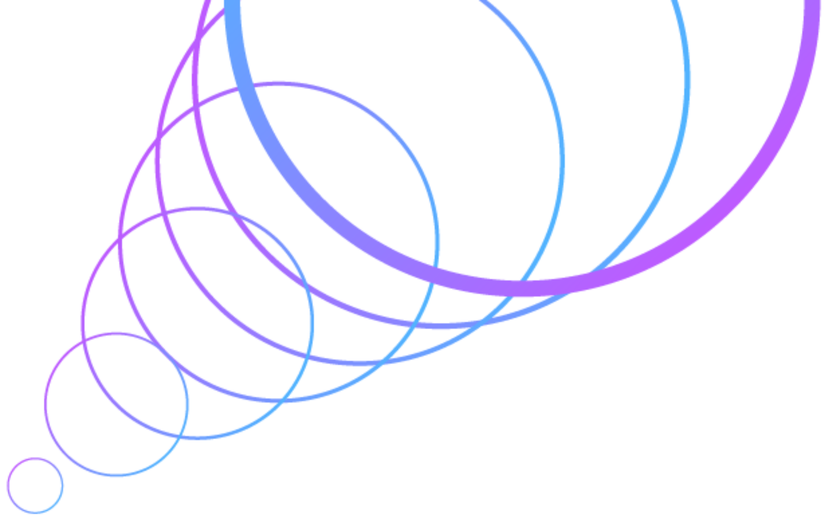
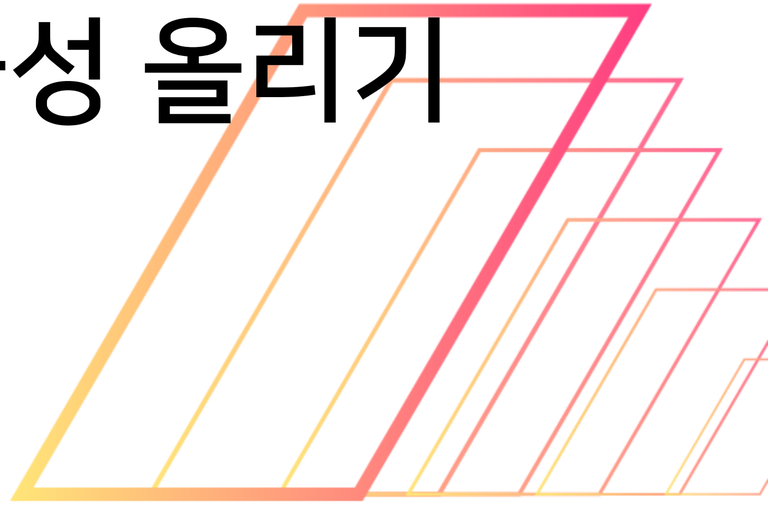




DEVIEW 2021

피그마 API로 하나 되는 우리

: 피그마 API를 활용해 협업 효율성 올리기

임송이 데이터라이즈

CONTENTS

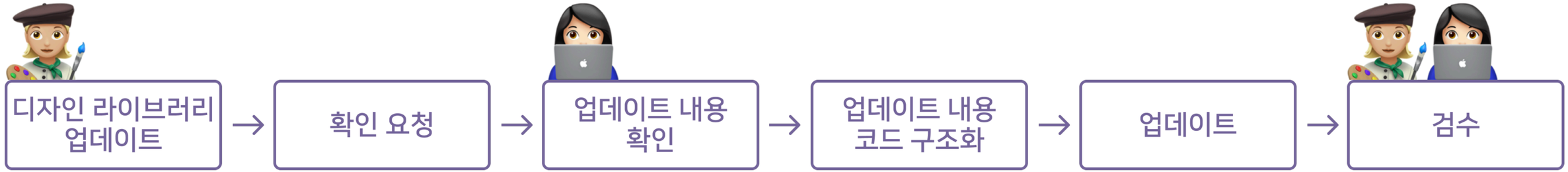
고민의 시작

고민 1. 커뮤니케이션 비용

DEVIEW 2021

작업 과정에서 요청, 확인, 반영 후 알림 등 커뮤니케이션 비용이 지속적으로 발생.

Before

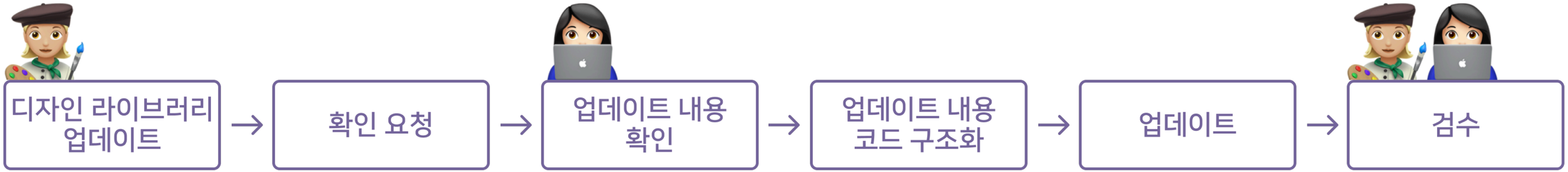


해결 1. 디자인 토큰 구축과 자동 업데이트

DEVIEW 2021

업데이트 후 확인, 반영해야 하는 단계를 단축해서 커뮤니케이션 비용을 줄임.

Before



After

고민 2. 작업 효율성

DEVIEW 2021

- 피그마를 활용한 작업과정에서 다양한 비효율 발생
  - 반복 작업, 사람이 일일이 수정하는 작업 등 필요하지만 효율이 아쉬운 부분이 발생
  - 여러 피그마 파일이 생겨나면서 파일의 통일성이 조금씩 달라지기 시작.

해결 2. 내부 피그마 플러그인 구축

작업의 효율성을 높일 수 있는 플러그인 구축

플러그인이란?

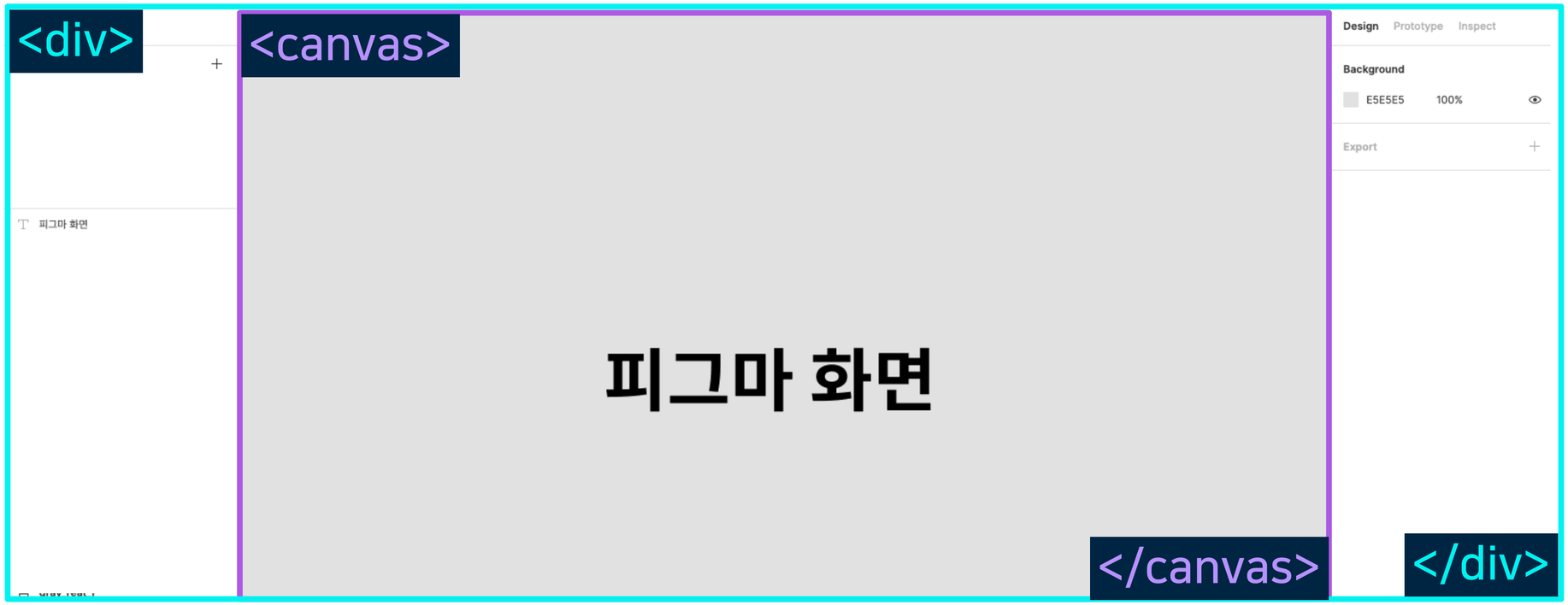
피그마에서 제공하는 기능 외에 유저가 커스텀으로 기능을 확장해서
사용할 수 있도록 피그마에서 제공하는 써드 파티 기능.

피그마 & 피그마API란?

피그마의 특징

DEVIEW 2021

- 웹 브라우저 기반의 디자인 툴.
- 피그마에서 제공하는 API를 통해 피그마 파일의 데이터를 가져오고 요소를 렌더링할 수 있음



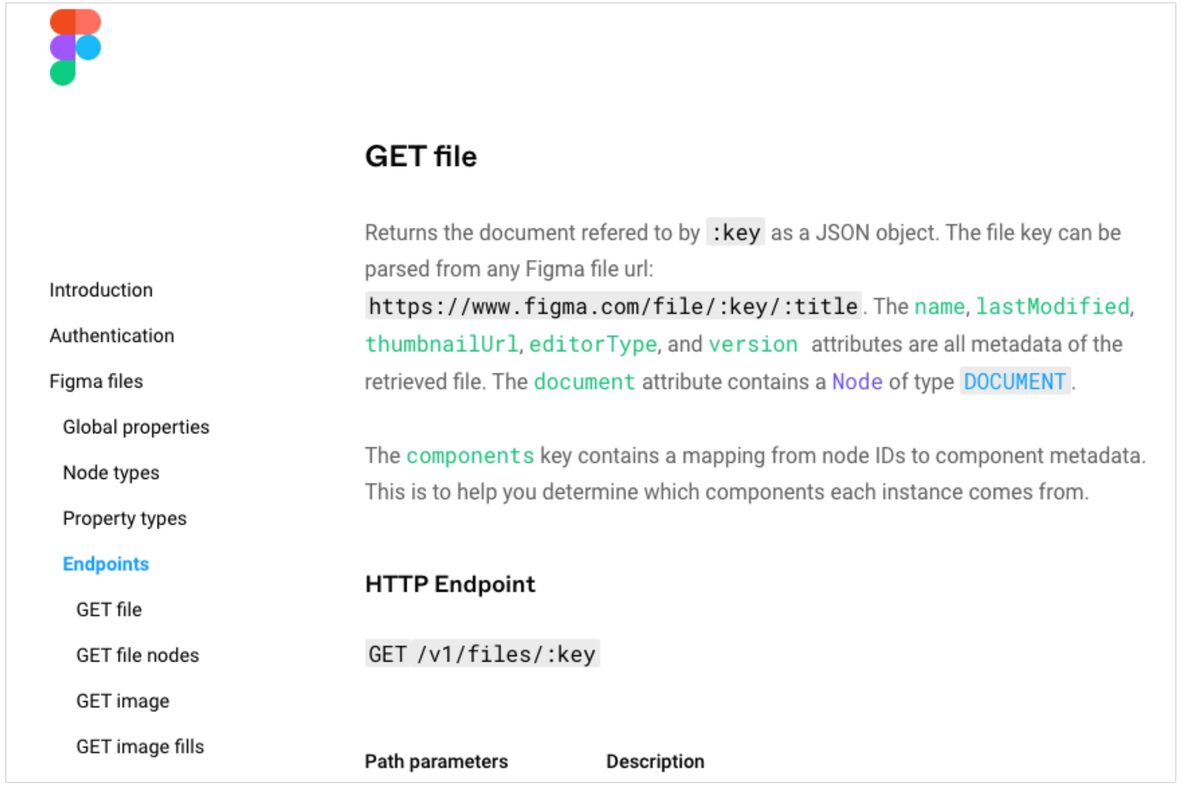
피그마에서 제공하는 API

DEVIEW 2021

Figma API

https://www.figma.com/developers/api

- 피그마 파일에 접근해 파일, 노드, 이미지, 주석 등 다양한 데이터를 가져올 수 있음
- 주로 GET만 가능한 API가 많음
- 피그마 파일 내에서 이벤트를 감지하는 Webhook API가 베타 오픈.
- 이번 프로젝트에서 주로 활용한 API는 FILE API



이미지 : www.figma.com/developers/api/ 문서 캡쳐

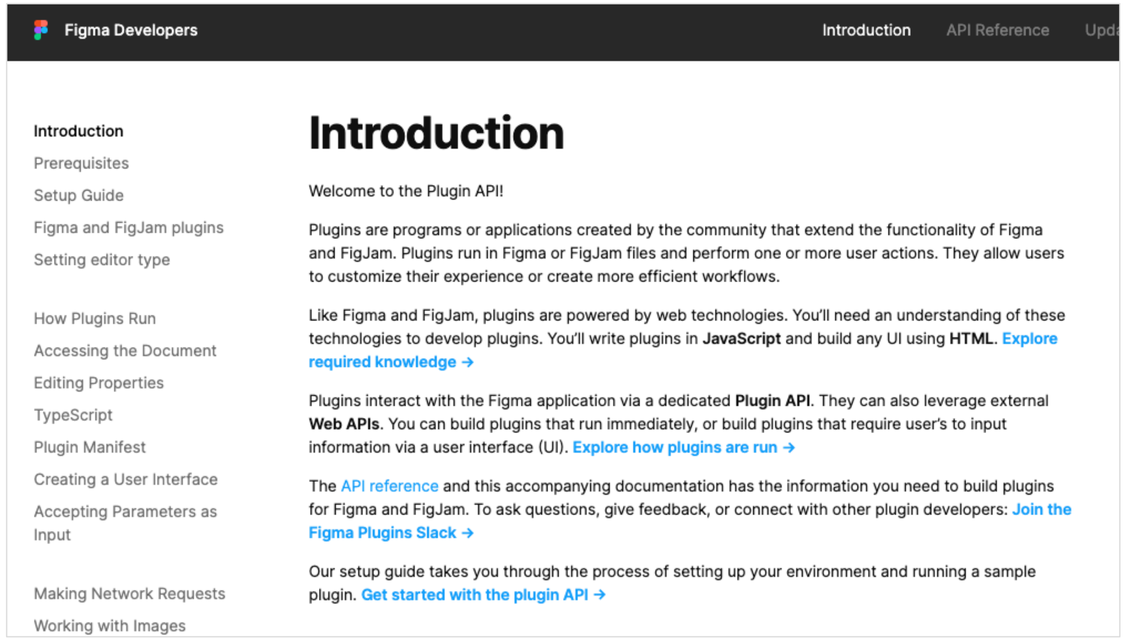
피그마에서 제공하는 API

DEVIEW 2021

Figma Plugin API

https://www.figma.com/plugin-docs

- 피그마 플러그인 제작을 위한 API
- 피그마 화면의 요소를 생성하거나 제어하는 기능 제공



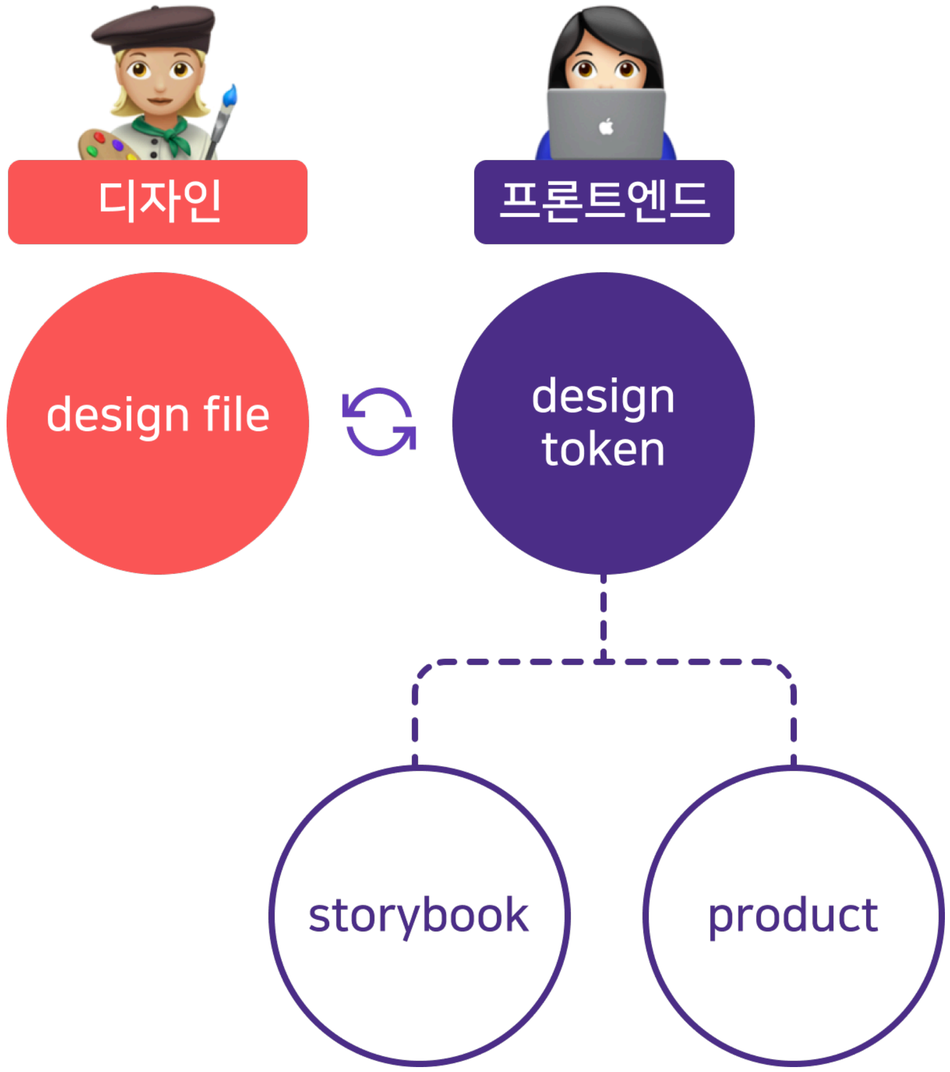
이미지 : www.figma.com/plugin-docs/ 문서 캡쳐

활용 1.

디자인 시스템 구축에 활용하기

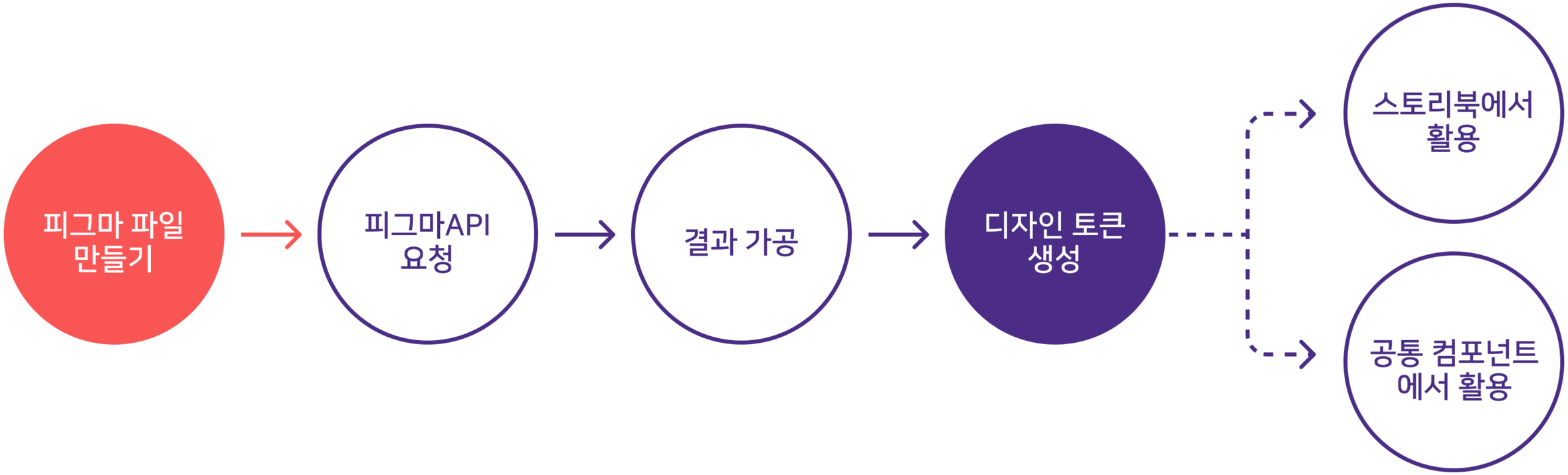
디자인 시스템 구조

- 피그마 디자인 파일과 디자인 토큰이 한 세트
- 디자인 토큰을 업데이트하면 토큰을 바라보는 모든 파일에 업데이트가 반영됨.



전체 과정

피그마 파일에서 추출한 데이터를 하나의 디자인 토큰 파일로 가공해 활용



1. 피그마에 디자인 라이브러리 구축하기

DEVIEW 2021

[1] 하나의 페이지 = 하나의 디자인 카테고리

피그마 화면	피그마 내부	JSON
File	DocumentNode	type: Document
Pages		
COLOR	PageNode	type: CANVAS
TYPOGRAPHY	PageNode	type: CANVAS
BUTTON	PageNode	type: CANVAS
INPUT	PageNode	type: CANVAS

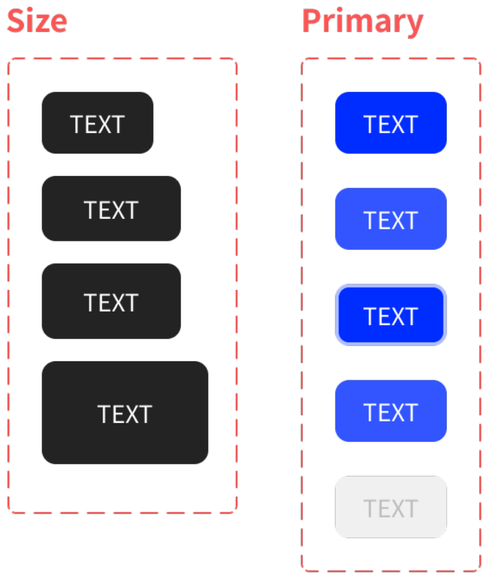
1. 피그마에 디자인 라이브러리 구축하기

DEVIEW 2021

[2] 각 페이지 안에 디자인 컴포넌트 만들기

디자인 라이브러리

디자인 토큰 피그마



1. 피그마에 디자인 라이브러리 구축하기

DEVIEW 2021

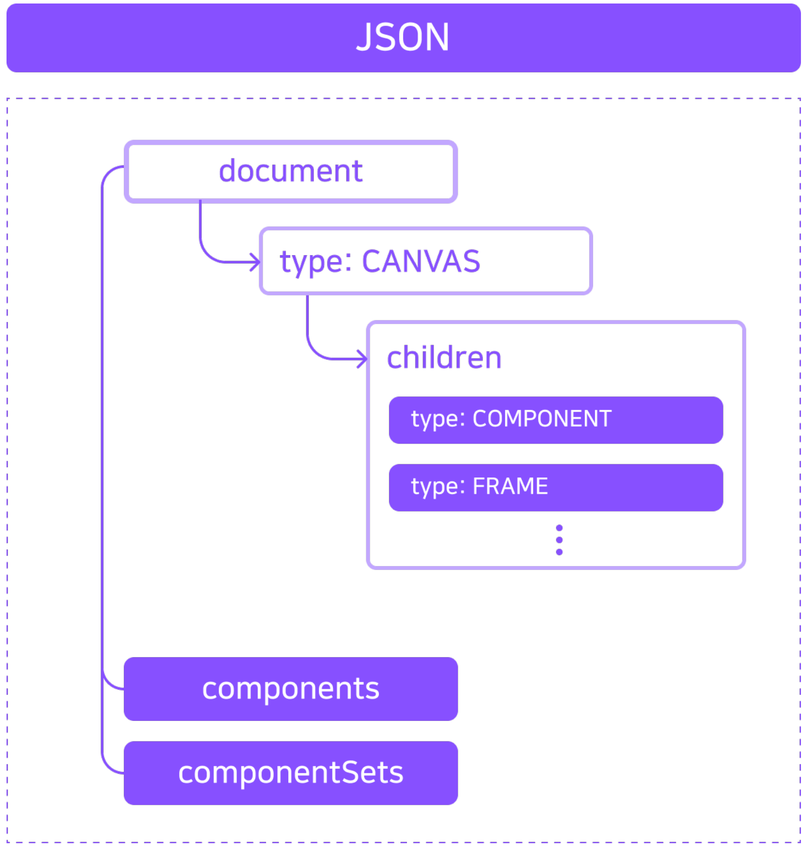
[3] 피그마의 Frame과 Component의 차이 이해하기

Frame

- 크기가 설정된 피그마의 레이어
- API 결과값에서 Frame 데이터는 페이지 노드 하위에 있음

Component

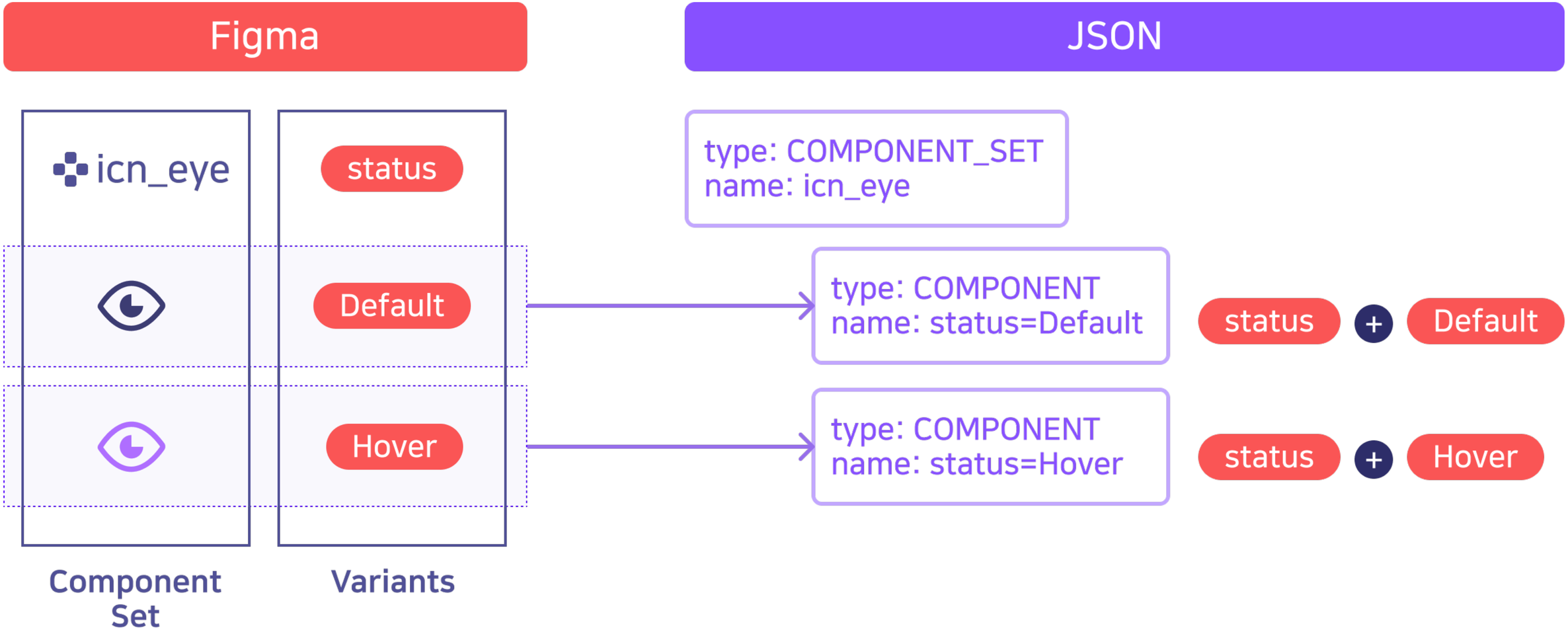
- 디자인을 재활용할 수 있는 틀
- 기준이 되는 요소가 변경되면 해당 요소에 즉시 반영
- API 결과값 최상단에서도 컴포넌트 리스트를 확인할 수 있음



1. 피그마에 디자인 라이브러리 구축하기

DEVIEW 2021

[4] 피그마의 Component와 Variants



2. API 요청하기

DEVIEW 2021

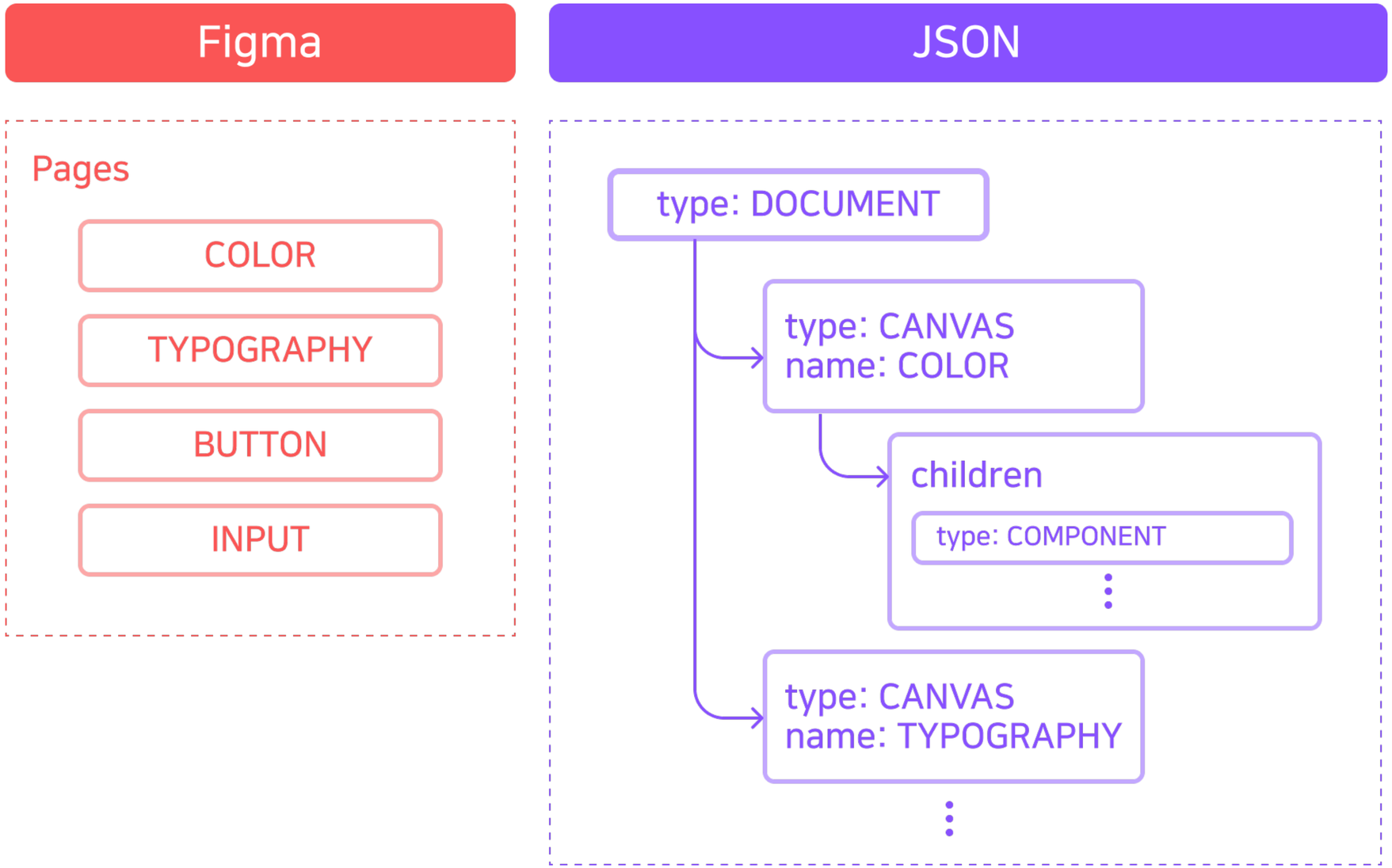
[1] API 호출하기

- 피그마 File API를 호출해서 피그마 파일의 모든 Node의 데이터를 받아옴.
- API 요청할 때 인증 정보와 파일 키가 필요함.
    - 피그마에서 발급받은 Access Token 또는 OAuth2 인증
- 피그마 파일 키는 피그마 파일 공유 링크에서 확인할 수 있다.

링크 주소 형식

https://www.figma.com/file/figma_file_key/file_name

2. API 요청 결과


Figma
JSON
Pages
COLOR
TYPOGRAPHY
BUTTON
INPUT
type: DOCUMENT
type: CANVAS
name: COLOR
children
type: COMPONENT
type: CANVAS
name: TYPOGRAPHY

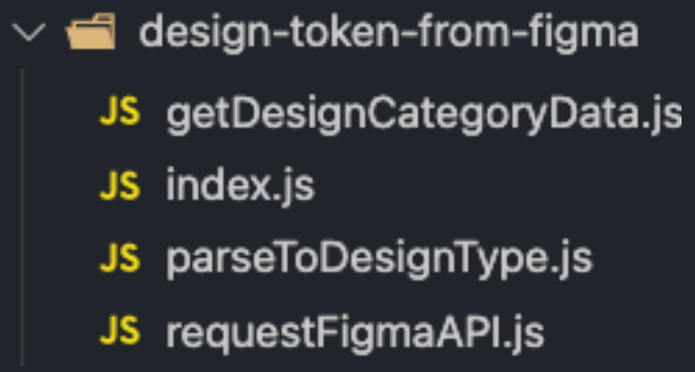
3. 데이터 가공하기

DEVIEW 2021

1. 데이터를 디자인 단위로 쪼개기
2. 페이지 단위에서 컴포넌트 분리
3. 컴포넌트 유형별로 필요한 요소 추출
4. 추출한 항목 가공하기
5. 전체 데이터 합치기

3. 데이터 가공하기

DEVIEW 2021



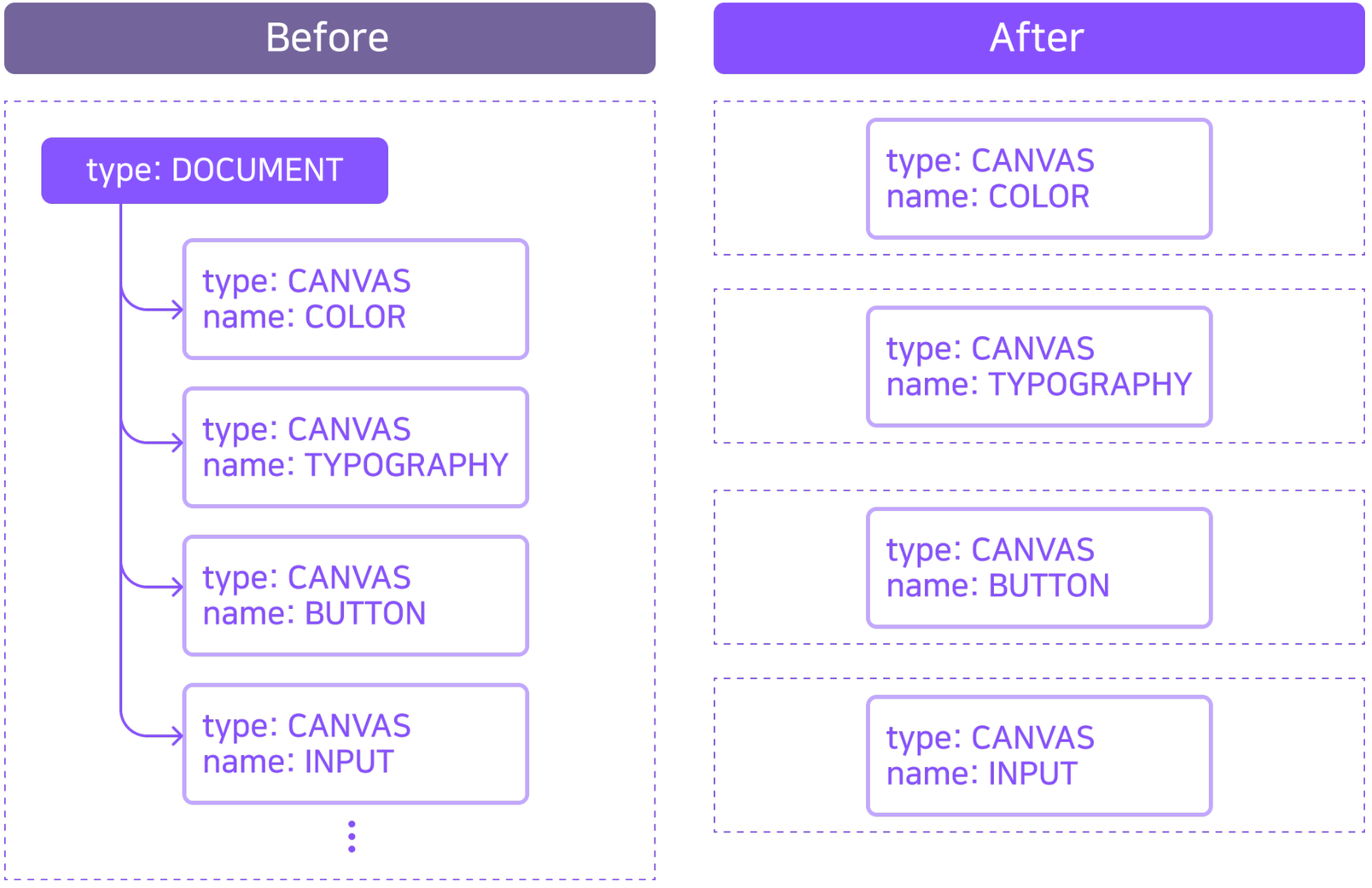
index.js	
requestFigmaApi.js	API 요청하기
getDesignCategoryData.js	페이지/컴포넌트 단위로 데이터 쪼개기
parseToDesignType.js	디자인 유형별로 데이터 가공함수 적용하기

3. 데이터 가공하기

DEVIEW 2021

[1] 데이터를 디자인 단위로 쪼개기

2 3 4 5



3. 데이터 가공하기

DEVIEW 2021

Figma

[Page]
Button

JSON

type: CANVAS
name: BUTTON

Primary

Secondary

Size

Children

type: COMPONENT_SET
name: Size

- type: COMPONENT
  name: status=Height36
- type: COMPONENT
  name: status=Height42

type: COMPONENT_SET
name: Primary

- type: COMPONENT
  name: status=Default
- type: COMPONENT
  name: status=Hover

type: COMPONENT_SET
name: Secondary

- type: COMPONENT
  name: status=Default
- type: COMPONENT
  name: status=Hover

name: status=Height36

name: status=Default

3. 데이터 가공하기

DEVIEW 2021

1 | 2 | [3] 컴포넌트 유형별로 필요한 요소 추출 | 4 | 5

Figma

TEXT
- borderColor
- backgroundColor
- labelColor

TEXT
- borderColor
- backgroundColor
- labelColor

TEXT
- borderColor
- backgroundColor
- labelColor

TEXT
- borderColor
- backgroundColor
- labelColor

TEXT
- borderColor
- backgroundColor
- labelColor

JSON

type: COMPONENT_SET
name: Primary

type: COMPONENT
name: status=Default

'status=Default'가 가진 속성 추출 O 추출 X

fills | strokes | strokeWeight | layoutMode

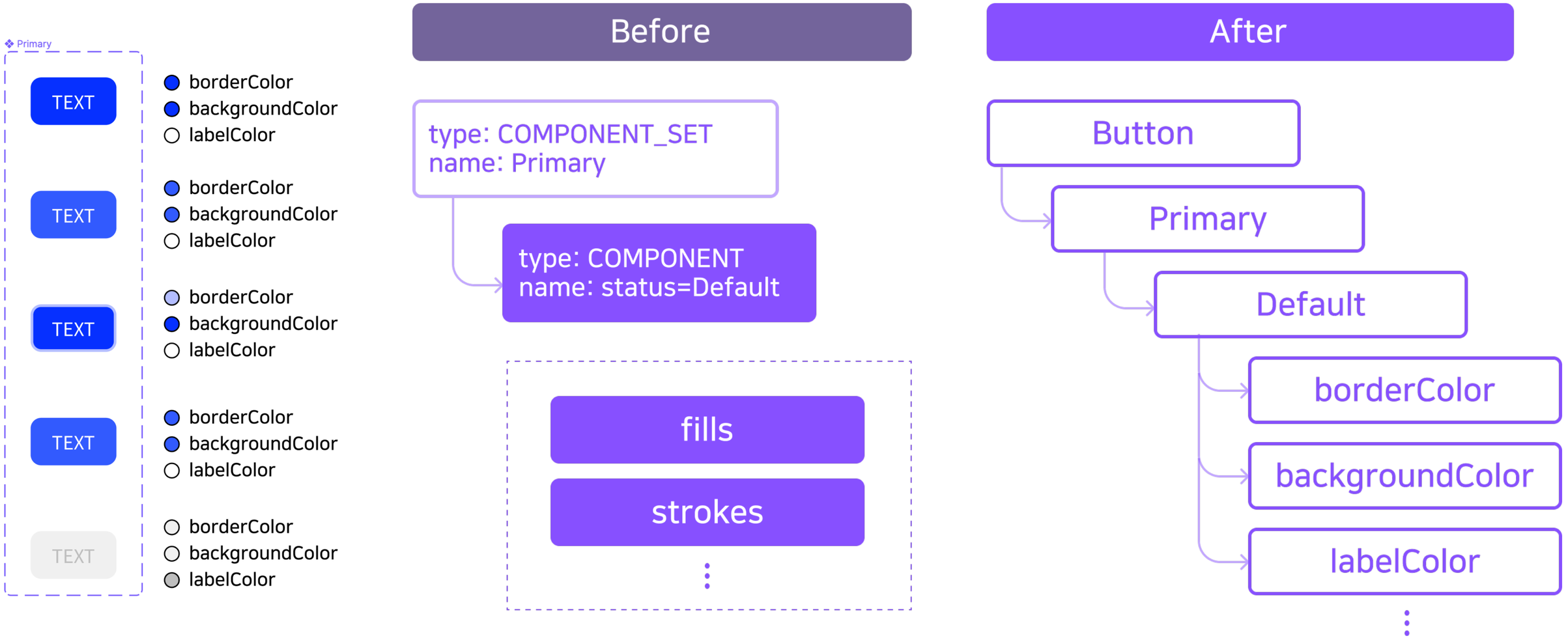
absoluteBoundingBox | counterAxisAlignItems | primaryAxisAlignItems

⋮

3. 데이터 가공하기

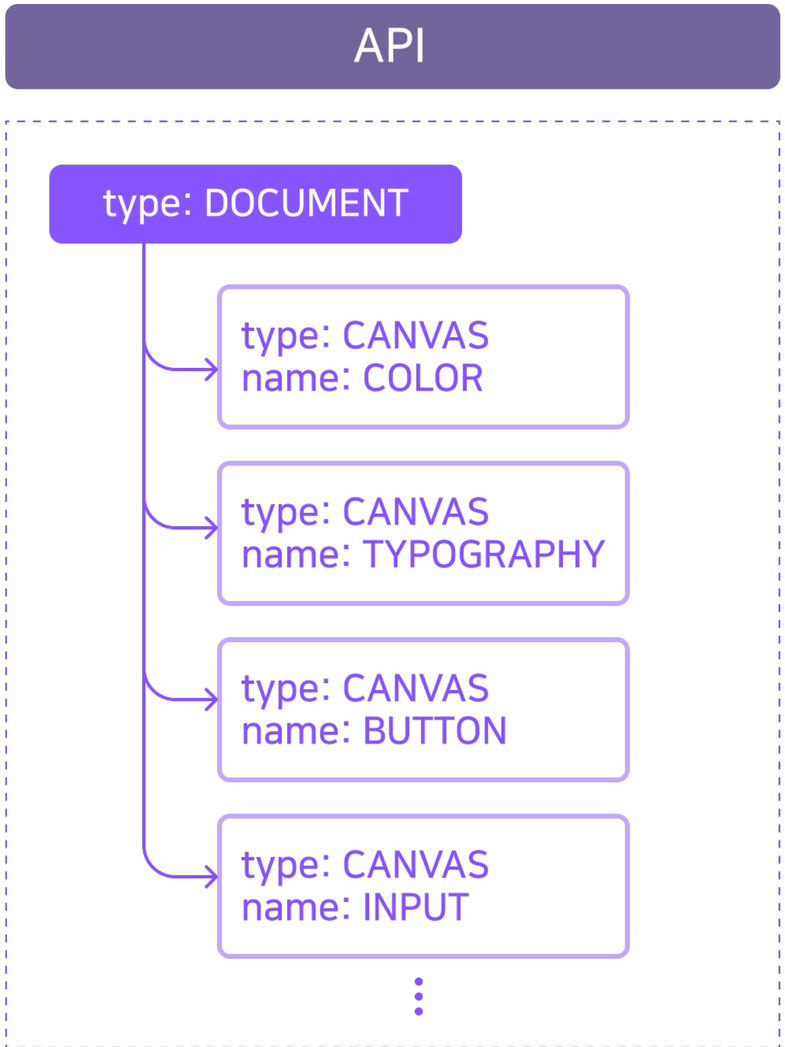
DEVIEW 2021

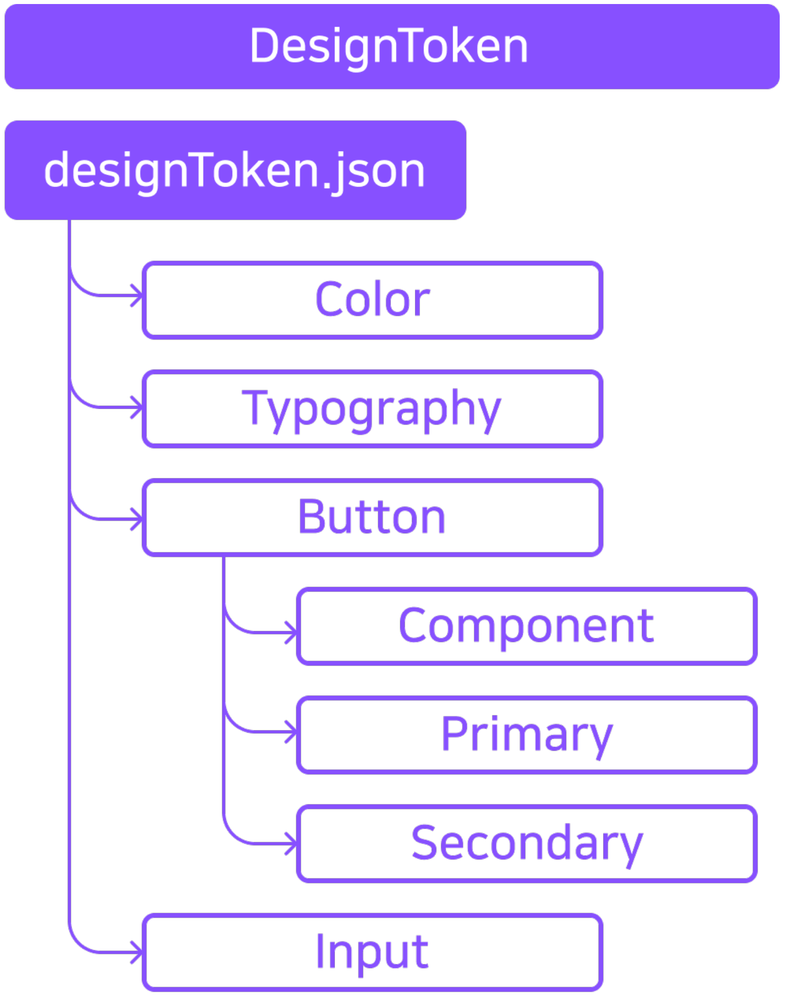
1 | 2 | 3 | [4] 추출한 항목 가공하기 | 5



3. 데이터 가공하기

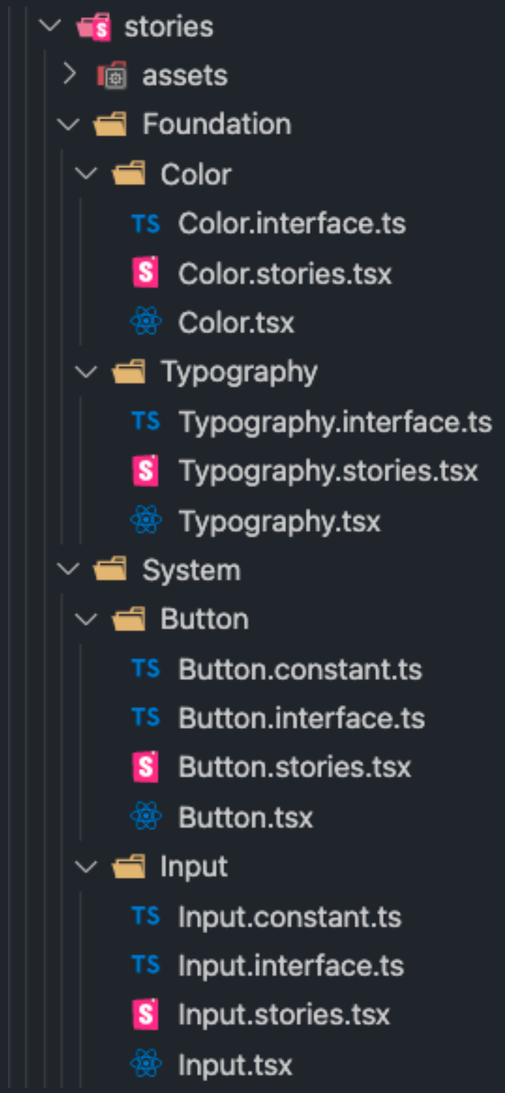
1 2 3 4 [5] 전체 데이터로 재병합





4. 활용하기 : 스토리북의 전체 구조

DEVIEW 2021



기준: React / TypeScript

파일 구조

DesignCategoryFolder

DesignCategory.stories.tsx

DesignCategory.tsx

-스토리북에서 렌더링하는 공통 컴포넌트를 작성

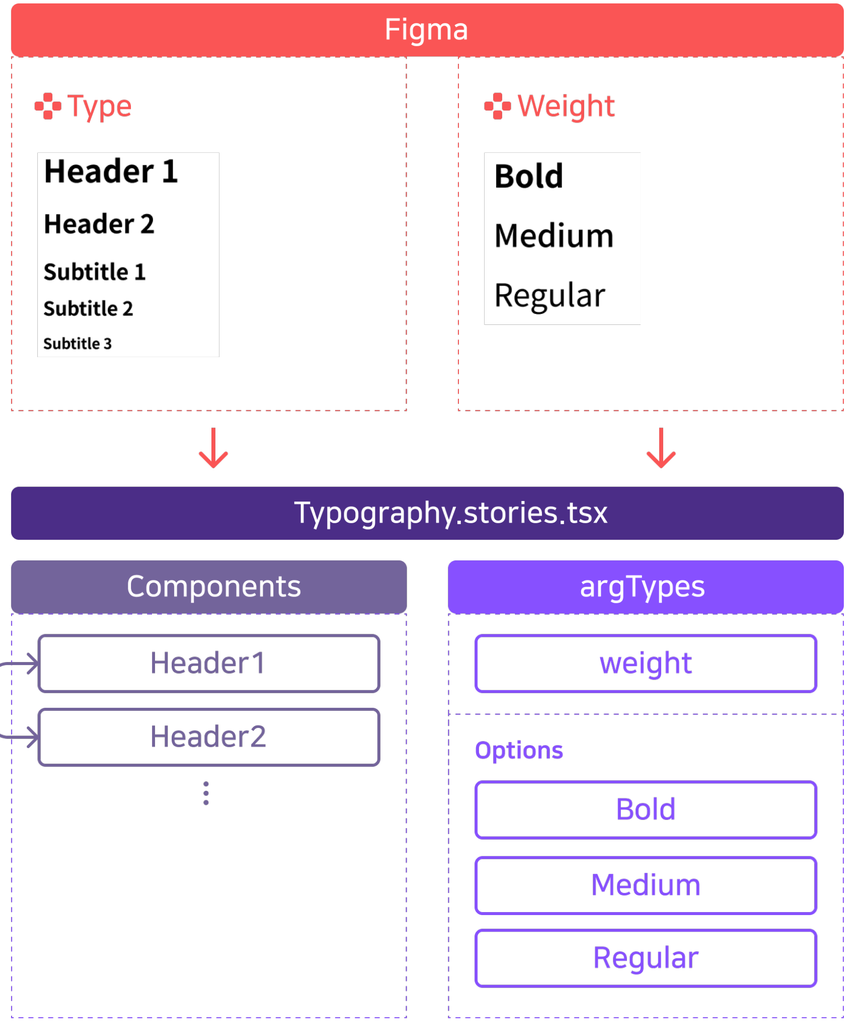
DesignCategory.interface.ts

- 스토리북/컴포넌트 렌더링에 필요한 타입/인터페이스 선언

4. 활용하기 : 스토리북 구축

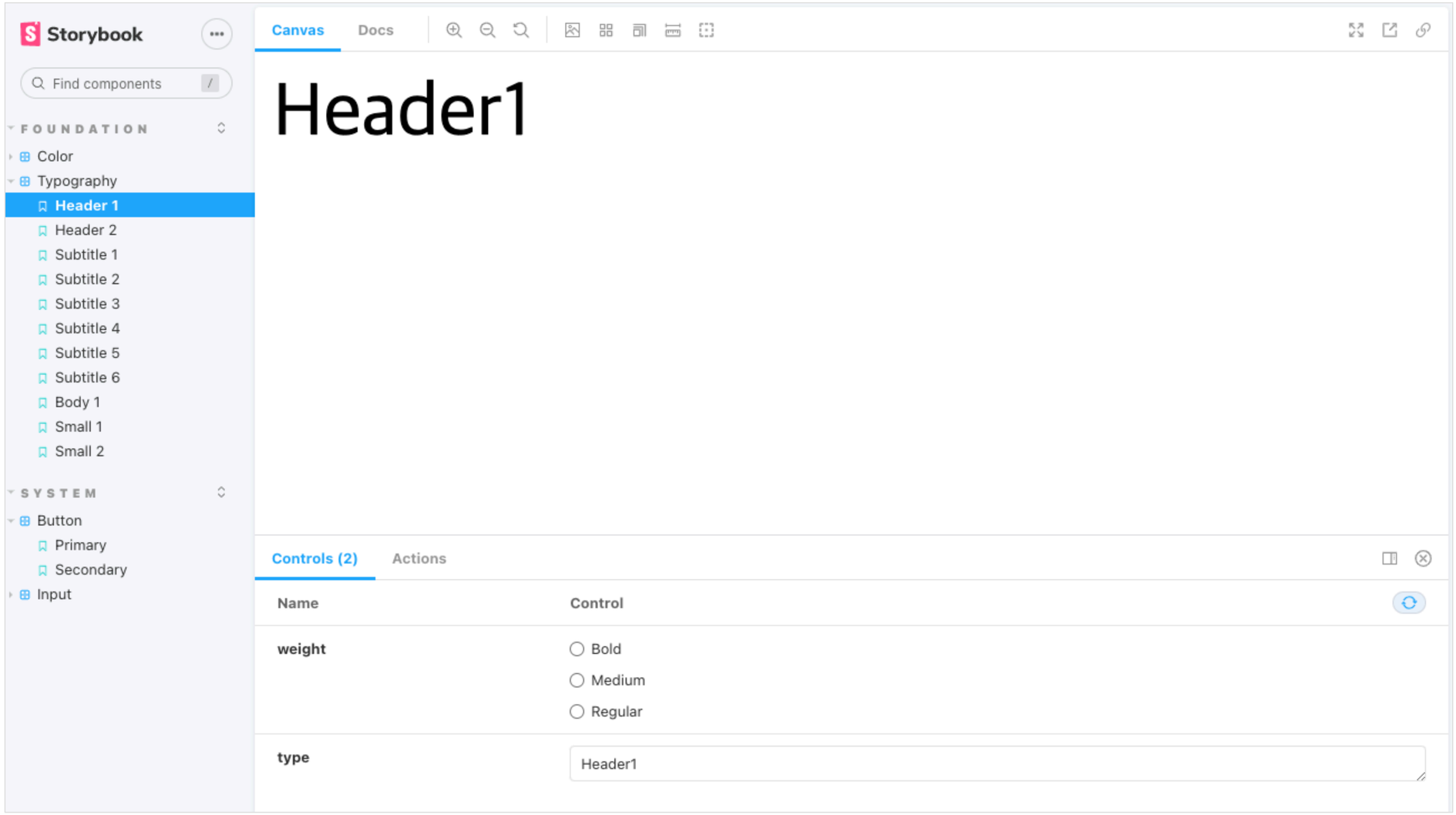
DEVIEW 2021

- assets
  - designToken.json
- stories
  - Typography.stories.tsx
  - Typography.tsx
  - Typography.interface.ts



4. 활용하기 : 스토리북 구축

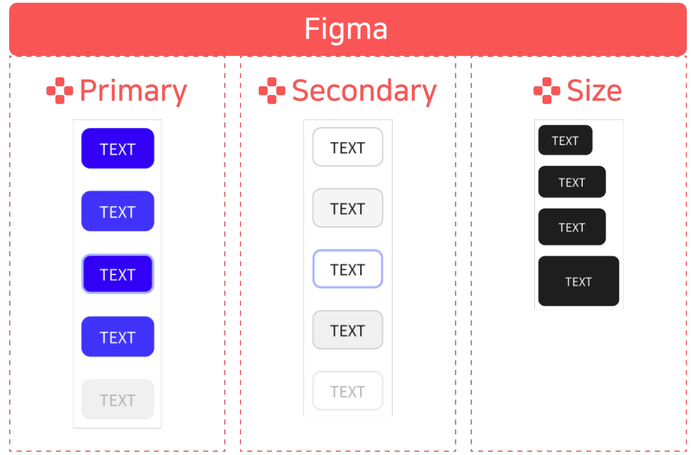
DEVIEW 2021

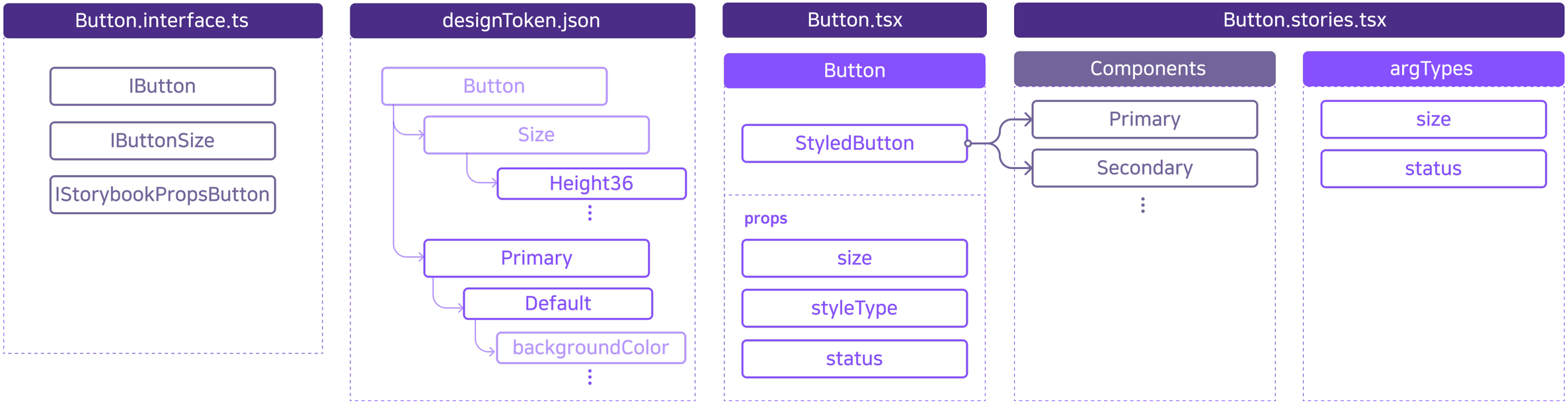


4. 활용하기 : 스토리북 구축

DEVIEW 2021

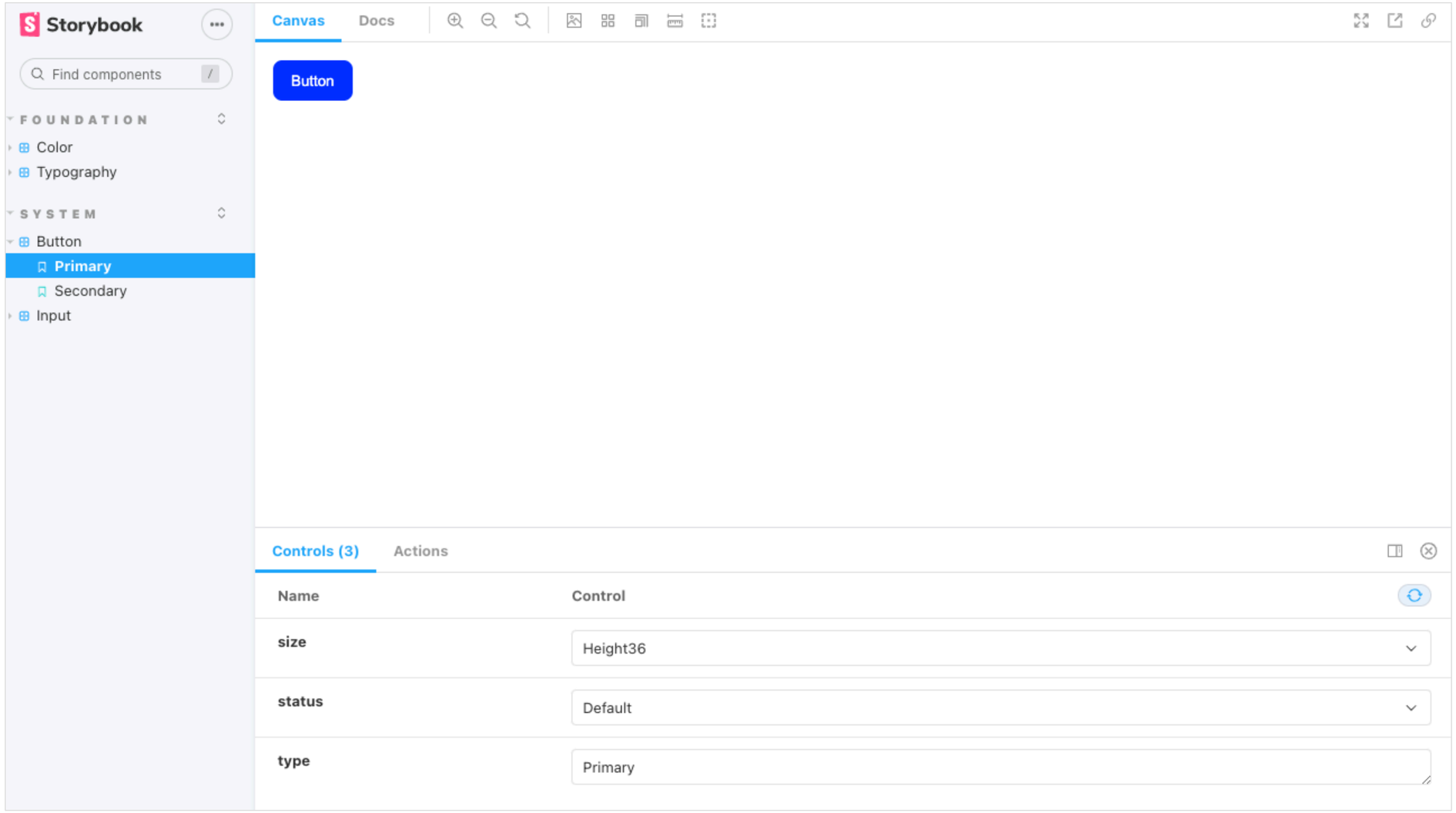
- assets
  - designToken.json
- stories
  - Button.stories.tsx
  - Button.tsx
  - Button.interface.ts





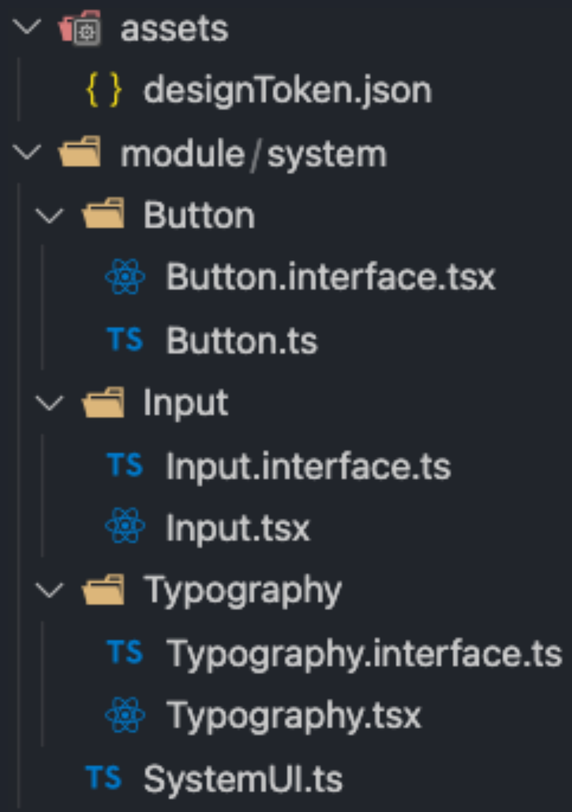
4. 활용하기 : 스토리북 구축

DEVIEW 2021



5. 활용하기 : 공통 컴포넌트 구축

DEVIEW 2021



기준: React / TypeScript

파일 구조

DesignCategoryFolder

DesignCategory.tsx

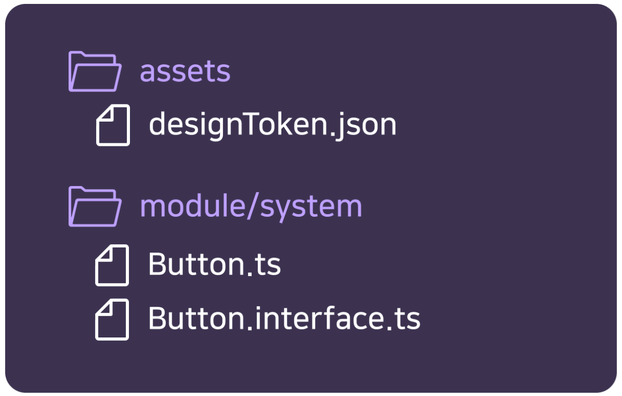
- 프로덕트에 사용되는 공통 컴포넌트를 선언

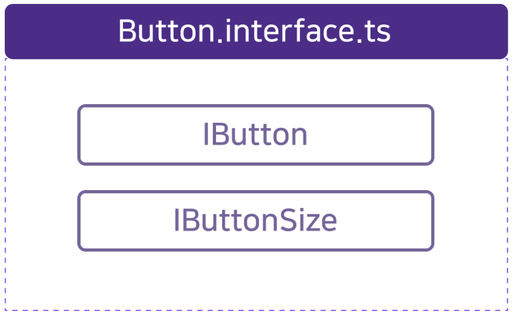
DesignCategory.interface.ts

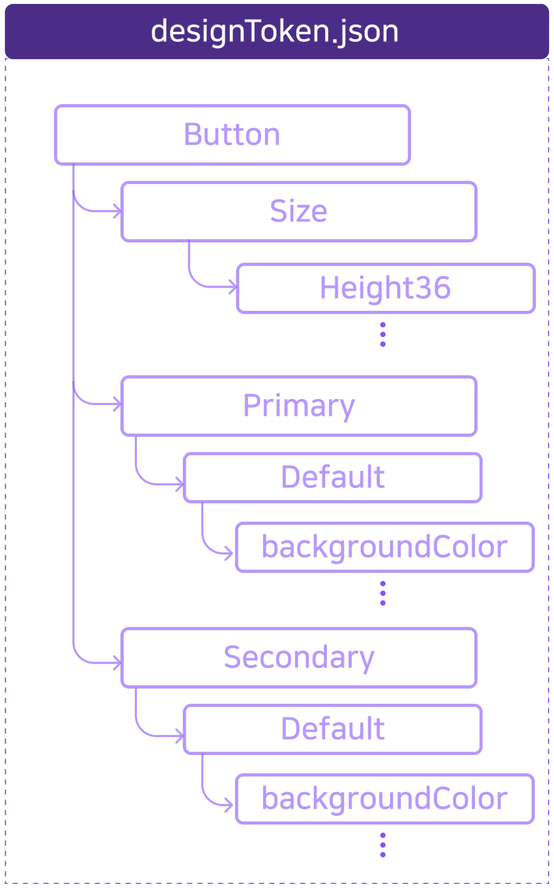
- 컴포넌트 렌더링에 필요한 타입/인터페이스 선언

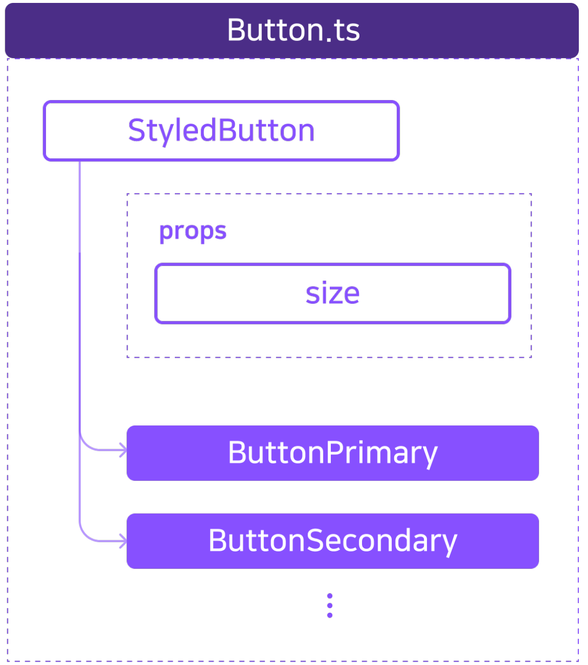
5. 활용하기 : 공통 컴포넌트 구축

DEVIEW 2021









5. 활용하기 : 공통 컴포넌트 구축

DEVIEW 2021

SystemUi.ts

export

- TypoHeader1
- TypoHeader2
- ⋮

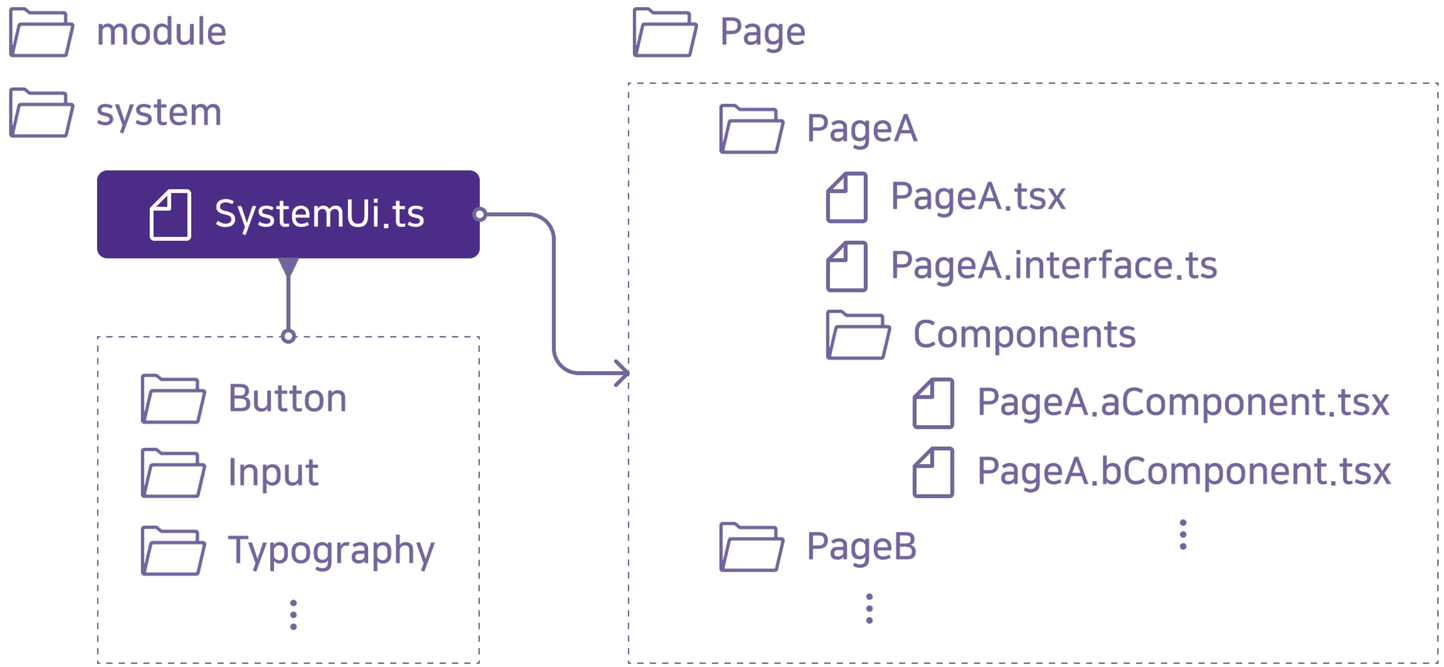
from Typography

export

- ButtonPrimary
- ButtonSecondary
- ⋮

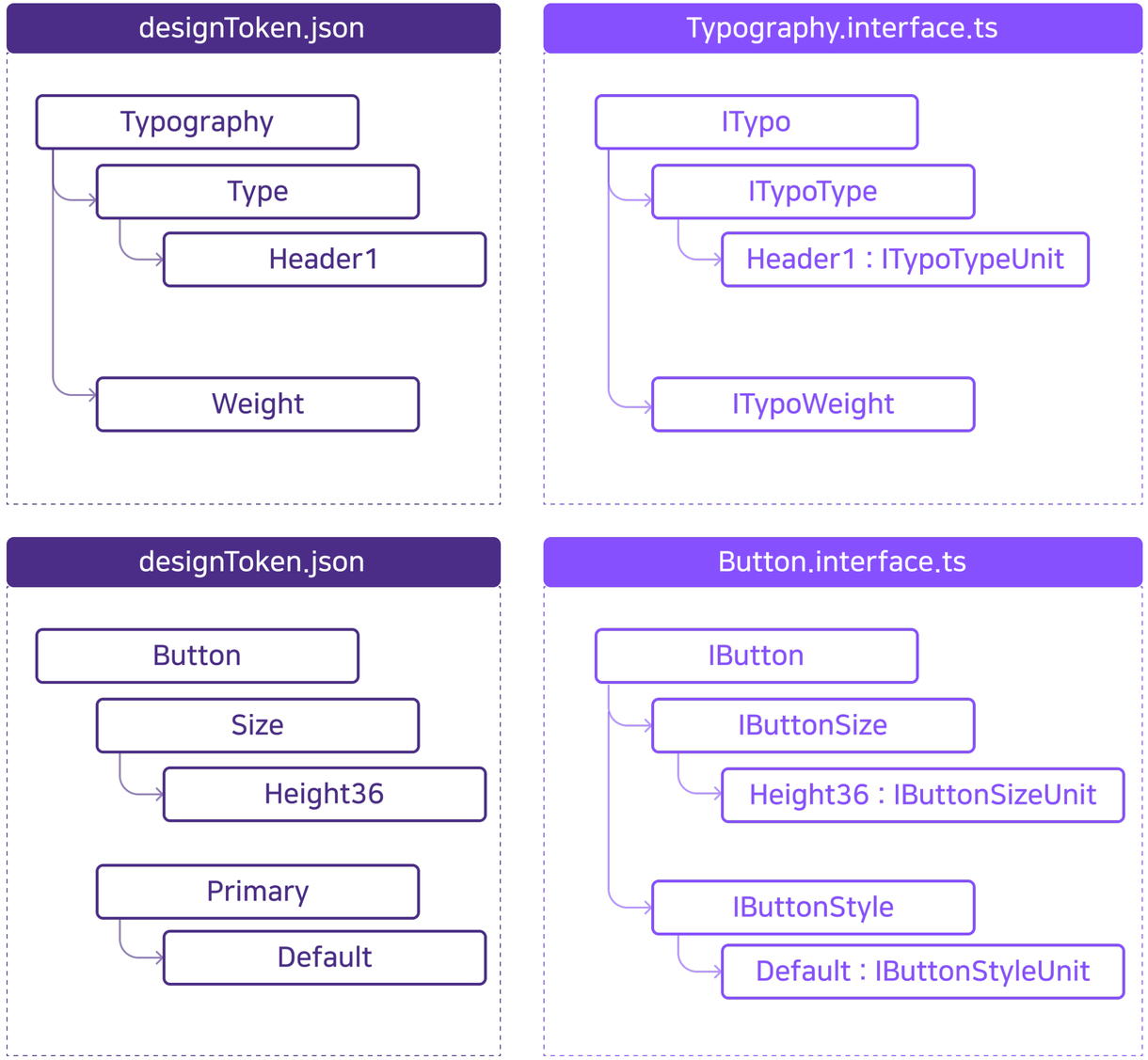
from Button

사용 구조



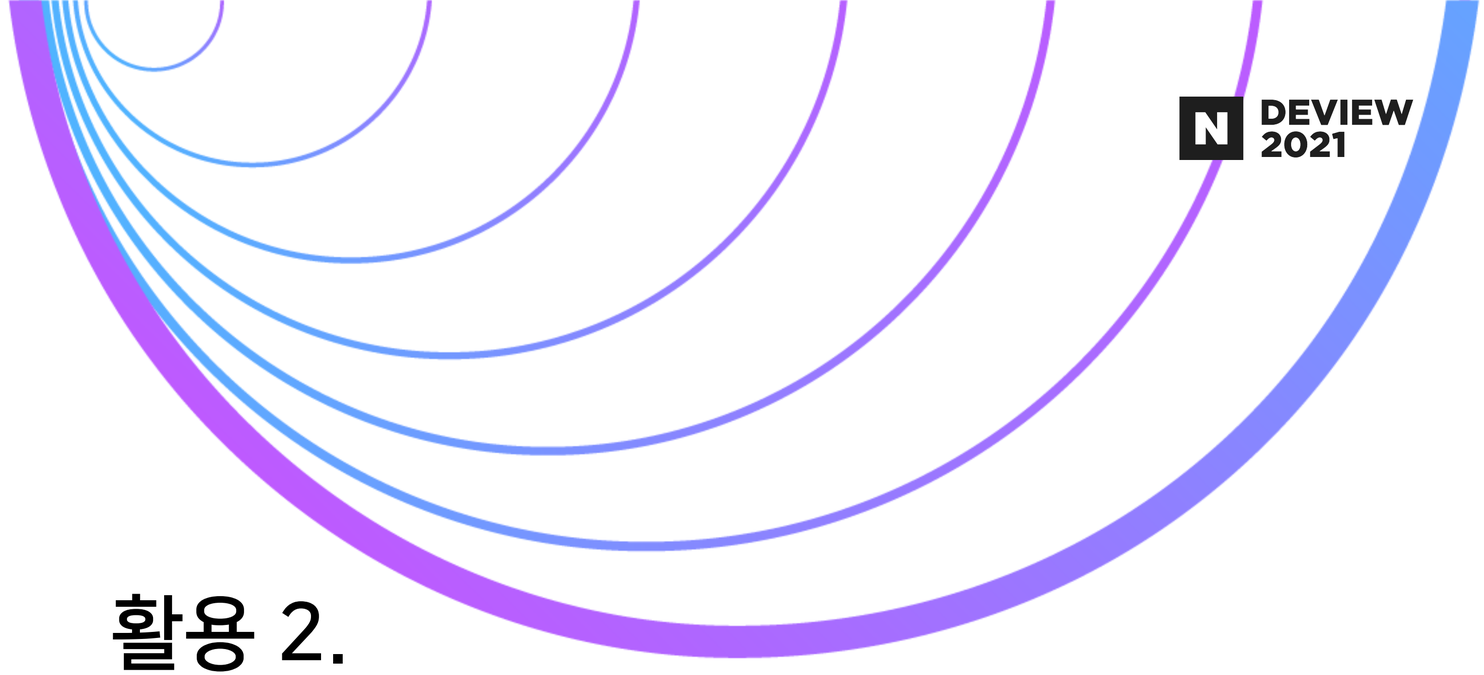
디자인 토큰의 타입 선언

타입스크립트로 디자인 토큰을 활용해 시스템을 구축할 때, 토큰의 구조에 맞춰 타입을 선언해야 함



효율적인 플로우 구축을 위해 설계 단계에서 고려하면 좋을 것들

- 디자인 카테고리, 디자인 요소별 필요한 항목 명확하게 나누기
- 토큰파일을 위한 피그마 파일을 만들 때 설명을 위한 요소는 가능하면 포함 X
- 피그마 페이지, 컴포넌트, 옵션에 대한 이름을 간결하고 명확하게 작성하기
- 디자이너와 함께 디자인 측면의 구조와 프론트엔드 측면의 구조 논의하기

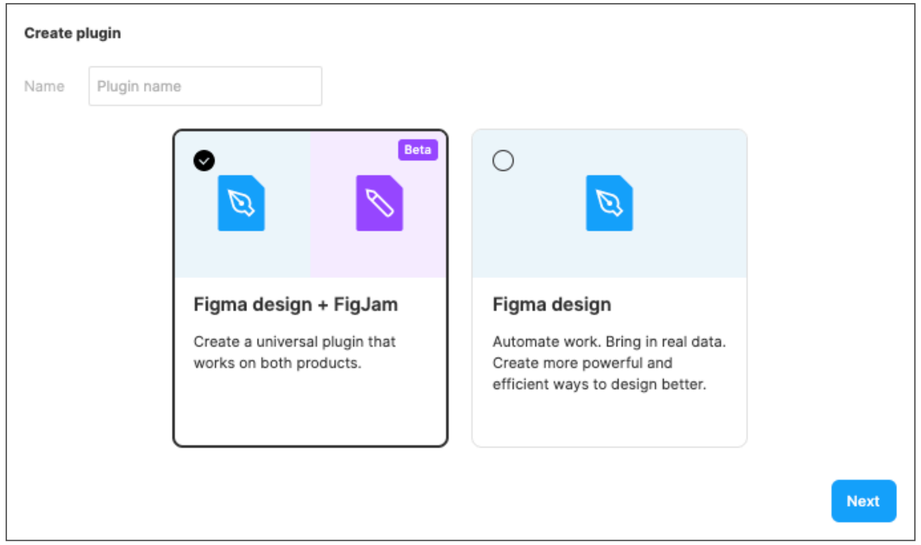


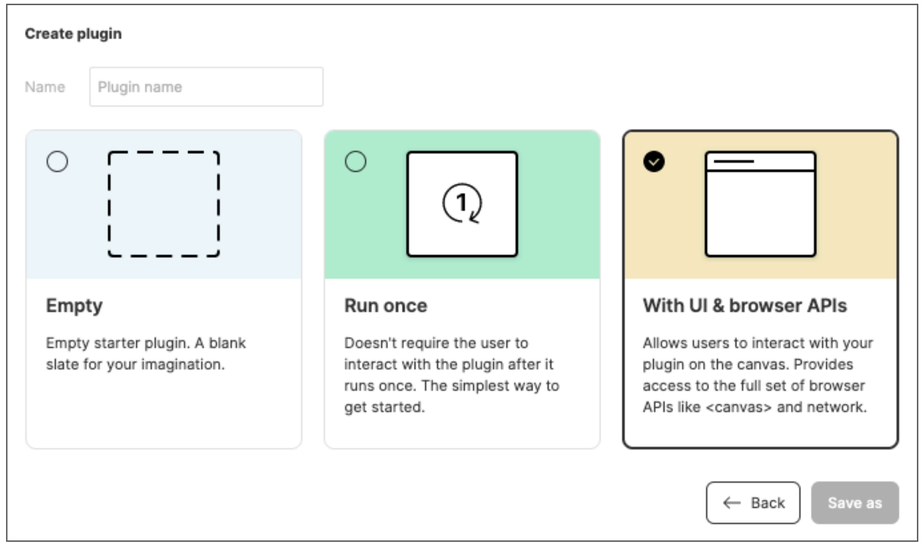
활용 2.

피그마 플러그인 활용하기

피그마 플러그인이란?

유저가 커스텀으로 기능을 확장해서 사용할 수 있는 써드 파티 기능.

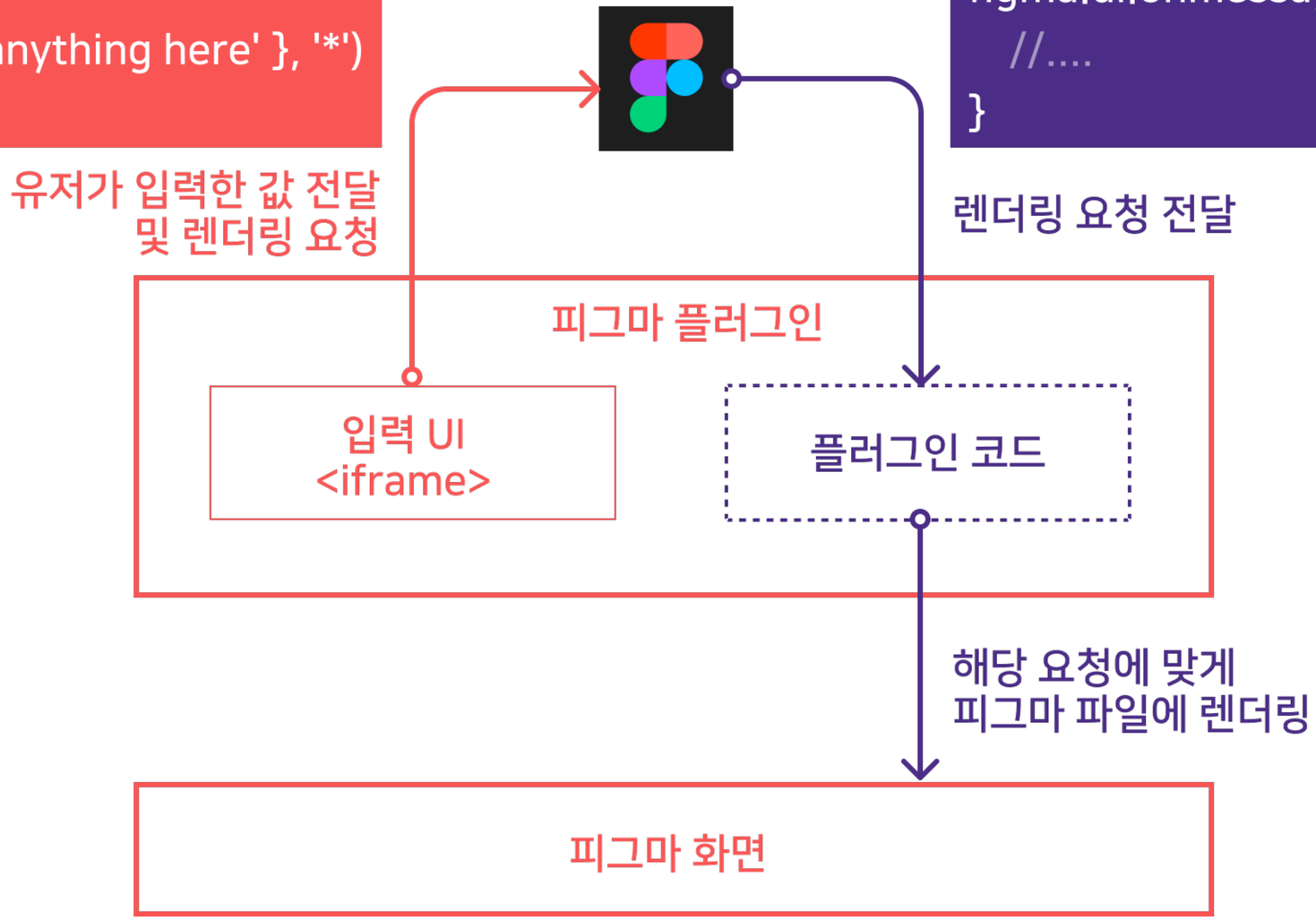




플러그인 구조


```
postMessage
({ pluginMessage: 'anything here' }, '*')
```

```
figma.ui.onmessage = (message) => {
  //....
}
```



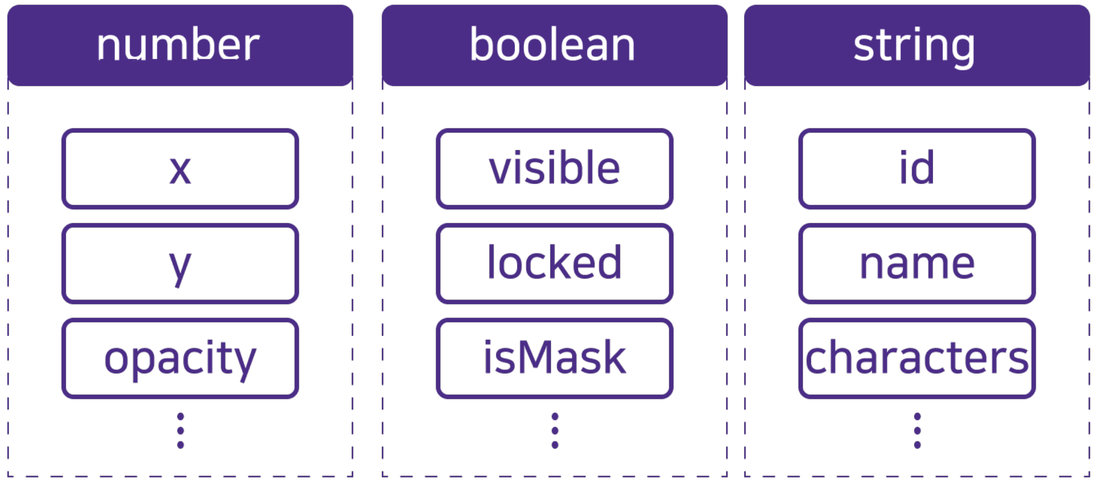
플러그인 구조

타입스크립트를 활용한 이유

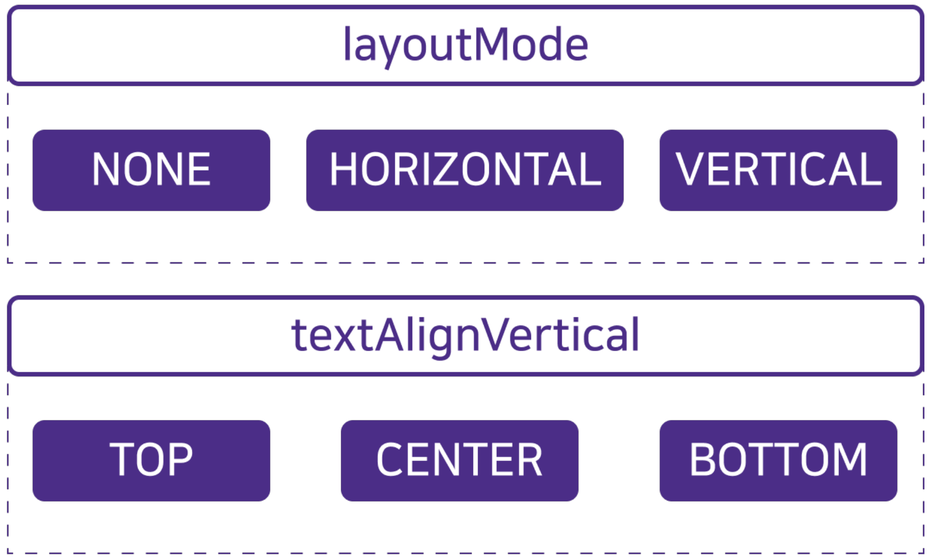
- 피그마에서 디자인 요소를 다룰 때 유저가 선택할 수 있는 옵션이 정해져 있음.
- 요소를 생성하고 다루기 위해서는 각 옵션마다 지정된 타입을 사용해야함.

피그마 항목 | 항목값의 타입

일반적인 타입의 항목들



특정 값으로만 설정해야 하는 항목들

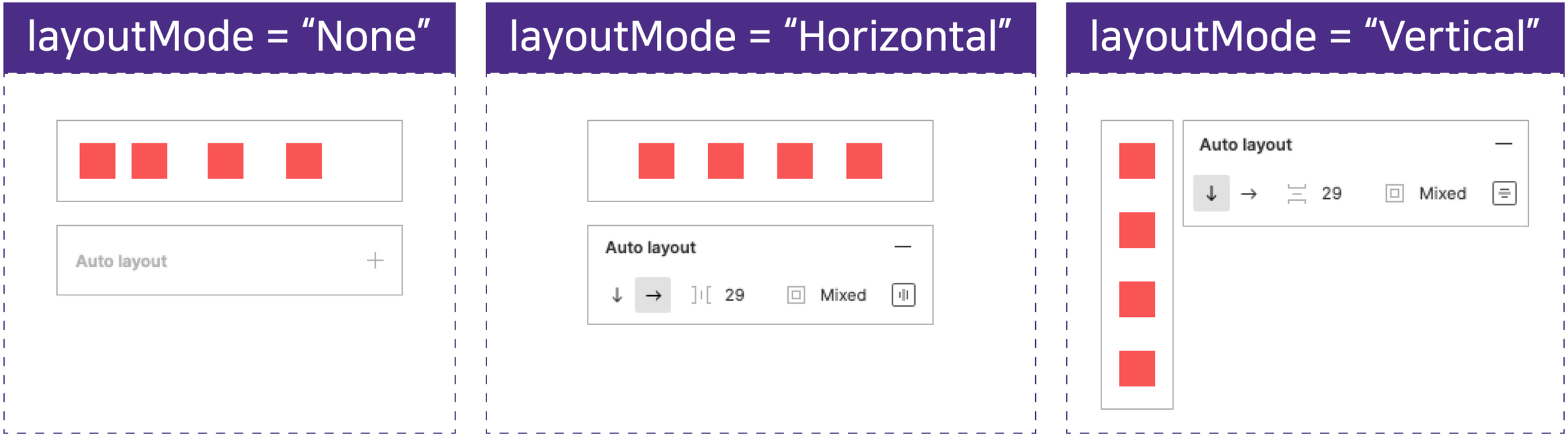


플러그인 구조

DEVIEW 2021

예시

layoutMode : “None” | “Horizontal” | “Vertical”



플러그인을 어떻게 활용할 수 있을까?

DEVIEW 2021

주로 활용한 포인트

- [x] 사람이 확인하고 입력하는 작업량 줄이기
- [x] 반복 작업 줄이기

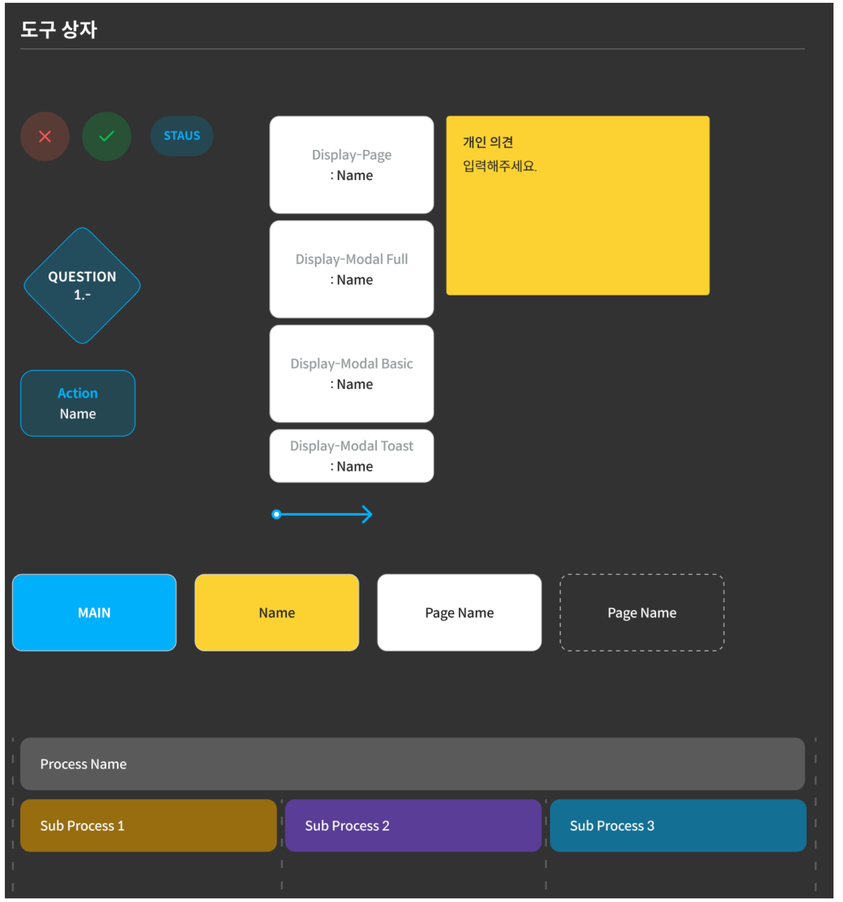
[사례] 회사 내부의 템플릿 구축하기

DEVIEW 2021

Before

- 많은 사람들이 피그마로 플로우 차트 제작
- 피그마에 익숙해져야 하고 시간이 오래 걸린다
- 각자의 방식으로 그리면서 통일성이 떨어짐

플로우 차트 도구 상자



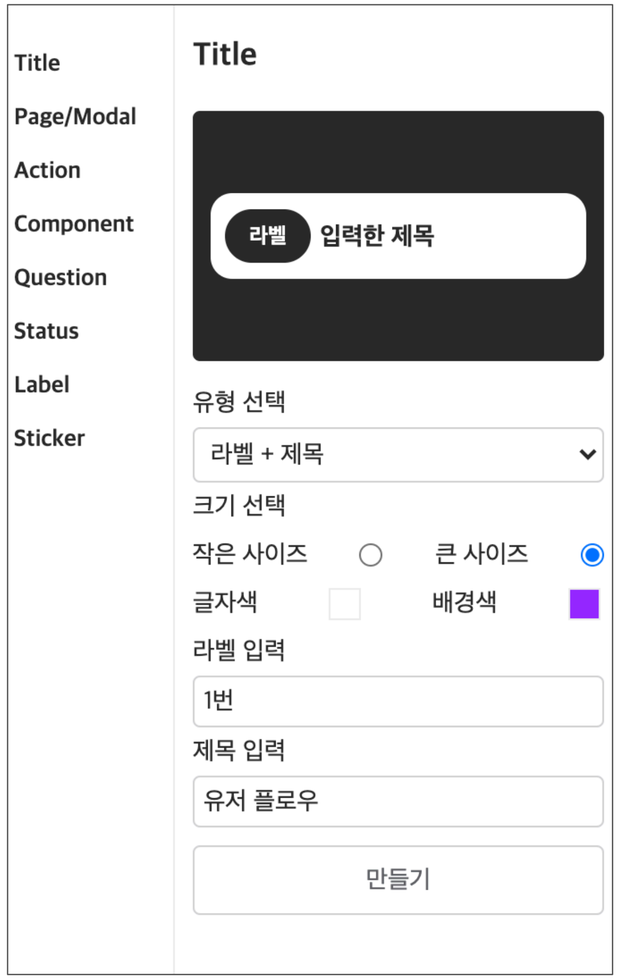
[사례] 회사 내부의 템플릿 구축하기

DEVIEW 2021

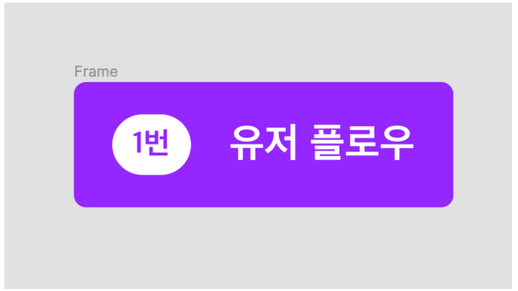
After

- [x] 플러그인에서 유형 선택 후 값을 입력
- [x] [만들기] 버튼을 클릭하면 해당 요소가 화면에 그려짐

플러그인



피그마 화면



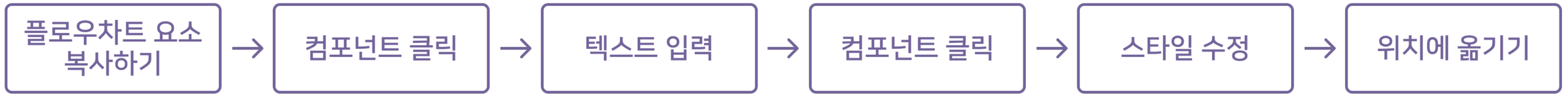
[사례] 회사 내부의 템플릿 구축하기

DEVIEW 2021

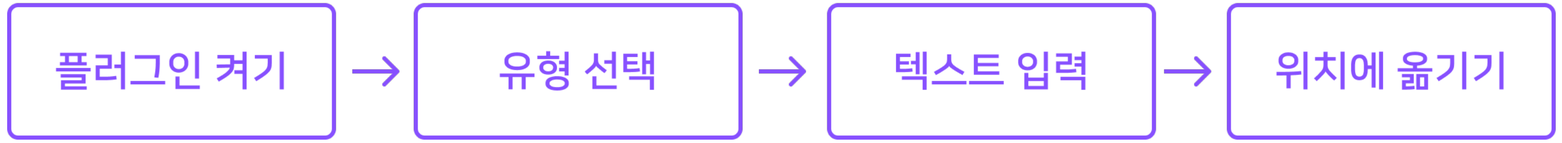
플러그인 코드 구조

어떤 점이 더 좋아졌을까?

Before



After



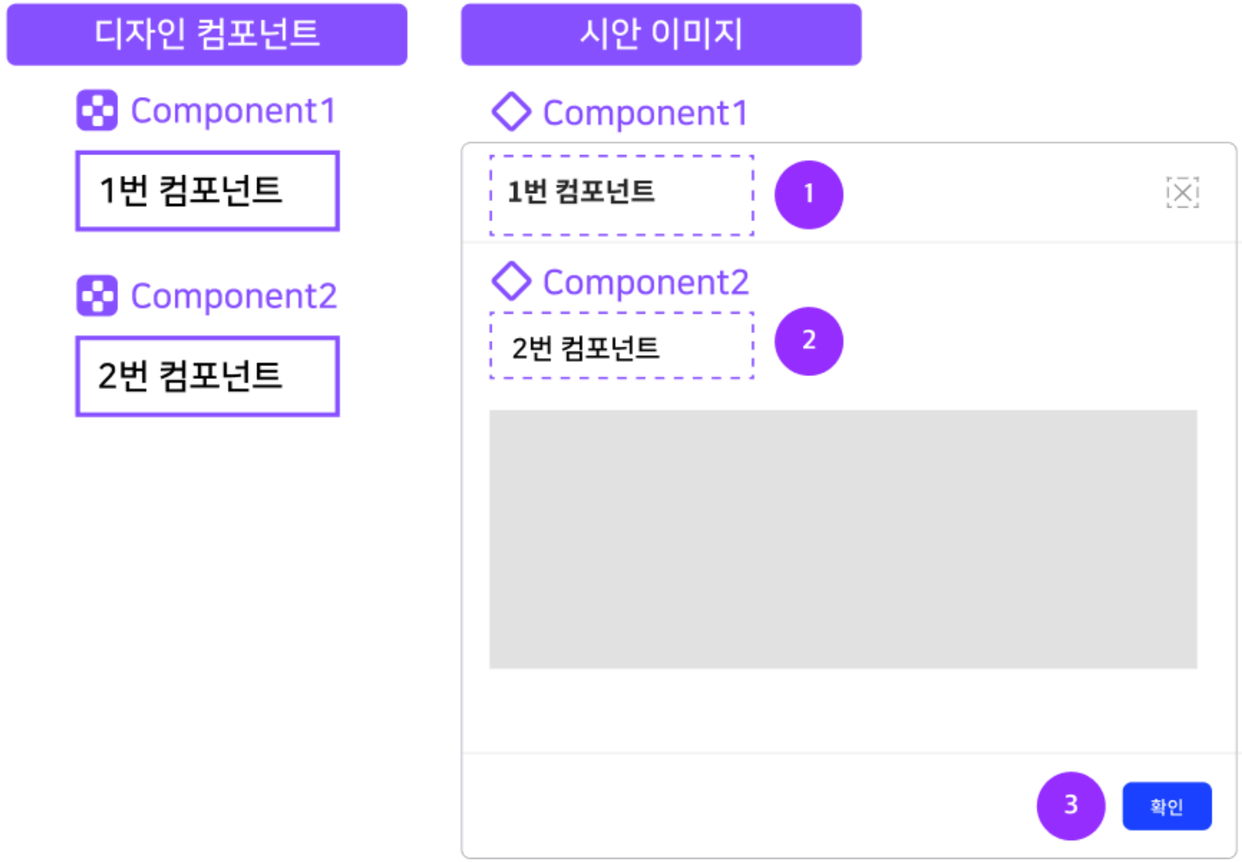
- [x] 피그마 문서 작성 효율이 높아짐
- [x] 회사의 공통된 템플릿 사용으로 구성원이 문서를 쉽게 이해/작성할 수 있음

[사례] 시안 텍스트 업데이트하기

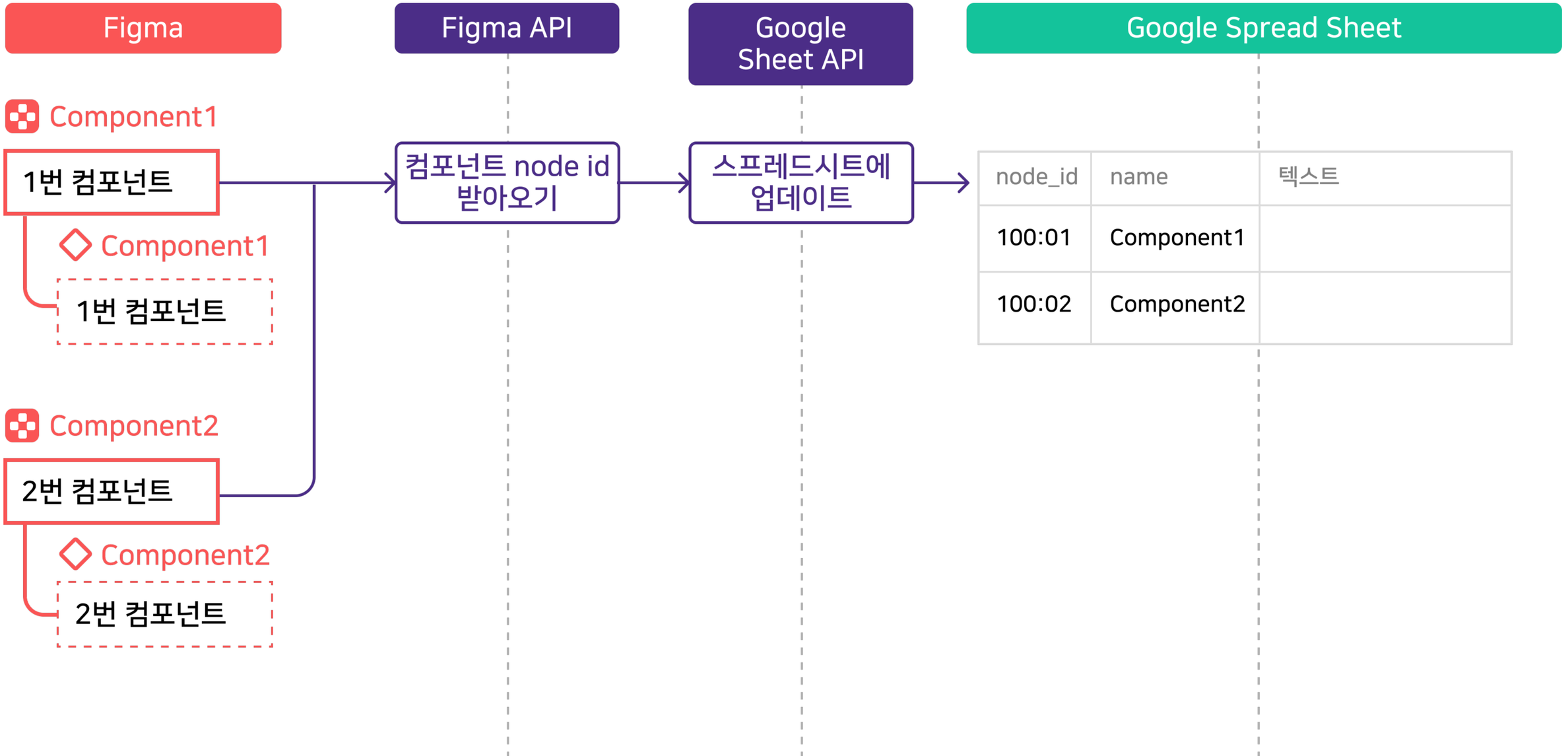
DEVIEW 2021

Before

- 시안 위에 번호를 붙여 텍스트 관리
- 텍스트는 구글스프레드 시트에서 관리
- 확인 후 수작업으로 업데이트 필요



[사례] 시안 텍스트 업데이트하기



[사례] 시안 텍스트 업데이트하기

DEVIEW 2021

Figma Plugin | Google Sheet API | Figma API | Figma

스프레드 시트 선택하기

스프레드 시트명

업데이트

스프레드시트 데이터 받아오기

node_id 일치하는 요소에 업데이트

node_id	name	텍스트
100:01	Component1	업데이트_1번 컴포넌트
100:02	Component2	업데이트_2번 컴포넌트

Component1

업데이트_1번 컴포넌트

Component1

업데이트_1번 컴포넌트

Component2

업데이트_2번 컴포넌트

Component2

업데이트_2번 컴포넌트

어떤 점이 더 좋아졌을까?

DEVIEW 2021

Before

UX writing용 시안 확인 → 시안에 번호 붙이기 → 스프레드시트에 번호, 분류 작성 → 텍스트 작성 → 확인 요청 →

확인 → 번호에 맞춰 텍스트 수정 → 확인 요청 →

검수

After

UX writing용 시안 확인 → 스프레드시트 업데이트 → 텍스트 입력 → 확인 요청 →

플러그인 클릭 →

검수

어떤 점이 더 좋아졌을까?

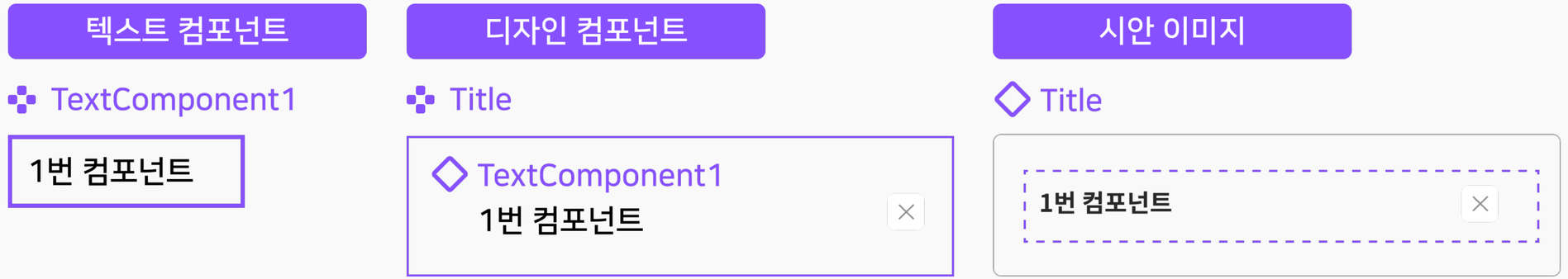
DEVIEW 2021

플로우 개선 & 플러그인 제작으로 개선된 점

- [x] 시안의 텍스트 업데이트 시간 단축
- [x] 작업의 효율성을 높임

주의사항

- [x] 시안을 작업할 때 미리 업데이트 과정을 고려해서 진행해야 한다

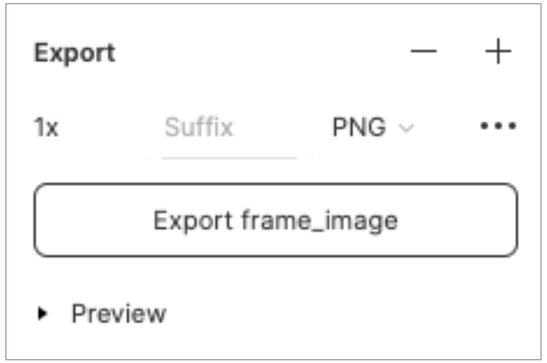


[사례] 이미지 한 번에 다운로드

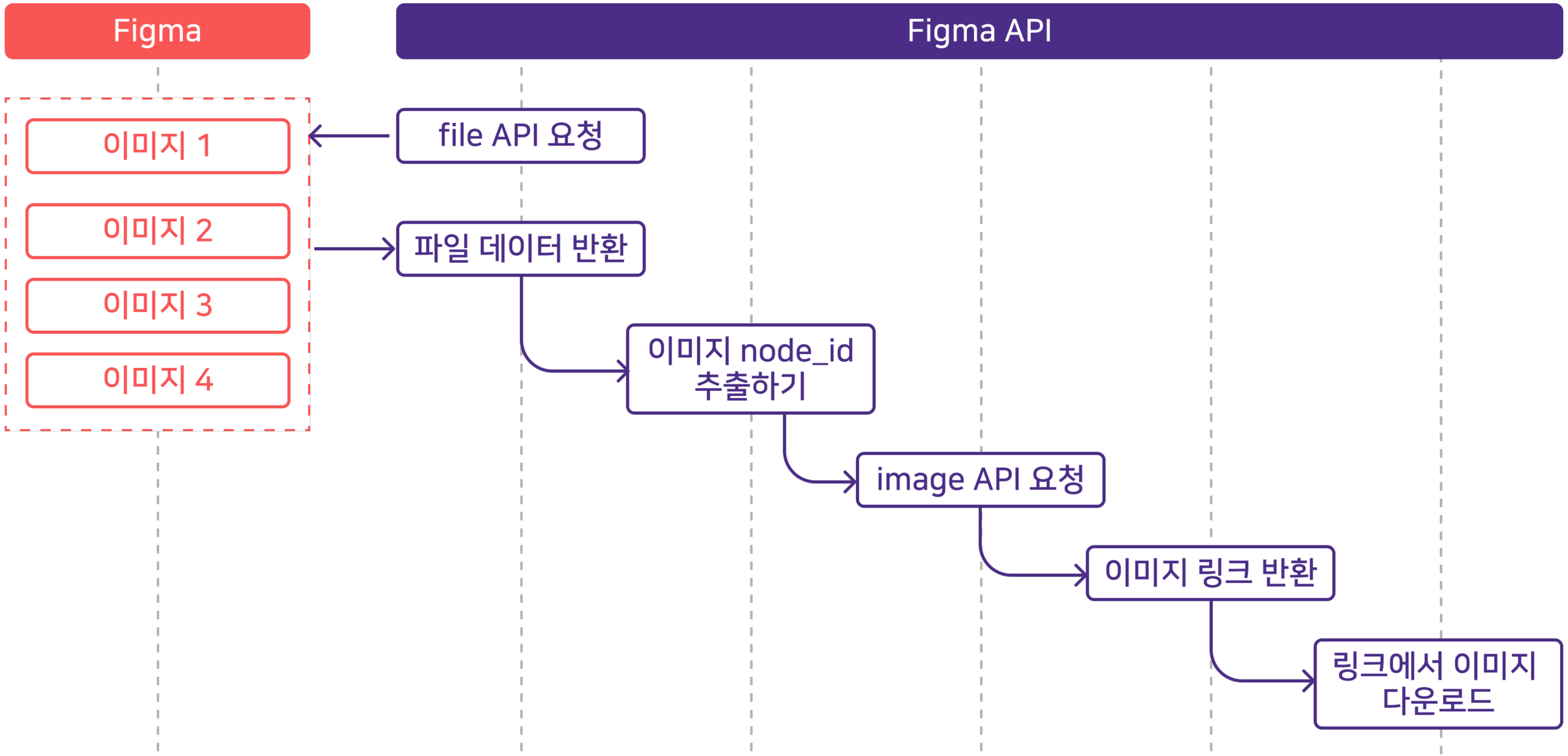
DEVIEW 2021

Before

- 피그마에서 제작한 요소를 이미지 파일로 다운로드 받아야 할 때가 있음
- 이미지를 다운로드 받으려면 각 이미지를 선택 후 Export해야 함
- 많은 이미지를 다운로드 받을 경우 일일이 선택해야 하는 번거로움



[사례] 이미지 한 번에 다운로드



마무리

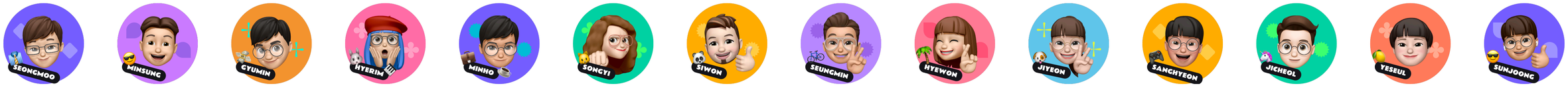
- 피그마 API를 활용하면 사람들과 좀 더 효율적으로 일할 수 있지 않을까?에 대한 고민으로 시작한 프로젝트
- 피그마는 계속 발전하는 디자인 툴이라 앞으로 API의 기능도 바뀔 수 있음
- 어떤 기술을 활용할 지는 바뀔 수 있지만 이 기술을 어떻게 활용해야 같이 일하는 사람들과 좀 더 효율적으로 일할 수 있을지에 대한 고민은 계속 필요하다.

마무리

DEVIEW 2021

데이터라이즈는 **이커머스를 위한 올인원 그로스 솔루션**을 만듭니다.
하루가 다르게 성장하는 로켓 속에서 **즐겁게, 스마트하게 같이 성장할 동료를** 기다립니다.

https://team.datarize.ai/



Thank You