Typesetting Manual and Specimen

Gramatika

Typesetting Manual and Specimen

Typesetting Manual and Specimen

Gramatika
Typesetting Manual and Specimen

Written and Designed by
Roman Gornitsky

Gramatika was initially developed as a typeface for the visual identity of V–A–C Foundation (Moscow/Venice) in close collaboration with Experimental Jetset and the V–A–C Foundation design team.

The author would like to thank Marieke, Erwin, Danny, Lyosha and all the others involved in the development process.

This is an online version of the manual.
You can order a printed version at
shop.temporarystate.net/gramatika-manual

Gramatika is available
for purchase at
typefaces.temporarystate.net

Published by The Temporary State
2020

ISBN
978-3-00-067256-9

Introduction

Total Design

Font Size: 75 mm

Gramatika Regular

Introduction

Kees Broos
Design: Total Design

Breda, 1983

The
Dutchman
walking around
with
an open
eye,

Introduction

will sooner or later (and whether he realizes it or not) encounter one of the many visual products of the Amsterdam design office Total Design (TD).

On his way to the Amro Bank, which has started the renewal of its interiors according to TD design. Wandering past the blue-green-red banners fluttering along the facade of the local Property Development Foundation and past the bright display windows of Randstad employment agency. A red van of Rapid Post passes him, followed by trucks of Boskalis and Calpam. In the cultural centre, posters of Stedelijk Museum Amsterdam and Museum Boymans Rotterdam are displayed. At the post office his eyes dwell upon the new red reception desk.

He has just seen a whole series of letters, symbols, colours and spatial designs, which all originate from Total Design. Even if he was to stay at home the whole day, chances are that a mailman delivers a letter carrying a stamp designed by a TD associate. The phone book, which he consults, is typographically arranged by Total Design, just as the text and illustrations of his Spectrum Encyclopedia.

While visiting museums and exhibitions, he certainly ran into the work of Total Design more than once. While escaping the wet Dutch winter to warmer regions, TD routing systems at Schiphol Airport make sure that he reaches his plane quickly and easily.

Introduction

Industry Standard

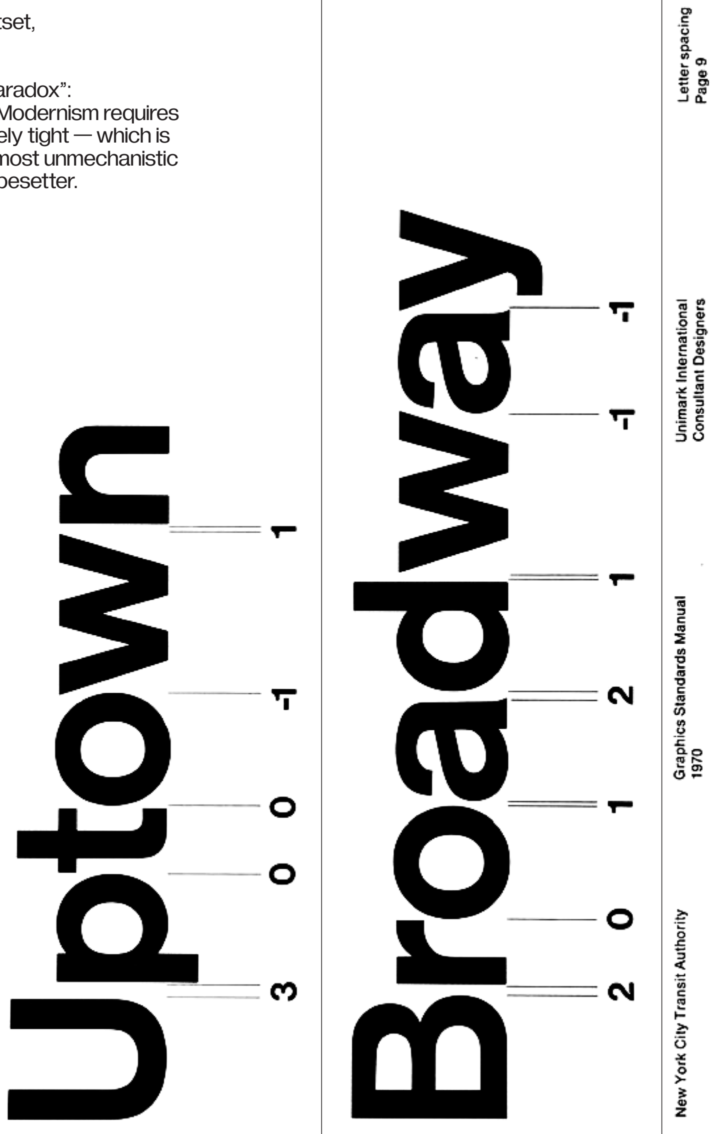
Throughout the 1960s and 70s the Dutch design studio Total Design produced a truly immense amount of work. As described before, their design was omnipresent and their client list seemingly endless. It will not be an understatement to say that the image of the entire country at that time was shaped by just one studio. Such scale comes with an obvious downtrade: only the key items and concepts could be worked through by the studio's world-renown partners (Wim Crouwel, Benno Wissing, Friso Kramer, Ben Bos, etc.), the rest of the work had to be delegated to an army of common employees. At its peak in the end of the sixties, the office of Total Design employed 45 "permanent associates" — not including the countless interns and freelancers. When the production reaches such scale, it's not possible anymore to strive for artistic excellence in each produced piece, but rather the issue of quality control becomes the main focus. An employee is not supposed to create a masterpiece, an employee is supposed to not make a mistake. To ensure the quality of graphic production, numerous manuals and guidelines were conceived — the industrial production of graphics required the industrial standardization:

> *At the start of our activities, Friso, Wim and I had realized very quickly, that in dealing with large projects, a number of things had to be standardized so that the arrangement of information could be more easily programmed, and more time would become available for handling intrinsic problems. If there were to be variations within the final products, we preferably searched for variations within a modular system, so that mutual relationships, interconnections, clustering and industrial production wouldn't demand follow-up care. This principle was applicable to architecture, industrial design and graphic design. That's the history of the birth of the grid! A cuckoo in the nest?*
>
> Benno Wissing,
> 1983

Helvetica was an essential element at the core of this system of "total" unity of style. Celebrated throughout the world by the International Style designers as a tool to orchestrate the entire design process, Helvetica does surprisingly little to streamline the actual typesetting process. On the contrary, since the Modernist typography requires it to be set as tightly as possible, it usually requires meticulous and tedious arrangement of every letter by hand:

> *To suggest that the way we use Helvetica is an easy way out typographically is ridiculous. We spend an enormous amount of time spacing, kerning, lining and positioning type. The fact that we use only a small variety of typefaces demands a certain discipline, a skillful precision, a focus on the finer details. It's certainly not that a-different-typeface-for-every-occasion attitude. Now, that would be an easy way out.*
>
> Experimental Jetset,
> 2003

Lets call it the "Helvetica Paradox":
The mechanistic image of Modernism requires Helvetica to be set extremely tight — which is only possible through the most unmechanistic manual labor of a skillful typesetter.



Measurement
of Type

From
Aa
to Яя
Zz

Font Size: 75 mm **Gramatika Regular** Line Height: 60 mm

Measurement of Type

Font Size

Working in InDesign, you can set the measurement units in your document to points[1], picas[2], inches, inches decimal, millimeters, centimeters, ciceros[3], agates[4], pixels[5], or even introduce your own measurement units. There is, however, one area to which these settings won't apply — type size will always be measured in points regardless of your settings. Why do fonts require their own measurement system and why can't their size be expressed in understandable metric units? I guess, it is some sort of a conspiracy: an obscure measurement system is designed to conceal the fact that the font size is, essentially, a lie. Because if you look at, for example, Times New Roman set in 30 mm size, you won't find a single measurement, that relates to the declared 30 mm size in any way (fig. 1). All the measurements seem to be completely random and do not relate to any measurements in any other typeface. In fact, it seems to be the type designer's duty to reinvent all the metrics whenever a new typeface is created. Often even the fonts by the same publisher won't feature a single matching value (fig. 2).

All of this makes modern digital fonts completely incompatible with each other: you can't put two different typefaces on the same grid. It also renders the creation of any universal typographic system impossible: you can't build a rational grid out of some unmeasurable ethereal substance, whose size is decided solely by what "feels right" (and therefore changes with every new task). For any true modernist, such environment is a constant lingering pain. The standard has to be introduced!

Technical Note

Every character of a digital typeface is described as a series of Bézier curves on a coordinate plane: each node and each curve handle is represented by a pair of numbers (see p. 9). To convert these abstract coordinates into the real world values, a parameter called UPM (Units per Em) is used. Usually it equals 1000, which means that if you, for example, set the type in 10 mm, 1000 units of a coordinate plane become 10 mm of physical measurements. So the letter "R" from the example on the opposite page, if set in 10 mm size, will have the height of exactly 6.5 mm.

Times New Roman has 2048 UPM and the caps height of 1356, therefore the letter "R" in 10mm Times New Roman has the height of roughly 6.6211 mm. There is a good reason to assume that all fonts have 1000 UPM and convert their inner measurements accordingly (unless you want to challenge your calculus skills a lot).

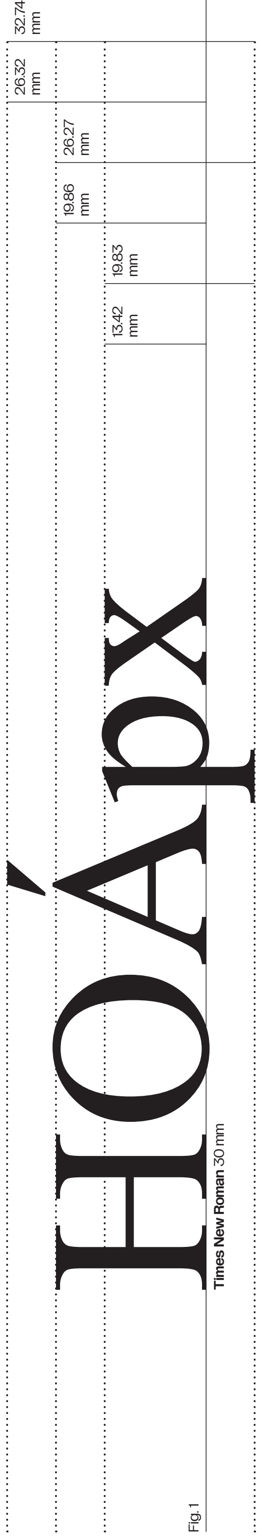


Fig. 1

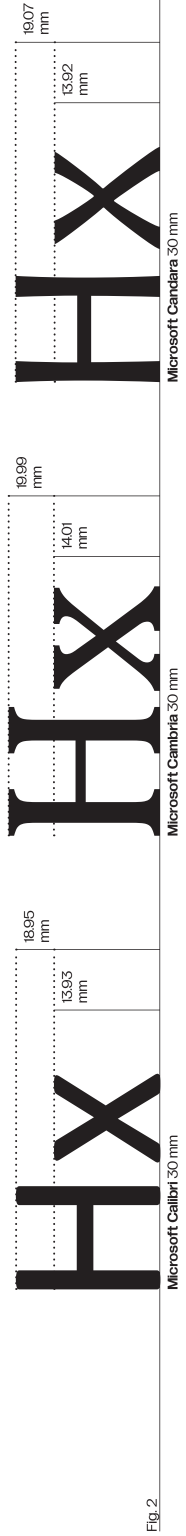


Fig. 2

[1] You can choose, whether you want to use PostScript points (roughly 0.3527 mm), "Traditional" points (roughly 0.3515 mm; "Traditional" stands for "American") or if your idea of a point is something completely different — you can assign the point to be anything between 0.315 and 0.42 mm.

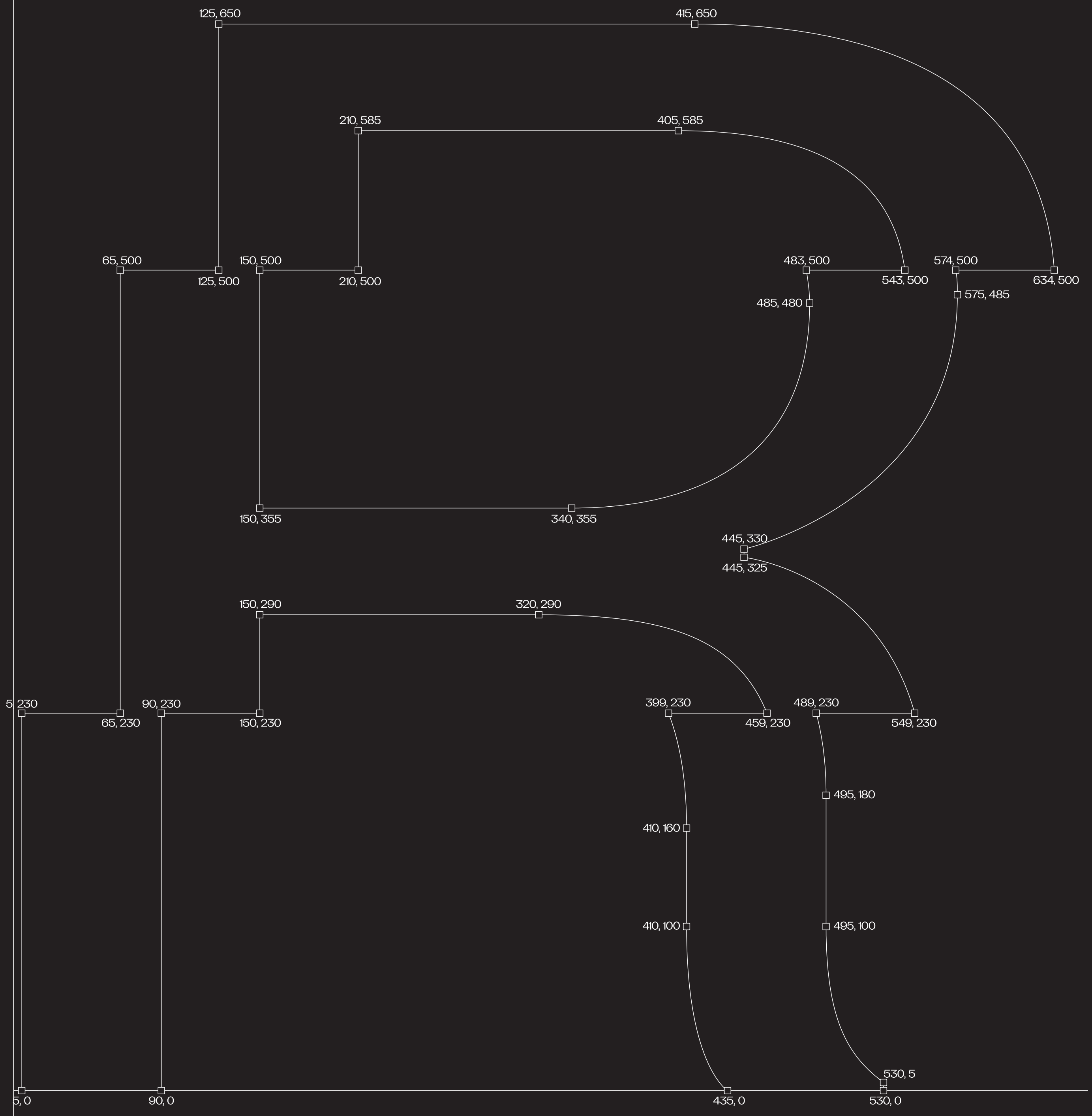
[2] One pica equals 12 points.

[3] One cicero also equals 12 points. Didot points, not PostScript points. Therefore, in InDesign 1 cicero equals 12.788 points.

[4] A bizzare measure of 1/14th of an inch. I'm not sure if anyone has ever made use of it in the entire history of InDesign.

[5] One pixel equals one PostScript point. A rather liberal assumption, to say the least.

Measurement of Type



Measurement of Type

Line Height

One might assume that all these seemingly random metrics are, in fact, carefully calculated values designed to ensure that the typeface looks good when the text is set on a line height matching the font size. Unfortunately, that cannot be further from truth. If you look at the specimen PDFs provided by the foundries themselves, you won't find sample text set in 12 point type on a 12 point line height. You will find 12 on 14 points (Colophon), 12 on 15.5 pt (type.today), 12 on 16 pt (Brownfox) and so on. The default value of "Auto" leading in InDesign is 120%, and it sets the 12 pt type on a 14.4 pt line. Online you can find style guides recommending anything between 120% and 180%. From my own typographic practice I can remember using values such as 9.95 on 12 pt, and I'm sure I'm not the only one here to get to the second decimal. But if I ever did set type in 12 on 12 pt, that must have been either a pure accident or a rebellious act of radical lazyness.

Modern digital typefaces are not designed to look good without leading[1]. The existence of leading in the world of metal typesetting was understandable, but in the world of digital typography it's hard to explain why we set the type more often on 140% rather than just 100% line height. Maybe in type designer's mind the type is set in one endless line of text, and there are simply no lines above and below. Or maybe he just doesn't care. Because why bother — it is, after all, the typographer's job to mold all this mess into comprehensible paragraphs. So, since we regularly encounter typefaces with something like 2048 UPM, 916 x-height and 1356 caps-height, set in sizes like 9.75 on 12 pt, it can be safely concluded: there is no logic in the modern typographic measurement system whatsoever.

Technical Note

It also has to be mentioned that different languages require drastically different line heights. For example, Russian has almost no ascenders or decenders, so it is usually set rather tight, while Slovak has a lot of ascenders, decenders and diacritics, so often it's not even possible to set it on a 100% line height — the diacritics will overlap the descenders even in lowercase (fig. 3).

Slovak Alphabet
Times New Roman,
100% line height

aáäbcčdď
dzdžeéfg
hchiíjklĺľmnň
oóôpqrŕsš
tťuúvwxyýzž

Russian Alphabet
Times New Roman,
100% line height

абвгдеёжз
ийклмнопрстуф
хцчшщъ
ыьэюя

Fig. 3

[1] This fact is considered so obvious, that a lot of foundries don't even mention the line height in their specimens. Only measuring the test sample with a ruler, one can find out, how the type is actually set.

Measurement of Type

210 mm

Uni

180 mm

While it is obvious that different typefaces cannot
have all of their metrics the same —
some have ascenders longer than the others
and the uppercase-to-lowercase ratio can also vary greatly —

150 mm

ver

120 mm

it is entirely possible to introduce
at least one measurement that will be the same across
all the fonts. As the most of our texts are written
in lowercase,

90 mm

sal x-

60 mm

it would make sense to pick the x-height as such a universal value,
fixing it at 500 units — exactly half of the font size.

30 mm

height

0 mm

Measurement of Type

Font Dimensions of Gramatika

Modernist sans-serifs, especially in display typography, are often set with negative leading, if any. These tight typographic compositions have to be arranged carefully, since one needs to constantly watch out for the ascenders not to overlap the descenders of the line above (and whenever this happens, the tribal dance with kerning, tracking and word spacing ensues). In order to mitigate this issue, the ascenders and descenders of Gramatika were made as short as possible, which also allowed to make the ascender and decender levels more pronounced and negate the demand for different line heights in different languages. While in most other typefaces "t" is usually shorter than "h", and "ř" aligns with neither (fig. 4), Gramatika keeps them all level — a joy for the Modernist eye. Needless to say, the uppercase height matches the ascender height. And finally, the uppercase diacritics are packed tightly to ensure that there will be no overlaps and the 100% line height can finally be set as the default (see pp. 13–14).

This 100% line height, while safe, might be not tight enough for some purposes. For such cases the line height of 80% of the font size is recommended (or, if you need to calculate the font size based on the line height, FontSize = 1.25 × LineHeight). With these settings the ascenders and descenders won't overlap, but watch out for the uppercase diacritics (pp. 15–16).

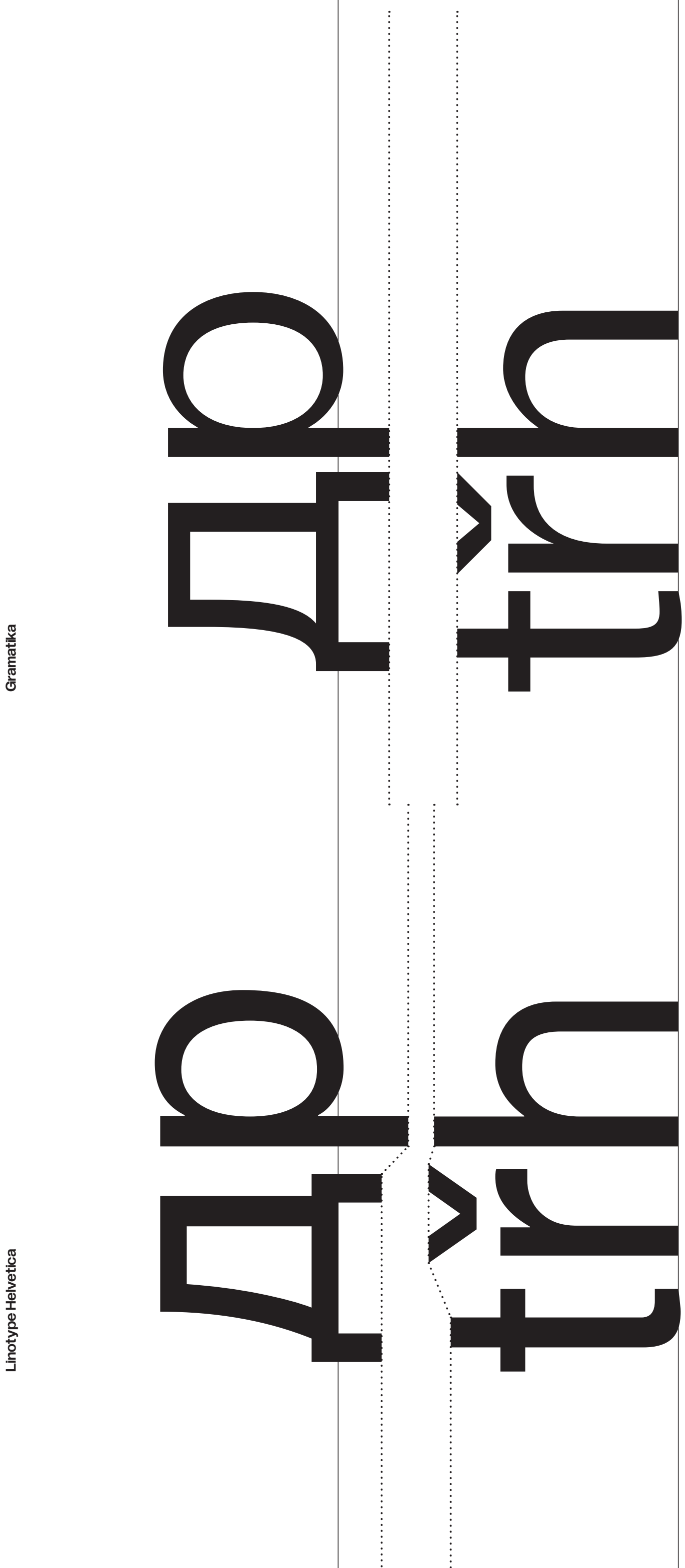
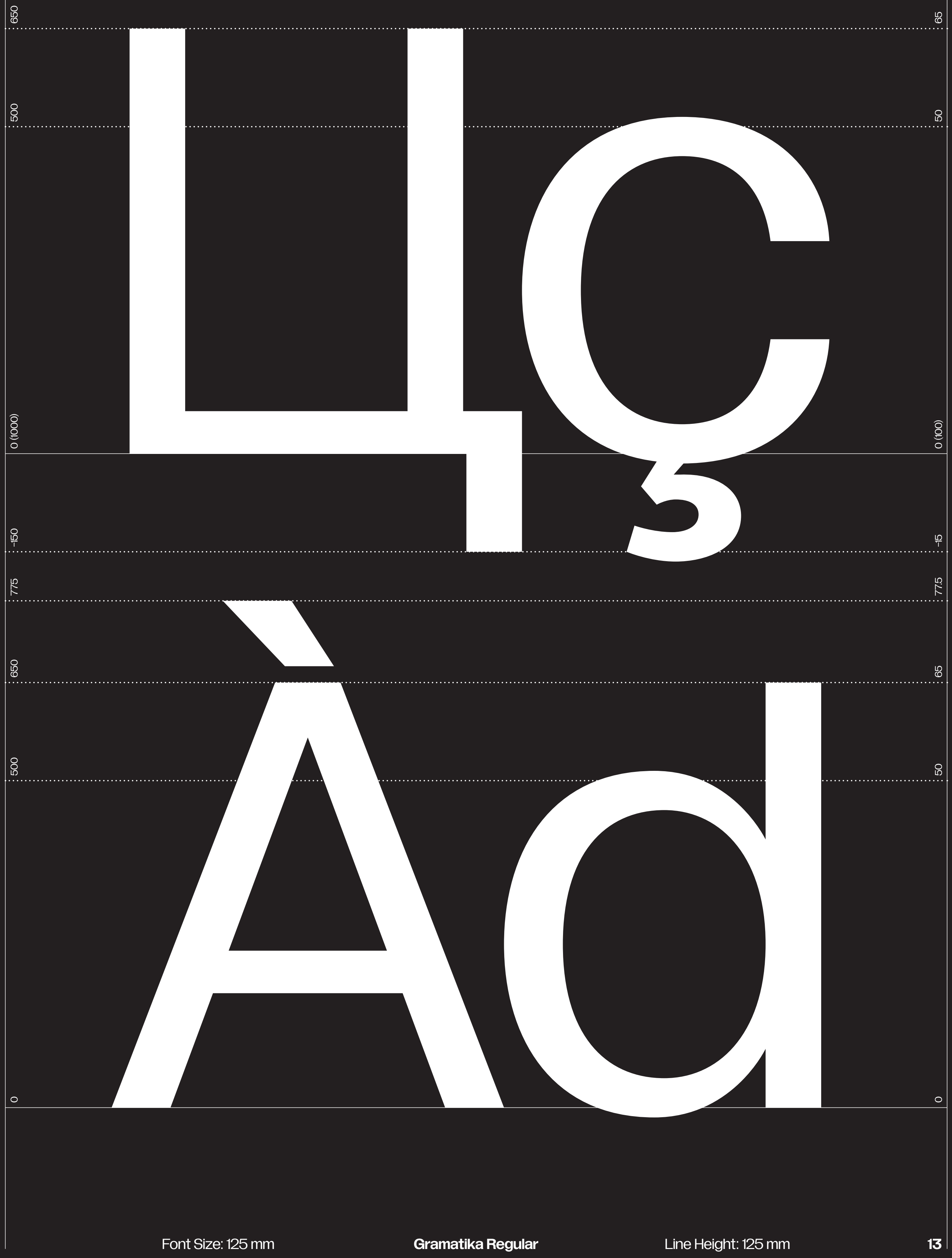


Fig. 4

Measurement of Type

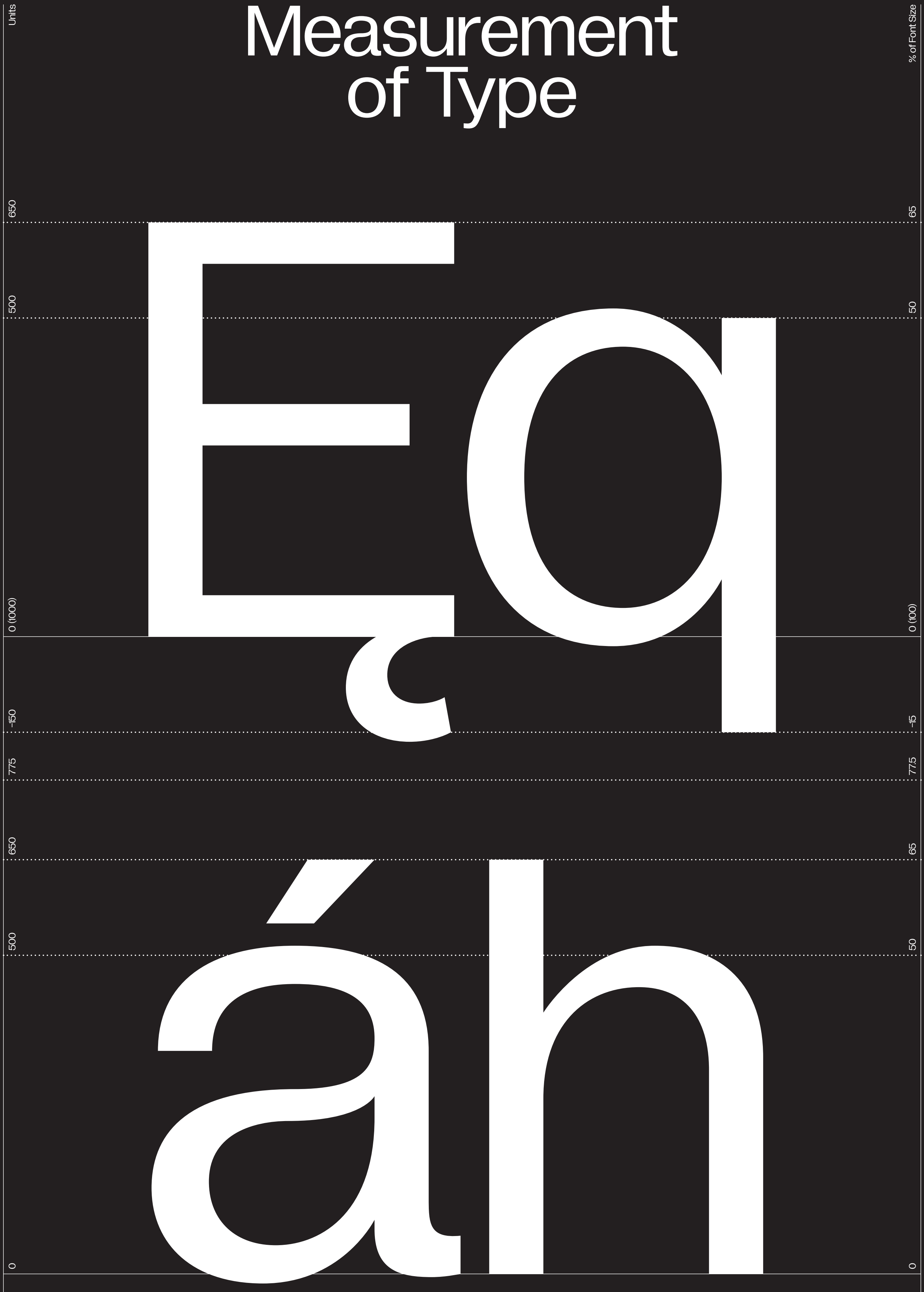


Font Size: 125 mm

Gramatika Regular

Line Height: 125 mm

Measurement of Type

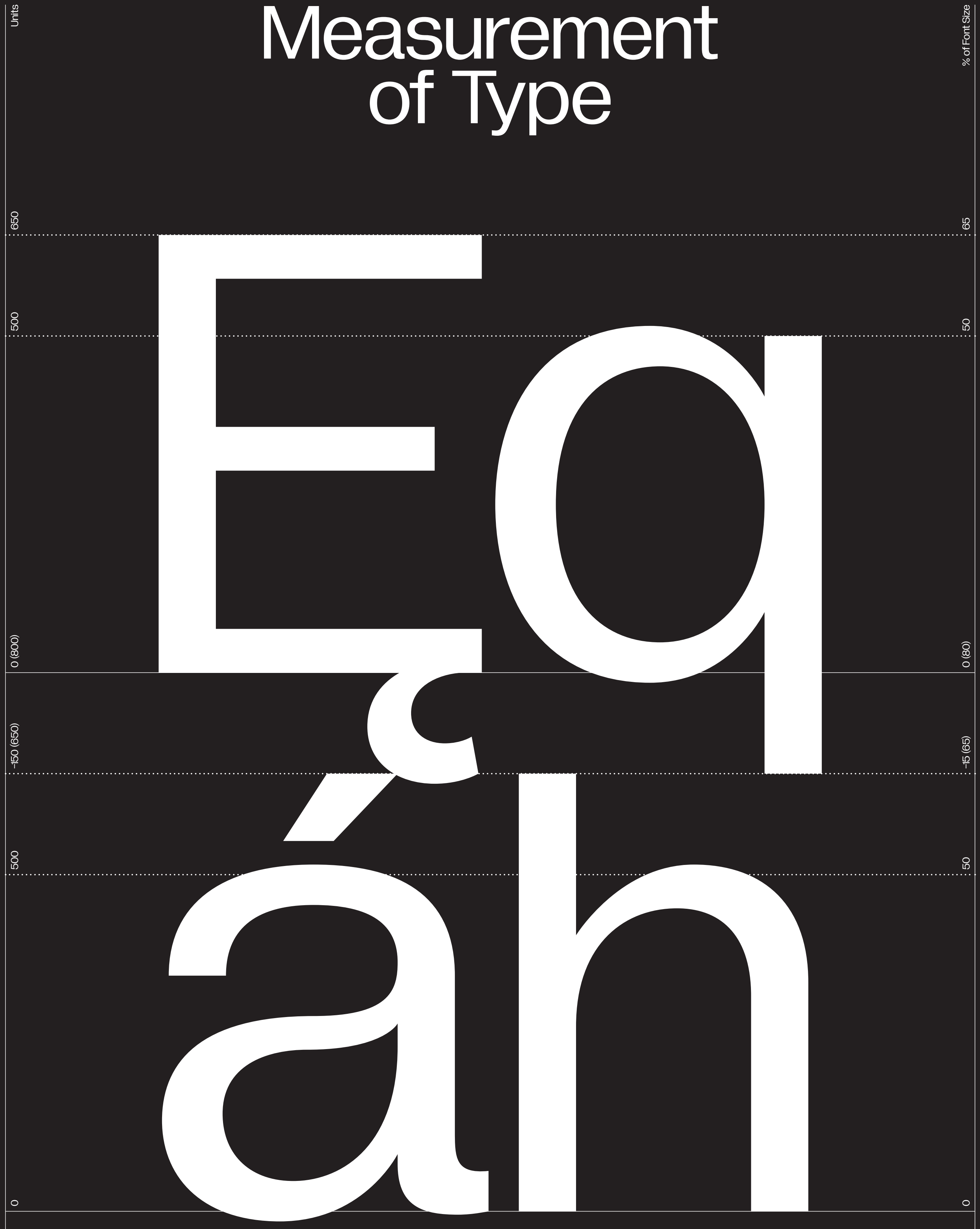


Font Size: 125 mm

Gramatika Regular

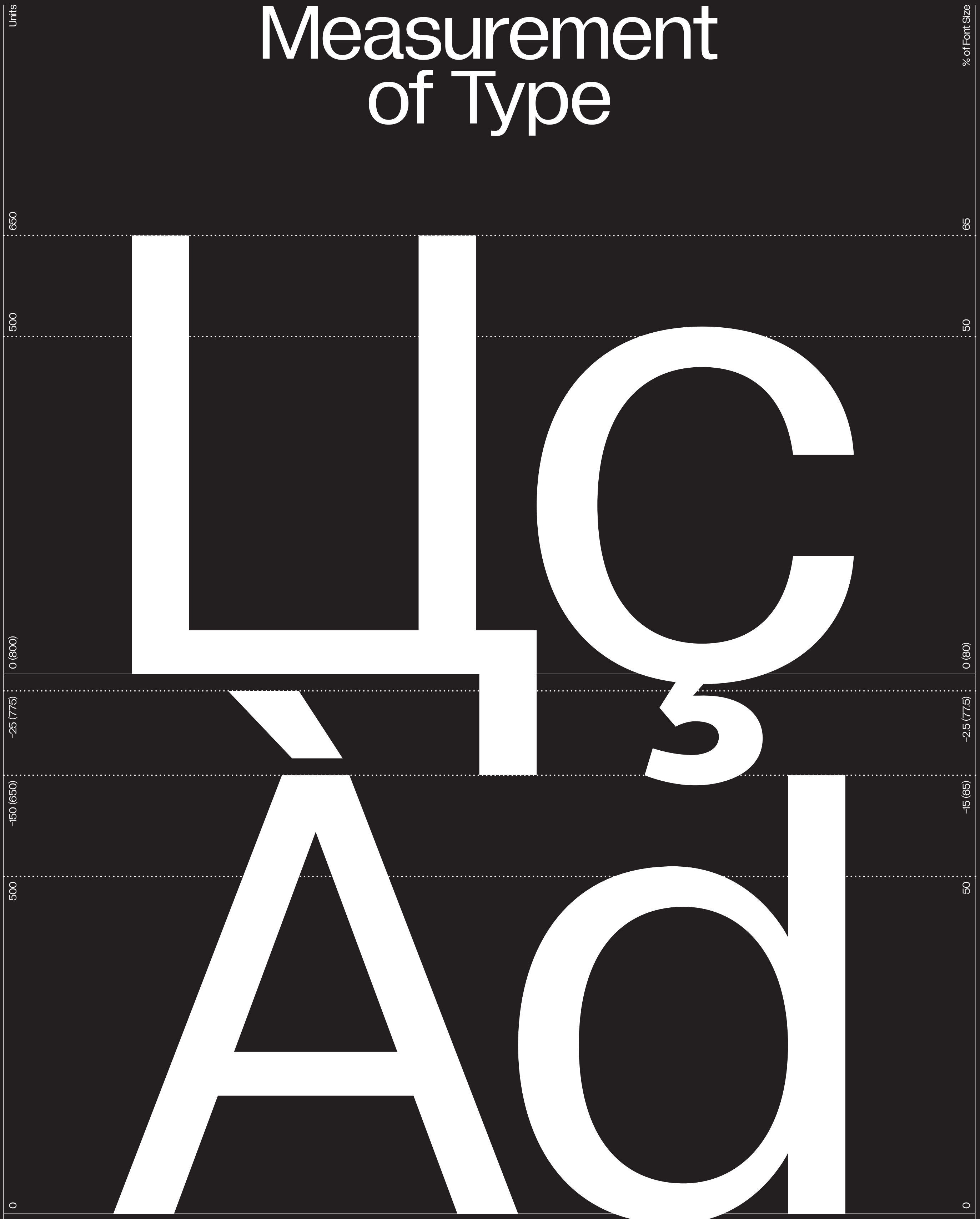
Line Height: 125 mm

Measurement of Type



Font Size: 125 mm **Gramatika Regular** Line Height: 100 mm

Measurement of Type



Font Size: 125 mm

Gramatika Regular

Line Height: 100 mm

Family Composition

Bold
Bold
Bold
Italic

Font Size: 75 mm

Gramatika Regular

Line Height: 60 mm

Family Composition

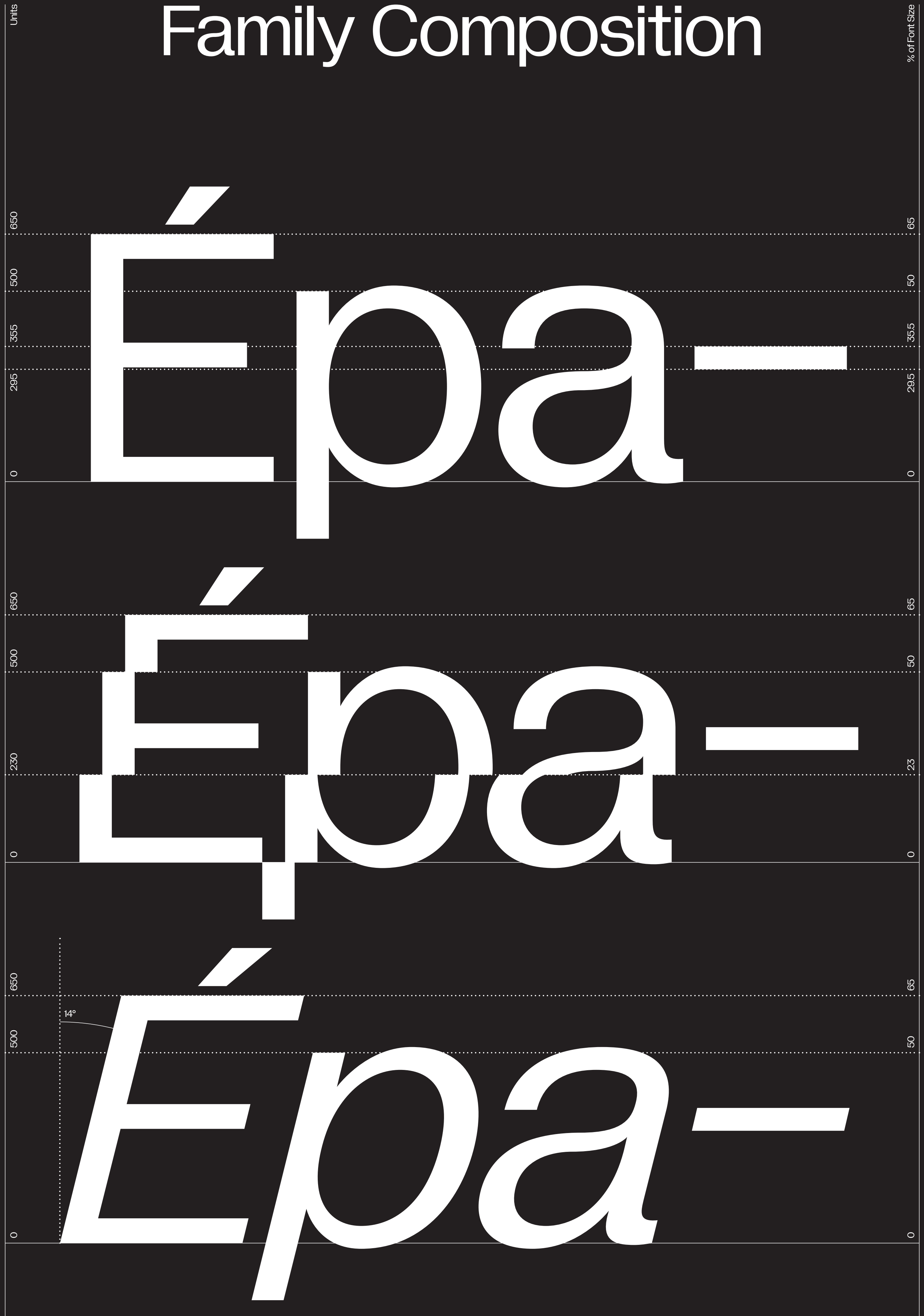
Two Italics

The type family currently consists of six styles: regular, two variations of regular “italic” styles, bold and two bold “italics”.

The first of the two “italic” styles is the usual for Modernist sans–serifs Slanted — it doesn’t have any special “true italic” lettershapes, but rather features the slanted variations of the upright characters. The second type of “italic” presents an alternative approach to constructing the incline — the upright characters are cut horizontally into pieces and shifted along the x–axis (therefore the name, “Shifted”), which creates an optical illusion of a slant. All “italic” styles share the metrics with their upright counterparts, so the text set, for example, in Bold will occupy exactly the same amount of space as Bold Shifted or Bold Slanted.

Finally, it should be mentioned, that all the symbol characters (pp. 36–47) are not affected by neither slant, nor shift, nor boldness.

Family Composition

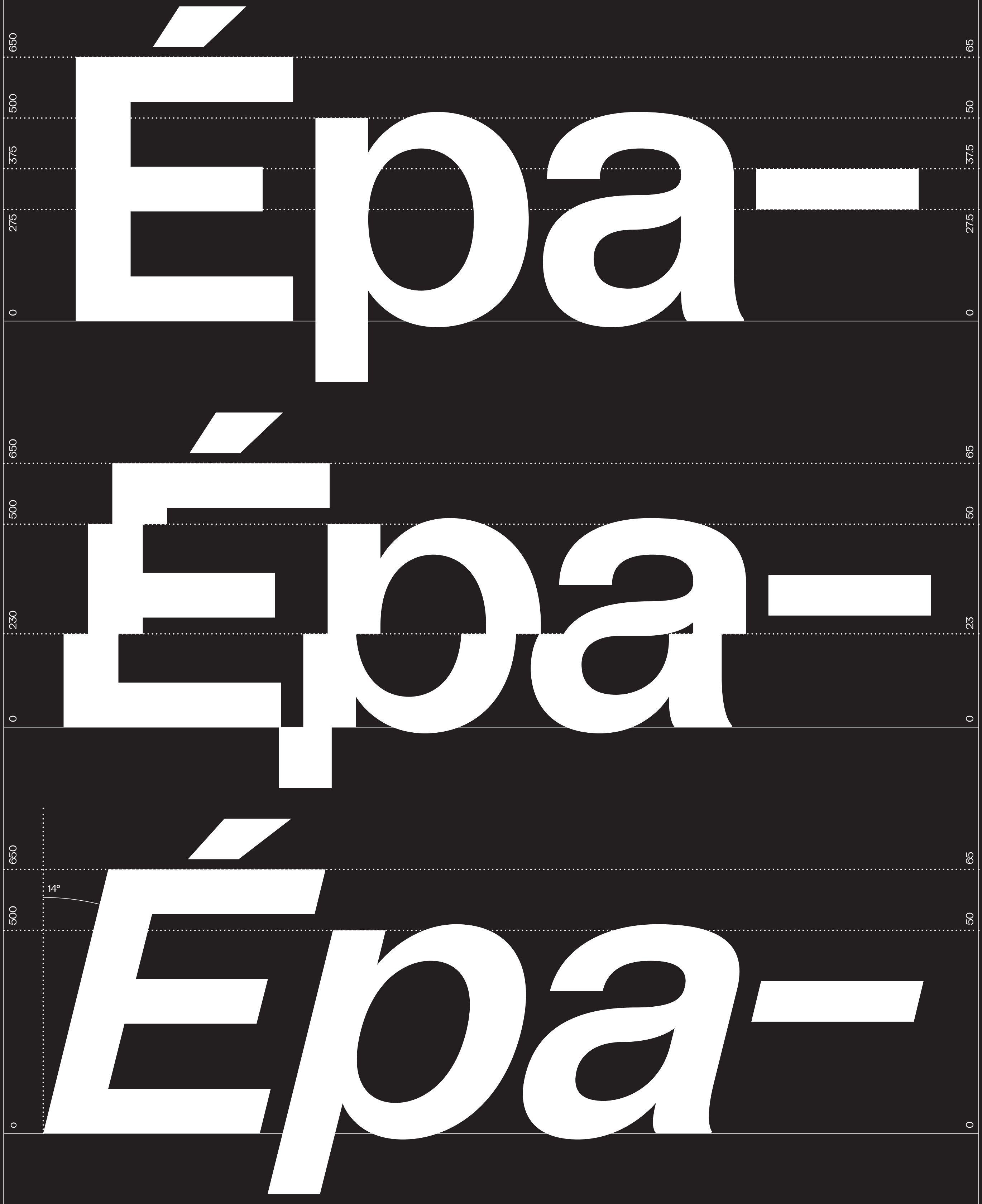


Gramatika Regular, Shifted and Slanted

Font Size: 75 mm

Line Height: 75 mm

Family Composition



Font Size: 75 mm

**Gramatika Bold,
Bold Shifted
and Bold Slanted**

Line Height: 75 mm

Language Support

Aa

Àà Áá

Ââ Ãã

Āā Ąą

Font Size: 75 mm **Gramatika Regular** Line Height: 60 mm

Language Support

Supported Languages in Cyrillic Script

A
Azerbaijani

B
Bashkir
Belarussian
Bosnian
Bulgarian
Buryat

Ch
Chuvash

D
Dungan

K
Kalmyk
Kazakh
Komi-Permyak
Kyrgyz

M
Macedonian
Mari
Moldovan
Mongolian

R
Russian
S
Serbian

T
Tajik
Tatar
Turkmen

U
Ukranian
Uzbek

Supported Languages in Latin Script

A
Afrikaans
Albanian

B
Basque

C
Catalan
Croatian
Czech

D
Danish
Dutch

E
English
Estonian

F
Filipino
Finnish
French

G
German

H
Hungarian
I
Icelandic
Indonesian
Irish
Italian

L
Latvian
Lithuanian

N
Norwegian

P
Polish
Portuguese

R
Romanian

S
Slovak
Slovenian
Spanish
Swedish

T
Turkish

U+04D0
Cyrillic Capital Letter A With Breve
Ӑ
U+04D1
Cyrillic Small Letter A With Breve
ӑ
U+0102
Latin Capital Letter A With Breve
Ă
U+0103
Latin Small Letter A With Breve
ă

Hyphens, Dashes and Features

elements
of
Style

Font Size: 75 mm

Gramatika Regular

Line Height: 60 mm

Hyphens, Dashes and Features

Transhuman Typography

It's impossible to make a design mistake on a typewriter — only a grammar mistake or a typo. Computers, though, are a whole different story. Even a "basic" program such as Microsoft Word dumps the entire typographic toolbox on the head of a (usually) completely unprepared user. Suddenly, he has to make a myriad of design decisions: choose the typeface, choose the font size, adjust the line height… Full justification of flush left? Indents? Drop caps? Tables, footnotes, pagination, images with captions, pie charts with legends… and for the creative ones: WordArt. Compared to the typewriter's humble repertoir of ALL CAPS, sperrsatz and underline (with only line breaks and tabulations for the spatial organization of the page), the possibilities of Microsoft Word are truly a maze. And while the professional typists of analog times would often undergo extensive training, the users of Microsoft Word are usually left alone in the typographic wilderness to figure things out themselves — no wonder that a well-typed document is an object of great rarity nowadays.

But blame the user! We all know: the user is an idiot and can't be trusted. Therefore an army of scripts watches over him carefully while he types. The beginnings of sentences are capitalized. Hyphens, when surrounded by spaces, are converted into emdashes. Anything that looks like a weblink is made blue and clickable. Lists receive proper indents and additional spacing according to the finest style manuals and proper quotation marks appear all by themselves. The user can't be trusted, but the automation will save us from bad typography. We just need more scripts, and the machine watching over the monkey with a keyboard will spit out Tschichold-worthy layouts, despite the monkey's best efforts to reign chaos.

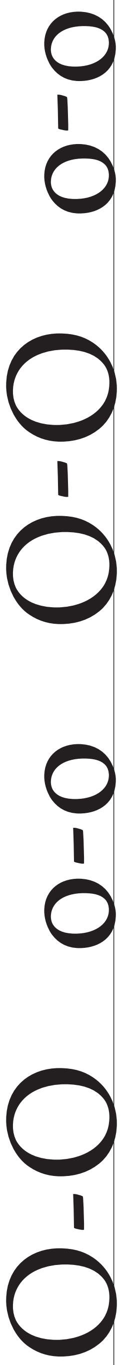
There are a lot of scripts for "typography enhancement" deployed on all levels already. Even inside the font files themselves there are OpenType "features", which carry out the duties for which human compositors were employed in the old days. They kern. They convert "fi" letter combination into a nice ligature. They add extra spacing when the type is set in all caps. They can spice up your layout with swashes and contextual alternates. They give you access to small caps, "true" ordinals and proper fractions. Unfortunately though, there is no real unity about how most of these features should be written and implemented. So every new typeface offers a new set of rules and demands your previous rulebook to be thrown away.

Vertical Alignment

One of the issues that OpenType features are trying to address, is the issue of vertical alignment. How high or low should the hyphen be positioned? The "uppercase" hyphen would be too high for the lowercase, and the "lowercase" hyphen usually appears dangling around the knee level of uppercase (fig. 5). To solve this problem, often an entire separate set of "uppercase" punctuation is drawn, which substitutes the default "lowercase" punctuation whenever OpenType "case" feature is activated. Which punctuation marks should have uppercase variations and which ones should not is not an easy question to answer. Hyphen, dashes, brackets and guillemets are the usual first candidates to receive an uppercase alternate. But should "·" have two options or is it always "uppercase"? What about "~"? "+"? And if we decide to make two versions of "+", does that mean two versions of the entire math symbols set: ÷, ×, =, ≈, ≠, ≡, etc.? Striving for the neatly organized perfection, you quickly find yourself buried under an insurmountable amount of characters to draw[1].

It would be interesting to find out what percentage of people are actually using the vertical alignment features. My guess: a small minority. To overcome this reluctance from the general public, it is tempting to force the substitutes to appear automatically, a sort of AutoCorrect: if there is a hyphen — yes → and it is preceded by an uppercase character — yes → and is followed by an uppercase character — yes → then use an uppercase hyphen instead of a regular one. But then you quickly realize that the combination "uppercase+whitespace+emdash+whitespace+uppercase" should also be scripted. And, probably, "uppercase+question mark+emdash+uppercase" as well… The list of substitutes grows, the errors are inevitable. Here's a radical proposal: trust the skill of the typographer. Skilled typographers will write their own GREP rules to lift the punctuation whenever and how they want it, amateurs will always find a way to mess up despite the most elaborate OT features. For both the scripts inside the fonts are useless[2].

Thankfully, another advantage of Gramatika's extremely low caps height is that there is no need for a separate uppercase punctuation set anymore. All the punctuation is positioned around the vertical middle of the caps, which does make the hyphen appear a bit upskewed in the lowercase, but hopefully one can find it a charming feature, rather than a nuisance. Afterall, isn't such a mechanistic simplification only suitable for a modernist sans-serif?



Arno Pro
with the default hyphens (left)
and with activated "case" feature (right)

Fig. 5

[1] And then comes the kerning: should the capitals have no kerning with the lowercase-punctuation set? Obviously not, since we have T-shirts and V-shapes. What about the brackets in mixed case? Maybe the uppercase punctuation should be kerned with lowercase as well? The kerning table grows exponentially and soon enough you start to envy the Chinese.

[2] Furthermore, it is well possible that the abundance of scripts may lead to the extinction of typographic skills — if the machine picks the right quotation marks for you automatically, there is no need to know anything about the quotation marks anymore.

Hyphens, Dashes and Features

Units

% of Font Size

650 65

325 32.5

0 0

D–sh

650 65

325 32.5

0 0

D–SH

650 65

325 32.5

0 0

d–SH

Font Size: 75 mm

Gramatika Regular

Line Height: 75 mm

Hyphens, Dashes and Features

More on Hyphens

Overall it is not entirely clear, why the machine should follow the rules of centuries-old typesetting manuals? Why our modern digital fonts are designed to typeset 18th century books? Why all these manicules, fleurons and double daggers? And — an outrageous question — why so many dashes?

> *. . . a normal font of roman or italic type includes at least three. These are the hyphen and two sizes of long dash: the en dash — which is one en (half of an em, M/2) in width — and the em dash — which is one em (two ens) wide. Many fonts also include a subtraction sign, which may or may not be the same length and weight as the en dash. And some include a figure dash (equal to the width of a standard numeral), a three-quarter em dash, and a three-to-em dash, which is one third of an em (M/3) in length.*
>
> Robert Bringhurst,
> The Elements of Typographic Style,
> 1992

I'm sure that somewhere there is a thick book entirely dedicated to the proper use of dashes and hyphens. It is, afterall, one of the very first things a typography student learns about the "proper" typesetting: there's no greater sin than to put a hyphen where the dash belongs. Unfortunately, our keyboards have no room (or rather, no desire) to fit all these dashes[1], so if you happen to be on Windows without a numpad, salvation isn't easy to obtain.

Refined typographers can argue about the dashes for hours: "A hairspace should be used around an emdash, not a regular space!" — "No, a hairspace would not suffice, a thinspace instead!". "An emdash in Times New Roman is too long, an endash should be used instead." — "No! An emdash is the only way, the endash is no less than a grammar mistake!" Meanwhile, type designers are trying to offer their own solutions, making the issue more and more intricate: Fedra has some space around an emdash, so presumably you don't need additional spacing. Times New Roman, on the opposite, has the emdash that sticks outside the character box ever so slightly. And Arial has a zero-space emdash[2]. Every new typeface you buy demands you to rewrite your typing manual. Furthermore, a truly tasteful hyphen should also look differently from a dash:

> *Most hyphens currently offered are short, blunt, thick, and perfectly level, like refugees from a font of Helvetica. This has sometimes been the choice of the designer, sometimes not. The double hyphen designed by Hermann Zapf in 1953 for his typeface Aldus, as an example, was omitted when the face was commercially issued in 1954. Foundry Centaur, designed by Bruce Rogers, had a hyphen inclined at 48°, but Monotype replaced it with a level bar when the face was adapted for machine composition in 1929. And the original Linotype issue of W.A. Dwiggins's Electra had a subtly tapered hyphen inclined at 7° from the horizontal; later copies of the face have substituted a bland, anonymous form.*
>
> Robert Bringhurst,
> The Elements of Typographic Style,
> 1992

Even in Arial the hyphen is different from the dashes — it is bolder. And while the dashes can have zero or even negative spacing, the hyphen is always spaced. Therefore, you can't make an emdash out of a hyphen by, say, stretching it to 200%. And most certainly two hyphens don't make an emdash:

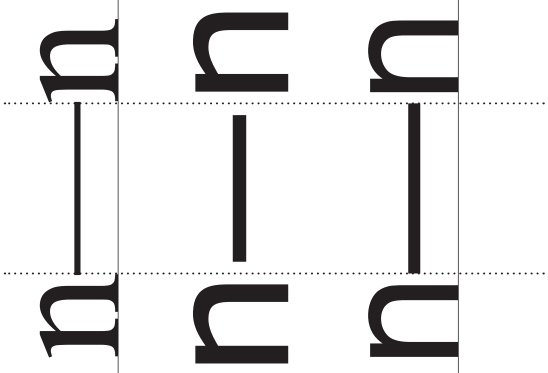
> *In typescript, a double hyphen (--) is often used for a long dash. Double hyphens in a typeset document are a sure sign that the type was set by a typist, not a typographer. A typographer will use an em dash, three-quarter em, or en dash, depending on context or personal style. The em dash is nineteenth-century standard, still prescribed in many editorial style books, but the em dash is too long for use with the best text faces. Like the oversized space between sentences, it belongs to the padded and corseted aesthetic of Victorian typography.*
>
> Robert Bringhurst
> The Elements of Typographic Style,
> 1992

But wait, why is that? In our age of automation, considering that "--" is already a widespread abbreviation for "—", why don't we just script this substitution the way f and i merge into a single ligature? In fact, in TEX this feature is implemented since 1980s:

> *for hyphen, type a hyphen (-);*
> *for an en-dash, type two hyphens (--);*
> *for an em-dash, type three hyphens (---);*
> *for a minus sign, type a hyphen in mathematics mode ($-$).*
>
> Donald Knuth,
> The TEXbook,
> 1984

Actually, you don't even need OT features or any other scripts to do that: you can just kern the "--" pair (if your hyphen is not tastefully inclined, that is). Or, how about that: a hyphen with no spacing. That will make two hyphens in a row merge, therefore it can be used as

-	a hyphen
—	an emdash
———	an ememememdash...



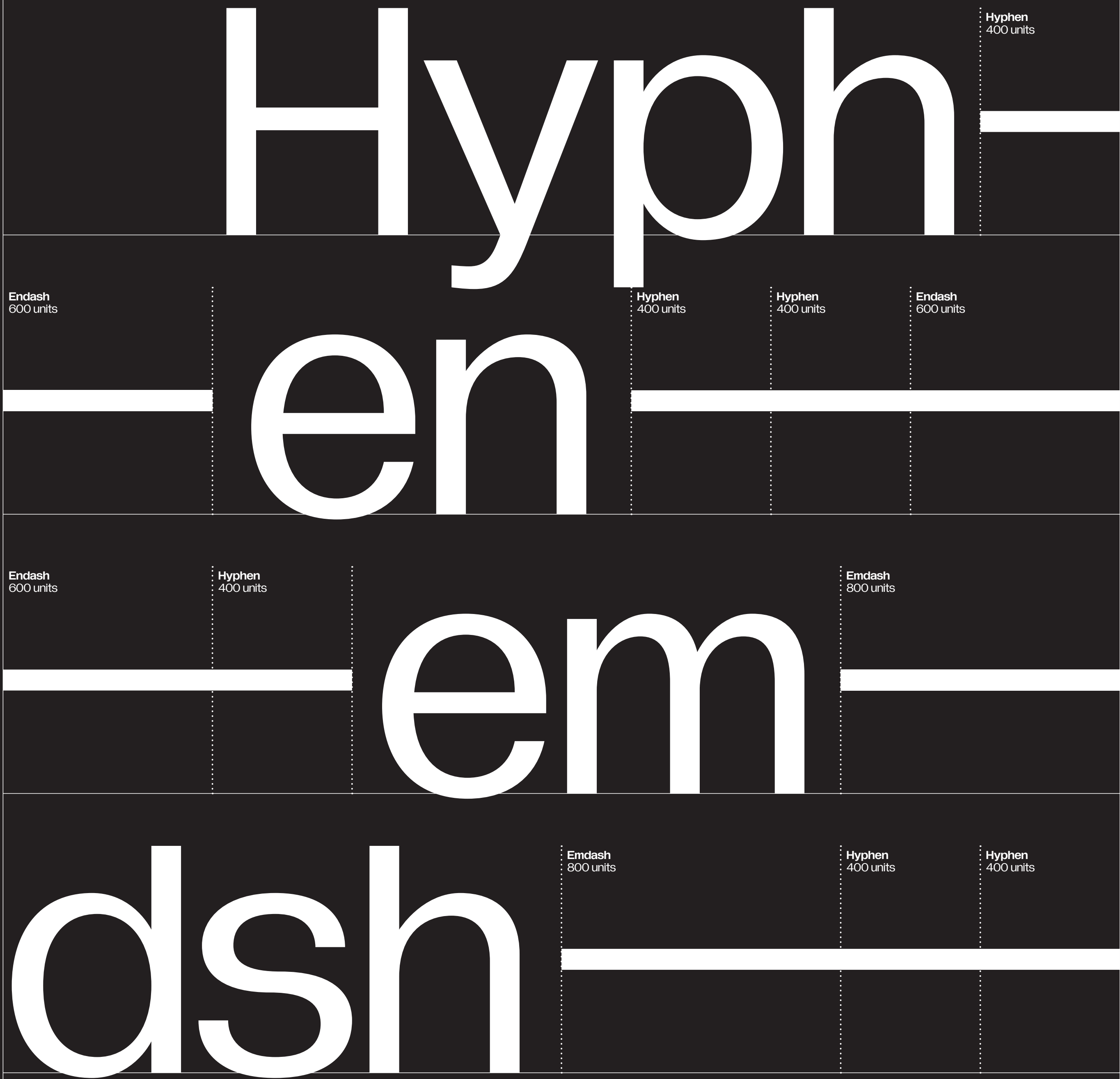
Times New Roman,
Fedra and Arial
22.5 mm

Fig. 6

[1] Actually, the full-sized keyboards have keys for both the hyphen and the minus, but both of them are mapped to the same character — the hyphen. In a similar manner, the multiplication key on the numpad is bound to an asterisk instead of a multiplication sign and the division button renders a slash. So, considering also the presence of almost entirely useless "Break" and "ScrollLock" buttons, the lack of space can hardly be justified as the main reason.

[2] A zero- or negative-space emdash has, arguably, another function: you can make a horizontal bar by typing several emdashes in a row.

Hyphens, Dashes and Features



Hyph‐

Hyphen
400 units

Endash
600 units

–en‐‐–

Hyphen
400 units

Hyphen
400 units

Endash
600 units

Endash
600 units

Hyphen
400 units

–‐em—

Emdash
800 units

dsh—‐‐

Emdash
800 units

Hyphen
400 units

Hyphen
400 units

Font Size: 56.25 mm Gramatika Regular Line Height: 45 mm

Hyphens, Dashes and Features

An (Almost) Modular Punctuation

The systematic widths of the three dashes (400→600→800 units wide) make it tempting to devise the widths of the rest of the punctuation set in a similarly modular way. It might eventually prove useful, for example, to have the space character 200 units wide, so that a hyphen is two spaces wide, endash is three spaces and emdash — four spaces. It might also be at some point convenient to have a period with the same width as the space. Comma, naturally, should have the same width as the period. And so should colon, semicolon, interpunct, etc.

Since an endash also serves as a minus, the rest of the math symbols should probably also be 600 units wide (+, ÷, ×, =, ≠, ≈, etc.). And since an asterisk is often used instead of a multiplication sign, maybe it should be 600 units wide as well. Such asterisk is too big, though, to build an asterism out of it. So maybe there should also be smaller sized asterisks, all sorts of them: * top, ⁎ bottom, ⁑ double and ⁂ asterism. That way they can be arranged and combined with each other into all sorts of constellations: ⁑⁑⁑⁑⁑⁑⁑ **⁎⁎**⁎⁎** ⁂⁂⁂*.

An underscore should, naturally, be the same width as the endash. Same should be the ‾ overscore, ⁀ tie and ‿ undertie, also all combinable with each other: __‿‿‿‾‾⁀⁀⁀.

But should the currency symbols be also made to fit this system? Is there any advantage in modular ampersand? Will modular guillemets serve any particular purpose? Probably, not. So in the end, Gramatika's modular punctuation system is rather inconsistent — some of the punctuation marks and symbols are systematic (p. 29), while others are not (fig. 7).

Superscripts, Subscripts, Ordinals, and Fractions

There are no special characters drawn for superscript or subscript, neither there are any specially designed ordinals or fractions. There is no need in them — if you carefully apply the correct type settings, the results will be just as good as any specially drawn characters (p. 30).

Underlines, Strikethroughs and Overlines

While the font files already contain all the information about the preferred underline position and thickness, most layout programs and browsers either read this data incorrectly or ignore it altogether. Furthermore, you might want to choose different underline or strikethrough settings depending on the context, diacritics or aesthethics (p. 31).

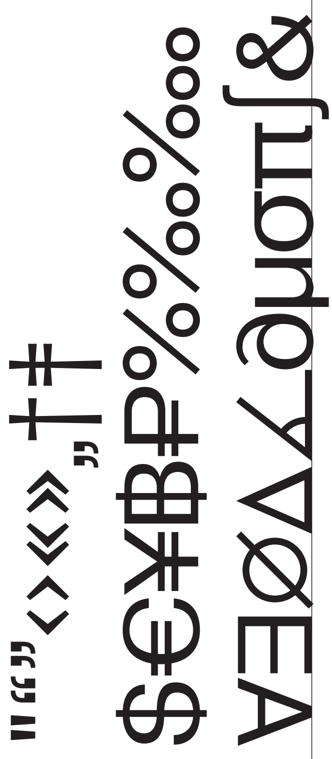


Fig. 7

Non–modular punctuation marks and symbols

Hyphens, Dashes and Features

200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 400 400 400 400 400

. , · , ‘ ’ ` ' : ; ¡ ! ! | ¦ ° – * ⁂

600 600 600 600 600 800 800 800

… • ¿ ? ? # — ⁂

600 600 600 600 600 600 600 600 600 600

‒ _ ‾ ‿ ⁀ ~ ␣ < > *

600 600 600 600 600 600 600 600 600 600

\ / + × ÷ ^ ± ≤ ≥ √

600 600 600 600 600 600 600 600 600 600

= ≠ ≈ ≡ ∵ ∴ ¤ ∑ § ¶

300 300 300 300 300 300 300 300 1000 1000 1000

( ) [ ] { } ⟨ ⟩ ∞ @ №

Hyphens, Dashes and Features

Énp— Ép2 —5

Superscript
scale 50%
baseline shift up by
0.4 × FontSize

Énp— Ép2 —5

Midscript
scale 50%
baseline shift up by
0.175 × FontSize

Énp— 2⁄3 —5

Fraction Top
scale 50%
baseline shift up by
0.325 × FontSize

Fraction Sign
scale 50%
baseline shift up by
0.175 × FontSize

Fraction Bottom
scale 50%
no baseline shift

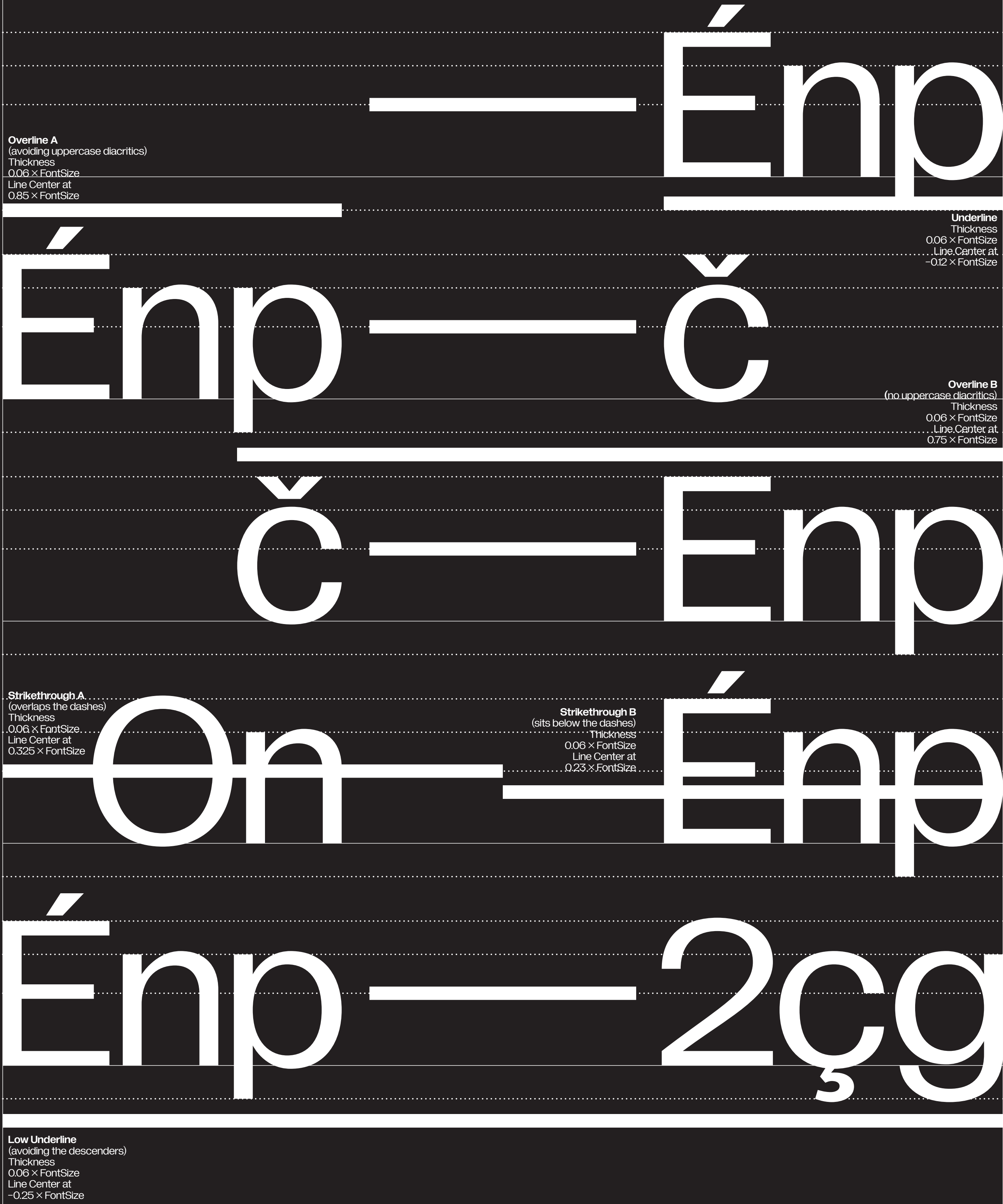
Énp— Ep2 —5

Subscript A
(no uppercase diacritics)
scale 50%
baseline shift down by
0.075 × FontSize

Énp— Ép2 —5

Subscript B
(in case of uppercase diacritics)
scale 50%
baseline shift down by
0.15 × FontSize

Hyphens, Dashes and Features



—Énp

Énp—č

č—Enp

On—Énp

Énp—2çg

Overline A
(avoiding uppercase diacritics)
Thickness
0.06 × FontSize
Line Center at
0.85 × FontSize

Underline
Thickness
0.06 × FontSize
Line Center at
−0.12 × FontSize

Overline B
(no uppercase diacritics)
Thickness
0.06 × FontSize
Line Center at
0.75 × FontSize

Strikethrough A
(overlaps the dashes)
Thickness
0.06 × FontSize
Line Center at
0.325 × FontSize

Strikethrough B
(sits below the dashes)
Thickness
0.06 × FontSize
Line Center at
0.23 × FontSize

Low Underline
(avoiding the descenders)
Thickness
0.06 × FontSize
Line Center at
−0.25 × FontSize

Hyphens, Dashes and Features

Figures

Considering the modular approach towards the punctuation, one might expect to find tabular figures in Gramatika. While it is true that the tabular figures would be the preferred option, their implementation comes into conflict with the overall aesthetics of the typeface. Tabular figures would unavoidably have irregular and gappy spacing, which contradicts the tight perfect fit of the rest of the typeface. In the end the choice is made in favor of aesthetics, so Gramatika may be not the most convenient font to typeset financial reports.

OpenType Features

There are only two OpenType features in Gramatika:

kern	kerns;
zero	converts the regular zero (0) into a dotted one (0).

Small Caps

The caps are already small enough.

Expansion of the Alphabet

From

to

Font Size: 75 mm

Gramatika Regular

Line Height: 60 mm

Expansion of the Alphabet

The Interface

A banal statement: in the digital age the range of symbols that we use in our daily life extends far beyond mere alphabet and punctuation. We can just as well “read”: a picture of a sheet of paper with a folded top right corner, a speech bubble with three blinking dots inside, all sorts of human silhouettes, hearts, stars, thumbs up and down, smiling and frowning faces, floppy disks, shopping carts, trash bins... Search, share, fullscreen, exit fullscreen, play, pause, shuffle, repeat, new window, close window, battery full, battery empty — all these concepts can be communicated without a single written word. Despite usually being used in typographic composition, these symbols are never included in the character sets of retail typefaces, so hundreds of icons have to be drawn and redrawn for every new website and for every new application.

Unsurprisingly so, as just a couple of decades ago our fonts were limited to a mere 256 characters (so, if you wanted to set the text in, for example, Czech and Spanish, you would have to buy two different fonts). There was simply no space for anything else except the basic alphabet and punctuation. The transition from 8-bit codepages to Unicode (with the adoption of OpenType format) was a great improvement, which finally allowed the fonts to expand beyond the needs of basic grammar. But this liberation came not without an issue itself, its biggest problem being its biggest advantage: as declared in the Unicode Character Encoding Stability Policy, “once a character is encoded, it will not be moved or removed.” Therefore U+2707 will always stand for “✇”, a symbol for the long abandoned tape drive. And we will always have the luxury of choice between a regular snowman (U+2603), a snowman without snow (U+26C4), and a black snowman (U+26C7). Thousands of new characters get added every year[1] to this endless field of aubergines, soccer balls and smiling faces — no type designer would be able to keep up with such a rapid expansion, even if he dedicates his entire life to making just one typeface. Unicode is too big for any typeface to cover[2], while the old 8-bit codepages are too small and no modern typeface is able to fit into Win-1252 anymore. Currently, there is no good way to describe a character set of a particular font. There is no easy way to say, for example: “This is a good typeface for navigation: it has icons for toilets and elevators, among with an extensive set of directional arrows.” We still have to either refer to the 8-bit codepages or dump the entire character set. In the first case anything outside the usual alphabet and punctuation spectrum gets omitted, in latter — pales into insignificance.

In fact, reference to the Unicode may not be a good way to describe a character set at all. First of all, modern OpenType fonts already have a lot of characters, that are not mapped to the Unicode — the majority of ligatures, swashes, contextual alternates, small caps, slashed zero, digit variations, etc. are not encoded. Second, and most important, from the designer’s perspective the choice of symbols in Unicode is rather questionable. There is a heart (U+2665), a heart outline (U+2661), a heart with an arrow (U+1F498), a heart with a ribbon (U+1F49D), a heart on an envelope (U+1F48C), a sparkling heart (U+1F496), a growing heart (U+1F497), a beating heart (U+1F493), two hearts (U+1F495), revolving hearts (U+1F49E), heart decoration (U+1F49F), heart exclamation (U+2763), broken heart (U+1F494), green heart (U+1F49A), yellow heart (U+1F49B), blue heart (U+1F499), orange heart (U+1F9E1)... but there is still no symbol for “share” icon?!

A lot of projects nowadays are trying to bridge this gap between typography and pictography. For example, The Noun Project offers a selection of over 3 million monochrome, reasonably well-drawn, and seemingly Helvetica-compatible icons: “From icons for Artificial Intelligence, to UI elements and Beyoncé, we’ve got you covered.” But the abundance is exactly the problem here: the spirit of Modernism demands finding universal solutions, not producing a myriad of bespoke fixes for any possible case. The thing that makes all these icon lists unattractively dull and seemingly endless (and, consequently, inconceivable for any type design project) is the total lack of abstraction. Icon design nowadays is most painfully pictorial, declaring that the only way to properly describe the act of adding an item to the virtual “shopping cart” is a picture of a hand, holding a box, putting that said box into a literal shopping cart with wheels. Anything less explanatory will confuse the user. Any true modernist would retort: less is more! In a well-designed context a simple downward arrow would suffice. Also, the progress always goes from a picture of a bull’s head towards the abstraction of a letter “A”: since the majority of users nowadays might not even know what a floppy disk is, 🖫 becomes nothing but a “save icon”, a symbol in itself. The resemblance to the real life object does not matter anymore and it is only a question of time now, before this symbol will transition into it’s abstract, non-pictorial shape (is it still too soon to switch to just something like ◳ already?).

Finally, there is beauty in ambiguity. Even such an inambiguous symbol as a magnifying lens can mean either “search” or “zoom in”, depending on the context. There doesn’t seem to be much confusion around it or a need to distinguish “the searching lens” from “the zooming lens”. So why can we not use more abstract, universal, multi-purpose symbols, then? Such context-dependent use is not only more efficient (covering more purposes with less characters, hopefully shrinking the character set to somewhat comprehensible size), but, I dare to say, is more poetic as well.

The composition of Gramatika’s character set, due to the lack of any acknowledged standard, relies solely on the intuition of its designer. It does not manifest any new standard, but invites the discussion about such standard: which new symbols do we really need and which ones can we abandon?

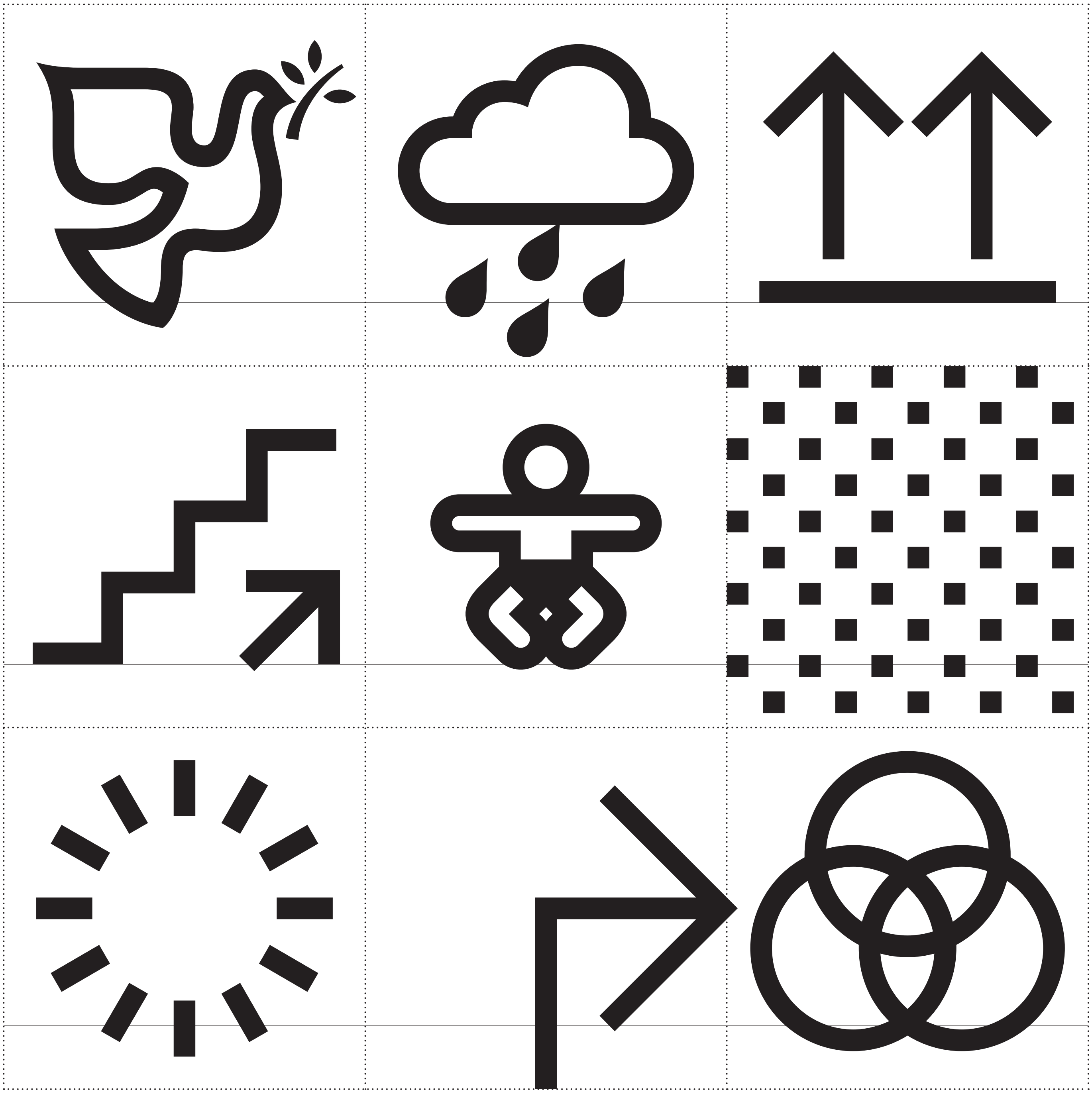
The symbols, icons and pictograms of Gramatika acknowledge the established conventions and avoid unnecessary reinventions (so yes, still a recognizable floppy disk for “save”, no radical simplifications), but try to depict any pictorial symbol in the most simple and abstract way possible (so, a magnifying lens is just a circle with a diagonal line, no fancy handle, no flare), while also introducing some ambiguous symbols, which don’t directly mean anything and therefore can be assigned to a wider spectrum of concepts (for example: ▚, ✳, ⚭, ◌).

All pictographic symbols have the width of 1000 units (so, the width equals the font size) and are designed for the 100% line height typesetting.

[1] Unicode 13.0, released on March 10, 2020, adds 5 930 new characters, for a total of 143 859 characters.

[2] Even such an immense type project as Arial (4 503 characters) covers only approximately 3% of the Unicode table.

Expansion of the Alphabet

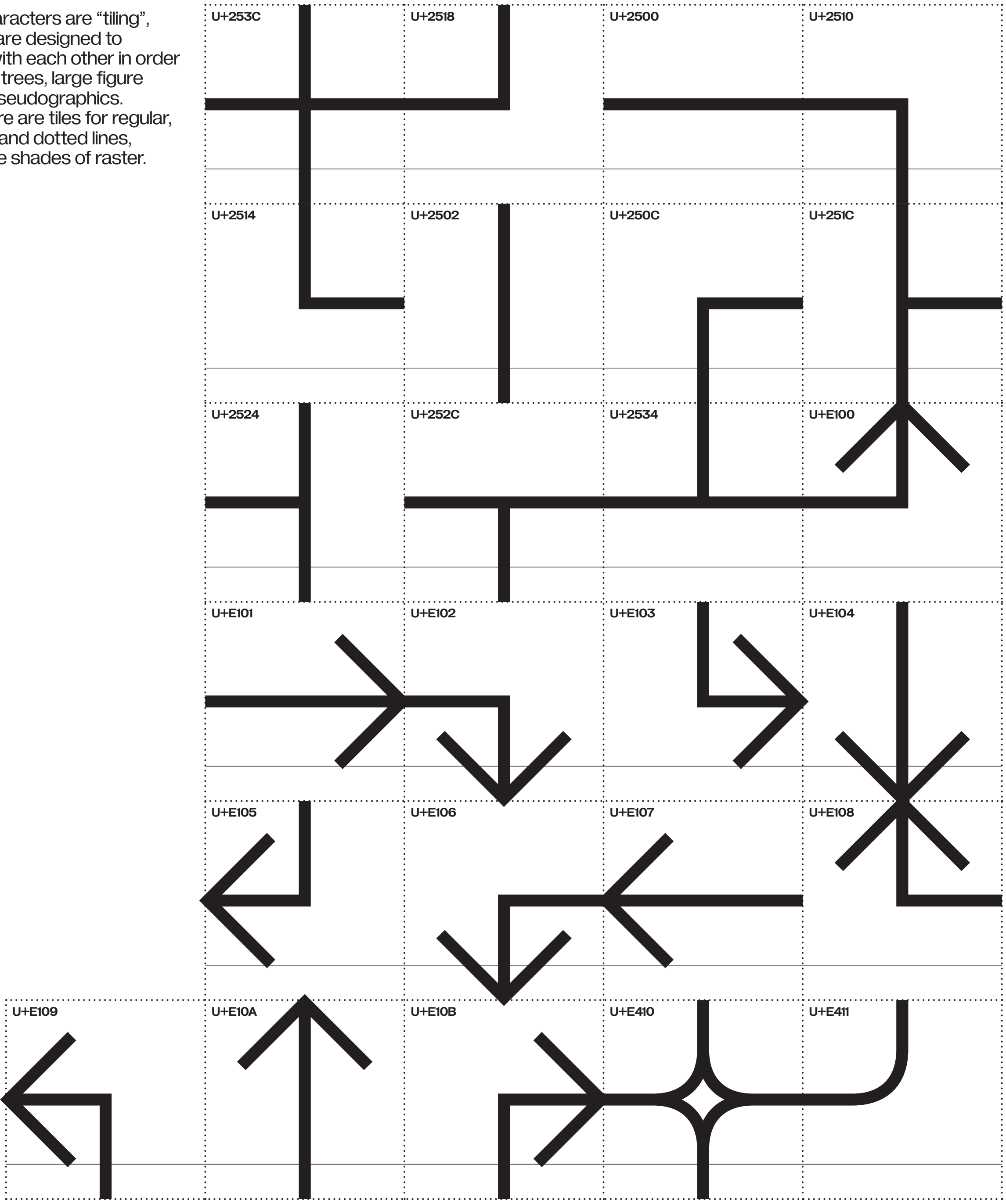


Expansion of the Alphabet

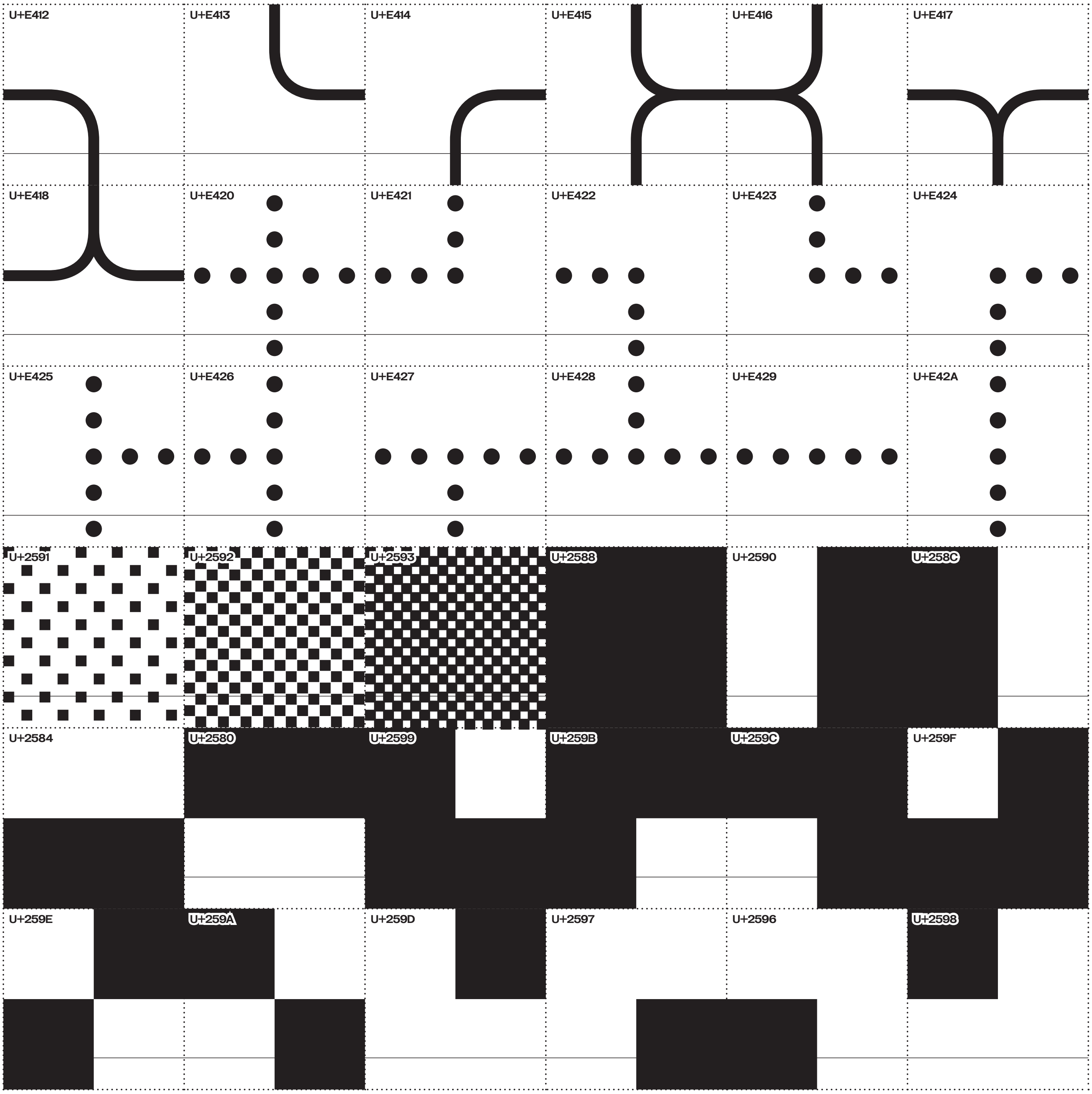
Tiles

Some of the symbol characters are “tiling”, which means that they are designed to combine and connect with each other in order to type frames, catalog trees, large figure brackets or any other pseudographics.

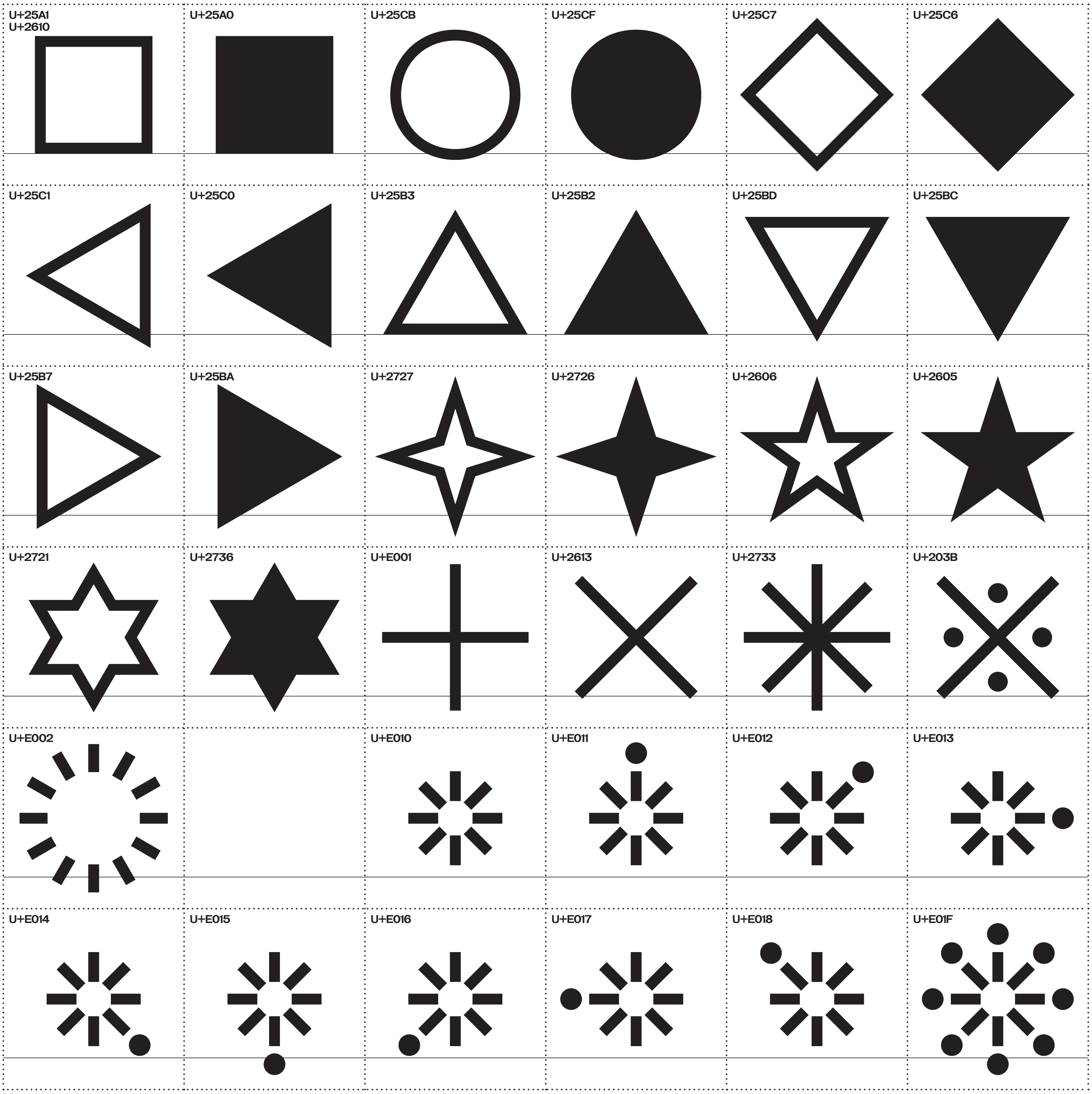
At the moment there are tiles for regular, arrowheaded, rounded and dotted lines, various blocks and three shades of raster.



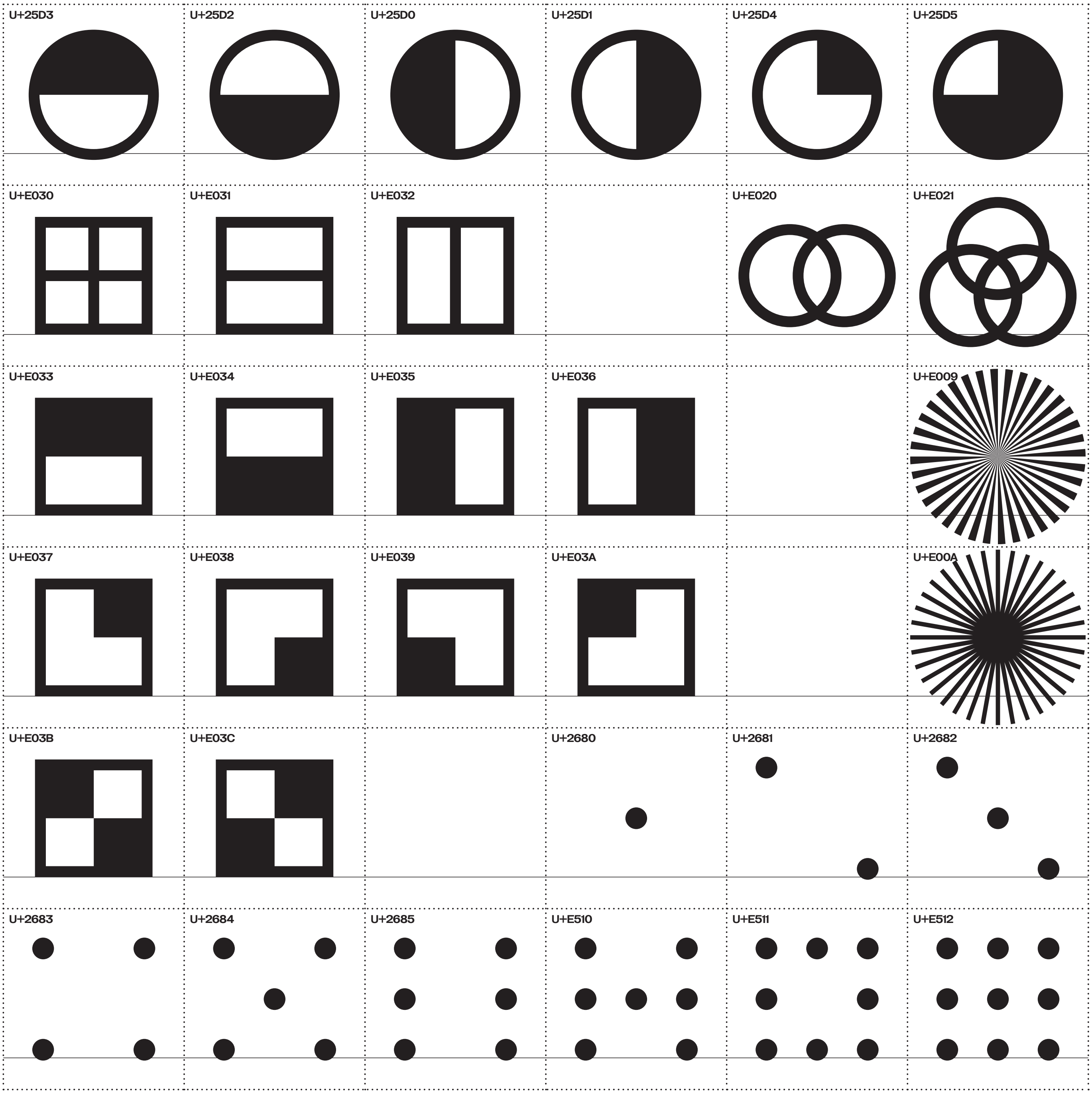
Expansion of the Alphabet



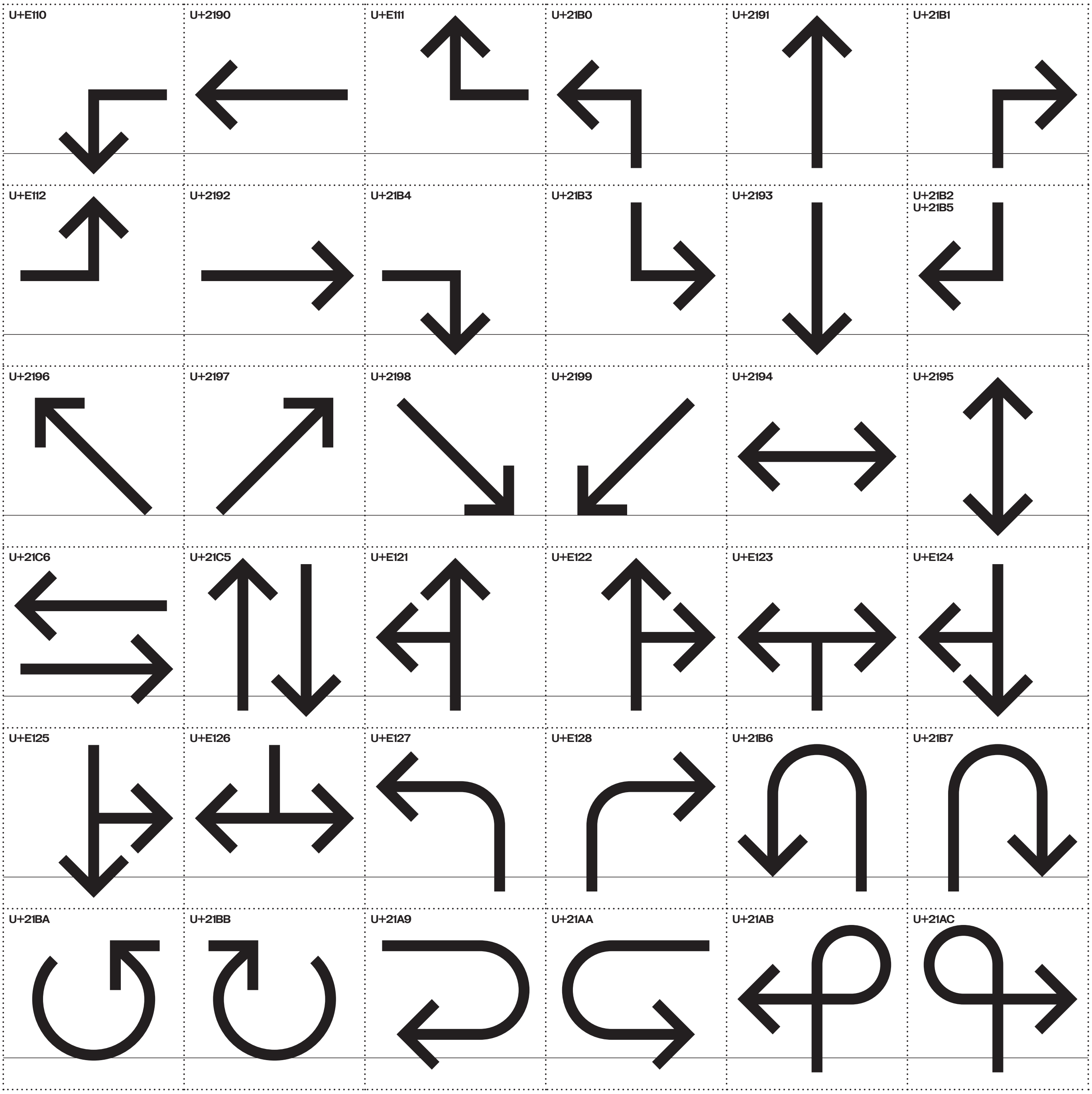
Expansion of the Alphabet



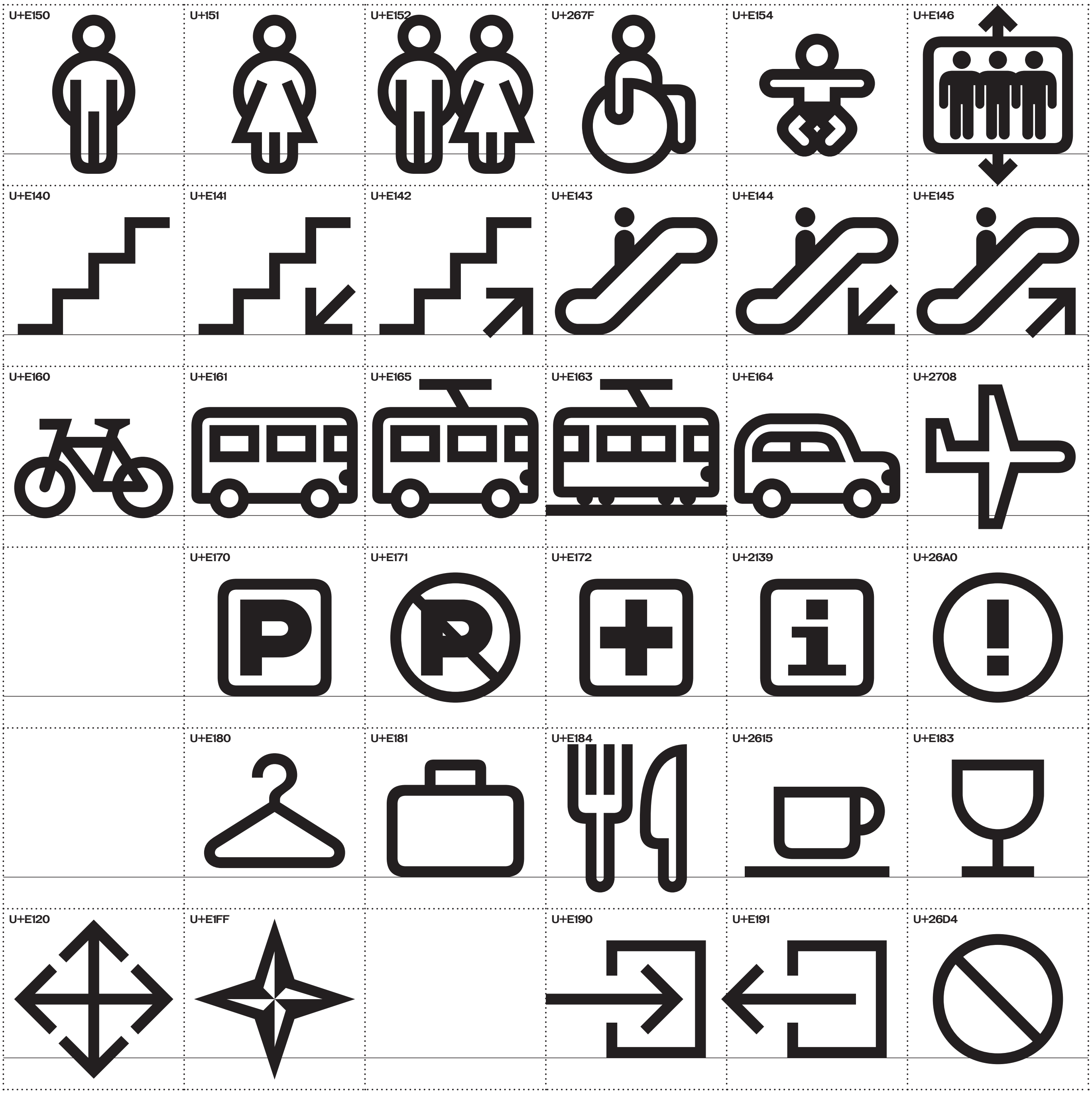
Expansion of the Alphabet



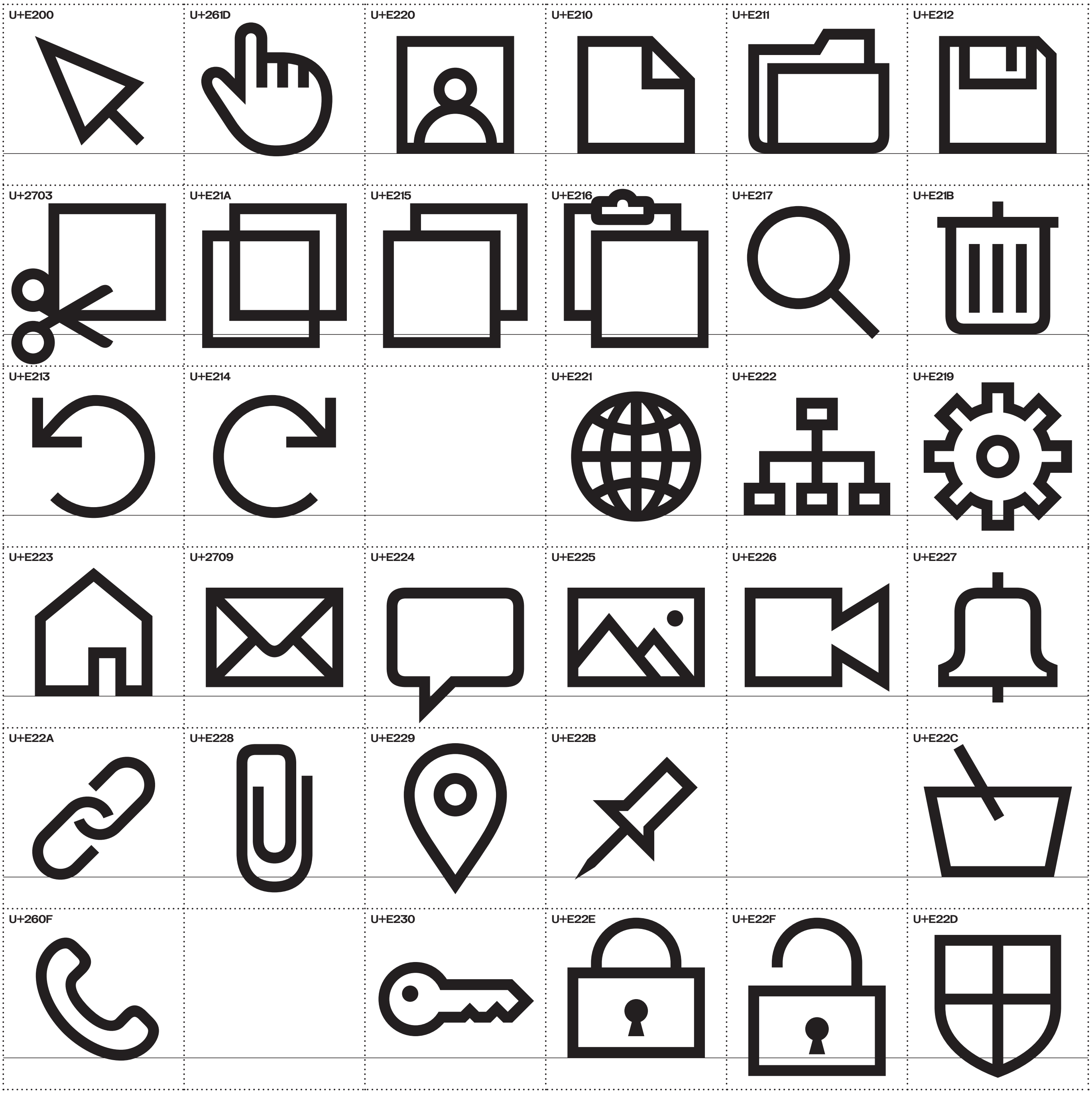
Expansion of the Alphabet



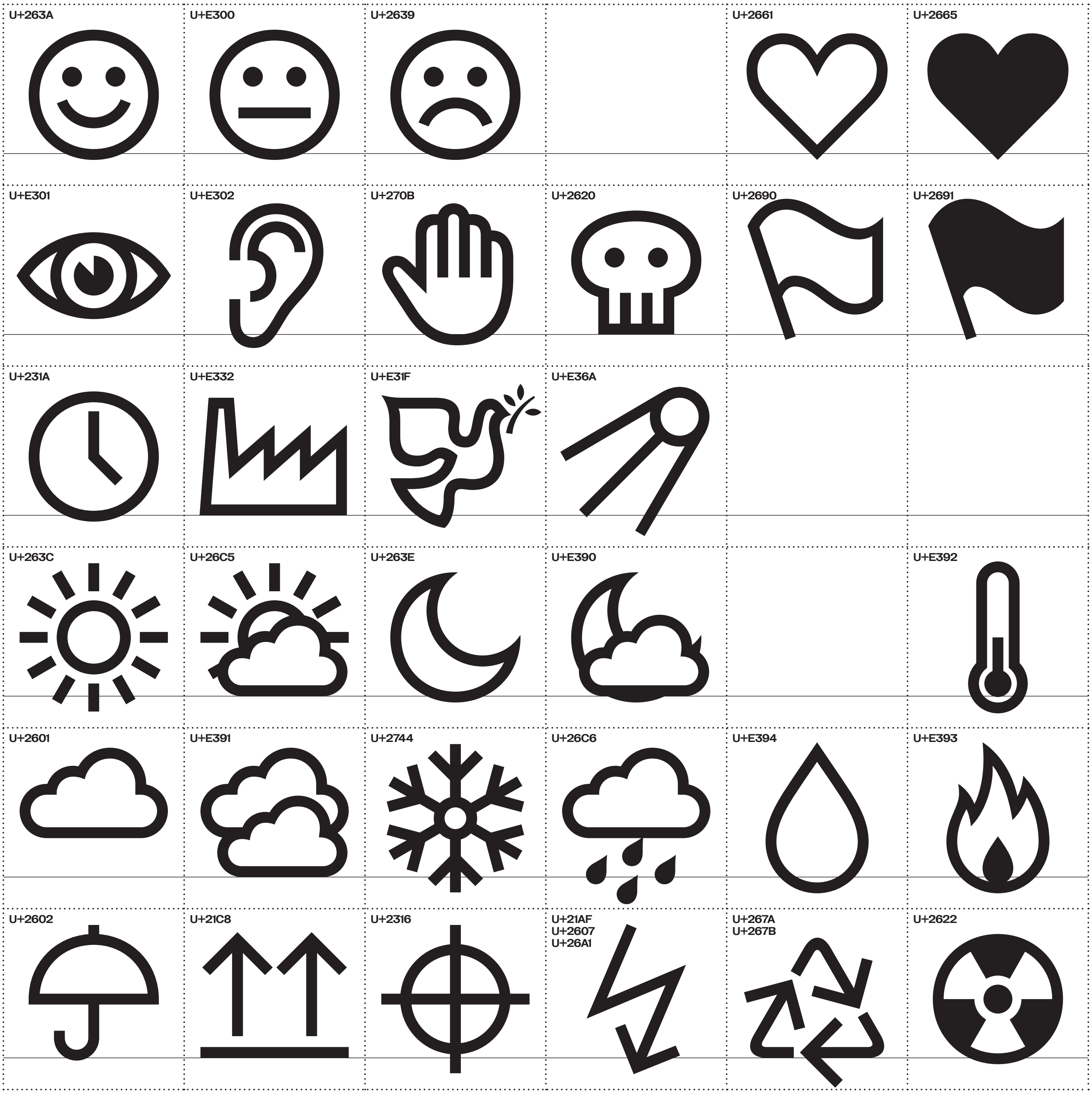
Expansion of the Alphabet



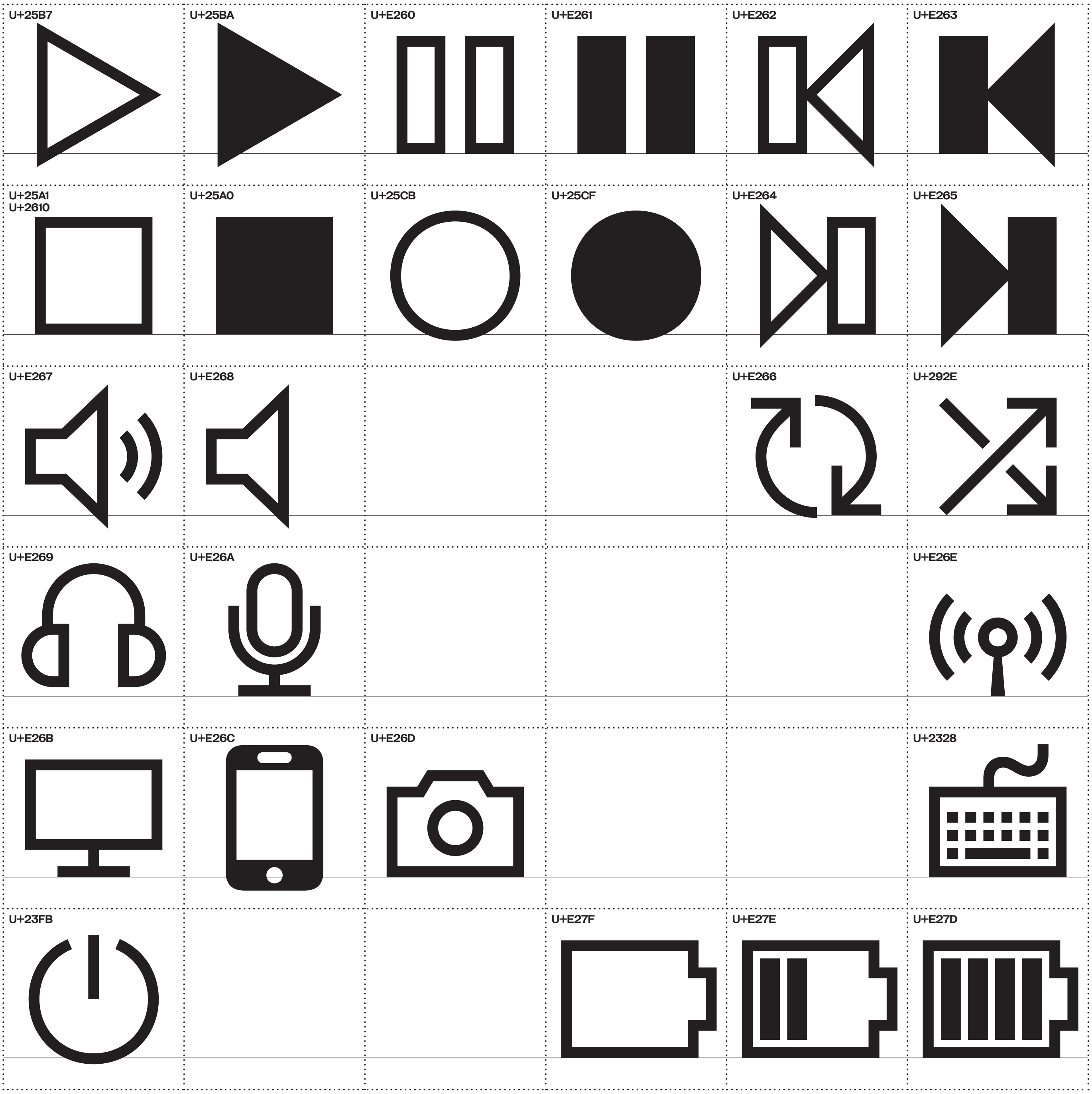
Expansion of the Alphabet



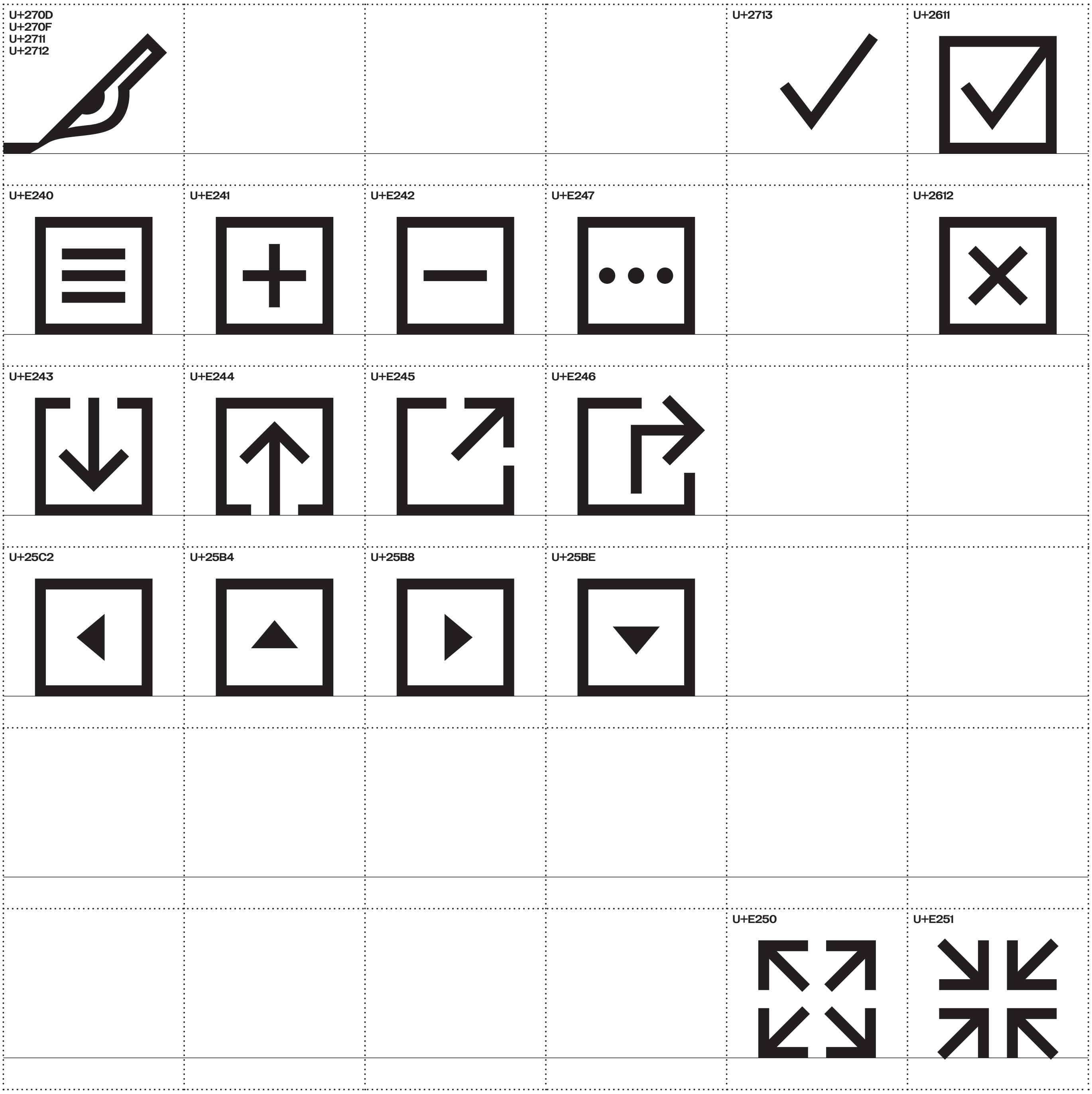
Expansion of the Alphabet



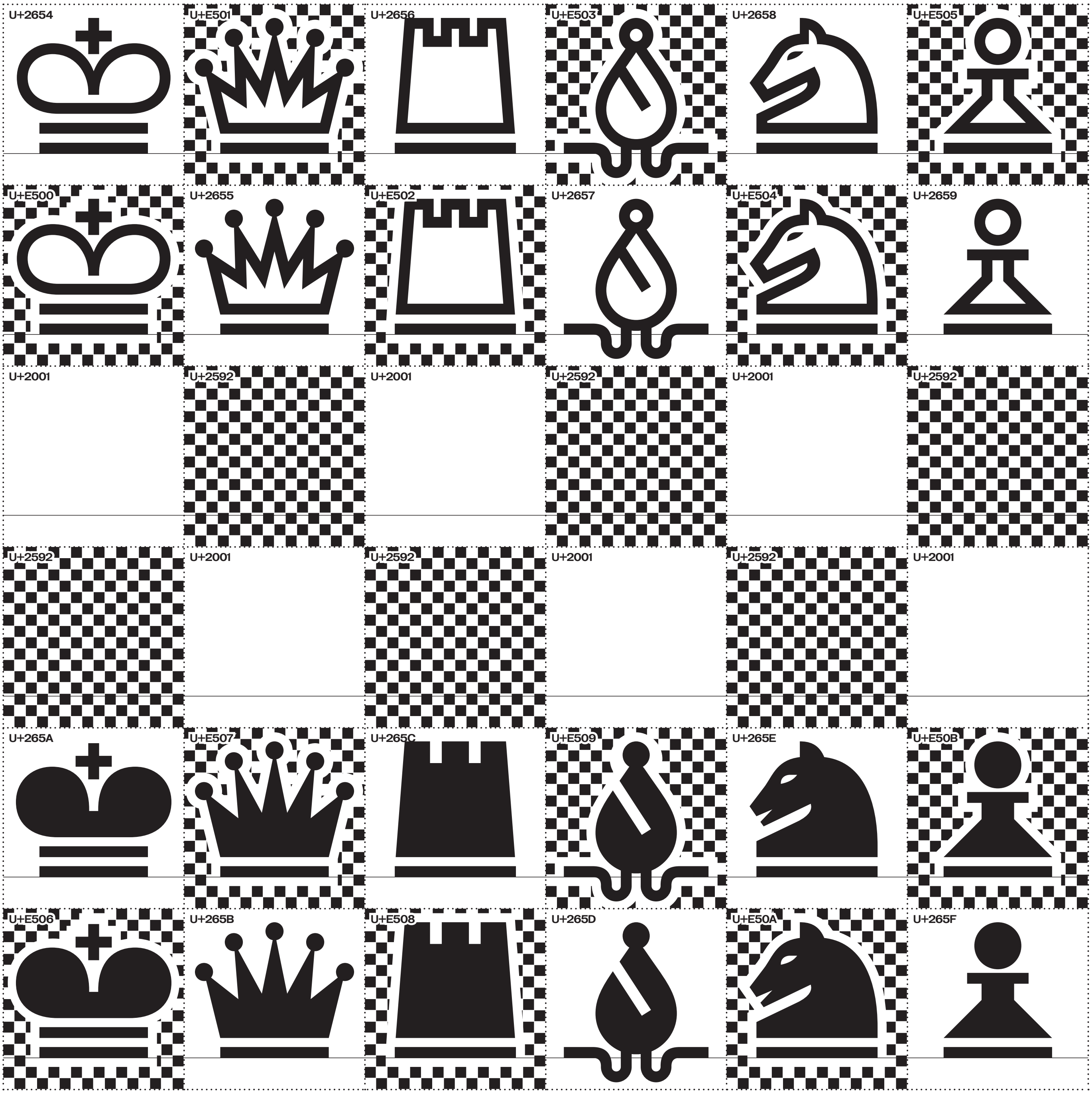
Expansion of the Alphabet



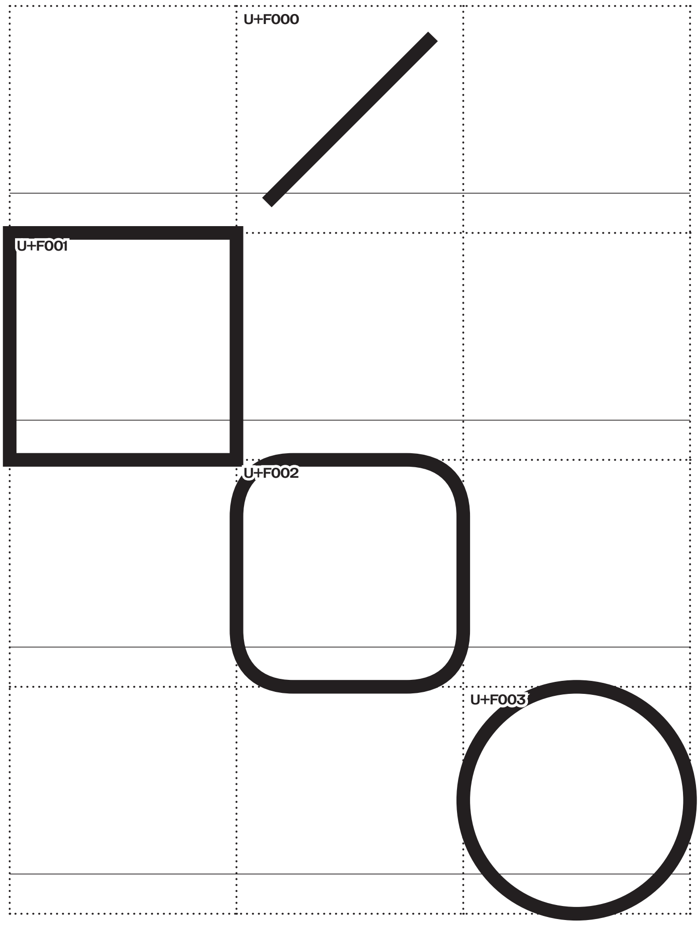
Expansion of the Alphabet



Expansion of the Alphabet



Expansion of the Alphabet



Overlays

Overlays are zero-width characters, which are designed to be combined with the previous pictogram in line. For example, the circle overlay in a line of regular text will be displayed like this: “◯”. Preceded by, for example, “ 🧍 ”, it will render “🧍◯”. Currently, there are four overlays: ⁄ □▢◯. They are combinable — you can use several of them at the same time: 🧍⁄□ — and tiling — putting a square overlay after each character of “☼⛅🌧” will result in “☼□⛅□🌧□”. The overlays can be used to create, for example, 🔇 Mute or ✋◯ Do Not Touch icons.

Notes

ISBN
978-3-00-067256-9

The
Temporary
State

2020