

Publishing variables

With

by Petr van Blokland

Scripting the many repeating details of a design project – like publishing this magazine – gives designers a chance to test the options without going through a lot of manual work.



Publishing variables with

Page Bot

by Petr van Blokland



What happens if designers learn to code?

A client calls for a thrilling project: The annual production of high-quality full-color travel guides in 40+ editions and 15+ languages for world-wide distribution. Saving on printing costs, only the black offset-plates change with each different language, so authors and translators get live feedback about the flow in such fixed layouts. And in addition, all pages are published online, getting their content from the same database.

Another client wants to create automated specimens for continuous testing during the design process of type, for *UFO*, *TTF*, *OTF* and the new *Variable Font* technology. And these specimens will also be used for marketing purposes and as templates for actual usage of the typeface.

Later in the year, the same client wants custom expansions to an existing typeface. Such a request includes not only the making of new drawings, but also new Open Type features, spacing and kerning. Not the kind of work type designers want to do entirely by hand. Yet, there is a big chance that standard type design applications do not fully support all the required functions for this project.

These are but a few examples where design and production cannot be done either manually or by

The **Droste Effect** (Dutch origin), known in art as "mise en abyme," is the sively appearing within itself. Recursion is an important tool in parametric design processes.

existing applications. Either the volume is too large, the design cycles too short or rules too complex for a single designer to finish in time. Or—due to the lack of freedom—the job is simply too boring to be a challenge.

What is design anyway?

Design as verb describes the process of searching for solutions that best match a given set of requirements, such as medium, purpose, usage and types of users. All within the given amount of time. At the start of a project for automated specimens, it is not clear what the best approach is: How do you predict the flow of text in columns with different widths and sizes? How do you visualize the number of scripts and languages that a typeface supports? The designer finds solutions to these questions by trial and error, step by step, sketching, prototyping, rejecting what doesn't fit, keeping what works and developing the rules along the way.

Once useful directions are discovered, they must be written out as rules for others to be useful. In plain English, an Art Director can explain some ideas for a series of images to a photographer, who then needs to interpret the ambiguous description into a sequence of actions to make the photos.

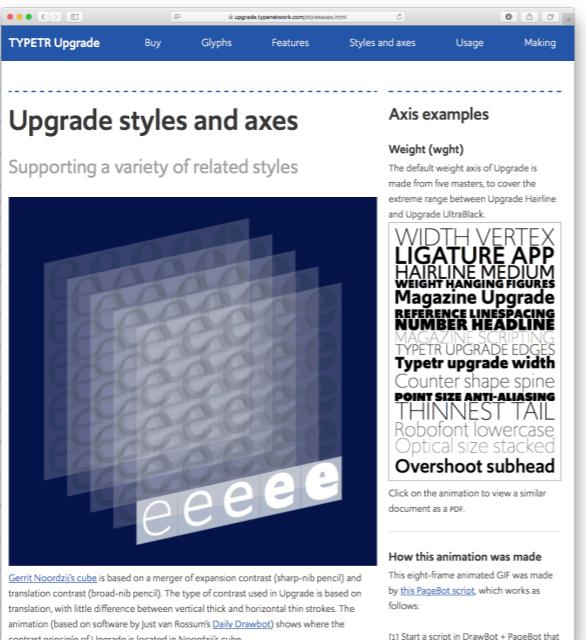
In a stricter context, rules are used by web browsers, composing the layout of a page to fit the limited size of mobile screens. Then, simply interpreting the English language is not accurate enough—the rules must be translated into strict executable instructions.

In any case, the rules are likely to contain phrases such as “For every photo you have, try to combine them in a frame. If that does not work, either scale some of them, hide them or move them to another page.”

Writing such rules in a structured language—such as Python or JavaScript—is called *Coding*. Designers unfamiliar with writing code may think designing and coding are polar opposites, but in reality, designers write rules all the time. Designers create loops (doing something for as long as needed) and conditions (doing something under the right circumstances) constantly, just with a different language.

How does that apply to type design?

Every design process breaks into parts that are mostly creative and parts that are mostly procedural. There is a difference between initial sketches—where the choices for contrast and serifs are made—and the production phase, where consistency in



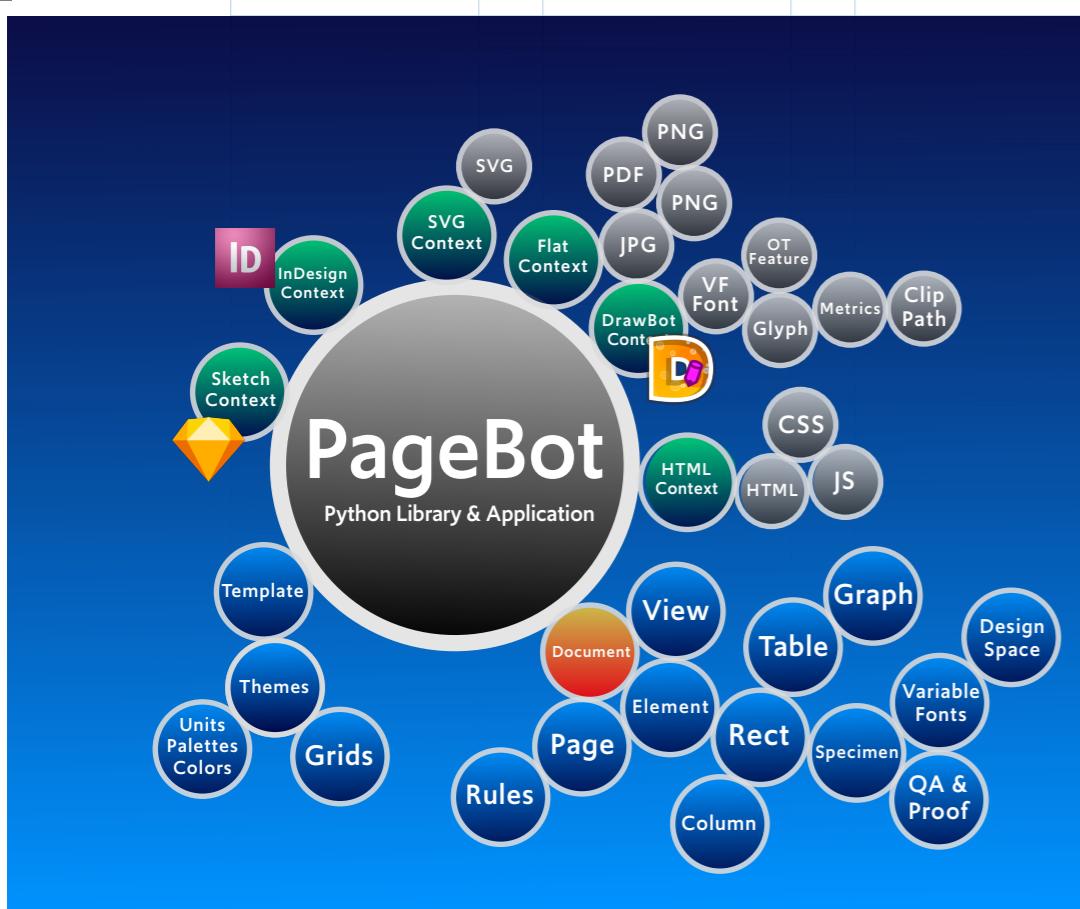
TYPETR Upgrade website

In the ongoing process of design, testing and presenting type, it is important to use the actual media that the typeface is intended for. The design part is figuring out how to write the code that matches the sketch.

The PageBot Library

PageBot is an MIT Open Source Python library for automated creations of publications. It can read and write many file formats, such as databases, images, animations and fonts. This allows for accurate parametric design, depending on platform and context, as type in print behaves very differently from type displayed on screen. The structure of typographic content for dynamic media—like websites—needs to be different from column layouts for fixed paper sizes.

Yet, it should not be the author or the designer to solve this problem over and over again. Eventually, their tools should contain all that knowledge.



weight and width is important. By definition, design applications can only contain functions that are mainstream. The variety of functions is so widespread and the potential market so small, type licenses won’t pay for it.

To put the reasoning in reverse: If a function is interesting enough to be implemented, it also becomes available for others to use. Just knowing how to operate the application is no longer enough to make a living.

What is the solution?

Designers can counter these kinds of developments by mastering the use of the tools better than anybody else. *By using them more frequently and differently from others. By understanding them inside and out, learning all their flaws and limitations. And most importantly, by knowing when they should not be used.*

Designers (graphic designers, typographers, type designers alike) need tools that allow them to automate repetitive tasks during their design cycles. For example, if a type designer needs to expand a handful of sketches into a complete type family, drawing all characters by hand, testing the result on proof-pages, going back, adjusting the originals, then manually expanding, the process will take too long. Sometimes a little chunk of code can speed up a very specific task, which exists for only a moment. For instance, while designing the *TYPETR Bitcount*, a vast number of pixels had to be aligned to the grid, different for roman and italic. That problem never came up again.

Scripts that automate such processes often span only a couple of lines, where they can save days or even weeks of work.

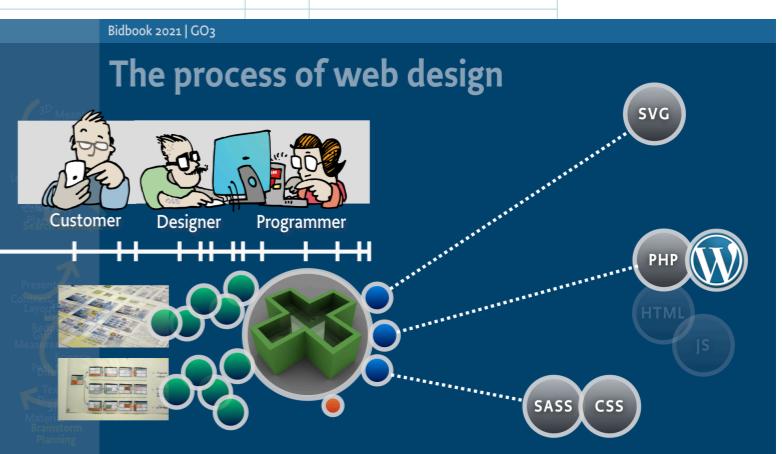
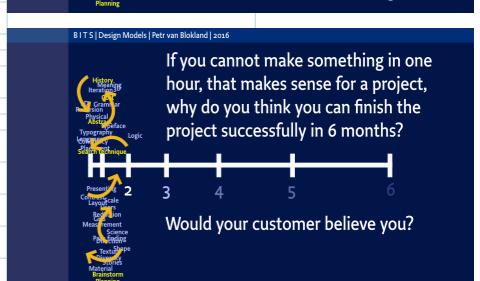
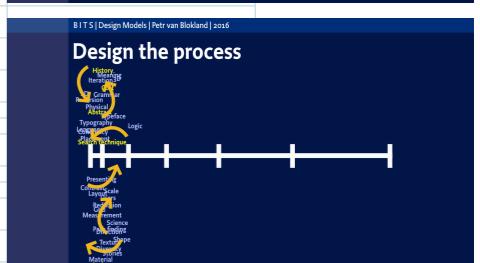
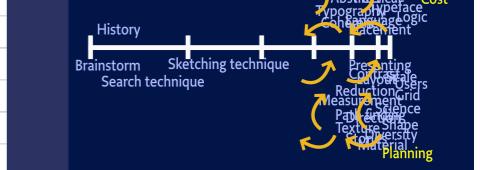
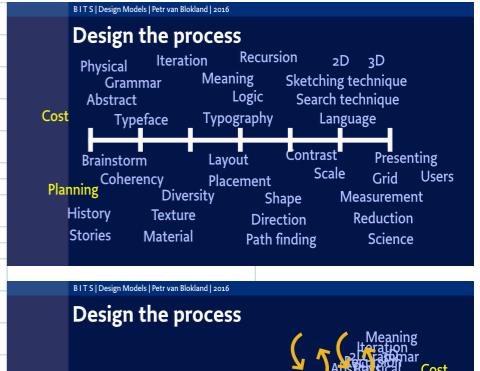
So designers should develop their own?

It is not realistic to expect that every designer will code their tools on the same level as professional programmers. Technology changes so fast that even professionals need to stay focused and select their specialization carefully. That is true for the “*big applications*.” But, there is an area where designers can surely benefit by writing their own code.

In web design, where the ability for designers to read and write CSS is an enormous advantage compared to those who send their pixel images off for programmers to convert into code. Web designers who think along the lines of responsive pages and conditional content have better control over the process than their colleagues who stick to designing websites in InDesign. Designers may not be writing the best possible code in the world when they are working on prototypes, but working in the same medium allows programmers to take over later in the process and finish the job. The same thing hap-

Doing and doing again
The common view of the process of design is to approach it as a linear operation. Research first, then a brief sketching phase, then production—all ending with a final presentation, barely making the deadline. This is the same, no matter if the whole process took 6 hours, 6 days, 6 weeks or a couple of months. In reality most of the work is done at the very end of a project, with little time left for enhanced views and experiments. Instead, doing a full circle of sketch-prototype-sketch, many design issues become clear earlier. If a designer cannot envision the entire process in a couple of hours, what could change later? Scripting tools like PageBot, help to generate prototypes and simulate the production process in an early stage, leaving space for the designer to iterate over new cycles.

Designing for the web
Similarly, when applied to web design, the use of rapid prototyping in a team is very effective to share knowledge and experience.



DesignDesign.Space

Which studies can we offer you in 2019?

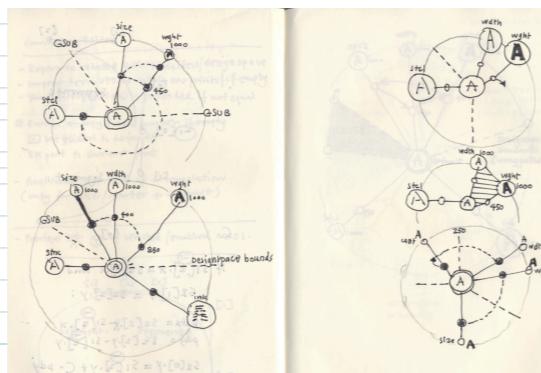
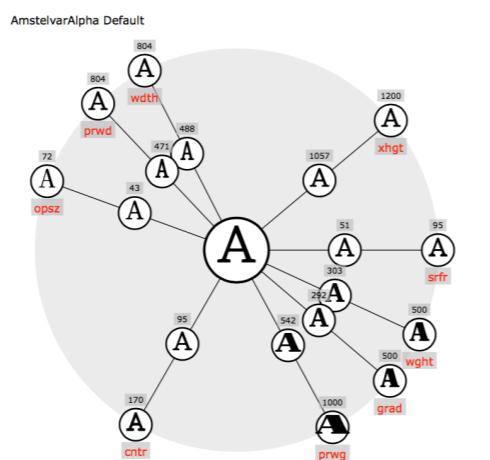
Discovering the basics of type design? Improving your sketching techniques? Starting a studio? Programming pages? Learning to code type specimens? Mastering typography? So are we.

It all happens in DesignDesign.Space studies

DesignDesign.Space as your coach

Coding as a design tool

For many designers, learning to program and code is often not an obvious choice. Thinking in parameters is assumed to limit their ideas and it's seen as something that programmers ("developers") do after the design is finished. However, automating the creation of prototypes early in the process gives much feed-



Swiss army knife
PageBot functions in many areas, from generating websites (top left), visualization of the design spaces for Variable Fonts (below),

TextCenter - Upgrade (18 styles)

Style: Upgrade-Semibold.ufo
Sample: Jilis kerning
TaTbTdTeTfTgThTiTjTkTlTmT
TnToTpTqTrTsTtTuTvTwTxTyTzT
TœTæTisTfiTflVaVbVcVdVeVfVg
aVbVivVivkVlVmVnVpVqVrVs

Values: 100 PPEM [ex], 120 % Leading, 4 Cursor, 10000 Glyphs per

Edit Mode: Spacing [s] Kerning [k]
Flags: Kerning [q], Kern markers [w], Metrics [m], Missing [?], Repeat [r], Color Scheme: Regular, Dia, Color, Scripts, Export

Zoo Gym Any XML
Mer Klem Jet him new Who
Bit Put Tan Cnt
See Our Typ Him
Ufo Nét See Pet

TYPETR Bitcount
Revived from a late 70's project to design the smallest possible matrix-letter (5x7), Bitcount is a good example where scripting helps to design and visualize the vast amount of variations that need to be reviewed in a Variable Font.

Scripted specimens
The Bitcount poster (left), the layered animations (right) and the proofing interpolating Variable Font axes (bottom) are all examples where scripting helped the designer through the design process by quick visual feedback on the various possible options.

pens with type design applications. In the past, websites were programmed and coded from scratch, costing thousands of dollars. With the availability of Open Source libraries such as *node.js*, *jquery* and *d3*, it has become possible to create complex websites with only a few lines of code.

The best of both worlds is to be standing on the shoulders of so many others, tapping into the scripting capabilities of existing applications.

Back to type design...

Where CSS and JavaScript are the obvious languages to define the rules, for type design the main programming language is Python. It writes scripts, extensions and plugins in all major type design applications.

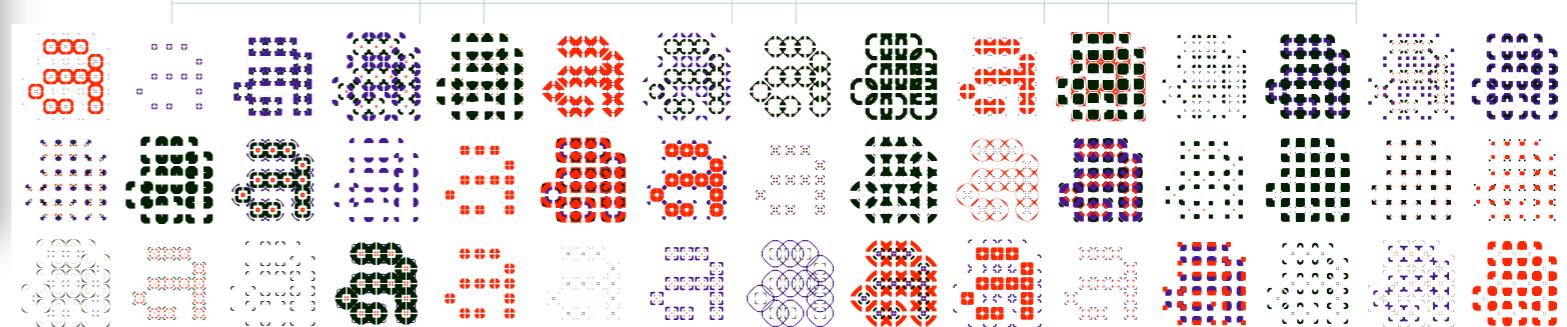
Many of these are available as Open Source on GitHub, supporting functions in the type design process that can be automated. Much more information can be found in the manuals on these subjects—there is too much to address in a single article.

The emphasis of this article is on another aspect besides the type design process itself: The education of design and the automated production of publications, such as the creation of type specimens for testing and presentation purposes.

Starting with DrawBot

Where many type design tools focus on the drawing process, not so much as been done for testing and the making of specimens.

In order to give better feedback to students when teaching them how to write their designs into code, *Just van Rossum* developed *DrawBot*, now maintained by *RoboFont* developer *Frederik Berlaen* as Open Source application *drawbot.com*, implementing a Python graphics library for designers. *DrawBot* is perfect for visualizing the working of algorithms, to experiment with Python code and to automate relatively simple tasks that otherwise would take too much time in the sketching process. The vast amount of export formats, ranging from PDF, PNG, JPG to animated GIF and movie formats makes





Digital letterpress
Revival of the 1967 letter-
press type specimen, by
the former Dutch book
printer "Koninklijke
Drukkerij Van de Garde,
Zaltbommel," designed by
Huib van Krimpen. Recre-
ated by PageBot.

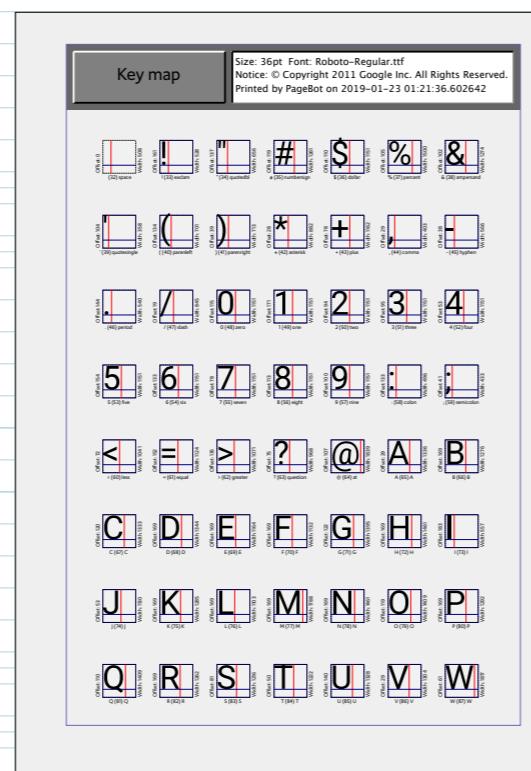
DrawBot an ideal tool for generative design and animations.

As easy as *DrawBot* is to start coding, the making of type specimens, magazines and websites needs more resident knowledge.

...then adding PageBot

To bridge that gap, Open Source *PageBot* offers an increasing amount of built-in knowledge, spanning several design disciplines. Since it can query font files to match proportions of letters with measures in typography, and since the dimensions of Variable Fonts are fully supported, all measures in a page interact and are covered by the same rules. *PageBot* not only offers scripting to make type specimens (ranging from the great designs of the past to new styles of today), it also is a solution to *Single Source Publishing*.

For example, the pages of this article, as well as several others in this magazine, were automatically composed and generated by *PageBot*, reading the article text from a *MarkDown* file, and then using Python instructions to compose the pages as *PDF*.



Römer

Scripting the past
Similar to the famous Berthold type specimen (right) and the American Type Foundry specimen from 1926 (above), existing specifications and presentations can be described parametric and put back into service. And meanwhile supporting modern requirements, such as Variable Fonts and responsive layout.

documents. However, other outputs, such as animations and websites, can be generated with the same content and the same scripts. *PageBot* will be under development for a while, with spin-offs in many directions, from specimen generators to website builders. Stay tuned.

TYPETR Collection

UPGRADE VAR REGULAR