Google Fonts

Google Fonts

opsz 18, wght 650, track -.5

Roboto Flex

Roboto Flex

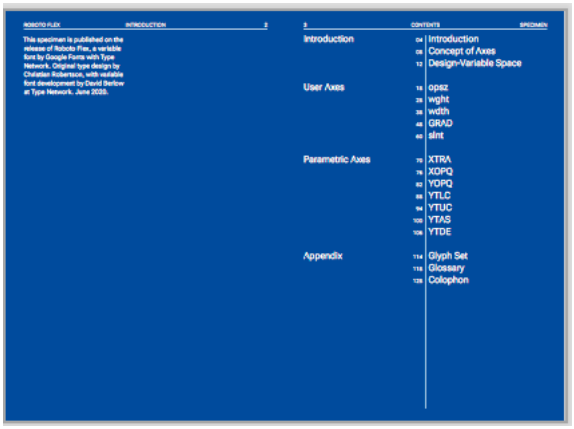
Specimen Book

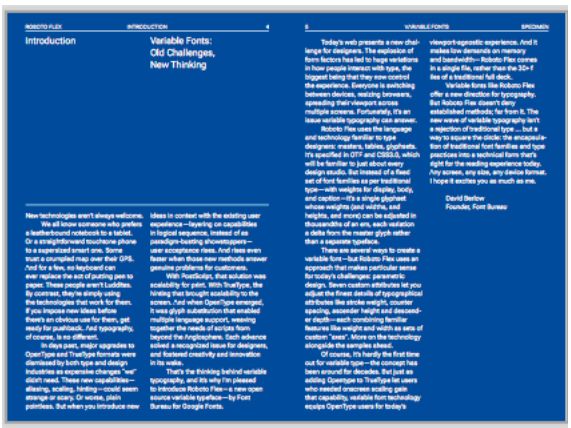
opsz wght wdth GRAD slnt XTRA XOPQ YOPQ YTLC YTUC YTAS YTDE

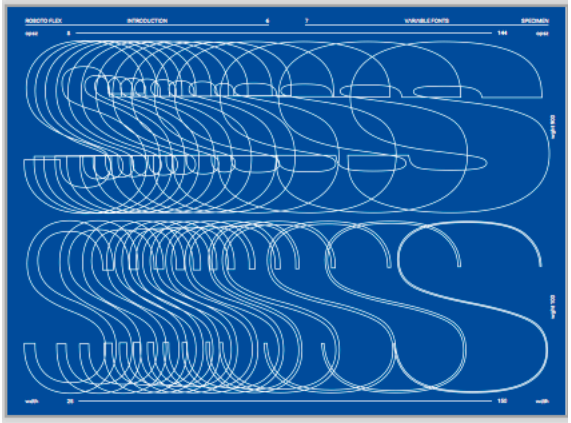
Roboto Flex Specimen Book Review

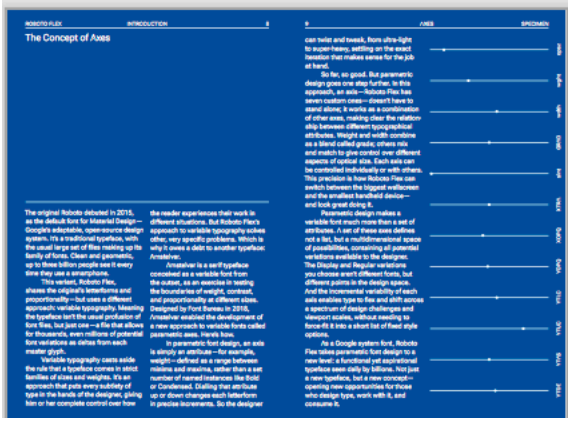
Wax Studios Google Roboto Book 061020.pdf

opsz 18, wght 650, track -.5









Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

I think there is no need to go into parametric space in creative display setting.

VARIABLE DESIGN SPACE

Roboto Flex designer David Berlow argues the most fundamental quality of type—its optical size—is a function of three attributes: weight, width, and height. Imagine them as a 3D box, but instead of width, length, and height, its dimensions are defined by width, visual weight and height.

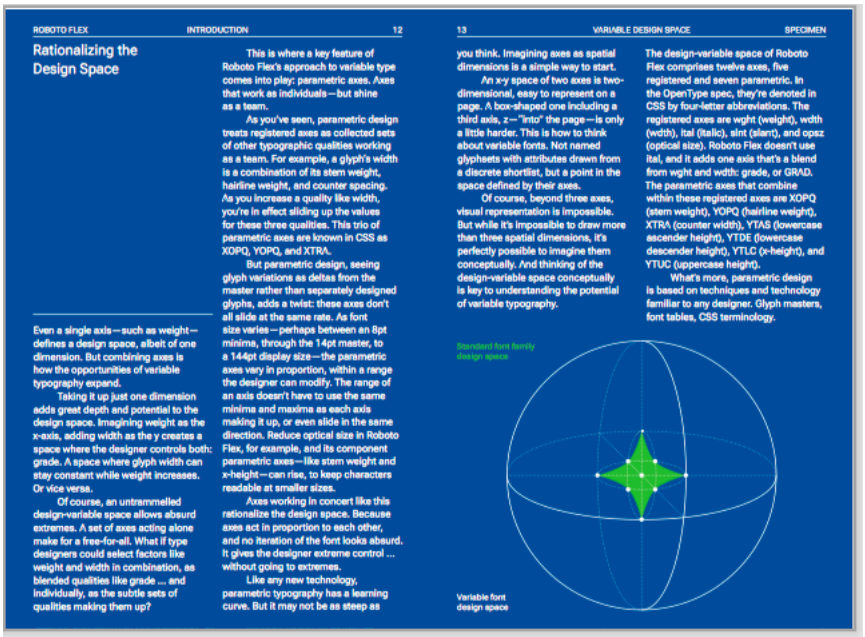
These basic attributes—weight, width, and height—are the most important factors contributing to the color and texture of a page. Which is why optical size is a registered axis: already part of the OpenType specification. (In CSS3.0, axes are denoted by a four-letter code, with optical size being opsz. (Common practice is to put registered axes in lowercase and capitalize others.)

There are five registered axes in OTF: weight, width, optical size, italic, and slant. Of these, Roboto Flex uses four—weight, width, optical size, and slant—plus a fifth that's a blend of weight and width, called grade (CSS: GRAD). (Grade is also part of Amstelvar. It's no coincidence Amstelvar and Roboto Flex play well together on the page: these different faces share a common philosophy.)

Sliding along any axis takes you to a different point in this design space—a narrower, taller, or heavier font variation. Because each attribute is a scale, and most axes in Roboto Flex are scaled as thousandths (milles) of an em, its design space contains not a dozen or so instances, but thousands.

Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



ROBOTO FLEX INTRODUCTION 12

Rationalizing the Design Space

Even a single axis—such as weight—defines a design space, albeit of one dimension. But combining axes is how the opportunities of variable typography expand.

Taking it up just one dimension adds great depth and potential to the design space. Imagining weight as the x-axis, adding width as the y creates a space where the designer controls both: grade. A space where glyph width can stay constant while weight increases. Or vice versa.

Of course, an untrammelled design-variable space allows absurd extremes. A set of axes acting alone make for a free-for-all. What if type designers could select factors like weight and width in combination, as blended qualities like grade … and individually, as the subtle sets of qualities making them up?

This is where a key feature of Roboto Flex's approach to variable type comes into play: parametric axes. Axes that work as individuals—but shine as a team.

As you've seen, parametric design treats registered axes as collected sets of other typographic qualities working as a team. For example, a glyph's width is a combination of its stem weight, hairline weight, and counter spacing. As you increase a quality like width, you're in effect sliding up the values for these three qualities. This trio of parametric axes are known in CSS as XOPQ, YOPQ, and XTRA.

But parametric design, seeing glyph variations as deltas from the master rather than separately designed glyphs, adds a twist: these axes don't all slide at the same rate. As font size varies—perhaps between an 8pt minima, through the 14pt master, to a 144pt display size—the parametric axes vary in proportion, within a range the designer can modify. The range of an axis doesn't have to use the same minima and maxima as each axis making it up, or even slide in the same direction. Reduce optical size in Roboto Flex, for example, and its component parametric axes—like stem weight and x-height—can rise, to keep characters readable at smaller sizes.

Axes working in concert like this rationalize the design space. Because axes act in proportion to each other, and no iteration of the font looks absurd. It gives the designer extreme control … without going to extremes.

Like any new technology, parametric typography has a learning curve. But it may not be as steep as

13 VARIABLE DESIGN SPACE SPECIMEN

you think. Imagining axes as spatial dimensions is a simple way to start.

An x-y space of two axes is two-dimensional, easy to represent on a page. A box-shaped one including a third axis, z—"into" the page—is only a little harder. This is how to think about variable fonts. Not named glyphsets with attributes drawn from a discrete shortlist, but a point in the space defined by their axes.

Of course, beyond three axes, visual representation is impossible. But while it's impossible to draw more than three spatial dimensions, it's perfectly possible to imagine them conceptually. And thinking of the design-variable space conceptually is key to understanding the potential of variable typography.

The design-variable space of Roboto Flex comprises twelve axes, five registered and seven parametric. In the OpenType spec, they're denoted in CSS by four-letter abbreviations. The registered axes are wght (weight), wdth (width), ital (italic), slnt (slant), and opsz (optical size). Roboto Flex doesn't use ital, and it adds one axis that's a blend from wght and wdth: grade, or GRAD. The parametric axes that combine within these registered axes are XOPQ (stem weight), YOPQ (hairline weight), XTRA (counter width), YTAS (lowercase ascender height), YTDE (lowercase descender height), YTLC (x-height), and YTUC (uppercase height).

What's more, parametric design is based on techniques and technology familiar to any designer. Glyph masters, font tables, CSS terminology.

Standard font family design space

Variable font design space

ROBOTO FLEX AXES 14

Axes

15 THE 12 AXES SPECIMEN

opsz
wght
wdth
GRAD
slnt

XTRA
XOPQ
YOPQ
YTLC
YTUC
YTAb
YTDE

USER AXES

PARAMETRIC AXES

Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

I think there is a good reason to show the range of the parmetric axes here.

Roboto Flex uses a set of 12 axes. So this book is organized the same way. First the five registered axes common to the OpenType Specification and CSS3.0, Then the seven "parametric" axes fundamental to Flex.

As you journey through this book, you'll see how each axis affects a glyph in isolation – but you'll also see how different axes work as a team, with sensible maxima and minima on one curtailing the risk of absurd extremes on another. It's an approach to variable typography that keeps all the moving parts connected and in proportion with each other.

What this means is there's an answer to your typographical challenge somewhere in the design space, waiting for you to discover it. Whatever that challenge is. The axes may be constrained. But the choices they enable are limitless.

opsz blends stem weight, hairline weight, counter width, and x-height as optical size. Its scale is based on familiar point sizes, from 8pt to 144pt, to allow for a huge range of styles.

opsz
8 – 144

BBbb

The **wght** registered axis controls overall glyph weight, ranging from 100 to 900 thousandths of an em. It's the axis instantly familiar to anyone with even a passing interest in type.

wght
100 – 900

BBbb

The **wdth** axis controls glyph width, within a range that lets the designer tune to fit line measure or type size without allowing absurdly wide characters.

wdth
25 – 151

BBbb

GRAD is a blended axis: weight and width acting in concert. It allows weight to rise without increasing width—leading to a range of different visual impressions on the page at different sizes.

GRAD
-1 – 1

BBbb

slnt allows the designer to fine-tune visual verticality. A narrow range of values (scaled in units roughly equivalent to degrees, from -10 to 0) offers a wide range of italic-style type without the

slnt
0 – 10

BBbb

The **XTRA** axis controls counter width. enabling precise justification. Its range is .323 to .603 of an em.

XTRA
323 – 603

BBbb

XOPQ is the axis for stem stroke weight, ranging from 27 to 175 milles of an em.

XOPQ
27 – 175

BBbb

YOPQ does the same for hairline stroke weight, with a range of 25 to 135. Minima and maxima prevent hairlines from disappearing at 8pt and below.

YOPQ
25 – 135

BBbb

YTLC covers x-height, and its range is from 416 to 570 milles of an em. It lets the designer increase lowercase height to levels that keep type readable even at tiny sizes.

YTLC
416 – 570

BBbb

YTUC deals with the height of uppercase glyphs, with extremes of 528 and 760. Again, visual size of small text is the main benefit.

YTUC
528 – 760

BBbb

The **YTAS** axis sets the height of lowercase ascenders, from 649 to 854.

YTAS
649 – 854

BBbb

YTDE sets the depth of lowercase descenders below the x-height, with values -305 to -98. Note the scale is negative.

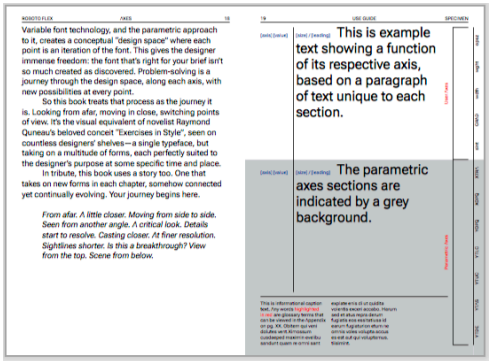
YTDE
-305 – -98

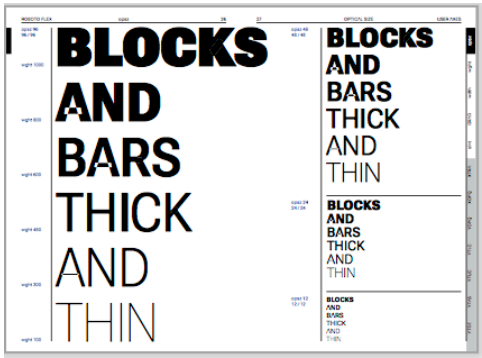
PPpp

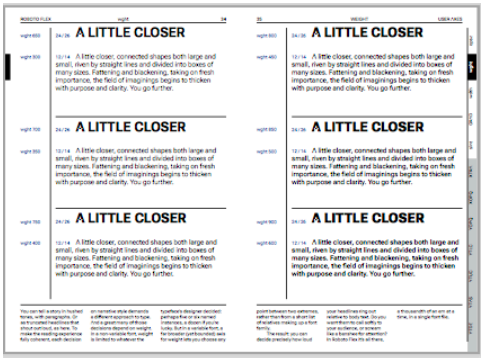
opsz wght wdth GRAD slnt XTRA XOPQ YOPQ YTLC YTUC YTAS YTDE

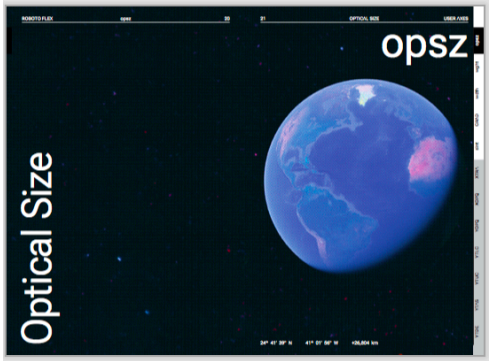
Roboto Flex Specimen Book Review

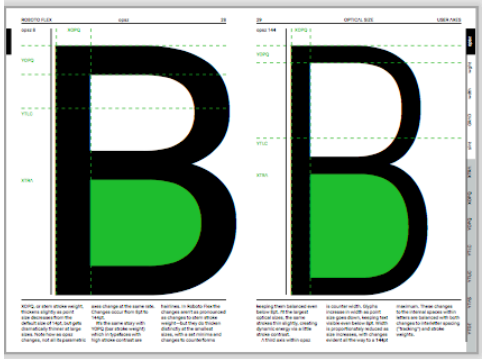
Wax Studios Google Roboto Book 061020.pdf

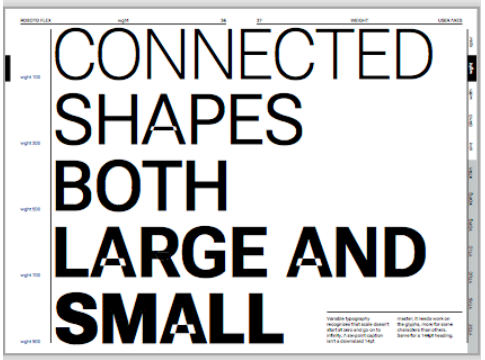


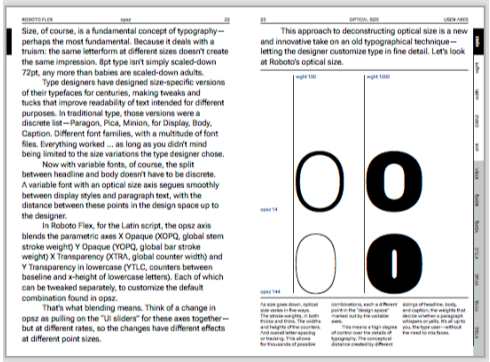


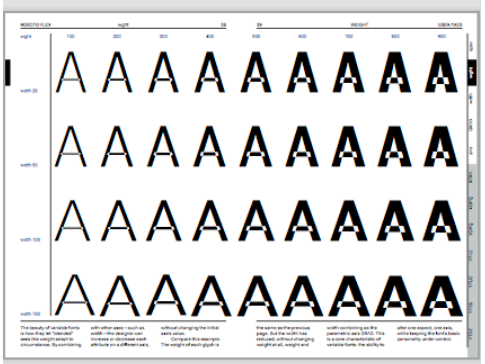


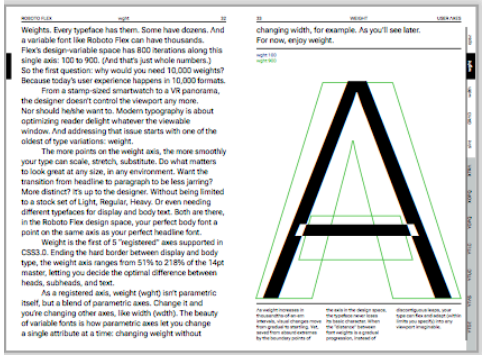


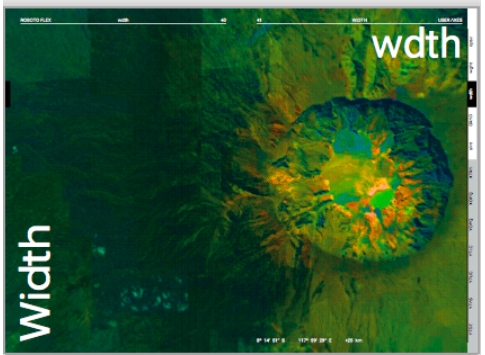


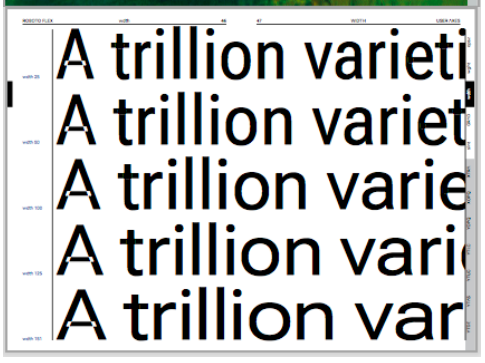








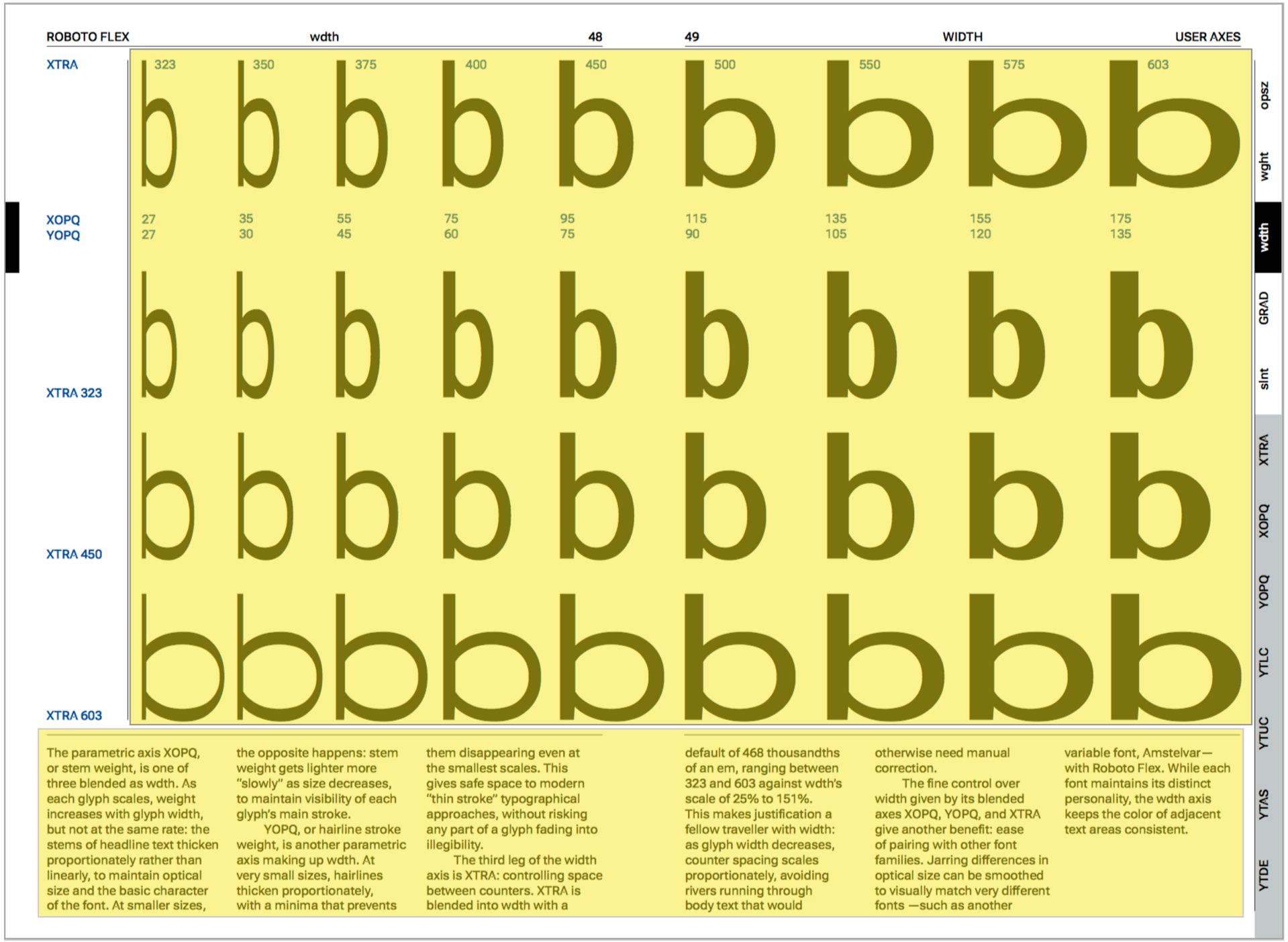




Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

page 48-49



The parametric axis XOPQ, or stem weight, is one of three blended as wdth. As each glyph scales, weight increases with glyph width, but not at the same rate: the stems of headline text thicken proportionately rather than linearly, to maintain optical size and the basic character of the font. At smaller sizes, the opposite happens: stem weight gets lighter more "slowly" as size decreases, to maintain visibility of each glyph's main stroke.

YOPQ, or hairline stroke weight, is another parametric axis making up wdth. At very small sizes, hairlines thicken proportionately, with a minima that prevents them disappearing even at the smallest scales. This gives safe space to modern "thin stroke" typographical approaches, without risking any part of a glyph fading into illegibility.

The third leg of the width axis is XTRA: controlling space between counters. XTRA is blended into wdth with a default of 468 thousandths of an em, ranging between 323 and 603 against wdth's scale of 25% to 151%. This makes justification a fellow traveller with width: as glyph width decreases, counter spacing scales proportionately, avoiding rivers running through body text that would otherwise need manual correction.

The fine control over width given by its blended axes XOPQ, YOPQ, and XTRA give another benefit: ease of pairing with other font families. Jarring differences in optical size can be smoothed to visually match very different fonts —such as another variable font, Amstelvar— with Roboto Flex. While each font maintains its distinct personality, the wdth axis keeps the color of adjacent text areas consistent.

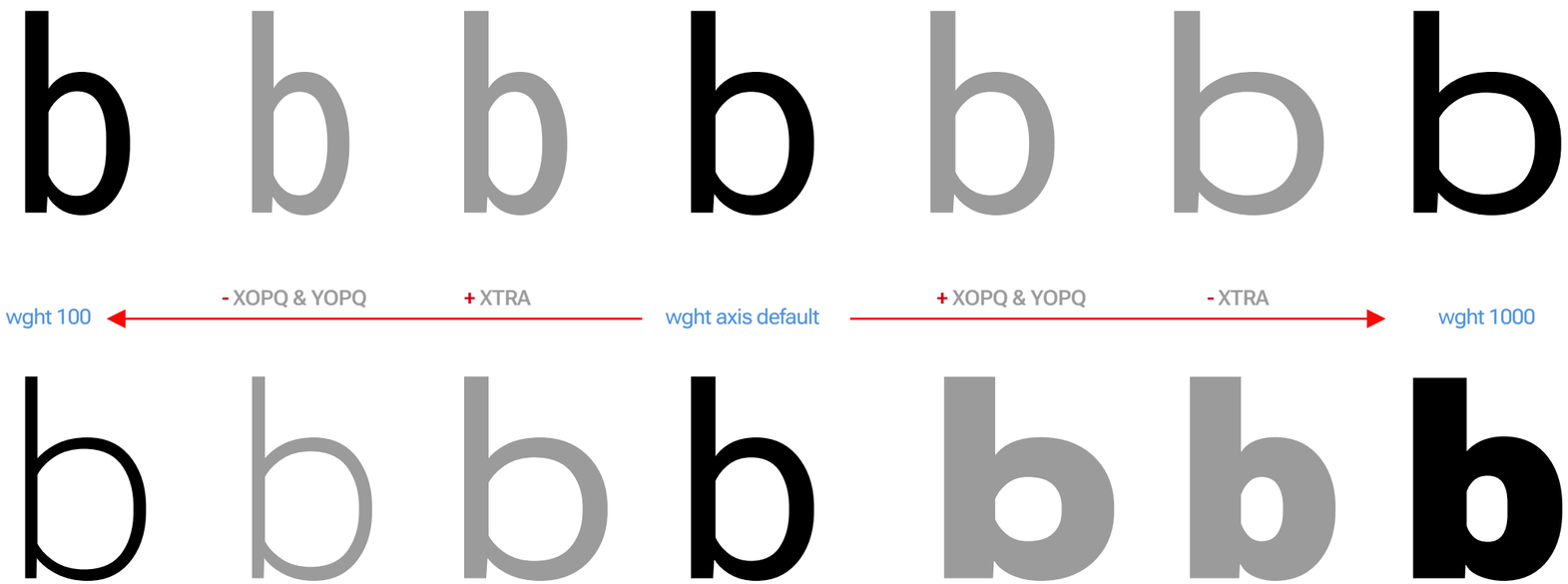
? Why is this here?

Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

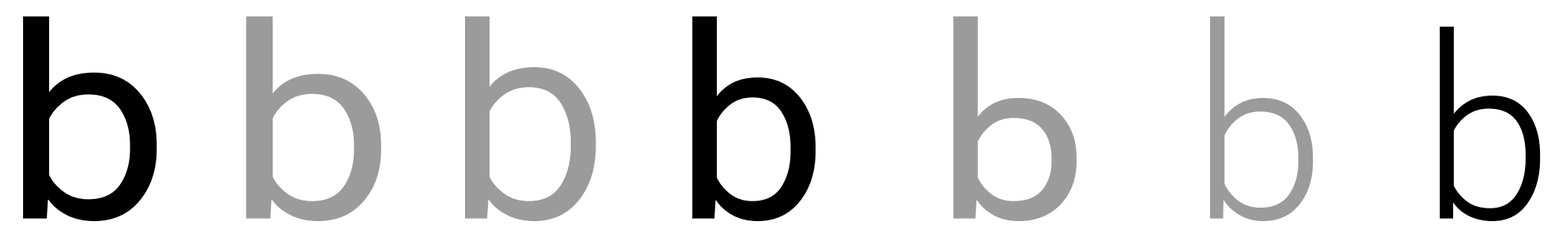
redo 48-49

wdth 50 ← - XOPQ - XTRA — wdth axis default — + XOPQ + XTRA → wdth 125



wght 100 ← - XOPQ & YOPQ + XTRA — wght axis default — + XOPQ & YOPQ - XTRA → wght 1000

opsz 6 ← + XOPQ & YOPQ + XTRA & YTRA — opsz 14 default — - XOPQ & YOPQ - XTRA & YTRA → opsz 144



The 3 rows above illustrate the parametric changes included in the width, weight and optical size axes. XTRA YTLC, XOPQ and YOPQ change in the grey columns, from the default in the center column, to the extremes of width, weight and optical sizes, in the far left and right columns.

In row one, the parametric axis XOPQ e.g., the main stem weight, is one of the blended axes of wdth, (top row). As width increases, XOPQ increases slightly, as width decreases XOPQ does too, again slightly. What controls the internal white space of glyphs, XTRA, makes most of the change to the width's increase and decrease. Other parametric axes, like those controlling glyph heights, remain unchanged as width changes.

The parametric axes that control both the main (XOPQ) and secondary stems (YOPQ), are the axes that control the weight axes, (in the second row). In addition, XTRA is part of the blend that helps maintain the proportion of weight to width, increasing as weight is reduced, and decreasing a weight is added. So width primarily changes XTRA, and weight changes the x and y stems controlled by XOPQ and YOPQ.

Optical size uses the width, weight a lowercase height controls combined, to size masters adjusted to produce wider, bolder and taller fonts for small uses, and narrower, lighter and slightly shorted styles for large sizes. This is shown in the third row, with intermediate styles shown with the changes made from the default style to the extremes by x and y direction parametric axes.

These additional axes can be used alone, to adjust for output considerations or to extend the effect of weight, width and optical size to the user's own extremes.

opsz wght **wdth** GRAD slnt XTRA XOPQ YOPQ YTLC YTUC YTAS YTDE

?

Move page to be between registered and parametric axes?