The Non-Latin scripts & typography

Kamal Mansour

1 Introduction

It is quite common in typographic terminology to divide the world's scripts into Latin and non-Latins. At first glance that might seem to be a reasonable categorization until one looks a bit closer to find that non-Latin consists of a huge number of diverse scripts. Imagine if we were to call Latin script blue, while calling all others non-blue. We would be gathering the remaining colors of the spectrum into one group without differentiation. Calling a color non-blue doesn't tell us anything about what it might be; is it orange, green, magenta, indigo?

However strange this terminology may be, there is good reason behind it. Gutenberg invented moveable type for Latin script; in particular, he focused on a certain Gothic style used at the time by German scribes. We know things didn't stop there. The printed forms of Latin script moved away from the handwritten forms over time, evolving into a distinct craft and discipline. In the process, the letter forms underwent considerable simplification. By the time typographic letters began to develop for other scripts, Latin type had reached a maturity possible only through time. Over the centuries, the machinery and techniques had been refined for the needs of Latin type and any newcomers to the game needed to adapt to the current state of things.

In the realm of non-Latin scripts, what are the main groups and families? Moving eastward from the origins of Latin script, we find Greek script, and further east, its close relative, Cyrillic. Hovering over the Caucasus, Armenian and Georgian raise their heads. Reaching from North Africa eastward, Arabic script occupies a large swath of the landscape that reaches as far as India. Near the juncture of Western Asia and North Africa, Hebrew has its home. In East Africa, Ethiopic script — in ancient times know as Ge'ez — has a sizable presence. Over the large territory of China, as well as in Taiwan, Chinese is the dominant script. Hangul, a syllabic script with some Chinese visual features, dominates in Korea. Meanwhile, further south in Japan, rules a unique hybrid writing system consisting of two syllabaries (katakana and hiragana), Chinese characters, in addition to Latin. The Indian subcontinent is home for a large family of scripts descended from ancient Brahmi writing. Over the centuries, the descendants split into two groups, northern and southern scripts. In the northern group, Devanagari, Bengali, and Gujarati scripts stand out. The southern group includes Tamil, Telugu, Kannada, Malayalam, and Sinhala. In Southeast Asia, we find more distant descendants of Brahmi in Thai, Lao, Khmer, and Myanmar scripts.

As close relatives of Latin, the Greek and Cyrillic scripts had more time to develop typographically than other non-Latins. In its earliest typographic forms, Greek initially imitated cursive scribal form before eventually settling on printed forms based on its classic origins. Classical and Biblical studies further established the need for Greek typefaces in the late nineteenth and early twentieth centuries.

Cyrillic, originally derived from Greek, was extensively revised and "europeanized" under the hand of Peter the Great. Then early in the twentieth century, Cyrillic was further simplified by purging it of unnecessary vestiges from Greek orthography and redundant characters. In an effort to further unite the huge territory of the Soviet Union, the government extended Cyrillic script to support the many non-Slavic languages of the various republics. Outside the Soviet Union, a certain number of Cyrillic typefaces were always available through the mainstream type foundries to satisfy the needs of academics in Slavic Studies and of emigrant communities.

Because of their common shapes and behavior, Greek and Cyrillic have always been easily accommodated on the same typesetting equipment as Latin script. Of course, the compositor had to be familiar with the script he was setting, with Polytonic Greek being the most demanding because of the variety of diacritics.

When it comes to typography, every script has its intricate details, and consequently, its own story of transition from handwriting to type. It would be interesting to tell every story, but that would be enough to fill many books. Without going into every detail, we can gain some understanding by surveying the types of problems that arose when adapting typesetting technology developed for Latin script to non-Latin scripts. Because scripts have developed as families, they share many attributes with other members of the family. We can take advantage of the similarities to identify recurring obstacles in the typographic development of non-Latin scripts. By citing judiciously chosen examples from a few scripts, we can readily present the gamut of significant difficulties.

Before delving into the range of technical challenges, we must first recognize the most common hurdle shared by all who endeavored to bring a non-Latin script into the typographic age: the fear of the exotic. Typography developed in Europe and it was

Europeans who first strove to take this craft to distant lands. Before the advent of sociology, anthropology, and linguistics, non-European cultures were seen as foreign, exotic, and difficult to fathom. Before understanding the written form of a language, one must first learn the spoken language and, to some extent, understand its culture.

To explore the gamut of typographic challenges, we will visit four scripts: Devanagari, Khmer, Arabic, and Chinese. In terms of visual impression and structure, each of these scripts differs greatly from the others.

2 Devanagari

We begin with Devanagari — the most commonly used script in contemporary India — that is used primarily for the Hindi and Marathi languages. Devanagari has a long history of development that began with the Sanskrit language centuries ago. The structure of Devanagari writing is based on the syllable which, in its simplest form, can be composed of a consonant with either an inherent or explicit vowel mark. As an example, let us consider the letter *ka*, which when written as क, includes the implied vowel *a*. If we want to write *kī*, we need to explicitly add the *ī* vowel ◌ी to *ka* क, resulting in की. Presented at a larger size, we can clearly show how the two components of the kī syllable overlap on the top:

क + ◌ी ⟶ की

Now, if we want to write *ku* instead of *kī*, the *u* vowel is placed below *ka* as follows:

क + ◌ु → कु

Marks other than vowels can also appear above and below letters; for instance, if we want to indicate that the vowel in *ka* sounds nasal, we would add a dot above (natively, *anusvara*) as in:

कं

Since Latin type was usually composed on a single horizontal line with gaps between letters, Devanagari presented quite a challenge because of its need for overlap between adjacent characters, in addition to upper and lower marks.

3 Khmer

Next, we consider Khmer, or Cambodian, writing. As a remote descendant of the Brahmi script, the structure of Khmer is also broadly based on the syllable, including the presence of an implied or explicit vowel; however, the phonetic nature of the Khmer language has also resulted in many additional features that go beyond the ancestral Brahmi model. The orthographic norms of Khmer script represent the range of simple to complex syllables by allowing its components to stack both horizontally and vertically. Like Devanagari, the simplest syllable consists of a consonant with an implied or explicit vowel, but because of the tonal nature of Khmer, marks indicating tone are sometimes also required. Khmer has the same needs as Devanagari for upper and lower marks, which result in more height and depth for syllable clusters. While Devanagari allows multi-consonant syllables, it also allows them to stack horizontally, if need be. On the other hand, Khmer requires the components of a multi-consonant syllable cluster to stack vertically.

Following is an example of a relatively simple syllable cluster consisting of a single consonant *no* followed by a vowel mark *i* above (u1793 + u17B7):

ន + ◌ិ ⟶ និ

As an example where additional depth is required, the following syllable cluster consists of a single consonantal letter followed by a subscript consonant that sits to its right while partially overlapping it below [*pha* + subscript *sa*]. The use of a subscript form indicates that the preceding consonant is not followed by a vowel (u1795 + u17D2 + u179F):

ផ ◌្ ស ⟶ ផ្ស

The character u17D2 indicates that the following consonantal symbol should appear in its subscript form.

There are also deeper cases where a vowel symbol *ie* surrounds and subtends a consonant *ko* followed by a subscript consonant *lo* (u1783 + u17D2 + u179B + u17C0). Note that the right part of *ie* is stretched to embrace the stacked consonants (*ko*+ subscript *lo* + *ie*):

ឃ ល ◌ៀ ⟶ ឃ្លៀ

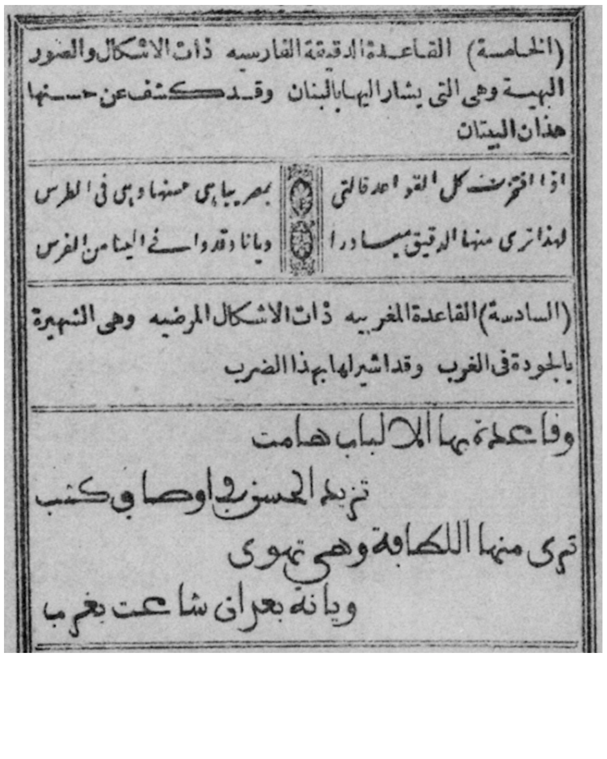
The above sequence could also carry an additional mark above it. Unlike Latin letters, Khmer characters have a predominantly vertical aspect ratio when arranged into syllable clusters. One can imagine the numerous challenges posed by Khmer script for those who wanted to implement it in metal type. How does one stack so many vertical elements along the baseline? If Latin and Khmer letters are put side by side, what sort of size ratio should be maintained between the two? Does one resort to using the largest leading needed throughout one document, or does one increase it only when required?

In digital fonts equipped with OpenType technology — or some other shaping software — the built-in shaping logic allows the user to enter text at the character level only, while watching as syllable clusters compose magically into their correct form. By moving from metal to digital type, we also leave the obstacles of metal behind while gaining new flexibility and fluidity of form.

4 Arabic

The cursive form of Arabic script is the one trait that differentiates it from most others. Arabic writing never developed typographic "block" letters, but has remained cursive from its inception to this day.

Spread over many centuries and a broad geographic terrain, many different styles of Arabic script have developed. Under Ottoman rule, when it came time to adapt one of the Arabic styles to type, the Naskh style prevailed. In due time, many other styles were implemented, but Naskh has remained the reference style for running text. Because of its rich variety of contextually variant forms, Naskh style could only be typeset by a compositor well versed in the intricacies of the style. Showing a variety of styles available for manual setting, the following image is taken from an 1873 catalog of the Amiriyah Press in Cairo:

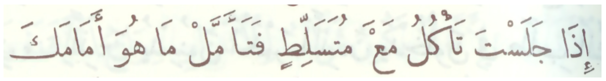

(الخامسة) القاعدة الدقيقة الفارسيه ذات الاشكال والصور
البهيه وهى التى يشار اليها بالبنان وقد كشف عن حسنها
هذان البيتان

اذا افترست كل القواعد فالتى بمصر يباهى حسنها وهى فى الطرس
لهذا ترى منها الدقيق مبادرا وبانا وقدوا فى اليناء من الفرس

(السادسة) القاعدة المغربيه ذات الاشكال المرضيه وهى الشهيرة
بالجودة فى الغرب وقد اشير اليها بهذا الضرب

وقاعدة بها الالباب هامت
تزيد الحسن فى اوصاف كتب
ترى منها اللطافة وهى تهوى
وبانه بعران شاعت بغرب

Each of the above styles required that the compositor thoroughly master its shaping rules.


إِذَا جَلَسْتَ تَأْكُلُ مَعَ مُتَسَلِّطٍ فَتَأَمَّلْ مَا هُوَ أَمَامَكَ

Most Arabic text is set without vowel marks, so setting fully vocalized text (with optional marks above and below the letters) as in the above sample was (and still is) laborious, and therefore avoided because of the extra expense.

With the advent of mechanized typesetting in the early to mid-twentieth century, styles had to be simplified to fit the new size constraints of magazines and keyboards. This trend to simplify persisted, reaching its peak with the arrival of phototypesetters where the limited number of character slots was dictated by Latin fonts.

In the present day, one can say with good confidence that Arabic script stands to benefit greatly from digital fonts equipped with OpenType technology in several ways. First of all, the oversimplification of the past century can be discarded because OpenType technology is capable of emulating the various calligraphic styles of Arabic script. Since the global adoption of the Unicode Standard in the 1990s, Arabic text is stored as a sequence of alphabetic characters that are independent of style. The embedded shaping logic of each font reflects its style, while text input remains invariant. By setting a string of text in a particular font, one can expect it to take the shape mandated by the rules of that style. For instance, in the following sample, each line consists of the same text set in three distinct fonts. Note that the second and third lines show words that are set on an incline with respect to the baseline — a calligraphic feature ably rendered in OpenType.

لمّا كان الاعتراف بالكرامة المتأصلة في جميع أعضاء الأسرة البشرية وبحقوقهم المتساوية الثابتة

لما كان الاعتراف بالكرامة المتأصلة في جميع أعضاء الأسرة البشرية وبحقوقهم المتساوية الثابتة

لما كان الاعتراف بالكرامة المتأصلة في جميع أعضاء الأسرة البشرية وبحقوقهم المتساوية الثابتة

5 Chinese

We now take a look at the fourth script, Chinese, which is nowadays shared mainly by the Chinese and Japanese languages, albeit with some stylistic differences. Unlike the three previously examined scripts, Chinese characters require no additional marks nor any contextual shaping. Once designed, a character can simply be typeset as is. The primary obstacle for Chinese script lies in the thousands of characters needed to set everyday text. Chinese writing has a

…Inherent dignity and…inalienable rights of…the human family is the foundation of freedom

…Inherent dignity and…inalienable rights of…the human family is the foundation of freedom

Figure 1: Typesetting in a standard Latin font vs. Zapfino.

long history of exacting calligraphic styles that require great effort to master. The following sample shows Song, one of the most common styles.

鉴于对人类家庭所有成员的固有尊
严及其平等的和不移的权利的承认,

Despite the graphic complexity of its characters, Chinese script is typically classified among the "simple" scripts because the typesetting of Chinese text has no complexities in terms of its character shapes.

To write Japanese, Chinese characters (*kanji*) are intermixed with two phonetic syllabaries called *hiragana* and *katakana* to show grammatical affixes, as well as foreign words and names. In typical Japanese text, about half the characters consist of kana (*hiragana* or *katakana*). The following text is taken from the first line of the Universal Declaration of Human Rights. The more fluid and curvier character are *hiragana*.

人類社会のすべての構成員の固有の
尊厳と平等で譲ることのできない権

In the twentieth century, ingenious methods and typesetting equipment were devised to handle the large character sets needed for Chinese and Japanese in their home lands. However, the keyboards and magazines of the mechanical typesetters of Linotype and Monotype were designed for limited character sets. In addition, the cost of developing an adequate set of Chinese characters was high enough to stop any Western companies from entering the field until the early era of digital typesetters in the late 1980s, when Monotype entered the marketplace.

Once digital methods prevailed in the Chinese and Japanese markets, more local enterprises were able to enter the typographic field and thrive. Digital text entry now depends on sophisticated software that exploits the systematic structure of Chinese character in terms of its radicals, strokes, or phonetics to simplify the job for the user.

6 Degrees of complexity

Now that we have taken a quick glimpse at several scripts and their traits, we need to examine what inferences we can draw for the world of scripts.

Scripts vary greatly in their level of complexity. A typical Latin typeface is quite simple because of its relatively small character set and the near absence of shaping rules, while a calligraphic font such as Zapfino depends on a set of complex rules; see fig. 1

This example clarifies how complexity can often be based in a particular style, and is not necessarily fundamental to the structure of the script itself. Some scripts are relatively complex because of their intrinsic shaping rules, while their input requirements are simple due to a small character set. The line composition of Chinese text is relatively simple because all characters fit within a square and require no additional marks. On the other hand, the labor needed to create all the characters is disproportionately high, and input methods are complex by necessity because of the large character set. An Arabic font whose style requires at most four shapes for each letter is simple in comparison to one whose style requires a rich set of contextual variants. Complexity in scripts is a matter of gradation. There is no need to simplistically put all scripts into two heaps, simple or complex.

In this brief overview, we have seen some salient features of each of the following scripts: Devanagari, Khmer, Arabic, and Chinese. Although there are no visual similarities among these scripts, they do share a common history: each of them came into the twentieth century with a typographic tradition of varying durations and strengths. While this is also true of other scripts, there are many others that came into the twenty-first century with either a typographic false start or none at all. There are many language communities who had their own traditional script but whose populations were too small to initiate and sustain a typographic practice. Other communities had established a traditional practice of metal typesetting during the nineteenth or twentieth centuries, but could not transition to newer technology such as phototypesetting as metal technology became obsolete.

Let us now roll back the calendar by about four decades to recount how we got to the current state of typography. As digital technology was maturing in the 1970s and 1980s, a silent software revolution was starting to overtake typography. Donald Knuth's METAFONT and TEX started making digital typography directly accessible to academics. This transition enabled a wave of academic publications that had earlier been throttled by ever-rising costs of professional typesetters. In the mid-1980s, the wave of WYSIWYG word-processing began at Xerox, but was later made available to the masses through the Apple Macintosh. The concurrent publishing of the PostScript language, and its use as a standard font format, wrested digital typography from the hands of proprietary typesetting equipment and brought it into the public arena. High-resolution printing — well, 300 dpi at the time — suddenly appeared in big and small offices alike, dragging along with it the word *font* into everyday language.

The first stage of the typographic revolution enabled the definition of character shapes through software instead of hardware. As affordable character-drawing software became available in the early 1990s, more typographic aficionados were drawn to the field as new practitioners. The new software made it possible to produce consistent character sets, properly formatted as fonts destined to create high-resolution imprints on paper. Because of the notable difference in resolution, what appeared on the computer screen was only an approximation of the final image produced by a laser printer on paper. In the end, the paper image was the more important one. After all, until that time, typography's aim was to leave an imprint on the relatively permanent medium of paper.

By making letter forms in software instead of metal, the wall of separation between Latin and non-Latin characters collapsed. After all, outlines defined in the form of polynomials have no physical barriers like molds and grooves, and are even free to overlap and intertwine. The domain of digital character shapes proved itself to be adaptable to any script.

As impressive as the first stage of digital typography was, it was only a first step forward. Even though it allowed us to produce typographic images without resorting to metal, it still had significant deficiencies which were not immediately apparent to all. Once stored on a computer, one could use a text document to produce numerous impressions of it over time. In a sense, we can consider the document a digital equivalent of the typographic galley for metal type, but unlike the galley, the early digital documents were somewhat unreliable. In the early 1990s, typical text documents used inherently ambiguous character codes; for instance, the code 192 (decimal) represented *À* in a Latin-1 code set, while it meant *ῶ* in a Greek set. With the right software and fonts in place, such a document could produce a faithful image of the intended content, but it was not a dependable, unambiguous archive of the original text.

Around that time, a group of technologists involved with multilingual computing were discussing how to achieve the until-then elusive goal of an unambiguous, global character set. They wanted a digital environment where code 192 would represent only one character, and every character of every language would be assigned a unique code. Gone would be the world of ambiguous character codes, and all scripts, Latin and non-Latin, would be on equal footing. This group of visionaries laid down the foundation for the Unicode Standard, which eventually came to be recognized as the one global character set by all the major makers of systems and software.

Once Unicode was accepted by Apple and Microsoft, the slow transition away from ambiguous codes began. Before and during this transition, the transfer of text documents between different systems was laborious. Any characters outside the 7-bit ASCII range could be garbled in transit unless they had been deliberately filtered. For instance, code `0xE8` in Mac Roman represented *Ë*, while in Windows Code Page 1252 it stood for *è*. As the use of personal computers continued to penetrate every aspect of life, it became clear that such a contradictory situation was untenable in the long run. Change was on the horizon, but it was slow in coming.

In the mid-1990s, as the presence of the Unicode Standard made itself felt across the globe, it met resistance in many different regions because it was perceived as foreign-born. To be accepted everywhere, advocates and practitioners of the Standard had to demonstrate its efficacy and suitability before many different national and linguistic communities. As this defense of the Standard was mounted, it soon became apparent that an aspect of Unicode had turned out to be an unintended obstacle for some non-Latin scripts.

"Characters, not glyphs" — one of Unicode's primary design principles — was aimed at keeping plain Unicode-encoded text free of the entanglements of style. Characters represent abstract units such as letters of the alphabet, while glyphs depict the particular style and size of the character, as dictated by a specific font. Since Unicode represents only the characters of a string, the final shape and appearance of the display glyphs are relegated to the "text rendering" process.

In the mid-1990s, competing systems had different "rendering" processes that typically supported only a few scripts. The rendering component of systems turned out to be a bottleneck for "complex" scripts such as Devanagari and Arabic; though their texts could be readily encoded in Unicode, it was difficult to achieve good typographic results. In that period, Apple introduced a sophisticated typographic system called "Apple GX" which was capable of rendering many scripts and styles, but it ran only on Apple hardware. Soon thereafter, Microsoft countered with a competing system called "TrueType Open".

Every practitioner in the publishing community was faced with a difficult choice to make. It wasn't until 1996 that Adobe and Microsoft first agreed on OpenType, a font specification that integrated both PostScript Type 1 and TrueType font formats, in addition to a unified system supporting advanced typographic features. Finally, there was hope of producing cross-platform fonts in the near term, even for non-Latin fonts. As could be expected, this potential was achieved gradually over the following years.

Today, we can rightly claim that a large number of scripts can be correctly rendered and that the level of typographic sophistication is continually rising. Certainly, scripts such as Devanagari, Arabic, and Khmer can be correctly rendered on multiple digital platforms without any worry about portability. The use of Unicode for numerous scripts, and the languages they support, has been universally accepted. Each of the characters *Ȁ*, *ϖ̄*, *Ë*, and *ȅ* is represented by the same unique code in any document, on any platform.

At this auspicious juncture, one must ask if there is more to do. Have we reached a stage where all the scripts in the world are on an even footing? In the Internet era, what does it take for a script to become "digitally enabled"? To prepare text documents in a particular script, all its characters must first be listed in the Unicode Standard, each of them assigned a unique code. Then, to make such documents humanly readable, there must be at least one functional font that renders the text in its commonly recognizable form. Once these two requirements are met, documents in a given script — and all the languages it supports — can be created, read, distributed, and archived.

Even though Version 13.0 of Unicode includes 154 scripts, more than 100 scripts are known to be under consideration for future inclusion. It might come as a surprise to some that so many scripts remain unencoded. Some scripts might no longer be in use and require considerable research to produce a definitive character set, while others might belong to a minority population that does not have the resources to prepare the documents needed to propose their inclusion in the Standard. Since it was established in 2002, the Script Encoding Initiative (SEI) has been preparing and presenting such proposals for scores of less privileged scripts (see `linguistics.berkeley.edu/sei/scripts-encoded.html`). Over the years, SEI has been instrumental in bringing at least 70 scripts into Unicode. Without such sponsorship, many scripts have little chance to enter the digital era.

About 10 years ago, Google launched the Noto Project (`google.com/get/noto`) to build OpenType fonts for the scripts in Unicode, and to make the fonts freely available to all under an open source license. Noto fonts were required not only to support the character set for each script, but also to make use of advanced typographic features that render the script in its expected native appearance. So far, at least 120 scripts are supported by Noto fonts. For scripts that were already widely covered, the Noto fonts simply broadened the variety of styles available, while for other scripts, they opened the door to the digital domain.

Not all scripts are on even footing yet — a great many have just begun their "digital lives". As always, improvements will need to be made, and additional scripts will have to be encoded. All the same, I can say with confidence that this is a fruitful and auspicious moment for the whole spectrum of global scripts, from red to violet.

◇ Kamal Mansour
`Kamal.Mansour (at) {monotype,me} dot com`

Production notes

Karl Berry

This article was typeset with XƎLaTeX, using the `polyglossia` package and its commands `\texthindi` and `\textkhmer` for the Devanagari and Khmer examples. The input text was UTF-8.

We used the Noto Serif Devanagari and Noto Serif Khmer fonts (regular weight). It was not feasible or desirable to install them as system fonts, so we specified them to `polyglossia` as filenames:

```
\setotherlanguages{hindi,khmer}
\newfontfamily\devanagarifont[Script=Devanagari]
  {NotoSerifDevanagari-Regular.ttf}
\newfontfamily\khmerfont[Script=Khmer]
  {NotoSerifKhmer-Regular.ttf}
```

With the addition of `,Renderer=HarfBuzz` in the `\newfontfamily` calls, the results obtained with LuaLaTeX were identical. ◇