The LaTeX Workshop

"A Practical Guide to Typesetting"

by: Mustafa Youldash, PhD

The LaTeX Workshop

"A Practical Guide to Typesetting"

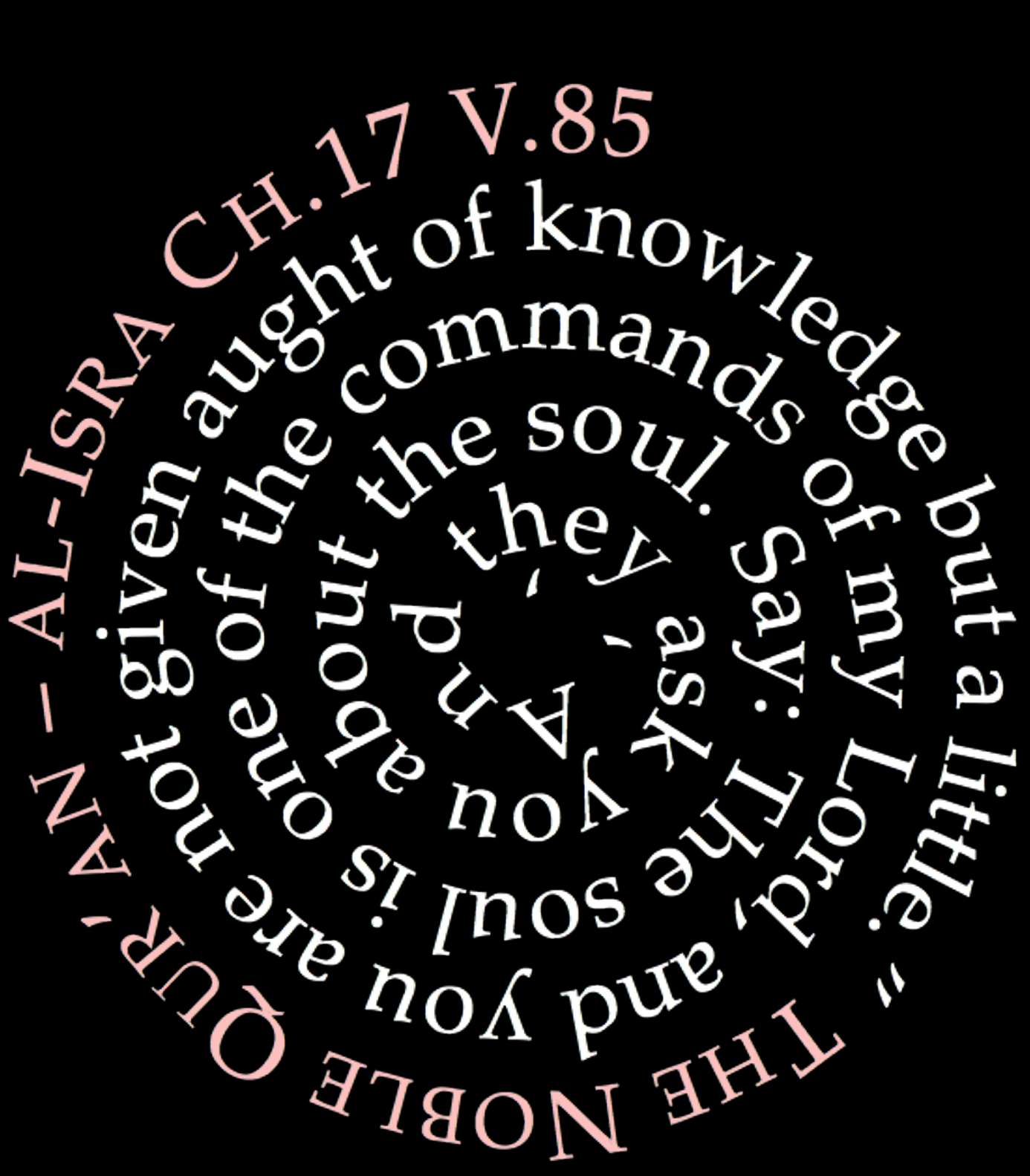
special thanks: CCSIT, IAU

The LaTeX Workshop

جامعة الإمام عبد الرحمن بن فيصل

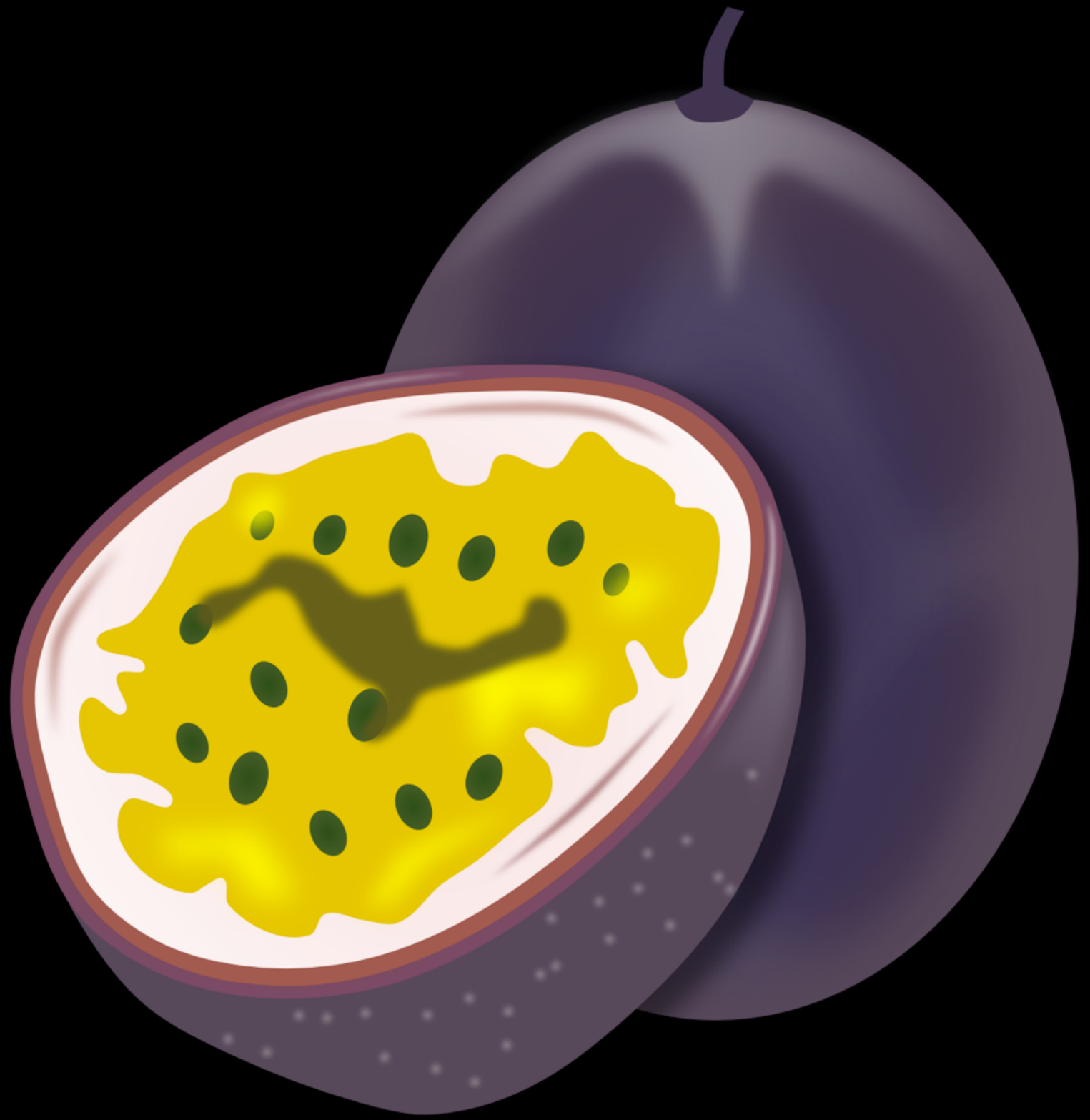
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

Mustafa Youldash, PhD
mmzyouldash@iau.edu.sa



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

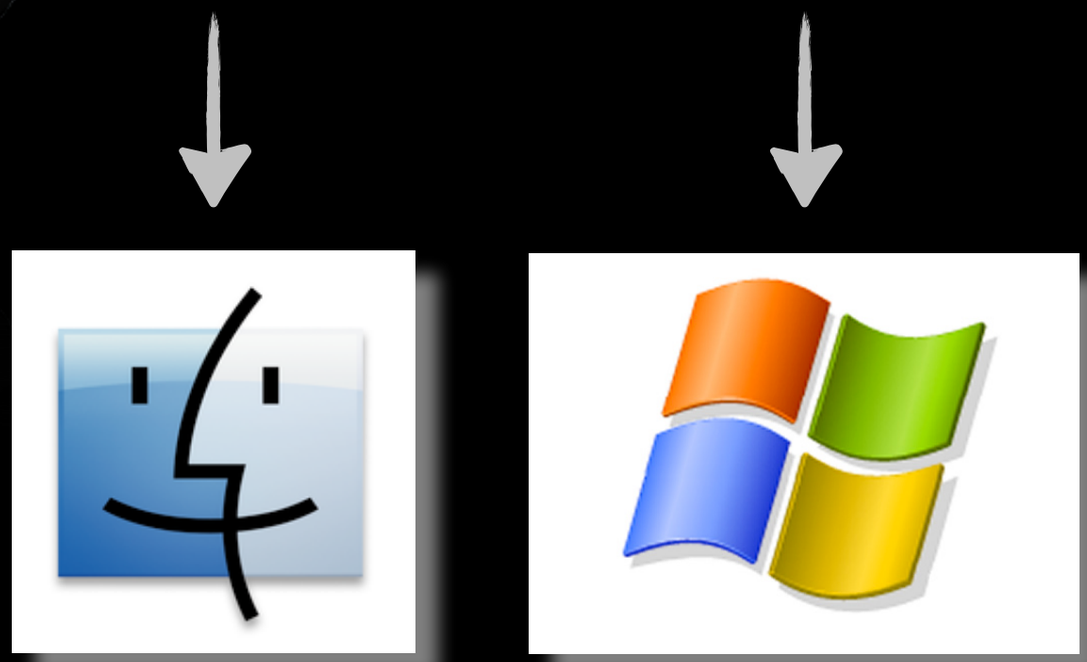
Let's
do this!



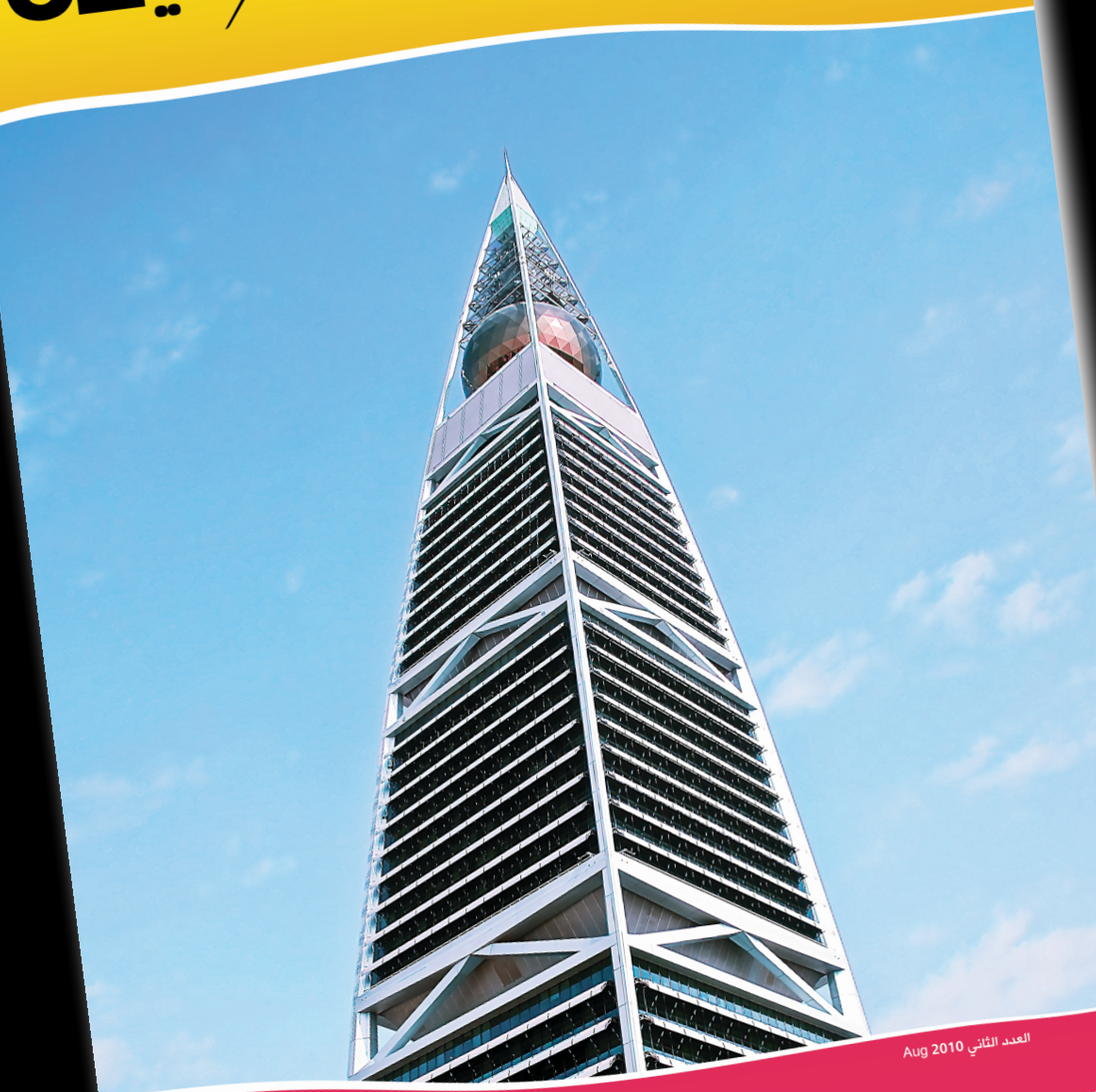
- Why LaTeX?
- Prerequisites
- Writing and Typesetting LaTeX Files
- Bibliographies
- Further to LaTeX...

Why?

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



saudi 2 oz
سعودي زد



العدد الثاني Aug 2010

A Love Affair with Computer Typesetting

By: Mustafa Youldash

If you have never experienced using anything other than regular word processing applications such as Word from Microsoft Office, Writer from OpenOffice.org, WordPerfect from Corel, Pages from Apple iWork, or any other program in that sense for producing documents, then you will inevitably be in a state of shock when you "typeset" your first document using sophisticated systems like LaTeX (pronounced "lay-tech" or "lah-tech"). Perhaps, it is worthy to note that the experience itself can be described, utterly as "a bit of a rocky patch". So why should we even bother with such systems in the first place?

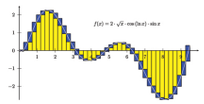
Imagine the phrase "typesetting" as a process by which characters are assembled into formatted text for the purpose of producing prints or "graphic objects". Prior typesetting as we know it today, typical machines were invented (e.g. a typewriter) and were used extensively to serve that simple purpose; by setting text (by hand) using individual letters of a type. Today, typesetting on a computer requires the prior process of designing fonts and storing them in some manner, and of course it involves retrieving the stored letters and ordering them according to a language's orthography for visual display.

To fully grasp the idea about typesetting, consider an author, a book designer, and a typesetter for example. An author compiles and delivers a typed manuscript to a publisher (or publishing company). The publisher then decides on a suitable layout (i.e. fonts, number of columns, column width and height, line spacing, headers, etc.) to be applied on the document. Meanwhile, the book designer finds out what the author had in mind while writing his/her manuscript, and then decides on chapter headings, citations, examples and illustrations, mathematical formulae (if applicable), and other components based on both designer's expertise, and original contents of the manuscript. The book designer finally appends his/her instructions onto the manuscript and then delivers it straight to the typesetter, whom is responsible for typesetting the book according to the embedded instructions.

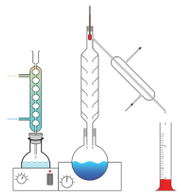
Fortunately, you do not have to go and seek the aid of a book designer, a publisher, nor human typesetter every time you have something simple like an article, an essay, or even a large-scale project like a masters or doctoral thesis to accomplish. With LaTeX, you play the role of designer, publisher, and typesetter. As a matter of fact, LaTeX pretty much grants you the power to render your imagination as a living and sensible reality. To put things into perspective, LaTeX is nothing but a mere computer program, which enables anyone to typeset and print his/her work at the highest typographical quality, using a predefined, professional layout. The program was originally written in the early 1980s by Leslie Lamport, and incorporates TeX (pronounced "tech") as its typesetter (formatting) engine. Like LaTeX, TeX is another computer program, created by Donald E. Knuth for typesetting text and mathematical formulae. Since LaTeX is nothing more than a program, you still have to supply all the relevant information, that would resemble the logical structure of your intended work. In simple terms, every time you use LaTeX, you communicate and interact with it via series of "LaTeX commands". Nevertheless, it is worthy to note that such systems have a rather steep learning curve (i.e. rather than relying on simple and straightforward WYSIWYG "what you see is what you get" editors like Word, instead you find yourself spending much time (especially as a beginner) getting your hands dirty and adopting a way that is based on allot of command lines in a structured fashion). The succeeding paragraphs will briefly outline most of what gives LaTeX over any conventional WYSIWYG editor a cutting edge (i.e. a better choice).

We begin by mentioning portability, which is achieved in multiple ways: (i) An actual LaTeX file is merely an ordinary text file (filename.tex or filename.ltx), of which is just about the most portable format in computing. (ii) LaTeX has been implemented on just about every mainstream platform you could name today (e.g. Microsoft Windows, Linux, and Mac OS X). (iii) The default output file for LaTeX is a device independent (DVI) file format. Such format was around well before portable document format (PDF) files. Another thing is that these output files can be easily viewed via software readers (e.g. Adobe Reader). Of course you can directly produce PDFs by typesetting with pdfTeX, which is another program that is part of the TeX online distributions.

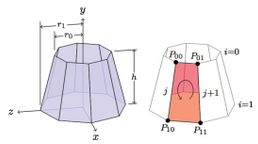
Another interesting matter is the separation of content and style. Obviously this may not make any sense, possibly because most Word users are not fully aware why this so beneficial. When producing your LaTeX document, you are concentrating on the content itself. You introduce structure "explicitly" by informing LaTeX when a new section begins for instance, and you do not spend so much time trying to decide how section headers should look like. This is of course accomplished at a later stage. In contrast, an average Word user would immediately highlight a given section header, and then apply formatting to it (maybe a larger font? maybe underline?, etc). Ironically, this will then have to be manually applied to every header, and of course this is without a doubt "tedious". On the other hand, if you rely on LaTeX, you simply define how different elements within your document should look. If you fancy a change, you only change the style definitions once! and then the presentation of the document will be updated automatically. Someone in this case might jump up and say: "Word does in fact include a similar Styles feature". However, because this is optional, people often do not realize it at first. LaTeX forces you to declare document semantics, which is why you can rely on it to produce a consistent looking publications. →



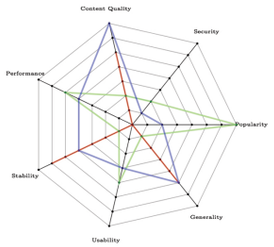
Plotting Equasions



Chemical Instruments



3D Cone



Spiderweb Diagram

Flexibility, is yet another factor I personally enjoy the most. This implies that with LaTeX, you can pretty much do just about anything imaginable! Over the years, the LaTeX community have been working very hard to give us an overwhelming selection of TeX packages for expanding the potential; packages that are freely accessible via the comprehensive TeX archive network (CTAN).

As far as control is considered, LaTeX grants you total control over the presentation of your work. For example, imagine yourself running into a situation where you quickly become frustrated by Word's rather "unintelligent" or "complex" user interference, while you waste considerable amount of time trying to get one figure that you know will fit at the bottom of a page, but you get stunned from Word "refusing" to place it there!

Output, as mentioned earlier, is surely what stacks LaTeX above any conventional word processor. In fact, it is difficult to disagree that the output generated from LaTeX (e.g. PDFs) is far superior to what Word can produce. This is emphasized greatest when you have to prepare your work with rather "more complex" mathematical content, which is a major strength for LaTeX. LaTeX also incorporates much better methods for text kerning, hyphenation and justification, that simply would make your output far more professional. In addition, it takes little effort to convert your work into a PostScript (i.e. filename.ps) or PDF file, or to images in that manner (JPEGs, GIFs, TIFFs, PNGs, EPSs, etc).

Scalability, as an advantage, can be expressed as working with documentation that requires to be "sliced" into smaller chunks, or in other words: into sub-documents. With LaTeX, you can continue your research, while writing your chapters into separate documents that would end up forming one output file. With this approach, better performance is highly achievable! Especially when you include your table of contents, table of figures, cross-references, indexes and bibliographies at one stage, then you suddenly decide to rearrange your chapters (e.g. chapter 3 becomes 4 and vise versa).

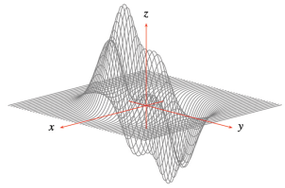
If I am not mistaken, stability is probably one of the strongest gestures of LaTeX. This is due to the very nature of its implementation for being "mature", and virtually bug-free. When writing up parts of you work, there is very little risk of you ever losing your original source text. Where as with conventional word processors, almost any tool within its integrated environment is capable of corrupting your file if it causes a crash.

When it comes down to cost, you may rest assured that LaTeX comes for free.. Yes, FREE! This is of course opposing to most commercial word processing application, where you still have to purchase your own copy of Word, or even a copy of EndNote just for the sake of managing your references and relevant citations. Having said that LaTeX is free does not emphasize that all front-end programs are completely free. For example, you may download and install TeX editors like TeXnicCenter for Windows, or TeXShop for the Mac. However you may consider buying other software that would be, in a sense, better in terms of productivity like TextMate on Mac OS X, and so forth. Regardless of what program you wish to use for all your typesetting needs, the only thing you have to consider is simplicity.

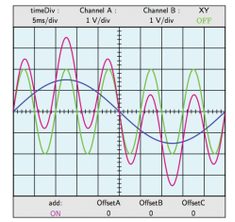
In the end, it is not surprising to see that modern typesetting systems such as TeX, LaTeX, even the more compelling versions like XeTeX, ConTeXt, PSTricks, TikZ and PGF, Beamer, and many more that are for sure, not enough to be covered in this article. Such systems have indeed touched many fields such as Science, Technology, Engineering, Medicine, Arts, Design, Music, Gaming, and many more. Nevertheless, I owe you a confession! (I may seem to be promoting you to completely get rid of your favorite word processor. On the contrary, this matter really depends on your ability in coping with the complexity and strict rules, that of which you have to follow every time you use LaTeX. To some, word processors may be more than enough. To others, LaTeX would sound like a suitable alternative). Finally, I wish to share to you samples, which have been designed using some of the packages that are mentioned in this paragraph - as a way for bringing insight into your inquiring mind.

Recommended Reading

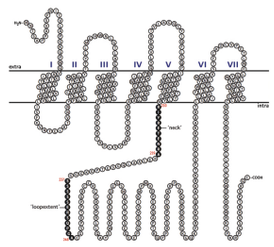
http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf (The Not So Short Into about LaTeX)
The LaTeX Companion (Tools and Techniques for Computer Typesetting)
LaTeX Graphics Companion, (2nd Edition)
Guide to LaTeX (4th Edition)
http://tug.org/ (the Comprehensive TeX Archive Network - CTAN)
http://en.wikibooks.org/wiki/LaTeX (Wiki Book about LaTeX)
http://www.youldash.net/default/LaTeX.html (How to Install LaTeX)
http://www.texample.net/ (Good LaTeX Examples)



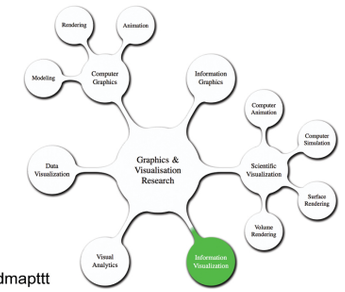
3D



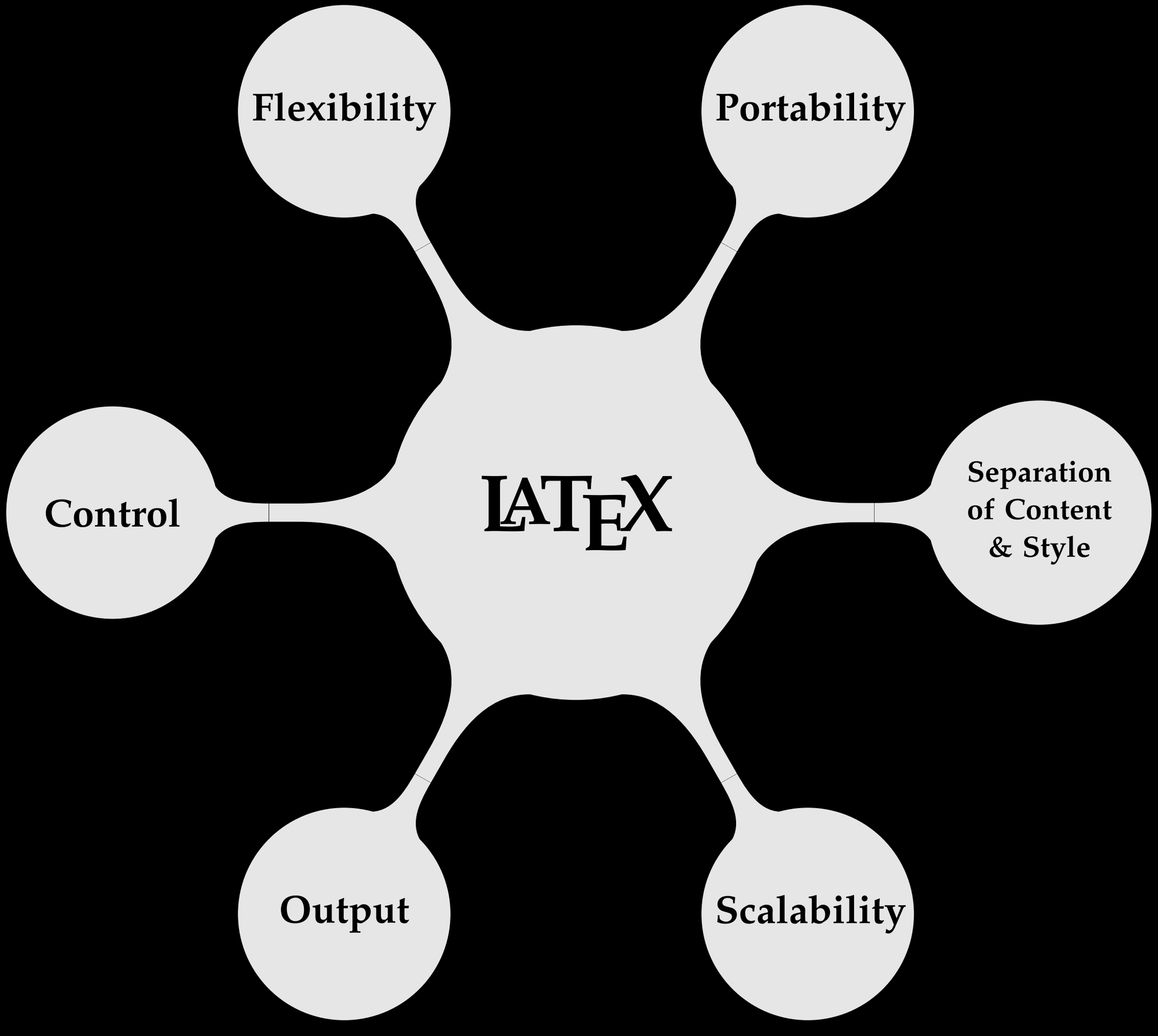
Oscilloscope



Membrane Protein Topology Plot



Mindmapttt


Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

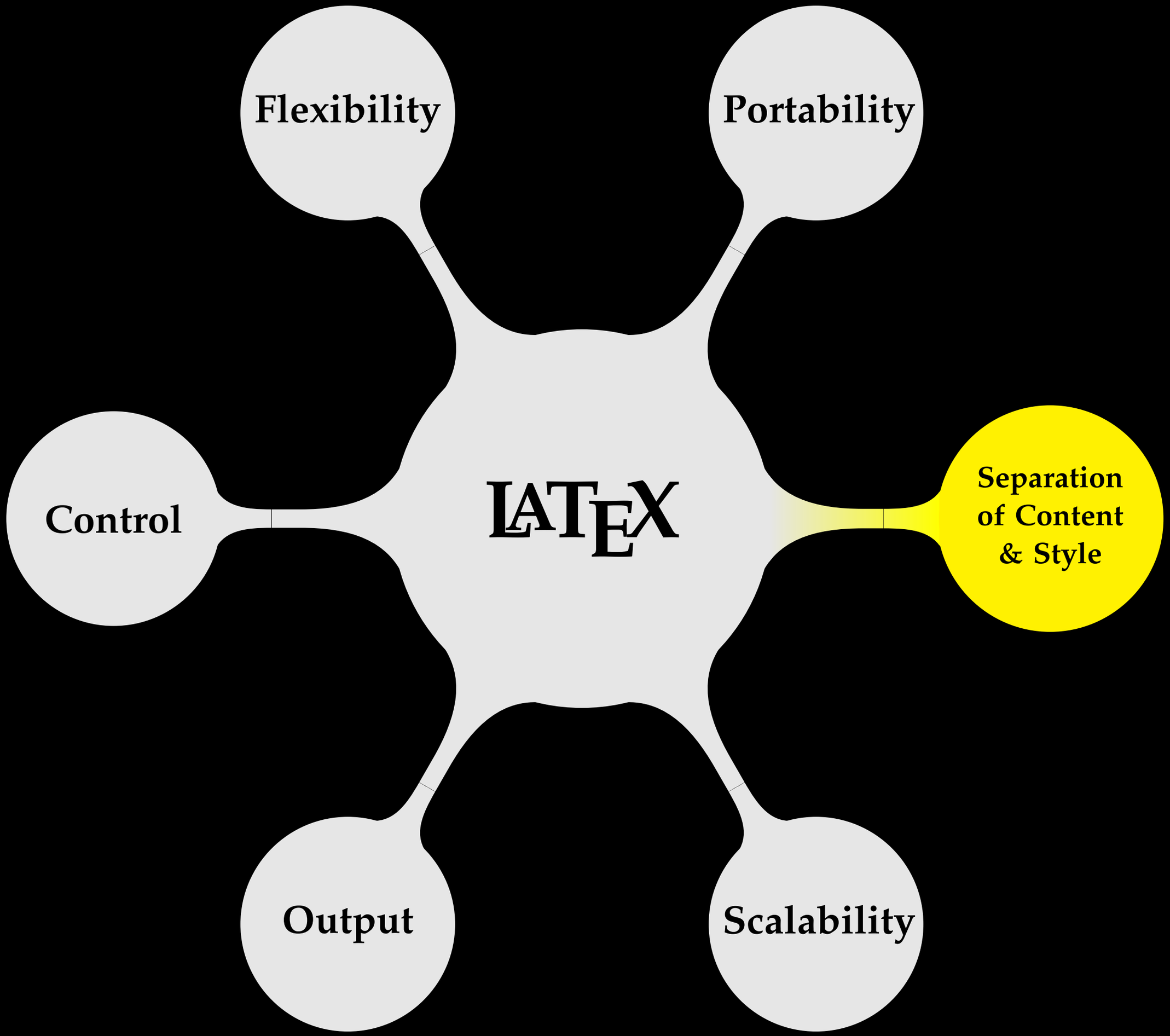
Flexibility

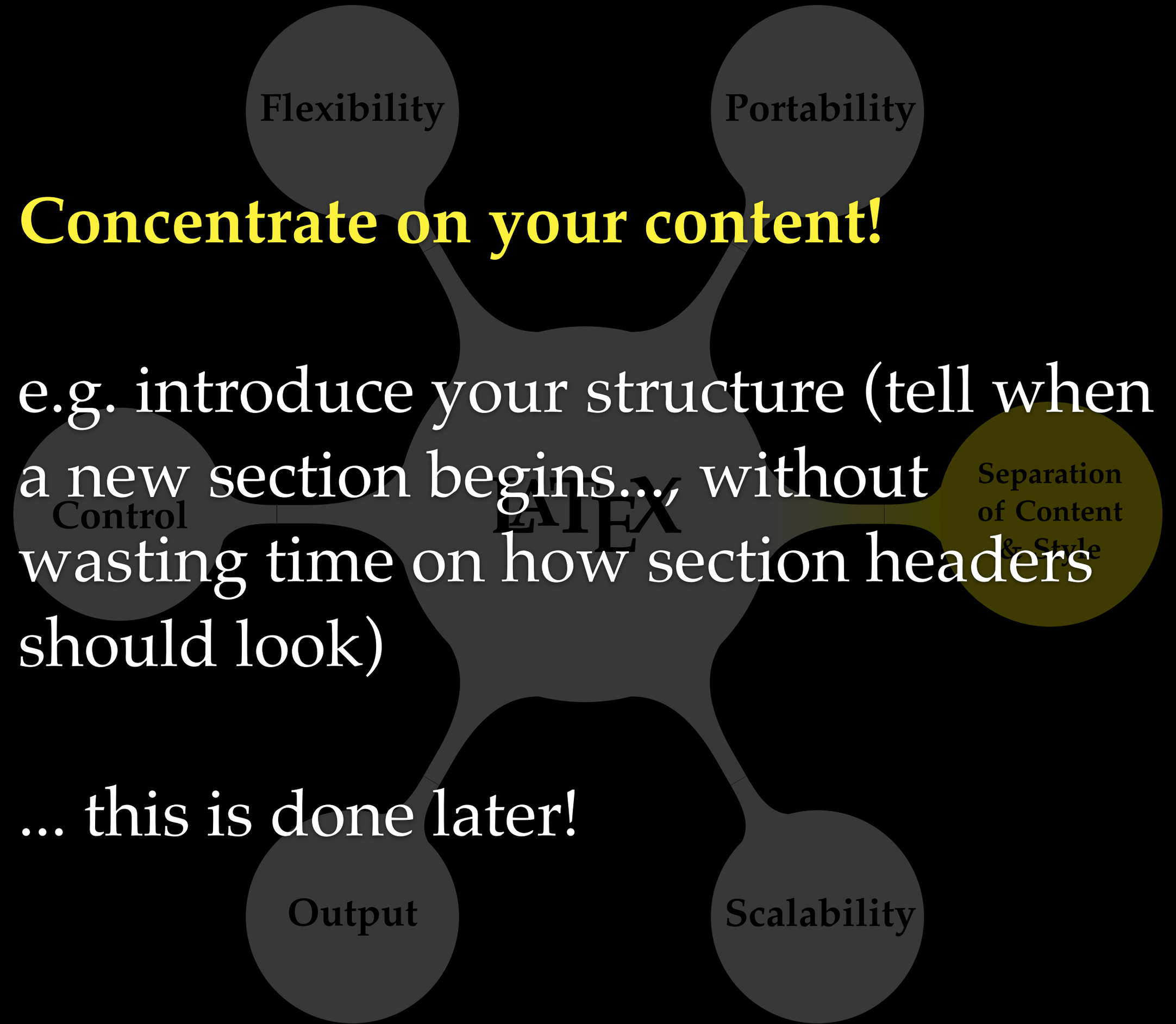
Portability

http://bit.ly/MediumLaTeX

Control

Separation of Content & Style

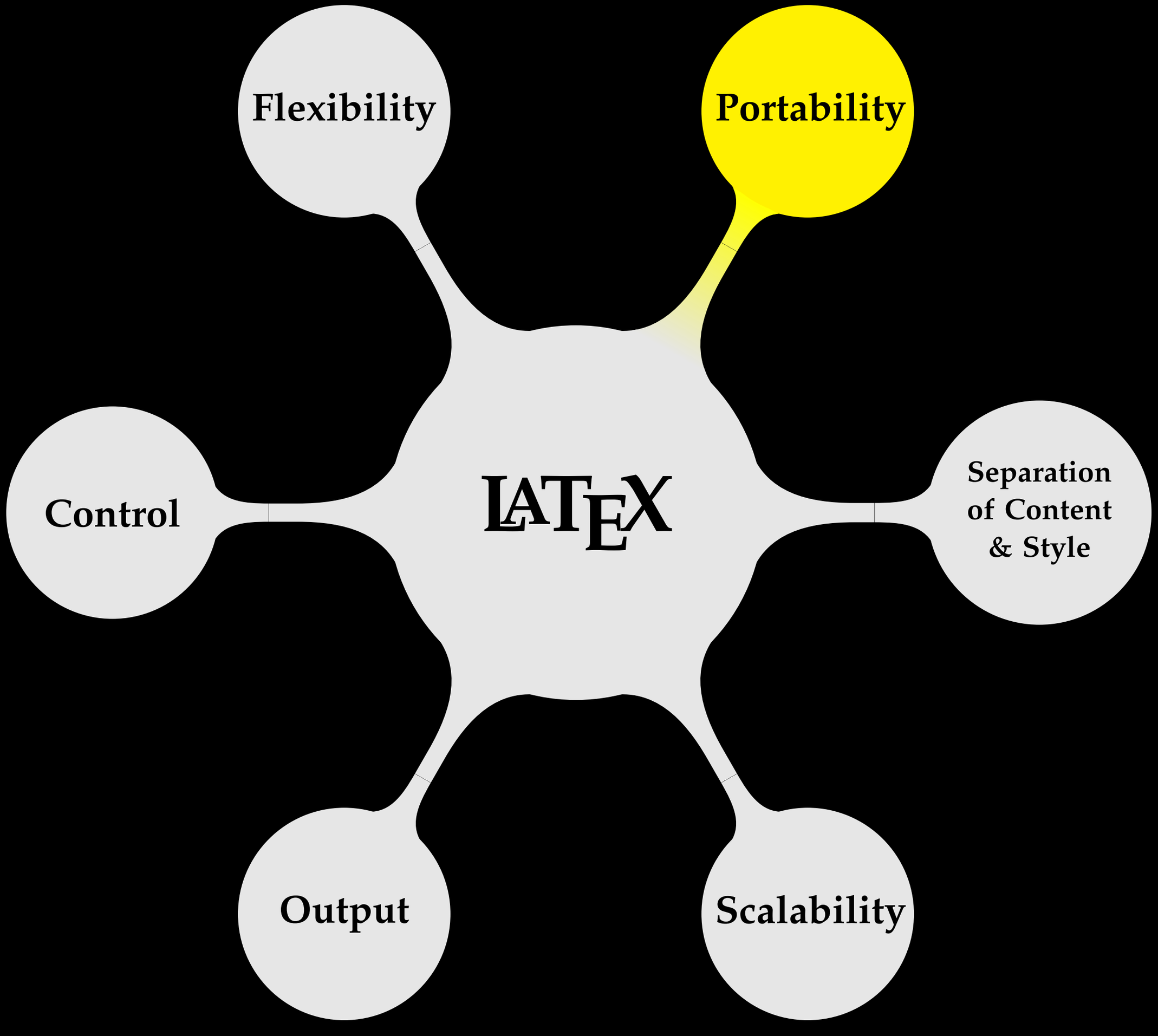

Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

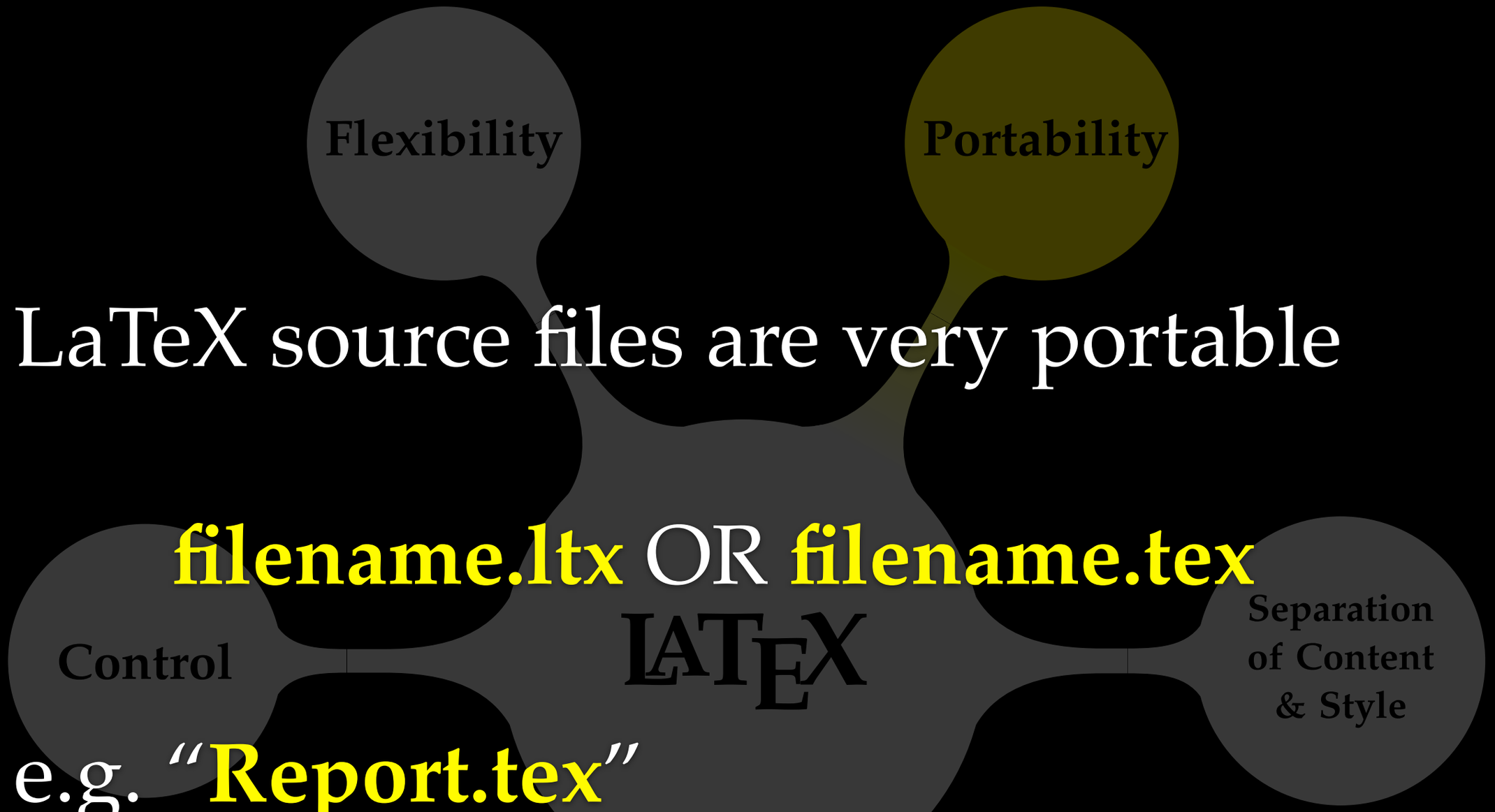


Concentrate on your content!

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

... this is done later!

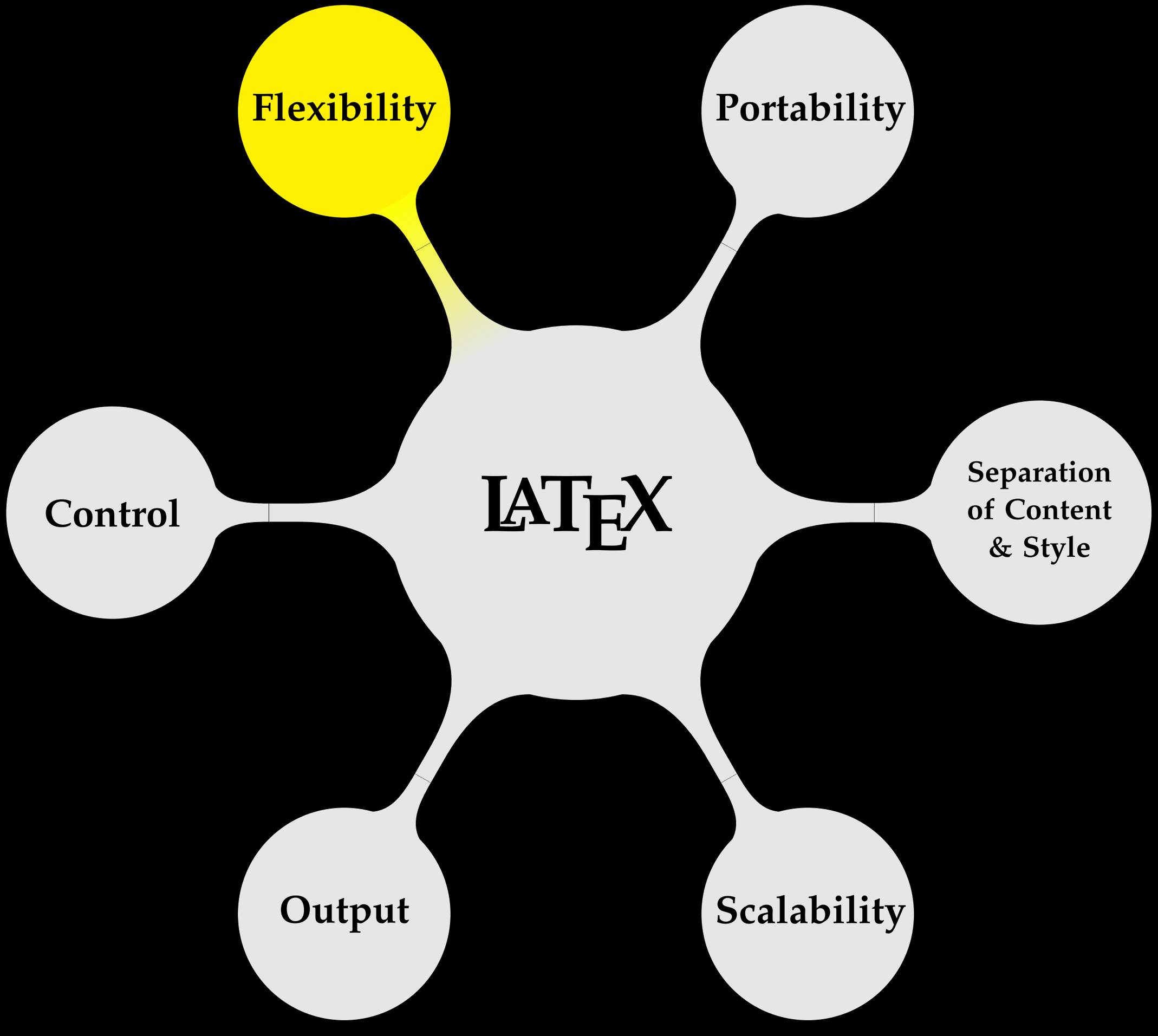

Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability



LaTeX source files are very portable

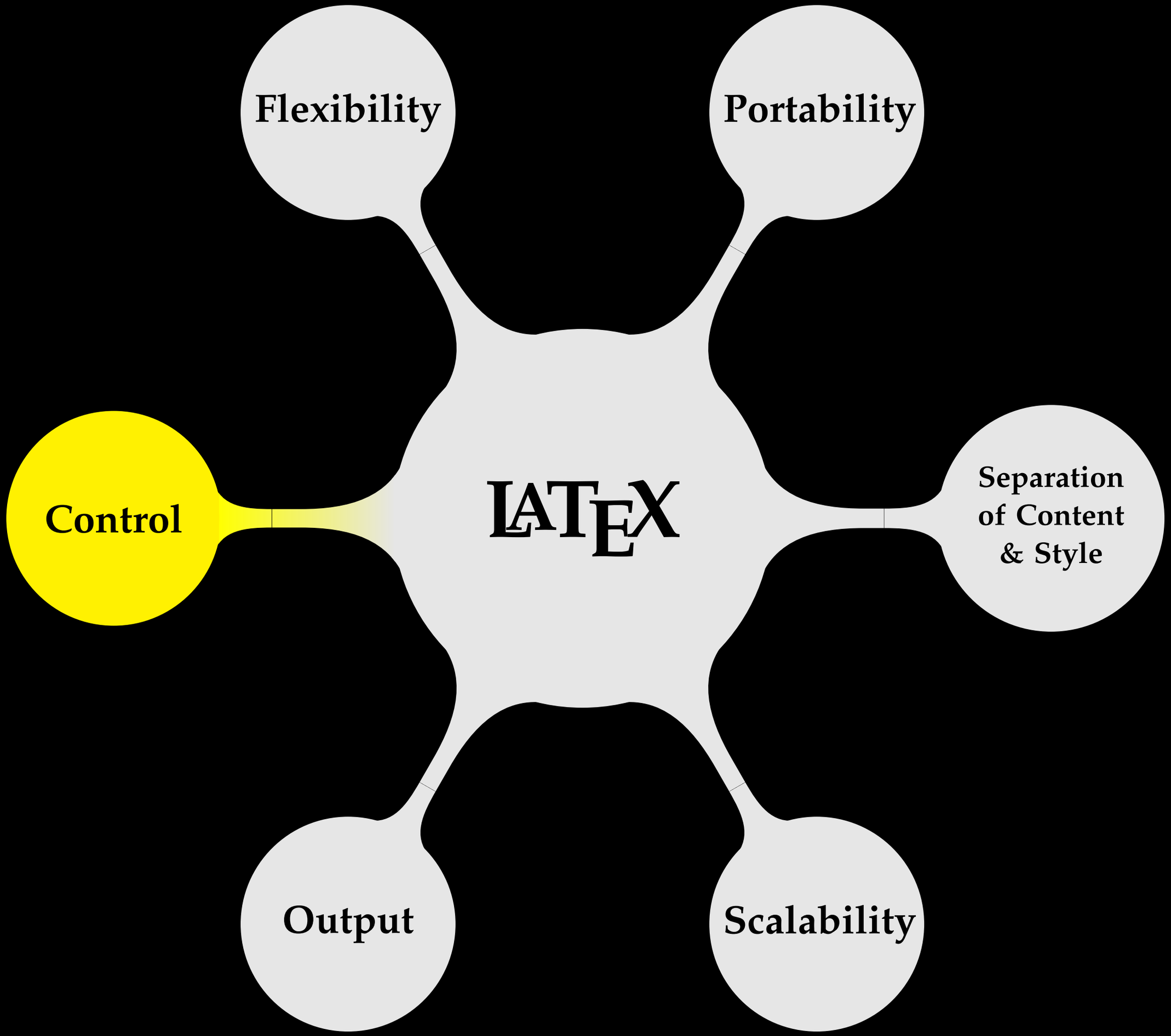
filename.ltx OR **filename.tex**

e.g. "**Report.tex**"
can be compiled on Windows, Linux,
and Mac systems


Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

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

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


Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

Flexibility

Portability

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

Control

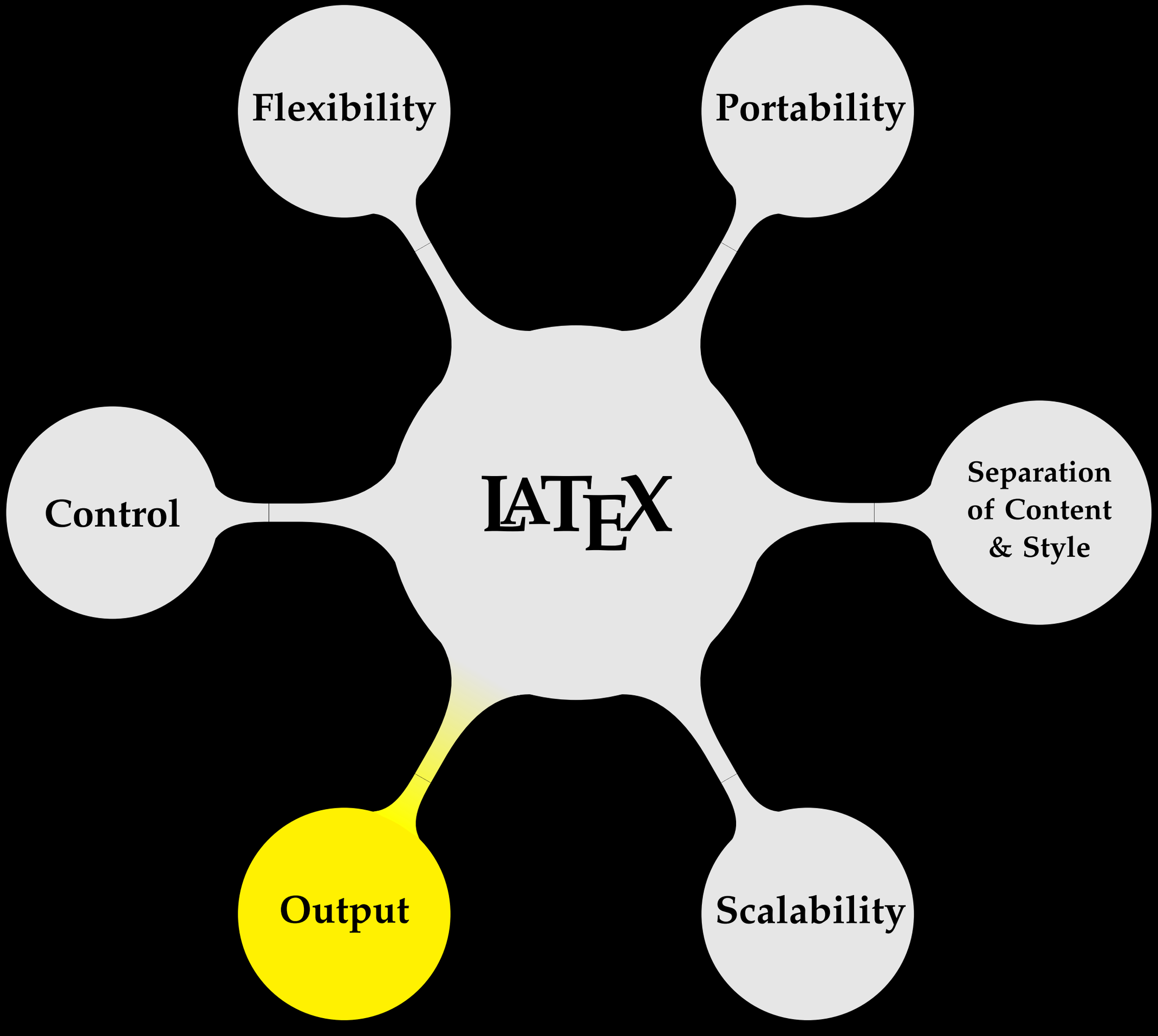
LaTeX

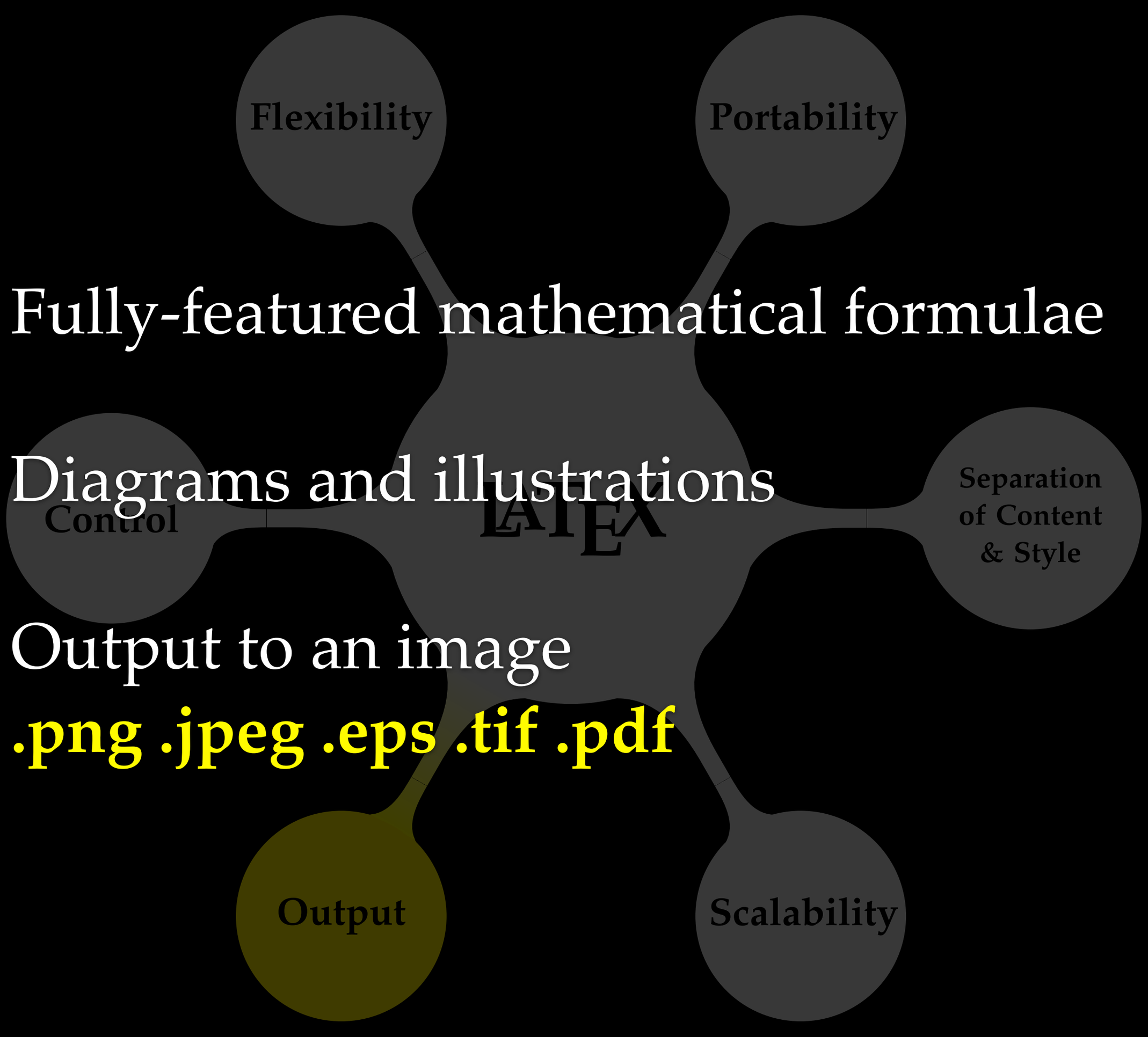
Separation of Content & Style

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

Output

Scalability


Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

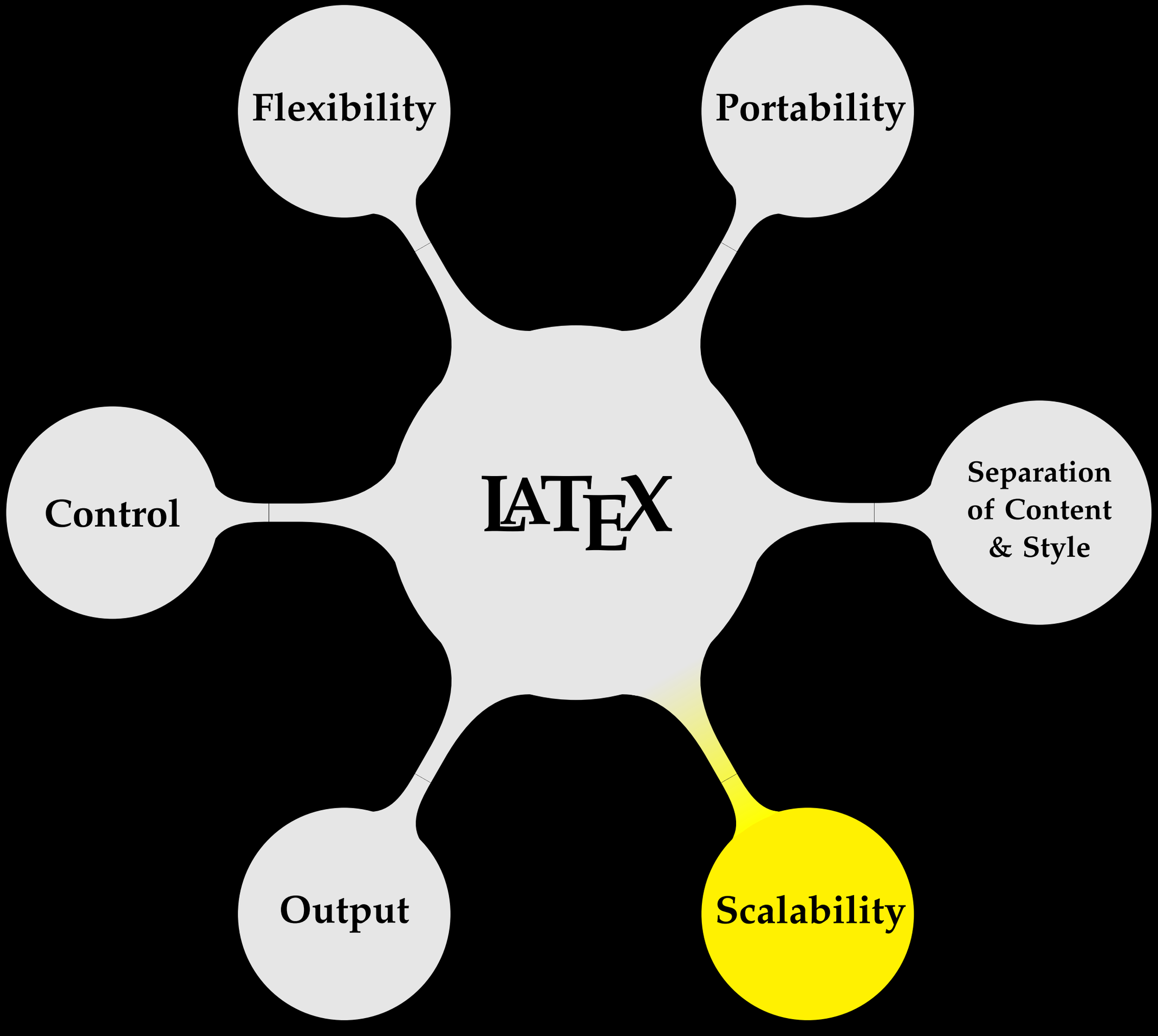


Fully-featured mathematical formulae

Diagrams and illustrations

Output to an image

.png .jpeg .eps .tif .pdf


Flexibility
Portability
Control
LaTeX
Separation of Content & Style
Output
Scalability

Flexibility

Portability

Very little risk of you ever losing your original sources

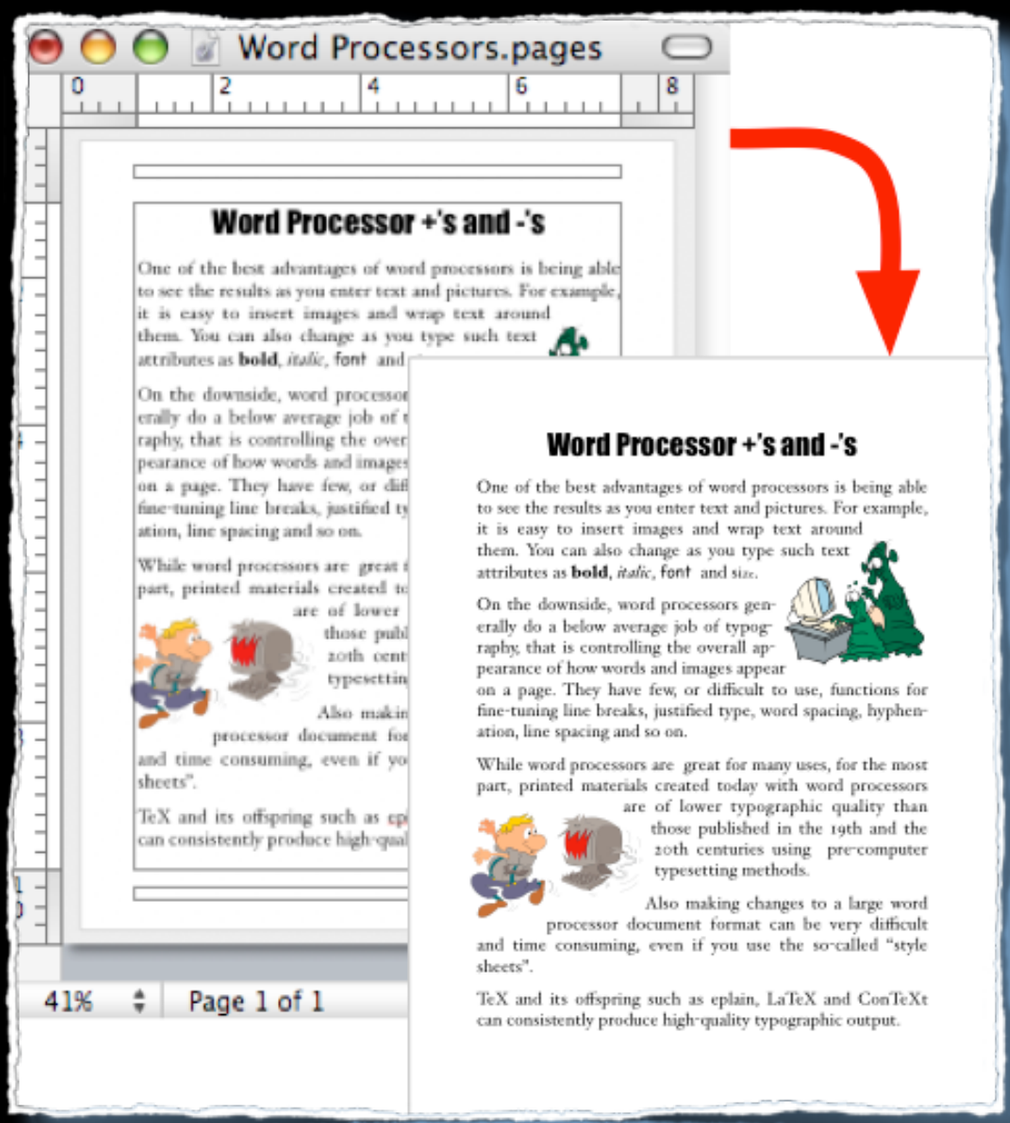
Separation of Content & Style

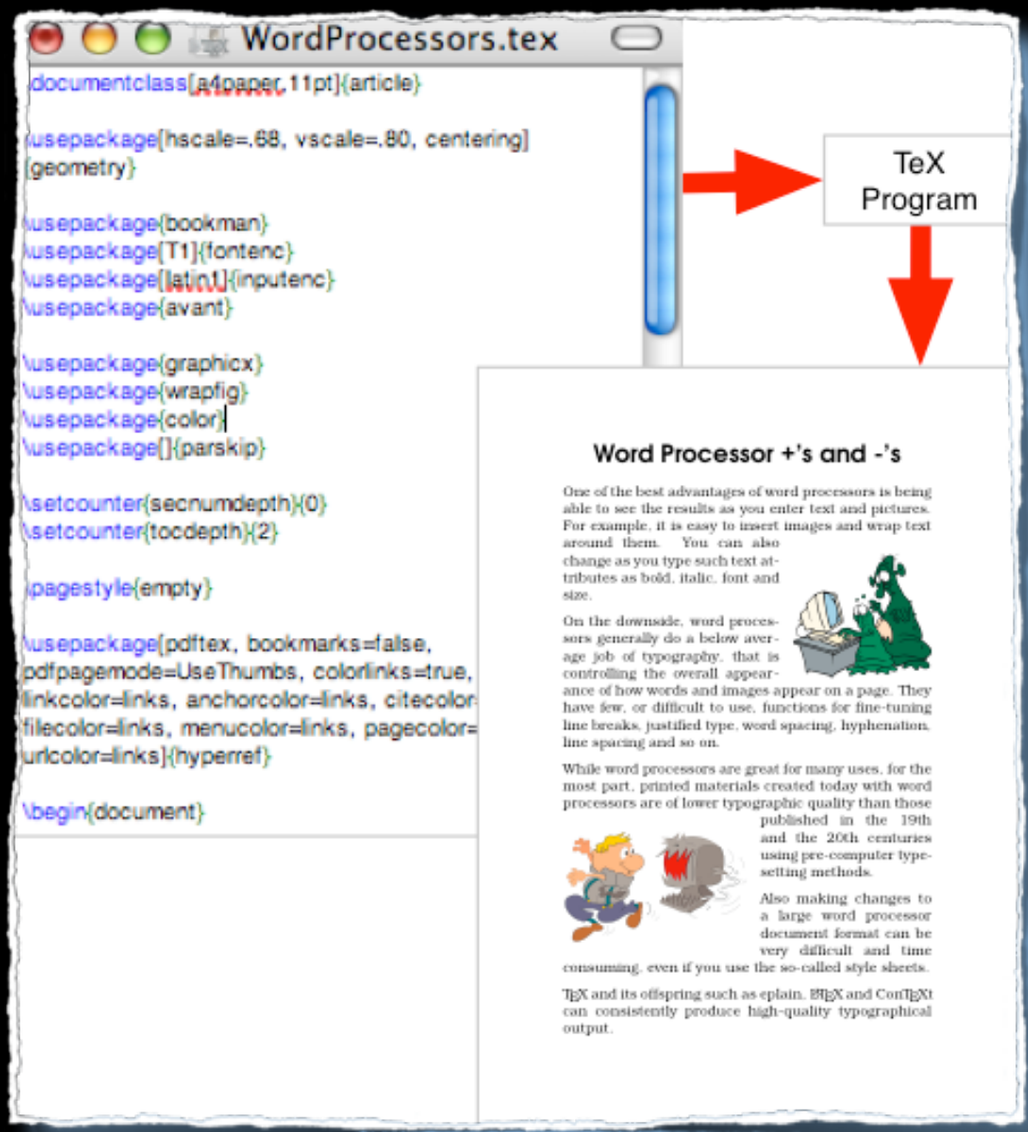
Very scalable with large "**sub-documents**" like in Word

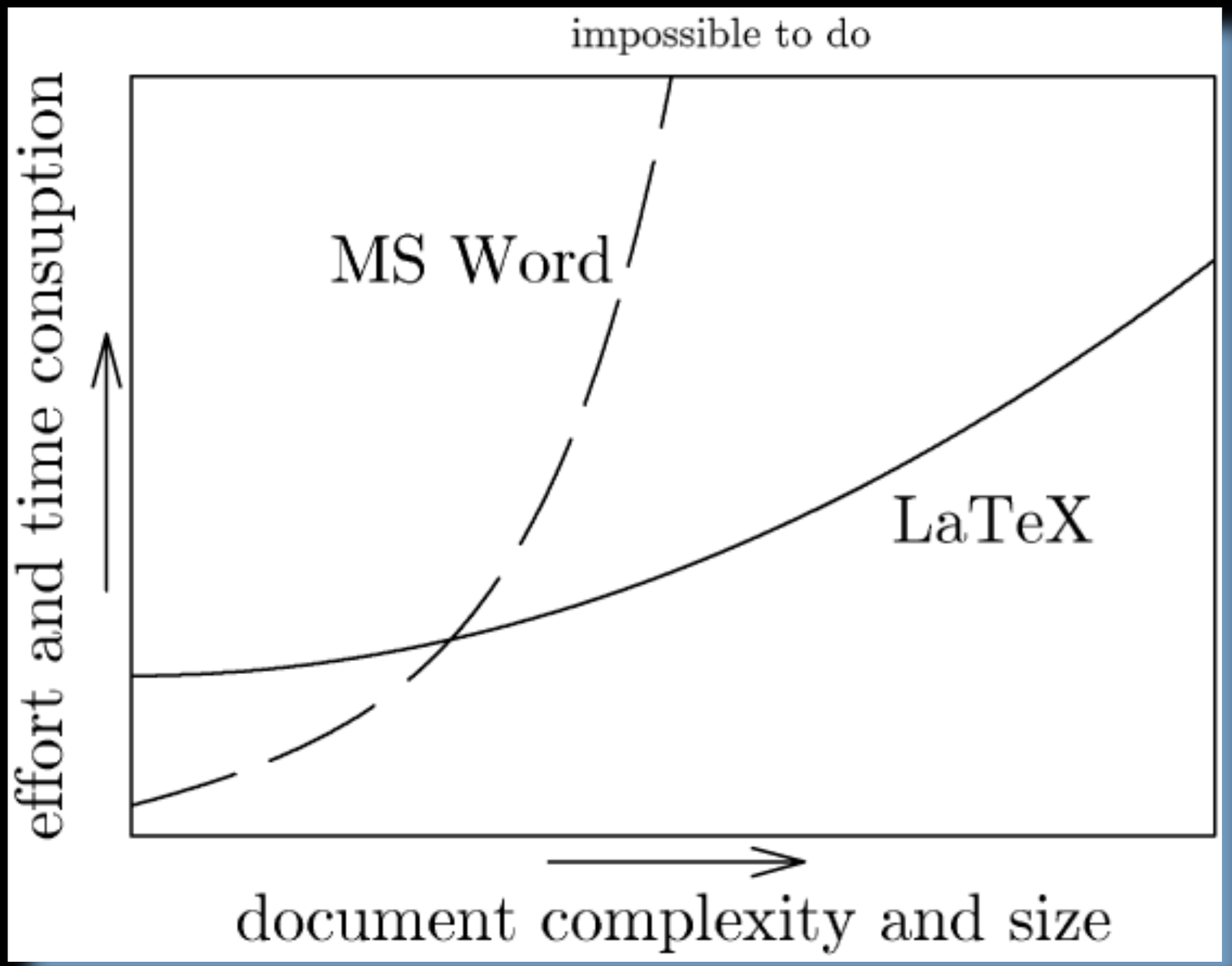
Never bother with MS Office viruses!

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
MS Word
LaTeX
effort and time consuption
document complexity and size

MiKTeX

EndNote

LaTeXiT

JabRef reference manager

BibDesk

TextMate

LyX

TextEdit

TeXShop

TeXworks

TeXnicCenter

LaTeX Equation Editor

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)).

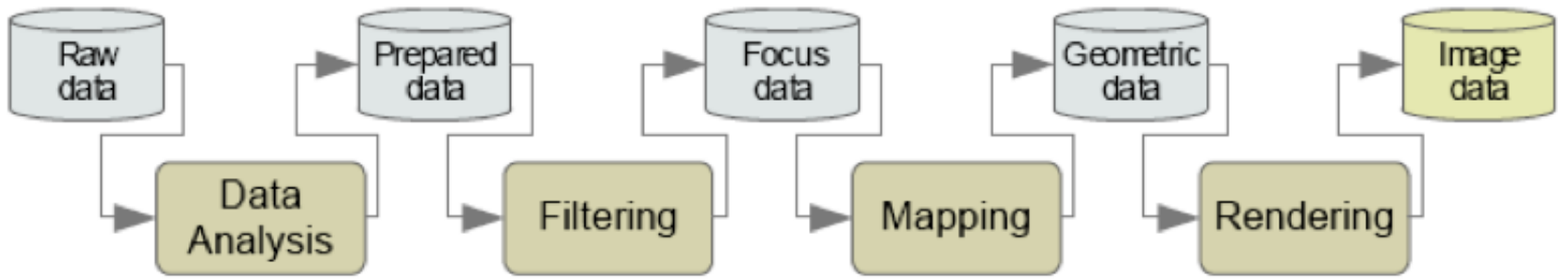


Figure 1.2: The Visualization Pipeline.

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:

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

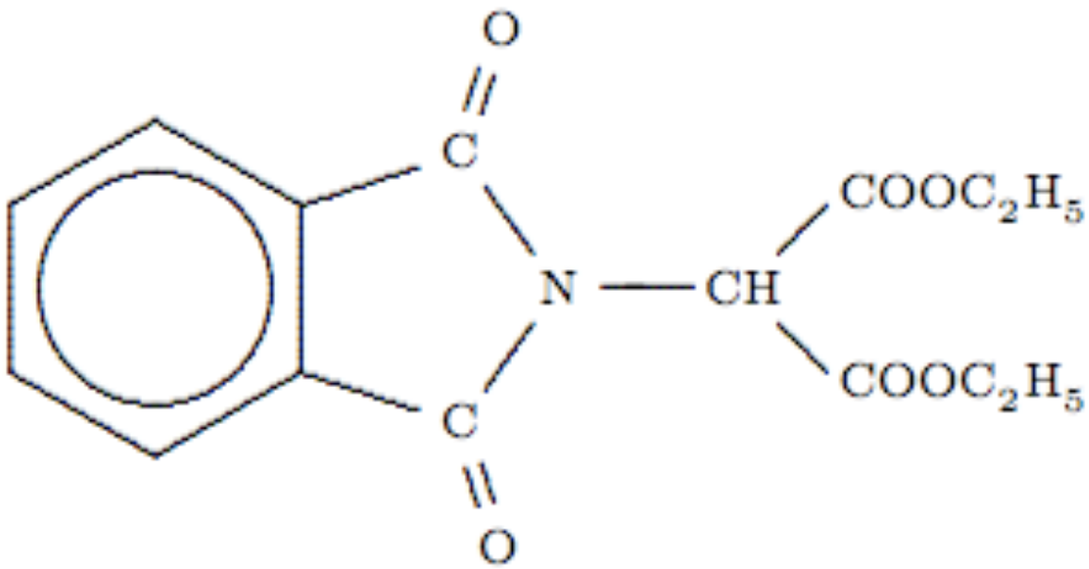
where N represents the number of original observations in the higher D space, and n on the other hand represents the number of projected observations in the lower d space. For instance, a transformation which maps the point (x, y, z) in $\Re^3$ to the point (x, y) in $\Re^2$ is a projection onto the 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

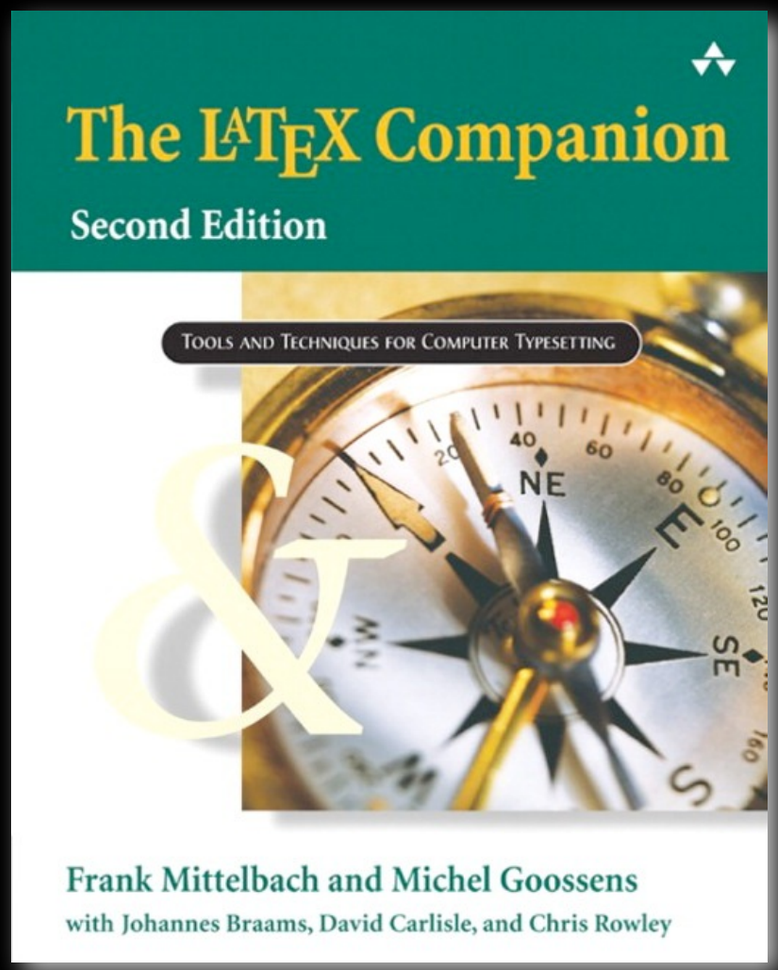
Arciniegas, I., Daniel, B. and Embrechts, M. J. (2001). Exploring financial crises data with self-organising maps (som), *in* N. Allinson, H. Yin, L. Allinson and S. J. (eds), *Advances in Self-Organising Maps*, Vol. 39-46, SpringerLink.

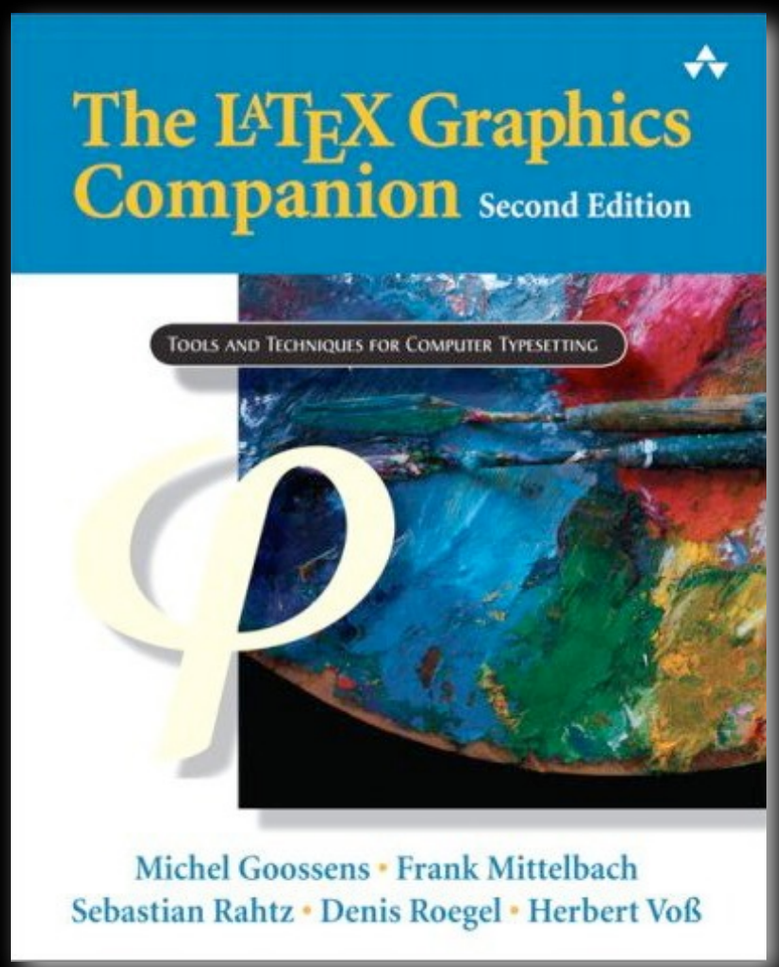
Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA.

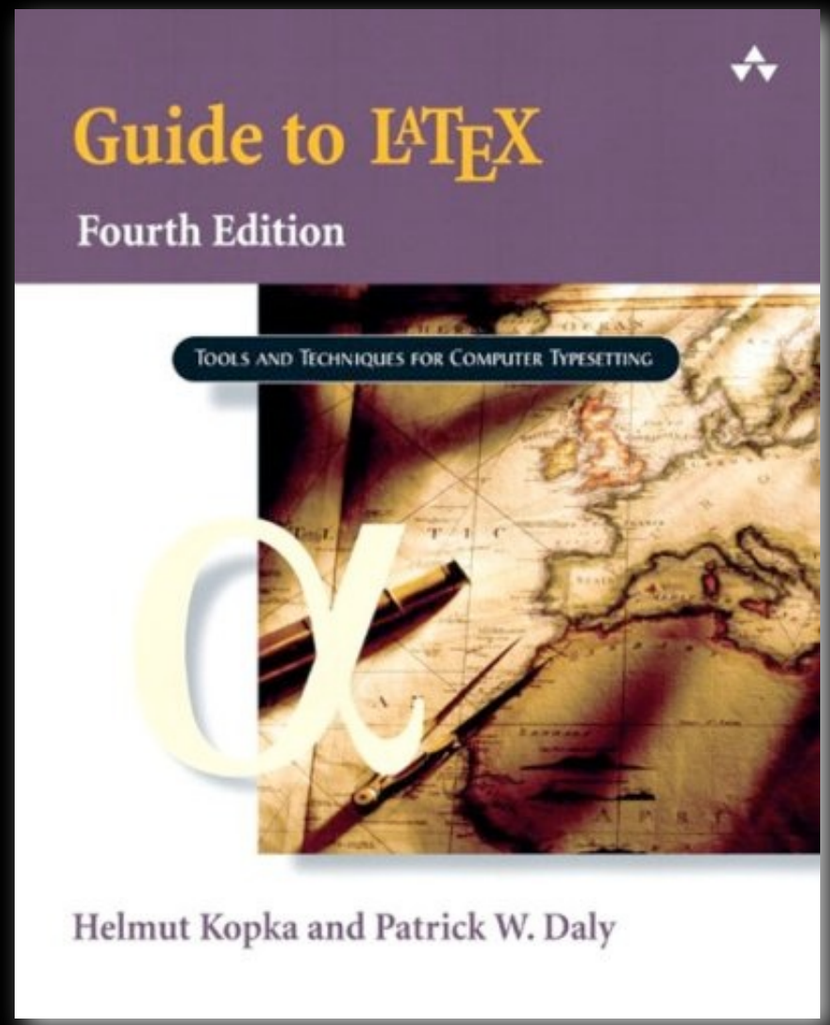
Bishop, C. M., Svensén, M. and Williams, C. K. I. (1998). Gtm: The generative topographic mapping, *Neural Computation* **10**: 215–235.

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


The LaTeX Companion
Second Edition
Tools and Techniques for Computer Typesetting
Frank Mittelbach and Michel Goossens
with Johannes Braams, David Carlisle, and Chris Rowley


The LaTeX Graphics Companion Second Edition
Tools and Techniques for Computer Typesetting
Michel Goossens • Frank Mittelbach
Sebastian Rahtz • Denis Roegel • Herbert Voß


Guide to LaTeX
Fourth Edition
Tools and Techniques for Computer Typesetting
Helmut Kopka and Patrick W. Daly

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