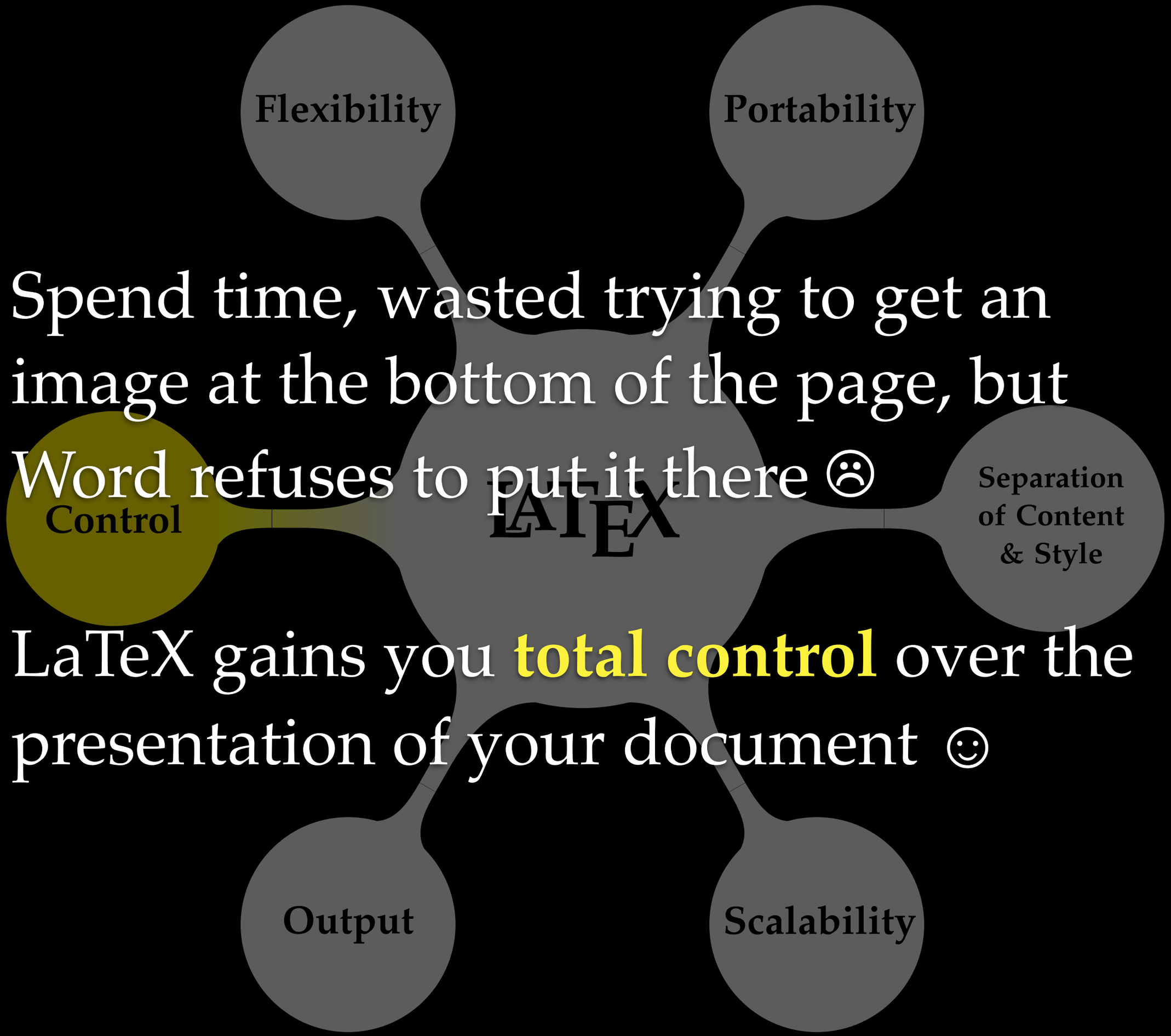
LaTeX

Separation of Content & Style

... via an overwhelming selection of packages or “small programs” (**FREE**)

Output

Scalability



Spend time, wasted trying to get an image at the bottom of the page, but Word refuses to put it there ☹

LaTeX gains you **total control** over the presentation of your document ☺

Flexibility

Portability

Fully-featured mathematical formulae

Diagrams and illustrations

Control

LaTeX

Separation of Content & Style

Output to an image

.png .jpeg .eps .tif .pdf

Output

Scalability

Flexibility

Portability

Very little risk of you ever losing your original sources

Very scalable with large "**sub-documents**" like in Word

Never bother with MS Office viruses!

LaTeX

Control

Separation of Content & Style

Output

Scalability

Word Processors.pages

Word Processor +'s and -'s

One of the best advantages of word processors is being able to see the results as you enter text and pictures. For example, it is easy to insert images and wrap text around them. You can also change as you type such text attributes as **bold**, *italic*, font and size.

On the downside, word processors generally do a below average job of typography, that is controlling the overall appearance of how words and images appear on a page. They have few, or difficult to use, functions for fine-tuning line breaks, justified type, word spacing, hyphenation, line spacing and so on.

While word processors are great for many uses, for the most part, printed materials created today with word processors are of lower typographic quality than those published in the 19th and the 20th centuries using pre-computer typesetting methods.

Also making changes to a large word processor document format can be very difficult and time consuming, even if you use the so-called "style sheets".

TeX and its offspring such as eplain, LaTeX and ConTeXt can consistently produce high-quality typographic output.

41% Page 1 of 1

WordProcessors.tex

```
\documentclass[a4paper,11pt]{article}

\usepackage[hscale=.68, vscale=.80, centering]
{geometry}

\usepackage{bookman}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{avant}

\usepackage{graphicx}
\usepackage{wrapfig}
\usepackage{color}
\usepackage[]{parskip}

\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{2}

\pagestyle{empty}

\usepackage{pdftex, bookmarks=false,
pdfpagemode=UseThumbs, colorlinks=true,
linkcolor=links, anchorcolor=links, citecolor
filecolor=links, menucolor=links, pagecolor=
urlcolor=links]{hyperref}

\begin{document}
```

TeX Program

Word Processor +'s and -'s

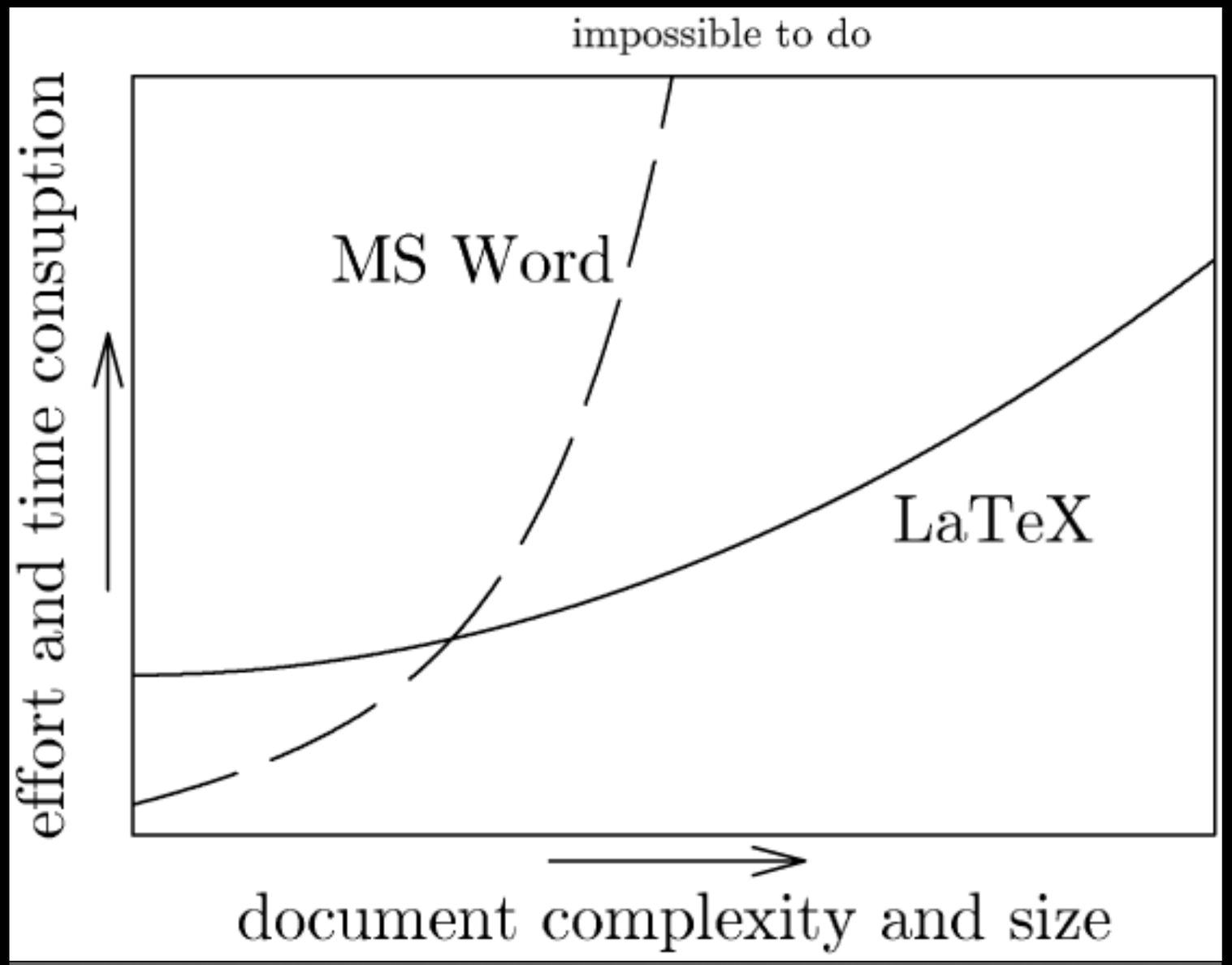
One of the best advantages of word processors is being able to see the results as you enter text and pictures. For example, it is easy to insert images and wrap text around them. You can also change as you type such text attributes as bold, italic, font and size.

On the downside, word processors generally do a below average job of typography, that is controlling the overall appearance of how words and images appear on a page. They have few, or difficult to use, functions for fine-tuning line breaks, justified type, word spacing, hyphenation, line spacing and so on.

While word processors are great for many uses, for the most part, printed materials created today with word processors are of lower typographic quality than those published in the 19th and the 20th centuries using pre-computer typesetting methods.

Also making changes to a large word processor document format can be very difficult and time consuming, even if you use the so-called style sheets.

TeX and its offspring such as eplain, LaTeX and ConTeXt can consistently produce high-quality typographical output.


impossible to do
effort and time consuption
MS Word
LaTeX
document complexity and size

MiKTeX

EndNote

LaTeXiT

JabRef reference manager

BibDesk

TextMate

LyX

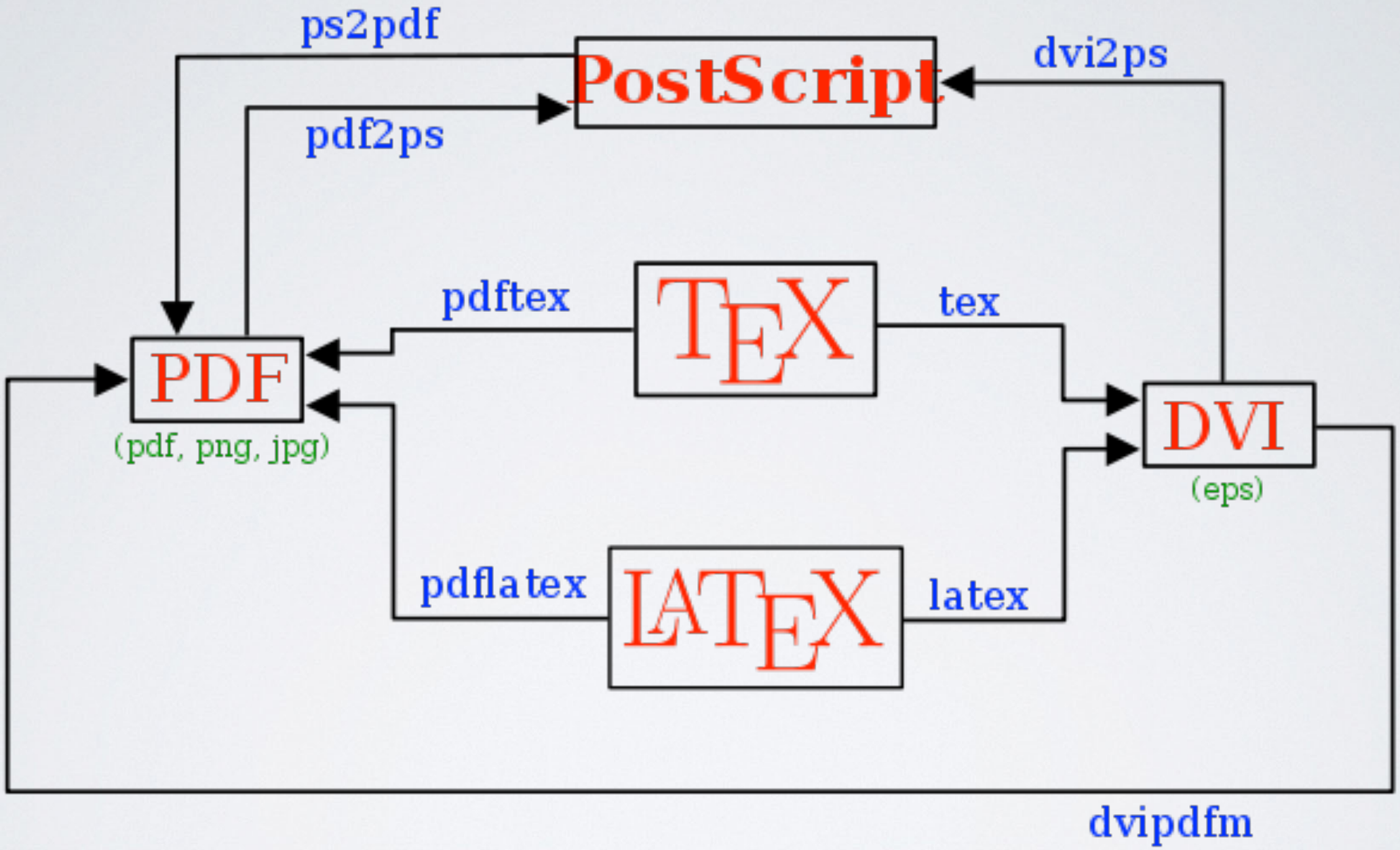
TextEdit

TeXShop

TeXworks

TeXnicCenter

LaTeX Equation Editor


ps2pdf
PostScript
dvi2ps
pdf2ps
pdftex
TEX
tex
PDF
(pdf, png, jpg)
DVI
(eps)
pdflatex
LATEX
latex
dvipdfm

Algorithm 1.0.1: BUILDGRAPHBASEDMST(V, E)

```
if n = 2
  then { if X[0] > X[1]
           then { T ← X[0]
                  X[0] ← X[1]
                  X[1] ← T
  else if n > 2
  then { m ← ⌊n/2⌋
         for i ← 0 to m − 1
           do A[i] ← X[i]
         for i ← m to n − 1
           do B[i] ← X[i]
         comment: Now sort the subarrays A and B
         BUILDGRAPHBASEDMST(m, A)
         BUILDGRAPHBASEDMST(n − m, B)
         i ← 0
         j ← 0
         for k ← 0 to n − 1
           do { if A[i] ≤ B[j]
                then { X[k] ← A[i]      (i)
                       i ← i + 1
                else { X[k] ← B[j]      (ii)
                       j ← j + 1
```

produce the following output:

On lines (i) and (ii) of Algorithm 1.0.1, we determine the kth element of the sorted array.

By supplying data that are relevant to the hypotheses, confirmed results are taken into account, and therefore the whole scenario may be taken up to the next

Article.tex

Typeset | LaTeX | Macros | Tags | Templates

```
\section{Introduction}
Throughout history, \ldots Word sucks!

\begin{table}[htp]
    \centering
    \begin{tabular}{c|c|c|c|c|c|c}
        {} & a & b & c & d & e & f \\
        \hline
        a & 0 & 184 & 222 & 177 & 216 & 237 \\
        \hline
        b & 184 & 0 & 45 & 123 & 128 & 200 \\
        \hline
        c & 222 & 45 & 0 & 129 & 121 & 203 \\
        \hline
        d & 177 & 123 & 129 & 0 & 40 & 86 \\
        \hline
        e & 216 & 128 & 121 & 40 & 0 & 83 \\
        \hline
        f & 237 & 200 & 203 & 86 & 83 & 0
    \end{tabular}
    \caption{Synthetic data set denoting pairwise
    \label{tab:Table_SyntheticData}
\end{table}
```

1 Introduction

Throughout history, . . . Word sucks!

	a	b	c	d	e	f
a	0	184	222	177	216	237
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	40	86
e	216	128	121	40	0	83
f	237	200	203	86	83	0

Table 1: Synthetic data set denoting pairwise distances between five nodes.

Article.tex

Typeset | LaTeX | Macros | Tags | Templates

```
\maketitle
\tableofcontents

\section{Introduction}
Throughout history, \ldots Word sucks!

\begin{figure}[htb]
    \centering
        \includegraphics[height=1.175in]{Figures/VisualizationPipeline.png}
    \caption{The Visualization Pipeline.}
    \label{fig:Figures_VisualizationPipeline}
\end{figure}

\subsection{Reasons}
You still have to pay for it :(
```

For any IV tool to generate its designated output whether in static form (images for example), or dynamic (such as animated scenes with or without interactivity), it must follow an architecture that is formally referred to at hand as the *visualization pipeline*. One typical example is illustrated in Figure 1.2 (Courtesy of Aigner et al. (2008)).

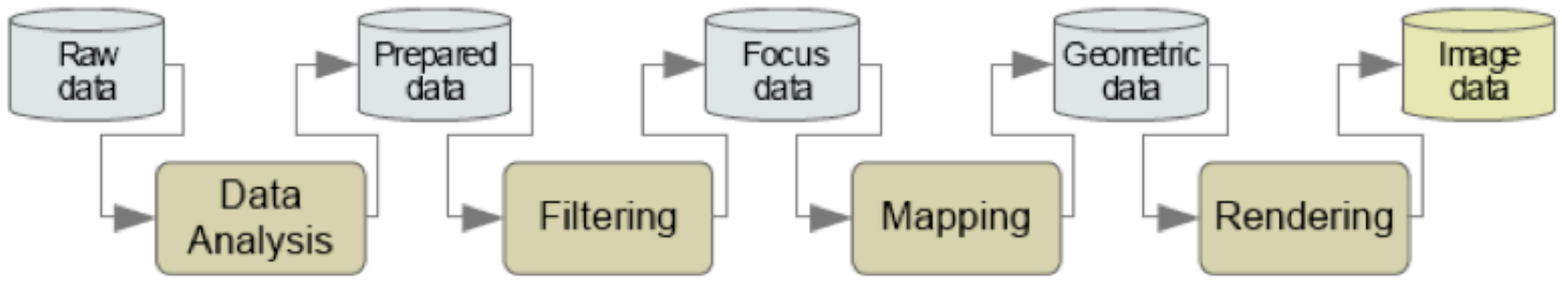


Figure 1.2: The Visualization Pipeline.

Proposition 4.1. *There is an exact sequence of sheaves on* $\widetilde{U}$*:*

$$\begin{array}{ccccc}
\mathcal{N}(Z-E) & \overset{\alpha}{\hookrightarrow} & \mathcal{I}_E\Omega^1(\log E) & & \\
 & & & \searrow^{\beta} & \\
 & & & & \left(\Omega^2(\log E)\big/\wedge^2\mathcal{N}(2Z)\right)\otimes\mathcal{O}(-Z-E) \\
 & & & & \downarrow{\scriptstyle\gamma} \\
 & & & & \Omega^3\otimes\mathcal{O}_{P+N-2Z}(-2Z)
\end{array}$$

Proof. See the gigantic diagram for the big picture. Since $\mathcal{O}(R)\approx\mathcal{O}(-Z)$ (by multiplication by h; see below), we have:

$$\begin{aligned}
&\left(\Omega^2(\log E)\big/\wedge^2\mathcal{N}(2Z)\right)\otimes\mathcal{O}(-Z-E) \\
&\quad\approx\ \Omega^2(\log E)\otimes\mathcal{O}(-Z-E)\big/\wedge^2\mathcal{N}(2Z)\otimes\mathcal{O}(-Z-E) \\
&\quad\approx\ \mathcal{I}_E\Omega^2(\log E)\otimes\mathcal{O}(R)\big/\wedge^2\mathcal{N}(2Z-E)\otimes\mathcal{O}(R),
\end{aligned}$$

and since $\wedge^3\mathcal{N}(3Z-E)\otimes\mathcal{O}(2R)=\mathcal{I}_E\Omega^3(\log E)\otimes\mathcal{O}(2Z-P-N+2R)$ (this will be shown later in this paper),

vector space $\Re^d$ would be represented as the following Framework:

$$P(u) \Leftarrow v \begin{cases} \forall u \in \Re^D & u = (u_1, u_2, \ldots, u_N) \\ \forall v \in \Re^d & v = (v_1, v_2, \ldots, v_n) \\ D \geq d \end{cases} \tag{2.1}$$

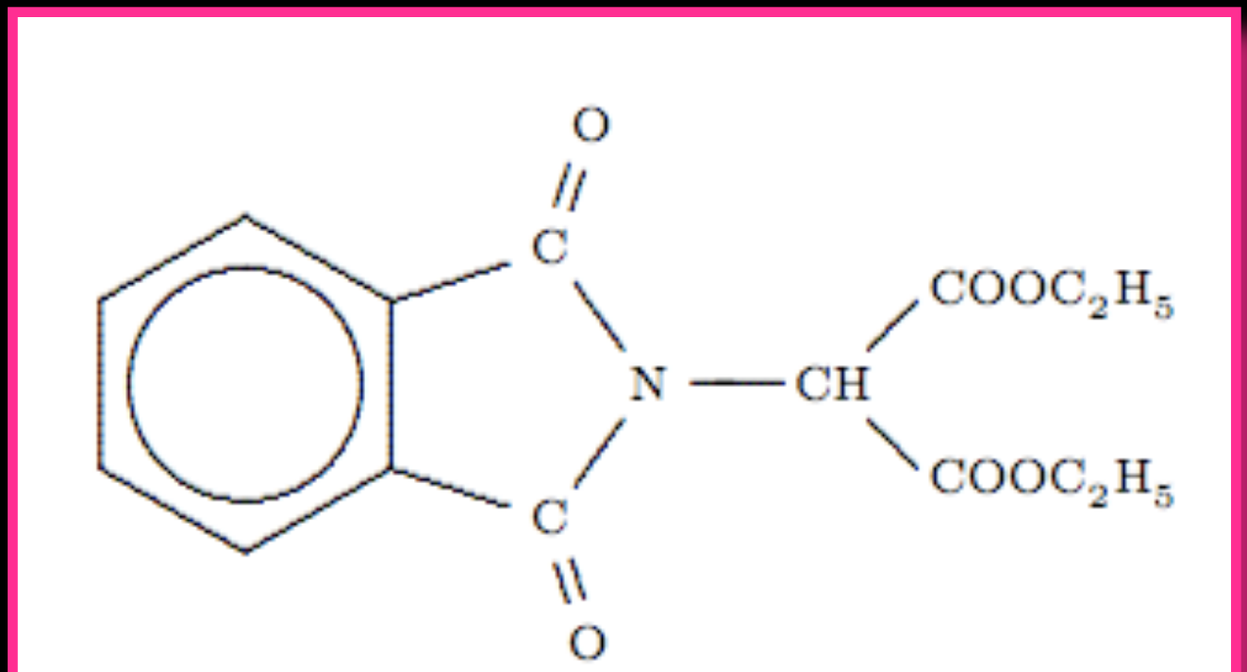
where N represents the number of original observations in the higher D space, and n on the other hand represents the number of projected observations in the lower d space. For instance, a transformation which maps the point (x, y, z) in $\Re^3$ to the point (x, y) in $\Re^2$ is a projection onto the xy plane (revealed in Figure 2.1). This function would be represented as the following matrix:

$$P \Leftarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$v_i = w_{i,1} \cdot u_1 + w_{i,2} \cdot u_2 + \ldots + w_{i,N} \cdot u_N \qquad \text{for } i = 1, 2, \ldots, n \tag{2.2}$$

or

$$v = \mathbf{W} \cdot u \tag{2.3}$$


O
C
N
CH
COOC2H5
COOC2H5
C
O

References

Arciniegas, I., Daniel, B. and Embrechts, M. J. (2001). Exploring financial crises data with self-organising maps (som), *in* N. Allinson, H. Yin, L. Allinson and S. J. (eds), *Advances in Self-Organising Maps*, Vol. 39-46, SpringerLink.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA.

Bishop, C. M., Svensén, M. and Williams, C. K. I. (1998). Gtm: The generative topographic mapping, *Neural Computation* **10**: 215–235.

Biswas, G., Jain, A. K. and Dubes, R. C. (1981). Evaluation of project algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-3**: 701–708.

Resources

Wiki book [LaTeX]

http://en.wikibooks.org/wiki/LaTeX

TeX - LaTeX Stack Exchange [wiki]

http://tex.stackexchange.com

TikZ and PGF Examples [forum]

http://www.texample.net

The “Not So Short” Introduction to LaTeX [PDF]

http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

BibTeX [Bibliography Management]

http://bibtex.org

Excel-to-LaTeX [Excel Macro]

http://www.ctan.org/pkg/excel2latex