

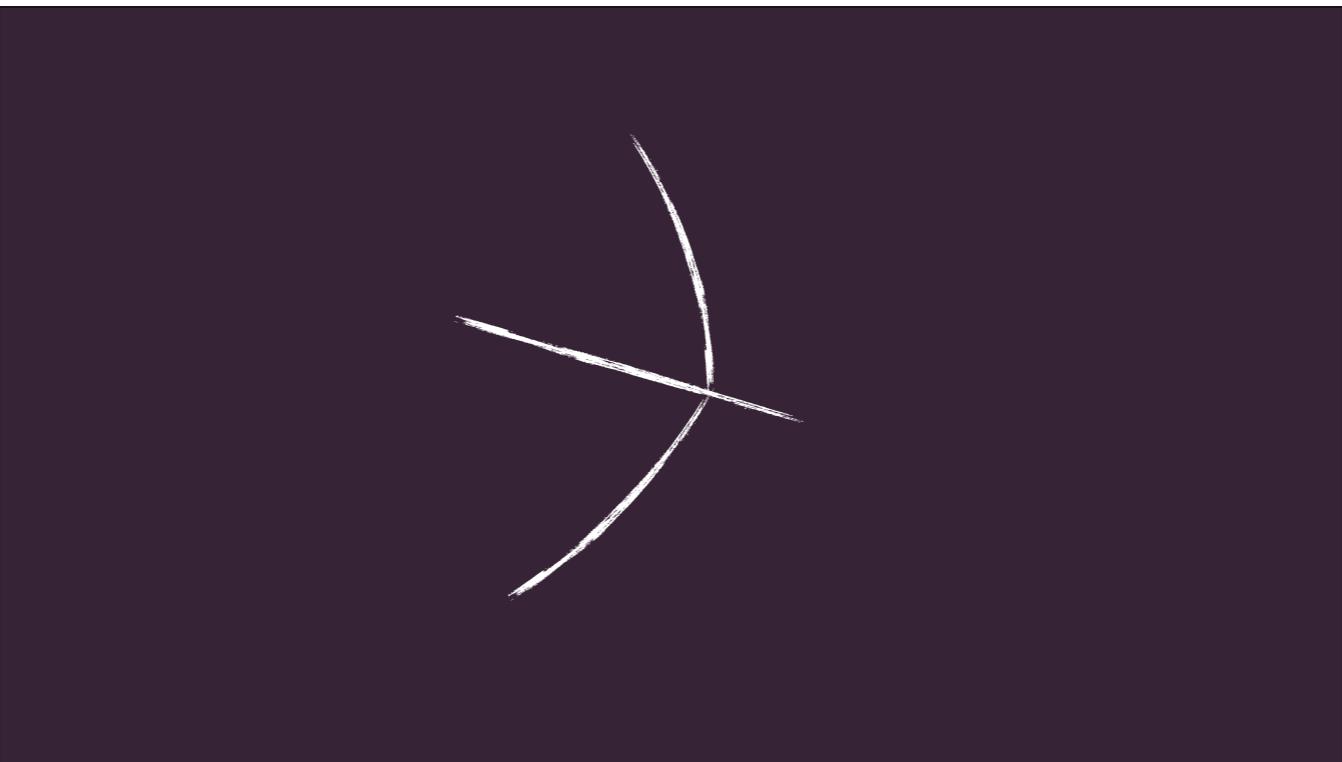


Designing Systems

Theory, Practice, and the Unfortunate In-between

@paulrobertlloyd

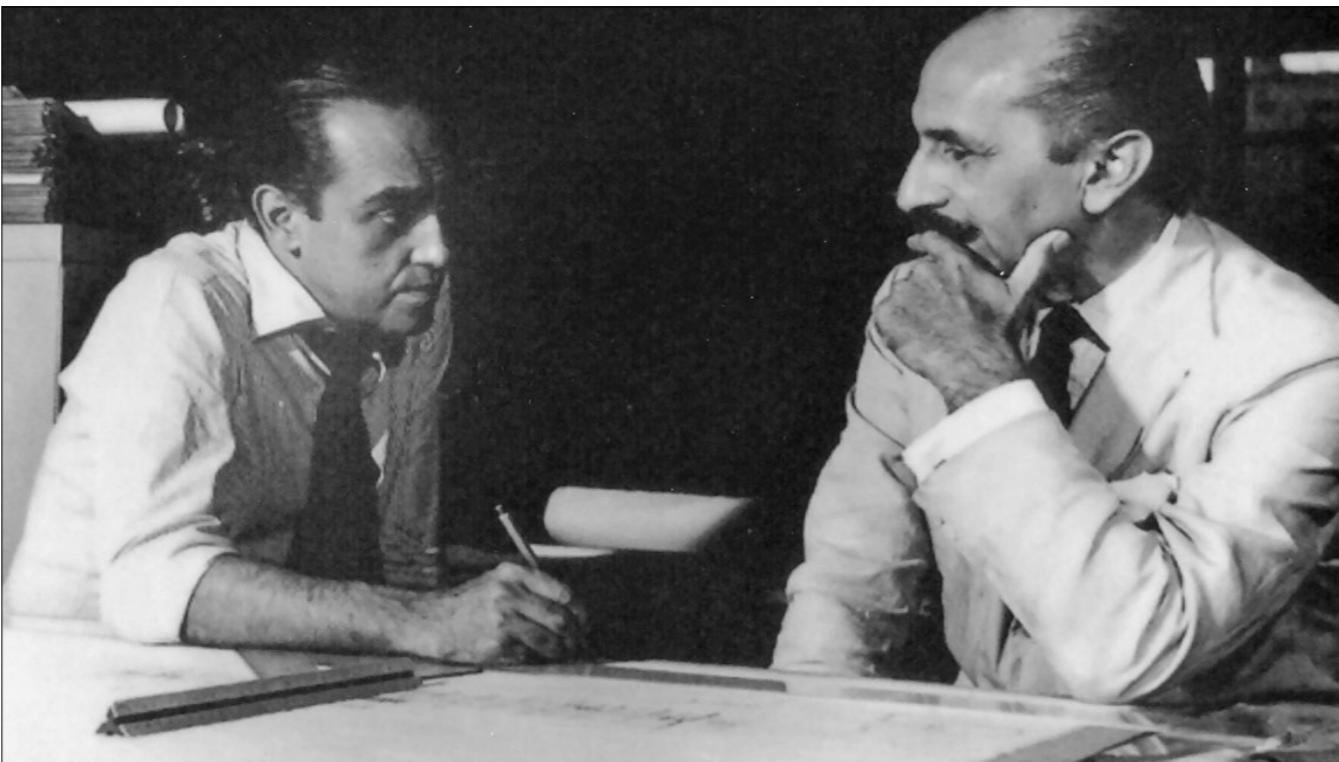
Smashing Conference Freiburg / 13 September 2016



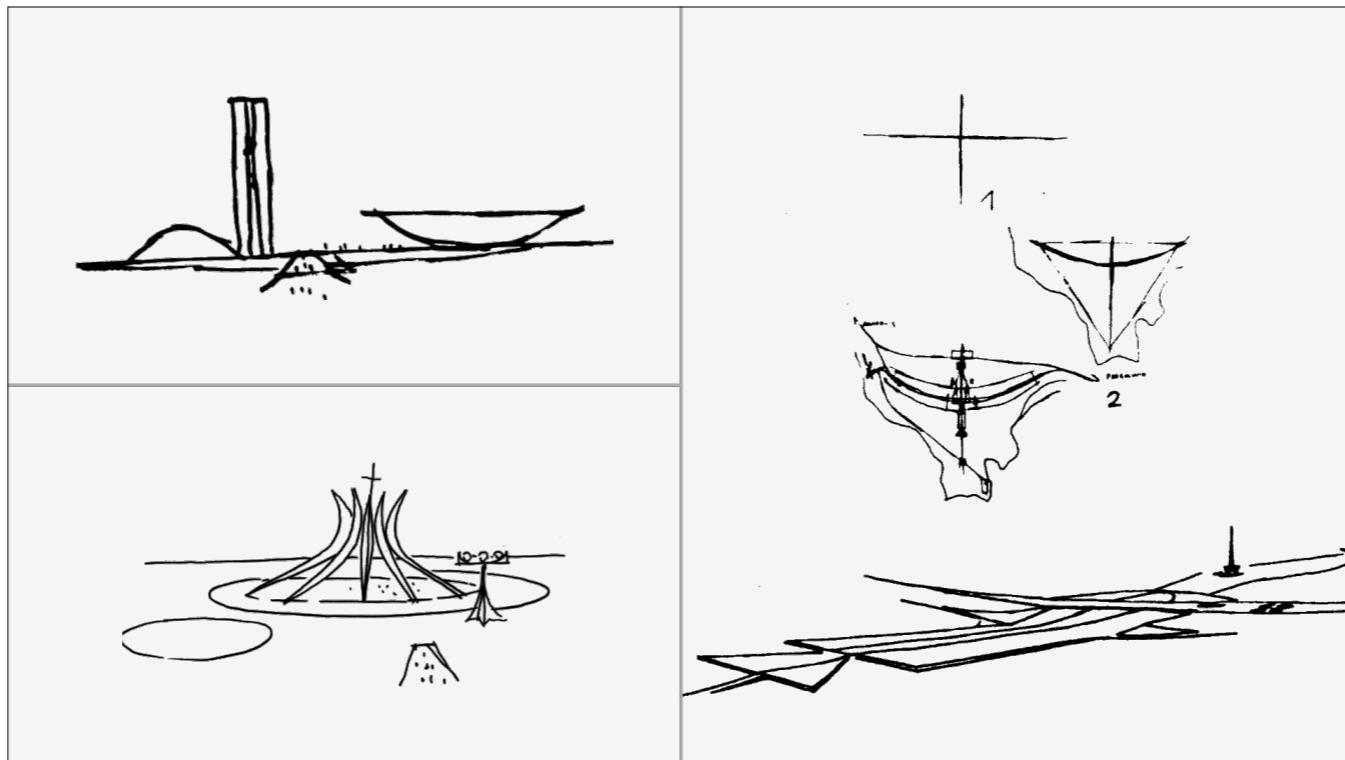
- I want you to imagine you were designing a city.
- What shape would it be. How might you layout its roads and major features?
- How about one shaped like a bird?



- Such a city exists, it is the capital of Brazil, and it is called Brasília.
- You might have thought Brazil's capital was in Rio de Janeiro and indeed it was until 1960.
- But as a coastal city, it was seen as vulnerable to attack, and the country's constitution had long stated that its capital should move to a more central location.
- It wasn't until Juscelino Kubitschek was elected president in 1956 however, that its construction would be ordered.
- Kubitschek had campaigned on a promise of "fifty years of prosperity in five", and the creation of Brasília a key pledge.



- Kubitschek invited the architect Oscar Niemeyer (on the left here) to design the new capital's civic buildings, such as the Presidential Palace, National Congress, Supreme Court and its Cathedral.
- Lúcio Costa (shown on the right) won the contest to arrange its urban plan, and is the person responsible for that bird like layout.
- Niemeyer and Costa had worked together on a number of projects previously.
- Both were admirers of Le Corbusier, the Swiss-French architect and pioneer of modern urbanism.
- Le Corbusier believed good architecture could lead to good societies.
- He advocated laying out cities formally and within rigid zones, and promoted a series of standardised housing types that he termed "machines for living in".
- Yet until the creation of Brasília, his ideas had never really been tested, and certainly not on the scale of an entire city.



- Tasked with designing a new capital, and with Brazil's vast and unpopulated interior as their canvas, Costa and Niemeyer could finally turn these ideas into practice.
- But, beyond acting as a symbol for a braver, more optimistic age, they also believed their design could transform Brazil's heavily stratified society into a more egalitarian one.
- In this model utopia, governors and ambassadors would live next door to janitors and labourers, and everyone would use the same entrances and share the same spaces.
- Yet Costa's plan contained a fundamental flaw: its fixed layout limited the amount of residential space available.



Photograph: Marcel Gautherot

- As people from across the country moved to the city in search for work, its population grew rapidly, and Brazil's deep class structures, subtly racial in origin, soon reasserted themselves.
- The workers who had helped build the city were soon banished from it.
- After a long fight they were allowed to remain — but only in distant satellite towns, where they remained segregated from the middle-classes, spending their meagre salaries on long commutes into and out of the city.
- Those servants that remained in the city, lived in apartments far worse than they might have done otherwise.
- **Reality had spoilt utopia.**
- Reflecting on their plan, the art critic Robert Hughes said...



The Shock of the New: Trouble in Utopia
BBC Television, 1980

- “Nothing dates faster than people’s fantasies about the future. This is what you get when perfectly decent, intelligent and talented men start thinking in terms of space, rather than place, and about single rather than multiple meanings. It’s what you get when you design for political aspirations and not real human needs. You get miles of jerry-built platonic nowhere infested with Volkswagens. This, one may reverently hope, is the last experiment of its kind. The utopian buck, stops here.”



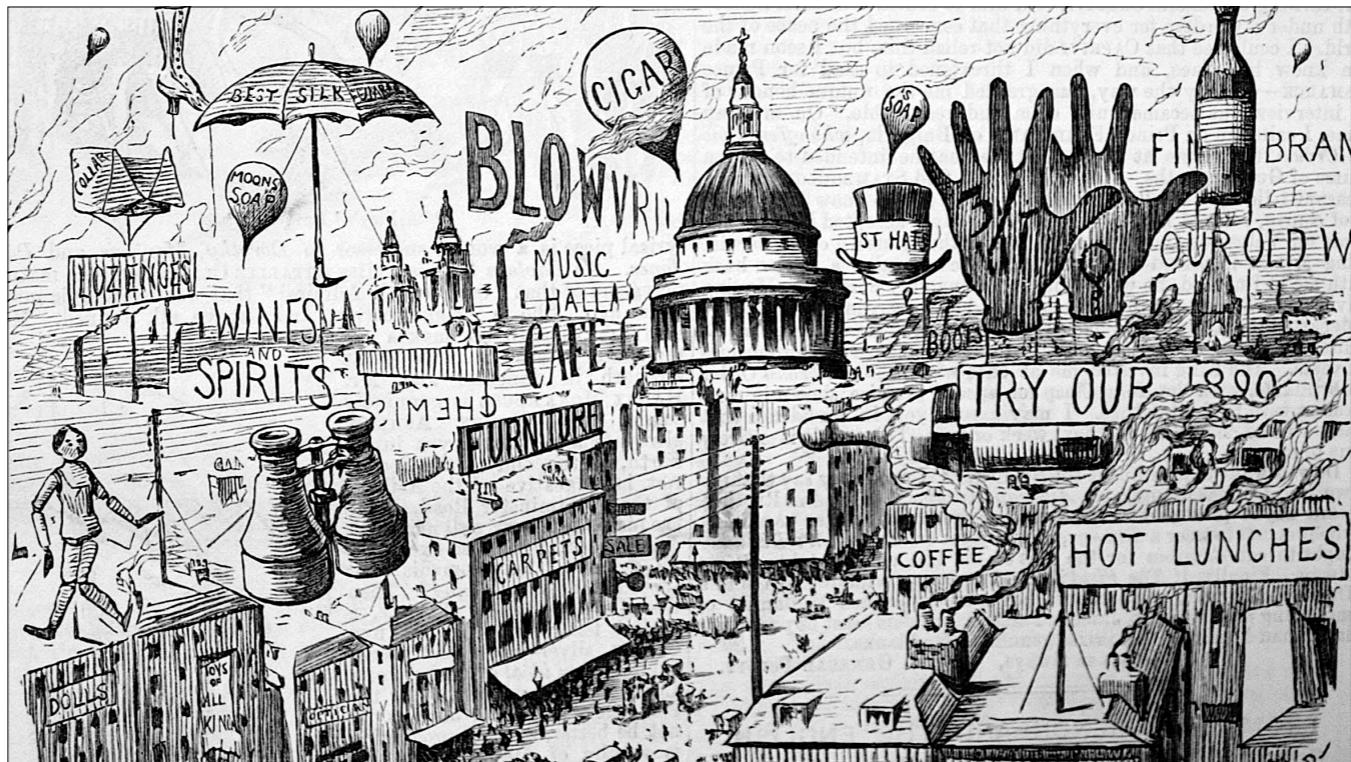
- While the barren central plateau of Brazil were a canvas for Costa's ego, the same might be said of Sir Christopher Wren's plans for London.
- Ravished by the great fire in 1666, he envisioned the creation of a beautiful new Baroque city, with wide sweeping avenues, grand Piazzas, and a street plan not too dissimilar to that of Paris.
- Yet, London is a peculiar, somewhat unruly beast, and not even the great Sir Christopher Wren could tame it.
- Before he even had a chance to enact his plans, residents had begun rebuilding their properties, and followed much of the old medieval layout as they did so.
- However, if anyone could claim to have bought order to London, it would be that of a quiet, shy and retiring solicitor from Lancashire, called Frank Pick.



The future of London cannot be an accident like the past. If it is to hold together, to remain a workable, manageable unit, it must now be planned, be designed, be organised.

— Frank Pick, 1926

- When Pick moved to London in 1906, the capital was home to over six million people, and its chaotic transport infrastructure was run by a number of separate companies.
- Pick had been employed by one of these, the Underground Electric Railways of London (UERL).
- At great difficulty and expense, they had recently constructed three new tube lines, and also electrified the original steam powered underground District Railway.
- The only problem was, very few people were willing to use these new services, and so Pick was put in charge of their publicity.

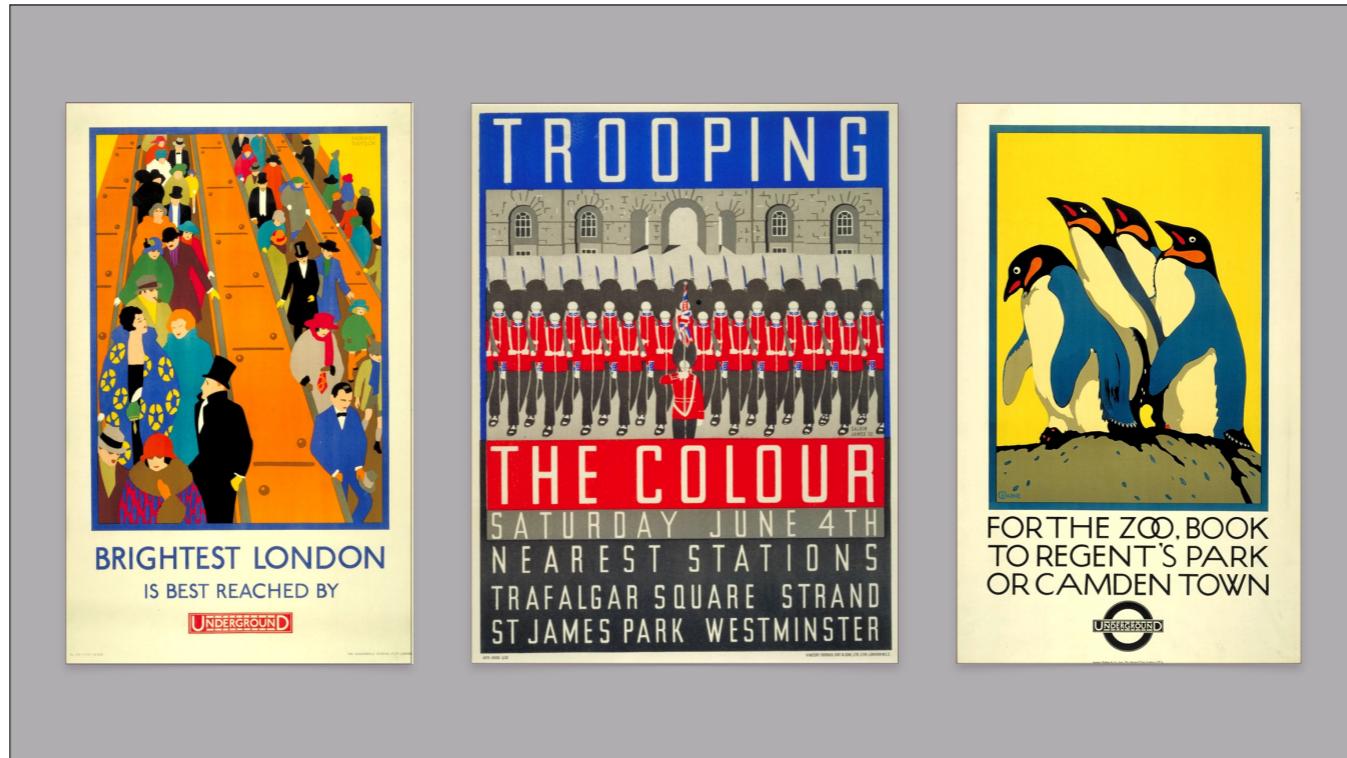


- At this time, you might call the design aesthetic eclectic, if you were being generous.
- This is an image from the sartorial magazine Punch in 1890, which may give you some idea how people perceived visual communication at the time.
- London's transport was not immune to this.
- On underground platforms, signage and advertising would appear side by side, arranged in a hotchpotch sort of manner, with a variety of different typefaces and styles.
- Not only was this uninviting, but it made the system harder to navigate, and hardly inspired confidence in this new mode of transport.



Photograph: London Transport Museum

- Unlike Costa or Niemeyer, Pick wasn't a designer or artist, so judged what worked through trial and error.
- His first instinct was to improve the appearance of stations. He achieved this by standardising the different poster sizes, limiting their number, and arranging them on a grid.
- Anticipating an increase in passengers visiting the city for the 1908 Olympic Games — and to ensure stations names would stand out — he placed a large red circles behind each name board.
- This would later inform the design of the now familiar underground logo, known as the Roundel.

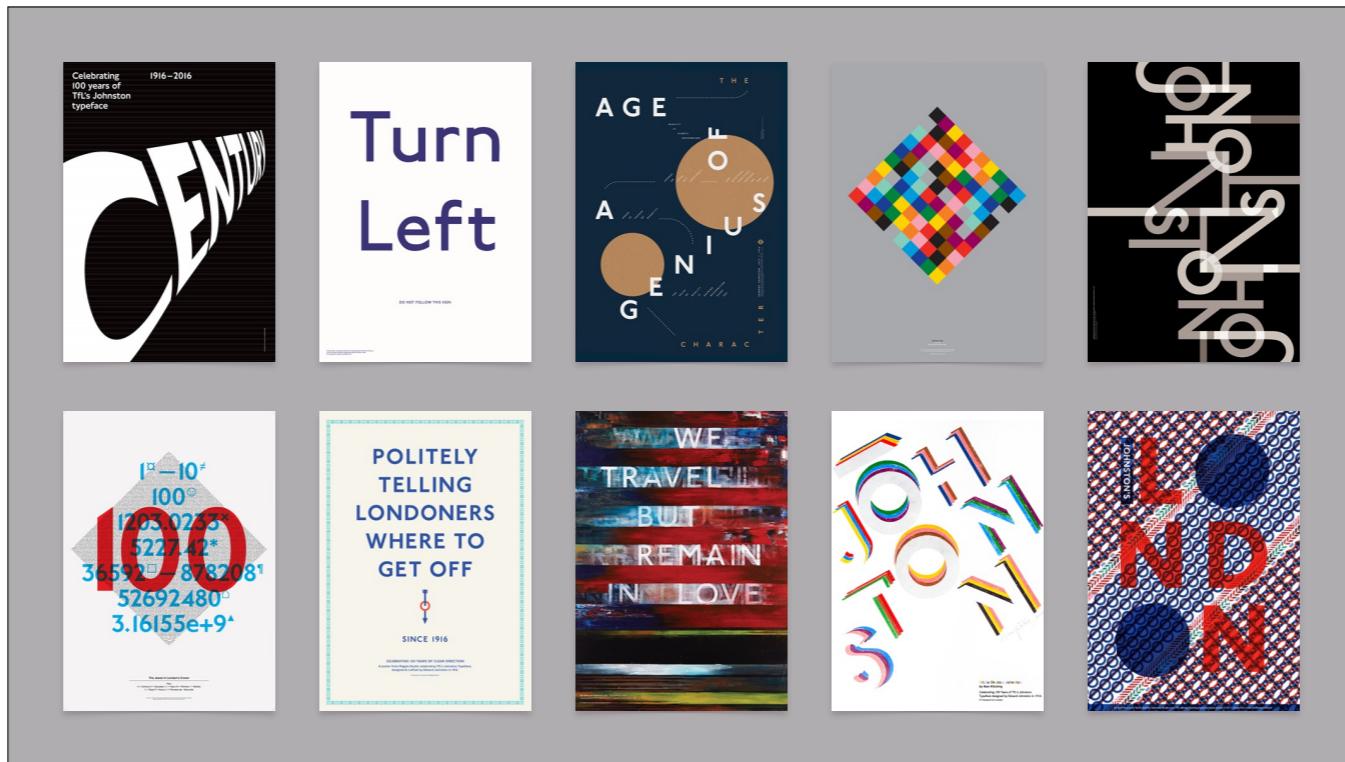


- Stating that “art has to come down off its pedestal and earn its living”, Pick then commissioned artists to design beautiful posters for the company.
- Rather than nag at passengers, these posters would instead inspire them, advertising different places to visit and attractions they could attend across London.
- In a certain way, these posters helped to humanise the network, and before long, people would refer to the tube as the longest art gallery in London.



Photograph: Alexander Baxevanis

- Pick also wanted a proprietary typeface to distinguish the Underground's publicity and announcements from that of other advertising.
- So he commissioned Edward Johnston to design this new font, with a brief stating that:
 - this should "have the bold simplicity of the authentic lettering of the finest periods"
 - and belong "unmistakably to the twentieth century".
- Johnston's typeface was inspired by the simple, hand drawn letters he had spotted on the side of a tradesmen's wagon, and its introduction proved to be a dramatic change.
- Some of Johnston's contemporaries derided his creation, calling it a betrayal of well-designed lettering.



- Yet his typeface survives to this day, and this year Transport for London is celebrating the 100th anniversary of its introduction.
- One of the ways they decided to celebrate, was by commissioning 10 artists to each design a poster celebrating the typeface, which seems rather fitting.



- As Pick rose up through the ranks, his influence grew. Before long, he wasn't just commissioning posters and typefaces, but buildings too.
- In 1927, he commissioned the architect Charles Holden to design a new head office, 55 Broadway, as well as a series of iconic stations on the extended Piccadilly line.
- By 1933 Pick had become the chief executive of a new combined public body that would be responsible for all public transport across London and its suburbs, and remained in charge until shortly before his death, in 1941.
- During his 30 years managing London's transport, Frank Pick had instigated what today we would call a corporate identity programme — dare I say, design system — one that in some respects is still being used today.



The test of the goodness of a thing is its **fitness for use**. If it fails on this first test, no amount of ornamentation will make it any better; it will only make it more expensive, more foolish.

— Frank Pick, 1916

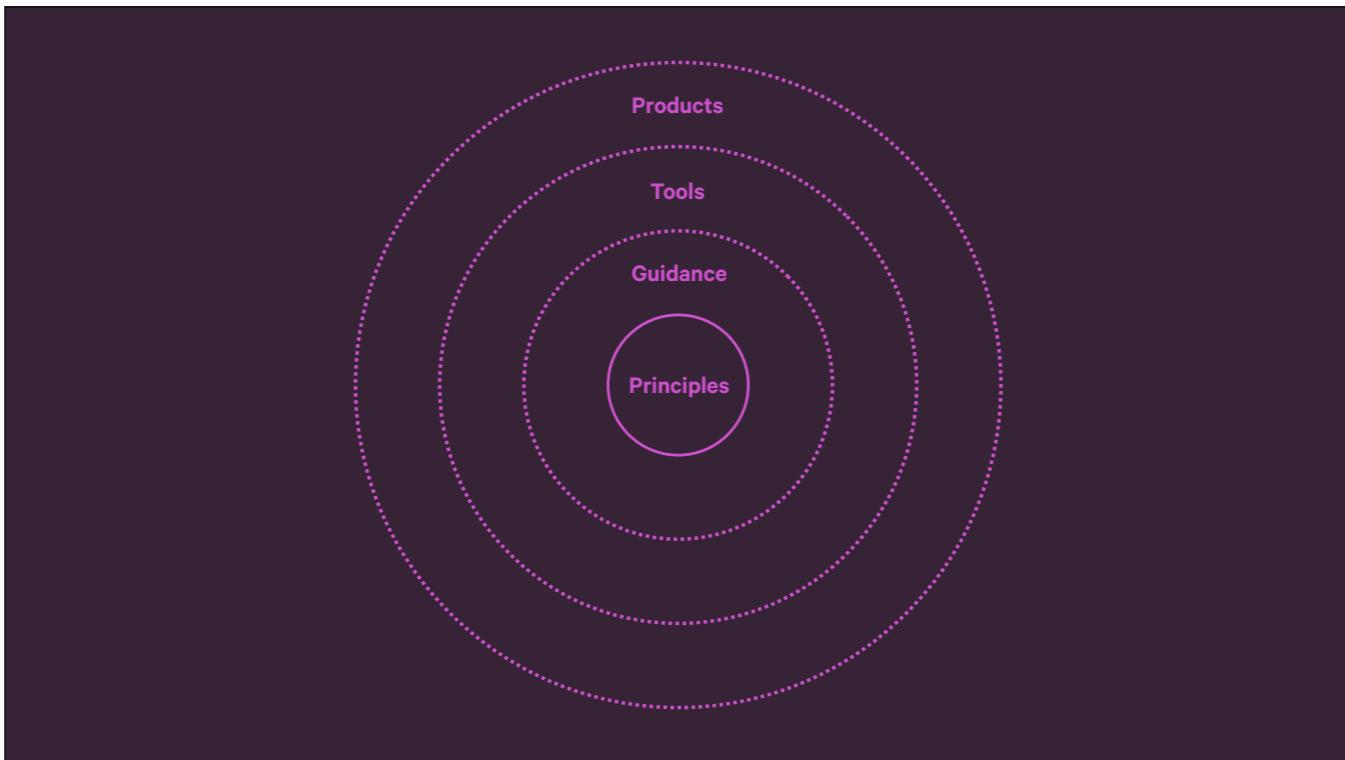
- In my mind, Frank Pick was the Steve Jobs of his time.
- Like Jobs was driven by insanely great product experiences...
- ...Pick was driven by a single over-arching principle: fitness for use.
- Be it the posters he commissioned Johnston's typeface or Holden's architecture, all needed to satisfy this principle.
- Anything crafted should express an honest and practical usefulness beyond mere aesthetic appearance.
- Furthermore, Pick sought to learn what worked using an evidence-based approach, one which placed passengers at the centre of his plans.



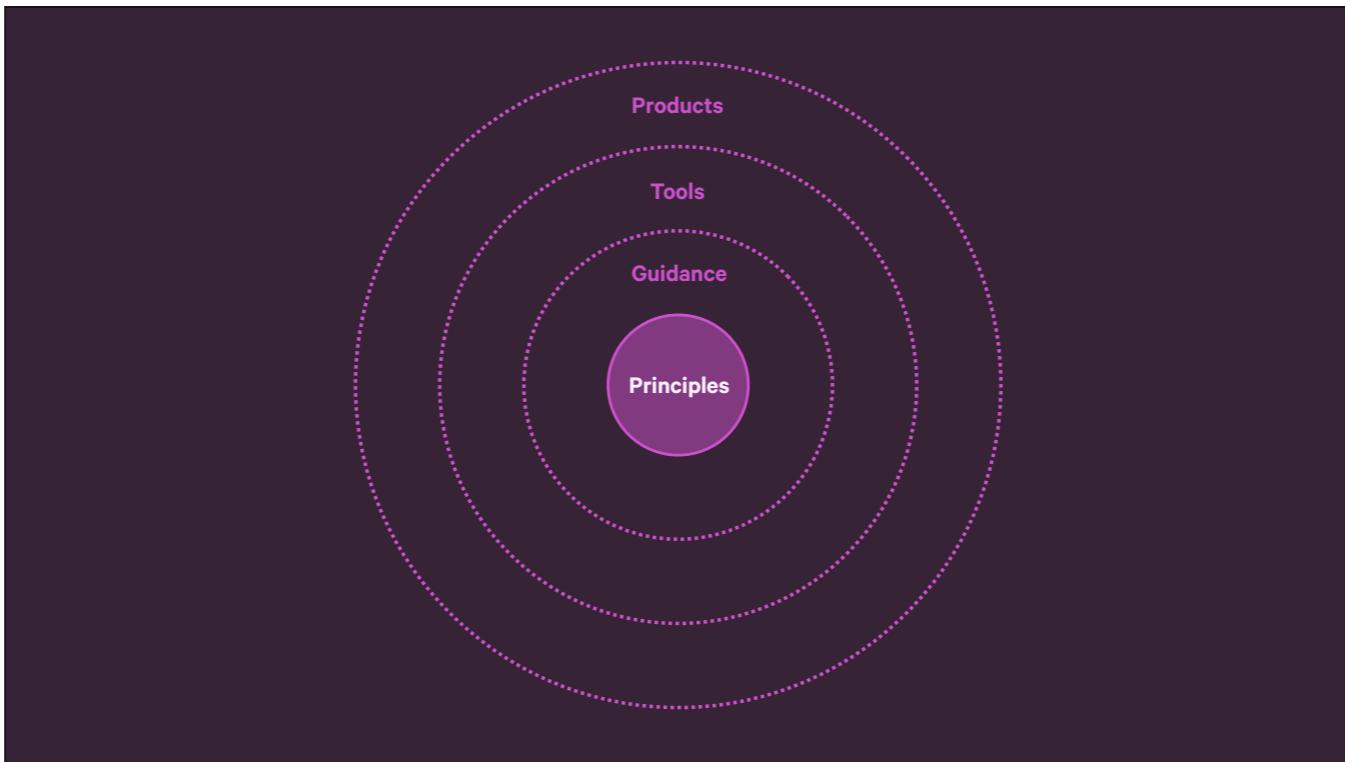
- Much like London, the web is also a chaotic and somewhat unruly place.
- Douglas Crockford called the web “the most hostile software development environment ever imagined”.
- To avoid building digital Brasílias, I suggest we heed the example set by Frank Pick and bring order to this chaos by grounding our decisions within the context of use.

What is a design system?

- What do we actually mean when we say ‘design system’?
- Over the last few months, this has become my favourite question to ask people.
- In discussing this with friends and colleagues, no one was able to give me a definitive answer.



- Today, I want to suggest that we consider a design system in its broadest sense, as an holistic collection of guidelines, tools and other artefacts, that together can help an organisation deliver coherent and consistent products and experiences.
- Also, that we imagine these systems being constituted of a series of layers, each concerned with a particular role and purpose.



- And much like Frank Pick's test for fitness of use, **these layers should radiate** from a core set of design principles.

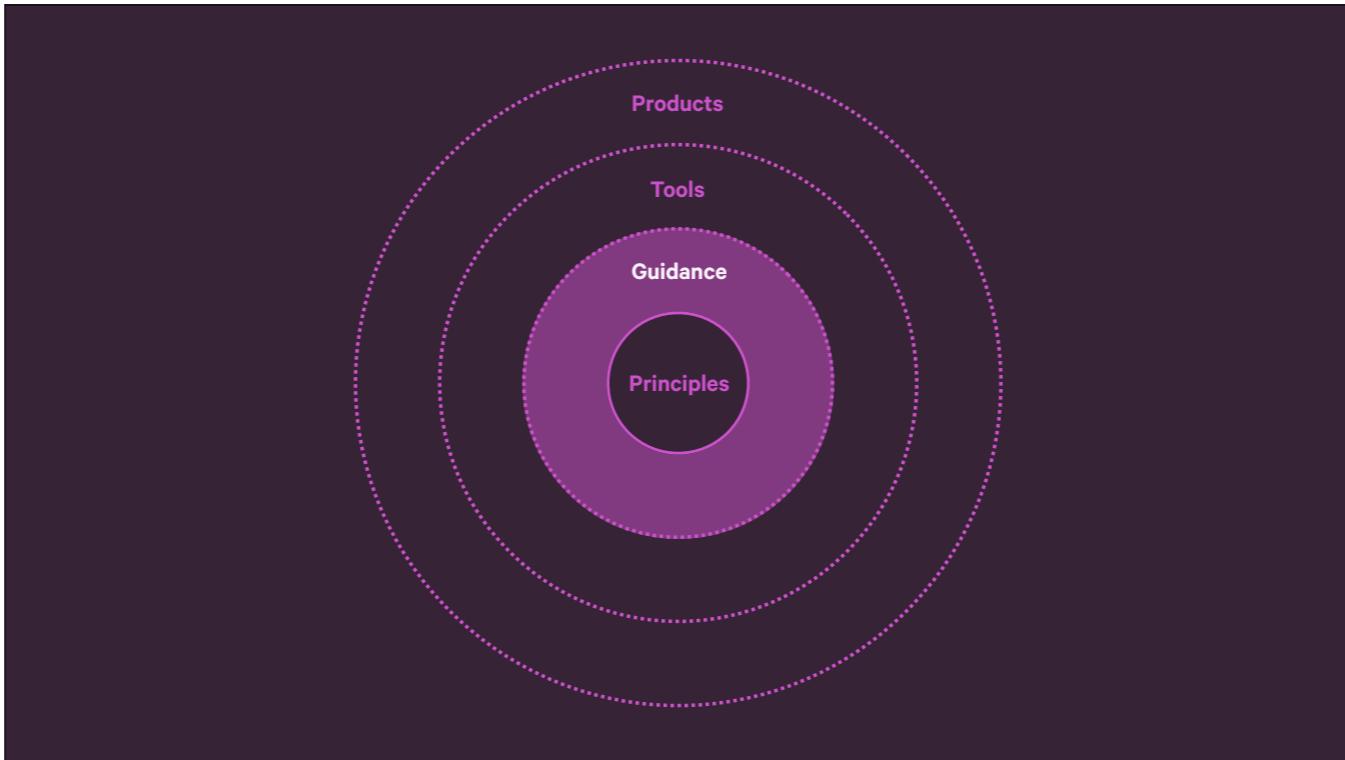
Design Principles

Design principles are short, insightful phrases that act as guiding lights and support the development of great product experiences.

Design principles enable you to be true to your users and true to your strategy over the long term.

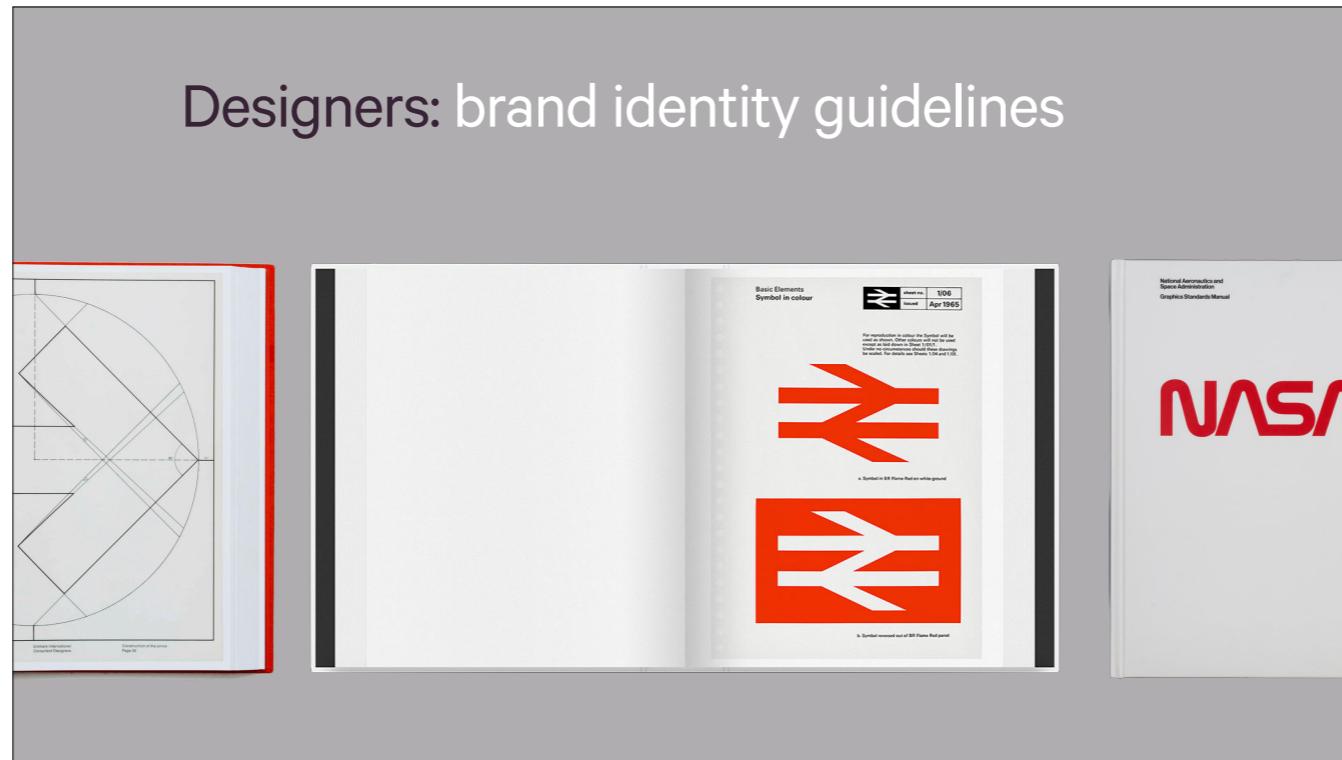
— Kate Rutter

- Kate Rutter calls design principles:
 - “*short, insightful phrases that act as guiding lights and support the development of great product experiences*”
 - that “*enable you to be true to your users and true to your strategy over the long term.*”
- Design principles should be focused primarily on user needs.
- They should also be distinctive and unique to organisation.
- Everyone wants their product to be easy to use, but what marks your product out as being different from its competitors?
- Creating these early on will help you make decisions later.
- Consider them the constitution for your design system.



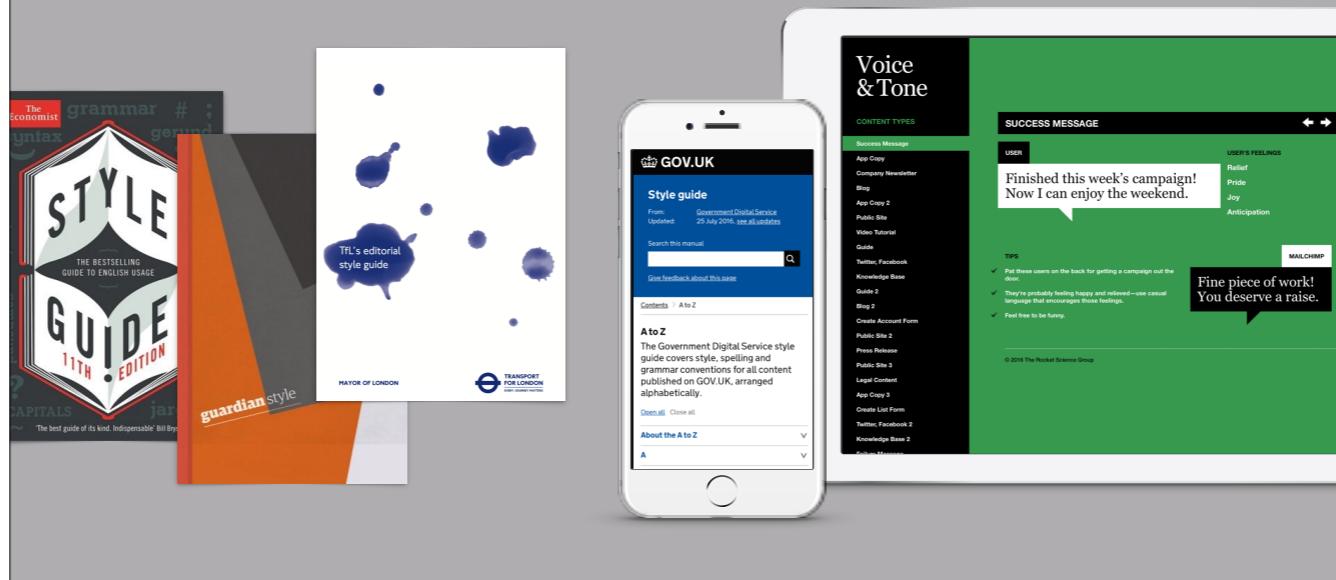
- Once these principles are in place, you can then start to consider a set of practitioner-centred guidelines.
- These are a set of artefacts that document how different practitioners should embed these principles within their own work.
- Examples of guidelines that might be considered within this layer include...

Designers: brand identity guidelines



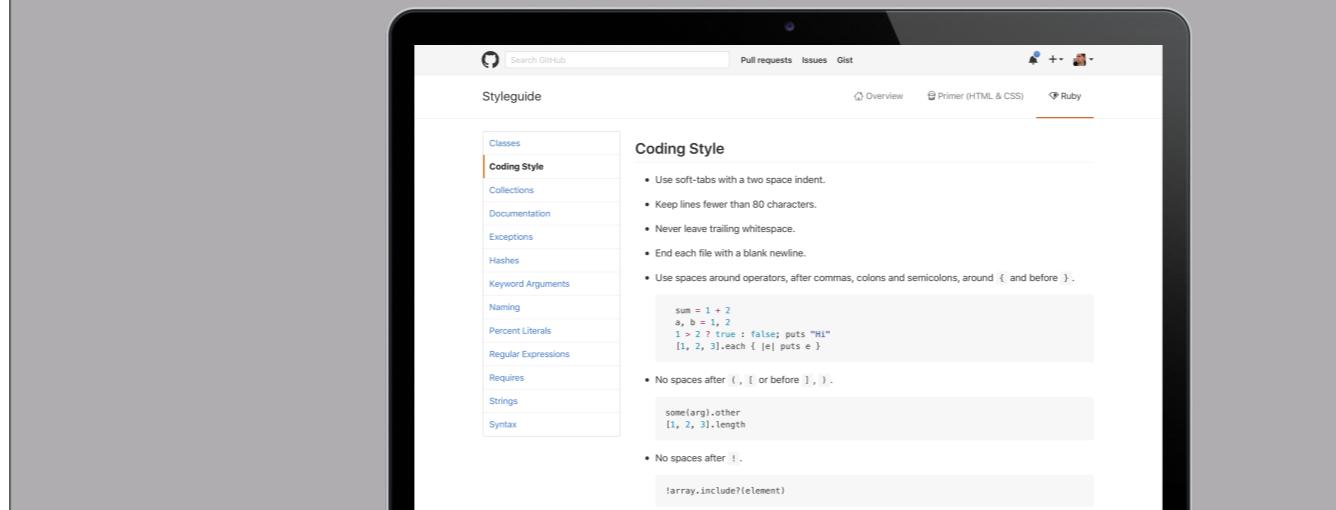
- Brand identity guidelines, which may form the backbone of how we think about an organisation's appearance.
- There's been a renewed interest in brand identity manuals of the past, such as those for New York Subway, British Rail and Nasa and many others.
- These documents can cover everything from logo usage (including variants, spacing and placement), to fonts, colour palettes, textures, patterns, iconography, photography etc.
- While brand consultants may no longer deliver multi-page tomes like these anymore, the guidance they do offer should be no less exhaustive.

Writers: tone of voice & style guides

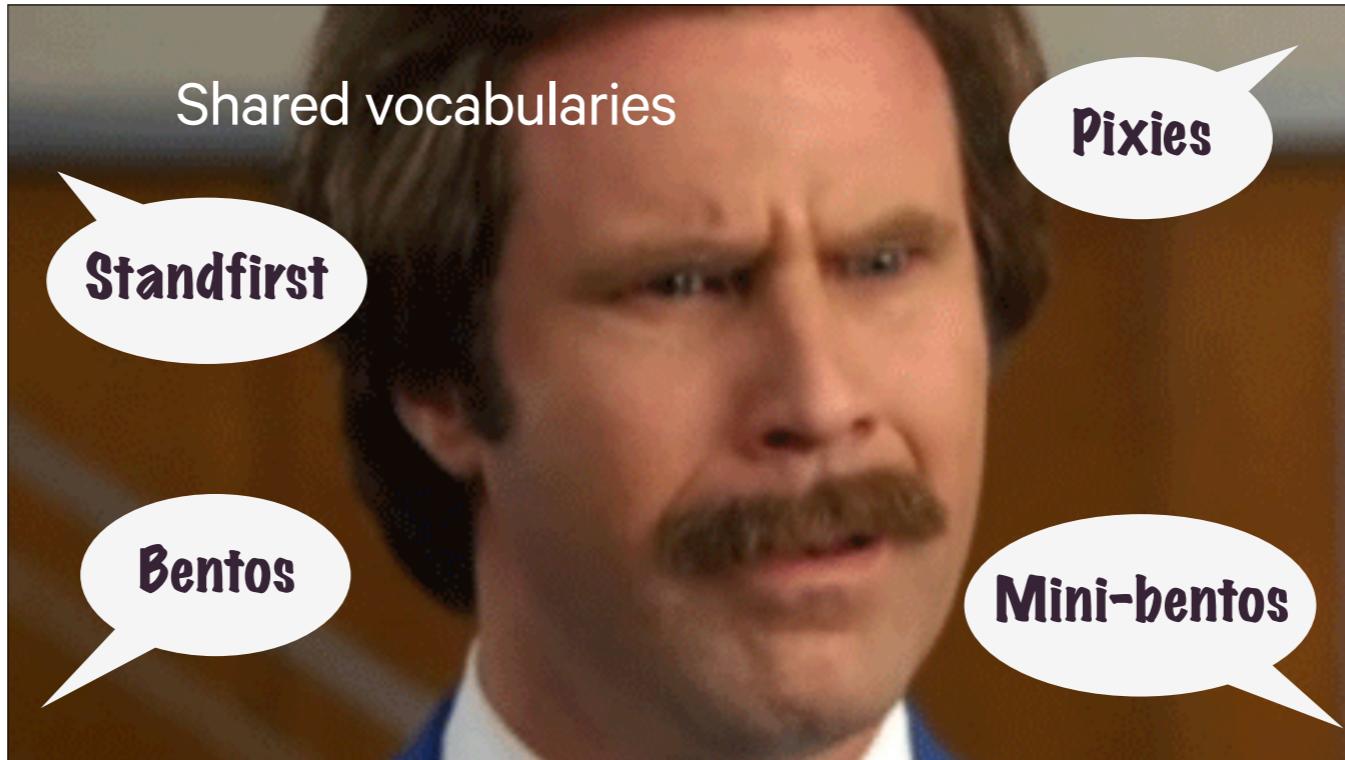


- Writers may create and use editorial and content guidelines.
- Style guides — in the original sense — cover preferred usage of particular words and grammatical choices.
 - These are important for content-heavy publications like newspapers and government information.
- Tone of voice guidelines speak less to the words, and more about sentences, and how we communicate with customers.
 - Tone of voice is important as it can significantly contribute to the distinctiveness of your product or service.

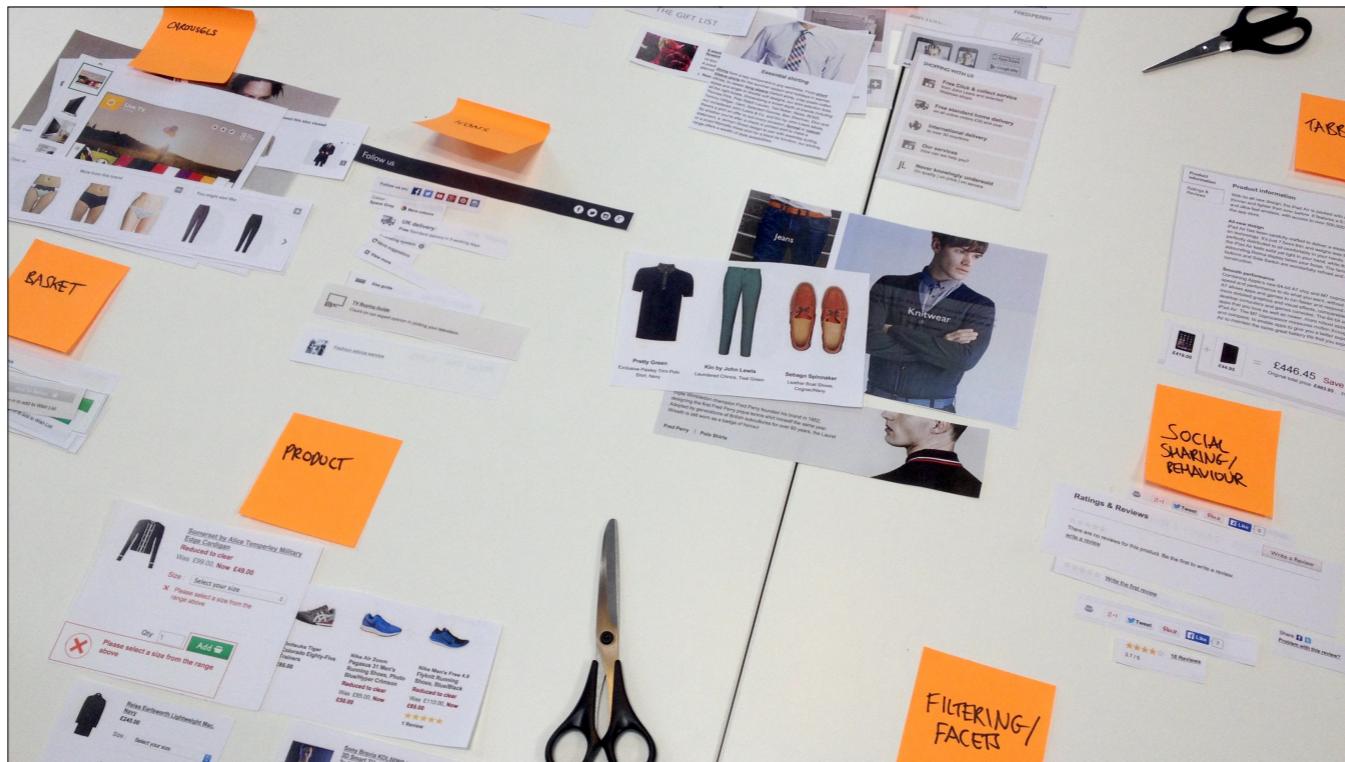
Developers: code styles/conventions



- Developers have guidelines and conventions too.
- While a products underlying code may not be seen by users, such guidance will ensure this remains consistent, predictable and maintainable.
- The correct answer to question “tabs or spaces” isn’t your personal preference, but established conventions on a particular project or within your organisation.



- The single idea that ties these artefacts together is that they document agreed conventions.
- They also help us establish a vocabulary that can be shared by everyone within a team.
- Organisations may have a number of shared vocabularies already.
- When I joined the Guardian, people would mention strange names like *pixies* and *super-pixies*, *bentos* and *mini-bentos*.
- They would regularly use industry terms like *stand firsts* and *lead ins*.
- To which your reaction, like mine might be “what did you say?”
- Naming things can be hard, but sometimes it can also be fun.
- But whatever names you do come up with, make sure they are documented somewhere and readily accessible, especially to new members of the team.



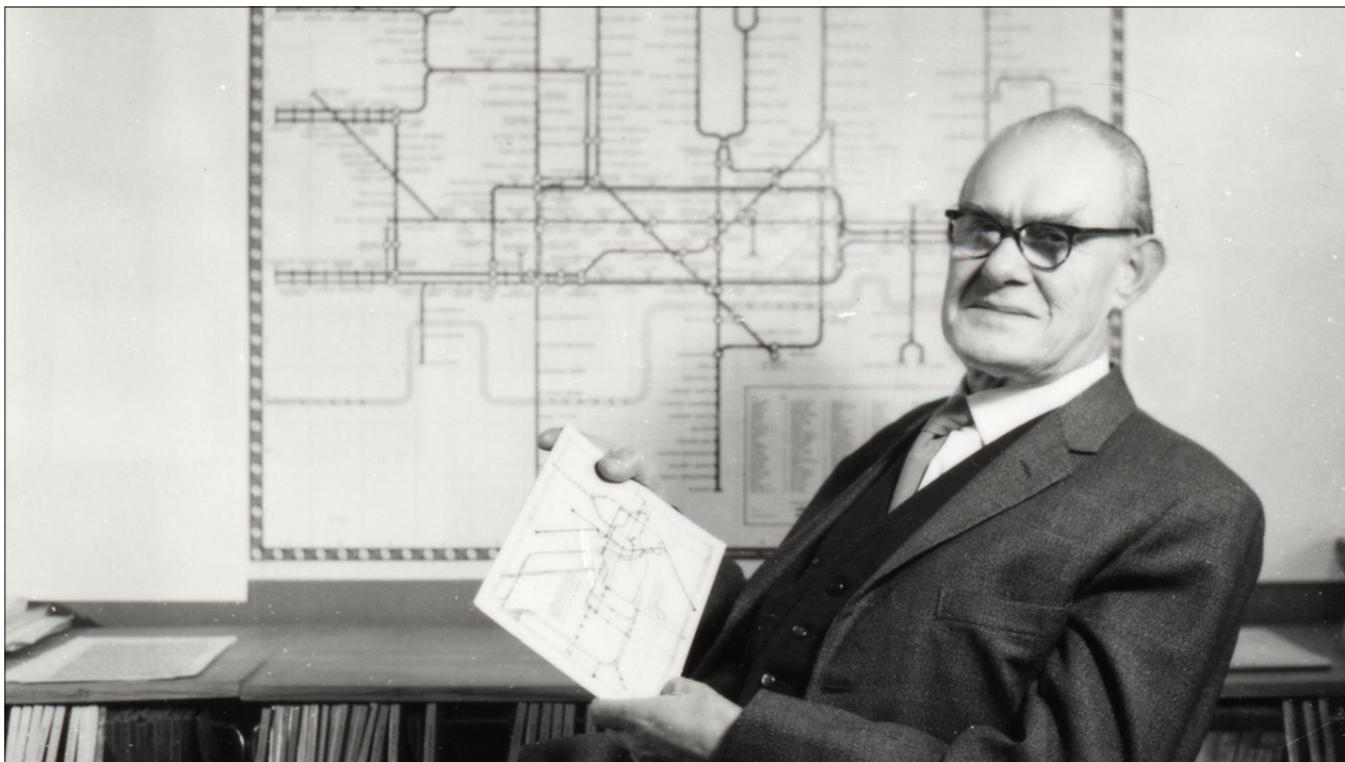
- If you are starting a project, you should work collaboratively to decide upon conventions and names for things.
- Nathan touched on this yesterday, and my colleague Charlotte Jackson has described a similar collaborative workshop you can use to help a team name their components:
 1. Cut up an interface into its component parts
 2. Group similar components
 3. Everyone comes up with a name for each group
 4. Once everyone has thought of a name, reveal them!
- If everyone has suggested the same name, well then that's likely to be a good candidate.
- If not, then the discussion that follows will be just as useful.

Design systems and Postel's Law

Policing a design system never works in my experience. It never works because people don't like rigid systems, being told what to do, and will straight up do the opposite. Being liberal in accepting things into the system, and being liberal about how you go about that, ensures you don't police the system. **You collaborate on it.**

— Mark Boulton

- The idea of working together to build a shared understanding, brings to mind a recent article by Mark Boulton.
- He suggested that a successful design system should adopt Postel's Law:
 - *being liberal in accepting things into the system, and being liberal about how you go about that, ensures you don't police the system. You collaborate on it.*
- I'm sure Frank Pick would agree.



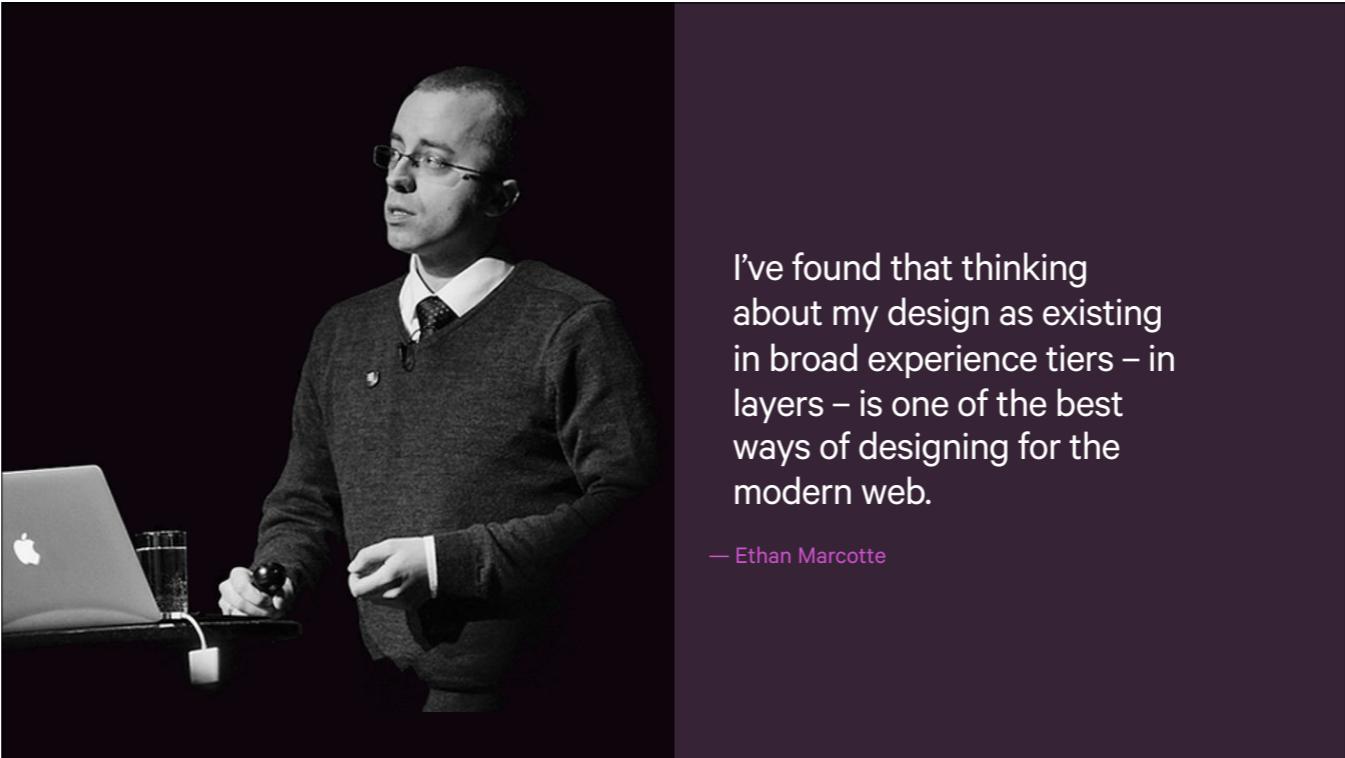
- After all, the most recognisable piece of design created during his tenure, the Underground map, was not instigated by Pick at all.
- Rather, it was a young engineering draftsman working on the network called Harry Beck, that came up with the idea.
- Here he is with his map a number of years later: he seems rightly chuffed with his little idea!
- The key realisation Beck had, was that passengers cared little about the geography above ground, they cared about the connections.
- So, he did away with the geographical accuracy and features (apart from the River Thames), and instead drew the various tube lines on vertical, horizontal and diagonal axes, marking intersections with a diamond shape.
- It's not the features that matter, but the connections. We'll come back to that thought later.

Components

- But first, I want to talk a little bit about components.



- I was fortunate enough to visit Brasilia in 2011.
- One of the things I observed was the repeating architectural motifs used by Niemeyer on his different signature buildings:
 - Elongated straight lines sat alongside soft curves...
 - Buildings appear suspended in air, or floating on water...
 - ...and have dramatic entrances, with long ramps that would ascend or descend into reception spaces...
- These repeating patterns are similar to the way we now think about building for the web.
- By creating identifiable and reusable components, you can piece these together in different combinations, yet ensure a product has a consistent and predictable feel to it.



I've found that thinking about my design as existing in broad experience tiers – in layers – is one of the best ways of designing for the modern web.

— Ethan Marcotte

- So how might we approach components in the context of a design system?
- Much like our model for a design system, we can consider a front-end component as being made of separate layers, or concerns too.
- By taking each concern on its own, we can focus on those guidelines and vocabularies that are applicable to it.

Component concerns



- The first concern, is content.
- Reduce any webpage to its bare essentials, and you're left with text.
- Even the most complex applications, rely on text labels and messaging.

Movie rating

The average rating is 4 out of 5 stars, from a total of [12 reviews](/reviews).

Add your score

Your name: []

Rating: () 1 () 2 () 3 (•) 4 () 5

[] Remember my details

(Submit rating)

- This is why I like to start building a component by writing plain text only.
 - I may use Markdown to give me a limited set of structural markers.
 - And for forms, buttons and inputs, I can use an ASCII-art like syntax.
- But otherwise, these constraints keep me focused on the content.
- And of course, I'll be looking to any editorial style guides, and tone of voice guidelines.

Component concerns



- The next step, is to think about meaning.
- For this, we'll **introduce a number of shared vocabularies** to help us.

```
<meter value="4" min="0" max="5">4 out of 5 stars</meter>
```

- Of course, there is one vocabulary that has already been defined for us: HTML.
- By using HTML's relatively limited palette of elements, again forces us to focus.
- *I loved Aaron's talk yesterday* — there is so much we can add to a component, just with sensible/discriminable element usage.
- It's worth reviewing the options available, especially given the range of new elements added in HTML5.
 - For example, on a recent project I used the meter element to describe a rating.

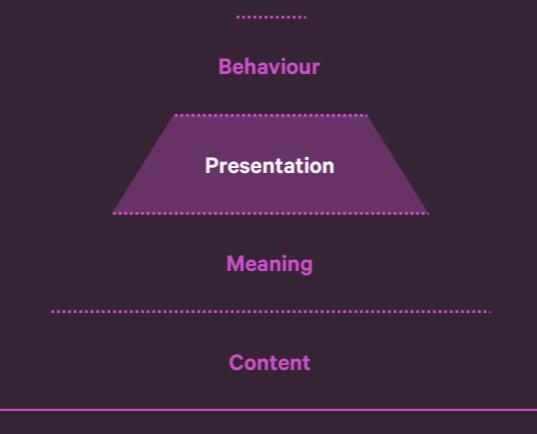
```
<section class="c-rating">
  <h1 class="c-rating__title">Movie rating</h1>
  <span class="u-hidden">The average rating is</span>
  <meter class="c-rating__value" value="4" min="0" max="5">
    4 out of 5 stars
  </meter>
  <span class="u-hidden">from a total of</span>
  <a class="c-rating__reviews" href="/reviews/">12 reviews</a>
</section>
```

- Yet HTML alone can only get us so far.
- To describe components in relation to our own internal vocabularies, we can use the class attribute.
- Within this attribute, we can also express these terms by following any established naming conventions.
- For example here I'm using the BEM naming methodology, and single letter class prefixes to indicate rule type (as per CSS name-spacing, described by Harry Roberts).

```
.c-[module-name] {...}  
.c-[module-name]__body {...}  
  
// Header  
.c-[module-name]__header {...}  
.c-[module-name]__title {...}  
  
// Content  
.c-[module-name]__main {...}  
.c-[module-name]__content {...}  
.c-[module-name]__items {...}  
.c-[module-name]__item {...}  
  
// Footer  
.c-[module-name]__footer {...}
```

- On the topic of naming, I also like to use naming meta patterns.
- These are repeated yet unrelated approaches for naming groups of elements.
- Here's a common meta pattern I like to use for naming child elements
- Note that I'm reusing a number of names used in HTML — their usage in relation to the component is similar, so to me it doesn't make sense to reinvent the wheel.
- Leban Hyde refers to this as 'the workhorse with no name', while Dave Rupert calls this the 'every module ever scaffold'.
- Repeating patterns like this make our code more predictable, easier to understand.

Component concerns



- Once I reach the presentation layer, it's here where things get really interesting, as I start to translate the visual design language into code.

```
// Colour palette
$color-brand--crimson: #c00;
$color-brand--mustard: #fc0;

$color-neutral--darkest: #222;
$color-neutral--darker: #444;
$color-neutral--dark: #666;
$color-neutral--mid: #888;
$color-neutral--light: #bbb;
$color-neutral--lighter: #ddd;
$color-neutral--lightest: #eee;
```

- First, I start by abstracting common design specifications into a set of variables.
- I am using colours as an example here, but the concepts can apply to other values such as typography and spacing.
- Saving values as variables keeps our code DRY, but on their own, they lack meaning.
- As Nathan wrote once: “Variables give us options, but they don’t give us decisions”

```
// Colour assignments
$color-text: $color-neutral--darkest;
$color-background--light: $color-neutral--darkest;
$color-background--dark: $color-neutral--darkest;

$color-link: $color-brand--crimson;
$color-link--hover: darken($color-brand--crimson, 10%);
$color-link--active: lighten($color-brand--crimson, 10%);
```

- Assigning our value variables to a second named set, gives them context.
- If we change our colour values later, we only need to change them in one place (hopefully).
- Again, this approach also makes our code easier to reason with.

```
@mixin typeset($preset, $level) {  
  @if ($preset == title) {  
    font-family: $typeface-serif;  
    font-weight: bold;  
    letter-spacing: 0.0025em;  
    text-transform: uppercase;  
  
    @if ($level == 1) {  
      font-size: 1em;  
      line-height: 1.25em;  
    }  
    @if ($level == 2) {  
      font-size: 2em;  
      line-height: 2em;  
    }  
  }  
}
```

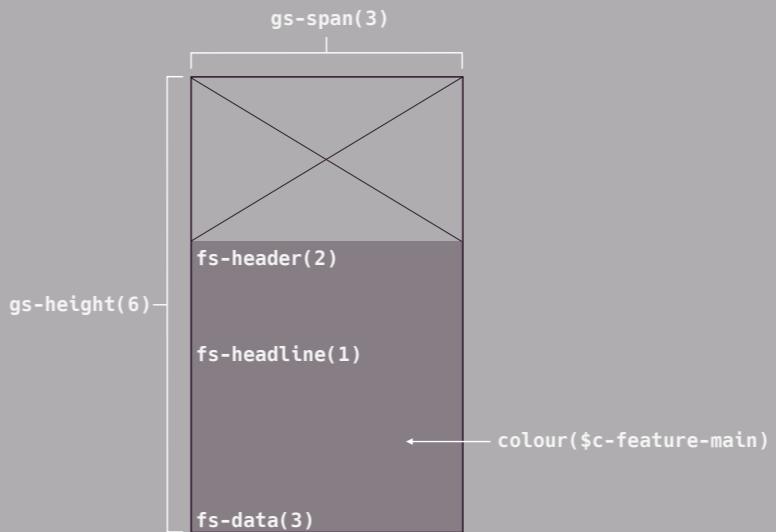
- For more complex aspects of the visual language, we can start to involve mixins.
- Something I've been doing lately is creating mixins in to manage my different typographic presents and sizing scales.

```
.c-module__title {  
  font: bold 2em/1 Georgia, serif;  
  margin-bottom: 1.5em;  
  padding-top: 0.75em;  
  letter-spacing: 0.0025em;  
  text-transform: uppercase;  
  color: #222;  
}  
  
.c-module__title {  
  @include typeset(title, 2);  
  margin-bottom: ($baseline * 4);  
  padding-top: ($baseline * 2);  
  color: $color-text;  
}
```

- The idea is to reach a point where we are no longer giving rules intangible values...
- ...but almost producing something more akin to a **design specification**, from which we can infer intent.

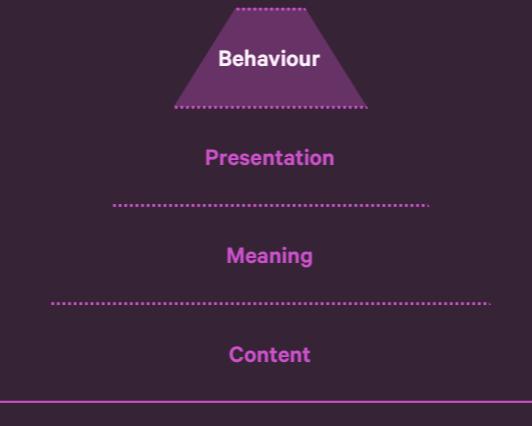
Guardian Style Sheets (GUSS)

github.com/guardian/guss



- This is similar to the system used at the Guardian.
- GUSS is a set of Sass mixins and other helpers that meant we could use our internal naming vocabulary, within our code.
- With GUSS, components could be described using a **language familiar to everyone** on the team, be they designers, developers, product managers and other stakeholders.
- I think there is a temptation to over-use tools like Sass and other pre-processors, and I wasn't so keen on them at first.
- However, once I understood that these as tools can allow use to embed our design language in code, it really clicked.

Component concerns



- As we reach the top of the stack, we can start to look at how our components behave.
- JavaScript isn't my forte, so I won't dwell too long on this.
- But even here, we can introduce naming conventions, and use similar concepts of starting simple and building up an experience in layers.
- There are also a number of aspects that I have not touched on here, that will likely transcend these layers.
- For example, considering the accessibility of a component:
 - thinking about the choice of language and readability is a content concern.
 - and ensuring text is legible and has suitable colour contrast is a presentation concern.

Connections

- So that's how I tend to think about building my components.
- But remember, it's not the features that matter, but the connections.



- Let's return to Brasília.
- While Niemeyer's buildings are gorgeous, they do tend to suffer from an emphasis on form over function.
- For example: this building, the National Congress, has this ramp at the front that looks like the entrance.
- In fact it is purely decorative; a point reinforced by a barrier that sits half way up it to prevent you climbing it — and the roof occupied by armed guards!



- Costa's layout suffers similarly. Along the main axis, there are these large areas of green space.
- I'm sure these satisfied an objective of providing open space for people to relax in; yet this **simplistic placement** neglects to consider that to reach these spaces, you need to cross six lanes of traffic.
- These things make sense on their own, but don't quite work when put together.



Always design a thing by considering it in its next larger context – a chair in a room, a room in a house, a house in an environment, an environment in a city plan.

— Eliel Saarinen

- A quote from Finish architect Eliel Saarinen's comes to mind...
 - *“Always design a thing by considering it in its next larger context – a chair in a room, a room in a house, a house in an environment, an environment in a city plan.”*
- In our discussion of components, we don't talk about chairs, but we tend to talk a lot about Lego.



- This is the real world equivalent of object-orientated programming: separate blocks that can be pieced together in any number of combinations.
- This is a useful analogy for explaining modular design to a client, or somebody not familiar with this approach...
- ...but look closely, and you'll find some shortcomings in this comparison.

The wrong analogy?



- Composite
- Static
- Encapsulated
- Clear affordances



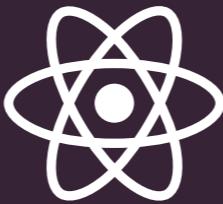
- Separate concerns
- Dynamic
- Leaky
- Difficult to reason with

- A Lego brick is...
 - a **composite**, made of a single material, ABS plastic,
 - **static**, its state doesn't change under normal operating conditions,
 - **encapsulated**; its colour/shape won't alter depending on the bricks attached to it,
 - and has clear **affordances**, you can understand its properties by looking at it.
- A front-end component is...
 - composed of **different concerns**, (as previously discussed),
 - **dynamic**, maybe intentionally through interaction, or affected by external factors,
 - **leaky**: the very first initial of CSS describes the opposite approach, rules able to *cascade*,
 - **difficult to reason with**, especially if code is in separate files, with little/no documentation.

Bringing encapsulation to the web



BEM
en.bem.info



React
facebook.github.io/react



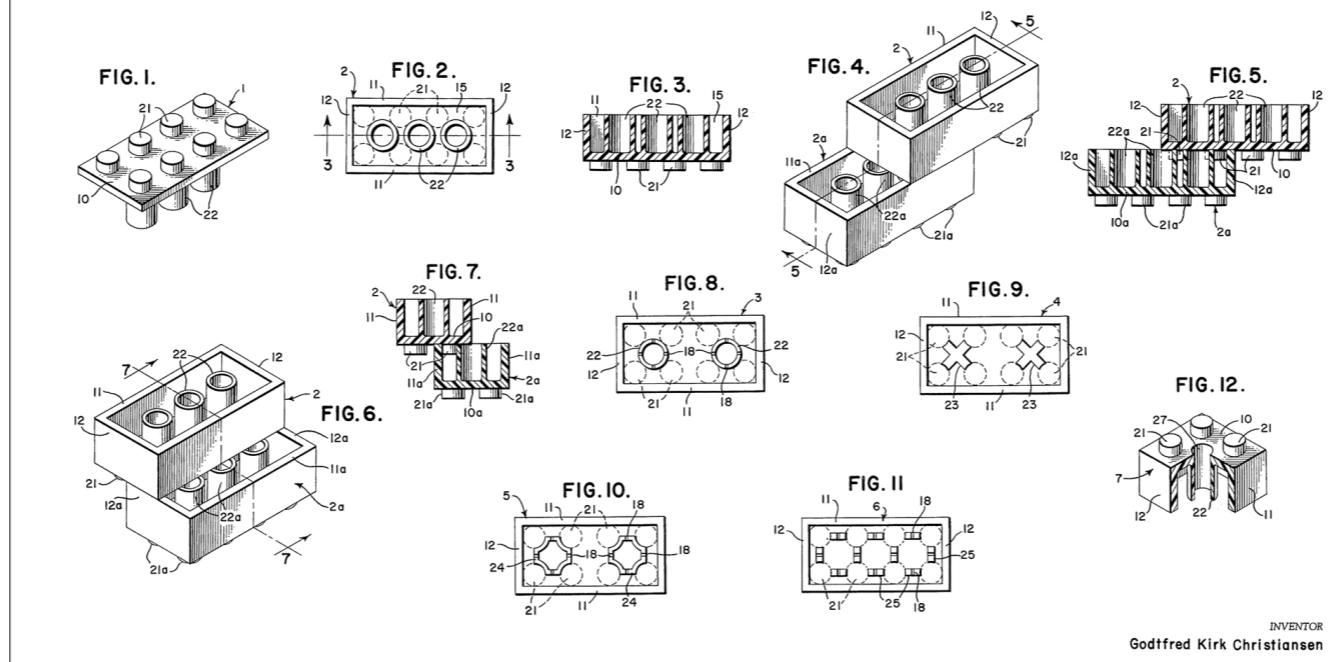
Web Components
webcomponents.org

- In the quest to build modular systems with greater predictability like Lego, libraries and tools seek to address these differences.
- We see approaches to CSS that seek to undermine the cascade, be that via name-spacing or applying styles directly.
- React components combine HTML and JavaScript into a single JSX file, and CSS modules bringing that concern into play too.
- These approaches are entirely understandable, and not necessarily wrong.
- ...but, it does feels like we're constantly working against the grain of the web.

Oct. 24, 1961
Filed July 28, 1958

G. K. CHRISTIANSEN
TOY BUILDING BRICK

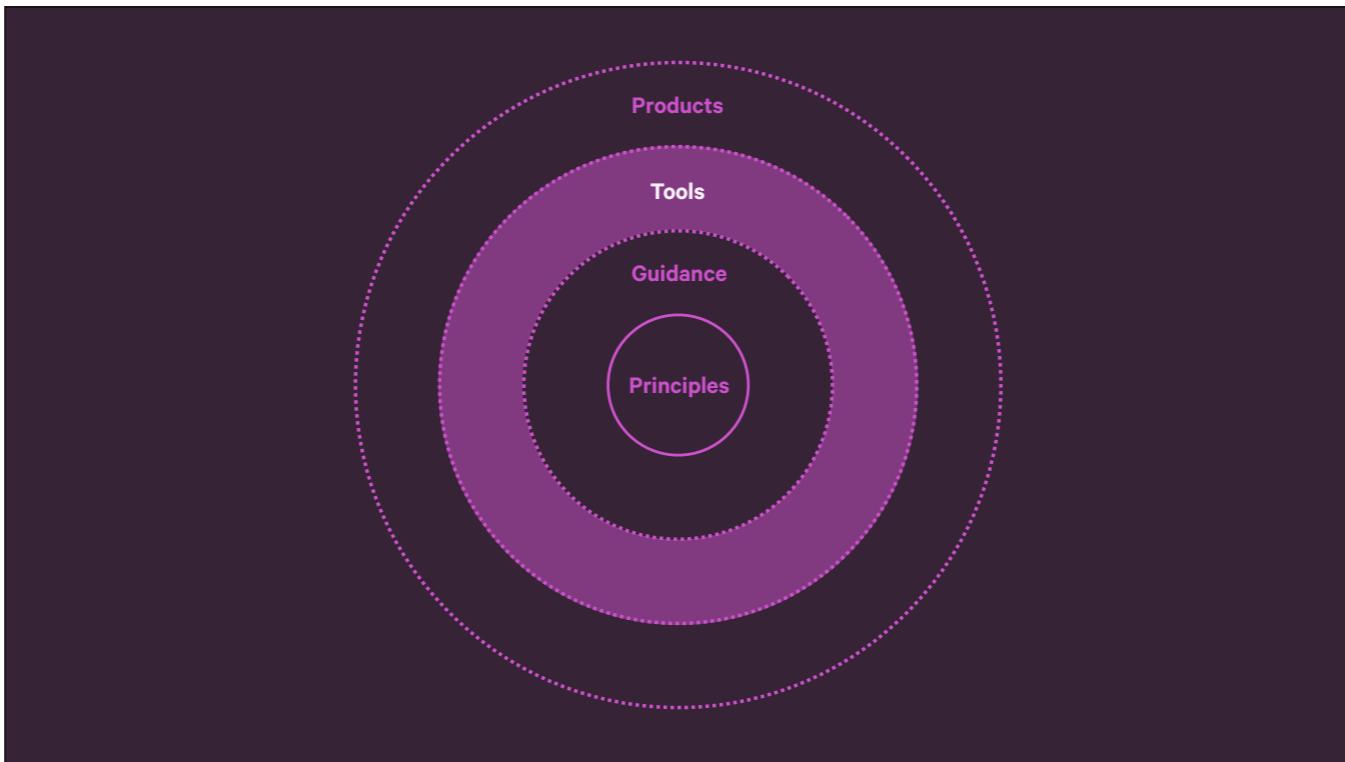
3,005,282
Sheet 1



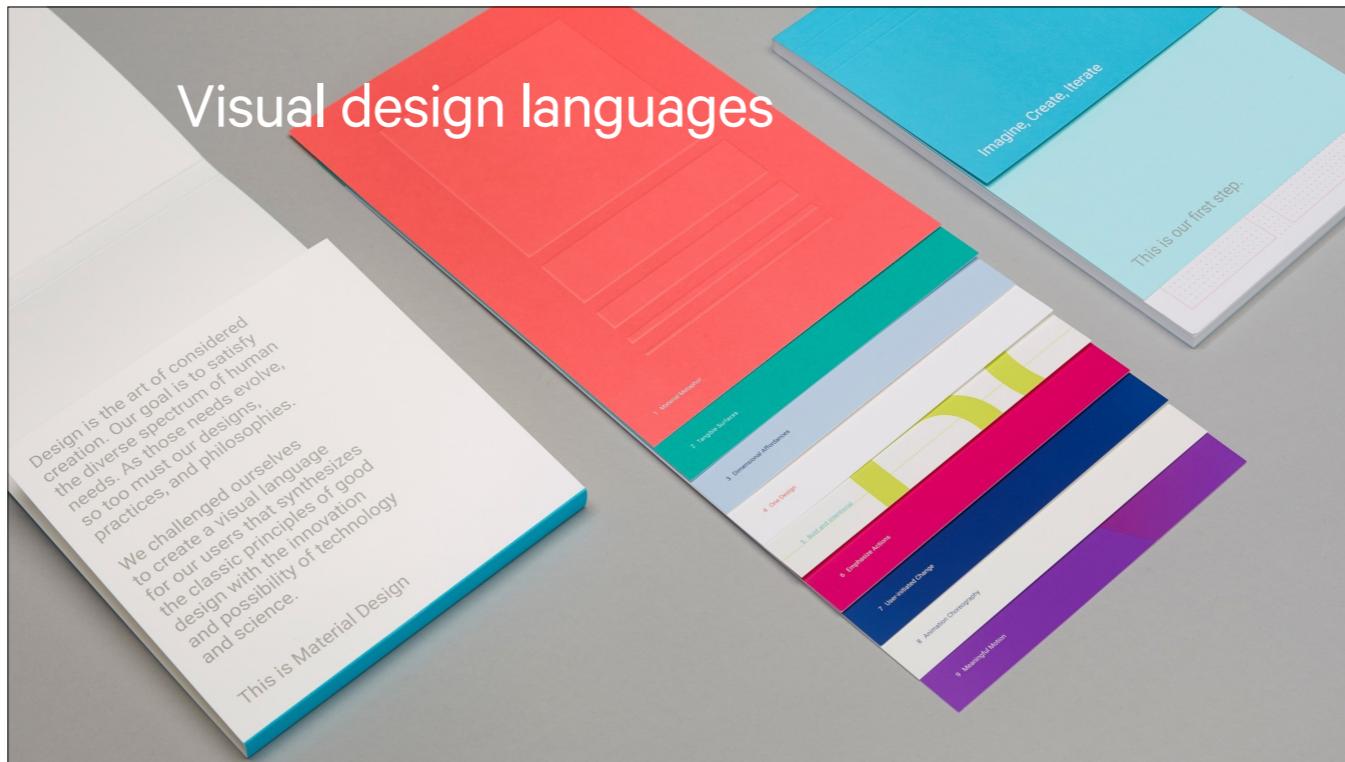
INVENTOR
Godtfred Kirk Christiansen

- Maybe Lego is the right example, but we've just been looking at the wrong part of it.
- The most important aspect of Lego is not the bricks, but the system of tubes and stubs that holds them together.
- New brick shapes have been added to the system over the years, yet a brick manufactured today will still connect with one of the first produced in 1958.
- What are the equivalent concepts within the realm of front-end development?
- What might we discover if we were to focus more on the relationships between components, rather than the components themselves?
- Who knows, maybe the cascade could become a powerful ally, rather than pernicious foe?

Composition



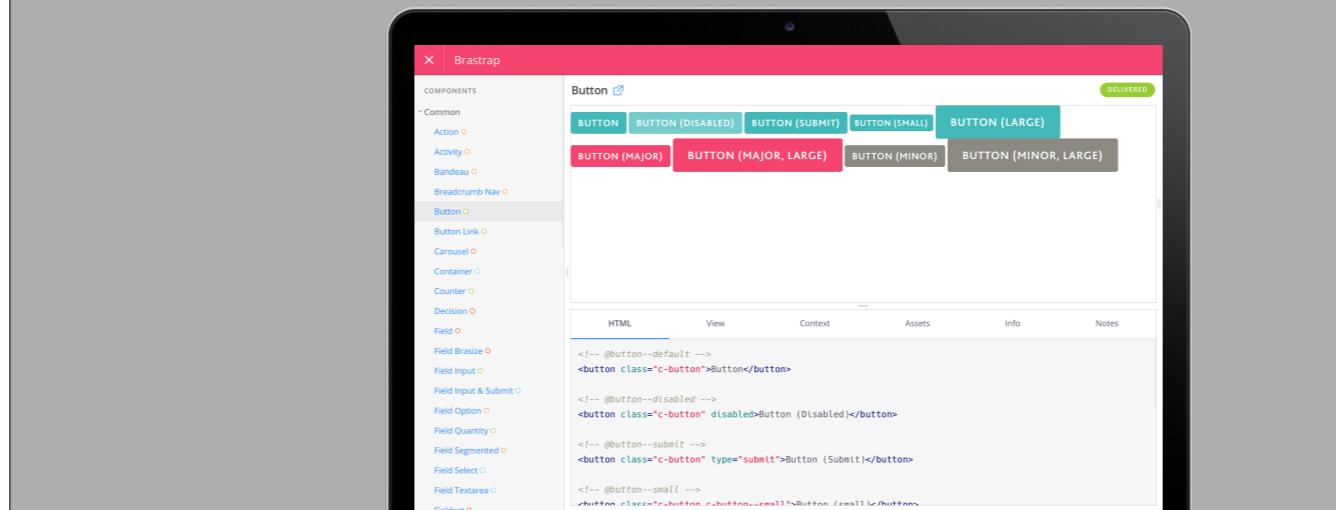
- To the penultimate layer in our design system: tools.
- A complete design system needs to provide a series of tools, that go beyond low-level guidance (the what)...
- and enable people to implement these guidelines in a meaningful way (provide the 'how').



- I'd suggest that a visual design language is a subset of a design system, and I place it within this layer.
- These are similar to brand identity guidelines, but concentrate more on the **application** of brand identity, and provide guidance on composition, order and placement.

Component libraries

<http://fractal.build>



- Another tool we might include in this layer is component libraries.
- Now, within a larger organisation you may want to build your own tools, but there are a number of freely available alternatives that can help us, and also learn from.
- One of which I've been working with a lot is called Fractal, built by Mark Perkins at Clearleft.
- I urge you to check it out if you haven't already—it's really nice!

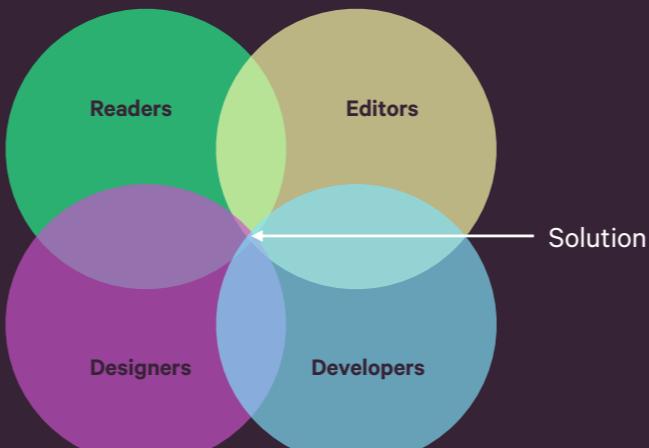
The container model

next.theguardian.com/blog/container-model-blended-content



- But I want to talk about a different kind of tool we used at the Guardian, called the container model.
- First, some context.

Confluence of interests

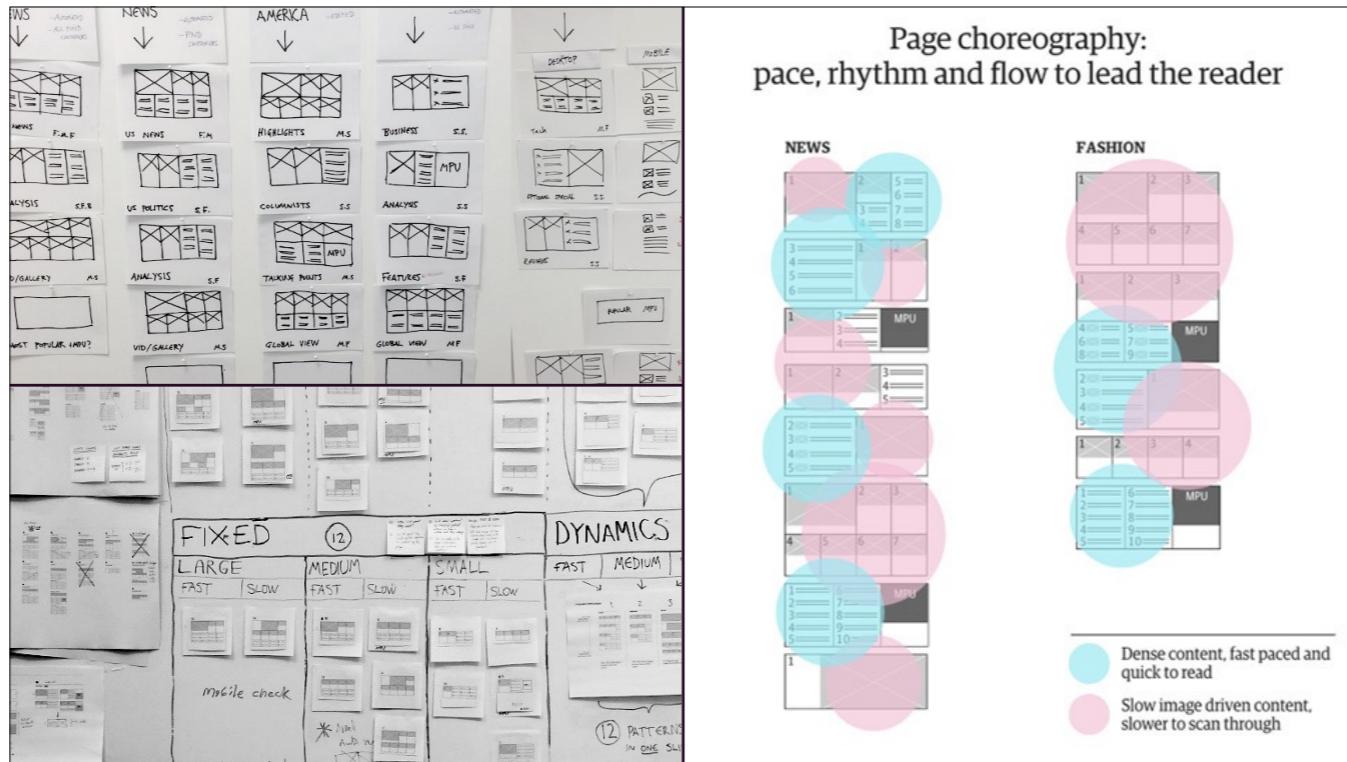


- As part of the responsive redesign, one of the team's central aims was to improve the discovery of different types content across the website.
- However, we needed to take into account the sometimes conflicting needs between different stakeholders:
 - **Readers** wanted predictable means of finding content.
 - **Editors** wanted to highlight the content they were keen on promoting.
 - **Creative director** wanted a distinctive design infused with Guardian's renowned print design values.
 - **Developers** wanted a system that wouldn't be overly complex or hard to maintain.
- Somewhere in the middle, a solution needed to be found.

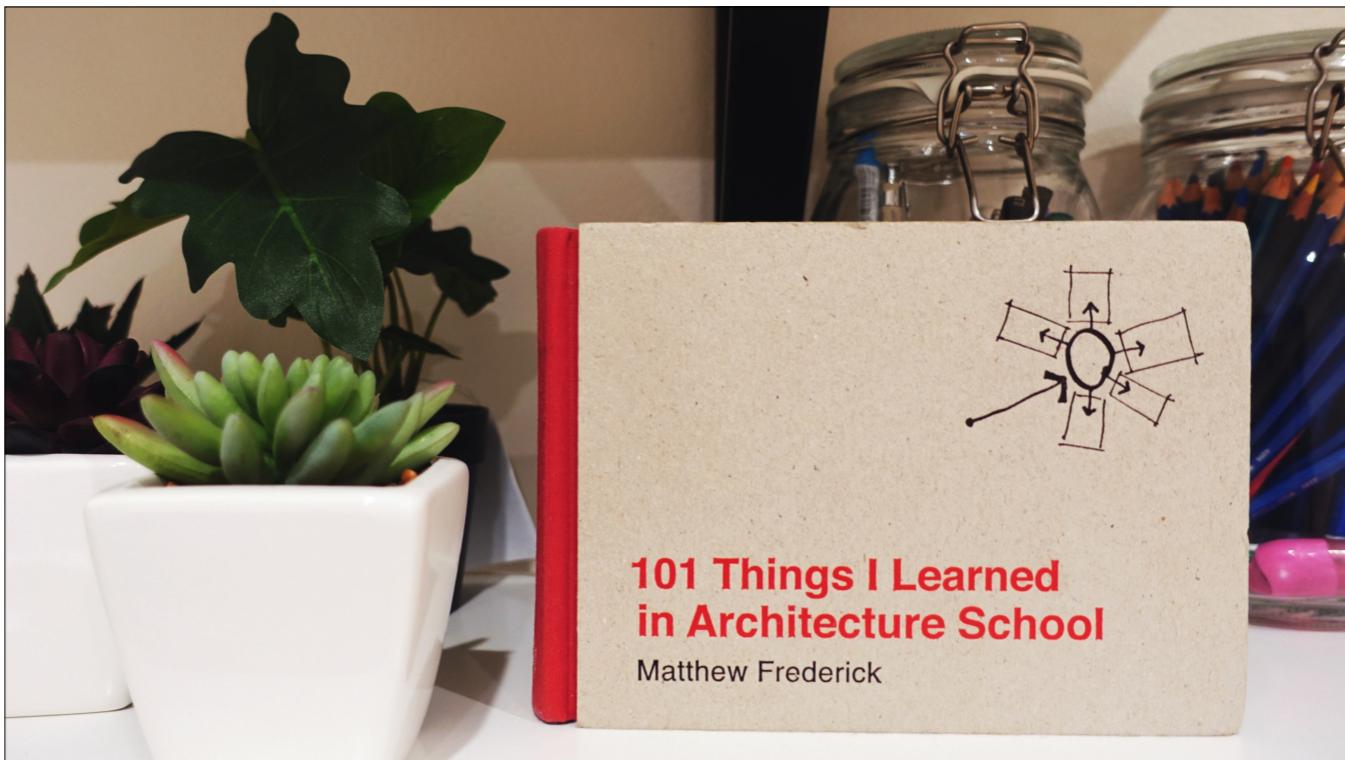
The container model



- Enter, the container model. This brings a systematic approach to page composition.
- In short:
 - there are items (which in this case each representing a single story),
 - these can be arranged into a number of different horizontal slices,
 - which themselves can be combined to create containers.
- Each container would revolve around a particular theme, say people in the news, or features, or videos.
- By building up each container from separate items and slices, gave us an incredible amount of flexibility, and reduced the amount of code needed to maintain it.
- Best of all, because these containers are self-contained, they can be shared across a number of different pages and in varying positions; for example the ‘most popular stories’ container on the homepage, could also be shown at the foot of an article page.
- In retrospect this seems pretty straight forward, but turns out, this only solved half the problem.



- We also needed to consider that:
 - The type of story could affect its **presentation**,
 - The newsworthiness of a story could affect its **position**,
 - Whether a container was curated or dynamically generated may affect its **layout**,
 - Also needed to consider the overall **pacing** of a page,
 - ...all while ensuring any solution would work across a range of **viewports**.
- Practically everyone on the team had a go at solving this puzzle.
- In the end, we managed to find a solution; partly because the team suddenly faced a six-week deadline for shipping something. Always helps.
- Also: we split the task up: one person focusing on the creation of blueprints for different container types, another focusing on the design of the individual items.



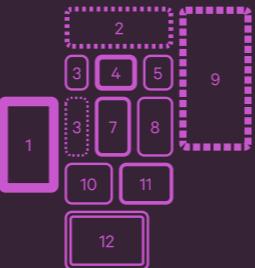
- In preparing for this talk, I've been re-reading this fantastic little book by Mark Frederick called *101 Things I learned in Architecture School*.
- I recommend getting it if you can, as there is so much in there I think relates to the work we do on the web.

Three levels of knowing



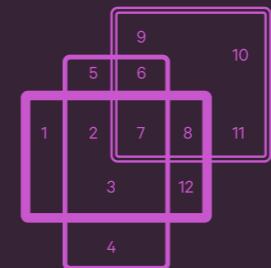
Simplicity

3 elements used to create 3 spaces



Complexity

12 elements required to create 12 spaces



Informed simplicity

3 elements combined to create 12 spaces

- One of the things he talks about is the Three levels of knowing:
 - **Simplicity:** the world view of a child, fully engaged in his own experience, and happily unaware of what lies beneath the surface of immediate reality.
 - **Complexity:** characterises the world view of an adult, in which there is an awareness of the complexity, but an inability to discern the patterns and connections.
 - **Informed simplicity:** an enlightened world view, in which a person has developed the ability to recognise and create patterns within complex mixtures.
- This perfectly encapsulates an experience I've become all too familiar with when creating a design system.
- You kind of have to work with the pieces for a long time, trying out different combinations until you finally discover the answer.

Change

- The final thing I want to talk about, is change.



- This poster is currently on display on the London Underground, which shows how the roundel logo has evolved over the last century.
- Frank Pick departed London Transport in 1941, and of course, over the years that followed, fashions have changed.
- How might it change over the next 100 years?

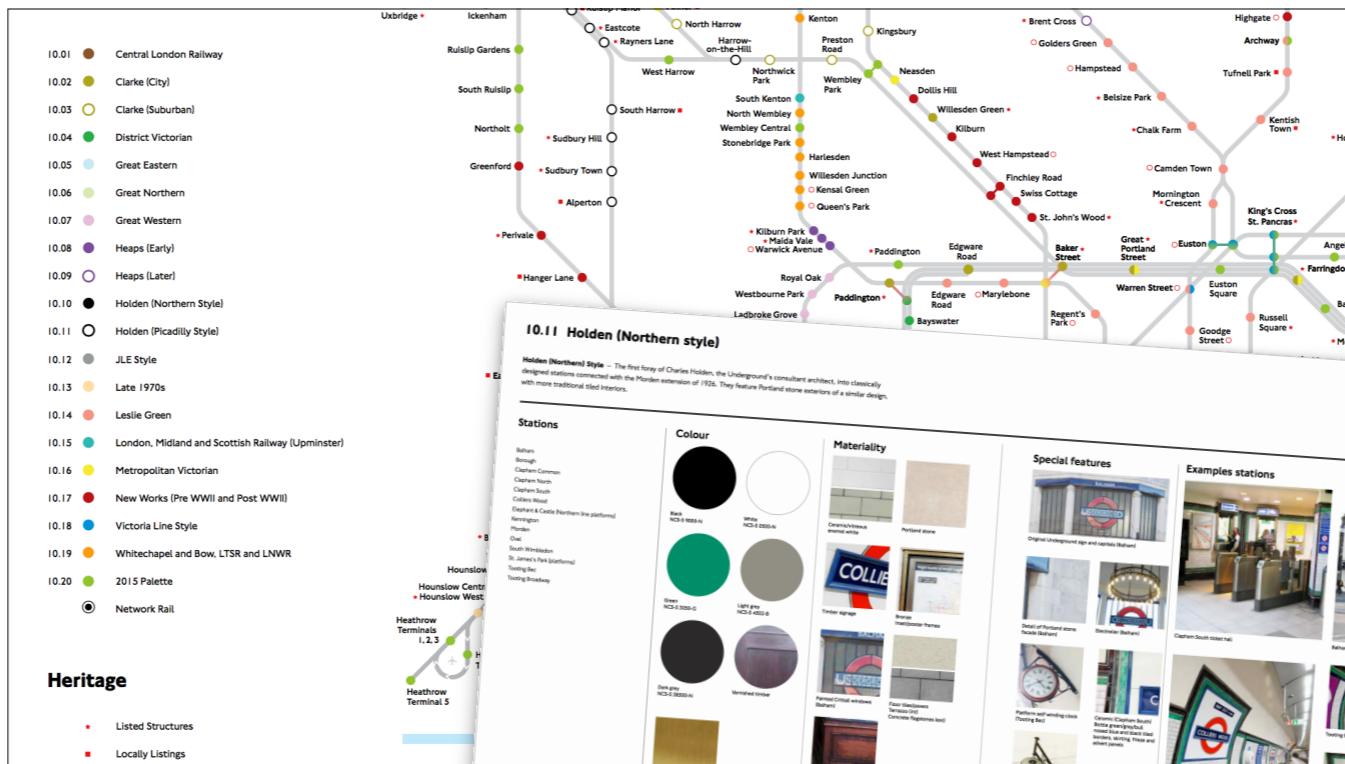


Photograph: Kirk Bauer

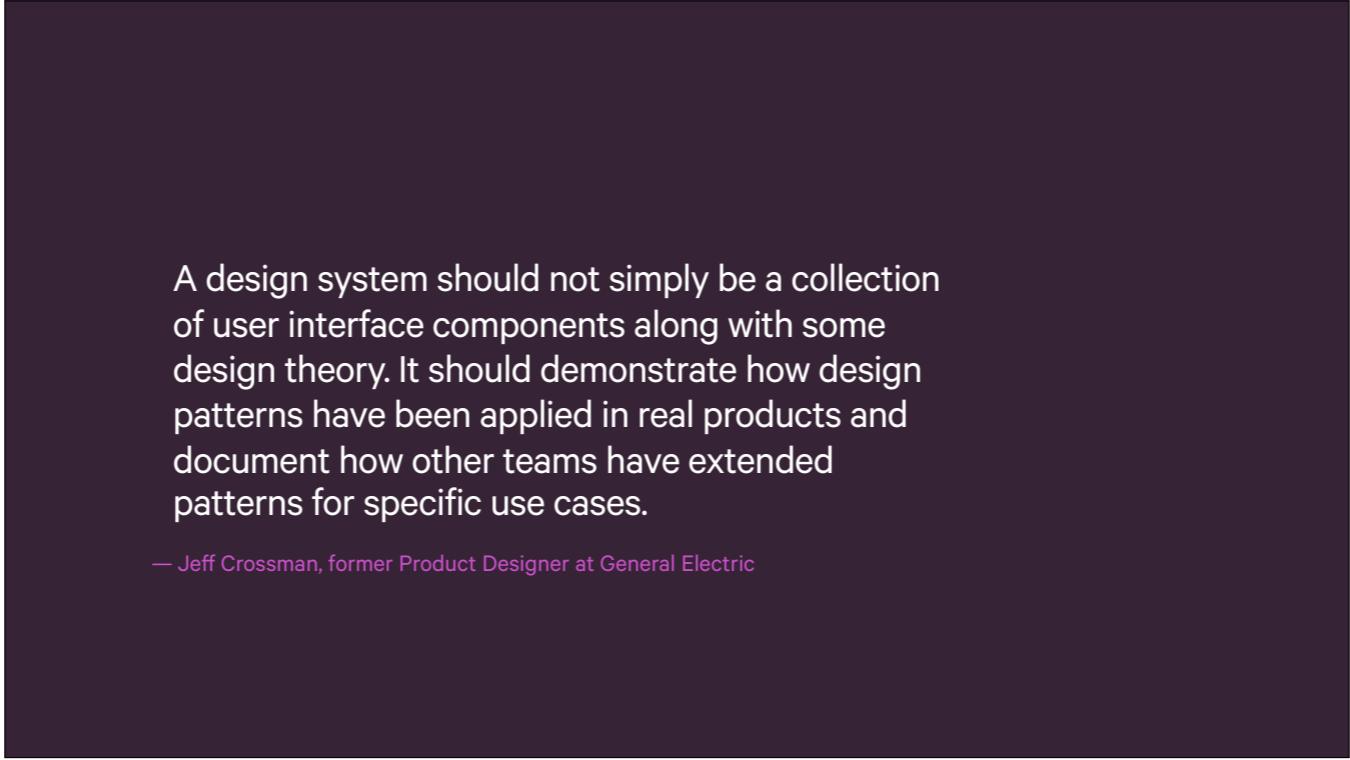
- Not only do fashions change, but so do priorities.
- This is a photo of the underground taken in 1997 – very different from how it looks today.
- It's worth acknowledging, that Pick's success came in part due to the support he received from the Underground's chairman, Lord Ashfield, who was willing to support Pick's various endeavours, seeing that they achieved results.
- But in the immediate decades following the departure of both Pick and Lord Ashfield, and with greater private car ownership and declining passenger numbers and revenue, managers prioritised price over quality — and it showed!



- Thankfully, the last 15 years have seen all those factors reverse, so once again the network is getting the attention it deserves.
- New lines are being built, such as the Elizabeth line, and existing stations are also being upgraded and repaired, too.
- The way Transport for London are managing this renewal is really interesting.
- One of the tools they've introduced is a set of 9 new design principles.
- Collectively called the Station Design Idiom, these principles underpin design across the Underground network.
- They are applied to every station and on every project, from small-scale repairs, to major refurbishments and new stations, and are used as a checklist throughout the design process.



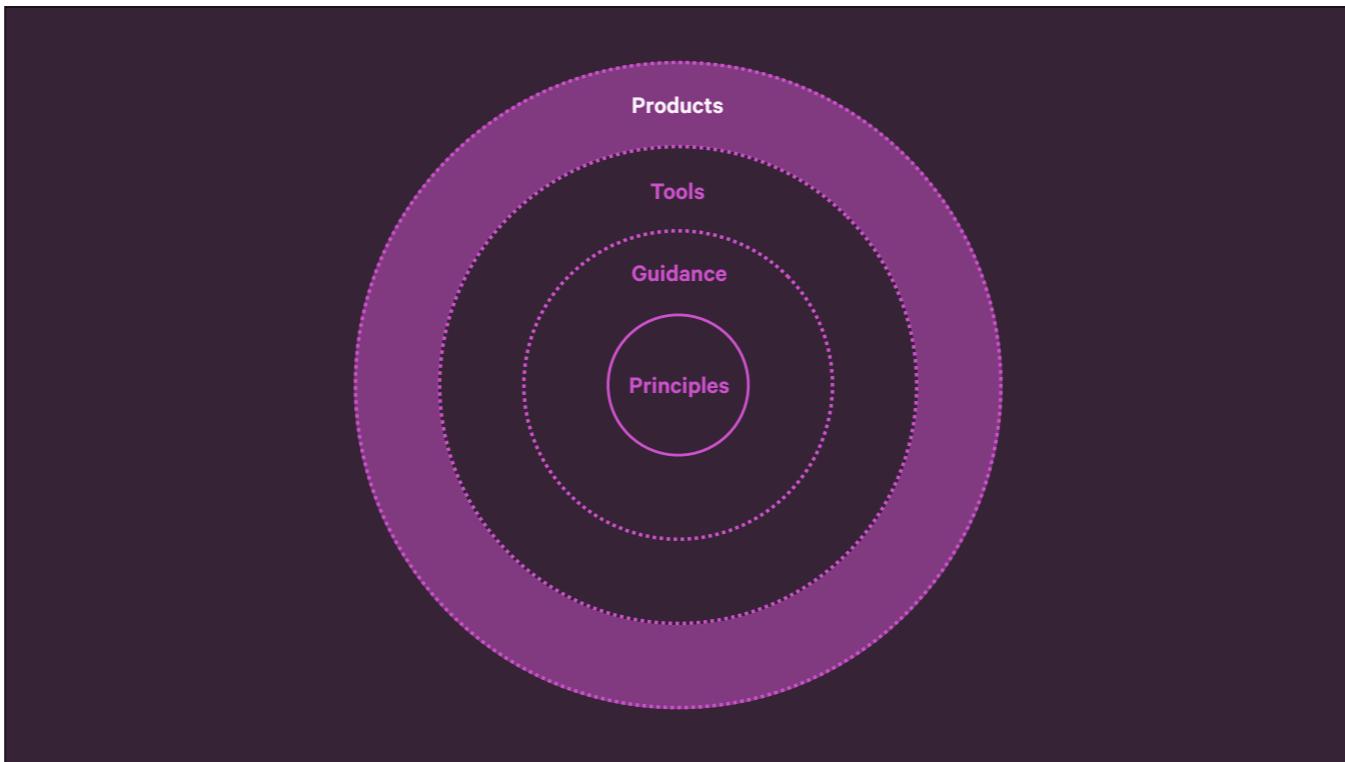
- Forming part of the idiom, there's also a new map that gives contractors a station-by-station guide to the different architectural styles used throughout the network, ensuring they use the right paint colours and finishes.
- Rather than attempt to impose a single style across the network, Transport for London have decided to celebrate the past instead.
- They are deliberately maintaining a diversity of styles, and in doing so, making them an integral part of a broader design system.
- This seems like a pragmatic, and reasonable approach to me.



A design system should not simply be a collection of user interface components along with some design theory. It should demonstrate how design patterns have been applied in real products and document how other teams have extended patterns for specific use cases.

— Jeff Crossman, former Product Designer at General Electric

- Jeff Crossman, a former product designer at GE wrote:
 - *A design system should not simply be a collection of user interface components along with some design theory. It should demonstrate how design patterns have been applied in real products and document how other teams have extended patterns for specific use cases.*
- I think Jeff touches on an important point here; the products of a design system will be the ultimate arbiters of its success.
- Beyond the immediate success (or failure) of these products, as they evolve to meet the changing needs of our customers, so the systems supporting them will need to be revised as well.

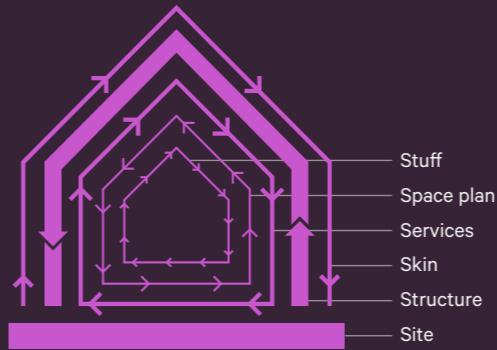


- So, the final layer in our design system, is the products that emanate from it.
- I really do feel that we should consider the products of a design system, as an integral part of it.
- These products — this surface layer emanating from all the work and thought below — is what users actually see.

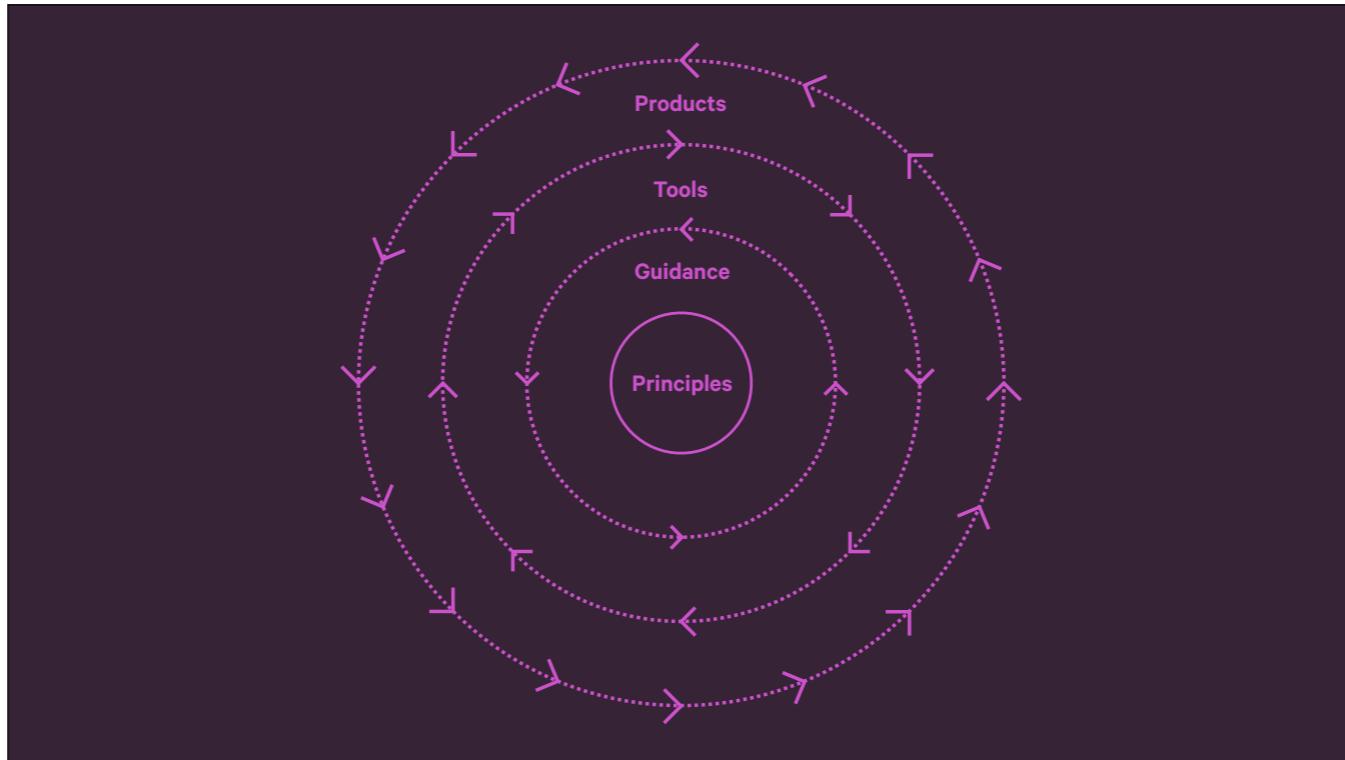
Shearing layers

Our basic argument is that there isn't any such thing as a building. A building properly conceived is several layers of longevity of built components.

— Frank Duffy, quoted in *How Buildings Learn*



- In architecture, there's this concept called shearing layers.
- This was coined by architect Frank Duffy to describe the different timescales that effect a building.
- He stated that:
 - *Our basic argument is that there isn't any such thing as a building. A building properly conceived is several layers of longevity of built components.*



- Perhaps there isn't any such thing a design system, but several layers of longevity of agreed conventions and tooling?
- So, at the surface, it is our products that are likely to change more frequently, maybe even daily.
- As our product strategies change to meet an evolving understanding of our users, so the tools that we built to support them will need to change too.
- And while our principles should remain fairly consistent, much like the United States constitution, they should still be open to periodic amendment and review.

Collaboration Components Connections Composition Change

- So to recap, today I have talked about:
 - Building a design system through **collaboration**
 - Embedding a shared vocabulary in our front-end **components**
 - Exploring the **connections** between those components
 - Devising tools that can help us create **compositions**
 - Ensuring we consider **change** an integral part of a design system
- And right here is the problem. Look! Notice how I've managed to use words beginning with a C for each of these sections?
- We designers can't help doing stuff like this!
- We love to discover patterns. We live for creating rules, defining hierarchies.



- But the world isn't like that. It's complex.
- And so are humans! We are emotional beings, not always reasonable, or rational.
- The web, an expression of humanity, is also chaotic, fluid and unpredictable.
- So, when we are designing for others, be they residents of a city, passengers on a transport network, or customers using a website, I think we should actually embrace this complexity.
- For complexity is the only thing that will ground our decisions to reality...
- ...the unfortunate space that lies between theory and practice.



Danke!

[@paulrobertlloyd / paulrobertlloyd.com](https://paulrobertlloyd.com)

© Attribution, Non-Commercial, Share Alike