

## บทที่ 4

# ข้อกำหนดเบื้องต้นในการใช้งาน X<sub>Y</sub>TEX

### 4.1 บทนำ

ผู้อ่านได้เรียนรู้แล้วว่าเอกสาร X<sub>Y</sub>TEX ได้จากการประมวลผลไฟล์ “รหัสต้นฉบับ” (source) หรือ “สคริปต์” (script) ซึ่งเดิม TEX และ L<sup>A</sup>TEX ใช้ไฟล์ชนิดที่เข้ารหัสเป็น “แอสกี” (ASCII <sup>1</sup>) แต่เนื่องจากเกิดข้อจำกัดในด้าน “ภาษาต่างชาติที่ไม่ใช่ภาษาอังกฤษ” รวมทั้ง การที่ไม่สามารถใช้งานฟอนต์ได้หลากหลาย ทำให้มีการริเริ่มพัฒนา X<sub>Y</sub>TEX โดยเปลี่ยนมาใช้การเข้ารหัสแบบ “ยูนิโคด” (Unicode <sup>2</sup>) มีข้อดีที่ทำให้สามารถสร้างไฟล์เอกสาร ด้วยภาษาใดๆ ก็ได้ และใช้งานฟอนต์ได้ทั้งแบบ “โพสสคริปต์” (PostScript) “ทรูไทป์” (TrueType) และ “โอเพ่นไทป์” (OpenType) ไฟล์รหัสต้นฉบับ หรือสคริปต์ สำหรับ X<sub>Y</sub>TEX สามารถเตรียมขึ้นด้วยโปรแกรม “เท็กซ์เอดิเตอร์” (text editor) ใดๆ ก็ได้ ในบทนี้ จะแนะนำให้อ่านได้ทราบข้อกำหนดที่จำเป็น ในการสร้างสคริปต์ ผู้อ่านไม่มีความจำเป็นที่จะต้องเตรียมไฟล์ใดๆ รวมทั้ง ไม่จำเป็นที่จะต้องสามารถนำส่วนหนึ่งส่วนใดของคำสั่งต่างๆ ในบทนี้ไปใช้งานได้ เพียงแต่สามารถเข้าใจข้อกำหนดเบื้องต้นเหล่านี้ได้ ก็นับว่าเพียงพอแล้ว เมื่อผ่านการเรียนรู้ในบทต่อๆ ไป ความรู้ในบทนี้จะเป็นส่วนประกอบที่จะทำให้เข้าใจการทำงานได้มากขึ้น ในท้ายที่สุดผู้อ่านจะสามารถจดจำและเข้าใจรายละเอียดต่างๆ ในบทนี้ได้เองเมื่อผ่านการเรียนรู้โครงสร้างของ X<sub>Y</sub>TEX ไปแล้ว

<sup>1</sup>ASCII = American Standard Code for Information Interchange

<sup>2</sup>ซึ่งมีหลายแบบ แต่ตัวที่นิยมกันในขณะนี้คือ “UTF-8”

## 4.2 ชื่อคำสั่งและอาร์กิวเมนต์ (Command names and arguments)

คำสั่ง (commands) คือ คำเฉพาะที่ใช้ในการบอกหรือสั่งให้ X<sub>Y</sub>TeX ดำเนินการอย่างใดอย่างหนึ่งกับไฟล์เอกสาร อาทิ การสั่งให้ X<sub>Y</sub>TeX พิมพ์สัญลักษณ์พิเศษบางตัวออกมา เช่น สัญลักษณ์ทางคณิตศาสตร์ หรือเครื่องหมายโลโก้ต่าง ๆ ตลอดจนการเปลี่ยนแปลงแบบตัวอักษรหรือฟอนท์ หรือจัดรูปแบบการแสดงผลต่าง ๆ เป็นต้น คำสั่งใน X<sub>Y</sub>TeX แบ่งออกได้เป็น 3 ประเภทใหญ่ ๆ คือ

- คำสั่งอักษรเดี่ยว (single characters) เป็นคำสั่งที่ใช้ตัวอักษรพิเศษตัวเดียว ในการทำงานประกอบไปด้วย # \$ & ~ \_ ^ % { และ } อักษรเหล่านี้ล้วนมีหน้าที่เฉพาะของตัวเอง จะอธิบายรายละเอียดในโอกาสต่อไป<sup>3</sup>
- คำสั่งอักษรเดี่ยวนำหน้าด้วยเครื่องหมายทับกลับทาง หรือ “แบคสแลช” (backslash) ใช้ในการพิมพ์สัญลักษณ์ หรือตัวอักษรพิเศษบางตัวที่ไม่สามารถพิมพ์ได้ เนื่องจาก ตัวอักษรเหล่านั้นอาจเป็นคำสั่งอักษรเดี่ยว อาทิ \\$ ใช้สำหรับพิมพ์เครื่องหมายดอลลาร์ “\$” ออกมา เป็นต้น
- คำสั่งนำหน้าด้วยเครื่องหมายทับกลับทาง คำสั่งใน X<sub>Y</sub>TeX ส่วนใหญ่ เป็นคำสั่งชนิดนี้ ใช้สำหรับจัดรูปแบบการแสดงผล หรือทำหน้าที่บางอย่างที่ต้องการ เช่น \XeTeX สำหรับพิมพ์สัญลักษณ์ X<sub>Y</sub>TeX หรือ \large สำหรับการพิมพ์ตัวอักษรด้วยขนาดใหญ่กว่าปกติ เป็นต้น

คำสั่งต่าง ๆ เหล่านี้ บางครั้งมีการเพิ่มเติม “ตัวเลือก” (option) ในลักษณะอาร์กิวเมนต์ (arguments) ใส่ไว้ภายในเครื่องหมายวงเล็บปีกกา (curly braces) หรือวงเล็บเหลี่ยม (brackets) ท้ายคำสั่ง เพื่อกำหนดลักษณะการแสดงผล ซึ่งยังสามารถจำแนกคร่าว ๆ ได้เป็น “อาร์กิวเมนต์ตัวเลือก” (optional) กับ “อาร์กิวเมนต์บังคับ” (mandatory) โดยทั่วไป สามารถเขียนลักษณะของคำสั่งต่าง ๆ ใน X<sub>Y</sub>TeX ที่มีการใช้อาร์กิวเมนต์ได้เป็น

```
\name[optional]{mandatory}
```

<sup>3</sup>ควรทราบว่า การพิมพ์ตัวอักษรเหล่านี้ ปะปนไปในข้อความโดยมิได้ตระหนักว่า เป็นอักษรคำสั่ง ก็อาจก่อให้เกิดผลลัพธ์แปลก ๆ ได้

โดยที่ ส่วนประกอบต่างๆ มีความหมายดังนี้

- name หมายถึง “ชื่อคำสั่ง” ที่เรียกใช้ พิมพ์ด้วยอักษรธรรมดา (letters) เท่านั้น และสิ้นสุดคำสั่ง ด้วยอักขระที่ไม่ใช่ตัวอักษร (non-letters) อาทิ ช่องว่าง (blank) หรือเครื่องหมายแสดงการใช้อาร์กิวเมนต์อย่าง [ และ { เป็นต้น
- optional คือ “อาร์กิวเมนต์ตัวเลือก” อาจมีได้หลายค่า สามารถละเว้นไม่ได้ หากไม่ใช้ตัวเลือกนี้เลยก็สามารถละเว้นไม่พิมพ์เครื่องหมายวงเล็บนี้ได้ด้วย
- mandatory คือ “อาร์กิวเมนต์บังคับ” ต้องกำหนดค่าอย่างใดอย่างหนึ่งลงไป ไม่ใส่ไม่ได้ คำสั่งบางอันอาจใช้ตัวบังคับเลือกมากกว่า 1 ตัว เวลาใช้งานต้องพิมพ์ตัวเลือกบังคับเหล่านี้ให้ครบถ้วนตามลำดับ สลับตำแหน่งกันไม่ได้

*ข้อควรระวัง* คำสั่งใน XeTeX จะสิ้นสุดเมื่อมีช่องว่าง หรืออักขระที่ไม่ใช่ตัวอักษร โดยจะถือว่า ช่องว่างหลังคำสั่งเหล่านั้น เป็นตัวบอกตำแหน่งสิ้นสุดคำสั่ง และจะดูเหมือนเป็นส่วนหนึ่งของคำสั่ง เมื่อพิมพ์คำสั่งปะปนไปกับข้อความอื่นๆ หากไม่ระมัดระวัง ก็อาจได้ผลลัพธ์ไม่ตรงกับความต้องการ เช่น การพิมพ์คำสั่ง `\XeTeX logo` ผลลัพธ์ที่จะได้คือ XeTeXlogo เป็นต้น หากต้องการให้มีช่องว่างระหว่างคำว่า XeTeX และคำว่า logo ก็ต้องระบุลงไปอย่างชัดเจนด้วยการใช้เครื่องหมายวงเล็บปีกกา หรือเครื่องหมายทับกลับทางประกอบลงไปด้วย อาทิ `\XeTeX{} logo` หรือ `\XeTeX\ logo` หรือ `{\XeTeX} logo` ซึ่งจะได้ผลลัพธ์เป็น XeTeX logo เป็นต้น

### 4.3 สภาพแวดล้อม (Environments)

นอกเหนือจากคำสั่งชนิดที่พิมพ์ปะปนไปกับข้อความดังแสดงในหัวข้อที่ผ่านมาแล้ว XeTeX ยังมีคำสั่งอีกประเภทหนึ่งซึ่งเขียนในลักษณะโครงสร้าง หรือสภาพแวดล้อม (environments) โดยมีคำสั่งเปิด (begin) และคำสั่งปิด (end) คล้ายแท็กคำสั่งในภาษา HTML โครงสร้างเป็นดังนี้ คือ

```
\begin{ชื่อของคำสั่งสภาพแวดล้อม}
...
...
\end{ชื่อของคำสั่งสภาพแวดล้อม}
```

การกำหนด หรือควบคุมการทำงาน จะมีผลอยู่เฉพาะเพียงภายในสภาพแวดล้อมเท่านั้น คำสั่งในลักษณะนี้ มีอยู่จำนวนมาก รายละเอียดจะได้กล่าวถึงในโอกาสต่อไป

## 4.4 การประกาศ (Declarations)

“การประกาศ” (declarations) เป็นคำสั่งที่ทำหน้าที่กำหนด หรือแก้ไขค่า “พารามิเตอร์” สำหรับการแสดงผลต่าง ๆ อาทิ การกำหนดให้พิมพ์ข้อความ (ที่กำลังจะเกิดขึ้นหลังการประกาศ) ด้วยตัวอักษรเข้ม หรืออักษรเอียง เป็นต้น โดยมีได้พิมพ์หรือแสดงผลลัพธ์ใด ๆ โดยตรง ค่าพารามิเตอร์ที่ถูกกำหนดจะมีผลต่อการทำงานนับตั้งแต่ประกาศ และจะมีผลอยู่ไปจนกระทั่งมีการกำหนดค่าใหม่ให้กับพารามิเตอร์ตัวเดียวกัน แต่หากมีการประกาศข้อกำหนดไว้ภายใน “สภาพแวดล้อม” ก็จะมีผลอยู่เฉพาะภายในสภาพแวดล้อมนั้นเท่านั้น

อาทิเช่น คำสั่ง `\bfseries` เป็นการประกาศใช้ “ตัวอักษรเข้ม” (bold face) ภายหลังจากประกาศคำสั่งนี้ ก็จะมีผลให้ข้อความต่าง ๆ ที่ต่อท้ายคำสั่งนี้กลายเป็นตัวอักษรเข้มทั้งหมด จนกว่าจะมีการประกาศ `\normalfont` เพื่อกำหนดให้มีการใช้ตัวอักษรแบบปรกติ

สังเกตตัวอย่างการใช้คำสั่งต่อไปนี้

### ตัวอย่างสคริปต์

```
To prove the idea, we designed the MVE, a new user interface device, with regards
to the environment-centered concept. The MVE is a VE that provides
motion. It was designed to be a vehicle for environment-centered navigation.
```

เมื่อนำไปประมวลผล จะให้ผลลัพธ์ดังนี้

#### ตัวอย่างผลการคอมไพล์

To prove the idea, we designed the MVE, a new user interface device, with regards to the environment-centered concept. **The MVE** is a VE that provides motion. It was designed to be a vehicle for environment-centered navigation.

จะสังเกตเห็นว่าข้อความในประโยคที่สอง ตั้งแต่คำว่า “The MVE” เป็นต้นไปจะกลายเป็นตัวอักษรเข้ม หากต้องการให้เพียงแค่คำว่า “The MVE” เป็นตัวอักษรเข้ม ต้องประกาศคำสั่ง `\bfseries` ด้วยวิธีการอย่างใดอย่างหนึ่งในลักษณะดังนี้

#### การกำหนด `\bfseries` แบบที่ 1

To prove the idea, we designed the MVE, a new user interface device, with regards to the environment-centered concept. `{\bfseries The MVE}` is a VE that provides motion. It was designed to be a vehicle for environment-centered navigation.

#### การกำหนด `\bfseries` แบบที่ 2

To prove the idea, we designed the MVE, a new user interface device, with regards to the environment-centered concept. `\bfseries The MVE \normalfont` is a VE that provides motion. It was designed to be a vehicle for environment-centered navigation.

เมื่อประมวลผล จะให้ผลลัพธ์เหมือนกัน ดังนี้

#### ผลการคอมไพล์ สคริปต์ทั้ง 2 แบบ

To prove the idea, we designed the MVE, a new user interface device, with regards to the environment-centered concept. **The MVE** is a VE that provides motion. It was designed to be a vehicle for environment-centered navigation.

## 4.5 หน่วยวัดความยาว (Lengths)

การกำหนดหน่วยวัดความกว้าง ความยาวต่าง ๆ สำหรับจัดรูปแบบการเรียงพิมพ์ด้วย  $\text{X}_{\text{TeX}}$  แบ่งออกเป็น 2 แบบคือ การกำหนดความยาวแบบตายตัว (fixed lengths) และการกำหนดความยาวแบบยืดหยุ่น (rubber lengths)

### 4.5.1 การกำหนดความยาวแบบตายตัว (Fixed Lengths)

การกำหนด “ความยาวแบบตายตัว” (fixed length) คือ การกำหนดค่าความยาว โดยไม่มีการเผื่อระยะขาดเหลือ ขนาดของความยาวที่กำหนด สามารถเป็นได้ทั้งค่าบวก (+) และค่าลบ (-) โดยต้องกำหนด “หน่วยวัด” (unit) ให้กับค่าที่กำหนดด้วย ซึ่งมีให้เลือกใช้ดังต่อไปนี้

cm	centimeter	=	เซนติเมตร (cm)
mm	millimeter	=	มิลลิเมตร (mm)
in	inch	=	นิ้ว (ความยาว 1 in = 2.54 cm)
pt	point	=	พอยต์ (ความยาว 1 in = 72.27 pt)
bp	big point	=	บิกพอยต์ (ความยาว 1 in = 72 bp)
pc	pica	=	พิคา (ความยาว 1 pc = 12 pt)
dd	didot point	=	ดีโดพอยต์ (ความยาว 1,157 dd = 1,238 pt)
cc	cicero	=	คิกเอโร (ความยาว 1 cc = 12 dd)
em		=	ขนาดความยาวเท่ากับความกว้างของตัวอักษรเอ็มพิมพ์ใหญ่ (M)
ex		=	ขนาดความยาวเท่ากับความสูงของตัวอักษรอีพิมพ์เล็ก (e)

ตัวเลขทศนิยมใน  $\text{X}_{\text{TeX}}$  สามารถเขียนได้ทั้งแบบอังกฤษ (English) ซึ่งใช้จุดทศนิยม (period) เป็นตัวแบ่ง และแบบยุโรป (European) ซึ่งใช้เครื่องหมายจุลภาค (comma) กล่าวคือสามารถกำหนดความยาวขนาด 10.5 เซนติเมตรได้เป็น 10.5cm หรือ 10,5cm ทั้งสองแบบ

คำสั่งกำหนดค่าความยาวให้กับพารามิเตอร์ใด ๆ ใน  $\text{X}_{\text{TeX}}$  มีลักษณะดังนี้

```
\setlength{\length_name}{length_spec}
```

โดยที่ *length\_name* คือ ชื่อพารามิเตอร์ที่ต้องการ และ *length\_spec* คือ หน่วยความยาวที่กำหนด อาทิเช่น การกำหนดค่าพารามิเตอร์ `textwidth` ให้มีขนาด 15 cm สามารถทำได้โดยประกาศ `\setlength{\textwidth}{15cm}` เป็นต้น

#### 4.5.2 การกำหนดความยาวแบบยืดหยุ่น (Rubber Lengths)

ความยาวแบบยืดหยุ่น (rubber lengths) คือ ความยาวชนิดที่มีการกำหนดระยะเพื่อให้สามารถบีบ (shrink) หรือขยายออก (stretch) ได้เล็กน้อย มีโครงสร้างคำสั่งคล้ายการกำหนดความยาวแบบตายตัว แต่มีการเพิ่มเติมระยะเพื่อในการกำหนดค่าความยาวด้วยโครงสร้างดังนี้

*nominal\_value plus stretch\_value minus shrink\_value*

อาทิเช่น คำสั่ง `\setlength{\parskip}{1ex plus0.5ex minus0.2ex}` หมายถึง การกำหนดค่าพารามิเตอร์ `parskip` ให้มีขนาดความยาว 1ex โดยที่สามารถยืดขนาดออกได้เป็น 1.5 เท่าของขนาดเดิม และหดลงได้เหลือ 0.8 เท่าของขนาดเดิม โดยเทียบกับขนาดความสูงของตัวอักษร e

#### 4.6 อักขระพิเศษ (Special Characters)

มีข้อควรทราบที่สำคัญ เกี่ยวกับวิธีการเขียนสคริปต์ต้นฉบับ หรือ “ซอร์ส” ของ X<sub>Y</sub>TEX โดยให้การเข้ารหัสด้วย “ยูนิโค้ด” (Unicode) ซึ่งแตกต่างอย่างมากเมื่อเทียบกับการเข้ารหัสเป็น “แอสกี” (ASCII) ของ T<sub>E</sub>X กล่าวคือ ใน T<sub>E</sub>X (ในที่นี้หมายรวมถึง L<sup>A</sup>T<sub>E</sub>X ด้วย) ศาสตราจารย์นุญได้สร้างฟอนท์ขึ้นมาชุดหนึ่ง สำหรับใช้เป็นฟอนท์โดยปริยาย (default font) มีชื่อว่า “Computer Modern” และได้ใส่อัลกอริทึม (algorithm) พิเศษรวมไว้ใน T<sub>E</sub>X เพื่อทำหน้าที่ “ปรับปรุง” การแสดงผลสัญลักษณ์ และเครื่องหมายบางตัว ซึ่งนิยมใช้ในสมการคณิตศาสตร์และภาษาในแถบยุโรป อาทิ ตัวอักษรแสดงการออกเสียง (accents) สัญลักษณ์สกุลเงิน อักขระควบ (ligatures) หรือการพิมพ์อักษรสองตัวติดกันโดยวางตำแหน่งใกล้เคียงกันเป็นพิเศษ อัลกอริทึมที่กล่าวถึงนี้ อาศัยการแปลง “คำสั่ง” ที่กำกับอยู่ในสคริปต์ซึ่งเข้ารหัสไว้เป็นแอสกี เมื่อ X<sub>Y</sub>TEX ได้หันมาใช้ในการเข้ารหัสแบบยูนิโค้ด

และให้อิสระผู้ใช้งานเลือกฟอนท์ตามความชอบ ผลกระทบที่ตามมาก็คือ อัลกอริทึมเดิมที่รวมอยู่ใน TeX ก็จะไม่สามารถปรับปรุงการแสดงผล ได้สวยงามเช่นเดิม เนื่องจากใช้รหัส และฟอนท์ที่แตกต่างกัน ผู้ใช้งานที่เคยคุ้นเคยกับ TeX หรือ L<sup>A</sup>TeX มาก่อนก็อาจสับสนบ้าง ที่ไม่สามารถใช้คำสั่งที่เคยรู้มาก่อนได้เป็นปกติใน X<sub>Y</sub>TeX

อย่างไรก็ดี อันที่จริงแล้ว การใช้รหัสอักขระแบบยูนิโคด มีเจตนาให้ผู้ใช้งานพิมพ์อักขระที่ต้องการเข้าไปโดยตรงได้ด้วยตัวเอง โดยไม่ต้องใช้คำสั่ง (ที่เคยต้องใช้ใน L<sup>A</sup>TeX) กล่าวคือ หากผู้ใช้งานทราบ “รหัสตัวเลข” (numeric codes) ของอักขระที่ต้องการ ก็สามารถพิมพ์เข้าไปโดยตรงผ่านคีย์บอร์ด โดยไม่ต้องกำกับไว้ด้วยคำสั่งของ L<sup>A</sup>TeX เช่น แต่เดิม หากต้องการแสดงตัวอักษร “๕” สำหรับสกุลเงินปอนด์ ผู้ใช้งานต้องพิมพ์คำสั่ง “\pounds” แต่ใน X<sub>Y</sub>TeX ถ้าผู้ใช้ทราบว่า รหัสประจำตัวอักขระของสัญลักษณ์ ๕ คือ “0163” ผู้ใช้ก็สามารถกดปุ่ม “Alt” บนคีย์บอร์ดค้างไว้ แล้วพิมพ์ตัวเลข “0163” ก็จะได้อักขระ ๕ ที่ต้องการ วิธีการแบบนี้ จะไม่สามารถทำได้เมื่อบันทึกไฟล์สคริปต์ด้วยรหัสแอสกี ในแบบที่ L<sup>A</sup>TeX ทำ จึงนับเป็นข้อดีอย่างหนึ่งของ X<sub>Y</sub>TeX

อย่างไรก็ดี แม้วิธีการนี้จะสะดวก และทำให้สามารถใช้ประโยชน์จากรหัสยูนิโคดได้เต็มที่ แต่ก็ยังมีผู้ใช้งานจำนวนไม่น้อยที่คุ้นเคยกับวิธีเขียนคำสั่งของ L<sup>A</sup>TeX แบบเดิม ผู้พัฒนา X<sub>Y</sub>TeX จึงจำเป็นต้องหาทางช่วยเหลือ โดยการเพิ่มเติมวิธีสำหรับ “mapping” หรือ “เทียบโอน” คำสั่งเก่า ให้เข้ากับวิธีการแบบใหม่ โดยผู้ใช้ต้องเพิ่มเติมคำสั่ง “เทียบโอน” ไว้ในส่วนหัวสคริปต์ ดังนี้

```
\defaultfontfeatures{mapping=tex-text}
```

รายละเอียดวิธีการใช้งาน จะแสดงให้เห็นในบทต่อไป

#### 4.6.1 ช่องว่าง (Space or Blank Characters)

“ช่องว่าง” (space characters) เป็นอักขระพิเศษ ที่จะถูกใช้งานแตกต่างจากอักขระตัวอักษรอื่น ๆ โดยในขณะทำการประมวลผล อักขระช่องว่างจะถูกพิจารณาให้เป็นช่องว่างแบบยืดหยุ่น ที่สามารถถูกบีบหรือขยายออก ให้เหมาะสมกับการเรียงพิมพ์ในแต่ละบรรทัด X<sub>Y</sub>TeX มีกติกากในการใช้งาน



ช่องว่างดังนี้

- “ช่องว่างหนึ่งช่อง” มีค่าเท่ากับ “ช่องว่างหลายๆ ช่อง” เมื่อทำการประมวลผล จะมีเพียงอักขระช่องว่างตัวแรกเท่านั้น ที่ถูกนำไปใช้ ขณะเตรียมไฟล์รหัสต้นฉบับ หากต้องการช่องว่าง ผู้อ่านไม่มีความจำเป็นที่จะต้องพิมพ์ช่องว่างหลายๆ ช่อง เนื่องจากเพียงช่องเดียวก็เพียงพอแล้ว บ่อยครั้งที่การพิมพ์ช่องว่างไว้หลายๆ ช่องเป็นประโยชน์ในแง่ช่วยให้ไฟล์รหัสต้นฉบับอ่านเข้าใจได้ง่าย สะดวกในการแก้ไข มากกว่าการหวังผลลัพธ์ของงานพิมพ์
- “ช่องว่างต้นบรรทัด” ไม่มีผลต่อการทำงาน ช่องว่างเหล่านี้จะถูกละเว้นในขณะประมวลผล
- “ช่องว่างหลังคำสั่ง” ไม่มีผลต่อการทำงาน ช่องว่างเหล่านี้จะถูกละเว้นในขณะประมวลผล
- “จุดสิ้นสุดบรรทัด” จะถูกพิจารณาให้มีลักษณะเหมือนเป็นช่องว่าง

#### 4.6.2 เครื่องหมายอัญประกาศ (Quotation Marks)

สำหรับ TeX และโปรแกรมในสายพันธุ์ เมื่อเราต้องการพิมพ์เครื่องหมาย “อัญประกาศ” จะไม่ใช้วิธีการพิมพ์เครื่องหมายอัญประกาศ ที่อยู่บนแป้นพิมพ์ซ้ำสองครั้ง กล่าวคือ เมื่อต้องการพิมพ์เครื่องหมายอัญประกาศ คร่อมข้อความจะไม่ใช้แป้น ” (double quote) หรือ ’ (single quote) ซึ่งอยู่ข้างซ้ายของปุ่ม “Enter” เพราะเมื่อประมวลผลแล้ว จะแสดงผลเป็น ”สวัสดี” หรือ ’สวัสดี’ สังเกตทิศทางของเครื่องหมายอัญประกาศ ใน TeX จะใช้แป้น 2 ตัวคือ ‘ (grave accent) และ ’ (single quote) ซึ่งอยู่ต่างแป้นพิมพ์กัน<sup>4</sup> อาทิ ‘สวัสดี’ และใช้การพิมพ์เครื่องหมาย ‘ หรือ ’ ซ้ำกัน 2 ครั้งเมื่อต้องการพิมพ์ว่า “สวัสดี” เป็นต้น<sup>5</sup>

<sup>4</sup>เครื่องหมาย ‘ (grave accent) อยู่บนแป้นมุมบนซ้ายของคีย์บอร์ด ดังนั้นผู้อ่านที่ใช้ระบบปฏิบัติการ MS Windows และได้กำหนดให้ใช้แป้น grave accent เป็นตัวสลับภาษาบนคีย์บอร์ด ควรเปลี่ยนไปใช้แป้นอื่นแทน

<sup>5</sup>การพิมพ์ในลักษณะนี้ อาจฝืนความรู้สึกของผู้อ่านที่คุ้นเคยกับการใช้โปรแกรมประมวลผลคำอย่าง MS Word บ้าง เนื่องจากโปรแกรมเหล่านั้น มักเปลี่ยนเครื่องหมาย ” ที่เหมือนกัน 2 อัน บนแป้นพิมพ์ให้เป็น “ และ ” โดยอัตโนมัติ โดยที่ผู้ใช้งานพิมพ์เครื่องหมายดังกล่าวด้วยแป้นพิมพ์ตัวเดียวกัน อย่างไรก็ตามผู้อ่านที่กำลังหัดใช้ XeTeX ควรฝึกฝนให้คุ้นเคยกับวิธีการพิมพ์ตามแบบฉบับของ TeX เพื่อลดความสับสน และปัญหาอื่นๆ ที่อาจตามมา

### 4.6.3 อักขระคำสั่ง (Command Characters)

ดังที่ได้กล่าวมาแล้วในหัวข้อ 4.2 เรื่อง “คำสั่งอักขระเดี่ยว” เมื่อต้องการพิมพ์อักขระเหล่านี้ลงในเอกสาร จะต้องใส่เครื่องหมาย “ทับกลับทาง” (backslash) นำหน้าไว้ เพื่อบอกให้  $\text{X}\text{T}\text{E}\text{X}$  รู้ว่าไม่ใช่คำสั่ง คำสั่งสำหรับพิมพ์ตัวอักขระคำสั่ง มีดังต่อไปนี้

$\$ = \$$        $\& = \&$        $\% = \%$        $\# = \#$        $\_ = \_$        $\{ = \{$        $\} = \}$

### 4.6.4 เครื่องหมายพิเศษ (Special Characters)

เครื่องหมายพิเศษ คือ เครื่องหมายที่ไม่ปรากฏอยู่บนแป้นพิมพ์ สามารถเรียกใช้งานได้อย่างน้อย 2 วิธี คือ วิธีดั้งเดิมของ  $\text{T}\text{E}\text{X}$  คือการใช้คำสั่ง และวิธีการพิมพ์เข้าไปในสคริปต์โดยตรงของ  $\text{X}\text{T}\text{E}\text{X}$  วิธีการหลังนี้ จำเป็นต้องรู้ “รหัสตัวเลข” (numeric codes) ของตัวอักขระ เมื่อต้องการใช้ ให้กดแป้น “Alt” (alternate) ค้างไว้ แล้วพิมพ์ตัวเลขรหัส 4 ตัว ดังแสดงในตารางต่อไปนี้

ตารางที่ 4.1: อักขระพิเศษ คำสั่ง และรหัสประจำตัว

อักขระ (character)	คำสั่ง (command)	รหัสประจำตัว (numeric code)
À	$\backslash\{A\}$	0192
È	$\backslash\{E\}$	0200
Ì	$\backslash\{I\}$	0204
Ò	$\backslash\{O\}$	0210
Ù	$\backslash\{U\}$	0217
à	$\backslash\{a\}$	0224
è	$\backslash\{e\}$	0232

อักขระพิเศษ คำสั่ง และรหัสประจำตัว (ต่อ)

อักขระ (character)	คำสั่ง (command)	รหัสประจำตัว (numeric code)
ì	\`{i}	0236
ò	\`{o}	0242
ù	\`{u}	0249
Á	\'{A}	0193
É	\'{E}	0201
Í	\'{I}	0205
Ó	\'{O}	0211
Ú	\'{U}	0218
Ý	\'{Y}	0221
á	\'{a}	0225
é	\'{e}	0233
í	\'{i}	0237
ó	\'{o}	0243
ú	\'{u}	0250
ý	\'{y}	0253
Â	\^{A}	0194
Ê	\^{E}	0202

---

 อักขระพิเศษ คำสั่ง และรหัสประจำตัว (ต่อ)
 

---

อักขระ (character)	คำสั่ง (command)	รหัสประจำตัว (numeric code)
İ	<code>\^{I}</code>	0206
Ō	<code>\^{O}</code>	0212
Ū	<code>\^{U}</code>	0219
â	<code>\^{a}</code>	0226
ê	<code>\^{e}</code>	0234
î	<code>\^{i}</code>	0238
ô	<code>\^{o}</code>	0244
û	<code>\^{u}</code>	0251
Ã	<code>\~{A}</code>	0195
Ñ	<code>\~{N}</code>	0209
Õ	<code>\~{O}</code>	0213
ã	<code>\~{a}</code>	0227
ñ	<code>\~{n}</code>	0241
õ	<code>\~{o}</code>	0245
Ä	<code>\" {A}</code>	0196
Ë	<code>\" {E}</code>	0203
Ï	<code>\" {I}</code>	0207

---

อักขระพิเศษ คำสั่ง และรหัสประจำตัว (ต่อ)

อักขระ (character)	คำสั่ง (command)	รหัสประจำตัว (numeric code)
Ö	\{O}	0214
Ü	\{U}	0220
ÿ	\{Y}	0159
ä	\{a}	0228
ë	\{e}	0235
ï	\{i}	0239
ö	\{o}	0246
ü	\{u}	0252
ÿ	\{y}	0255
Œ	\OE	0140
œ	\oe	0156
ß	\ss	0223
∅	\O	0216
ø	\o	0248
Å	\AA	0197
å	\aa	0229
Æ	\AE	0198

## อักขระพิเศษ คำสั่ง และรหัสประจำตัว (ต่อ)

อักขระ (character)	คำสั่ง (command)	รหัสประจำตัว (numeric code)
æ	<code>\ae</code>	0230
þ	<code>\TH</code>	0222
þ	<code>\th</code>	0254
Ð	<code>\DH</code>	0208
ð	<code>\dh</code>	0240
Š	<code>\v{S}</code>	0138
š	<code>\v{s}</code>	0154
Ž	<code>\v{Z}</code>	0142
ž	<code>\v{z}</code>	0158
Š	<code>\S</code>	0167
†	<code>\dag</code>	0134
‡	<code>\ddag</code>	0135
¶	<code>\P</code>	0182

“รหัสตัวเลข” ที่แสดงในตาราง 4.1 นี้ เป็นเพียงบางส่วนของรหัสทั้งหมด<sup>6</sup> ควรทราบว่า บ่อยครั้งเมื่อผู้ใช้งาน TeX ต้องการใช้อักขระบางตัวที่ไม่ได้อยู่ใน “แพ็คเกจมาตรฐาน” ก็จำเป็นต้องเรียกใช้แพ็คเกจพิเศษอื่นๆ อาทิ เมื่อต้องการใช้สัญลักษณ์ทางคณิตศาสตร์บางตัว ก็อาจเรียกใช้” แพ็คเกจ “amssymb” โดยใช้คำสั่ง “`\usepackage{amssymb}`” สังเกตตอนต้นไฟล์ตัวอย่าง xetex-004.tex ที่ได้แสดงไว้แล้ว

<sup>6</sup>[http://en.wikipedia.org/wiki/List\\_of\\_XML\\_and\\_HTML\\_character\\_entity\\_references](http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references)

#### 4.6.5 การใช้แพ็คเกจเครื่องหมายอื่น ๆ

ที่ผ่านมา มีผู้พัฒนาแพ็คเกจสำหรับอำนวยความสะดวกในการแสดงสัญลักษณ์ และเครื่องหมายต่าง ๆ ในเอกสารที่สร้างด้วย T<sub>E</sub>X และ L<sub>A</sub>T<sub>E</sub>X ใหม่มากมาย และเนื่องจาก X<sub>Y</sub>T<sub>E</sub>X ถูกพัฒนาต่อจาก T<sub>E</sub>X จึงมีความสามารถในการใช้งานแพ็คเกจเหล่านี้ไปด้วย ในกรณีของ X<sub>Y</sub>T<sub>E</sub>X แม้ผู้ใช้งานจะสามารถพิมพ์สัญลักษณ์เหล่านี้โดยอาศัย “รหัสตัวเลข” ดังที่แสดงในหัวข้อที่ผ่านมา แต่สำหรับผู้ใช้งานจำนวนไม่น้อย การจดจำสัญลักษณ์ที่ต้องการ “เป็นคำสั่ง” แทนการจดจำเป็นตัวเลข ก็อาจเป็นอะไรที่สะดวกมากกว่า การทำความรู้จัก “แพ็คเกจ” บางตัวที่มีอยู่ใน L<sub>A</sub>T<sub>E</sub>X แม้จะเป็นเทคนิคแบบเดิม แต่ก็น่าจะเป็นประโยชน์ให้เราสามารถใช้งาน X<sub>Y</sub>T<sub>E</sub>X ได้กว้างขวางมากขึ้น

แพ็คเกจเหล่านี้มีวิธีการใช้งานคล้ายกัน กล่าวคือ ผู้ใช้ต้อง “เรียกใช้แพ็คเกจ” ไว้ในตอนต้นสคริปต์ อาทิ การเรียกใช้แพ็คเกจ textcomp ต้องประกาศไว้ที่ส่วนหัวไฟล์ดังนี้

```
\usepackage{textcomp}
```

ทั้งนี้ สัญลักษณ์ที่สามารถใช้ใน X<sub>Y</sub>T<sub>E</sub>X ได้ มีเป็นจำนวนมาก ผู้เขียนไม่สามารถนำมาแสดงในหนังสือฉบับนี้ได้ทั้งหมด ผู้อ่านควรศึกษาเพิ่มเติมจากเว็บไซต์ของ CTAN <sup>7</sup>

L<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> มาตรฐาน (ไม่ต้องเรียกแพ็คเกจเพิ่มเติม) :

^	=	\textasciicircum	~	=	\textasciitilde
*	=	\textasteriskcentered	\	=	\textbackslash
	=	\textbar	{	=	\textbraceleft
}	=	\textbraceright	•	=	\textbullet
©	=	\textcopyright	†	=	\textdagger
‡	=	\textdaggerdbl	\$	=	\textdollar
...	=	\textellipsis	-	=	\textemdash
-	=	\textendash	;	=	\textexclamdown

<sup>7</sup><http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>

>	=	\textgreater	<	=	\textless
♀	=	\textordfeminine	♂	=	\textordmasculine
¶	=	\textparagraph	·	=	\textperiodcentered
¿	=	\textquestiondown	"	=	\textquotedblleft
"	=	\textquotedblright	'	=	\textquoteleft
'	=	\textquoteright	®	=	\textregistered
§	=	\textsection	£	=	\textsterling
□	=	\textvisiblespace	_	=	\textunderscore

**แพ็คเกจ textcomp :**

฿	=	\textbaht	¢	=	\textcent
¢	=	\textcentoldstyle	₯	=	\textcolonmonetary
₭	=	\textcurrency	\$	=	\textdollar
\$	=	\textdollaroldstyle	₫	=	\textdong
€	=	\texteuro	f	=	\textflorin
₲	=	\textguarani	₺	=	\textlira
₪	=	\textnaira	₱	=	\textpeso
£	=	\textsterling	₩	=	\textwon
¥	=	\textyen	Ⓟ	=	\textcircledP
©	=	\textcopyleft	©	=	\textcopyright
®	=	\textregistered			

**แพ็คเกจ eurosym :**

€	=	\geneuro	€	=	\geneuronarrow
€	=	\geneurowide	€	=	\officiaeuro

นอกจากนี้ ยังมีแพ็คเกจอื่นๆ อีกมาก ซึ่งขอละไว้ไม่กล่าวถึงในที่นี้



#### 4.6.6 วันที่ (Dates)

เมื่อต้องการพิมพ์ “วันที่” ลงในเอกสารโดยอัตโนมัติ (ตามวันที่ทำการประมวลผลสคริปต์) ทำได้โดยใช้คำสั่ง `\today` ณ ตำแหน่งข้อความที่ต้องการได้ อาทิ เช่น เมื่อสั่งคำสั่งดังกล่าว ณ ตำแหน่งนี้จะได้ผลลัพธ์เป็น “July 2, 2009” ซึ่งเป็นวันที่ต้นฉบับนี้ถูกประมวลผล เป็นต้น

### 4.7 การปรับแต่งข้อความ (Fine-tuning text)

#### 4.7.1 การเว้นช่องว่าง (Spacing)

เมื่อต้องการแทรก “ช่องว่าง” (space) ลงในข้อความ สามารถทำได้ด้วยคำสั่ง 2 แบบต่อไปนี้ โดยที่ “*space*” คือ ระยะความยาวที่ต้องการ อาทิ 2cm หมายถึง ความยาว 2 เซนติเมตร เป็นต้น

- `\hspace{space}` เป็นการเว้นช่องว่างขนาด *space* แต่จะไม่มีผลต่อการทำงาน หากเรียกใช้คำสั่งที่ต้นบรรทัด กล่าวคือ จะตัดช่องว่างทั้งหมดที่ต้นบรรทัดทิ้งไป
- `\hspace*{space}` คำสั่งนี้จะแทรกช่องว่าง ขนาด *space* ณ จุดที่มีการเรียกใช้งานทันที โดยไม่สนใจว่าคำสั่งปรากฏอยู่ ณ ตำแหน่งใด

สังเกตสคริปต์ และผลลัพธ์ต่อไปนี้

```
\hspace*{2cm}ทดสอบการทำงาน a \hspace{3cm}ทดสอบ
\hspace{2cm}ทดสอบการทำงาน b \hspace{3cm}ทดสอบ
```

- ผลลัพธ์ -

ทดสอบการทำงาน a	ทดสอบ
ทดสอบการทำงาน b	ทดสอบ

นอกจากนี้ ยังสามารถใช้คำสั่ง `\hfill` ซึ่งเป็นคำสั่งแบบลดรูป ของ `\hspace{\fill}` ในการเว้นช่องว่าง ชนิดผลักให้ข้อความที่อยู่ด้านซ้ายและขวาของคำสั่งนี้ ชิดขอบด้านซ้ายและขวาตามลำดับ สังเกตการเขียนคำสั่งต่อไปนี้

ด้านซ้าย `\hfill` ด้านขวา

- ผลลัพธ์ -

ด้านซ้าย

ด้านขวา

หากใช้คำสั่ง `\hfill` แบ่งข้อความออกเป็น 3 ส่วน ก็จะมีผลทำให้สามารถจัดการแสดงผลในลักษณะ 3 คอลัมน์ คือ ซิดซ้าย กลาง และซิดขวา ดังนี้

ด้านซ้าย `\hfill` กลาง `\hfill` ด้านขวา

- ผลลัพธ์ -

ด้านซ้าย

กลาง

ด้านขวา

#### 4.7.2 การเติมจุดไข่ปลา และเส้นบรรทัด (Dots and Ruled Line)

นอกเหนือไปจากการเว้นช่องว่าง ดังที่ได้กล่าวไว้แล้ว ผู้อ่านสามารถเติมจุดไข่ปลา หรือเส้นบรรทัดระหว่างช่องว่างใดๆ ได้ด้วยคำสั่ง `\dotfill` และ `\hrulefill` ตามลำดับ อาทิ

ด้านซ้าย `\dotfill{}` ด้านขวา

ด้านซ้าย `\dotfill{}` กลาง `\dotfill{}` ด้านขวา

ด้านซ้าย `\hrulefill{}` กลาง `\hrulefill{}` ด้านขวา

- ผลลัพธ์ -

ด้านซ้าย ..... ด้านขวา

ด้านซ้าย ..... กลาง ..... ด้านขวา

ด้านซ้าย \_\_\_\_\_ กลาง \_\_\_\_\_ ด้านขวา

### 4.7.3 การเว้นบรรทัด (Line Breaking)

เมื่อสิ้นสุดข้อความ และต้องการขึ้นบรรทัดใหม่ สามารถทำได้โดยอาศัยเครื่องหมาย “\” เป็นตัวบอกให้ X<sub>Y</sub>TeX รู้ว่าสิ้นสุดบรรทัดแล้ว และยังสามารถกำหนด “ช่องว่าง” (space) ระหว่างบรรทัดเพิ่มเติม นอกเหนือจากการขึ้นบรรทัดใหม่ด้วยคำสั่ง “\” ได้อีกด้วย

คำสั่งเว้นบรรทัดแบบกำหนดช่องว่างเพิ่มเติม สามารถทำได้ 2 แบบ คือ

`\\[space]`

หากสิ้นสุดบรรทัดนี้ แล้วบังเอิญหมดหน้ากระดาษพอดี ต้องขึ้นหน้าใหม่ X<sub>Y</sub>TeX จะตัดช่องว่างที่กำหนดไว้ใน space ทิ้งไป แล้วเริ่มต้นหน้าใหม่ที่บรรทัดแรกของหน้า

`\\*[space]`

จะไม่สนใจการขึ้นหน้าใหม่ แต่จะเว้นช่องว่างระหว่างบรรทัดตามขนาดที่กำหนด

อาทิเช่น คำสั่ง `\\*[10cm]` จะมีผลให้สิ้นสุดบรรทัด ณ ตำแหน่งที่เรียกใช้คำสั่งนี้ จากนั้นเว้นช่องว่างไว้ 10 เซนติเมตร โดยไม่สนใจว่าจะมีการขึ้นหน้าใหม่หรือไม่ หากมีการขึ้นหน้าใหม่ก็ยังคงเว้นช่องว่างต่อไปจนครบความยาว 10 เซนติเมตร จึงจะเริ่มข้อความในบรรทัดต่อไป

นอกจากนี้ยังสามารถใช้คำสั่ง `\newline` สำหรับการขึ้นบรรทัดใหม่ได้อีกด้วย คำสั่งนี้ไม่มีตัวเลือกกำหนดช่องว่างหลังการขึ้นบรรทัดใหม่ และเมื่อจำเป็นต้องขึ้นหน้าใหม่ก็จะเริ่มต้นบรรทัดต่อไปที่บรรทัดแรกของหน้า

ทั้งคำสั่ง `\\` และ `\newline` ใช้ได้ในข้อความแต่ละย่อหน้าเท่านั้น ไม่สามารถประกาศใช้ลอยๆ ระหว่างย่อหน้าแต่ละอันได้

#### 4.7.4 ช่องว่างในแนวตั้ง (Verticle Spacing)

สามารถกำหนด “ช่องว่างในแนวตั้ง” (vertical space) ระหว่างย่อหน้า ได้ด้วยคำสั่ง `\vspace` ซึ่งมี 2 ลักษณะคือ

`\vspace{space}`

ลักษณะพิเศษคือ หากต้องขึ้นหน้าใหม่ระหว่างช่องว่างนี้  $\text{X}_{\text{TeX}}$  จะตัดช่องว่างตามความยาวที่กำหนดให้กับ `space` ทิ้งไปแล้วเริ่มต้นหน้าใหม่ที่บรรทัดแรกของหน้า

`\vspace*{space}`

จะไม่สนใจการขึ้นหน้าใหม่ แต่จะเว้นช่องว่างระหว่างย่อหน้าตามขนาดที่กำหนด

ผู้อ่านยังสามารถใช้คำสั่ง `\fill` ในลักษณะเดียวกับ `\hfill` ที่ได้อธิบายไปแล้วอีกด้วย เพียงแต่ในกรณีนี้จะจัดการแสดงผลในแนวตั้งแทน

คำสั่งอื่นๆ ที่ใช้ในการเว้นช่องว่างระหว่างย่อหน้าประกอบไปด้วย `\bigskip` `\medskip` และ `\smallskip`

#### 4.7.5 การขึ้นหน้าใหม่ (Page Breaking)

คำสั่งสำหรับการขึ้นหน้าใหม่ คือ `\pagebreak[num]` โดยที่ `num` คือ หมายเลขแสดงระดับความสำคัญในการขึ้นหน้าใหม่ มีค่าระหว่าง 0 ถึง 4 ตัวเลขมาก หมายถึงระดับความสำคัญมาก ซึ่งจะระบุถึงระดับการบังคับให้มีการขึ้นหน้าใหม่

คำสั่งที่น่าจะทำความเข้าใจในการใช้งานได้ง่าย และควบคุมการทำงานได้แม่นยำกว่า คือ คำสั่ง `\newpage` ซึ่งจะมีผลให้ขึ้นหน้าใหม่หลังสิ้นสุดคำสั่ง

#### 4.7.6 เครื่องหมายติ๊ก (Hyphens and Dashes)

เครื่องหมาย “ยติ๊ก” (hyphens) มี 3 แบบ ตามขนาดความยาวของเครื่องหมายคือ

- “-” ขนาดสั้น (hyphen) ใช้สำหรับการแบ่งคำที่มีความยาวเกินกว่าจะพิมพ์ให้หมดในบรรทัดเดียวกันได้ โดยปกติ เครื่องหมายติ๊กขนาดสั้นนี้ จะถูกสร้างขึ้นโดยอัตโนมัติ ในขณะที่ประมวลผล ไม่จำเป็นต้องพิมพ์ขึ้นเอง สร้างขึ้นจากการพิมพ์เครื่องหมาย “-” บนแป้นพิมพ์เพียง 1 ครั้ง
- “—” ขนาดกลาง (en dash) ใช้สำหรับการเขียนช่วงของตัวเลข เช่น 1—10 เป็นต้น สร้างขึ้นจากการพิมพ์เครื่องหมาย “-” บนแป้นพิมพ์ติดกัน 2 ครั้ง ดังนี้คือ “--”
- “——” ขนาดยาว (em dash) ใช้สำหรับการแบ่งวรรคตอนในภาษาอังกฤษ ซึ่งโดยปกติภาษาไทยคงไม่ใช้เครื่องหมายนี้ สร้างขึ้นจากการพิมพ์เครื่องหมาย “-” บนแป้นพิมพ์ติดกัน 3 ครั้ง ดังนี้คือ “---”

### 4.8 สรุป

ในบทนี้ ผู้เขียนได้อธิบายให้ผู้อ่านทราบหลักการกว้างๆ ในการประกาศใช้คำสั่งต่างๆ ของ X<sub>Y</sub>TEX ซึ่งแบ่งประเภทคำสั่งออกเป็น 3 ลักษณะ คือ (1) “คำสั่งอักษรเดี่ยว” มีลักษณะเป็นตัวอักษรตัวเดียว ซึ่ง X<sub>Y</sub>TEX ใช้ควบคุมการทำงานบางอย่าง (2) “คำสั่งอักษรเดี่ยวที่นำหน้าด้วยเครื่องหมาย \” และ (3) “คำสั่งอื่นๆ ที่นำหน้าด้วยเครื่องหมาย \” การประกาศใช้คำสั่งเหล่านี้ มีทั้งแบบพิมพ์ปะปนไปในข้อความต่างๆ และแบบที่ต้องกำหนดเป็น “สภาพแวดล้อม” ซึ่งมีคู่คำสั่ง “เปิด” และ “ปิด” ในลักษณะ `\begin{ชื่อของคำสั่งสภาพแวดล้อม} ... \end{ชื่อของคำสั่งสภาพแวดล้อม}`

คำสั่งของ X<sub>Y</sub>TEX มีอยู่เป็นจำนวนมาก รวมทั้งมีรายละเอียดวิธีการใช้งานที่ต่างๆ กัน ผู้เขียนคงไม่สามารถรวบรวมรายการคำสั่งทั้งหมดในหนังสือเล่มนี้ได้ การจะสามารถใช้งาน X<sub>Y</sub>TEX ได้เต็มประสิทธิภาพ ผู้อ่านจึงจำเป็นต้องติดตาม “คู่มือการใช้งานแบบออนไลน์” ซึ่งมีรายละเอียดที่สำคัญที่ควรทราบ คู่มือเหล่านี้ สามารถค้นหาได้จากอินเทอร์เน็ต โดยเฉพาะจากเว็บไซต์ของ CTAN