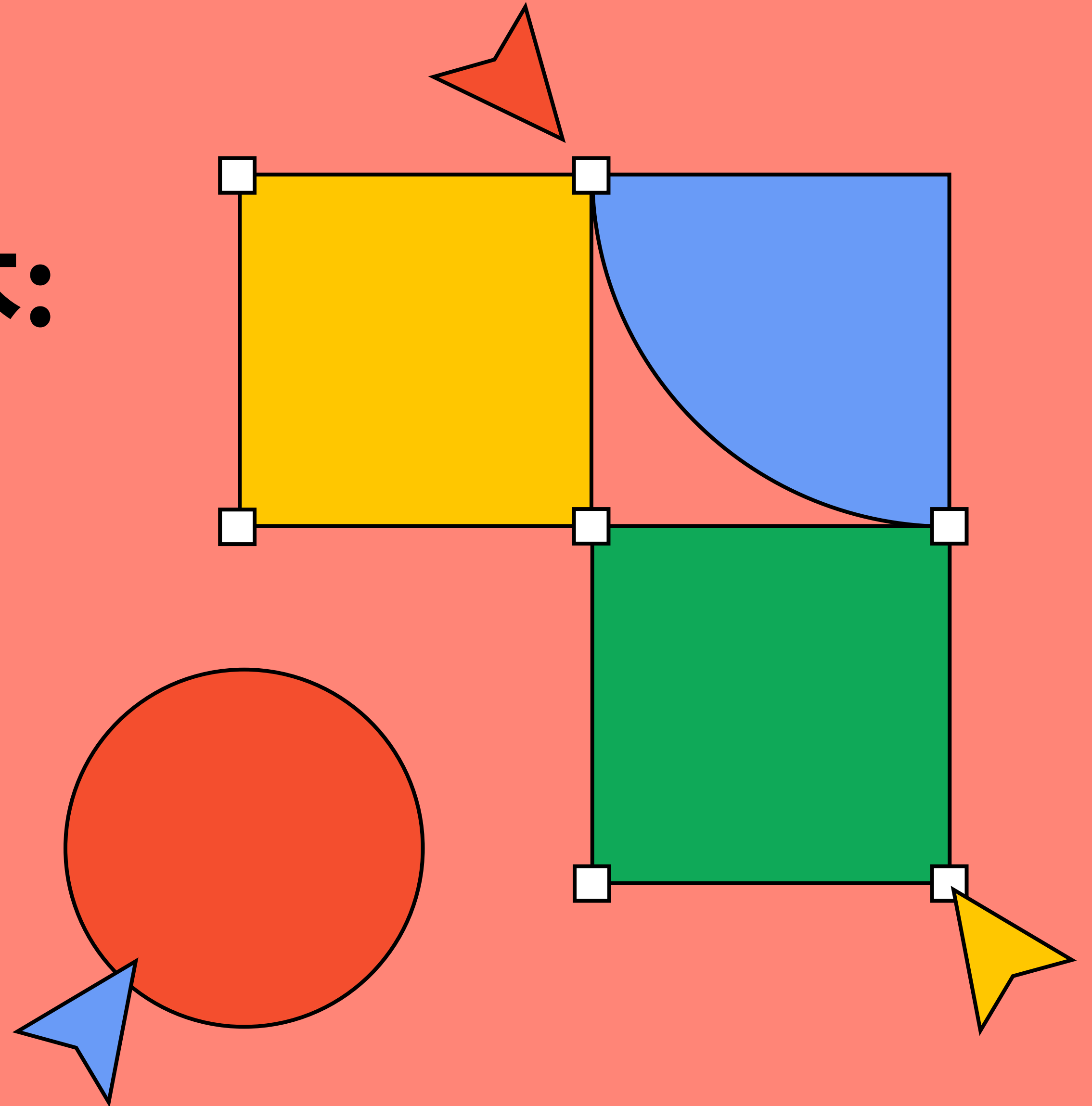


Figma

デザインシステムの基本: デザインシステムとは?



目次

デザインシステムとは一体何なのか? →

なぜデザインシステムを使うのか? →

デザインシステムの簡単な歴史 →

デザインシステムは必要か? →

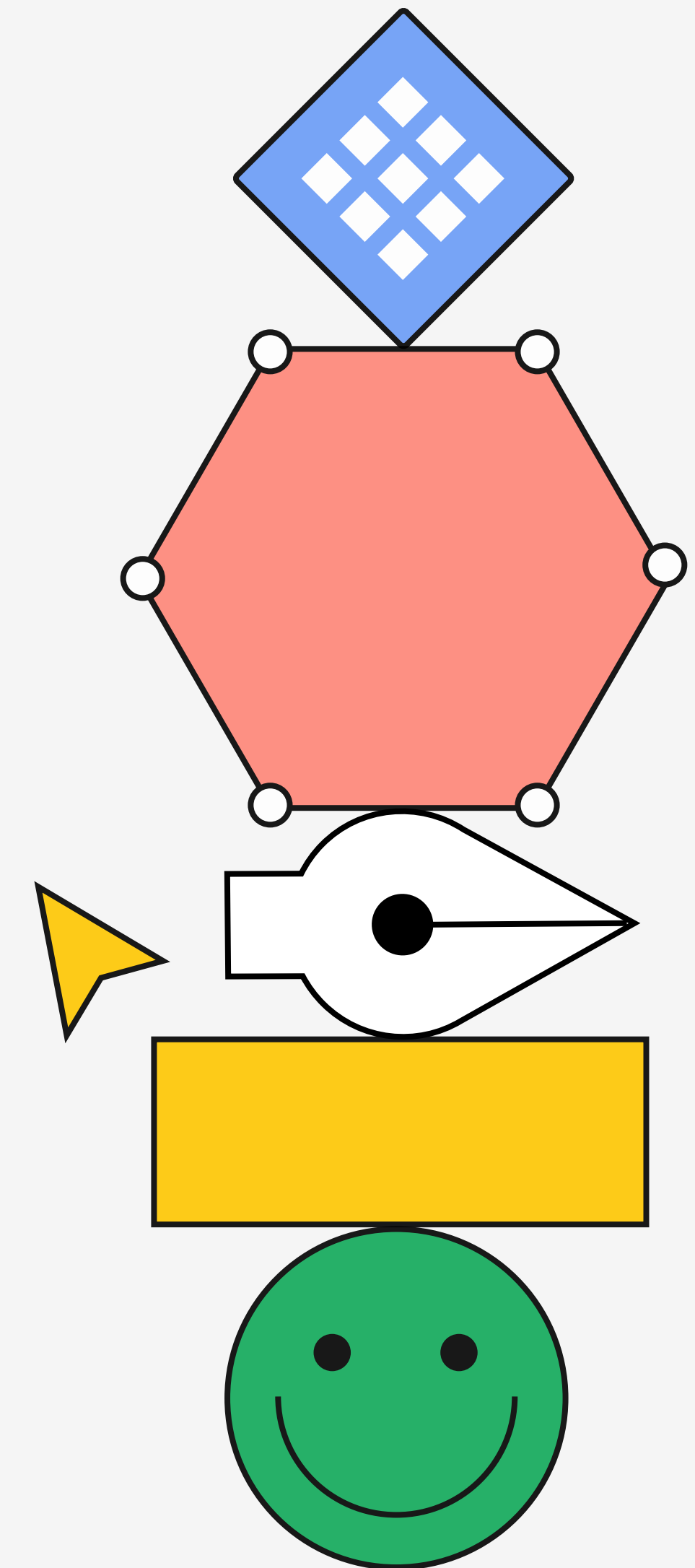
このシリーズは、デザインシステムの基礎を扱うデザインシステム入門コースを発展させたものです。

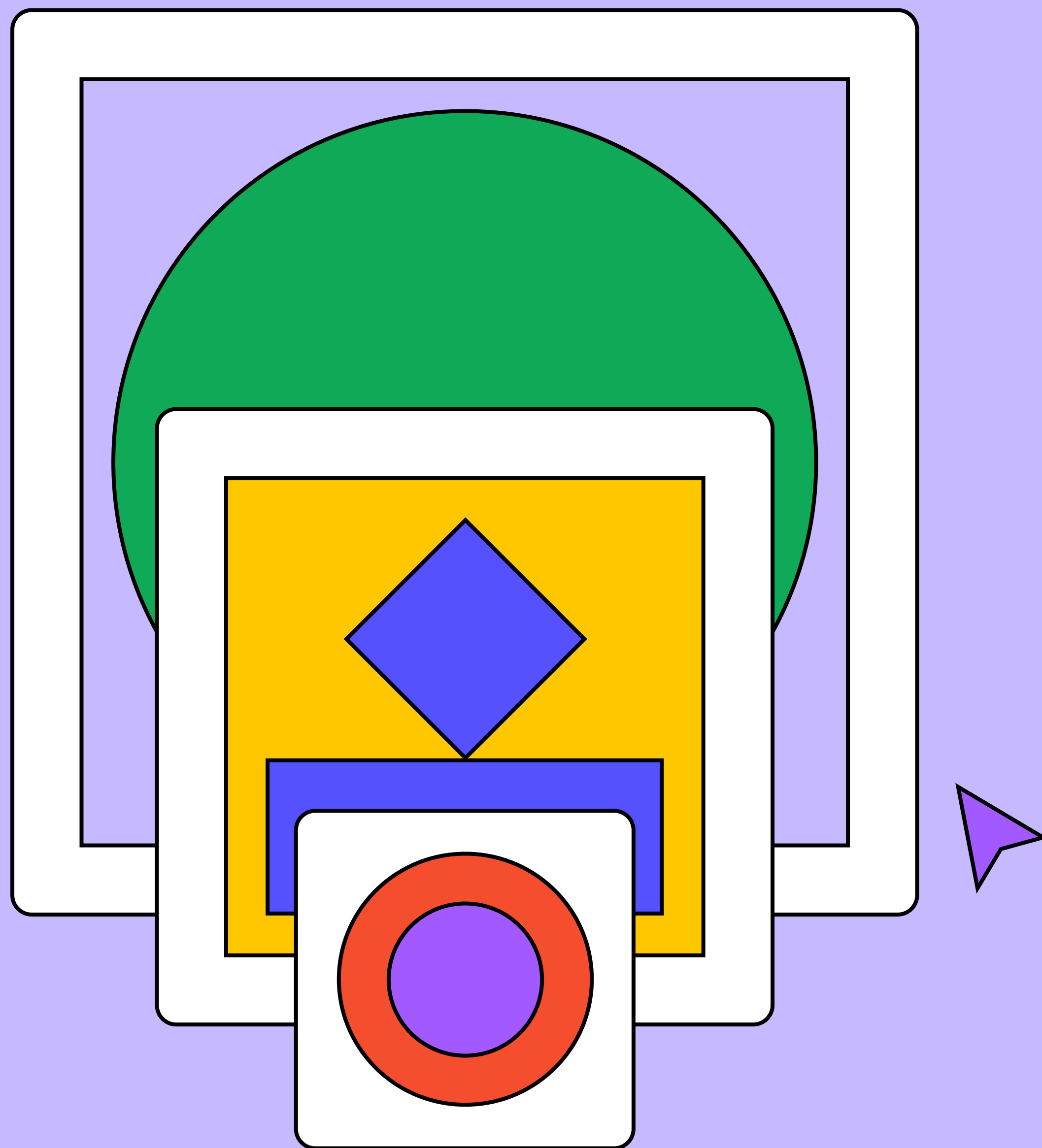
[YouTubeでもご覧いただけます →](#)

デザインシステムの基本、デザインシステムとは何か、デザインシステムはどのように機能するか、デザインシステムはデザイン方法をどのように変えることができるかについて学びましょう。

デザインシステムは、モバイルアプリやWebサイトからテレビ画面や車のダッシュボードのインターフェースまで、私たちが毎日使用する製品の縁の下で力持ちになっていることがよくあります。それは製品デザインのDNAとして機能し、ユーザーが操作するエクスペリエンスそのものを定義する原則と要素をコード化します。色とアイコンが意味を伝えます。ボタンは、一貫したサイズとシェイプで実装されます。明確で簡単に理解できる言葉が用いられます。デザインシステムがうまく機能していれば、エンドユーザーはデザインシステムをほとんど意識せずにすむのです。

デザインシステムがなければ、一貫性の欠如という危機に陥り、迷路を進むことになり、混乱、ブランドの希薄化、ユーザーの不満につながる可能性があります。このシリーズでは、デザインシステムの基本を説明し、デザインシステムが不可欠である理由を分析し、お客様のデジタルプレゼンスを高めデザインの混乱を防ぐデザインシステムの作成をご案内します。





デザインシステムとは一体何なのか？

デザインシステムの中核となるのは、製品やエクスペリエンスのルック&フィールの一貫性を保つための一連の構成要素と標準です。これは、デジタル製品を作成する複雑なプロセスを通じてチームを導く、統一された言語と構造化されたフレームワークを提供する青写真と考えてください。

デザインシステムは、製品やインターフェースを大規模にデザイン・構築する際に、要素やパターンを繰り返し作成するのに費やす時間を短縮するのに役立ちます。



デザインシステムのコンポーネント

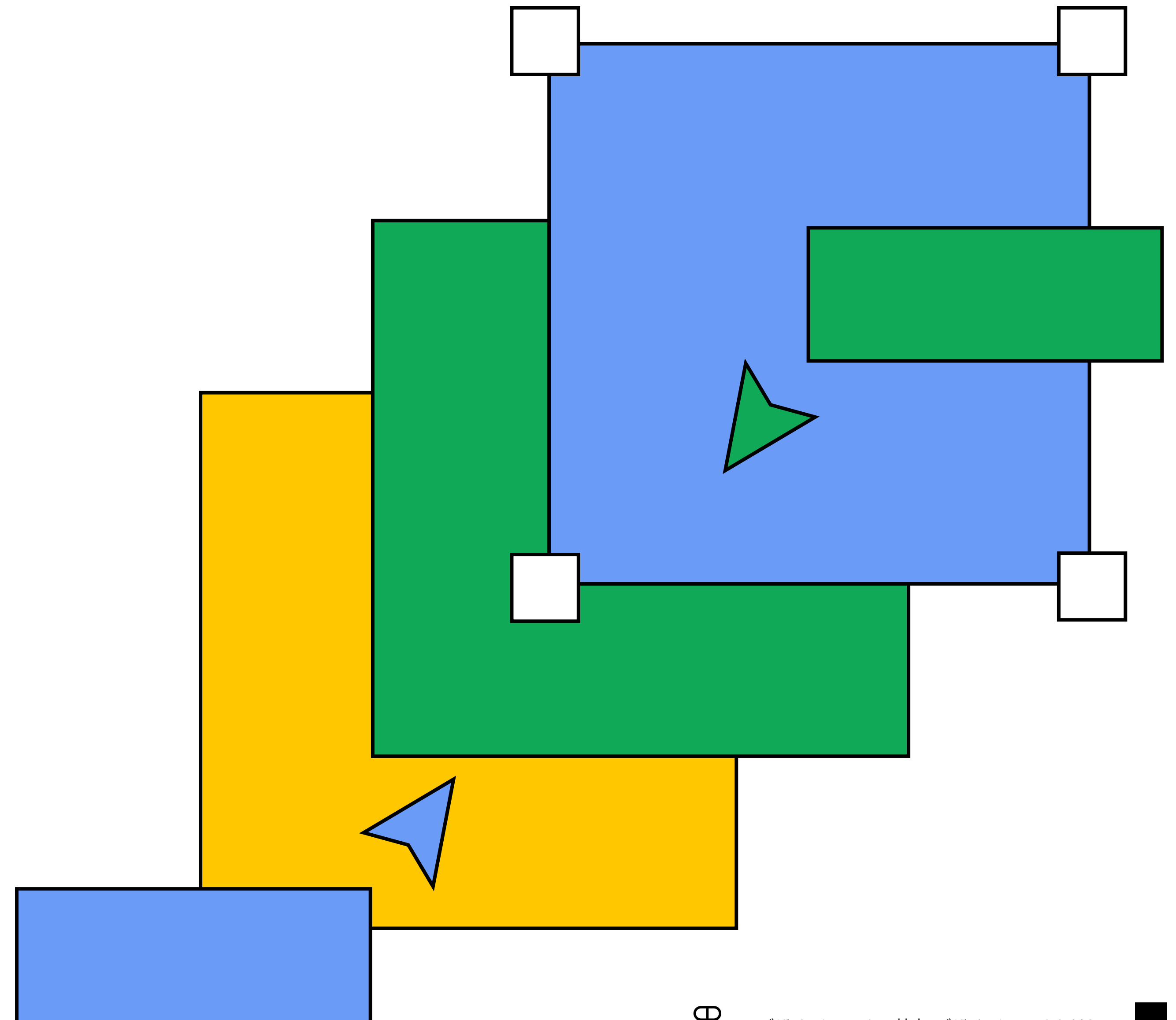
デザインシステムの階層は、次のように考えることができます。

1. デザインシステム

このトップレベルのコンテナは、包括的なガイダンスであり、製品エコシステムのリソースとプロセスの、進化し続ける基本的なコレクションです。これには、技術仕様、デザイントークン、ドキュメント、ベストプラクティスを含めることができます。また、UXデザインの意思決定や製品開発の指針となる基本原則やプロセスを含めることもできます。

デザインシステムとスタイルガイドの区別:

デザインシステムとスタイルガイドはしばしば同じ意味で用いられますが、デザインシステムがコーディング標準やユーザビリティなど、より包括的なものを指すのに対し、スタイルガイドは、主に色、タイポグラフィ、画像などの視覚的要素に焦点を当てたサブセットです。



2. コンポーネントとパターンライブラリ

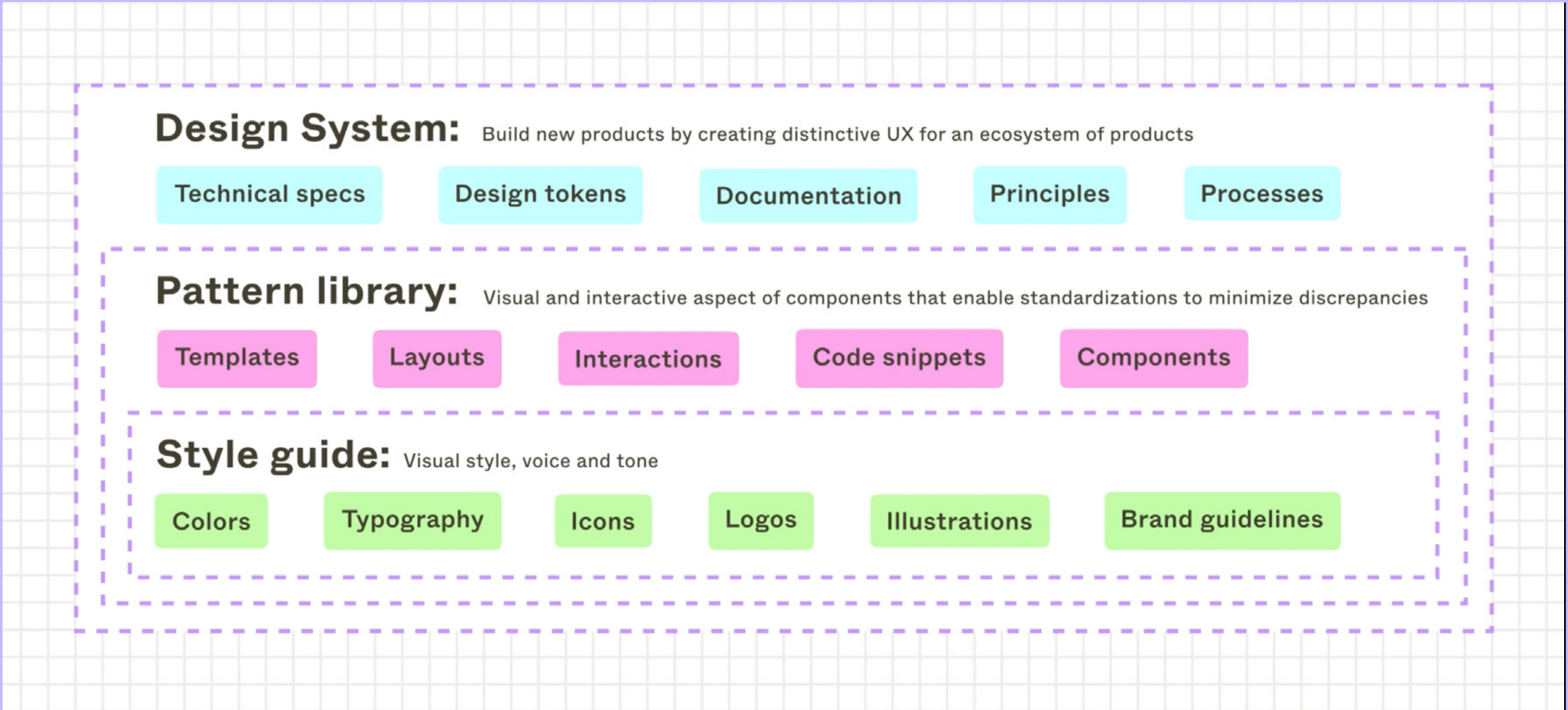
コンポーネントとパターンライブラリは、製品の共通のインターフェースや動作についての情報を伝える、再利用可能な視覚要素とインタラクションパターンです。テンプレート、レイアウト、インタラクションパターン、コードスニペット、コンポーネントなどが含まれ、詳細なドキュメントでサポートされています。

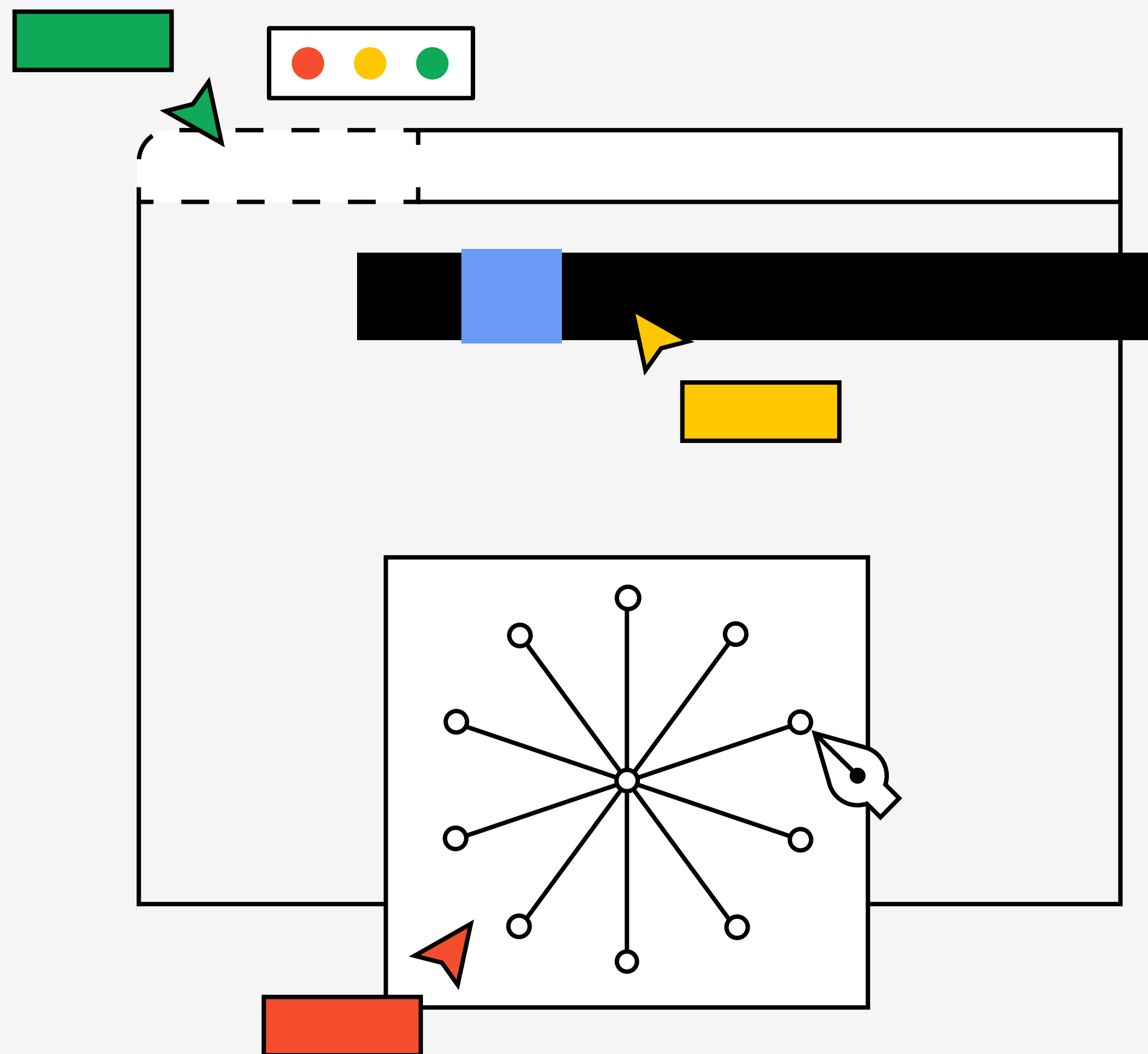
3. 基本要素

基本要素は、要素の外観や、色やタイポグラフィーなどの製品の全体的な調子やトーンなど、視覚的な言語を設定します。通常、アイコン、ロゴ、イラスト、アクセシビリティとブランドのガイドラインも含まれます。

コンポーネントとパターンライブラリの違い:

どちらもデザインシステムに不可欠ですが、コンポーネントライブラリはボタンや入力フィールドなどのUI要素に重点を置いているのに対して、パターンライブラリはナビゲーションフローやデータ表示などのより広範なデザインソリューションに対応しています。





デザインシステムとUX: ただの美学ではありません

デザインシステムは美学を抑圧し、デザイナーの創造性を制限し、デザイン全体に一律の均質性を強制するという俗説があります。デザインシステムは、繰り返すパターンを解決し、他の課題の解決に向かえるようにすることで、デザイナーをサポートするものです。

Figmaで共有・管理されているデザインシステムを使用することで、デザイナーは同じデザインを何度もコピー&ペーストすることなく、一貫したコンポーネントを再利用し、モードや画面サイズをシームレスに切り替えるバリエーションを設定できます。更新を1か所で行えば、システム全体に反映されるため、全員が同じ認識を持つことができます。こうしてデザイナーは解放され、次の問題に注意を向けたり、別の問題に深く取り組んだりすることができ、最終的にはより直感的でアクセスしやすく、楽しいエクスペリエンスを生み出すことができます。

誤解を解く:

デザイナーアドボケイトであるAna Boyerがデザインシステムの採用を妨げている可能性のある**6つの俗説を否定**。

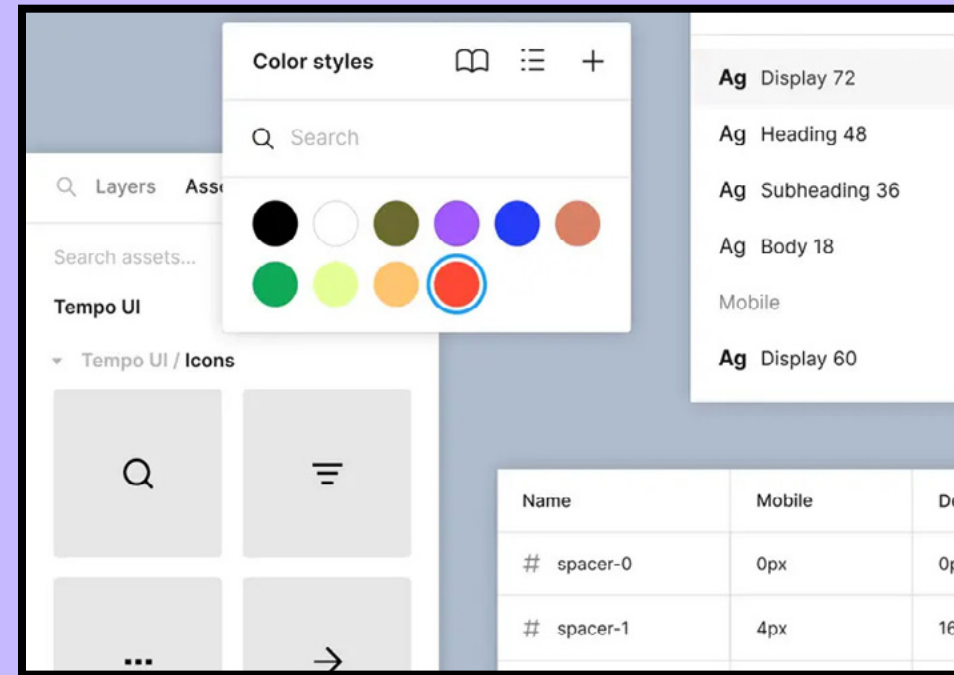


なぜデザインシステムを使うのか？

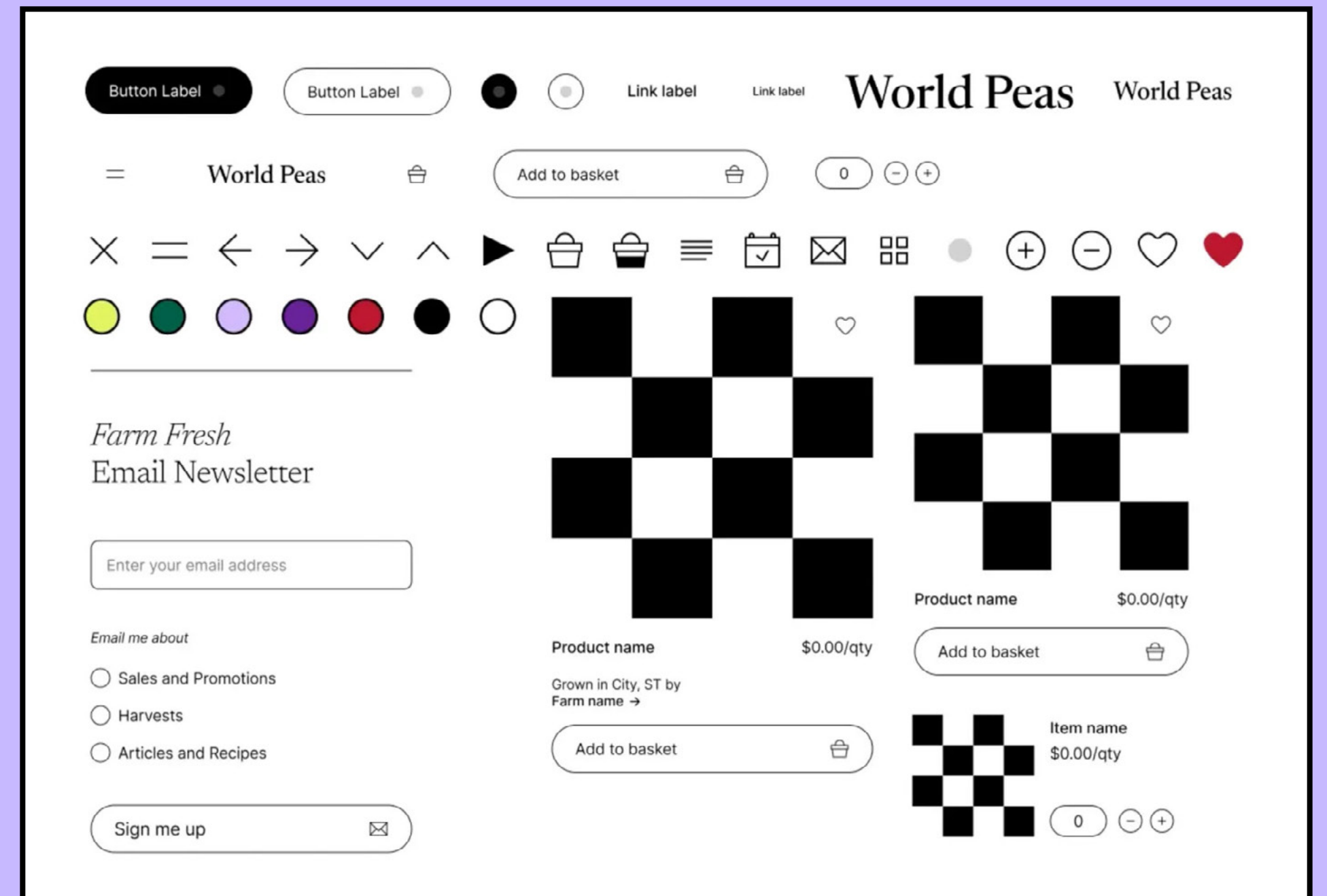
デザインシステムの影響力は、ワークフローを合理化し、製品全体の一貫性を確保し、部門横断的なチーム間のコラボレーションを促進する点にあります。小規模から始める場合でも、複数のプラットフォームに拡張する場合でも、デザインシステムを使用すると、機能のデザインだけでなく、大きなものを構築する場合でも、チームはより少ないリソースでより多くのことを行うことができます。

デザインシステムは、デザインの冗長性を減らし、開発プロセスを加速する唯一の確かな情報源です。デザインシステムを使用すれば、デザイナーはコンポーネントの作り直しに費やす時間を短縮し、代わりにブランドが承認した開発用オプションのライブラリから引き出して、デザインを迅速に構築できます。コンポーネントがコード、トークン、アニメーションプリセットを使用してデザインされている場合、開発者はそれらを機能的でアクセス可能なコードにわずかな時間で変換できます。デザインシステムを活用する企業は、製品開発ライフサイクルの変革を実際に経験し、市場投入までの時間を短縮し、まとまりのあるユーザーエクスペリエンスを実現しています。

新人デザイナーにとって、デザインシステムはオンボーディングツールとして機能します。デザインシステムから製品と原則について学び、より早く貢献できるようになります。システムが成熟するにつれて、デザインシステムはチームに共通のビジョンと言語を提供し、理解を深め、より一貫性のある製品を作ることができるようにします。これにより、製品を使用する人びととの信頼関係が深まり、時間の経過とともにより熱心で忠実な顧客ベースを築くことができます。



Figmaを使用すると、チームはスタイル、バリアブル、コンポーネントを標準化できるため、色からパディングまで、すべてを製品やブランド全体にシームレスに拡張できます。



コンポーネントライブラリには、製品の構成要素が含まれています。これには、個々のコンポーネント、レイアウトとテンプレート、インタラクションパターンが含まれる場合があります。



デザインシステムの簡単な歴史

デザインシステムのご概念は、テクノロジーそのものとともに成熟してきました。デザインシステムのルーツは、グラフィックデザインと印刷メディアの黎明期にさかのぼり、この時、スタイルガイドとタイプグラフィの標準によって、システムティックデザインのアイデアが初めて導入されました。20世紀半ばにブランディングが企業にとって不可欠になると、一貫したアイデンティティの必要性から、包括的なブランドガイドラインが策定されました。

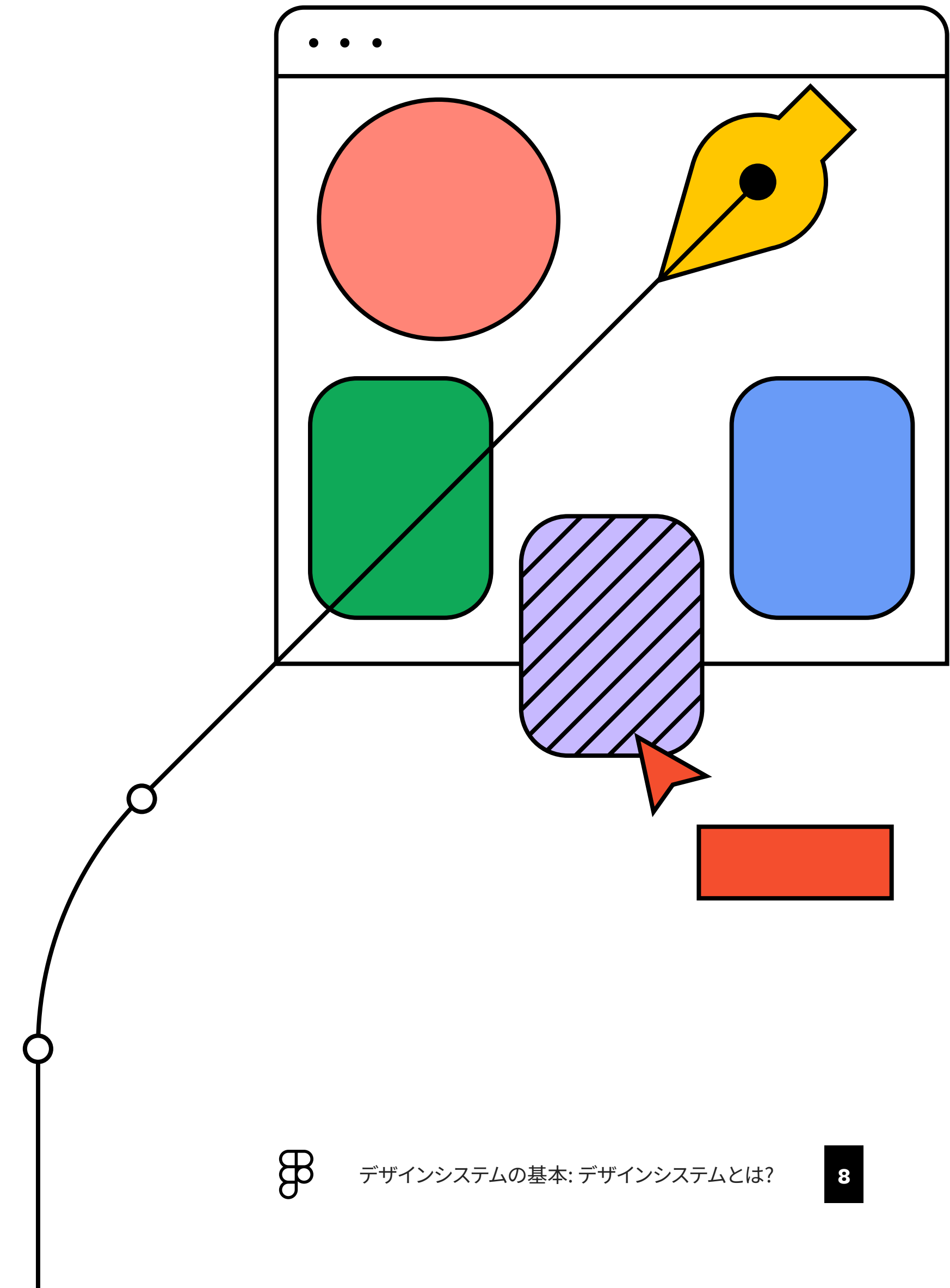
デザインシステムの誕生:

「システム」または「パターン」の使用について最初に言及されたのは、1960年代後半の **NATOソフトウェアエンジニアリング会議** においてであり、主にChristopher Alexander氏の功績とされています。Murray Silverstein氏、Sara Ishikawa氏との共著『A Pattern Language(パターンランゲージ)』は、建築における相互に関連し合うパターンについて論じており、今日私たちが「デザインシステム」と呼ぶものの起源であると考えられています。

デザインにとっての小さな一歩:

1976年1月にNASA長官Richard Truly氏によって前書き付きで発行された **NASA Graphics Standard Manual (NASAグラフィックス標準マニュアル)**。

デジタル革命により、これらのシステム原理は紙からピクセルに形を変え、広大なウェブや急成長するアプリ市場に合わせて複雑さを増しました。1987年のAppleに続いて、Google、IBM、Microsoftなどの大企業が、その後、無数のユーザーインターフェースのルック&フィールを決定することになる包括的なデザイン言語のパイオニアとなりました。これらのシステムは、美的統一性だけでなく実用性も重視して作成されており、開発をスピードアップし、UIデザインを合理化するための明確なドキュメントと再利用可能なパターンを提供し、デジタル製品チームの急増するニーズに対応しています。



2013年にBrad Frost氏の[アトミックデザイン手法](#)が導入されたことで、デザインシステムの構造は再定義されました。このアプローチは、UIコンポーネントに対する考え方に革命をもたらし、原子、分子、有機体へと明確な階層に整理しただけでなく、デザインチームと開発チームに共通の語彙を提供しました。

このシステムチックなデザインの探求は、デバイスプラットフォームの急速に変化する状況へのデザインチームの対応方法に影響を与え始めました。FigmaのデザイナーアドボケイトのChad Bergmanにとって、それは極めて重要な瞬間でした。

「私は[Twitter Bootstrapフレームワーク](#)の経験と、Bryan Haggerty氏、Laura Kalbag氏、Josh Clark氏の著作や講演、またBootstrapやZurb Foundationなどのフレームワークから影響を受け、標準的かつ再利用可能なアセットやドキュメントを作成するというこのアイデアを中心に、同僚のグループを結集し始めました。これが後に私が手掛けることになる多くのデザインシステムの最初のものになります」

このジャーニーは、2000年代初頭のパターンライブラリの作成からレスポンシブデザイン手法の採用まで、この分野のより広範なトレンドを反映しています。これらのマイルストーンは、現代のデジタルツールキットに不可欠なコンポーネントとしてのデザインシステムの重要性を明確に示しています。今日、デザインシステムは、デジタル製品の作成を導く包括的なリソースであり、視覚的なまとまりをつくるだけでなく、さまざまなデバイスやプラットフォームで直感的に使用できるよう配慮されています。初歩的なガイドラインから洗練されたフレームワークへのこの進化は、テクノロジーとデザインのダイナミックな相互作用を示しており、効率性、一貫性、革新性へのあくなき探求を印象付けています。

言語の学習:

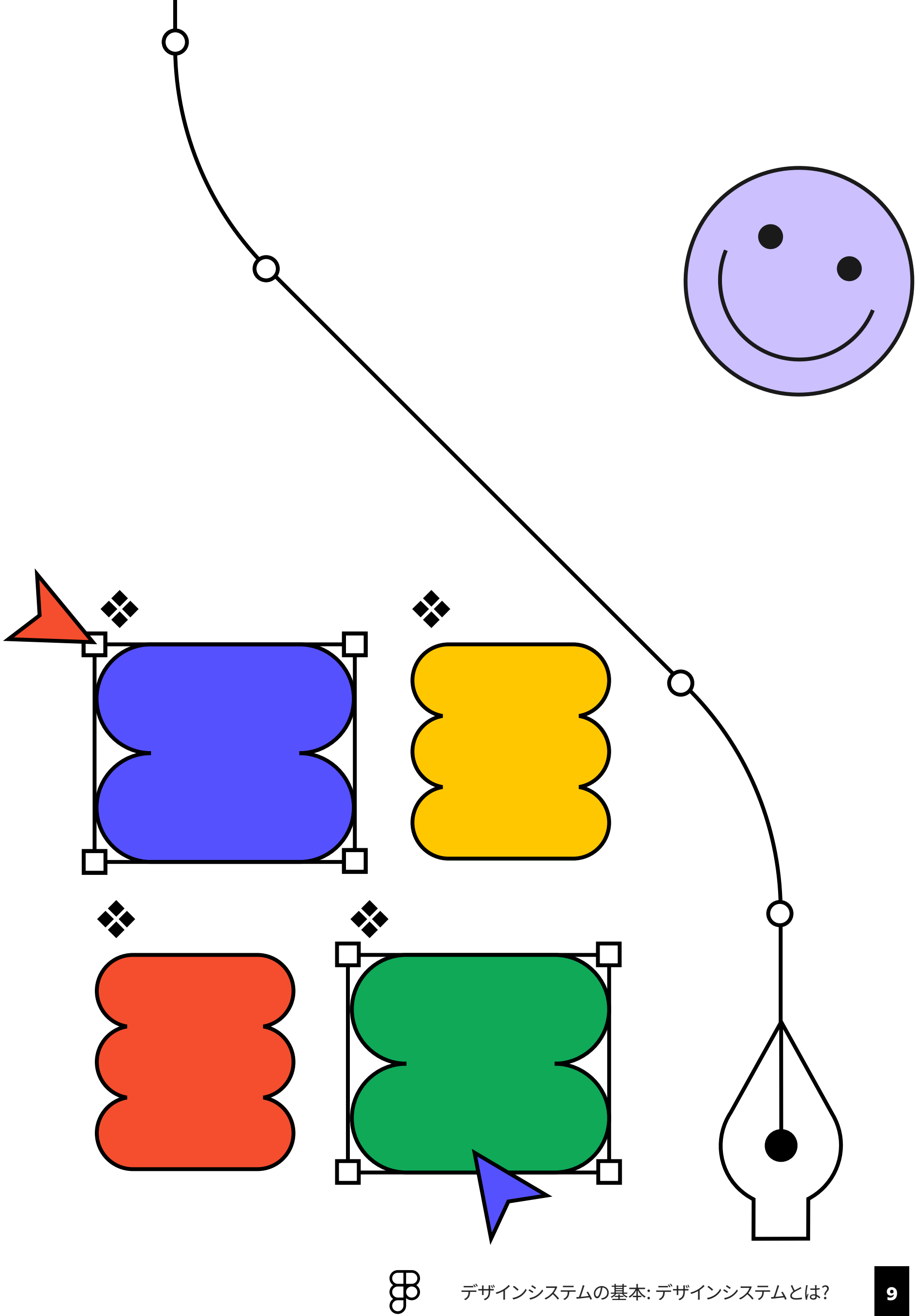
2006年、Yahoo!はYahoo! Design Pattern LibraryとYahoo! User Interface Library (YUI)を公開し、パターン言語への一般の人びとの関心を引き起こしました。GoogleのMaterial Designは、2014年に初めて自称「デザイン言語」になりました。[\(ウィキペディア\)](#)

言葉からパターンへ:

Rune Madsen氏は、[プログラミングデザインシステム](#)コースの一環として、[デザインシステムの歴史](#)を出版しました。

Building Blocks:

FigmaのデザイナーアドボケイトのChad BergmanとプロダクトマネージャーのJacob Millerが、デザインシステムに関するオフィスアワー、Building Blocksを主催しています。[YouTubeで過去のエピソードをご覧ください。](#)



デザインシステムが必要かどうかは、どのように判断するのか？

デザインシステムを実装するのに最適なタイミングは、厳密な科学によって特定されるわけではありません。むしろ、メリット、潜在的な課題、組織が直面する固有のハードルの組み合わせを考慮した上での戦略的な選択です。では、デザインシステムを実際を選ぶべき時とはどのような場合なのでしょうか？

デザインシステムを採用するかどうかは、特定のニーズに基づいて決定する必要があります。

不整合のチェック

さまざまなプラットフォームで製品のルック&フィールを精査してください。不快な不整合や、1つのエクスペリエンスから別のエクスペリエンスへと移行する際につながりやを欠くブランドアイデンティティを見つけたら、それはデザインシステムが必要かもしれないという兆候です。

多様なテーマやプラットフォームの検討

製品は、テーマ間でシームレスに切り替えたり(ダークモードとライトモード、異なるブランドなど)、さまざまなデバイスプラットフォームや画面サイズに適合させる必要がありますか？デザインシステムは、この複雑さを優雅に管理するための鍵となる可能性があります。

冗長性の削減

チームの時間を浪費する反復的なデザインタスクや、何度も発生する問題を把握しましょう。デザインシステムの標準化されたコンポーネントにより、プロセスを合理化できます。

コミュニケーションの強化

デザインに関するチームのコミュニケーション方法を確認しましょう。誤解が生じていないか、あるいは、デザイン要素を明確にするのに時間を浪費していないか？デザイン言語を共有することで、こうした難点を克服できます。

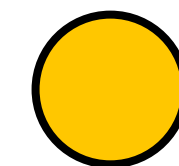
オンボーディングの簡易化

新しいチームメンバーのオンボーディングプロセスについて考えましょう。堅牢なデザインシステムにより、この移行をより迅速かつ効果的にを行い、全員がより早く貢献できるようになります。

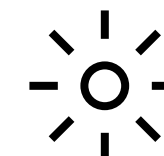
ライフサイクル効率の向上

製品のライフサイクルを振り返ってください。デザイン、プロトタイピング、更新をスピードアップする余地がある場合、デザインシステムは効率化のきっかけとなり、製品開発のあらゆる段階を強化することができます。

コミットする前に、チームの現在のワークフロー、製品のユーザーエクスペリエンス、解決が必要な特定の問題について評価しましょう。チームと徹底的に話し合うことで、ニーズに合わせたデザインシステムの必要性と潜在的な範囲を特定することができます。



Golden Yellow



Drop Shadow

Ag

Headline

一つのサイズがすべてに合うわけではない：

デザインシステムジャーニーを進める際には、すべてに適合するデザインシステムは存在しないことを念頭に置いてください。企業が異なればニーズも異なり、必要なソリューションも異なります。



デザインシステム導入時の課題の予測

デザインシステムには多くのメリットがありますが、課題がないわけではではありません。これらのハードルを理解することで、よりまとまりのある効率的なデザインプラクティスに向けたスムーズなジャーニーに備えることができます。

継続的なメンテナンス

まず、他の製品と同様に、デザインシステムにも、初期設定だけでなく、メンテナンスにも継続的な注意と労力が必要です。優れた製品が進化し続けるのと同じように、デザインシステムも真の意味で完成するということはありません。デザインシステムを最新かつ適切な状態に保つには、専用のリソースと時間が必要であり、それは長期的な取り組みになります。

後からやってくる満足感

デザインシステムの影響は計り知れませんが、その影響が明らかになっていくのには時間がかかります。このようにメリットが明らかになるのに時間を要することは、リーダーシップからの賛同を妨げることがあります。特にデザインシステムを導入するために、当面のプロジェクト目標からリソースやフォーカスを逸らせてしまう場合にそうなりがちです。目に見える見返りが明らかになる前に、時間の面でもチーム拡大の可能性の面でも、かなりの額の初期投資が必要となるなるかもしれません。

リソースの割り当て

適切なリソースを割り当てるためには、バランスを取ることが必要です。デザインシステムの効果を維持するためには、初期開発の段階を超えて、デザイナー、開発者、プロダクトマネージャーからの継続的なインプットが必要です。これは、優先順位を再考したり、デザインシステムが停滞しないようにチームを拡大したりすることを意味するかもしれません。

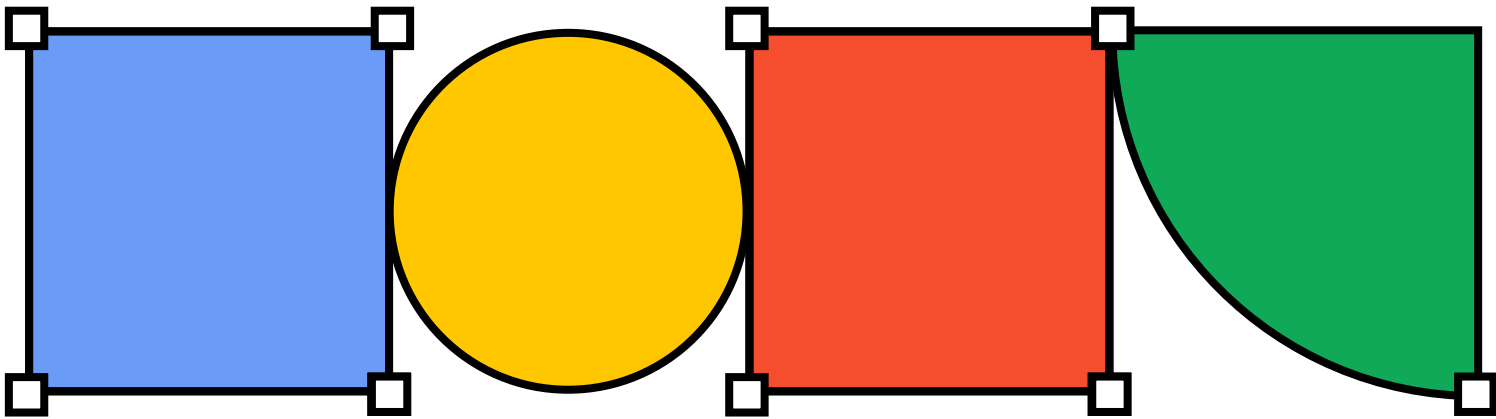
文化の変革

おそらく、より微妙で複雑な課題の1つは、組織全体でデザインシステムを統合するために必要な文化的な変化です。取り組むべきは、新しいツールやプロセスを採用することだけではありません。チームのコミュニケーション、コラボレーション、仕事に対する考え方を変えることも必要です。デザインシステムを組織の構造に本当の意味で組み込むには、その成功に尽力し、その価値を他の人に納得させることができるアドボケイト、つまり推進者が必要です。重要なのは、システムが成長し進化し続ける中で、コミュニティの変化するニーズや貢献を反映し、サービスを受け取る人びとによって形作られる生きたリソースになることです。

システムの社会化

デザインシステムに全員の参加を取り付ける過程は、上り坂のように感じるかもしれません。これを行うには、さまざまな利害関係者にその長期的な価値を教育し、納得させることが含まれます。そのためには、明確なコミュニケーション、その影響の実証、広範な採用のための戦略が必要です。これは多くの場合、デザインシステムを中心に、さまざまな部門や役割にまたがるコミュニティを作り、育てることを意味します。

これらの課題を前もって認識し、その対処方法を戦略化することで、デザインシステムの効率化を図り、人びとに受け入れてもらうための準備ができます。そのジャーニーは複雑かもしれませんが、その目的地(より一貫性があり、効率的でスケーラブルなデザインプラクティス)を考えれば、努力する価値が十分にあります。



切り替えをお考えですか？

デザイナーアドボケイトのClara Ujiieによる「**An insider’s guide to a seamless Figma migration (シームレスなFigma移行のためのインサイダーのガイド)**」では、Figmaに移行するための豊富なリソース、ヒント、ツールが紹介されています。



チーム全員のためのデザインシステム

チーム全体がアクセスすることのできる共有アセットやライブラリを作りましょう。Figmaのウェブベースのマルチプレイヤープラットフォームを使えば、デザインや開発を簡単にスケールさせることができます。

[Figmaのデザインシステム →](#)

