



WHY WE SHOULD USE TYPST AS ALTERNATION OF LATEX FOR TYPSETTING

CÓ PHẢI LATEX ĐÃ LỖI THỜI HAY?

Supervisor: LINH T. DƯƠNG
Student: DƯƠNG TUẤN LINH

June 10 2023

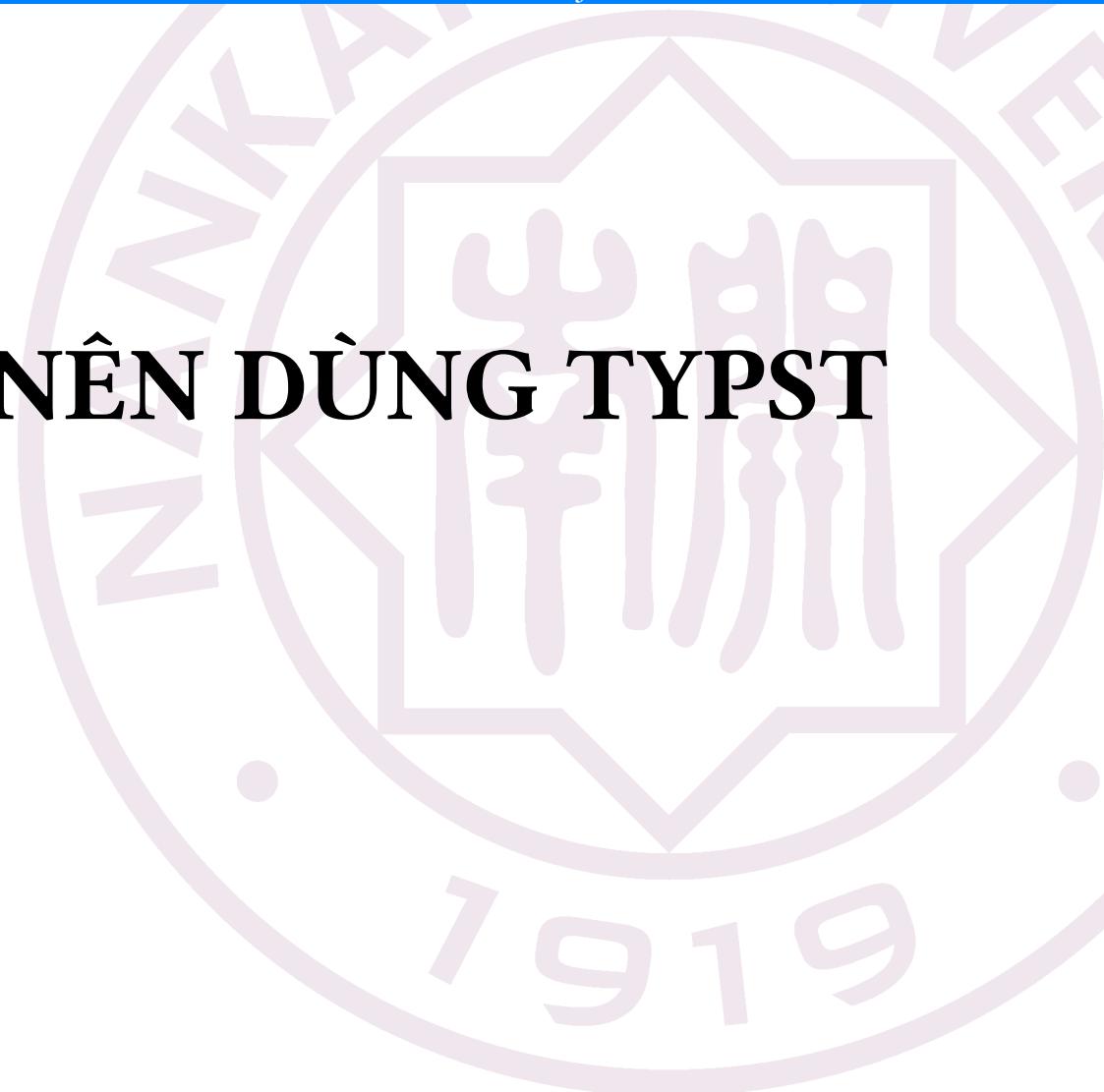
BÀI GIẢNG NÀY GỒM 2 PHẦN:

PHẦN 1. TẠI SAO NÊN DÙNG TYPST

**PHẦN 2. MỘT SỐ HƯỚNG DẪN CƠ BẢN
CHO THỦ VIỆN SOẠN THẢO SLIDE**



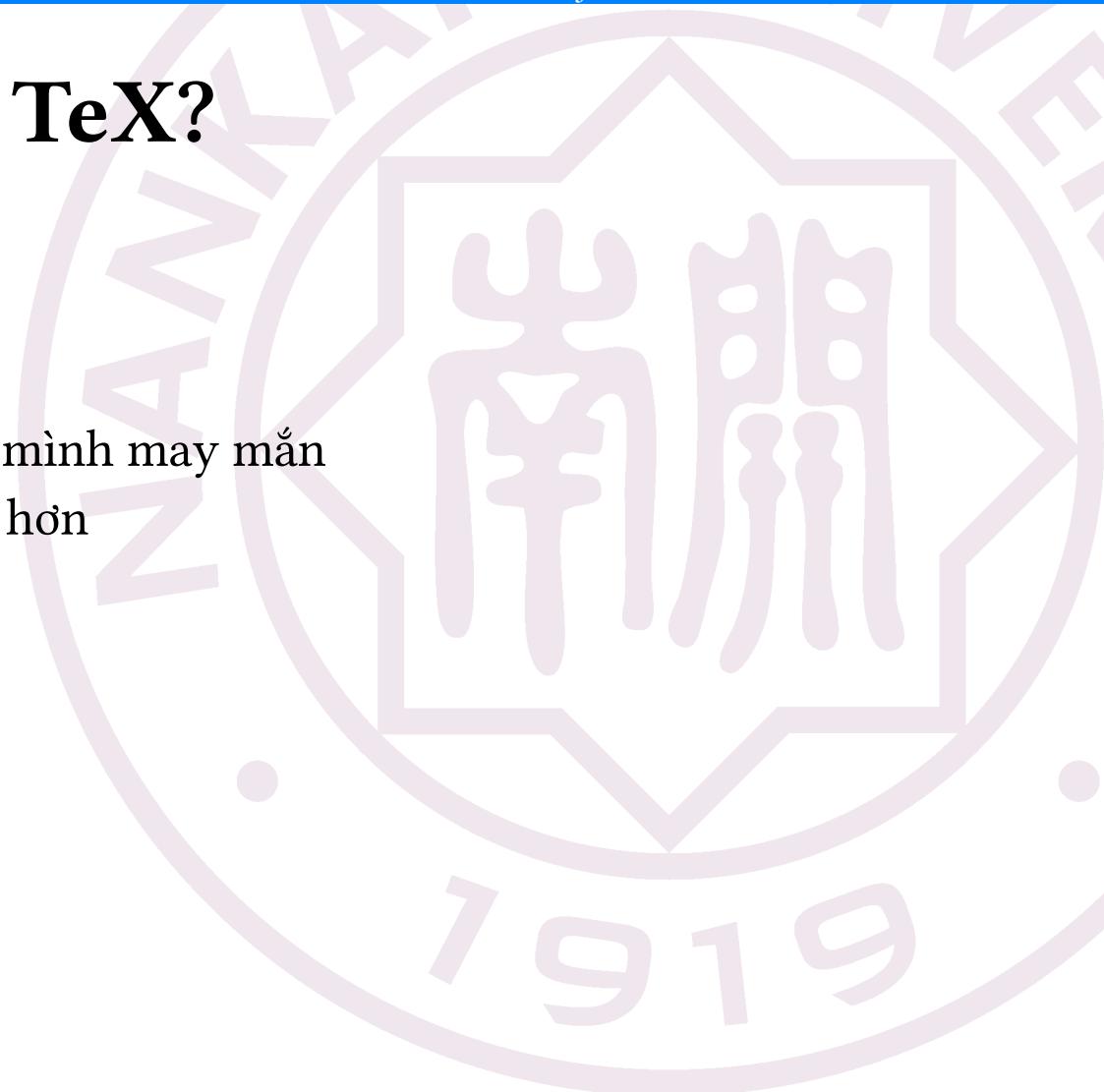
PHẦN 1. TẠI SAO NÊN DÙNG TYPST



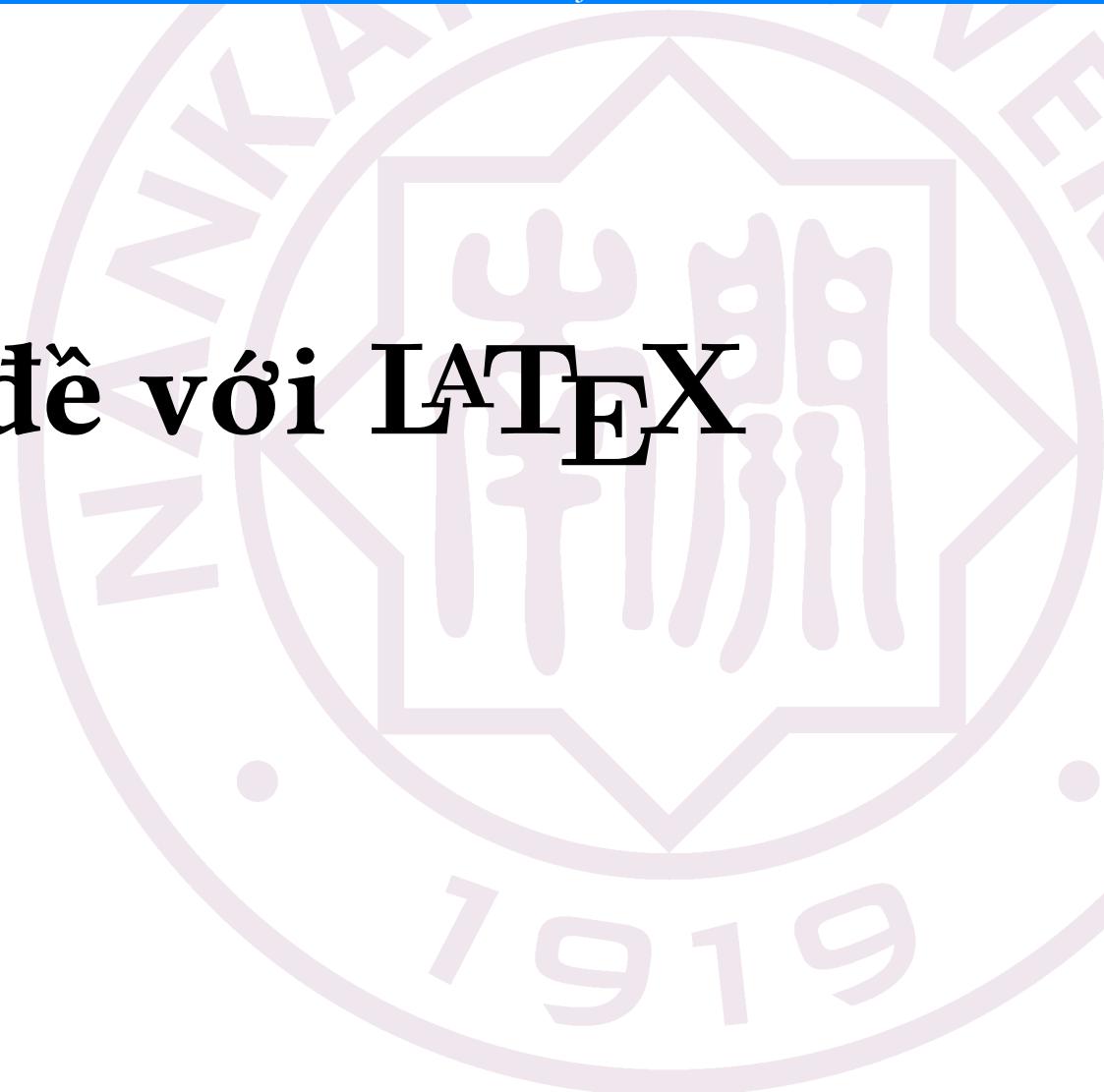


LUỢC SỬ PHÁT TRIỂN TeX?

- Ai đã từng sử dụng LaTeX? → Bạn sẽ thấy mình may mắn
- Ai có viết các API? → Làm cho nó dễ dàng hơn



Một số vấn đề với LATEX





Cha đẻ của L^AT_EX là ai????



Donald E. Knuth¹ (born. 10. Jan. 1938)

- Tác giả của TeX và METAFONT, phát triển từ 1977.
- Tác phẩm nổi tiếng nhất bên cạnh TeX: The Art of Computer Programming
- Đặc biệt phát triển TeX cho cuốn sách của anh ấy vì anh ấy có những yêu cầu thẩm mỹ đặc biệt mà các nhà xuất bản không đáp ứng được





Tiếp theo có những ai đóng góp vào sự phát triển của L^AT_EX?????



Leslie Lamport² (born 7. Feb. 1941)

- Tác giả của LaTeX, phát triển từ đầu những năm 1980
- LaTeX là tập hợp các macro để mở rộng và đơn giản hóa TeX





NHỮNG VẤN ĐỀ TỒN ĐỌNG CỦA LATEX

1. Kích thước chương trình lớn
2. Lựa chọn trình biên dịch
3. Lỗi khó hiểu





Kích thước của chương trình lớn?

```
% du -sch /usr/share/texmf-dist/* | sort -hr
2,5G    insgesamt
1,9G    /usr/share/texmf-dist/fonts
499M    /usr/share/texmf-dist/tex
58M     /usr/share/texmf-dist/scripts
44M     /usr/share/texmf-dist/tex4ht
24M     /usr/share/texmf-dist/bibtex
15M     /usr/share/texmf-dist/metapost
7,5M    /usr/share/texmf-dist/dvips
3,8M    /usr/share/texmf-dist/xindy
3,4M    /usr/share/texmf-dist/ls-R
2,6M    /usr/share/texmf-dist/asymptote
1,7M    /usr/share/texmf-dist/context
516K    /usr/share/texmf-dist/omega
344K    /usr/share/texmf-dist/makeindex
```

So với 21MB của trình biên dịch của Typst...

```
% du -sch /usr/bin/typst
21M    /usr/bin/typst
21M    insgesamt
```

- Kích thước cài đặt có thể thay đổi rất nhiều, 300MB..7GB, tuỳ vào các thư viện như TexStudio với texlive hay MiKTex,...
- vô cùng nhiều gói (đây vừa là thuận lợi nhưng cũng là khó khăn)





Sự đa dạng của chương trình TeX

“LATEX” không phải là một chương trình duy nhất mà còn:

- pdfLaTeX
- LuaTeX
- XeTeX
- MikTeX
- KaTeX
- ...





Thông báo lỗi của Typst, một số ví dụ điển hình

Typst:

\$a+b

```
error: expected dollar sign
└ test.typ:1:5
  1 | $a+b
    ^
```





Thông báo lỗi của LATEX

LATEX:

```
\documentclass{article}
```

```
\begin{document}
```

```
$a+b
```

```
\end{document}
```

```
Rc files read:  
/path/.config/latexmk/latexmkrc  
Latexmk: This is Latexmk, John Collins, 17 Mar. 2022. Version 4.77, version: 4.77.  
Latexmk: applying rule 'pdflatex'...  
Rule 'pdflatex': File changes, etc:  
    Changed files, or newly in use since previous run(s):  
    /path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex  
    test.tex  
Rule 'pdflatex': The following rules & subrules became out-of-date:  
    pdflatex  
-----  
Run number 1 of rule 'pdflatex'  
-----  
-----  
Running 'pdflatex -synctex=1 -interaction=nonstopmode -file-line-error -recorder "/path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex"'  
-----  
This is pdfTeX, Version 3.141592653-2.6-1.40.24 (TeX Live 2022/Arch Linux) (preloaded format=pdflatex)  
restricted \write18 enabled.  
entering extended mode  
(/path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex  
LaTeX2e <2021-11-15> patch level 1  
L3 programming layer <2022-04-10> (/usr/share/texmf-dist/tex/latex/base/article.cls  
Document Class: article 2021/10/04 v1.4n Standard LaTeX document class  
(/usr/share/texmf-dist/tex/latex/base/sizel0.clo)) (/usr/share/texmf-dist/tex/latex/l3backend/l3backend-pdftex.def) (./test.aux)  
/path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex:5: Missing $ inserted.  
<inserted text>  
$  
1.5  
[1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map}] (./test.aux)  
(see the transcript file for additional information)</usr/share/texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb>  
Output written on test.pdf (1 page, 13646 bytes).  
SyncTeX written on test.synctex.gz.  
Transcript written on test.log.  
Latexmk: If appropriate, the -f option can be used to get latexmk  
to try to force complete processing.  
Latexmk: Getting log file 'test.log'  
Latexmk: Examining 'test.fls'  
Latexmk: Examining 'test.log'  
Latexmk: Log file says output to 'test.pdf'  
Latexmk: Errors, so I did not complete making targets  
Collected error summary (may duplicate other messages):  
    pdflatex: Command for 'pdflatex' gave return code 1  
    Refer to 'test.log' for details
```





So sánh thông báo lỗi giữa Typst và LATEX

Typst	LATEX
#set par(leading: [Hello]) ~~~~~ expected integer, found content	\baselineskip=Hello Missing number, treated as zero. Illegal unit of measure (pt inserted).
#heading() ^^ missing argument: body	\section Missing \endcsname inserted. Missing \endcsname inserted. ...

(Nguồn: Mädje, Laurenz: “Typst – A Programmable Markup Language for Typesetting.” Luận văn thạc sĩ Đại học Kỹ thuật Berlin (Technische Universität, 2022.))



Giải pháp cho vấn đề này là
Typst





Typst



Martin Haug
(Người phát triển ứng dụng nền web)



Laurenz Mädje
(Người phát triển trình biên dịch)

<https://typst.app/about/>, (Cập nhật lần gần nhất: 26.05.2023, 10:16)

- Năm 2019, dự án được bắt đầu tại Technische Universität Berlin
- Lí do chính là sự thất vọng khi sử dụng L^AT_EX





Mục Đích của việc phát triển Typst

“Während bestehende Lösungen langsam, schwer zu bedienen oder einschränkend sind, ist Typst sorgfältig entworfen, um leicht erlernbar, flexibel und schnell zu sein. Dafür haben wir eine komplett eigene Markupsprache und Textsatzengine von Grund auf entwickelt. Dadurch sind wir in allen Bereichen des Schreib- und Textsatzprozesses innovationsfähig.”

1. Tạm dịch:

“Trong khi các giải pháp hiện có chậm chạp, khó sử dụng hoặc có hạn chế, Typst đã được thiết kế cẩn thận để dễ học, linh hoạt và nhanh chóng. Để làm được điều đó, chúng tôi đã phát triển một ngôn ngữ đánh dấu và hệ thống xuất bản văn bản hoàn toàn riêng từ đầu. Điều này giúp chúng tôi có khả năng đổi mới trong tất cả các khía cạnh của quá trình viết và xuất bản văn bản.”

<https://www.tu.berlin/entrepreneurship/startup-support/unsere-startups/container-profile/startups-2023-typst>, (Truy cập lần gần nhất: 03.05.2023, 10:13)





So sánh một chút giữa Typst và L^AT_EX

L^AT_EX

```
\documentclass{article}  
  
\begin{document}  
    \begin{enumerate}  
        \item Dies  
        \item Ist  
        \item Eine  
        \item Liste!  
    \end{enumerate}  
    \end{document}
```

Typst

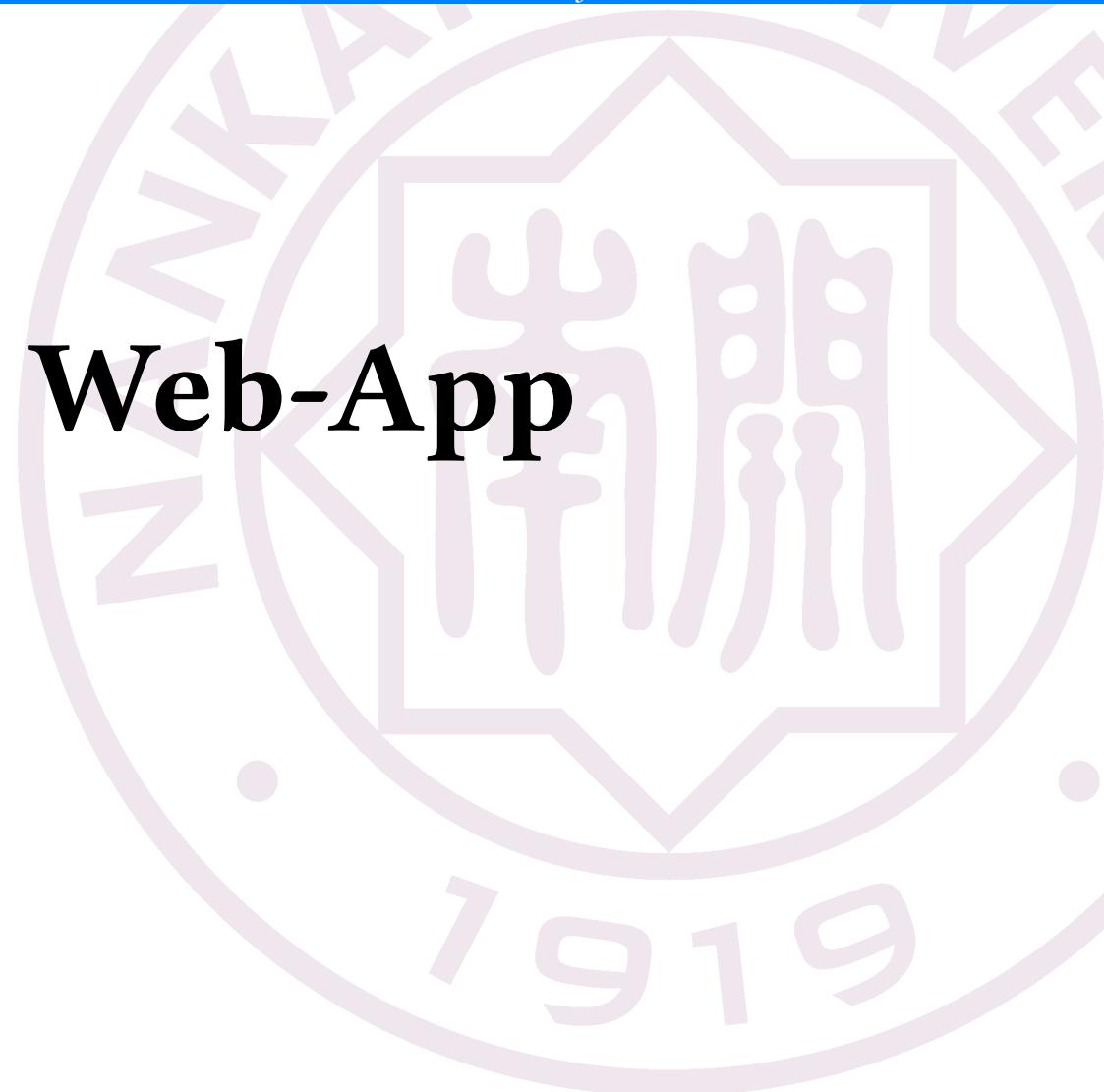
- + Cái này
- + Là
- + một
- + danh sách!

Kết quả

1. Cái này
2. Là
3. một
4. danh sách!



Ứng dụng Web-App





Nào chúng ta cùng tiến lên!

Thuận lợi:

- Tất cả các tập tin trực tuyến
- Các dự án khác nhau có thể được tạo ra
- Trình soạn thảo trực tuyến tốt
- Làm việc trên các tệp theo nhóm cùng một lúc
- Tài liệu tích hợp

<https://typst.app/>

Tài khoản tạm thời (với $1 \leq N \leq 15$):

- E-Mail: typstseminarN@byom.de
- Passwort: typstseminarN

Đường dẫn tới: <https://github.com/linhduongtuan/DTU-typst-presentation>

- Mở trình duyệt và trang web
- Tạo tài khoản của riêng bạn



Duong

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

19

Định dạng trang cơ bản





Định dạng trang cơ bản

- Tiêu đề
- Đậm, nghiêng
- danh sách
- Những bức ảnh
- Đặt và hiển thị các quy tắc
- Thay đổi kích thước phông chữ và màu sắc
- Căn chỉnh văn bản
- Viết công thức Toán học (tài liệu có ký hiệu)





Heading level

= **Heading 1!**

== **Heading 2!**

==== **Heading 3!**

Text ~ Câu chữ

Đoạn văn mới!

1. **Heading 1!**

1.1. **Đầu mục 2!**

1.1.1. **Đầu mục 3!**

Text

Đoạn văn mới!





Đoạn văn

= Tiêu đề`thú của tôi

Một đoạn mới bắt đầu trong Typst ngay khi có một dòng trống trong mã.

Thật không may, theo mặc định, các đoạn văn được căn trái, không được căn đều. Chúng ta sắp học cách thay đổi điều đó!

1. Tổng Quan Tài Liệuj

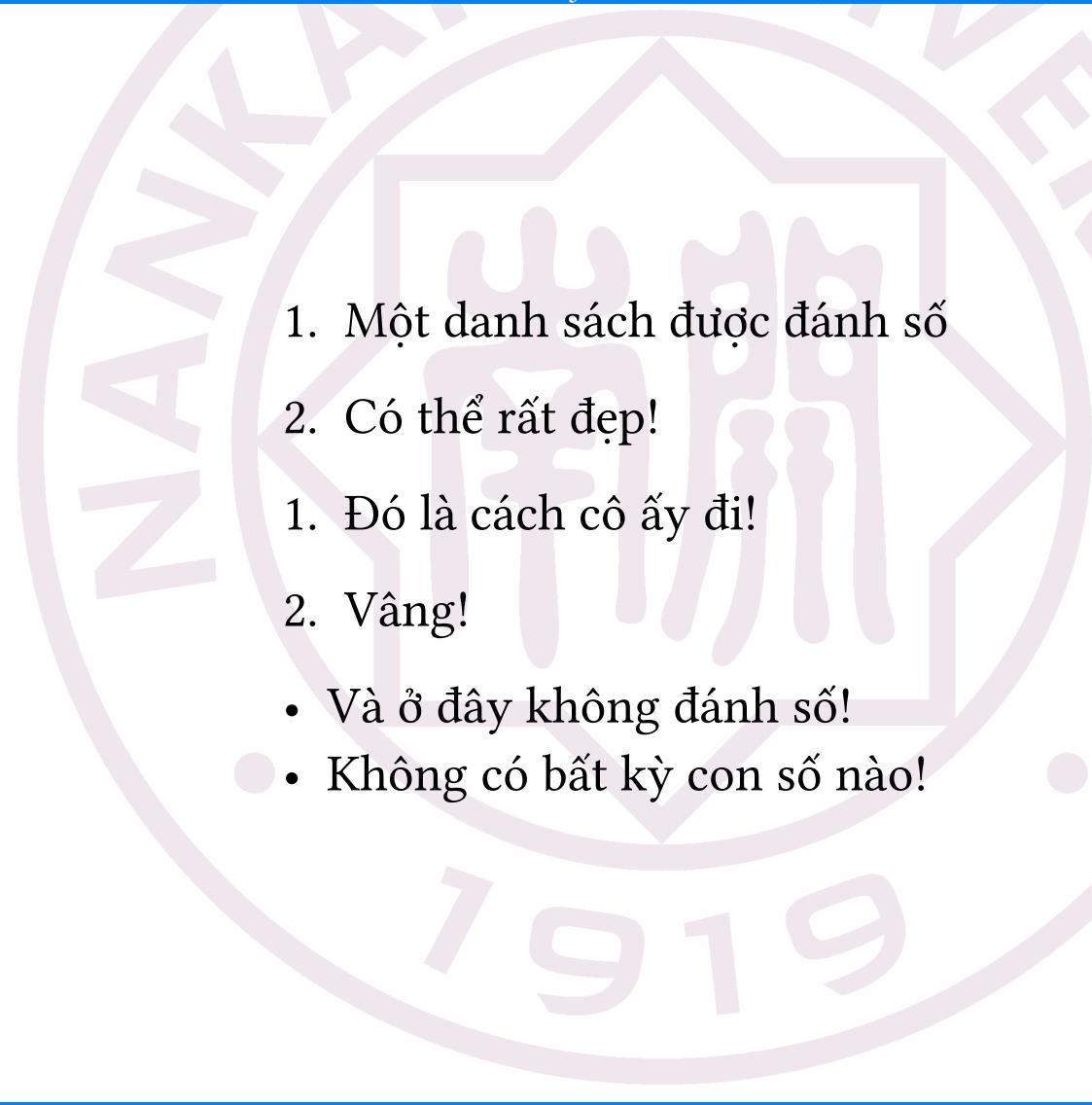
- Một đoạn mới bắt đầu trong Typst ngay khi có một dòng trống trong mã.
- Thật không may, theo mặc định, các đoạn văn được căn trái, không được căn đều. Chúng ta sắp học cách thay đổi điều đó!





Chú ý

- + Một danh sách được đánh số!
 - + Có thể rất đẹp!
1. Đó là cách cô ấy đi!
 2. Vâng!
- Và ở đây không đánh số!
 - Không có bất kỳ con số nào!

- 
1. Một danh sách được đánh số
 2. Có thể rất đẹp!
1. Đó là cách cô ấy đi!
 2. Vâng!
- Và ở đây không đánh số!
 - Không có bất kỳ con số nào!





Kiểu font chữ

```
#text(font: "Arial", [Xin chào!])
```

```
#text(font: "Courier New", [Xin chào!])
```

```
#text(font: "New Computer Modern", [Xin chào!])
```





Nội dung, chuỗi, tập lệnh

- Có hai loại nội dung trong Typst:
 - Nội dung trong [...]
 - Tập lệnh trong {...}
 - từ nội dung đến tập lệnh với #
 - Mọi tài liệu về cơ bản là nội dung

```
#strong([Đây là nội dung trong tập lệnh!])
```

```
#{
    strong([In đậm!])
    [Một lần nữa, đây là nội dung trong tập
lệnh!]
}

#{ 3/4 }
```

Đây là nội dung.

In đậm!Một lần nữa, đây là nội dung trong tập lệnh!

0.75





Định dạng font chữ⁴

Xin chào! **#strong**([Xin chào!])

Xin chào! **#emph**([Xin chào!])

Xin chào! **#super**([Xin chào!])

Xin chào! **#sub**([Xin chào!])

#text(fill: red, [Xin chào chữ màu đỏ!])

#text(fill: **rgb**("#ff00ff"), [Xin chào chữ màu hông!])

#text(fill: **rgb**("#ff00ff"), **strong**([Xin chào chữ màu hông!]))



Xin chào! Xin chào!

Xin chào! Xin chào!

Xin chào!^{Xin chào!}

Xin chào!^{Xin chào!}

Xin chào chữ màu đỏ!

Xin chào chữ màu hồng!

Xin chào chữ màu hồng!





Định dạng vị trí chữ

```
#align(left, [Xin chào!])
#align(center, [Xin chào!])
#align(right, [Xin chào!])
#align(right, strong([Xin chào!]))
```





Câu hỏi 1

Làm thế nào để bạn thực hiện những điều sau đây trong Typst?

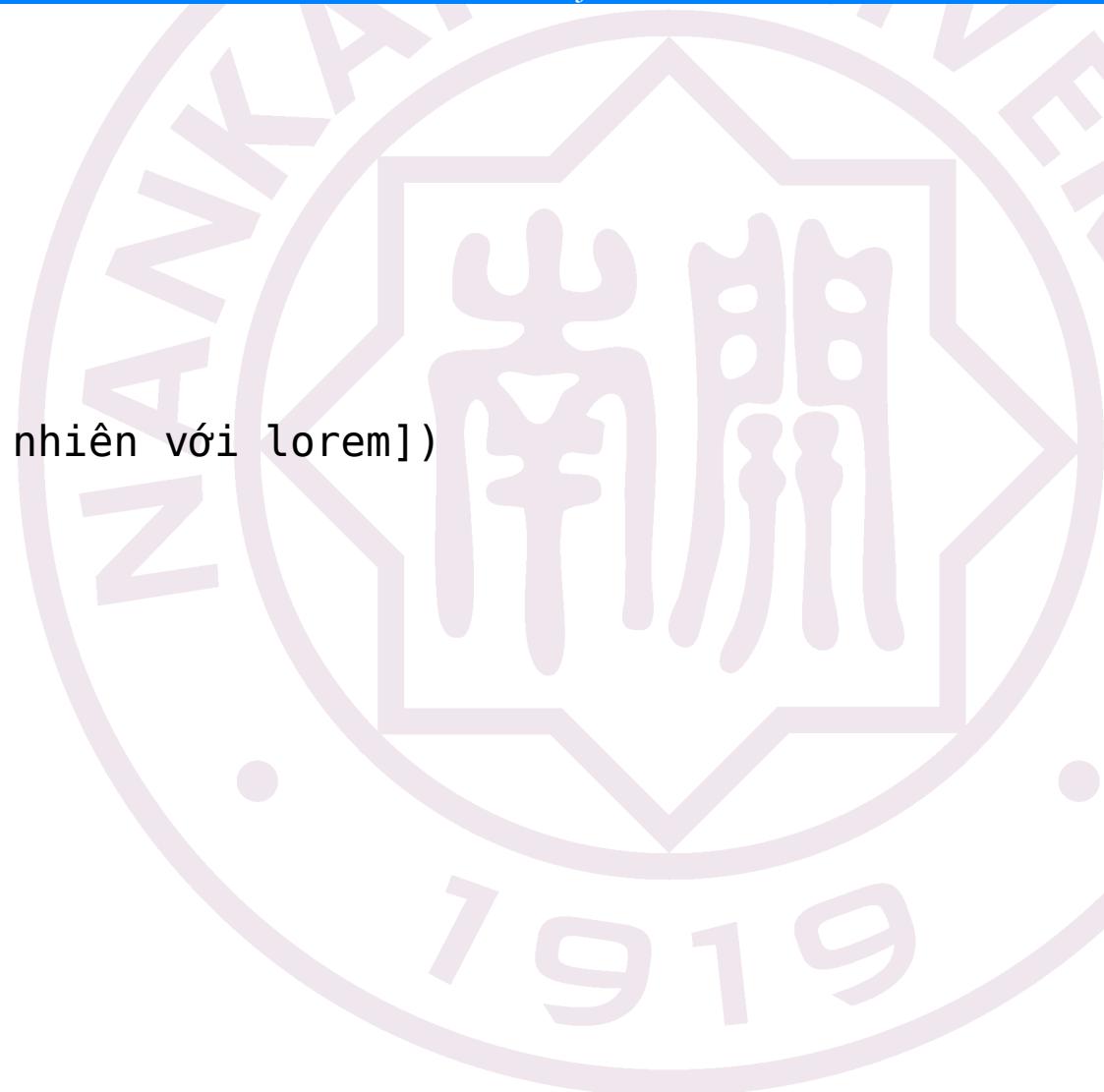
Trích đoạn từ ngẫu nhiên với lorem

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliquam quaerat.



Cách thực hiện 1

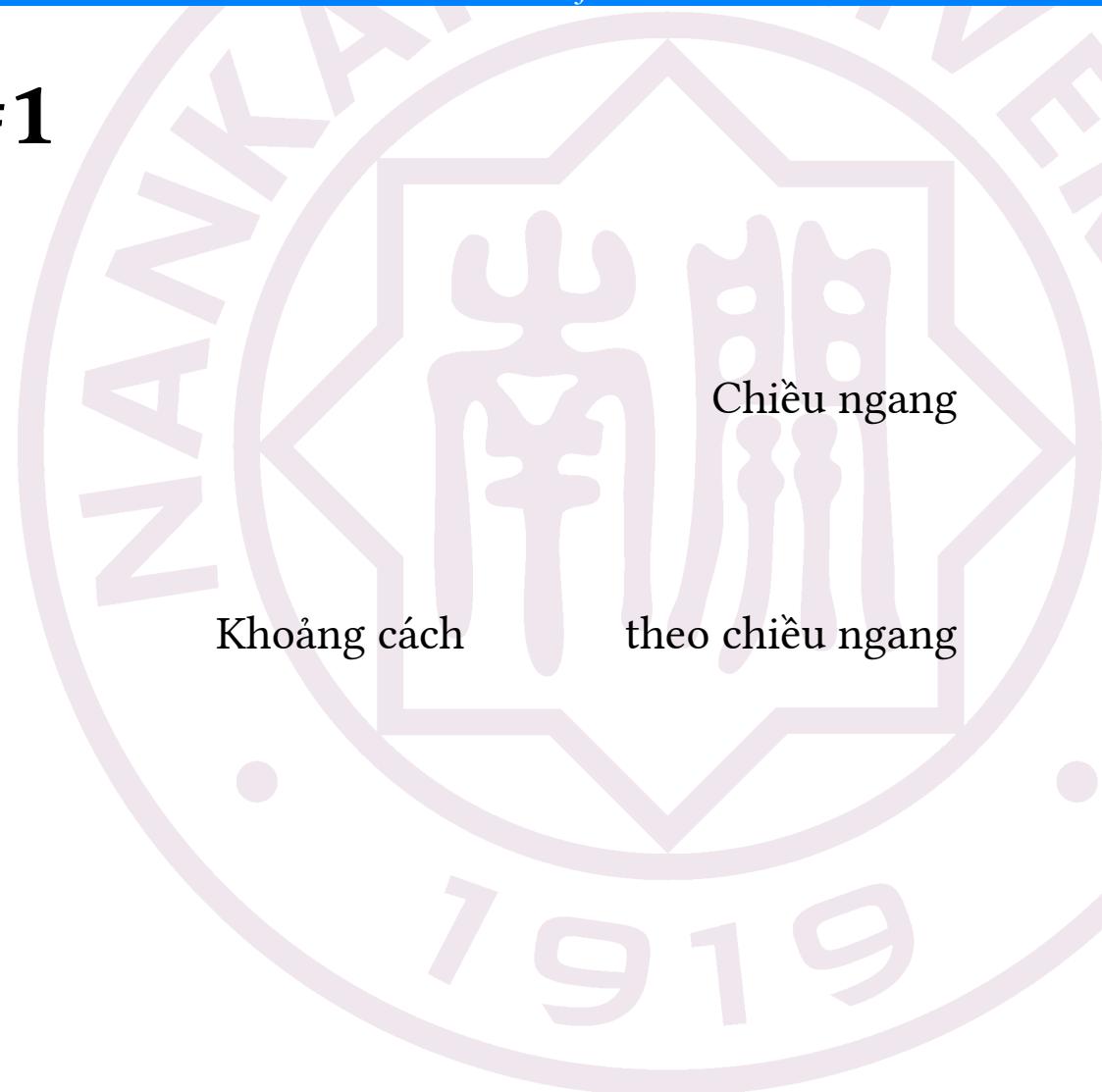
```
#align(center, emph[Trích đoạn từ ngẫu nhiên với lorem])  
#lorem(20)
```





Thiết lập khoảng cách #1

```
#align(right, [Chiều ngang])
#v(2cm)
Khoảng cách #h(2cm) theo chiều ngang
```





Thiết lập khoảng cách (Đọc)

Liệt kê ví dụ từ slide 21 với khoảng cách đọc.

- + Một danh sách được đánh số'
 - + Có thể' rất đẹp!
- #v(2em)
1. Đó là cách cô ấy đi!
 2. Vâng!
- #v(2em)
- Và ở đây không đánh số!
 - Không có bất kỳ con số nào!

Liệt kê ví dụ từ slide 21 với khoảng cách đọc.

1. Một danh sách được đánh số
 2. Có thể rất đẹp!
1. Đó là cách cô ấy đi!
2. Vâng!
- Và ở đây không đánh số!
 - Không có bất kỳ con số nào!





Thiết lập khoảng cách (Ngang)

- + Mọi đức tính hoàn hảo đều thuộc về Đức Chúa Trời vì bản chất hoàn hảo của Ngài.
- + Tồn tại hoàn hảo hơn là không tồn tại.
- + Vì vậy sự tồn tại là một tài sản hoàn hảo.
- + Vậy là có Chúa. #h(1fr) QED

1. Mọi đức tính hoàn hảo đều thuộc về Đức Chúa Trời vì bản chất hoàn hảo của Ngài.
2. Tồn tại hoàn hảo hơn là không tồn tại.
3. Vì vậy sự tồn tại là một tài sản hoàn hảo.
4. Vậy là có Chúa.

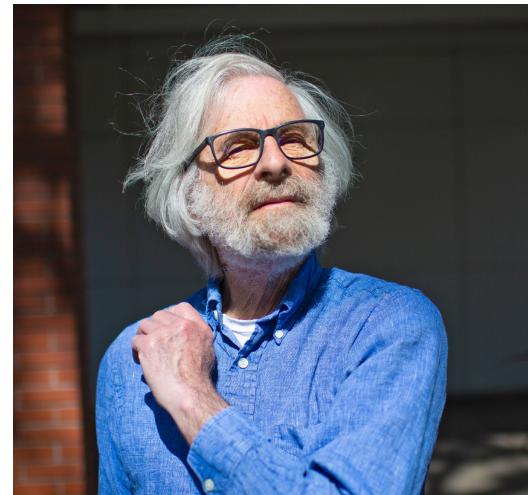
QED





Với ảnh

```
#image(height: 50%, "leslie_lamport.png")
```





#block() và #box()

Điều này có thể được thực hiện với `#block`: `#block(stroke: black, inset: 0.5em, [`#block()`]` tạo một dòng mới và có thể được ngắt giữa các trang. Nó có nhiều đối số tùy chọn.)
Đó là cách nó trông ra sao.

Điều này có thể được thực hiện với #block:

`#block()` tạo một dòng mới và có thể được ngắt giữa các trang. Nó có nhiều đối số tùy chọn.

Đó là cách nó trông ra sao.





#block() và #box()

Mặt khác: `#box(stroke: black, inset: 2pt, [#box()])` tạo ngắt dòng `#box(stroke: (bottom: black), inset: 2pt, [không])` và cho phép `#box(stroke: (bottom: black), inset: 2pt, [không])` ngắt giữa các trang. Nhưng bạn có thể sử dụng nó để đóng khung mọi thứ chặng hạn. Để gạch chân, bạn nên sử dụng `#underline[#underline]` thay thế.

Mặt khác: `#box()` tạo ngắt dòng không và cho phép không ngắt giữa các trang. Nhưng bạn có thể sử dụng nó để đóng khung mọi thứ chặng hạn. Để gạch chân, bạn nên sử dụng `#underline` thay thế.

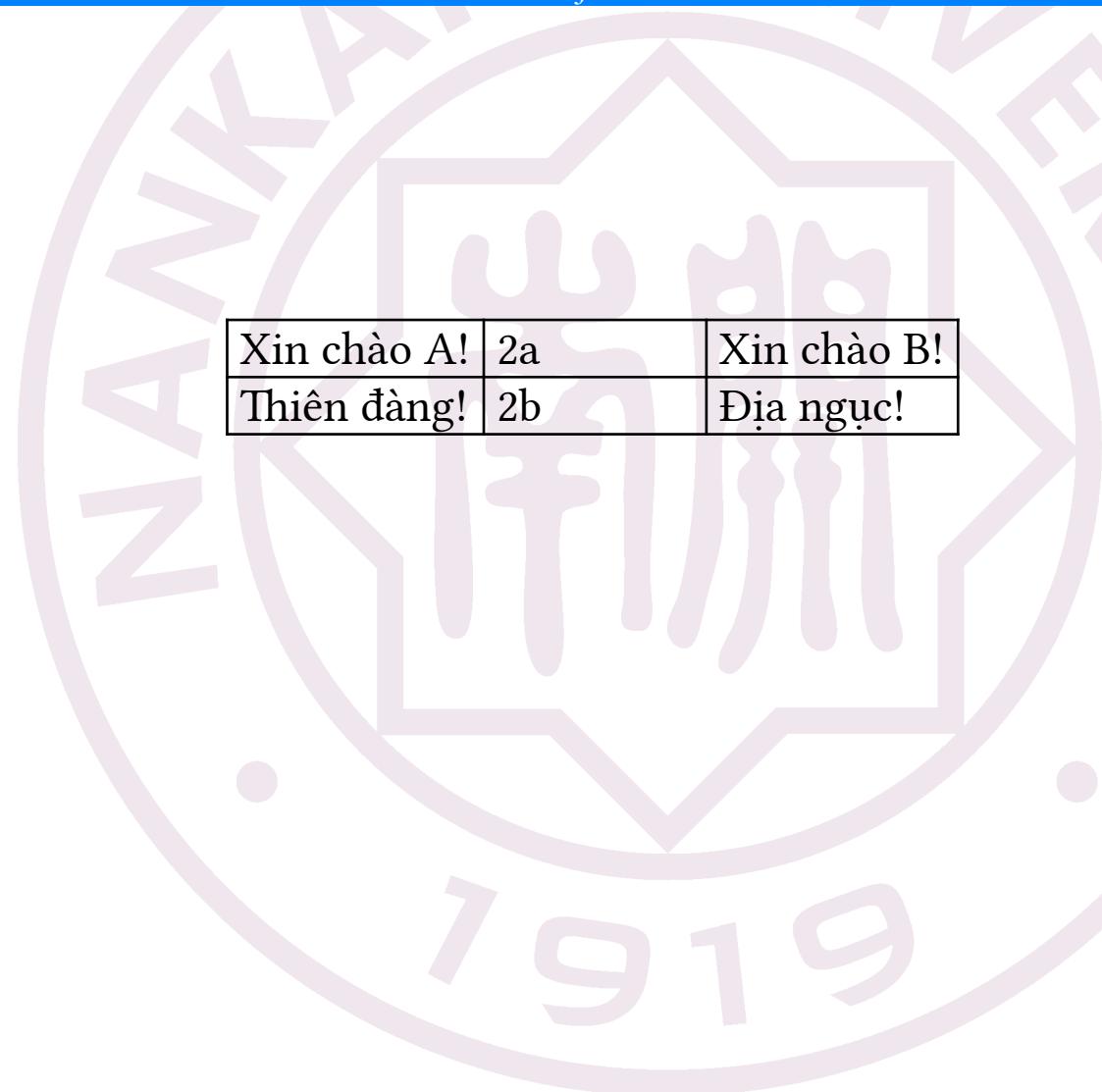




Bảng

```
#table(  
  columns: (auto, 3cm, auto),  
  [Xin chào A!],  
  [2a],  
  [Xin chào B!],  
  [Thiên đàng!],  
  [2b],  
  [Địa ngục!])
```

Xin chào A!	2a	Xin chào B!
Thiên đàng!	2b	Địa ngục!





Công thức toán

```
$ sum_(k=0)^n k = 1 + \dots + n $
```

```
$ A = pi r^2 $
```

```
$ "area" = pi dot.op "radius"^2 $
```

```
$ cal(A) :=  
{ x in RR | x "is natural" } $
```

```
$ frac(a^2, 2) $
```

$$\sum_{k=0}^n k = 1 + \dots + n$$

$$A = \pi r^2$$

$$\text{area} = \pi \cdot \text{radius}^2$$

$$\mathcal{A} := \{x \in \mathbb{R} \mid x \text{ is natural}\}$$

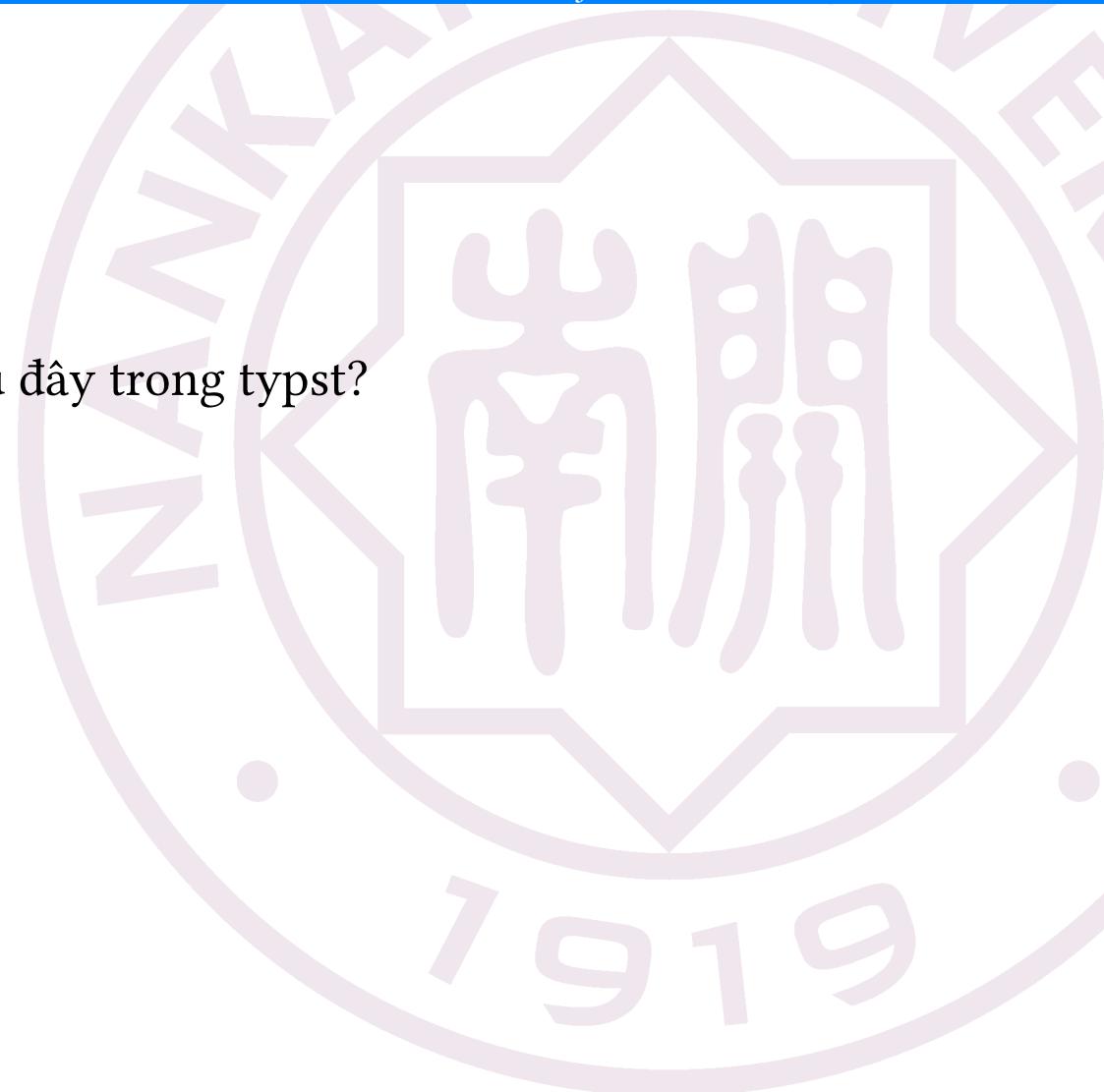
$$\frac{a^2}{2}$$



Câu hỏi 2

Làm thế nào để bạn thực hiện những điều sau đây trong typst?

Công thức	Cho là
$a^2 + b^2 = c^2$	Định lý Pythagoras
$c \leq a + b$	Không xác định





Cách thực hiện 2

```
#table(columns: (auto, auto),  
       strong[Công thức], strong[Cho là],  
       $a^2 + b^2 = c^2$, [Định lý Pythagoras],  
       $c <= a + b$, [Không xác định])
```





Thiết lập quy tắc⁶

Đây là phông chữ mặc định! Q

```
#set text(font: "New Computer Modern",
fill: blue)
```

Q Từ bây giờ mọi thứ hoàn toàn ở phông chữ khác và thậm chí cả màu xanh lam!

```
#set par(first-line-indent: 1.5em,
justify: true)
```

Từ bây giờ, mọi dòng đầu tiên của đoạn văn sẽ được thụt vào và căn đều!

Thực sự, tôi chắc chắn! #lorem(20)

Đây là phông chữ mặc định! Q

Q Từ bây giờ mọi thứ hoàn toàn ở phông chữ khác và thậm chí cả màu xanh lam!

Từ bây giờ, mọi dòng đầu tiên của đoạn văn sẽ được thụt vào và căn đều!

Thực sự, tôi chắc chắn! Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. #lorem(20)





Thiết lập quy tắc⁸

#show heading: set text(red)

==== Xin chào!

===== Thiên đàng!

// Từ trang giới thiệu

// cách sử dụng chính thức của Typst

#show "Project": smallcaps

#show "badly": "great"

*We started Project in 2019
and are still working on it.

Project is progressing badly.*

(_Chúng tôi bắt đầu Dự án vào năm 2019
và vẫn đang làm việc trên nó.

Dự án đang tiến triển không mấy khả quan._)

Xin chào!

Thiên đàng!

We started PROJECT in 2019
and are still working on it.

PROJECT is progressing great.

(Chúng tôi bắt đầu Dự án vào năm
2019 và vẫn đang làm việc trên nó.
Dự án đang tiến triển không mấy
khả quan.)



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

42

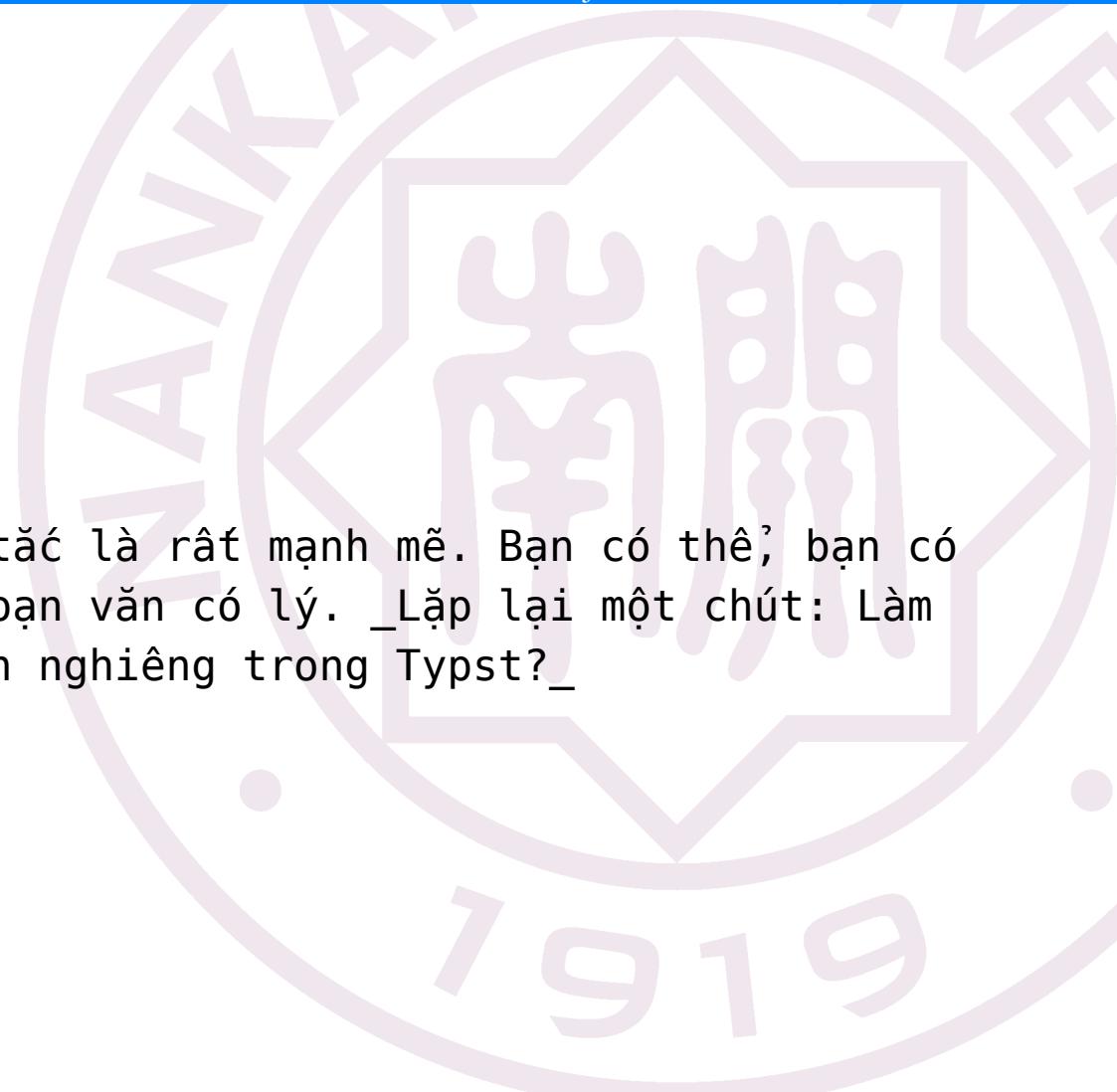


Câu hỏi 3

Làm thế nào để bạn thực hiện những điều sau đây trong Typst? Lưu ý: Quy tắc show không áp dụng cho `#lorem()`.

Trong typst hiển thị và thiết lập quy tắc là rất mạnh mẽ. Bạn có thể, bạn có thể gõ. **Typst** phải luôn được in đậm. Đoạn văn có lý. *Lặp lại một chút: Làm thế nào để bạn tạo lại một cái gì đó in nghiêng trong Typst?*





Cách thực hiện 3

```
#set par(justify: true)
#show "Typst": strong
```

Trong typst hiển thị và thiết lập quy tắc là rất mạnh mẽ. Bạn có thể, bạn có thể'gõ. Typst phải luôn được in đậm. Đoạn văn có lý. _Lặp lại một chút: Làm thế nào để bạn tạo lại cái gì đó in nghiêng trong Typst?_



```

\documentclass[14pt,a4paper]{extarticle}
\usepackage{bold-extra}
\usepackage{amssymb}
\usepackage[T1]{fontenc}
\usepackage[paperwidth=22cm,paperheight=5.5cm,
left=0.5cm,right=0.5cm,top=0.5cm,
bottom=0.5cm]{geometry}

\pagenumbering{gobble}
\setlength{\parskip}{0.65em}
\setlength{\parindent}{0pt}

\begin{document}
\noindent\textbf{\textsc{Definition 1.}} \textit{Sei $D \subset \mathbb{R}$ und sei $f: D \rightarrow \mathbb{R}$ eine Funktion. $f$ ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:}

\textit{Für alle $\epsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \epsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.}

\textit{Oder Alternativ: $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$}

\bigskip
(\LaTeX)
\end{document}

```

```

#set page(margin: 0.5cm, width: 22cm, height: 5.5cm)
#set text(size: 14pt, font: "New Computer Modern")
#set par(justify: true)

*#smallcaps([Definition 1.])* _Sei $D \subset \mathbb{R}$ und sei $f: D \rightarrow \mathbb{R}$ eine Funktion. $f$ ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:_

_Für alle $\epsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \epsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$._

_Oder Alternativ: $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$_

#v(1em)
(Typst)

```



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

45



DEFINITION 1. Sei $D \subseteq \mathbb{R}$ und sei $f : D \rightarrow \mathbb{R}$ eine Funktion. f ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:

Für alle $\epsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \epsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.

Oder Alternativ: $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$

(LATEX)

DEFINITION 1. Sei $D \subseteq \mathbb{R}$ und sei $f : D \rightarrow \mathbb{R}$ eine Funktion. f ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:

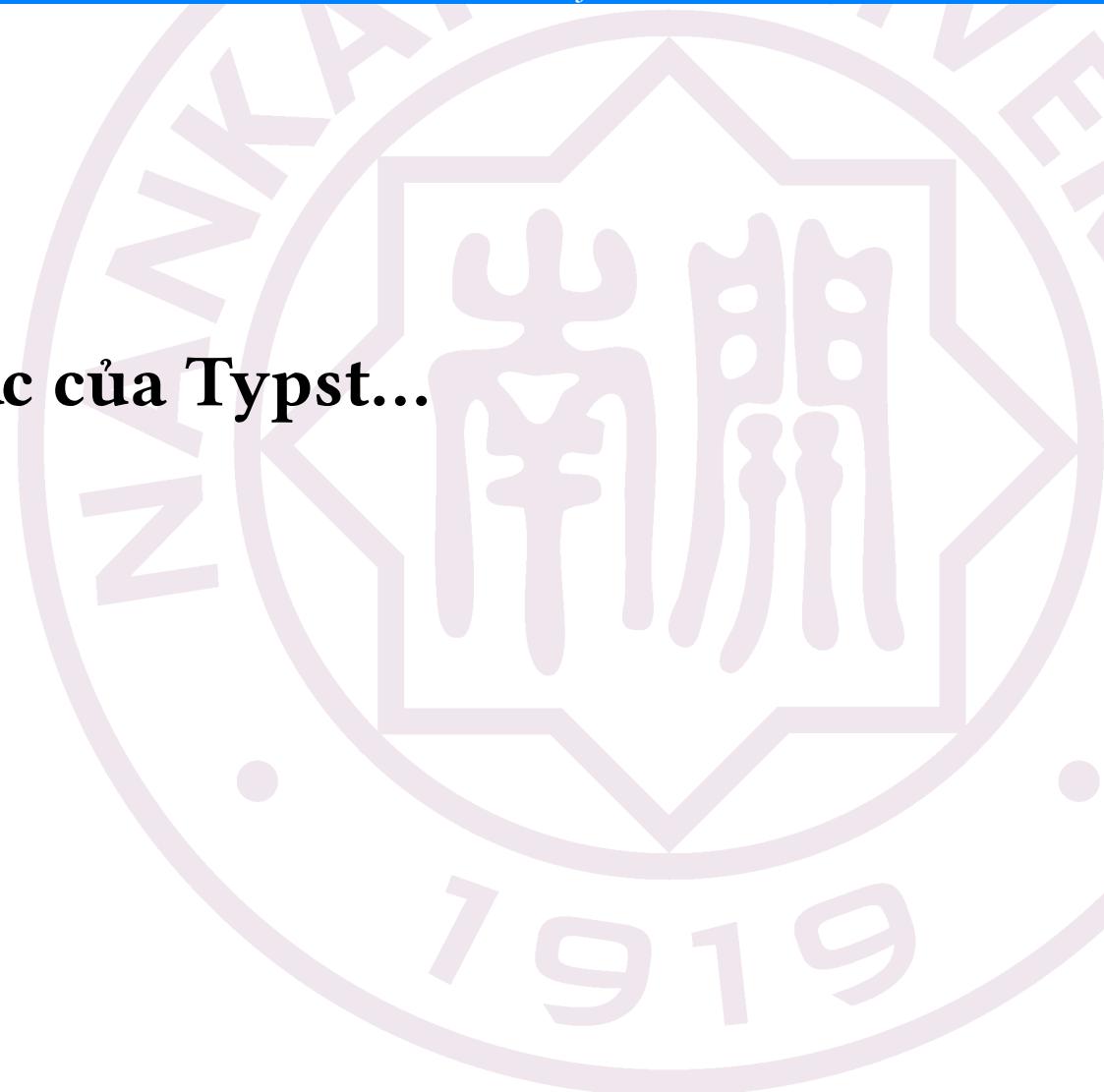
Für alle $\varepsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \varepsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.

Oder Alternativ: $\forall \varepsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \varepsilon$

(Typst)



Tài liệu tham khảo chính thức của Typst...





- <https://typst.app/docs> như một công việc tham khảo
- Tài liệu rất quan trọng!

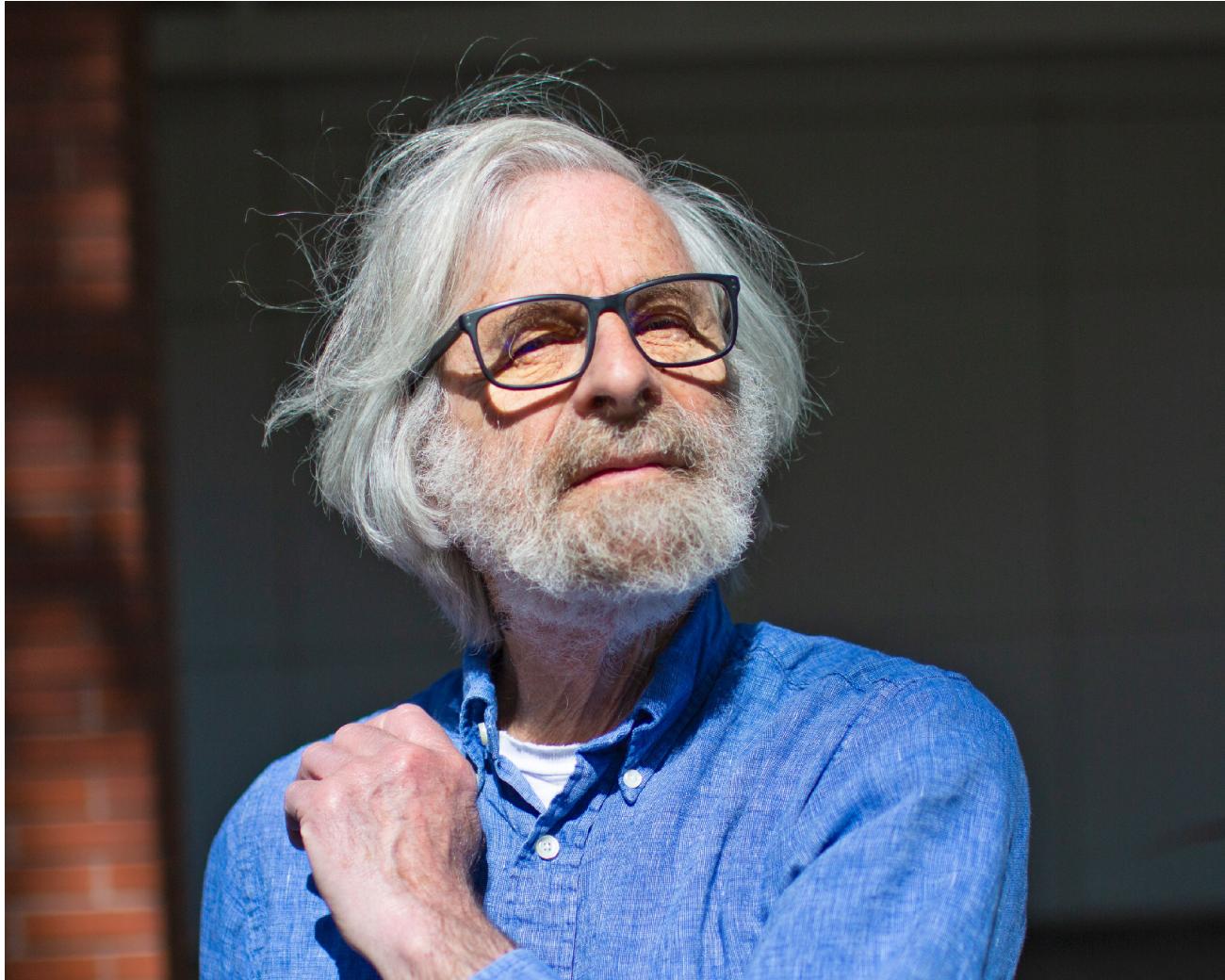


Ví dụ về tài liệu tham khảo: Hàm image



image có thể có nhiều chức năng hơn bạn nghĩ!

```
#image("leslie_lamport.png")
```





Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

49



Ví dụ về tài liệu tham khảo: Hàm `image`

`image` có thể có nhiều chức năng hơn bạn nghĩ!

```
#image(height: 50%, "leslie_lamport.png")
```





Ví dụ về tài liệu tham khảo: Hàm `image`

`image` có thể có nhiều chức năng hơn bạn nghĩ!

```
#image(fit: "stretch", width: 100%, height: 100%, "leslie_lamport.png")
```



 Search[☰](#) > Overview[Overview](#)[Tutorial](#)[Reference](#)[Changelog](#)[Community](#)

Overview

Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.

This documentation is split into two parts: A beginner-friendly tutorial that introduces Typst through a practical use case and a comprehensive reference that explains all of Typst's concepts and features.

We also invite you to join the community we're building around Typst. Typst is still a very young project, so your feedback is more than valuable.

[Tutorial](#)

Step-by-step guide to help you get started.

[Reference](#)

Details about all syntax, concepts, types, and functions.



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

52



image

Results

Image FUNCTION

Formatting CHAPTER

Box FUNCTION

Baseline PARAMETER OF
BOX

Padding FUNCTION

Background PARAMETER
OF PAGE

Table FUNCTION

Path PARAMETER OF IMAGE

Width PARAMETER OF
IMAGEHeight PARAMETER OF
IMAGE

> Overview

Overview

Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.

This documentation is split into two parts: A beginner-friendly tutorial that introduces Typst through a practical use case and a comprehensive reference that explains all of Typst's concepts and features.

We also invite you to join the community we're building around Typst. Typst is still a very young project, so your feedback is more than valuable.



Tutorial

Step-by-step guide to help you get started.



Reference

Details about all syntax, concepts, types, and functions.

Overview

[Tutorial](#)[Reference](#)[Changelog](#)[Community](#)



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

53



Reference > Reference > Visualize > Image

Overview

Tutorial

Reference

LANGUAGE

Syntax

Styling

Scripting

Types

CONTENT

Text

Math

Layout

Visualize

Circle

Ellipse

Image

Line

Path

Polygon

Rectangle

Square

image Element ?

A raster or vector graphic.

Supported formats are PNG, JPEG, GIF and SVG.

Example

```
#figure(  
  image("molecular.jpg", width: 80%),  
  caption: [  
    A step in the molecular testing  
    pipeline of our lab.  
  ],  
)
```



Figure 1: A step in the molecular testing pipeline of our lab.

Parameters ?

```
image(  
  string,  
  width: auto relative length,
```

ON THIS PAGE

Summary

Parameters

path

width

height

alt

fit



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

54



Parameters ⓘ

```
image(  
  string ,  
  width: auto relative length ,  
  height: auto relative length ,  
  alt: none string ,  
  fit: string ,  
) -> content
```

math string Required Positional ⓘ



width `auto` or `relative length` *Settable* ?

The width of the image.

height `auto` or `relative length` *Settable* ?

The height of the image.

alt `none` or `string` *Settable* ?

A text describing the image.

fit `string` *Settable* ?

How the image should adjust itself to a given area.

- `"cover"` The image should completely cover the area. This is the default.
- `"contain"` The image should be fully contained in the area.
- `"stretch"` The image should be stretched so that it exactly fills the area, even if this means that the image will be distorted.



Dương

Contact me via: duongtuanlinh@duytan.edu.vn

DUY TAN UNIVERSITY - ĐÀ NẴNG - VIET NAM

56

Ví dụ `#enum()`:

numbering `string` or `function` *Settable* ?

How to number the enumeration. Accepts a [numbering pattern or function](#).

If the numbering pattern contains multiple counting symbols, they apply to nested enums. If given a function, the function receives one argument if `full` is `false` and multiple arguments if `full` is `true`.

› View example

Ví dụ `#enum()`:

numbering string or function Settable ?

How to number the enumeration. Accepts a [numbering pattern or function](#).

If the numbering pattern contains multiple counting symbols, they apply to nested enums. If given a function, the function receives one argument if `full` is `false` and multiple arguments if `full` is `true`.

▼ View example

```
#set enum(numbering: "1.a")
+ Different
+ Numbering
  + Nested
  + Items
+ Style

#set enum(numbering: n => super[#n])
+ Superscript
+ Numbering!
```

- 1) Different
 - 2) Numbering
 - a) Nested
 - b) Items
 - 3) Style
- ¹ Superscript
² Numbering!





Câu hỏi 4

Tham số nào có thể được sửa đổi theo quy tắc set để định dạng danh sách liệt kê (#enum()) như sau? tài liệu!

- I. Tài liệu tham khảo số 1
- II. Tài liệu tham khảo số 2
- III. Tài liệu tham khảo số 3



Cách thực hiện 4

- ```
#set enum(numbering: "I.")
```
- + Tài liệu tham khảo số'1
  - + Tài liệu tham khảo số'2
  - + Tài liệu tham khảo số'3





## Câu hỏi 5

Tham số nào có thể được sửa đổi theo quy tắc set để định dạng một danh sách không được đánh số như thế này? Hãy đọc tài liệu tham khảo tài liệu!

- > Tài liệu tham khảo số 1
- > Tài liệu tham khảo số 2
- > Tài liệu tham khảo số 3





## Cách thực hiện 5

```
#set list(marker: ">")
- Tài liệu tham khảo số'1
- Tài liệu tham khảo số'2
- Tài liệu tham khảo số'3
```

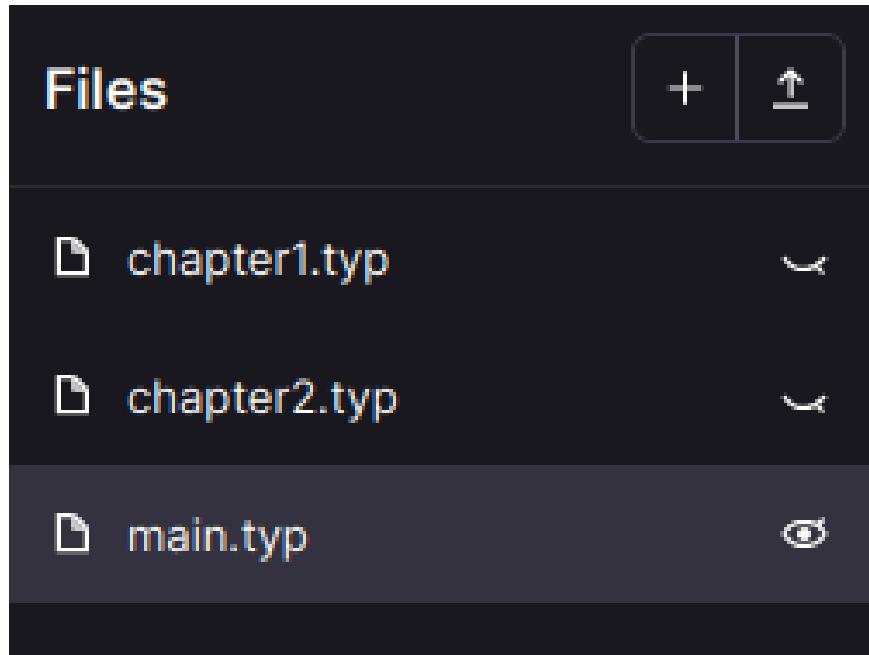
# Cách tạo mẫu và tập lệnh trong Typst





# Sử dụng nhiều tập lệnh





chapter1.typ:

**== Caption 1**  
`#lorem(10)`

chapter2.typ:

**== Caption 2**  
`#lorem(10)`

main.typ:

**= Cuôn sách tuyệt vời của tôi**

Đây, chapter!

```
#include "chapter1.typ"
#include "chapter2.typ"
```



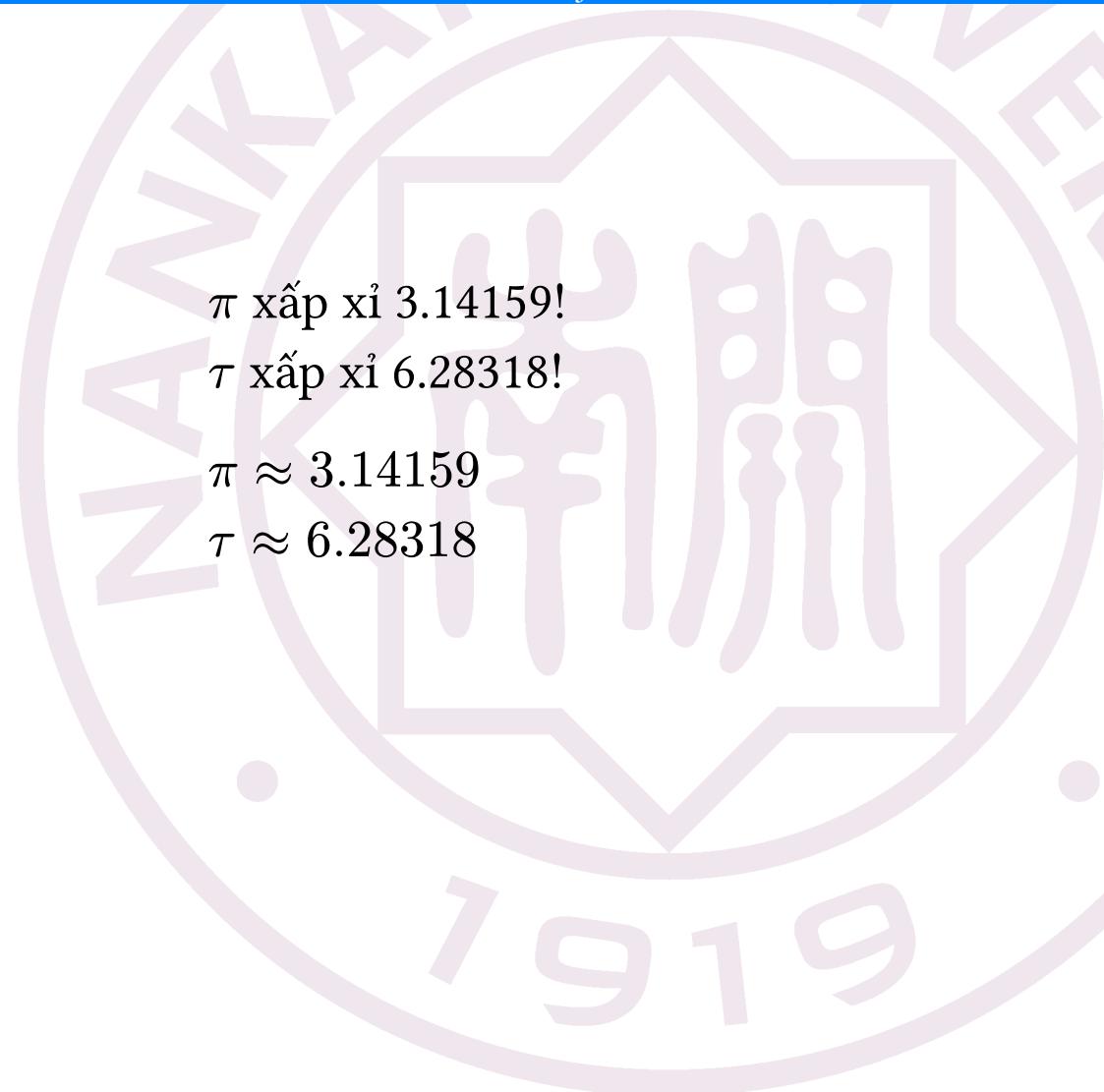
# Các hàm và biến #1

```
#let var = 3.14159
#let double(e) = {
 return 2*e
}

pi xấp xỉ #var!
tau xấp xỉ #double(var)

$pi approx var$ \
$tau approx double(var)$
```

$\pi$  xấp xỉ 3.14159!  
 $\tau$  xấp xỉ 6.28318!  
 $\pi \approx 3.14159$   
 $\tau \approx 6.28318$





# Các hàm và biến #2

Một lưu ý: chúng tôi đang làm việc với *tên hàm*.

Không hoạt động:

```
#let var = 2
#let change_var(new_v) = {
 var = new_v
 return var
}

change_var(100)
```

SAI:

```
#let change_var(new_v) = {
 var = new_v
 ^^^
```

variables from outside the function are  
read-only and cannot be modified



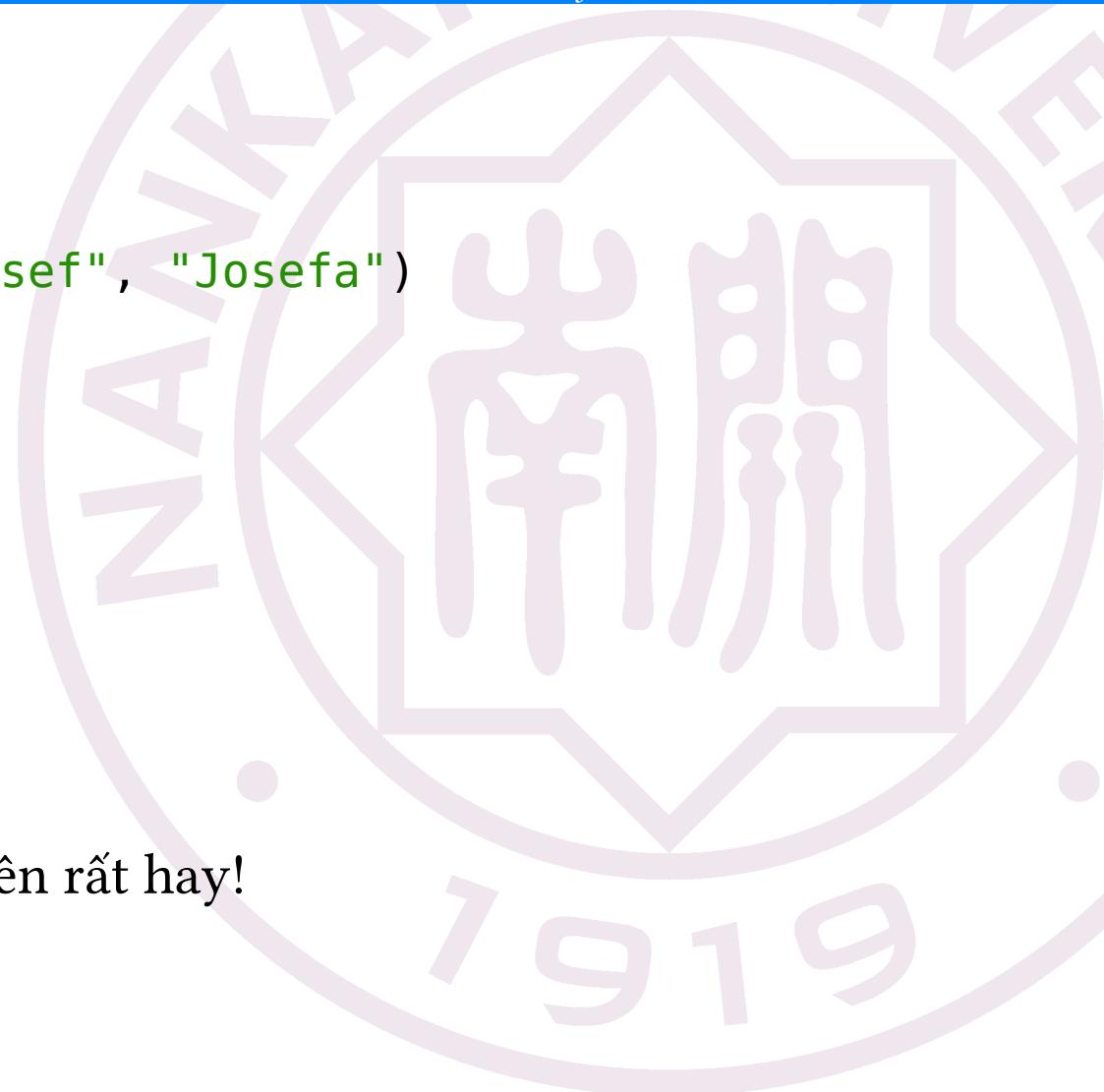


# Các hàm và biến #3

```
#let names = ("Peter", "Petra", "Josef", "Josefa")
#let greet(names) = {
 [Xin chào]
 names.join(last: " và ", ", ")
 [! #names.len() Tên rất hay!]
}

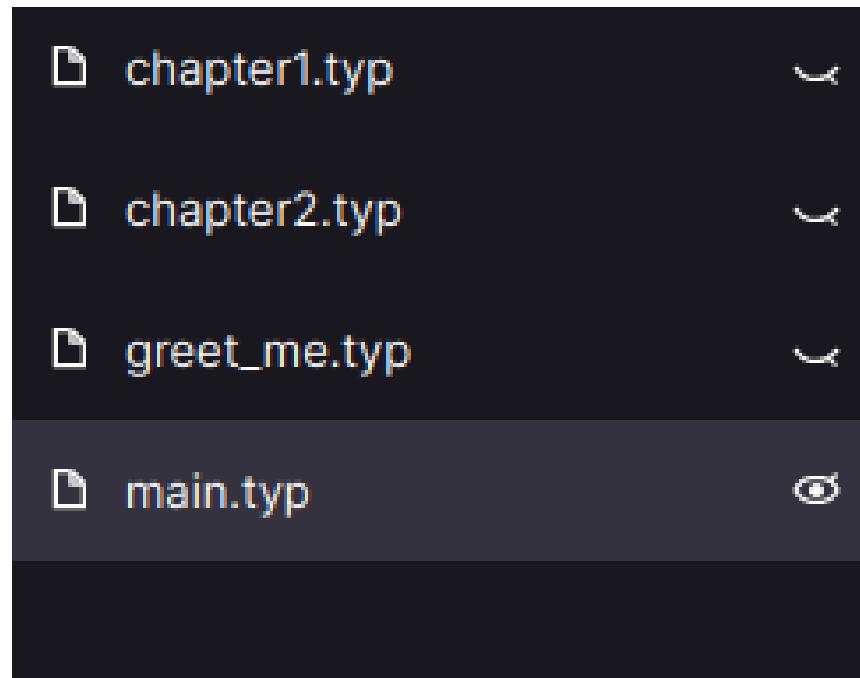
#greet(names)
```

Xin chào Peter, Petra, Josef và Josefa! 4 Tên rất hay!





# Hàm chức năng khác từ tệp bên ngoài

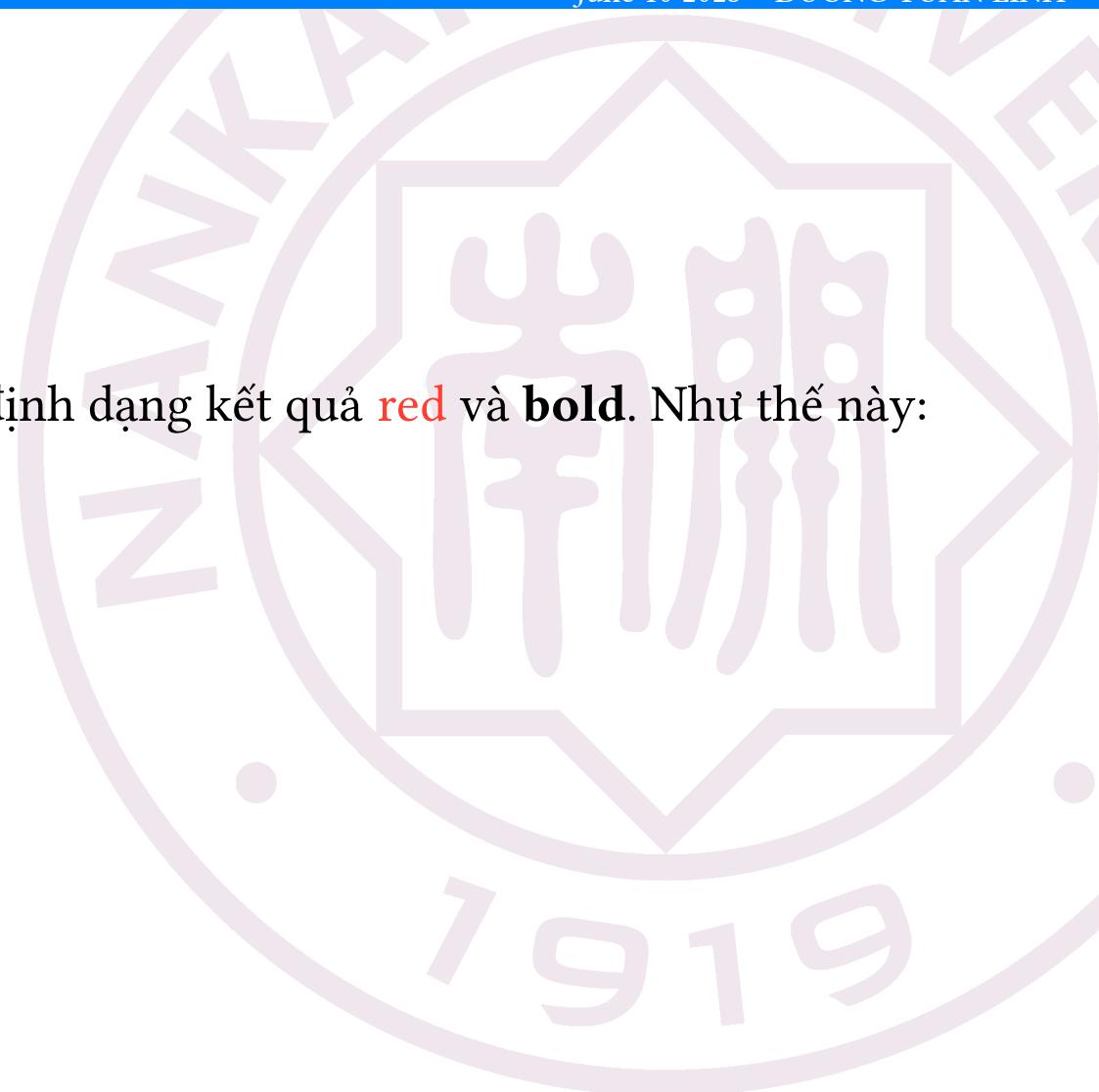


greet\_me.typ definiert:

- greet(names)
- double(n)

```
#import "greet_me.typ": *
#import "greet_me.typ": greet
```





## Câu hỏi 6

Chúng ta muốn tạo một hàm cộng hai số và định dạng kết quả **red** và **bold**. Như thế này:

```
#add(20, 40)
```

```
60
```



## Cách thực hiện 6

Cách phổ biến nhất:

```
#let add(a, b) = strong(text(red, #[(a+b)]))
#add(20, 40)
```

hoặc biến số thành chuỗi bằng #str():

```
#let add(a, b) = strong(text(red, str(a+b)))
#add(20, 40)
```

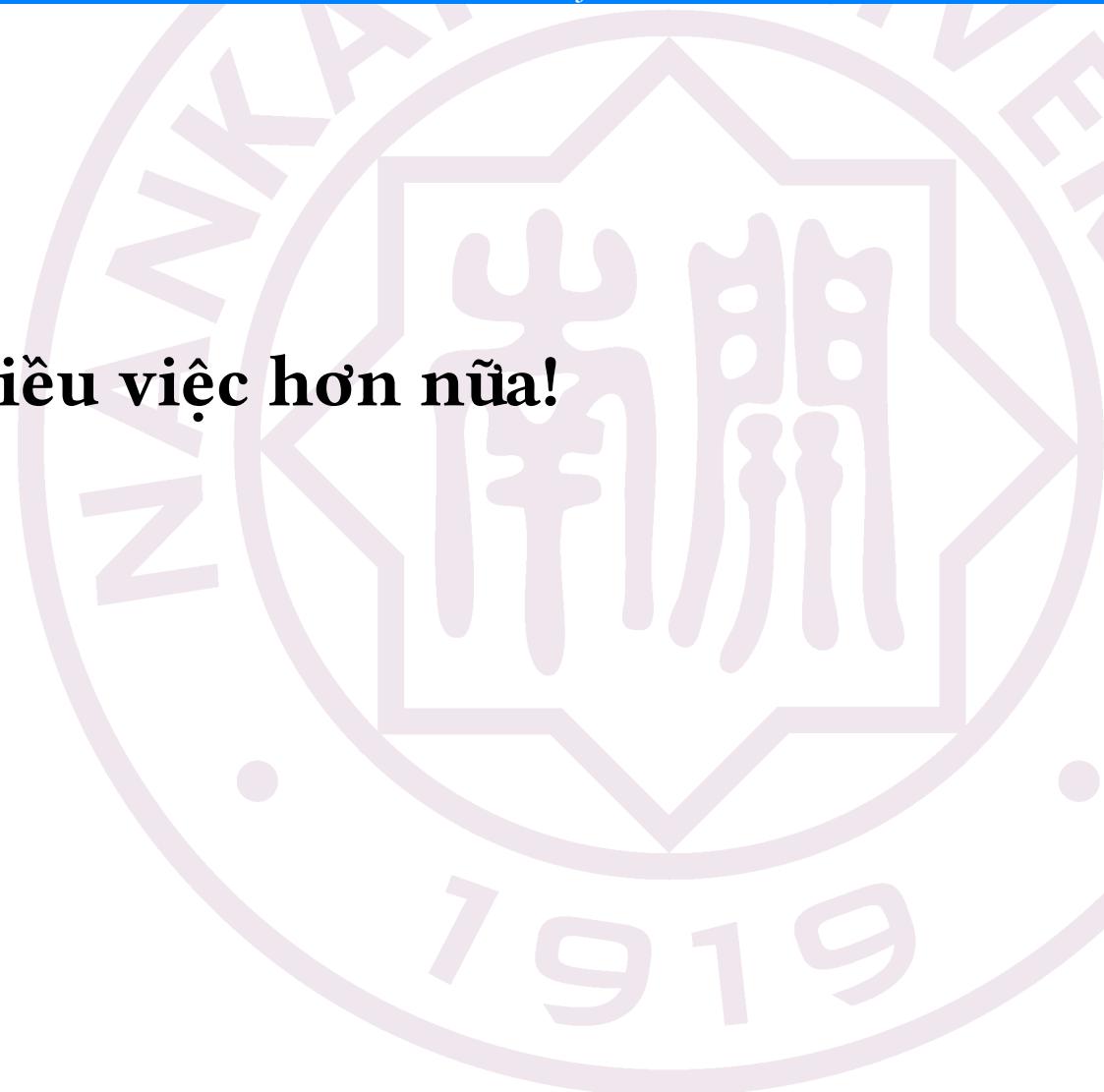
hoặc với dấu ngoặc:

```
#let add(a, b) = {
 let result = a+b
 strong(text(red, str(result)))
}

#add(20, 40)
```



Typst có thể còn làm được nhiều việc hơn nữa!





# Typst có thể còn làm được nhiều việc hơn nữa!

- Thư mục
- Số liệu và tài liệu tham khảo
- hiển thị tài liệu!!!
- Raytracing <https://github.com/elteammate>





# Hình và Tài liệu tham khảo <sup>10</sup>



@glacier shows a glacier. Glaciers are complex systems.

```
#figure(
 image("glacier.jpg", height: 80%),
 caption: [A curious figure.],
) <glacier>
```

Figure 1 shows a glacier. Glaciers are complex systems.



Figure 1: A curious figure.



# TÀI LIỆU THAM KHẢO<sup>12</sup>



works.bib:

```
@article{netwok,
 title={At-scale impact of the {Net Wok}: A culinarily
holistic investigation of distributed dumplings},
 author={Astley, Rick and Morris, Linda},
 journal={Armenian Journal of Proceedings},
 volume={61},
 pages={192--219},
 year={2020},
 publisher={Automattic Inc.}
}

@article{arrgh,
 title={The Pirate Organization},
 author={Leeson, Peter T.},
}
```

This was already noted by pirates long ago. @arrgh

Multiple sources say ...  
#cite("arrgh", "netwok").

```
#bibliography("works.bib")
```

This was already noted by pirates long ago. [1]

Multiple sources say ... [1, 2].

## Bibliography

- [1] P. T. Leeson, “The pirate organization.”
- [2] R. Astley, and L. Morris, “At-scale impact of the Net Wok: a culinarily holistic investigation of distributed dumplings,” *Armenian J. Proc.*, vol. 61, pp. 192–219, 2020.



# Raytracing

The image shows a code editor window with a dark theme. On the left is the code for a raytracer, and on the right is a rendered 3D scene. The code editor has tabs for Typst, File, Edit, View, Help, and a toolbar with bold, italic, underline, and alignment icons. The rendered scene shows several spheres of different sizes and colors (purple, green, red) on a reflective surface against a blue background.

```
1 let EPS = 1e-6;
2 let CLIPPING = 1e-2;
3 let INF = 1e6;
4
5 let sqrt(x) = {
6 if x < EPS { return x; }
7 let guess = x;
8 // newton's method, a lot of digits of accuracy
9 for i in range(10) {
10 guess = (guess + x / guess) / 2;
11 }
12 guess
13 }
14
15 let about_eq(a, b, eps) = {
16 return abs(a - b) < eps;
17 }
18
19 let dot(a, b) = { a(0) * b(0) + a(1) * b(1) + a(2) * b(2) }
20 let plus(a, b) = { (a(0) + b(0), a(1) + b(1), a(2) + b(2)) }
21 let minus(a, b) = { (a(0) - b(0), a(1) - b(1), a(2) - b(2)) }
22 let neg(a) = { (-a(0), -a(1), -a(2)) }
23 let div(a, x) = { (a(0) / x, a(1) / x, a(2) / x) }
24 let mul(a, x) = { (a(0) * x, a(1) * x, a(2) * x) }
25 let len(a) = { sqrt(dot(a, a)) }
26 let normalize(a) = { div(a, len(a)) }
27 let pointwise_mul(a, b) = { (a(0) * b(0), a(1) * b(1), a(2) * b(2)) }
28
29 let material(color, smoothness) = {
30 (color: color, smoothness: smoothness)
31 }
32
33 let sphere(center, radius, material) = {
34 (center: center, radius: radius, material: material)
35 }
36
37 let ray(start, direction) = {
38 (start: start, direction: normalize(direction))
39 }
```

Công cụ dò tia đầy đủ chức năng để kết xuất 3D.<sup>13</sup>

# TẠM KẾT





# Typst còn thiếu những gì?<sup>15</sup>

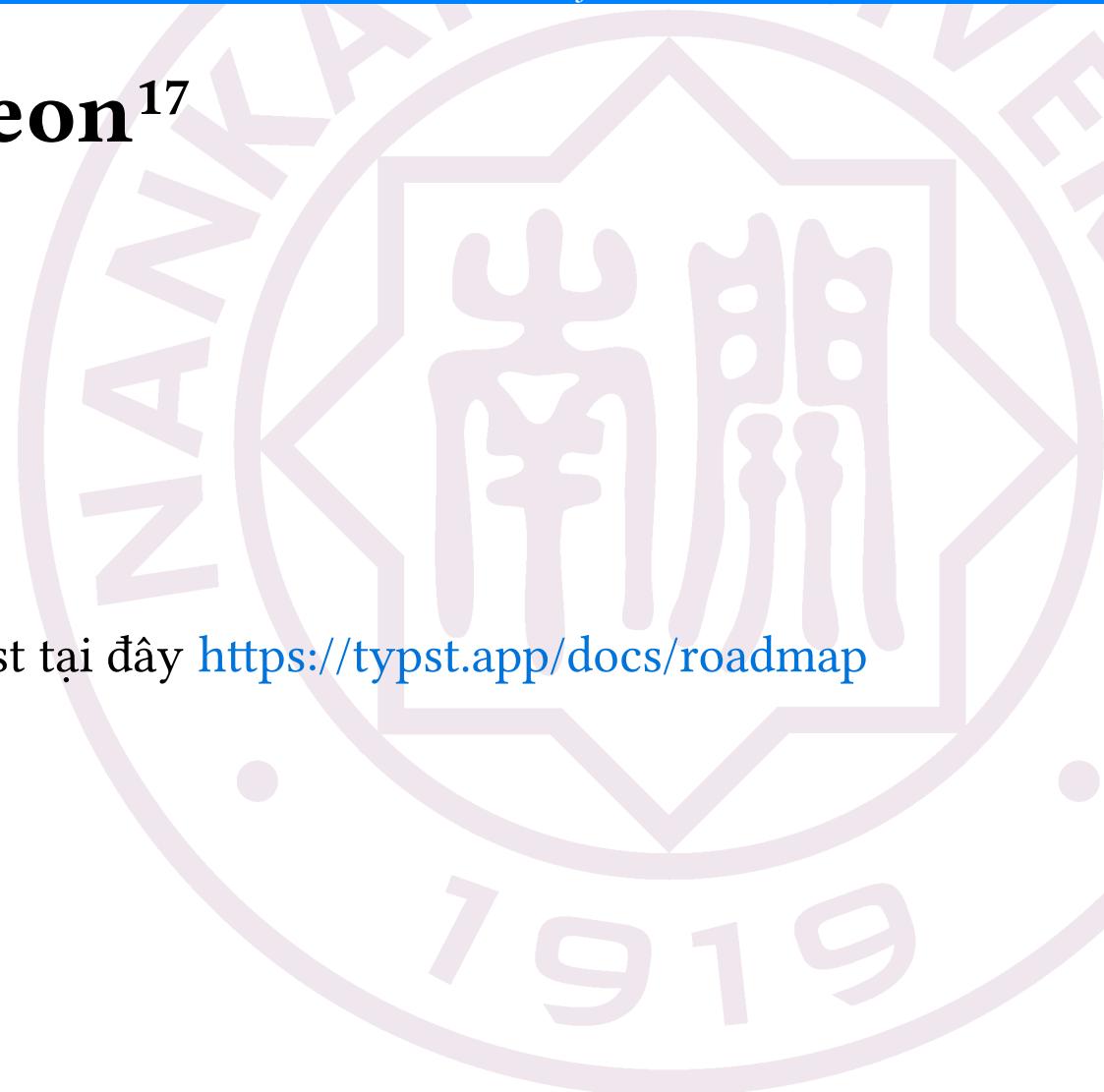
- Bản cập nhật version 0.4.0 ngày 10.06.2023
- Quản lý thư viện (Xem nhanh tại đây: GitHub)
- StackOverflow (Thảo luận tại đây: Discord)





# Một số cập nhật tiếp theon<sup>17</sup>

- Cải thiện công cụ bố trí
- Quản lý gói
- Cải thiện bố cục toán học
- Đầu ra HTML
- Tham khảo thêm về lộ trình phát triển Typst tại đây <https://typst.app/docs/roadmap>





# Ai (không) nên dùng Typst?



Việc cần làm: mở rộng và đánh giá nghiêm túc Typst để giải thích các câu hỏi:

- Tại sao tôi nên sử dụng Typst khi LaTeX đang chạy khắp mọi nơi?
- Vấn đề con gà và quả trứng
- Chúng ta không thể hào hứng với những công nghệ mới chỉ vì chúng ta nghĩ rằng chúng hay sao?
- Làm sao mọi thứ có thể bắt kịp nếu chúng ta cứ nghĩ, “Ừ, nhưng chúng ta đã từng làm khác đi.”?

LaTeX có nhiều gói hỗ trợ hơn. Typst thậm chí chưa hỗ trợ công cụ vẽ hình như TikZ (tôi biết có người đang phát triển công cụ tương đương).

- Các lựa chọn thay thế: Tạo SVG bằng TikZ và nhúng nó vào Typst hoặc sử dụng GraphViz
- <https://github.com/johannes-wolf/typst-canvas> làm thư viện cơ bản để vẽ các hình

Tổng quát: vấn đề con gà và quả trứng. Đôi khi bạn không thể làm những việc ngoài sở thích? Chúng ta không thể chỉ quan tâm đến các công nghệ mới sao? Tôi có thực sự cần tất cả “khả năng” không, con quái vật LaTeX?





# Ai (không) nên dùng Typst?

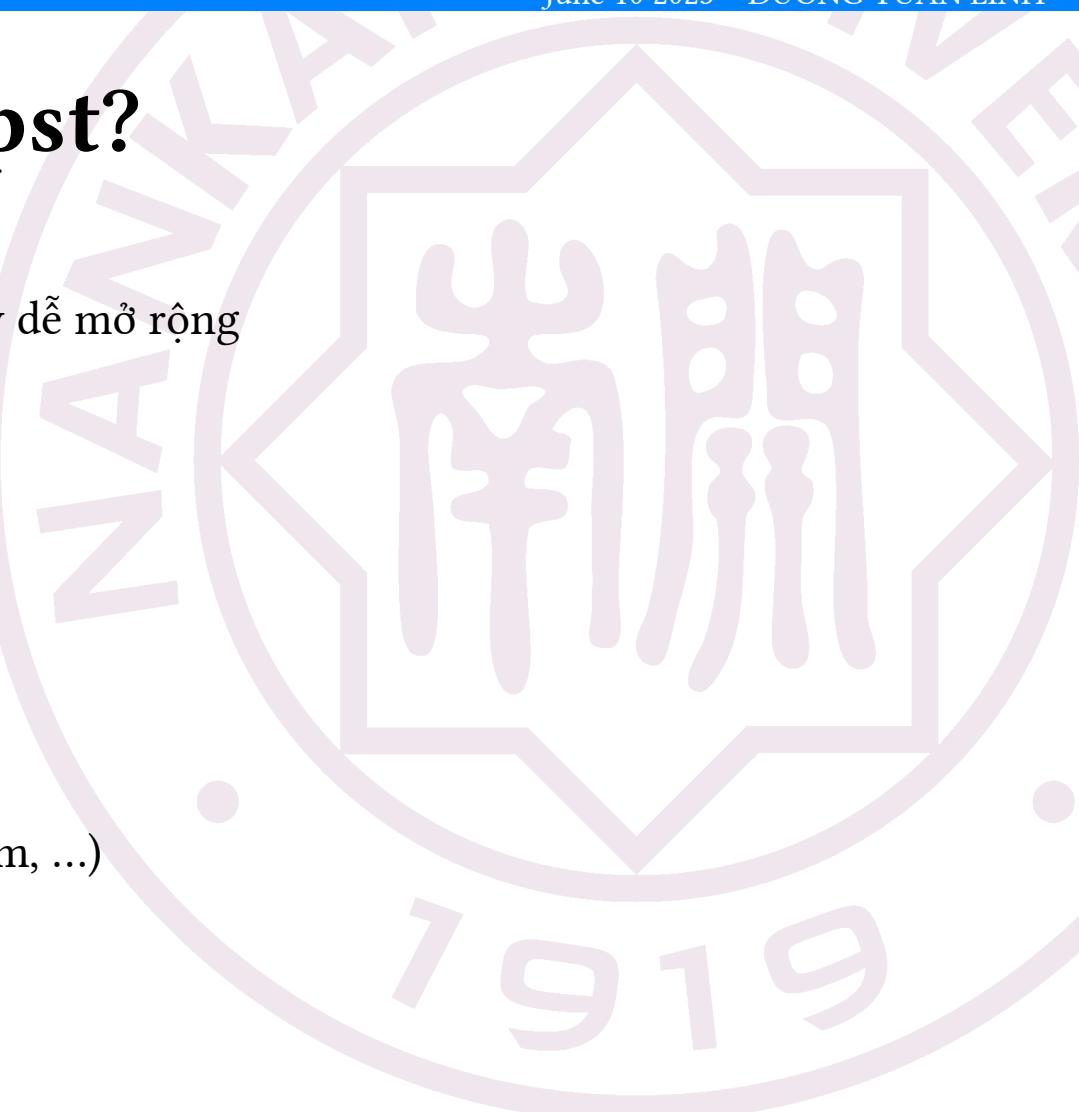
## Thuận lợi:

- ✓ **steile** Đường cong học tập
  - ✓ Lập trình viên có nhiều cách tiếp cận, và cực kỳ dễ mở rộng
  - ✓ Cộng đồng tích cực
  - ✓ Thời gian biên dịch nhanh
  - ✓ IDE trực tuyến
  - ✓ Thông báo lỗi dễ hiểu

## Khó khăn:

- ✗ Lập trình viên có nhiều cách tiếp cận khác nhau
- ✗ Bối cảnh phức tạp (chưa có hình nối)
- ✗ Các hàm thuận túy có thể nặng (trạng thái, bộ đếm, ...)
- ✗ Cho đến nay chưa có sự hỗ trợ từ các tạp chí lớn
- ✗ Chưa quản lý gói trung tâm

(Stand: 10.06.2023)





# Tiếp theo

Lưu ý: Toàn bộ bản trình bày này chỉ được tạo bằng Typst và Typst-LPS trên VSC.

- Tài liệu Typst: <https://typst.app/docs/>
- Hướng dẫn Typst chính thức: <https://typst.app/docs/tutorial>
- Typst Discord chính thức: <https://discord.gg/2uDybryKPe>
- Mã cho bài thuyết trình này và các ví dụ khác: <https://github.com/linhduongtuan/DTU-typst-presentation>
- Luận văn thạc sĩ về Typst của Laurenz Mädje: <https://www.user.tu-berlin.de/laurmaedje-programmable-markup-lingu-for-typesetting.pdf>
- Luận văn thạc sĩ về Typst của Martin Haug: <https://www.user.tu-berlin.de/mhaug/fast-typesetting-incremental-compilation.pdf>
- Lambdas, Bang và Bộ đếm: <https://typst.app/project/rpnqiqoQNfxXjHQmpJ81nF>
- Danh sách dự án về Typst: <https://github.com/qjcg/awesome-typst>



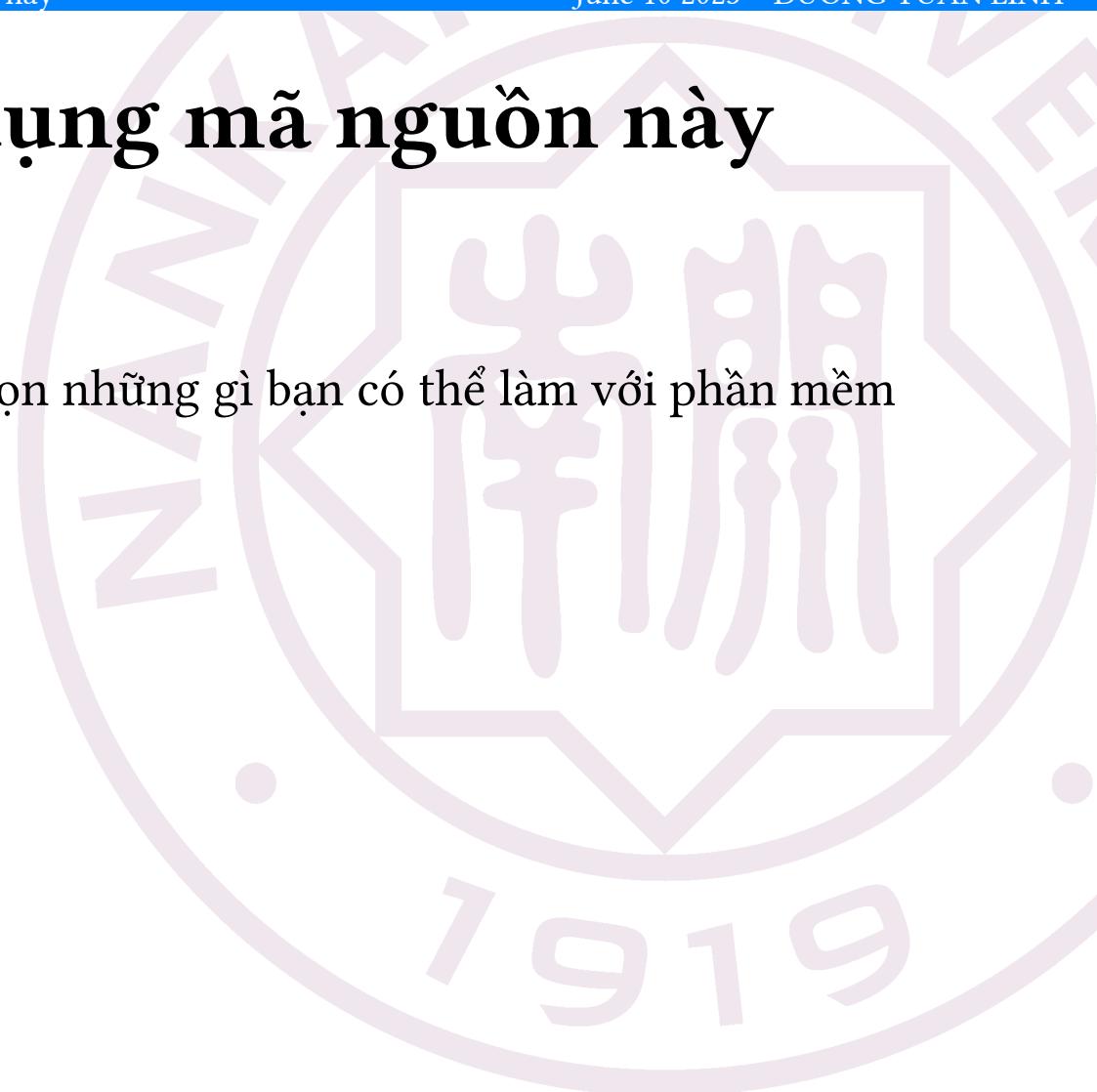
# PHẦN 2. MỘT SỐ HƯỚNG DẪN CƠ BẢN CHO THỦ VIỆN SOẠN THẢO SLIDE





# Về soạn thảo slides sử dụng mã nguồn này

- Phần trình bày này nhằm giới thiệu ngắn gọn những gì bạn có thể làm với phần mềm này bản mẫu.
  - Để có tài liệu đầy đủ, hãy đọc [online book](#).

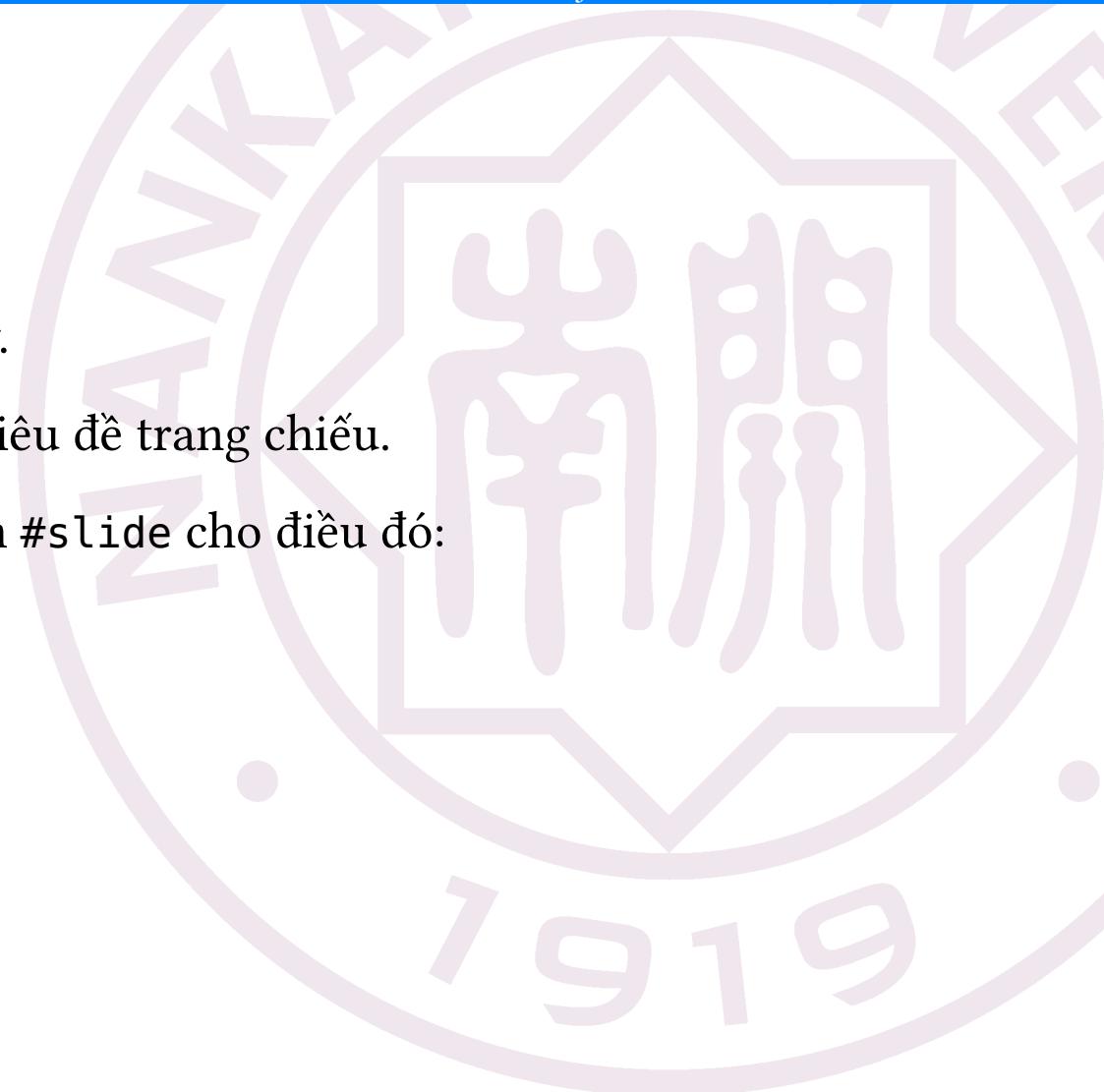




# Who are you

1. Hãy khám phá những gì chúng ta có ở đây.
  - Trên đầu trang chiếu này, bạn có thể thấy tiêu đề trang chiếu.
  - Chúng tôi đã sử dụng đối số title của hàm #slide cho điều đó:

```
#slide(title: "First slide")[\n ...
]
```



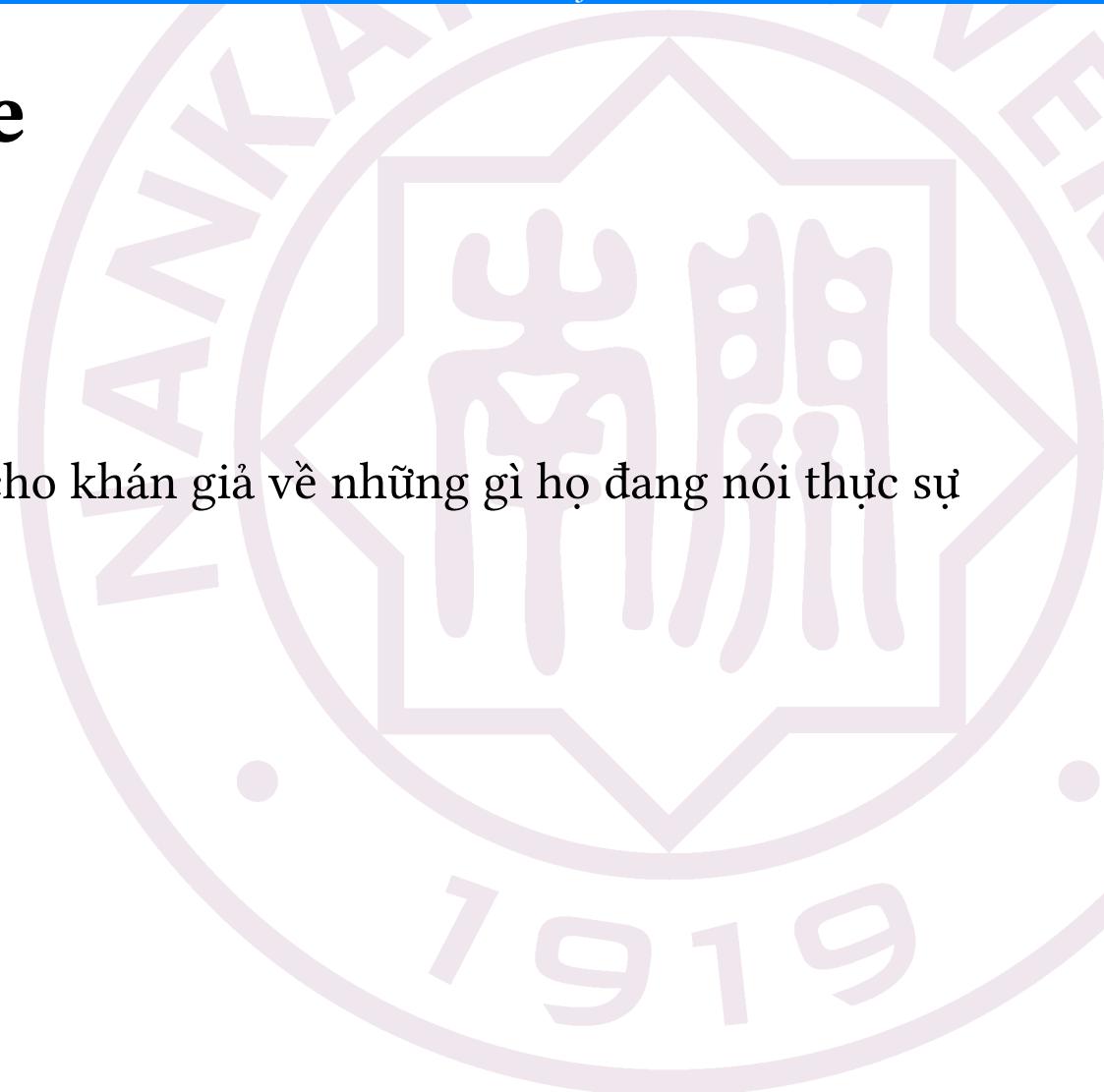
- Tiêu đề không bắt buộc, slide này không có tiêu đề.
- Nhưng bạn có để ý rằng tên phần hiện tại được hiển thị phía trên dòng trên cùng?
- Chúng tôi đã xác định nó bằng cách sử dụng  
`#new-section("Introduction").`
- Điều này giúp khán giả của chúng tôi không bị lạc sau một giấc ngủ ngắn.





# Phần bên dưới mỗi slide

- Bây giờ, hãy nhìn xuống dưới mỗi slide!
- Ở đó chúng tôi có một số thông tin chung cho khán giả về những gì họ đang nói thực sự tham dự ngay bây giờ.
- Bạn cũng có thể thấy số slide ở đó.



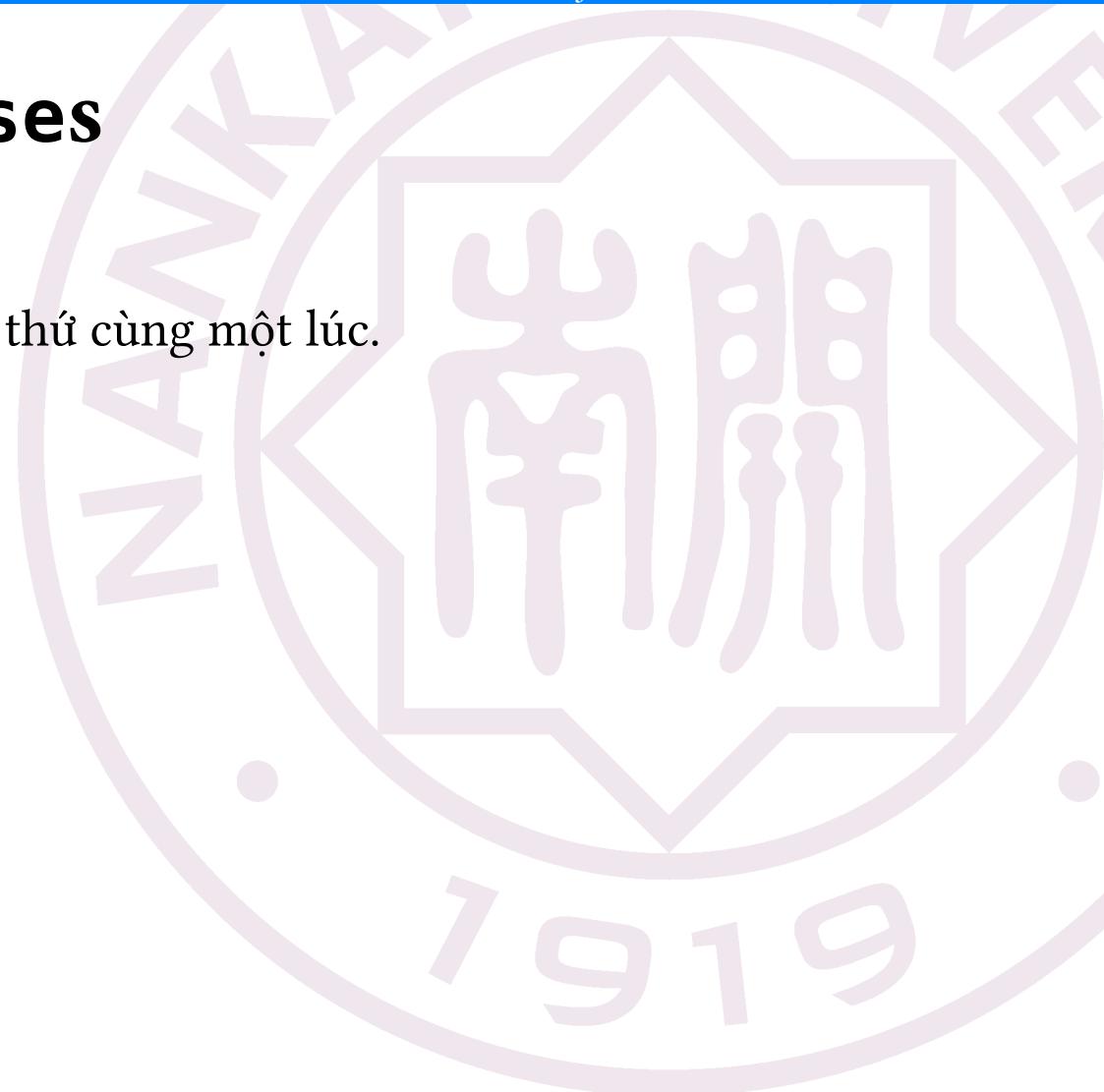
# Nội dung cho mỗi slide động





# Slide động với hàm pauses

- Đôi khi chúng ta không muốn hiển thị mọi thứ cùng một lúc.

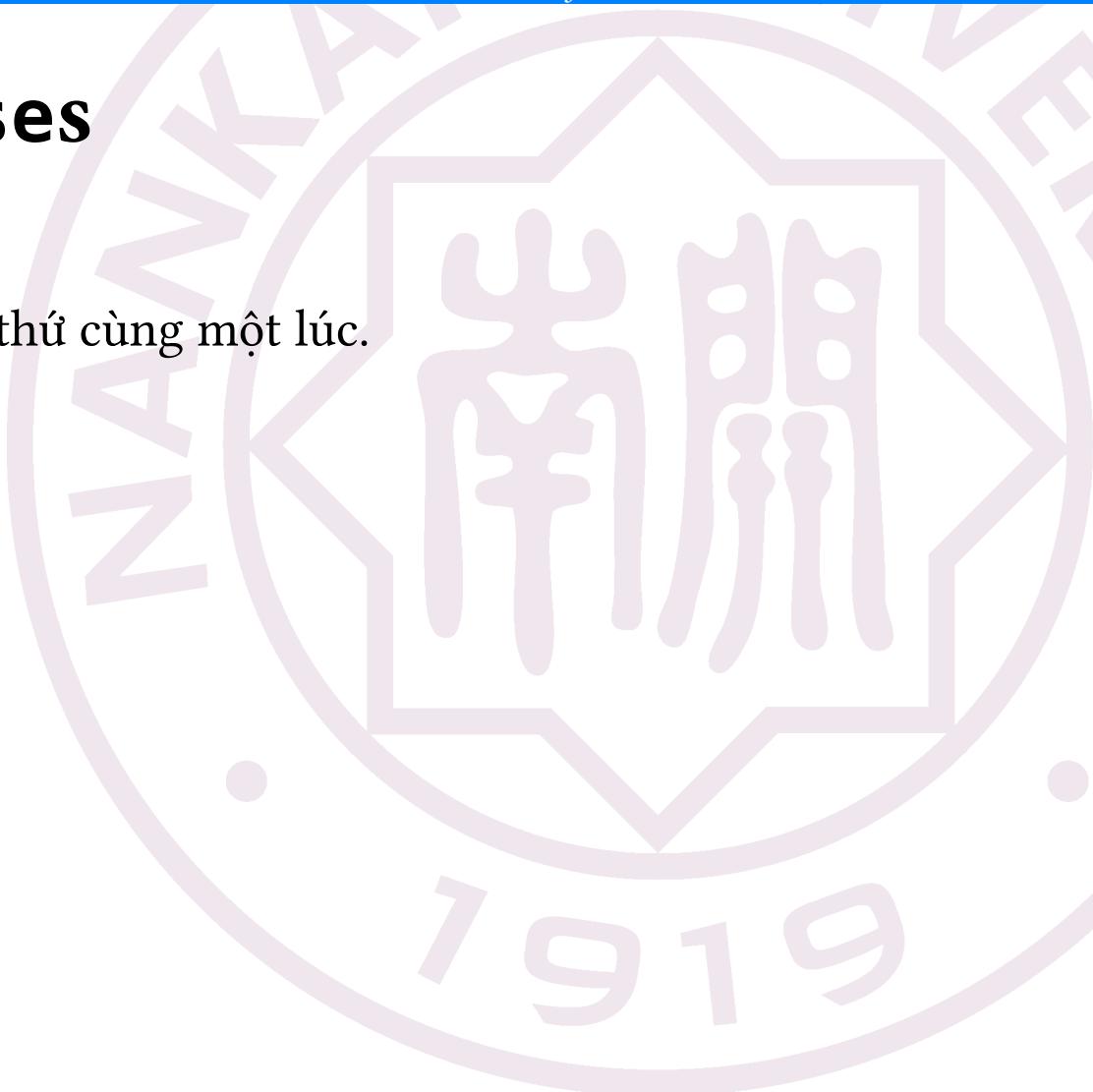




# Slide động với hàm pauses

- Đôi khi chúng ta không muốn hiển thị mọi thứ cùng một lúc.
- Đó là mục đích của chức năng pause!
- Sử dụng nó như sau:

#show: pause(n)





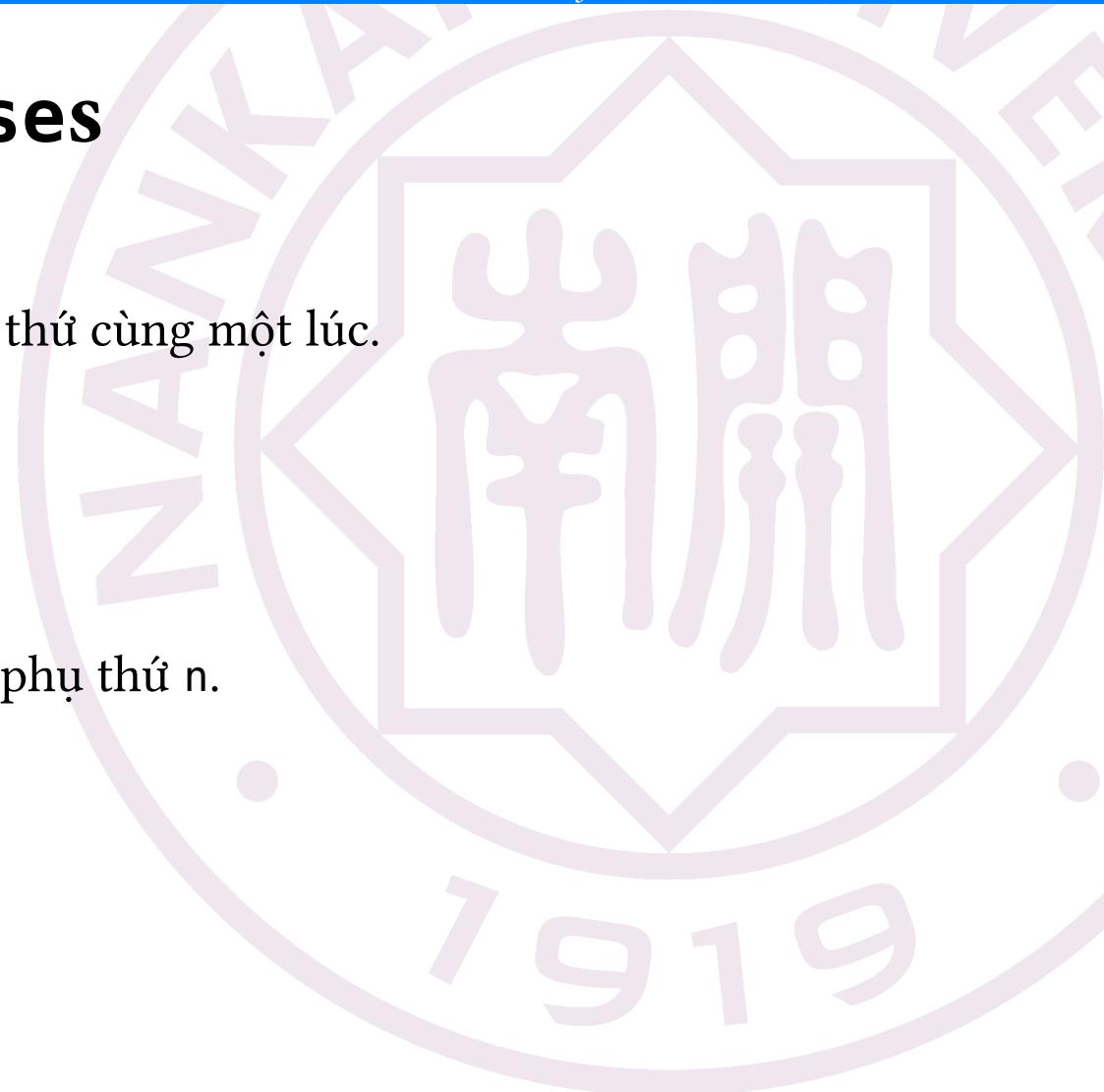
# Slide động với hàm pauses

- Đôi khi chúng ta không muốn hiển thị mọi thứ cùng một lúc.
- Đó là mục đích của chức năng pause!
- Sử dụng nó như sau:

#show: pause(n)

Nó làm cho mọi thứ sau nó xuất hiện ở trang phụ thứ n.

(Cũng lưu ý rằng số trang chiếu không thay đổi khi chúng tôi ở đây.)

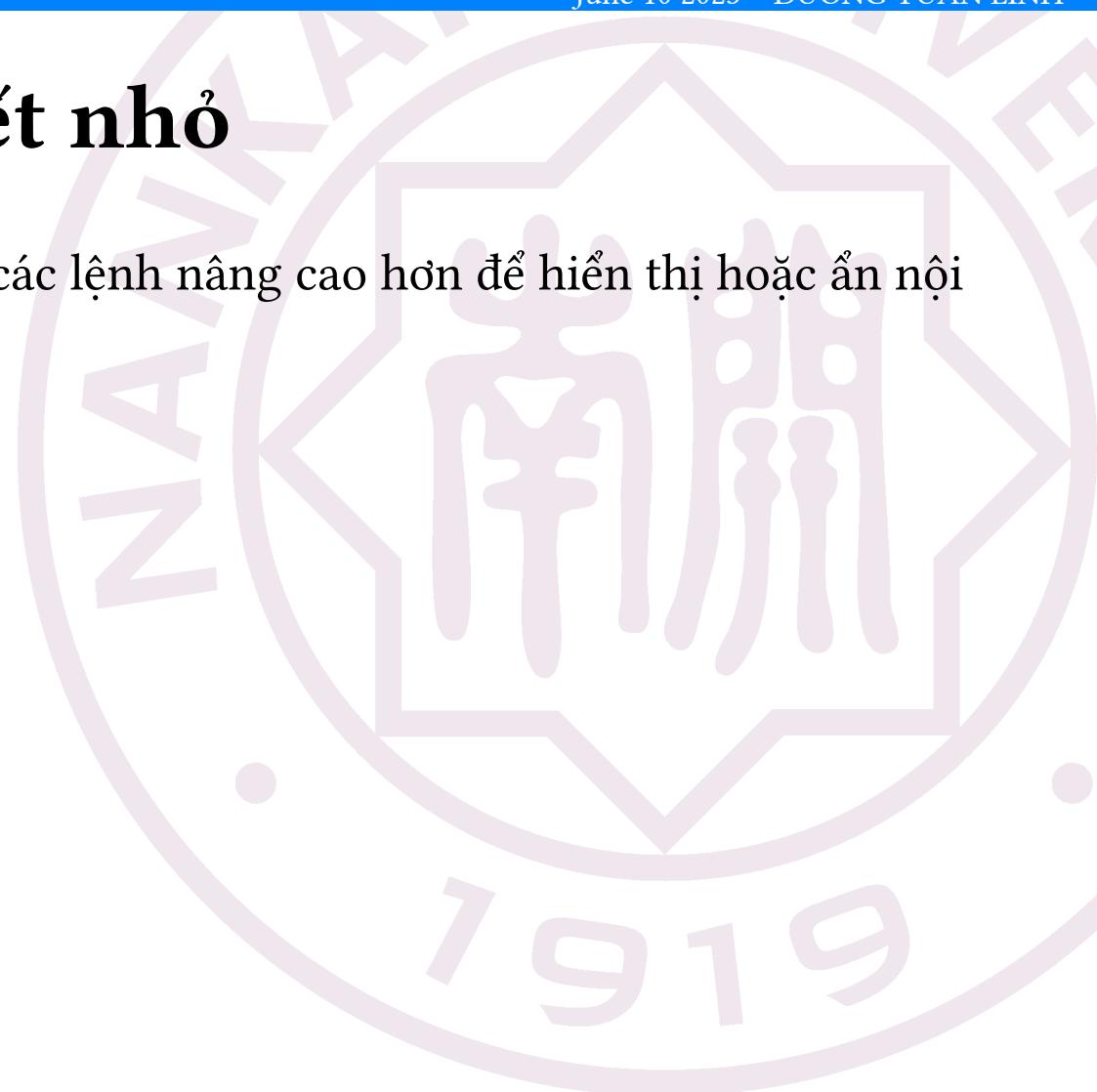




# Kiểm soát những chi tiết nhỏ

1. Khi #pause không đủ, bạn có thể sử dụng các lệnh nâng cao hơn để hiển thị hoặc ẩn nội dung.
2. Đây là những lựa chọn của bạn:
  - #uncover
  - #only
  - #alternatives
  - #one-by-one
  - #line-by-line

*Hãy khám phá chúng chi tiết hơn!*





# #uncover: Bảo tồn Khoảng cách

Sử dụng hàm `#uncover`, nội dung vẫn chiếm khoảng không gian, thậm chí nó sẽ không hiện nội dung.

Ví dụ , chỉ nhìn thấy ở slide phụ ("subslide") thứ hai.

Trong () đằng sau `#uncover`, bạn chỉ định *when* để hiển thị nội dung và trong [ ] sau đó bạn nói *what* để hiển thị:

```
#uncover(3)[chỉ nhìn thấy ở slide phụ ("subslide") thứ ba]
```





# #uncover: Bảo tồn Khoảng cách

Sử dụng hàm `#uncover`, nội dung vẫn chiếm khoảng không gian, thậm chí nó sẽ không hiện nội dung.

Ví dụ , Từ này chỉ nhìn thấy ở slide phụ ("subslide") thứ hai.

Trong () đằng sau `#uncover`, bạn chỉ định *when* để hiển thị nội dung và trong [ ] sau đó bạn nói *what* để hiển thị:

```
#uncover(3)[chỉ nhìn thấy ở slide phụ ("subslide") thứ ba]
```





# #uncover: Bảo tồn Khoảng cách

Sử dụng hàm `#uncover`, nội dung vẫn chiếm khoảng không gian, thậm chí nó sẽ không hiện nội dung.

Ví dụ , chỉ nhìn thấy ở slide phụ (“subslide”) thứ hai.

Trong () đằng sau `#uncover`, bạn chỉ định *when* để hiển thị nội dung và trong [ ] sau đó bạn nói *what* để hiển thị:

```
#uncover(3)[chỉ nhìn thấy ở slide phụ ("subslide") thứ ba]
```

```
chỉ nhìn thấy ở slide phụ ("subslide") thứ ba
```





# Các nguyên tắc hiển thị phức tạp

- Cho đến nay, chúng tôi chỉ sử dụng các chỉ mục trang chiếu con duy nhất để xác định thời điểm hiển thị nội dung nào đó.
- Chúng ta cũng có thể sử dụng các mảng số...

```
#uncover((1, 3, 4))[Hiển thị các slides thứ 1, 3, và 4]
```

Hiển thị các slides thứ 1, 3, và 4

...hay với một dictionary khi sử dụng beginning hay until:

```
#uncover((beginning: 2, until: 4))[Hiển thị các slides phụ 2, 3, và 4]
```





# Các nguyên tắc hiển thị phức tạp

- Cho đến nay, chúng tôi chỉ sử dụng các chỉ mục trang chiếu con duy nhất để xác định thời điểm hiển thị nội dung nào đó.
- Chúng ta cũng có thể sử dụng các mảng số...

```
#uncover((1, 3, 4))[Hiển thị các slides thứ 1, 3, và 4]
```

...hay với một dictionary khi sử dụng beginning hay until:

```
#uncover((beginning: 2, until: 4))[Hiển thị các slides phụ 2, 3, và 4]
```

Hiển thị các slides phụ 2, 3, và 4





# Các nguyên tắc hiển thị phức tạp

- Cho đến nay, chúng tôi chỉ sử dụng các chỉ mục trang chiếu con duy nhất để xác định thời điểm hiển thị nội dung nào đó.
- Chúng ta cũng có thể sử dụng các mảng số...

```
#uncover((1, 3, 4))[Hiển thị các slides thứ 1, 3, và 4]
```

Hiển thị các slides thứ 1, 3, và 4

...hay với một dictionary khi sử dụng beginning hay until:

```
#uncover((beginning: 2, until: 4))[Hiển thị các slides phụ 2, 3, và 4]
```

Hiển thị các slides phụ 2, 3, và 4





# Các nguyên tắc hiển thị phức tạp

- Cho đến nay, chúng tôi chỉ sử dụng các chỉ mục trang chiếu con duy nhất để xác định thời điểm hiển thị nội dung nào đó.
- Chúng ta cũng có thể sử dụng các mảng số...

```
#uncover((1, 3, 4))[Hiển thị các slides thứ 1, 3, và 4]
```

Hiển thị các slides thứ 1, 3, và 4

...hay với một dictionary khi sử dụng beginning hay until:

```
#uncover((beginning: 2, until: 4))[Hiển thị các slides phụ 2, 3, và 4]
```

Hiển thị các slides phụ 2, 3, và 4





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover("-2, 4-6, 8-") [Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

- 1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)
- 4 Tất cả các slides cho tới slide số 4
- 2- Từ slide thứ 2 trở đi
- 3 Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng

1-3 Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)

-4 Tất cả các slides cho tới slide số 4

2- Từ slide thứ 2 trở đi

3 Chỉ hiện thi slide số 3

```
#uncover("-2, 4-6, 8-") [Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```





# Nguyên tắc đơn giản với chuỗi

- Là một tùy chọn ngắn gọn, bạn cũng có thể chỉ định các quy tắc dưới dạng các chuỗi trong một tùy chọn đặc biệt với cú pháp.
- Được phân tách bằng dấu phẩy, bạn có thể sử dụng các quy tắc có dạng  
1-3    Từ slide phụ số 1 tới 3 (bao gồm cả slide số 2)  
      -4    Tất cả các slides cho tới slide số 4  
2-      Từ slide thứ 2 trở đi  
3      Chỉ hiện thi slide số 3

```
#uncover(" -2, 4-6, 8- ")[Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi]
```

Hiển thị các slides số 1, 2, 4, 5, 6, và tất cả các từ slide thứ 8 trở đi





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

```
#only("2-4, 6") [Hiển thị các slides 2, 3, 4, và 6]
```





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: `see how` phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

`#only("2-4, 6")` [Hiển thị các slides 2, 3, 4, và 6]

Hiển thị các slides 2, 3, 4, và 6





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

`#only("2-4, 6")` [Hiển thị các slides 2, 3, 4, và 6]

Hiển thị các slides 2, 3, 4, và 6





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

`#only("2-4, 6")` [Hiển thị các slides 2, 3, 4, và 6]

Hiển thị các slides 2, 3, 4, và 6





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

```
#only("2-4, 6") [Hiển thị các slides 2, 3, 4, và 6]
```





# #only: Bảo tồn không giữ khoảng cách

- Mọi thứ hoạt động với `#uncover` cũng hoạt động với `#only`.
- Tuy nhiên, nội dung hoàn toàn biến mất khi nó không được hiển thị.
- Ví dụ: phần còn lại của câu này di chuyển.
- Một lần nữa, bạn có thể sử dụng các quy tắc chuỗi phức tạp nếu muốn.

`#only("2-4, 6")` [Hiển thị các slides 2, 3, 4, và 6]

Hiển thị các slides 2, 3, 4, và 6





# #alternatives: Thay thế nội dung

- Bạn có thể bị cám dỗ khi thử

```
#only(1)[Ann] #only(2)[Bob] #only(3)[Christopher] likes #only(1)[chocolate] #only(2)
[strawberry] #only(3)[vanilla] ice cream.
```

Ann likes chocolate ice cream.

- Nhưng thật khó để thấy đoạn văn bản nào thực sự thay đổi vì mọi thứ Di chuyển xung quanh.
- Tốt hơn:

```
#alternatives[Ann][Bob][Christopher] likes #alternatives[chocolate][strawberry][vanilla] ice
cream.
```

Ann likes chocolate ice cream.





# #alternatives: Thay thế nội dung

- Bạn có thể bị cám dỗ khi thử

```
#only(1)[Ann] #only(2)[Bob] #only(3)[Christopher] likes #only(1)[chocolate] #only(2)
[strawberry] #only(3)[vanilla] ice cream.
```

Bob likes strawberry ice cream.

- Nhưng thật khó để thấy đoạn văn bản nào thực sự thay đổi vì mọi thứ Di chuyển xung quanh.
- Tốt hơn:

```
#alternatives[Ann][Bob][Christopher] likes #alternatives[chocolate][strawberry][vanilla] ice
cream.
```

Bob            likes strawberry ice cream.





# #alternatives: Thay thế nội dung

- Bạn có thể bị cám dỗ khi thử

```
#only(1)[Ann] #only(2)[Bob] #only(3)[Christopher] likes #only(1)[chocolate] #only(2)
[strawberry] #only(3)[vanilla] ice cream.
```

Christopher likes vanilla ice cream.

- Nhưng thật khó để thấy đoạn văn bản nào thực sự thay đổi vì mọi thứ Di chuyển xung quanh.
- Tốt hơn:

```
#alternatives[Ann][Bob][Christopher] likes #alternatives[chocolate][strawberry][vanilla] ice
cream.
```

Christopher likes vanilla ice cream.





# #one-by-one: An alternative for #pause

#alternatives là để #only khi hàm #one-by-one dùng để #uncover.

#one-by-one hoạt động tương tự như sử dụng #pause nhưng bạn cũng có thể trạng thái khi khám phá nên bắt đầu.

```
#one-by-one(start: 2)[one][by][one]
```

start cũng có thể được bỏ qua, sau đó nó bắt đầu với phần phụ đầu tiên:

```
#one-by-one[one][by][one]
```

one





# #one-by-one: An alternative for #pause

#alternatives là để #only khi hàm #one-by-one dùng để #uncover.

#one-by-one hoạt động tương tự như sử dụng #pause nhưng bạn cũng có thể trạng thái khi khám phá nên bắt đầu.

```
#one-by-one(start: 2)[one][by][one]
one
```

start cũng có thể được bỏ qua, sau đó nó bắt đầu với phần phụ đầu tiên:

```
#one-by-one[one][by][one]
one by
```





# #one-by-one: An alternative for #pause

#alternatives là để #only khi hàm #one-by-one dùng để #uncover.

#one-by-one hoạt động tương tự như sử dụng #pause nhưng bạn cũng có thể trạng thái khi khám phá nên bắt đầu.

```
#one-by-one(start: 2)[one][by][one]
one by
```

start cũng có thể được bỏ qua, sau đó nó bắt đầu với phần phụ đầu tiên:

```
#one-by-one[one][by][one]
one by one
```





# #one-by-one: An alternative for #pause

#alternatives là để #only khi hàm #one-by-one dùng để #uncover.

#one-by-one hoạt động tương tự như sử dụng #pause nhưng bạn cũng có thể trạng thái khi khám phá nên bắt đầu.

```
#one-by-one(start: 2)[one][by][one]
one by one
```

start cũng có thể được bỏ qua, sau đó nó bắt đầu với phần phụ đầu tiên:

```
#one-by-one[one][by][one]
one by one
```





# #line-by-line: Cú pháp dễ dàng với #one-by-one

- Đôi khi thật thuận tiện khi viết các nội dung khác nhau để khám phá một tại một thời điểm trong các dòng tiếp theo.
- Điều này đặc biệt hữu ích cho danh sách dấu đầu dòng, kiểu liệt kê và danh sách thuật ngữ.

```
#line-by-line(start: 2)[
 - first
 - second
 - third
]
```

start là tùy chọn và mặc định là 1.





# #line-by-line: Cú pháp dễ dàng với #one-by-one

- Đôi khi thật thuận tiện khi viết các nội dung khác nhau để khám phá một tại một thời điểm trong các dòng tiếp theo.
- Điều này đặc biệt hữu ích cho danh sách dấu đầu dòng, kiểu liệt kê và danh sách thuật ngữ.

```
#line-by-line(start: 2)[
 - first
 - second
 - third
]
```

- first

start là tùy chọn và mặc định là 1.





# #line-by-line: Cú pháp dễ dàng với #one-by-one

- Đôi khi thật thuận tiện khi viết các nội dung khác nhau để khám phá một tại một thời điểm trong các dòng tiếp theo.
- Điều này đặc biệt hữu ích cho danh sách dấu đầu dòng, kiểu liệt kê và danh sách thuật ngữ.

```
#line-by-line(start: 2)[
 - first
 - second
 - third
]
```

- first
- second

start là tùy chọn và mặc định là 1.





# #line-by-line: Cú pháp dễ dàng với #one-by-one

- Đôi khi thật thuận tiện khi viết các nội dung khác nhau để khám phá một tại một thời điểm trong các dòng tiếp theo.
- Điều này đặc biệt hữu ích cho danh sách dấu đầu dòng, kiểu liệt kê và danh sách thuật ngữ.

```
#line-by-line(start: 2)[
 - first
 - second
 - third
]
```

- first
- second
- third

start là tùy chọn và mặc định là 1.





# Các cách khác nhau để bao phủ nội dung

- Khi nội dung được che phủ, nó sẽ hoàn toàn ẩn theo mặc định.
- Tuy nhiên, bạn cũng có thể chỉ hiển thị nó bằng màu xám nhạt bằng cách sử dụng đối số mode với giá trị "transparent":
- Nội dung bị che phủ sau đó được hiển thị khác nhau.
- Mọi chức năng dựa trên uncover đều có đối số mode tùy chọn:

  - #show: pause(...)
  - #uncover(...)[...]
  - #one-by-one(...)[...][...]
  - #line-by-line(...)[...][...]





# Các cách khác nhau để bao phủ nội dung

- Khi nội dung được che phủ, nó sẽ hoàn toàn ẩn theo mặc định.
- Tuy nhiên, bạn cũng có thể chỉ hiển thị nó bằng màu xám nhạt bằng cách sử dụng đối số mode với giá trị "transparent":
- Nội dung bị che phủ sau đó được hiển thị khác nhau.
  - Mọi chức năng dựa trên uncover đều có đối số mode tùy chọn:
    - #show: pause(...)
    - #uncover(...)[...]
    - #one-by-one(...)[...][...]
    - #line-by-line(...)[...][...]





# Các cách khác nhau để bao phủ nội dung

- Khi nội dung được che phủ, nó sẽ hoàn toàn ẩn theo mặc định.
- Tuy nhiên, bạn cũng có thể chỉ hiển thị nó bằng màu xám nhạt bằng cách sử dụng đối số mode với giá trị "transparent":
- Nội dung bị che phủ sau đó được hiển thị khác nhau.
- Mọi chức năng dựa trên uncover đều có đối số mode tùy chọn:
- #show: pause(...)
- #uncover(...)[...]
- #one-by-one(...)[...][...]
- #line-by-line(...)[...][...]





# Themes





# Các diện mạo mỗi slide được chiếu ...

... được xác định bởi *theme* của bản trình bày.

- Bản trình diễn này sử dụng chủ đề mặc định.
- Chính vì vậy, tiêu đề slide và phần trang trí trên mỗi slide (có tên phần, tiêu đề ngắn, số trang, v.v.) trông giống như cách họ làm.
- Chủ đề cũng có thể cung cấp các biến thể, ví dụ...



... với hàm này!

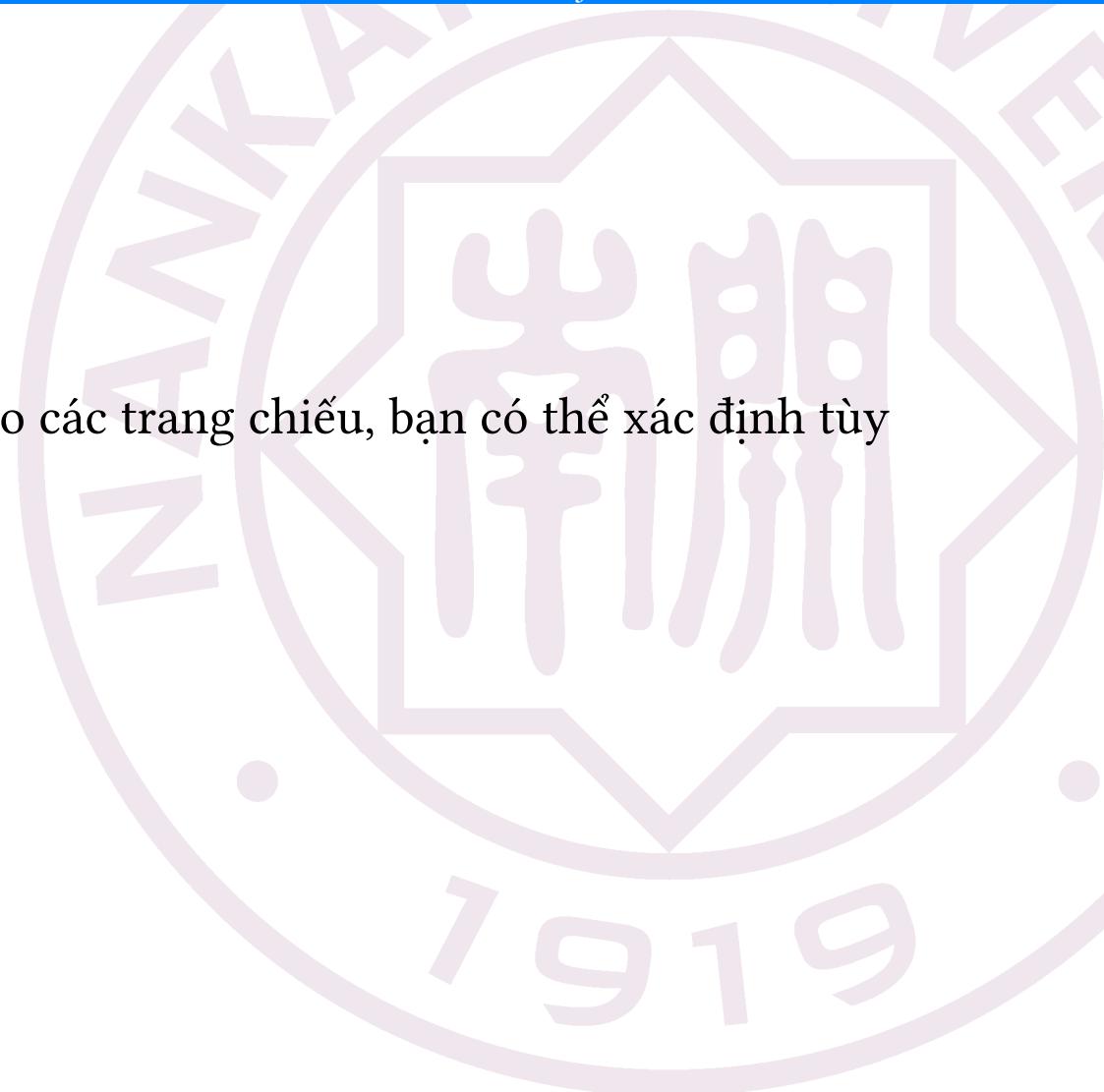
- Nó rất tối giản và giúp khán giả tập trung vào một điểm quan trọng.



# Your own theme?

Nếu bạn muốn tạo thiết kế của riêng mình cho các trang chiếu, bạn có thể xác định tùy chỉnh themes!

Xem [sách này](#) explains how to do so.



# TẠM KẾT



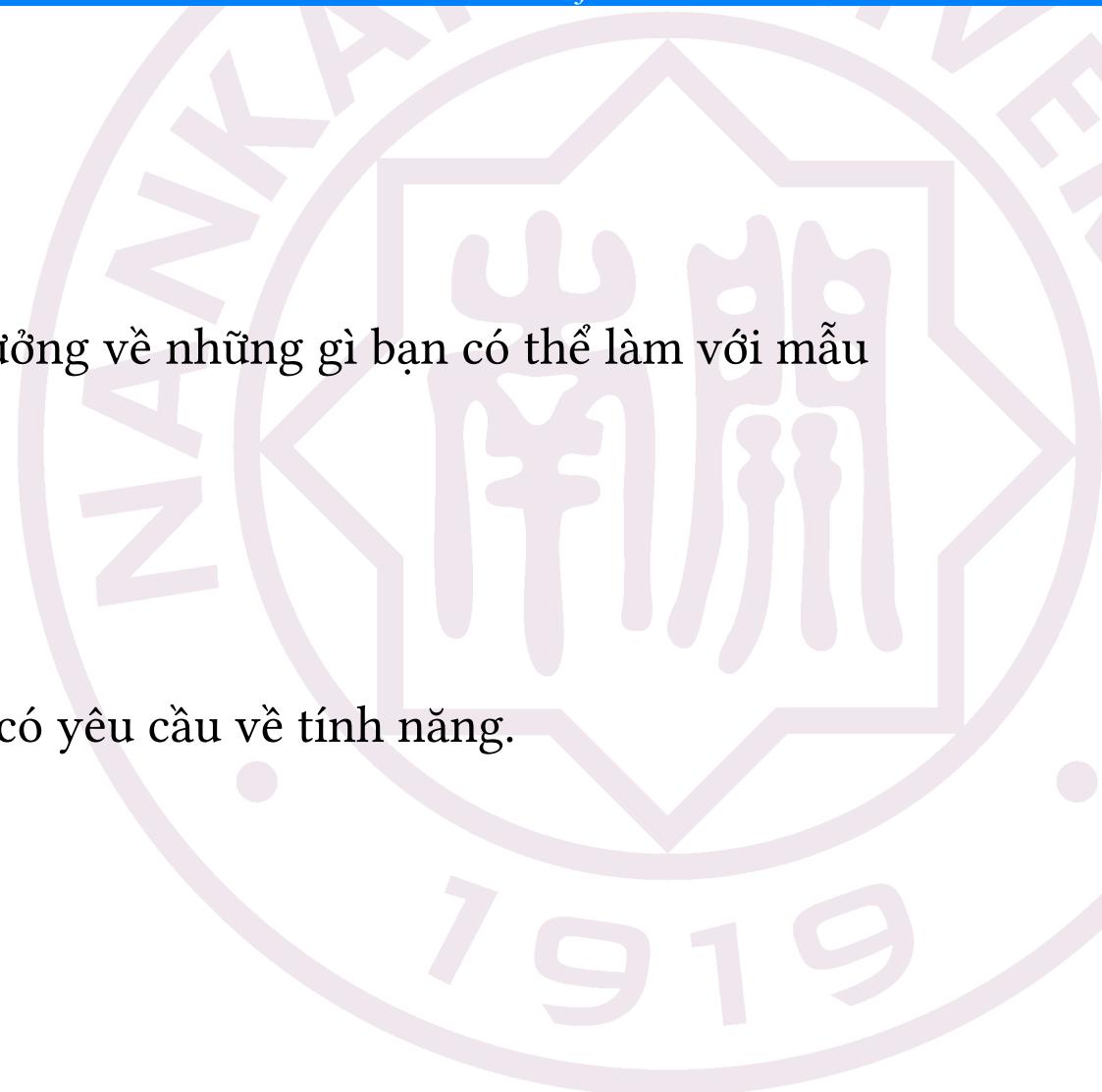


# Chúc các bạn học vui!

- Hy vọng rằng bây giờ bạn đã có một số ý tưởng về những gì bạn có thể làm với mẫu này.
- Mong các bạn cho mình nhiều sao

[GitHub](#) của mình tại đây

- Hoặc mở một issue nếu bạn gặp lỗi hoặc có yêu cầu về tính năng.



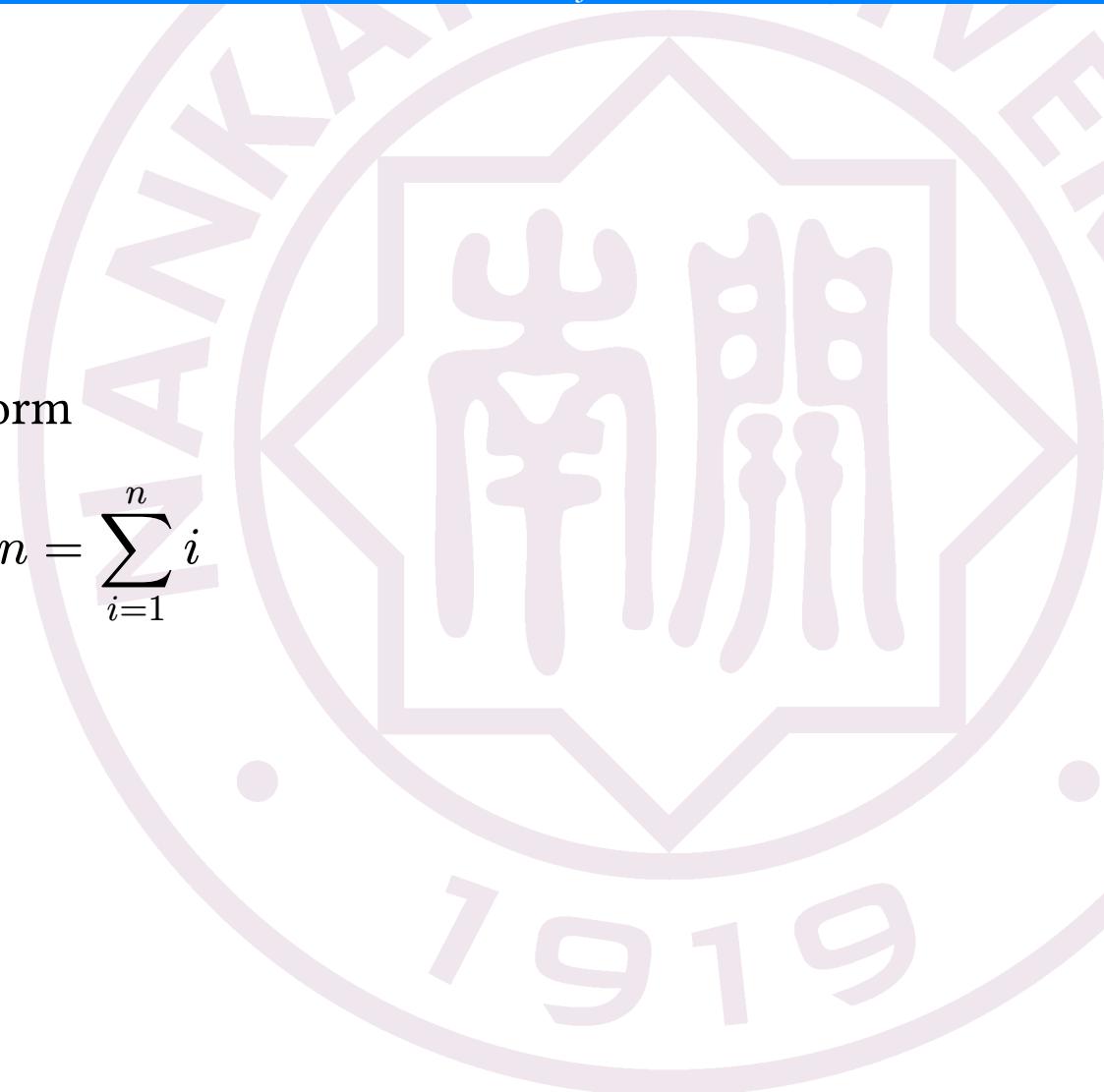
HÃY CHỨNG MINH ĐỊNH LÝ  
GAUSS, TỈNH DÂY NÀO CÁC  
BẠN!



# Problem statement

Let  $n \in \mathbb{N}$ . We are interested in sums of the form

$$1 + \dots + n = \sum_{i=1}^n i$$





# The theorem

I discovered that

$$\sum_{i=1}^n i = \frac{n(n + 1)}{2}$$

Let's prove that!



# CHỨNG MINH





# PHƯƠNG PHÁP CHỨNG MINH

We will prove the theorem by induction, following these steps:

1. base case
2. induction hypothesis
3. induction step





# Proof

**base case:** Let  $n = 1$ . Then  $\sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1 \checkmark$





# Proof

**base case:** Let  $n = 1$ . Then  $\sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1 \checkmark$

**ind. hypothesis:** Let  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$  for some  $k \geq 1$ .





# Proof

**base case:** Let  $n = 1$ . Then  $\sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1 \checkmark$

**ind. hypothesis:** Let  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$  for some  $k \geq 1$ .

**ind. step:** Show that  $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^k i + (k+1)$$





# Proof

**base case:** Let  $n = 1$ . Then  $\sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1 \checkmark$

**ind. hypothesis:** Let  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$  for some  $k \geq 1$ .

**ind. step:** Show that  $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^k i + (k+1) = \frac{k(k+1)}{2} + (k+1)$$





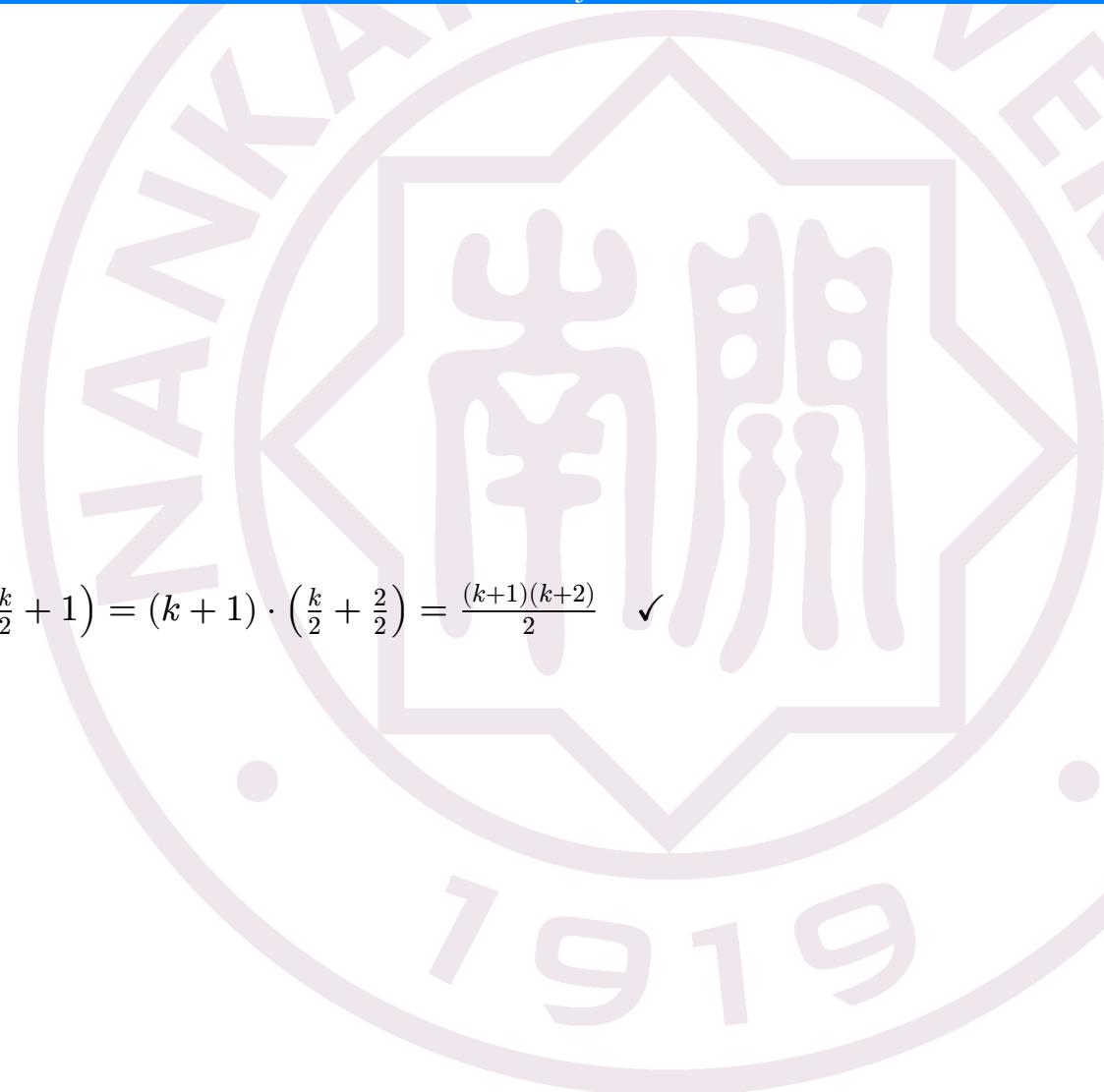
# Proof

**base case:** Let  $n = 1$ . Then  $\sum_{i=1}^1 i = \frac{1 \cdot 2}{2} = 1 \checkmark$

**ind. hypothesis:** Let  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$  for some  $k \geq 1$ .

**ind. step:** Show that  $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^k i + (k+1) = \frac{k(k+1)}{2} + (k+1) = (k+1) \cdot \left(\frac{k}{2} + 1\right) = (k+1) \cdot \left(\frac{k}{2} + \frac{2}{2}\right) = \frac{(k+1)(k+2)}{2} \checkmark$$



CHÚNG TA ĐÃ CHỨNG MINH  
XONG ĐỊNH LÝ GAUSS, TỈNH  
DÂY NÀO CÁC BẠN!