

Google Fonts  
Google Fonts

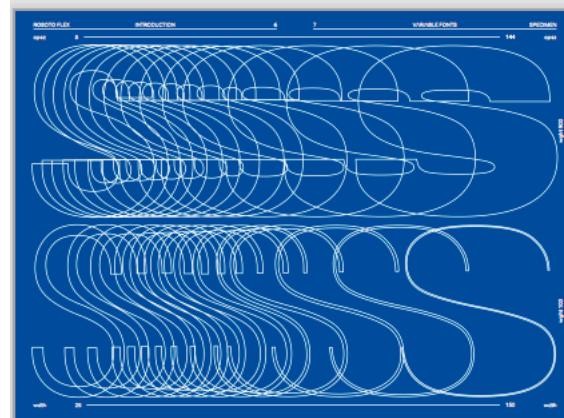
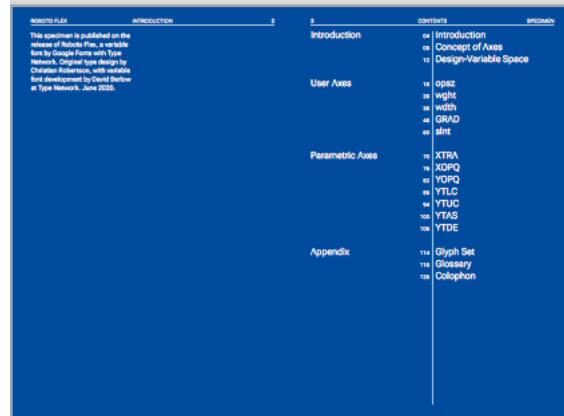
opsz 18, wght 650, track -.5

Roboto Flex  
Roboto Flex

# 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.

ROBOTO FLEX

INTRODUCTION

10

11

VARIABLE DESIGN SPACE

SPECIMEN



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

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 axes for those others as well. This set of parametric axes are known in CSS as XOPQ, YOPQ, and XTRA.

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. Add another weight axis, adding width as a *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 untrammeled 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?

**ROBOTO FLEX** **AXES** **14**

### Axes

13 **VARIABLE DESIGN SPACE** **SPECIMEN**

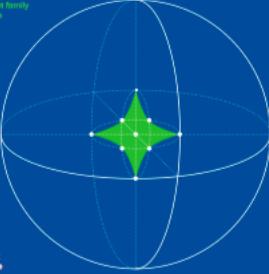
you think. Imagining axes as spatial dimensions is a simple way to start. An x-y space of two axes is two-dimensional, easy to represent on a page. A box-shaped one including a third axis, z—"into" the page—is only a little harder. This is how to think about variable fonts. Not named glysheets with attributes 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 hard to imagine 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.

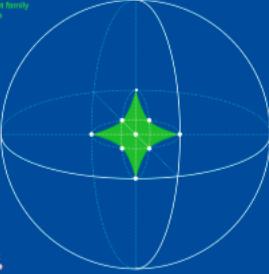
The design-variable space of Roboto Flex comprises twelve axes, five registered and seven parametric. In the OpenType spec, they're denoted in CSS by four-letter abbreviations. The registered axes are *wght* (*weight*), *wdth* (*width*), *ital* (*italic*), *slnt* (*slant*), and *opsz* (*optical size*).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



**15** **THE 12 AXES** **SPECIMEN**

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

USER AXES PARAMETRIC AXES

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

## ROBOTO FLEX

## AXES

16

I think there is a good reason to show the range of the parametric 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.

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.

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.

**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.

**sint** 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

opsz  
8 – 144

wght  
100 – 900

wdth  
25 – 151

GRAD  
-1 – 1

sint  
0 – 10

# BBbbb

# BBbbb

# BBbbb

# BBbbb

# BBbbb

17

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

XTRA  
323 – 603

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

XOPQ  
27 – 175

**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

**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

**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

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

YTAS  
649 – 854

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

YTDE  
-305 – -98

THE 12 AXES

SPECIMEN

# BBbbb

# BBbbb

# BBbbb

# BBbbb

# BBbbb

# BBbbb

# PPpp

opsz

wght

wdth

GRAD

sint

XTRA

XOPQ

YOPQ

YTLC

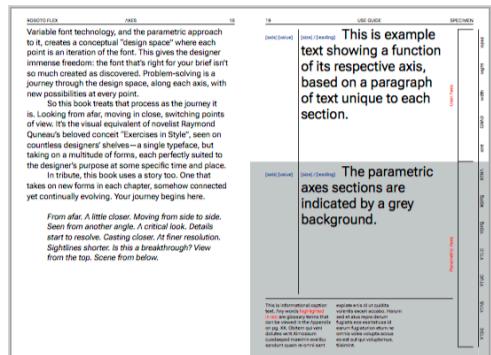
YTUC

YTAS

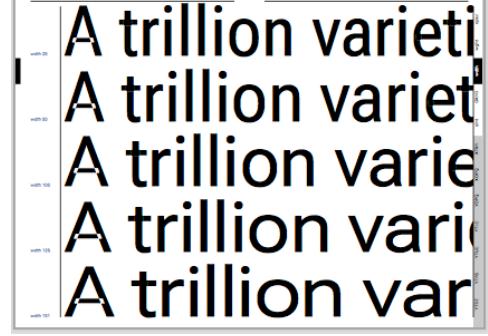
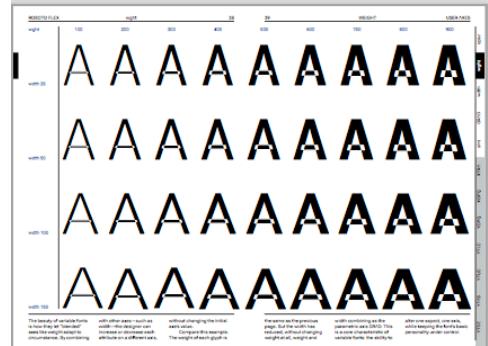
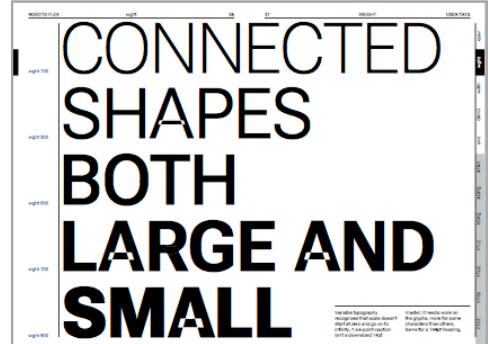
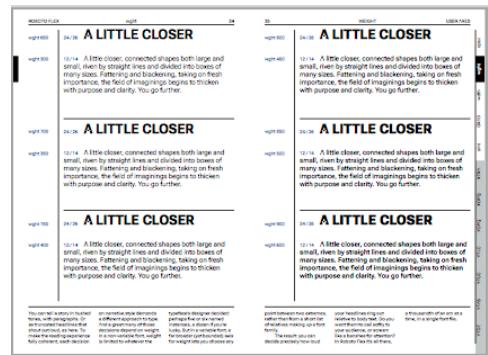
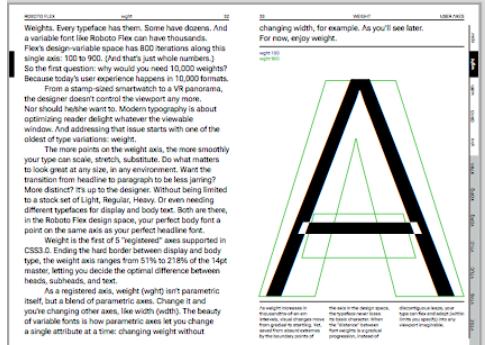
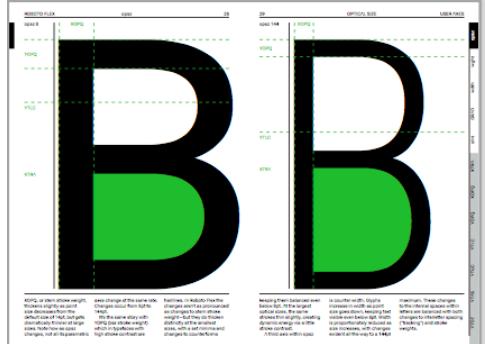
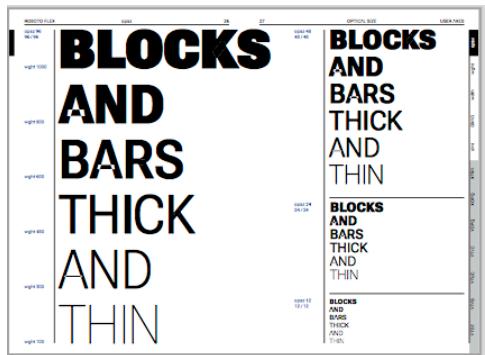
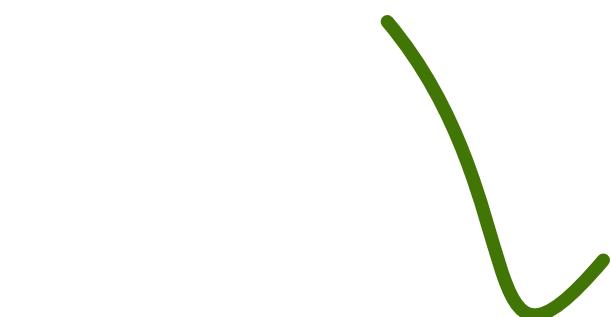
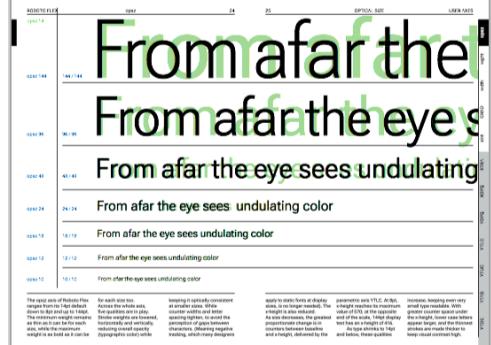
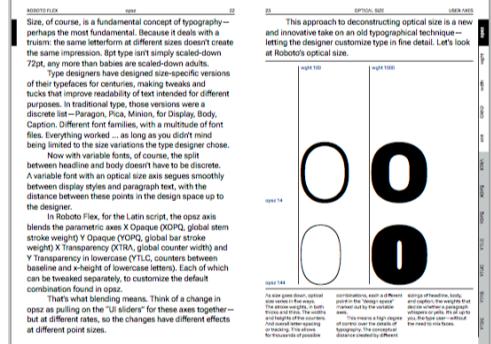
YTDE

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



## Optical Size



# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

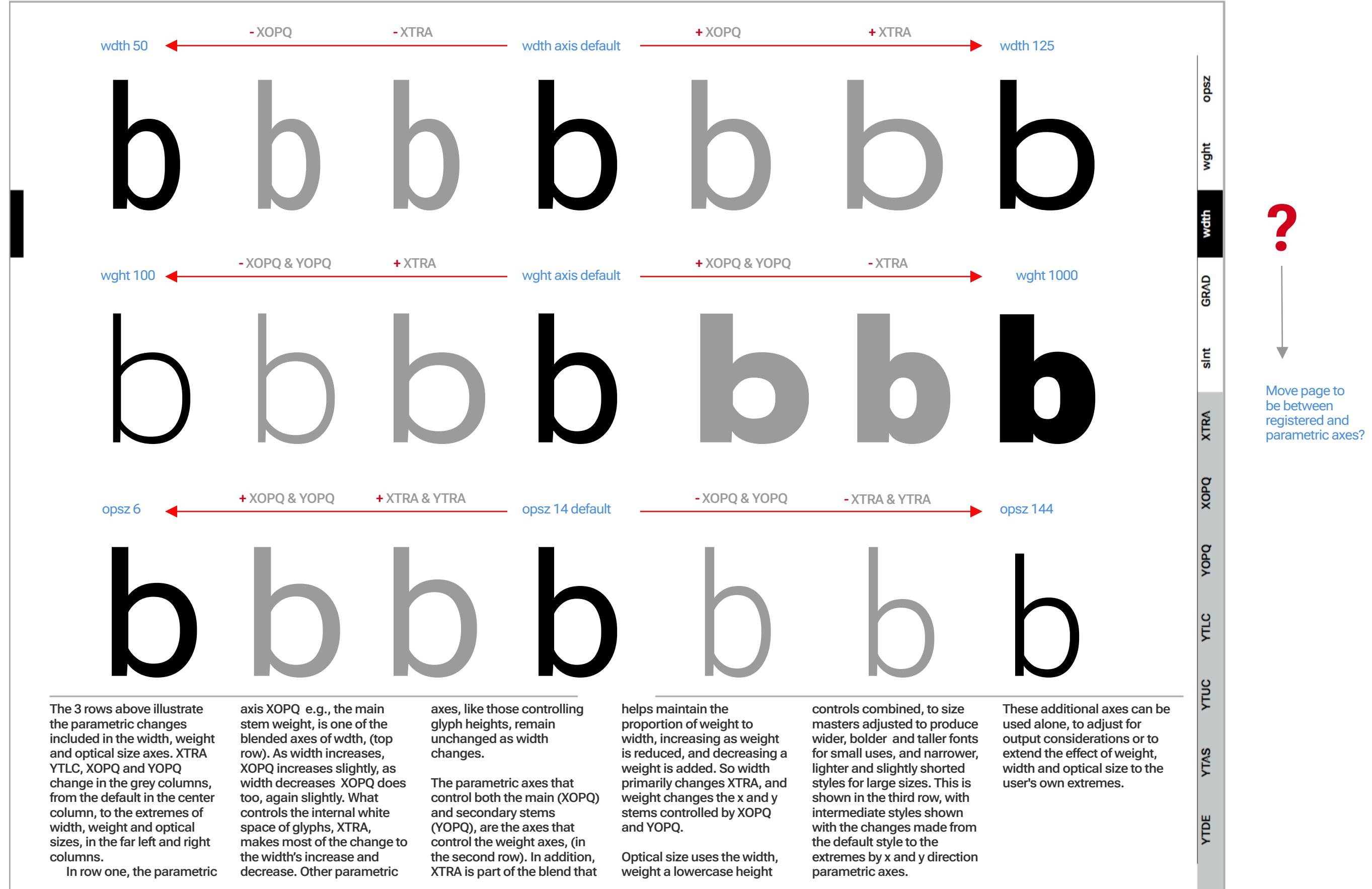
page  
48-49



# Roboto Flex Specimen Book Review

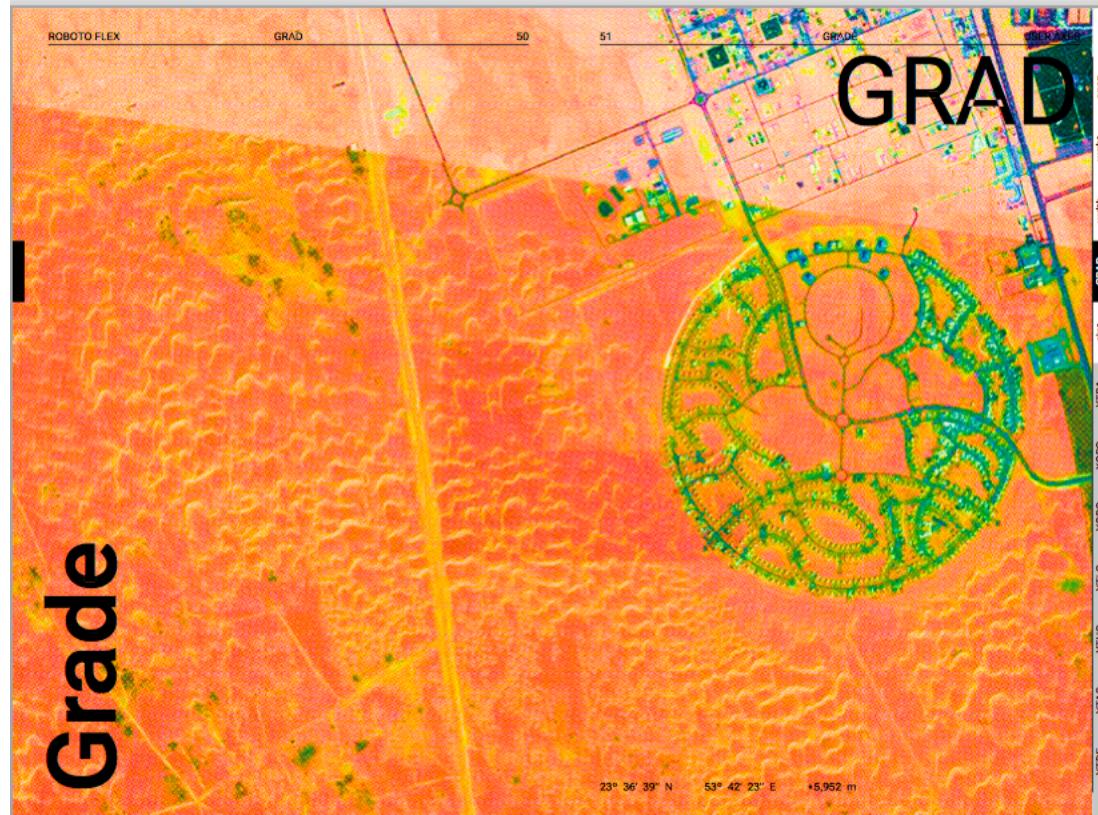
Wax Studios Google Roboto Book 061020.pdf

redo  
48-49



# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



## Important edit

The grade axis increases or decreases glyph weight without changing width—so keeping *glyph, word and line-endings constant in text layout*.

ROBOTO FLEX

GRAD

52

When you drill down into how readers perceive type, some factors matter more than others. Combine two of the big ones—weight (wght) and width (wdth)—and you get another of Roboto Flex's axes: grade. (CSS: GRAD.) Not a parametric axis, but a blended one.

The grade axis increases or decreases glyph weight without changing width—so keeping counter weight, justification, and kerning constant too. A simple idea. But with countless applications in the user experience. Accessibility. Affordance. And flexibility that goes all the way.

The term dates back to the 1990s. But the concept is far older, riffing on how old-school letterpress type varied with the viscosity of ink in use. The heavier the ink, the bolder the letterform on the printed page. And, of course, paper matters too: low-absorption stock meant greater ink spread, risking counters shrinking to dots and tails turning into blobs.

To this day, magazines use different stock for color and mono pages, making ink choice an art in the pursuit of a consistent reader experience. And while letterpress is far from dead, the modern idiom—more relevant to most designers—has inherited many of the same challenges.

An easy UI option to switch between Light and Dark Modes creates a harder situation for designers. Grade can be the fix, letting a typeface with thin strokes thicken up on a black background without changing position, allowing seamless switching between modes. While bolding an inline hyperlink when the user's cursor hovers over it—a feature of many website UIs—creates a problem when that increase in width makes the whole page dance a jig. Grade can solve it.

53

GRADE

USER AXES

The grade axis is recognized by a number of type players: Google, TypeNetwork, Font Bureau, Microsoft, Axis-Praxis, and Fontsmith among them. In this part of the Roboto Flex specimen, you'll see it in action across a number of use cases and reader scenarios.



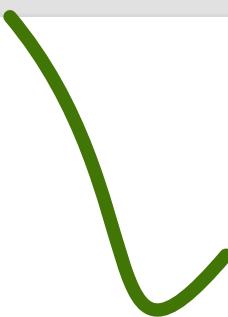
With grade, that perfect line length can stay constant at both extremes. You can do it manually, dialling down width as you weight up, or let the grade axis do it for you. The defined range in Roboto

Flex is from -1.0 to +1.0, with a default of zero. It avoids percentages, since grade is relative only to itself: a simple minus-to-plus scale keeps the axis linear and easy to reset to its default. If a percentage,

similar-sized changes in value—from 10% to 20% versus 50% to 60%—might create very different visual impressions.

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



ROBOTO FLEX	GRAD	54	GRADE	USER AXES
GRAD 1 30/36	A critical look hints at overarching structure, organizing principles. How can more become more without changing? Or less become more at the same time? But the eye sees no contradiction.	24/28 A hints at overall organizing principles. How can more become more without changing?	14/18 A critical look hints at overarching structure, organizing principles. How can more become more, without changing? Or less become more at the same time? But the eye sees no contradiction.	8/10
	Or less become more at the same time? But the eye sees no contradiction.		Or less become more at the same time? But the eye sees no contradiction.	
	at the same time? But the eye sees no contradiction.		at the same time? But the eye sees no contradiction.	

In these examples, it's the change in grade that's led to the changes in weight and width—the axis being a blend of both. Without a grade axis, the designer would have to balance weight and width by hand, a time-consuming chore. Dark Mode—and other accessibility reqs—are one area where grade makes all the difference. When contrast flips between extremes of light and dark, increasing the grade thickens strokes that might otherwise disappear visually. Ordinal numbers and punctuation are more visible, without affecting line measure or relative positioning of words: as weight goes up, width goes down. Even at extremes of the range (-1.0 on the left, +1.0 on the right) letter spacing and counter width stay constant, without the typeface losing any of its character.

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

FYI

Little or none of this text is being illustrated.

May be confusing for readers.

ROBOTO FLEX GRAD 56 57 GRADE USER AXES

GRAD 0 GRAD 1

# A critical look hints at overarching structure

When glyph widths, letter spacing, and kerning are fixed, emphasizing inline text—such as a link being hovered over—doesn't lead to the page joining the text refresher group; plus another option for making text stand out. Line length, of course, is identical in all cases.

Like other axes, grade adapts to circumstances set by the designer. As a blend of weight and width, you can change grade manually by dialling up a heavy weight, then combining with basic slanting—creating the same outcome as using the grade axis alone. And you can do the same with the parametric axes. Increasing stem weight and hairline weight (XOPQ and YOPQ) then decreasing counter width (XTRA) both produce type with greater weight on the same width.

Low-resolution screens are becoming rarer, but they're not extinct. And rendering text at any weight lower than 200 on a pixelated viewport is a thinned-out font. Of course, this brings together old and new—when letterpress printers adjusted inks for different paper stock, their goal was to keep type consistent across all of them. Roboto Flex's grade axis does no less for the digital user experience. The grade axis can also bring different scripts together in the same visual space without one dominating.

The same principle applies to fonts outside Roboto's Latin/Greek/Cyrillic scope. Languages whose typefaces have low visual size (such as Arabic) or high (such as Chinese) relative to Western script can be balanced visually by increasing grade, to create a pleasing visual impression when next to each other.

ROBOTO FLEX GRAD 58 59 GRADE USER AXES

GRAD 0.2



# Roboto Flex Specimen Book Review

# Wax Studios Google Roboto Book 061020.pdf



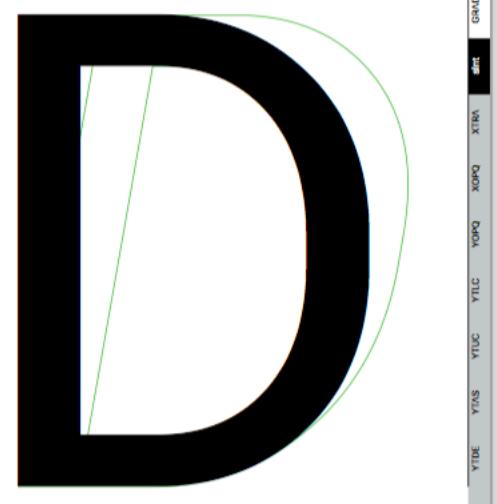
The two registered axes `slnt` and `ital` (slant and italic) might seem similar, but they're used quite differently. In brief: `slnt` acts on an existing glyph, modifying its deltas to create an oblique look. The italic registered axis, by contrast, normally substitutes that glyph for another. A variable font can use either axis, but generally not both.

In serif font families, italic glyphs tend to be qualitatively different to non-italics. Strokes, finials, and serifs aren't just different weights; they're different shapes. But sans serifs—like Roboto Flex—are another matter entirely. A slanted Roboto Flex glyph makes for an excellent oblique, and lessens the need for a specific set of italic glyphs, because it's deeply geometric to start with: no finials to finesse.

The `slint` attribute in CSS3.0 does not use milles (thousandths) of an em as its scale—nor could it. (A slanted glyph transforms by different quantities at its baseline and x-height.) Instead, its scale is best thought of as degrees of counterclockwise from the vertical. Since an oblique effect takes hold at very small values, its range is small too: 0 to -10. Note degrees are negative, but slant will be to the right.

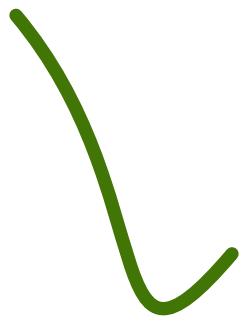
Let's look at Roboto Flex's slnt axis in use.

63	SLANT	USER AXES
sint 0		
sint 10		



# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



# Roboto Flex Specimen Book Review

# Wax Studios Google Roboto Book 061020.pdf

ROBOTO FLEX	slnt	66	67	SLANT	USER AXES
slnt 0	30 / 35	Seen from another angle. <i>Up becomes across in a symphony of tilt. Yet understanding rises as forms take on new shapes. We are moving into something, and its possibilities are endless.</i>	slnt 0 XTRA-468	12 / 14 Seen from another angle. Up becomes across in a symphony of tilt. Yet understanding rises as forms take on new shapes. We are moving into something, and its possibilities are endless.	wdth wdth
slnt 10			slnt 10 XTRA-468	12 / 14 Seen from another angle. Up becomes across in a symphony of tilt. Yet understanding rises as forms take on new shapes. We are moving into something, and its possibilities are endless.	GRAD GRAD
			slnt 10 XTRA-488	12 / 14 Seen from another angle. Up becomes across in a symphony of tilt. Yet understanding rises as forms take on new shapes. We are moving into something, and its possibilities are endless.	YTLC YTLC
				A great advantage of the slnt axis is the visual smoothness it brings to any block of text that mixes uprights with obliques. The slnt attribute applies its deltas without changing cap height or x-height. Even a large volume of oblique instances doesn't create a jolting or jerking impression; the obliqued glyphs are, after all, the same master letterforms, whatever the degree of variation. A slanted glyph in Roboto Flex takes up less horizontal space in a line when kerning and justification are in play—in common with other oblique variations. By combining slnt with XTRA (counter spacing) the designer can maintain a constant character count as the line measure—or not.	YPOS YPOS XOPO XOPO YTLG YTLG YTUC YTUC



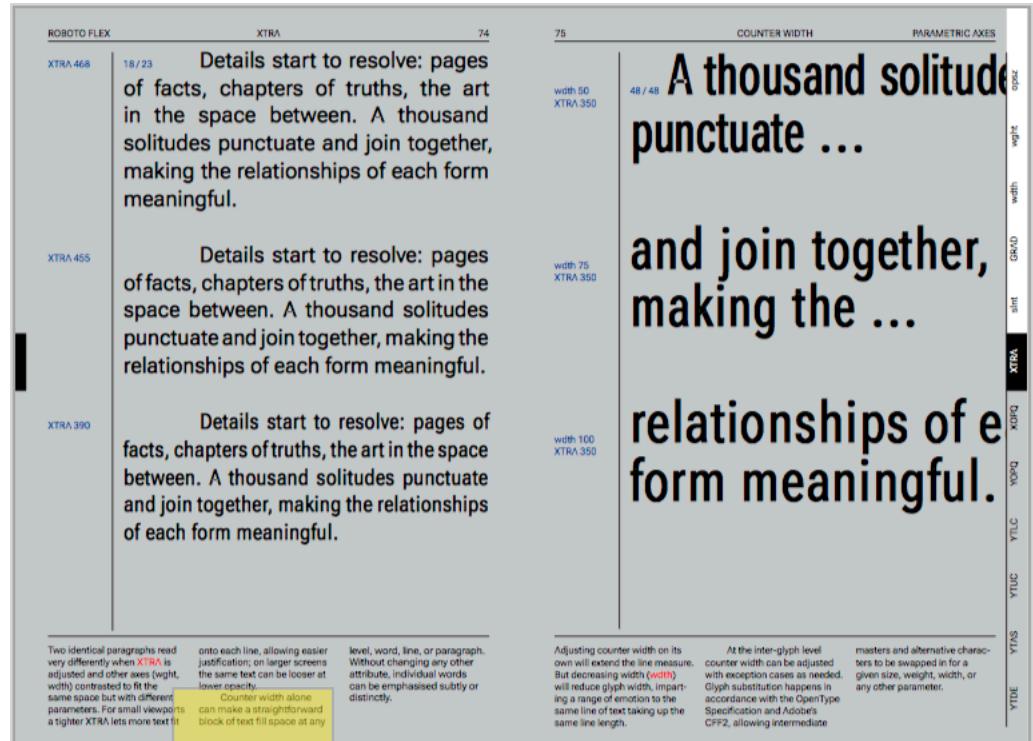
# Roboto Flex Specimen Book Review

# Wax Studios Google Roboto Book 061020.pdf



# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



Two identical paragraphs read very differently when **XTRA** is adjusted and other axes (wght, wdth) contrasted to fit the same space but with different parameters. For small viewports a tighter XTRA lets more text fit

onto each line, allowing easier justification; on larger screens the same text can be looser at lower opacity.

Counter width alone can make a straightforward block of text fill space at any

onto each line, allowing easier justification; on larger screens the same text can be looser at lower opacity.

Counter width alone can make a straightforward block of text fill space at any

level, word, line, or paragraph. Without changing any other attribute, individual words can be emphasised subtly or distinctly.



# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



ROBOTO FLEX XOPQ 78

In Roboto Flex, **XOPQ**—the axis for stem weight—blends with YOPQ and XTRA as the width registered axis. But as a parametric axis, it can also be adjusted as a singular quality, without causing a change in its companions. This lets designers fine-tune stem weight in isolation, allowing for huge flexibility in the range of variations in the design space.

These examples illustrate XOPQ this way: as a single parameter, ranging stem weight while keeping glyph width, counter width, and hairline stroke weight constant. Blended-axis changes (i.e. to width as a whole) are shown for context.

When **XOPQ** is changed in isolation, glyph and counter width remain the same, with the whole character set filling the same space for a given size.

The **XOPQ** axis changes stem weight in a range from 27 to 175 units of an em from the 14pt master. (Its companion axis YOPQ ranges from 23 to 135 to maintain stem weight heavier than hairline weight for all values of the blended with axis.) Note glyph width or justification does not change.

XOPQ 27 - 175

ROBOTO FLEX XOPQ 80

12 / 14 Casting closer, paragraphs and passages become distinct, all different yet in perfect context. What's this? A shifting of parts already solid? Small changes that make the whole work.

ROBOTO FLEX XOPQ 85 YOPQ 90 40 / 42 A shifting of parts already solid? Small changes that make the whole work.

ROBOTO FLEX XOPQ 85 YOPQ 95 48 / 48 A shifting of parts already solid? Small changes that make the whole work.

The minimum value of **XOPQ** remains readable down to its first, thickening the main stroke of each character down to a practical lower limit. At large sizes its maxima keeps Roboto from getting any thicker than its "companion" axis **YOPQ** when the two axes are at different sizes for example, very small caption text.

Please note that designer fine-tune the look of the font at different sizes for example, very small caption text.

STEM WEIGHT PARAMETRIC AXES

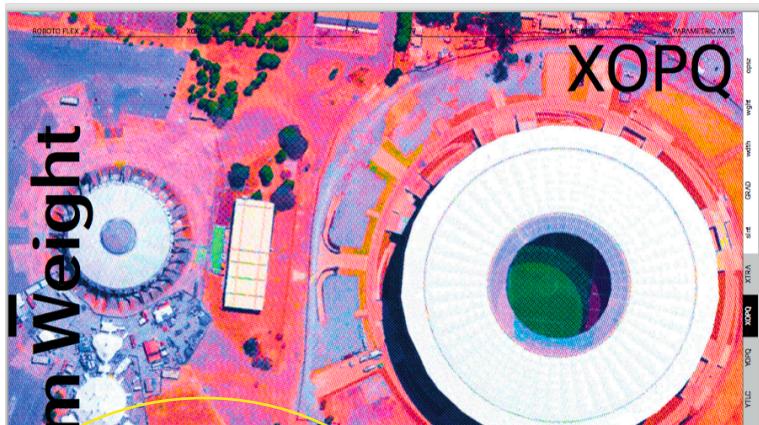
81 XOPQ 100 YOPQ 80 144 / 123

Small changes that make the whole work.

Stem weight, of course, can never fall to zero, so possibility stays where it belongs: outside the design space of Roboto Flex. Adjusting **XOPQ** between its extremes at all sizes, from 8pt to 72pt inclusive, means that the design remains. The **XOPQ** axis continues affecting optical size up to 144pt. Roboto Flex's complete set of diacritics for Latin, Greek, and Cyrillic character sets has been remapped to work within the variable font, each mark-based and weighted for the 144pt master to slide smoothly up and down with **XOPQ**. This ensures diacritics on accented characters do not look "pasted on."

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf

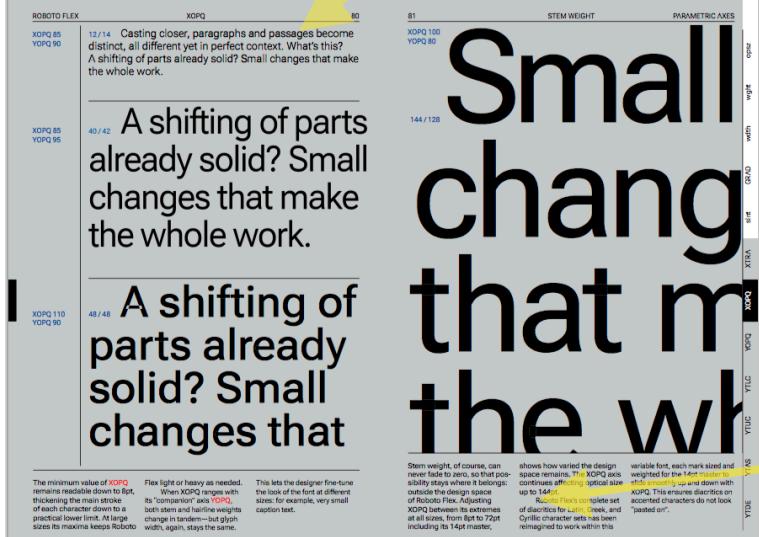
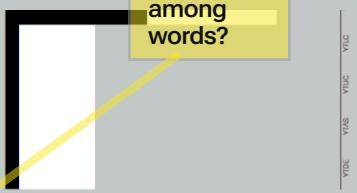


XOPQ

80

12 / 14 Casting closer, paragraphs and passages become distinct, all different yet in perfect context. What's this? A shifting of parts already solid? Small changes that make the whole work.

It looks like there are unwanted changes to the text's wght spec among words?



Small  
chang  
that m  
the wh

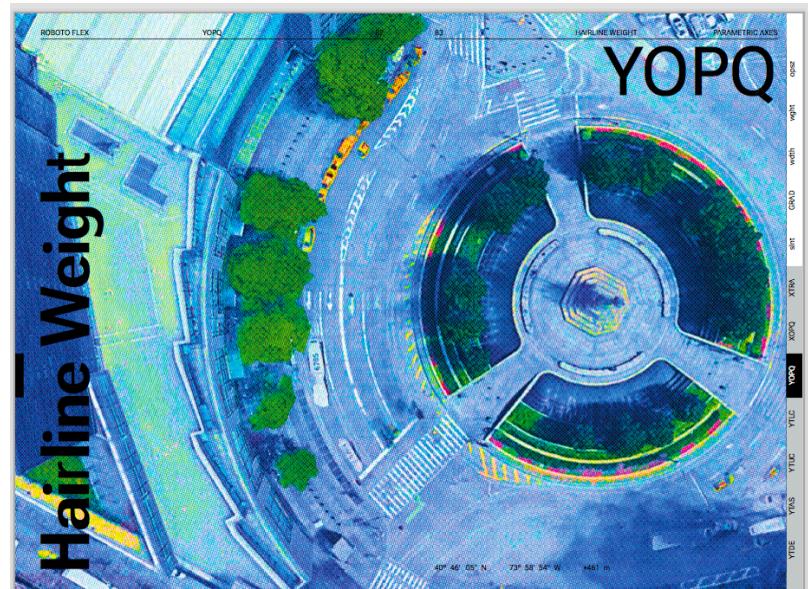
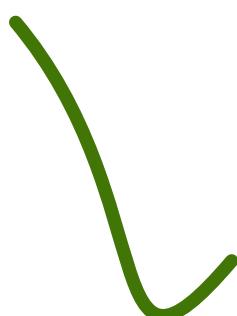
It looks like the text spec is changing from normal to condensed mid-caption

shows how varied the design space remains. The XOPQ axis continues affecting optical size up to 144pt.

Roboto Flex's complete set of diacritics for Latin, Greek, and Cyrillic character sets has been

# Roboto Flex Specimen Book Review

## Wax Studios Google Roboto Book 061020.pdf



ROBOTO FLEX YOPQ 83 HAIRLINE WEIGHT PARAMETRIC AXES

The companion axis to **XOPQ**, in CSS as **YOPQ**, covers hairline stroke weight. Once again, this quality can be adjusted independently of the other axes that blend as width: **XOPQ** and **XTRA**. Changing hairline weight alone offers personality changes to a block of text echoing a range of classic type styles from poster forms to letterpress.

These examples illustrate **YOPQ** in isolation, ranging hairline weight while keeping glyph width, counter width, and stem weight constant. Blended-axis changes (i.e. to **width** as a whole) are shown for context.

84

85 HAIRLINE WEIGHT PARAMETRIC AXES

When **YOPQ** is changed as a single axis, glyph, counter width, and stem weight are unchanged. The same character set filling the same space for a given size.

The **YOPQ** axis changes hairline weight in a range from 25 to 135 miles of an em (1px to 4pt stroke). The lower limit means hairline weight stays visible at all sizes. Note

YOPQ 25 – 135

ROBOTO FLEX YOPQ 86 HAIRLINE WEIGHT PARAMETRIC AXES

12 / 14 At finer resolution, sets of sentences emerge and assert individuality, the thinnest parts yet rich with content. Horizontals and verticals acting to delineate and demarcate, derive deepening definition.

12 / 14 At finer resolution, sets of sentences emerge and assert individuality, the thinnest parts yet rich with content. Horizontals and verticals acting to delineate and demarcate, derive deepening definition.

12 / 14 At finer resolution, sets of sentences emerge and assert individuality, the thinnest parts yet rich with content. Horizontals and verticals acting to delineate and demarcate, derive deepening definition.

12 / 14 At finer resolution, sets of sentences emerge and assert individuality, the thinnest parts yet rich with content. Horizontals and verticals acting to delineate and demarcate, derive deepening definition.

86

87 HAIRLINE WEIGHT PARAMETRIC AXES

12 / 14 At finer resolution, sets of sentences emerge and assert individuality, the thinnest parts yet rich with content. Horizontals and verticals acting to delineate and demarcate, derive deepening definition.

144 / 120

At fin  
resolu  
sets o  
sente

When teamed with its "companion" axis **XOPQ**, hairline and stem weight can change without the other attributes changing. Once more this offers opportunities for fine-tuning the font at different sizes, both display and body.

The maximum value of **XOPQ** is kept so the text readable down to 8pt, thickening hairlines down to a practical lower limit of 1px and no lower. At large sizes its maxima keeps **YOPQ** consistent with **XOPQ**, keeping the text readable down to 8pt, thickening hairlines down to a practical

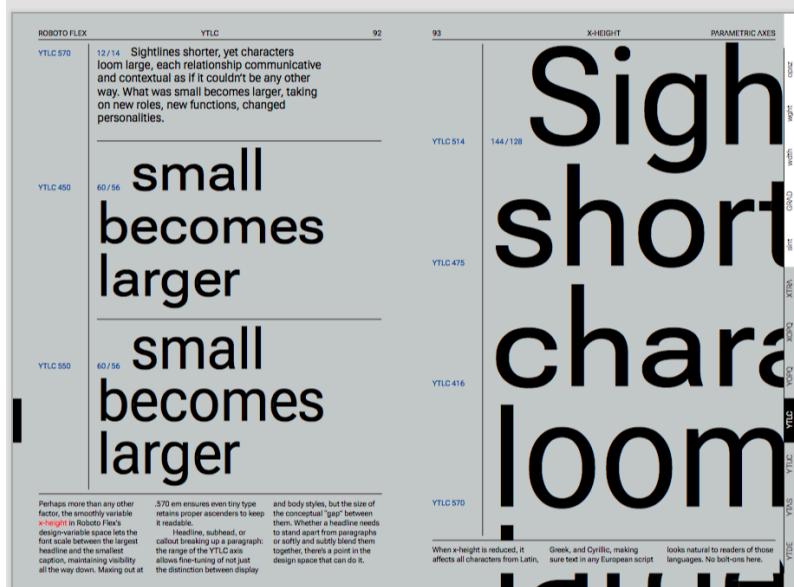
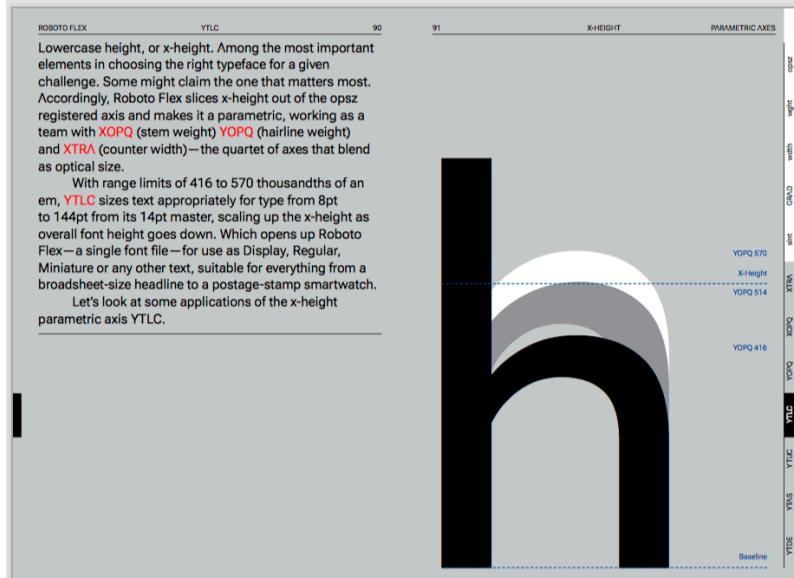
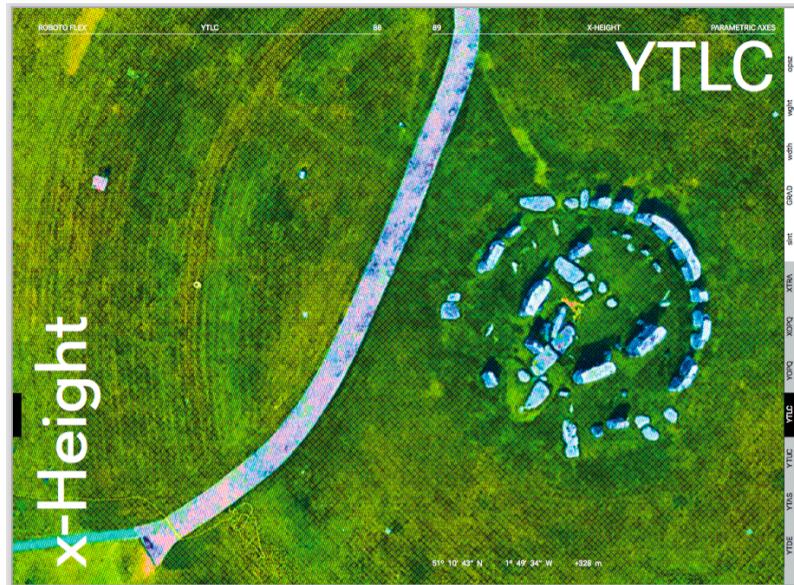
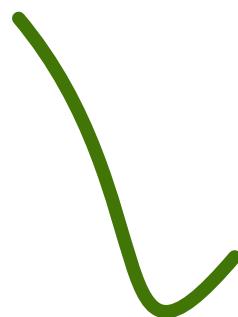
lower limit of 1px and no lower. At large sizes its maxima keeps **YOPQ** consistent with **XOPQ**, keeping the text readable down to 8pt, thickening hairlines down to a practical

135 either side of its 79 default at larger sizes shows how varied the space remains—the lower limit of 1px acts as a safety stop that prevents hairlines vanishing. The **YOPQ** axis continues affecting optical size down to 8pt.

VIDE YNS YTC YLC YRS YTR XTRA XOPQ YOPQ

# Roboto Flex Specimen Book Review

Wax Studios Google Roboto Book 061020.pdf



# Roboto Flex Specimen Book Review

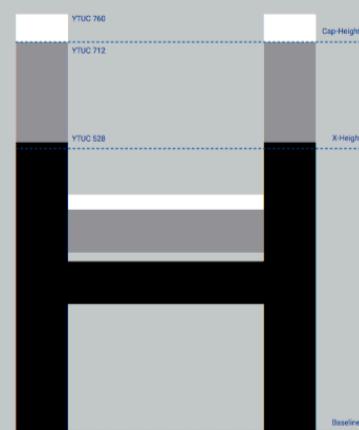
Wax Studios Google Roboto Book 061020.pdf



ROBOTO FLEX YTUC 95 CAP HEIGHT PARAMETRIC AXES  
Effective typography is about balance. Parametric axes give you fine control over the levers that control individual qualities, down to single thousandths of an em. Blend them with other axes, and you can slide those qualities up or down as one, like chords in a concerto. Some are subtle, some more distinct. Among the distinct in Roboto Flex is **YTUC**, or uppercase height.

Uppercase height is different to point size. Adjusting this axis doesn't alter width, weight or counter spacing. It simply makes the character taller, in a range from 528 to 760 miles of an em from a 712 default. That means more room below the default in Roboto Flex's design-variable space, to work smarter with lowercase axes like x-height. Let's get close to YTUC.

ROBOTO FLEX YTUC 96 CAP HEIGHT PARAMETRIC AXES  
The range of the YTUC was in Robot Flex is 528 to 760 miles (thousandths of an em), taking the type from a large percentage of the glyph down to barely more than x-height at default settings. YTUC offers some interesting opportunities for text color within the design space.



ROBOTO FLEX YTUC 98 CAP HEIGHT PARAMETRIC AXES  
16/14 BOUNDARY BREAKTHROUGH

YTUC 760 12/13 Is this a BREAKTHROUGH? Some kind of BOUNDARY? An EVENT HORIZON with EXIT ahead? More details emerge. Ideas and concepts easy as A, B, C or 1, 2, 3. Another move onward beckons. This is going to be BIG.

YTUC 528 12/13 Is this a BREAKTHROUGH? some kind of BOUNDARY? An EVENT HORIZON with EXIT ahead? More details emerge. Ideas and concepts easy as A, B, C or 1, 2, 3. Another move onward beckons. This is going to be BIG.



It looks like an unwanted line break.

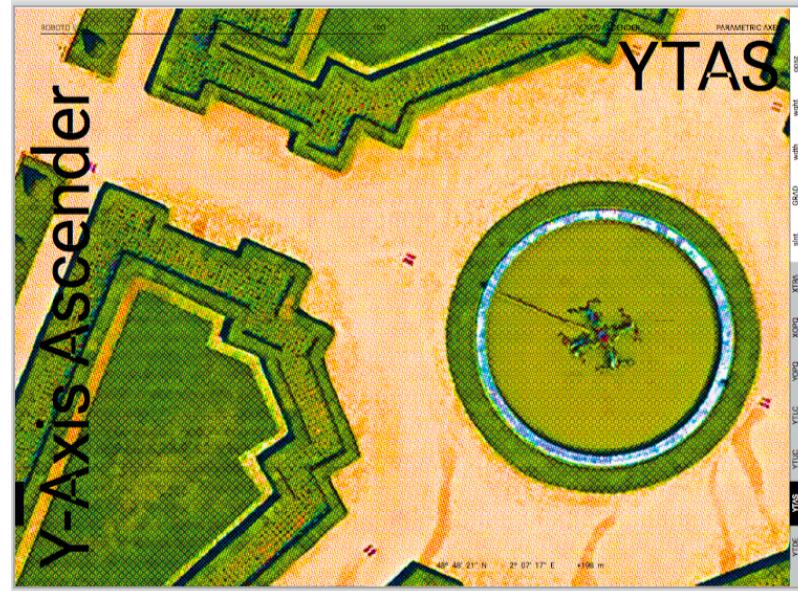
ROBOTO FLEX YTUC 99 CAP HEIGHT PARAMETRIC AXES  
16/14 BOUNDARY BREAKTHROUGH  
YTUC leads to a change in visual size. The range of YTUC is .550 to .850 of an em; the equivalent for lower case is .416 to .570. These extremes of the design space allow, for example, a quasi-uppercaser by minimizing YTUC while maximizing YTOS, or, by contrast, a hyper-uppercaser. The opposite. Between the extremes, of course, is a playground of potential. Diacritics in other languages look natural even in uppercase Roboto Flex, with marks that don't necessarily match the YTUC axis as height increases or decreases from the 14pt baseline. The result is a collection of rounded uppercase letters that are constant across all rounded letterforms, with many rounded

The same passage looks subtly different when adjusting YTUC alongside YTOS, resulting in a hyper-uppercaser. The opposite. Between the extremes, of course, is a playground of potential. Diacritics in other languages look natural even in uppercase Roboto Flex, with marks that don't necessarily match the YTUC axis as height increases or decreases from the 14pt baseline. The result is a collection of rounded uppercase letters that are constant across all rounded letterforms, with many rounded

Having horizontal curves at the top. Potted performances such as A and M match the shape of N, since they share a flat horizontal at the apex.

# Roboto Flex Specimen Book Review

# Wax Studios Google Roboto Book 061020.pdf

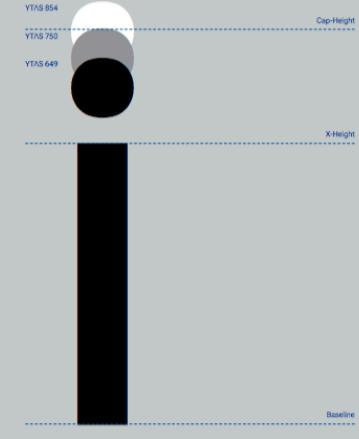


**ROBOTO FLEX** **YTAS**

Ascenders. One of the subtlest parts of typography, where a fractional adjustment makes for a sweeping transformation to the color of a page. That's why Roboto Flex pulls out lowercase ascender height as an individual axis, **YTAS**.

Scaled as .649 to .854 of an em and set by default

Scaled as .649 to .854 of an em and set by default at .750 on the 14pt master, it's not a huge range, but extending it upwards gives a classic airiness to l, f, and h when they start a word. While combining with the uppercase companion **YTUC** lets you play off cap and lower case height beautifully, even matching them precisely if that's the effect you want. All are points in the same design space. Let's go all the way up, with **YTAS**.



YT&S 675 10/13 View from the top: lovely forms giddily gliding by, strange concoctions of letters: libidibi, titikiki, everything on the up and up. A sense of hearing the conclusion.

YTAS 649	44 / 48	giddily gliding
YTAS 750	44 / 48	giddily gliding
YTAS 854	44 / 48	giddily gliding

YTAS 854 110 / 125

First at 649 dimensions, then at 854. Different values of **YTA** in Roboto Flex's design-variable space, without moving any other axis. Yet even at extremes, the typeface never loses its basic character. While Roboto Flex does not include a separate axis for line spacing—since so many applications do already—being able to control ascender (and descender) height offers much more control over the overall look.

of blunting into the line above.

The range of the YTAS parameter isn't huge, nor does it need to be. Constraining the dimensionality of this variable space keeps the limits of the font within reason – no lowercase letters form absurdly tall or short. Of course, the range is calibrated to play well with other elements of visual

in answer goes up visually of course, I ascended come close above. Right both escape give you deal with

When ascenders are distinct, the limit of line height, where both ascenders and descenders can be accommodated, is determined by the ascender Fleck's area for ascender and descender choices for it. Increasing ascender height creates interesting potential



# **Roboto Flex Specimen Book Review**

Wax Studios Google Roboto Book 061020.pdf