

Pure UI

maru@spoqa.com

2017년 8월 크리에이터 워크샵

스페셜 땡스 투

Pure UI

July 13, 2015 – 470,860 views

<https://rauchg.com/2015/pure-ui>



Adam Solove

Follow

Building web UIs. Everything should be faster.

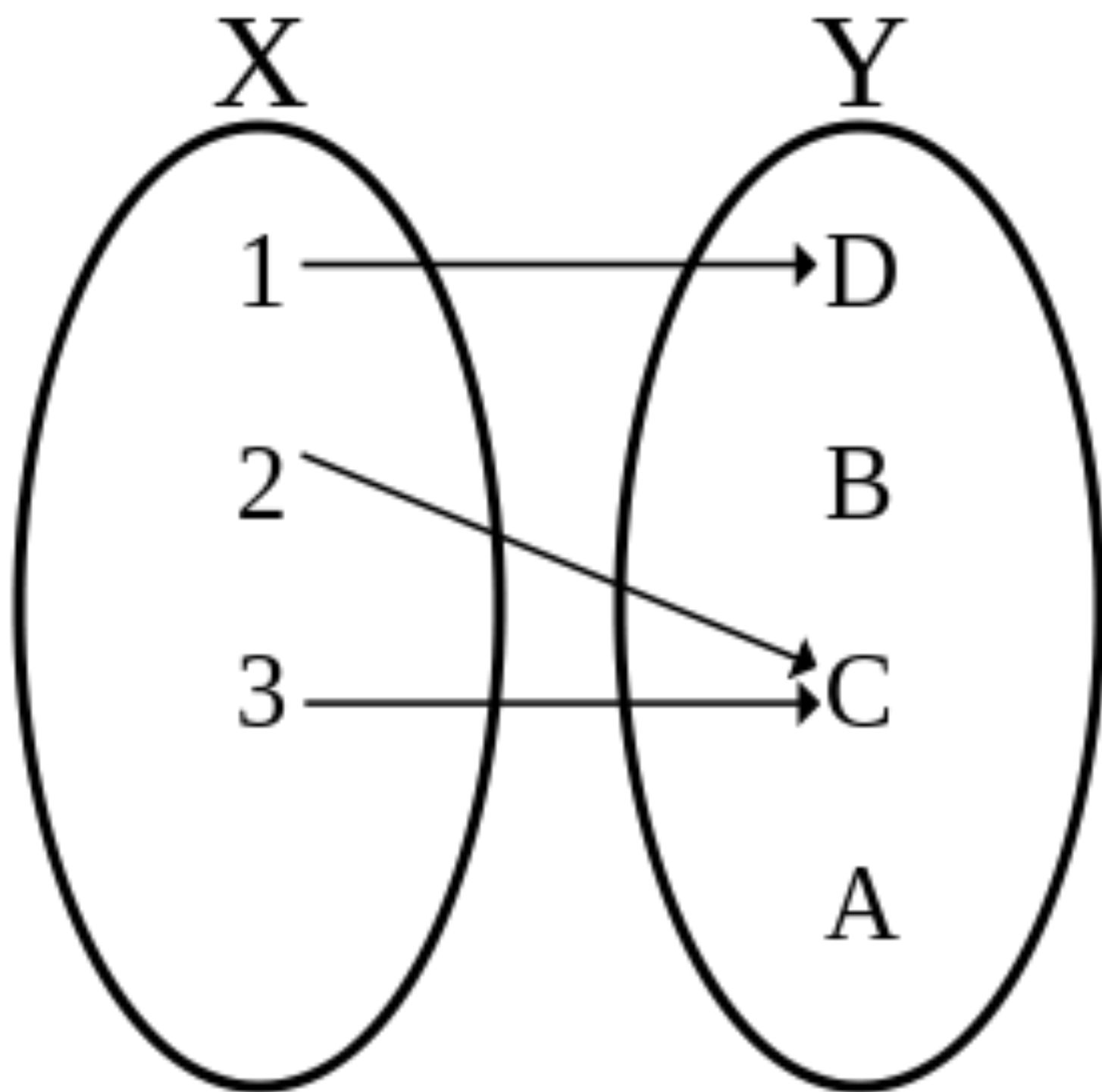
Jul 14 · 12 min read

Pure UI Control

<https://medium.com/@asolove/pure-ui-control-ac8d1be97a8d>

- 이 발표의 내용은 아래 두 에세이에서 소개된 아이디어의 요약이 거의 전부입니다.
- Guillermo Rauch - <Pure UI>
- Adam Solove - <Pure UI Control>
- 구체적인 액션 플랜을 제시하기보다는 아이디어를 소개하고 제안하는 정도로 기획했습니다.

함수



- 프로그래밍의 함수(=subroutine)가 아닌, **수학의 함수**
- 정의역 위의 독립 변수를 공역 위의 값으로 매칭해줍니다.
- 정의역 X - 공역 Y 사이의 함수 f 에 대해,
 x 가 같다면 $f(x)$ 의 값은 늘 동일합니다.

두 가지 시각

- UI는 화면에 그려진 컴포넌트들이다.
- 사용자의 상호작용이나 서버 응답 등의 이벤트가 발생했을 때, 프로그래머는 코드로 화면에 그려진 컴포넌트에 들어간 값이나 생긴 모습을 적절히 조작한다.
- UI는 기저에 깔린 데이터로부터 나온 결과물이다.
- 사용자의 상호작용이나 서버 응답 등의 이벤트가 발생했을 때, 해당 이벤트는 UI를 직접 변경하는 대신 기저에 깔린 데이터를 변경한다.
- 데이터가 변경되면 (함수의 인자가 바뀌었으므로) UI는 그에 걸맞게 업데이트된다.

UI를 액션의 대상으로 바라보기



- 페이지가 있습니다.
- 메시지 수정하기 버튼을 누르면 메시지 수정을 위한 팝업이 뜹니다.
- 우측 상단 ON/OFF 토글을 누르면 활성화 상태를 변경하고, UI의 ON/OFF 상태를 토글합니다.
- 리포트 영역에는 맨 처음에 스피너가 들어있습니다.
 - 리포트를 다 받아오고 나면 실제 리포트로 갈아치웁니다.
- 수신 고객 버튼을 누르면 수신 고객 팝업이 뜹니다.

UI를 액션의 대상으로 바라보기

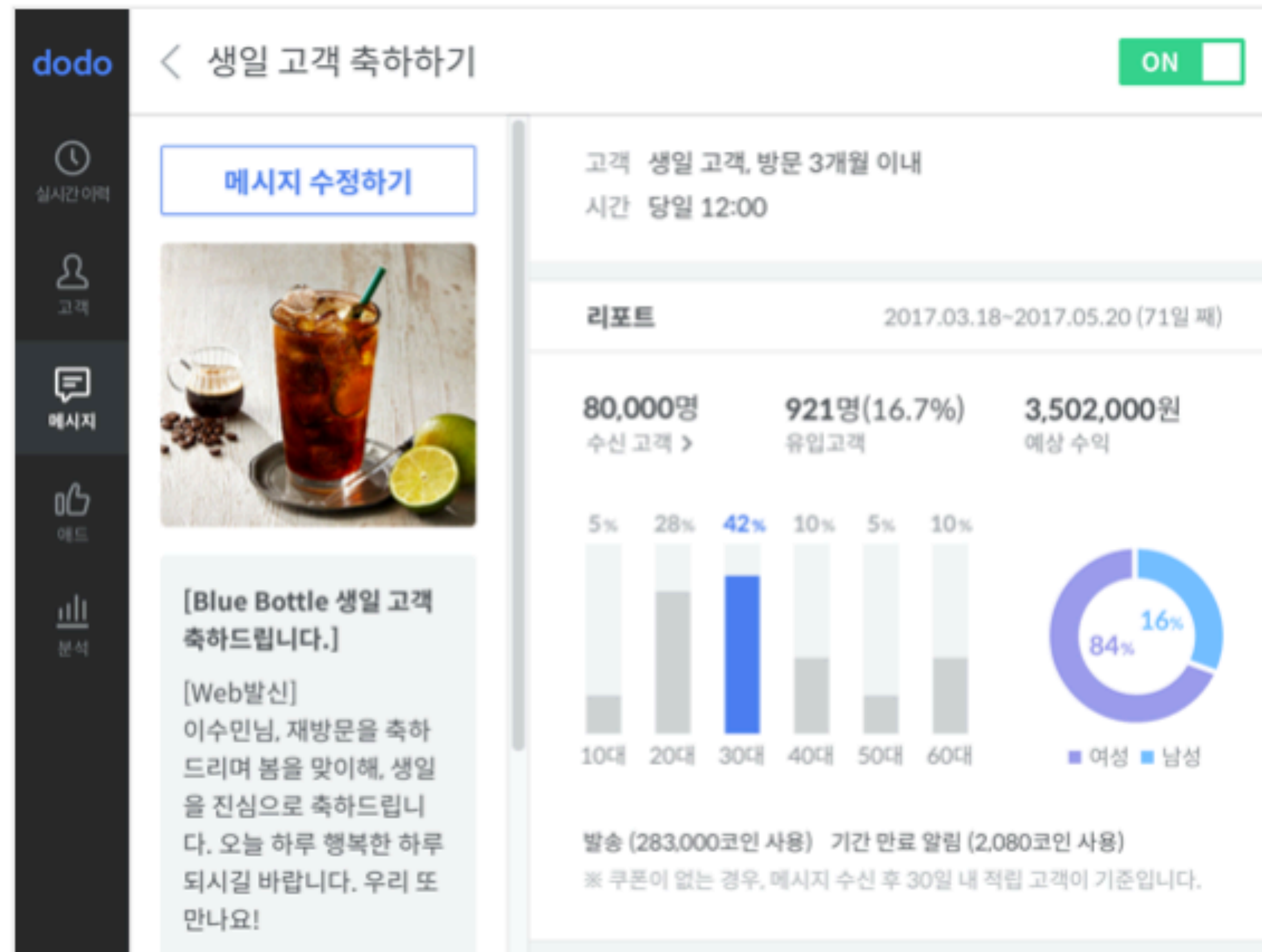
```
1  var $stateSelect = $('#state-select');
2  var $districtSelect = $('#district-select');
3
4  function updateDistrict(state) {
5      state = state || '';
6      $districtSelect.html('');
7      $.each(locationData[state], function (key) {
8          $districtSelect.append($option(key, '시/군/구'));
9      });
10     $districtSelect.val('');
11 }
12
13 $stateSelect.on('change', function () {
14     updateDistrict($stateSelect.val());
15     switchInput();
16 });
17
```

- 랜딩페이지 인바운드 폼에 사용되는 코드 일부
- 앞쪽 드롭다운(stateSelect)의 선택된 값이 바뀌면 그에 맞게 뒤쪽 시/군/구 드롭다운(districtSelect)의 가능한 옵션을 업데이트해주는 코드

지역

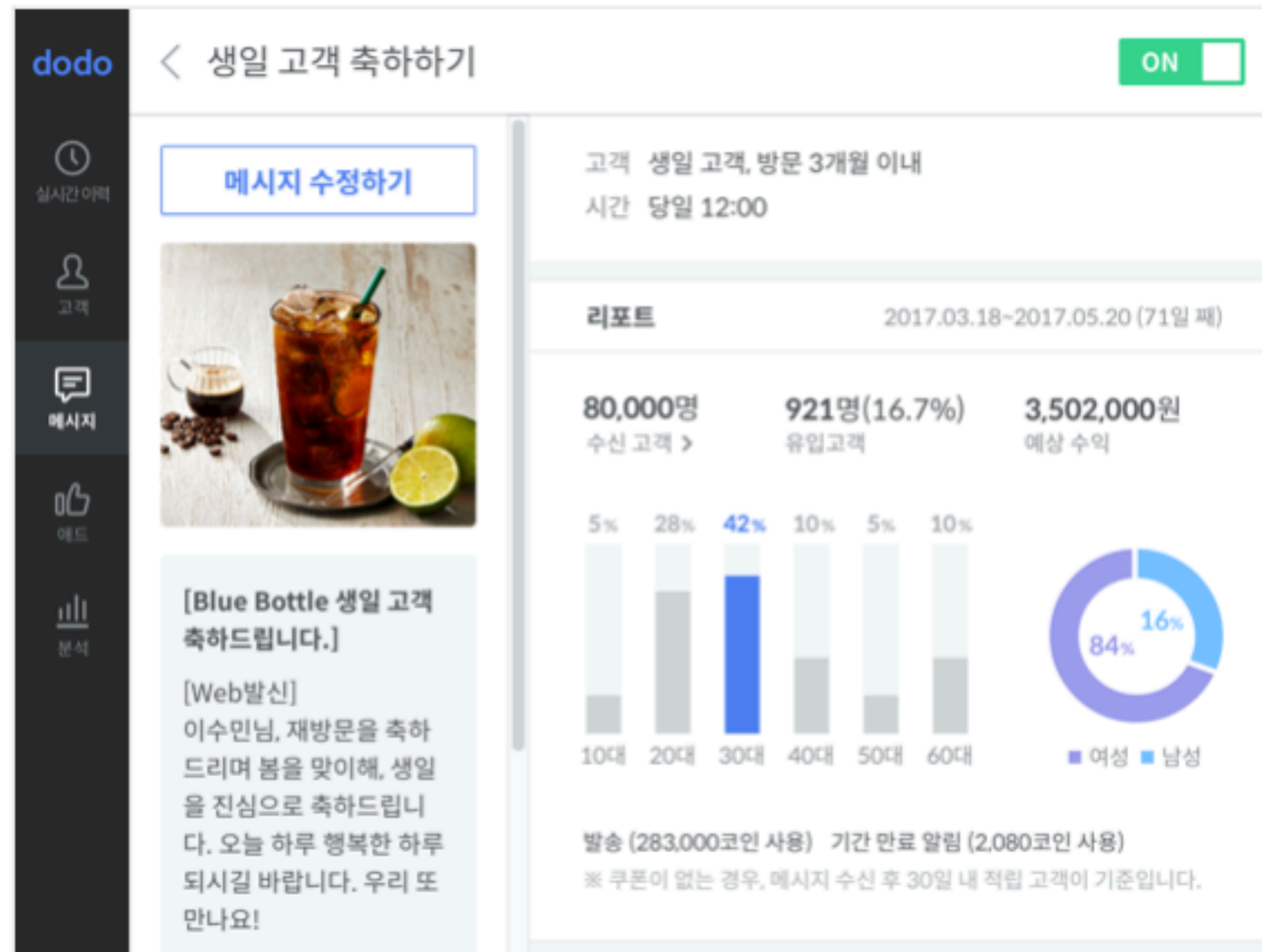
서울특별시	시/군/구
-------	-------

UI를 상태의 발현으로 바라보기



- 페이지가 있습니다.
이 페이지는 여러 상태를 들고 있습니다.
- 메시지 수정하기 팝업이 떠있는지 아닌지, 수신 고객 팝업이 떠 있는지 아닌지 여부를 나타내는 상태가 있습니다.
- ON/OFF 를 나타내는 상태가 있습니다.
- 리포트를 받아왔는지의 상태와 받아왔다면 받아온 리포트를 담고 있는 상태가 있습니다.
- 수신 고객 버튼을 누르면 수신 고객 팝업이 뜹니다.

UI를 상태의 발현으로 바라보기



- 예를 들어 아래와 같은 형태로 나타낼 수 있습니다.
- {
 popup: 없음 OR 수신 고객 OR 메시지 수정

 isPromotionActive: ON OR OFF

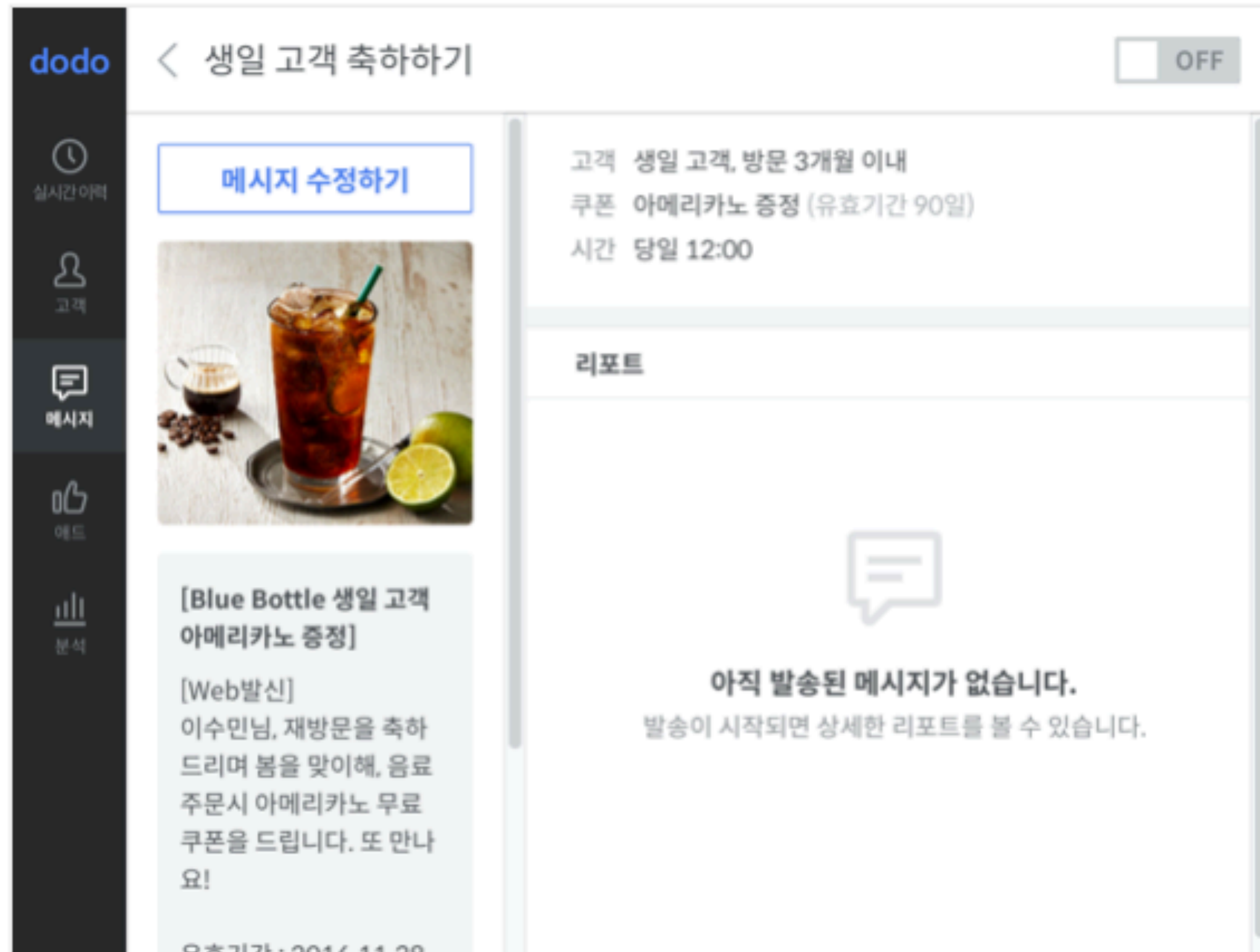
 report: 없음 OR 빈 리포트 OR 내용이 있는 리포트
}
- 이러한 상태의 조합은 페이지/컴포넌트와 1:1로 대응합니다.
- 지금까지 어떤 일이 있었는지는 알 바 아닙니다.

UI를 상태의 발현으로 바라보기



```
{  
  popup: 없음  
  
  isPromotionActive: ON  
  
  report: 내용이 있는 리포트  
}
```

UI를 상태의 발현으로 바라보기



```
{  
  popup: 없음  
  
  isPromotionActive: OFF  
  
  report: 빈 리포트  
}
```

UI를 상태의 발현으로 바라보기

쿠폰 수정 > 메시지 수정 > 발송 시각 수정

적립 40번 이상

유기농 주스 메뉴 중 레드 오렌지, 망고, 두리안, 딸기 중 택 1

쿠폰 받은 날로부터 30일간 유효

✓ 만료 3일 전 알림 보내기

쿠폰 종류

상품증정

유기농 주스 메뉴 중 레드 오

쿠폰 유효기간

30일

사용 가능 요일

☒ 평일 ☒ 토 ☒ 일

유효기간 만료 알림 *코인 추가사용

ON

유효기간 만료 알림이란?

취소

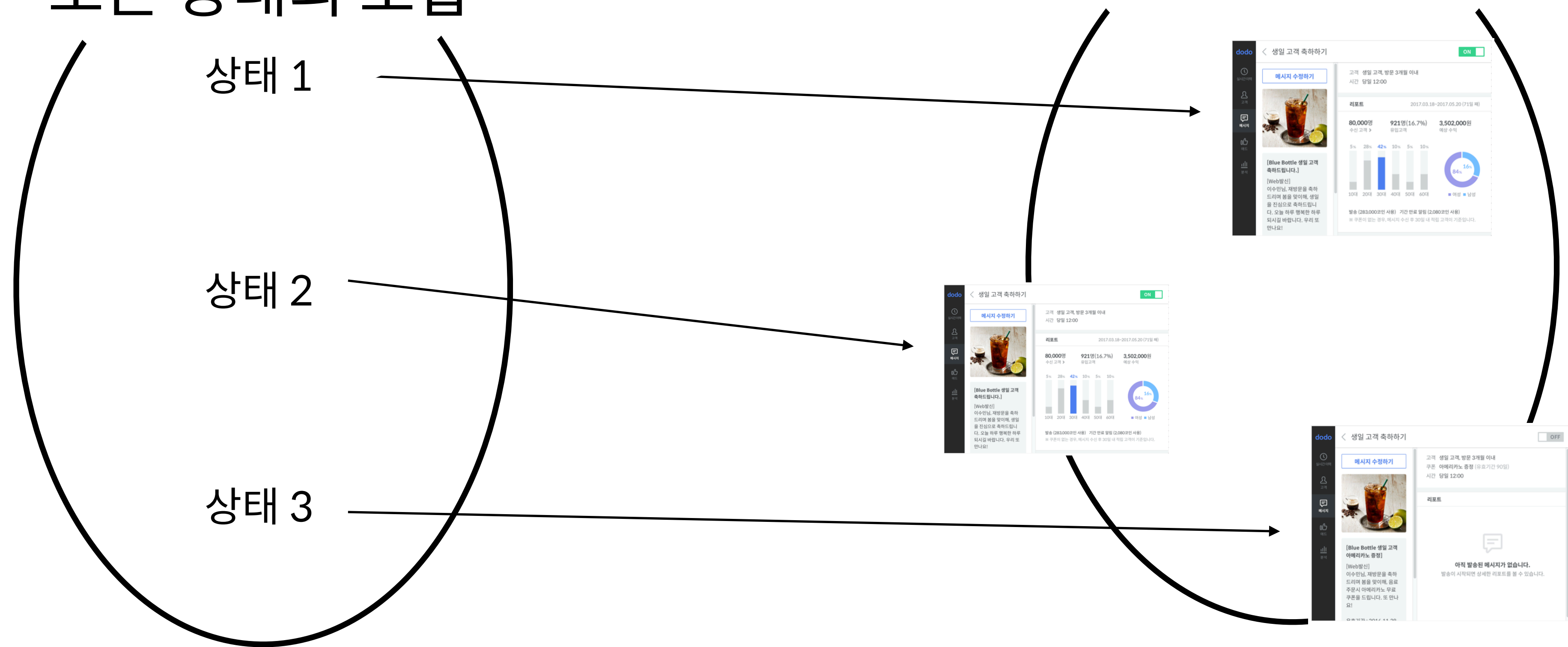
저장

```
{  
  popup: 수신 고객  
  
  isPromotionActive: 원들  
  
  report: 원들  
}
```

$$UI = f(\text{데이터})$$

페이지가 가질 수 있는
모든 상태의 조합

페이지가 가질 수 있는
모든 UI의 조합



같은 상태 x 에 대한 $f(x) = \text{UI의 값은 늘 동일합니다.}$

비교

- UI는 화면에 그려진 컴포넌트들이다.
 - 특정 시점의 화면을 보고 싶으면 일련의 행동을 모두 수행해야 함.
 - 어떤 화면은 화면이 처음 그려진 이후로 행해진 액션의 합이기 때문에, 디버깅이 까다롭다.
- UI는 기저에 깔린 데이터를 인자로 한 함수의 값이다.
 - 특정 시점의 화면을 보고 싶으면 인자(= 컴포넌트의 상태)를 조정해주면 됨.
 - 어떤 화면은 데이터가 특정 상태일 때 항상 같은 모습이기 때문에, 화면이 의도한 대로 나오고 있지 않다면 상태가 올바른지 보면 된다 🙌 디버깅이 쉬워진다.
 - 몇 년 전부터 프론트엔드 프로그래머 커뮤니티에서는 이 쪽이 훨씬 나은 접근법이라는 합의가 이뤄졌습니다.

준비물

- 상태를 받아 UI를 대신 그려주고, 상태가 바뀌면 UI를 업데이트 해 주는, 기존에는 사람이 하던 작업을 대신 해 줄 누군가
 - 우리 회사에서는 리액트님께서 이 역할을 맡아 주십니다 🙏
- 각 컴포넌트, 페이지 등이 가질 수 있는 상태의 **명시적인 목록**
 - 오늘 발표에서 (굉장히 대책없이) 제안하고 싶은 건 이것입니다.

상태의 명시적인 목록이 있으면...

- 구현을 하다가 ‘아차!’ 하는 일이 줄어든다.
 - 화면이 달라지게 만드는 모든 요소는 이미 전부 리스팅 되어 있다.
- 화면을 바트릴 일이 사라진다.
 - 모든 상태의 조합에 대해 UI가 나와 있는지 체크하는 것으로 충분하다.
- 의사소통 비용이 줄어든다.

상태의 명시적인 목록이 있으면...

- 디자이너 🧠 프로그래머

- 새로운 경우의 수를 커버하기 위한 화면을 그림

- 이 상태 필드가 추가되었습니다.
- 이 상태에 대응하는 화면은 이겁니다.

- 기존에 있던 화면을 변경함

- 이 상태의 화면이 변경되었습니다.

- 프로그래머 🧠 디자이너

- 구현하다보니 새로운 경우의 수 발견

- 이런 상태 필드가 필요합니다.
- 새로 생긴 상태에 대응하는 화면이 필요합니다.

안 하고 있는 건 아닙니다

dodo 고객

방문 이력, 고객, 메시지, 광고

검색

전체, 신규, 적립우수, 생일

2015.08.22~2017.08.28 상세필터

7,283명 최근 방문순

010-***3-2700 | 이지은 17.01.25
보유 80,000P (누적 292,000P)
적립 53회

010-***4-1932 17.01.25
보유 1,200P (누적 2392P)
적립 53회

010-***2-2913 17.01.25
보유 100P (누적 9,000P)
적립 21회

010-***3-1203 17.01.25

이지은
010-***3-2700 수정

80,000P 보유 포인트 > 53회 적립 2장 쿠폰 >

유형 대학생, 아침 방문
연령 20대 (1989. 03.26)
메모 홍익대생으로 평일 점심 12시 방문 고객, 주로 단체 4인 이상으로 방문하시는 고... 더보기

방문 이력 누적 292,000P

+ 1,200P 포인트 적립 17.02.05 21:03

+ 1,200P 포인트 적립

요소		명세	레이블
검색창		이름, 휴대전화 번호, 메모로 검색 가능	이름, 휴대전화
퀵필터	전체	기간 내 방문 1회 이상 한 고객 필터링	전체
	신규	기간 내 최초 적립한 고객 필터링	신규

고객 선택 > 쿠폰 설정 > 메시지 작성 > 최종 확인

테스트 문자 발송하기

고객 적립 횟수 100번~120번, 방문 3개월 이내
쿠폰 더모커도 자조 케이크 증정
유효기간 2017.03.26 ~ 2017.06.30 (총 90일)
범위 1,200명 수정

Blue Bottle 및 고객 아메리카노 증정

Web업선

이슈리노, 대만 무로 증정도 20%

총 4,000 코인 사용
1,200 코인이 부족함.03

코인 승인

여전, 예약 발송, 즉시 발송

05-21 최종확인(맞춤/코인 부족)

1w

고객 선택 > 쿠폰 설정 > 메시지 작성 > 최종 확인

테스트 문자 발송하기

고객 생일 고객, 10~30대 여성
쿠폰 1+1 치즈 케이크 증정 | 유효기간 30일

Blue Bottle 및 고객 아메리카노 증정

Web업선

이슈리노, 대만 무로 증정도 20%

총 4,000 코인 보유

코인 승인

여전, 예약 발송, 즉시 발송

05-24 최종확인(자동/생일)

1w

naominam 3 months ago

고객 필터

- 전체
- 적립 횟수
- n번-n'번
- 미방문
- 30일 이상
- 2개월 이상
- 3개월 이상
- 6개월 이상
- 누적 포인트
- nP-n'P
- 누적 구매금액
- n원-n'원
- 보유 포인트
- nP-n'P

rock 3 months ago

"이하"는 없고 "n~n`" 만 처리하
? @naominam

👍 🤔 😞

안 하고 있는건 아닌데 왜?

- 앱의 (또는 페이지의) 상태
 - 복잡한 앱이라면 몇 단계에 걸쳐 감싸져 있기 마련입니다.
- 구글 드라이브, 제플린 코멘트, 컨플루언스 테이블의 공통점
 - 여러 단계에 걸쳐 구조화된 데이터를 표현하기 위한 포맷이 아닙니다.
 - + 접근성이 그렇게 좋지는 않다고 생각합니다.

잠재적 대안 : JSON

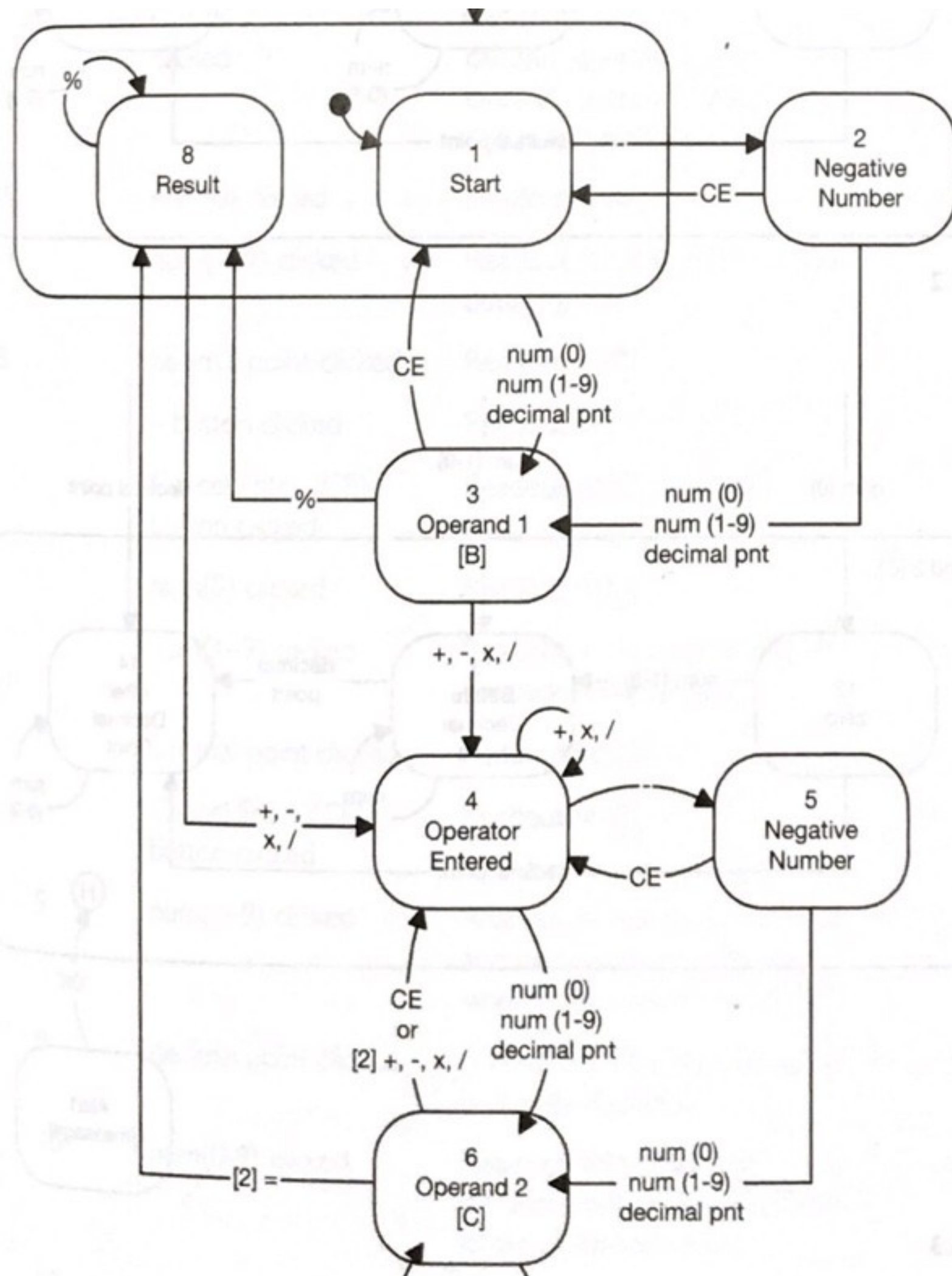
```
{ "widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}
```

- 구조화된 데이터를 표현하기에 적합
- 간단한 포맷이다.
- Github에 올려놓고 웹에서 바로 수정도 가능!
- 페이지마다 page.spec.json이 있다면?
- 더 좋은 방법이 있을 것 같긴 합니다.
- 뭐가 있을지 얘기해보고 싶어서 발표 중.

Bonus Track: 상호작용

- 어떤 시점에 어떤 화면이 보여야할지에 대한 리스트업이 끝났다고 칩시다.
- 그 화면에서 어떤 상호작용(= 데이터 조작)이 가능한지에 대해서는?
- 마찬가지로 상태로 나타낼 수는 있겠으나, 애는 생김새에 직접 영향을 주는 상태가 아닙니다.
 - 상호작용을 위한 상태를 별도로 관리?

Bonus Track: 상태 기계



- 가능한 상호작용의 경우의 수를 각각 상태로 나타내고, 어떤 경우에 상태 전이가 일어나는지 기술
- 비슷한 장점이 있다. (e.g. 해당 페이지/컴포넌트가 가질 수 있는 상호작용 상태를 모두 기술...)
- 보너스 트랙일 뿐입니다.

요약

- 최근 들어 React를 위시하여 많은 프론트엔드 라이브러리/프레임워크가 채택한, UI를 데이터의 함수로 바라보는 사고방식은 다양한 장점을 제공합니다.
- 우리도 (어느정도 수준으로) 이미 이렇게 제품 개발 하고 있습니다.
- 지금 하고 있는걸 더 잘하기 위해 앱을 개발하기 이전에, 그리고 개발하는 과정에서 상태 리스트업을 보다 간결하고, 접근성 높고, 명시적인 방법으로 해보는 건 어떨까요?
- 액션 플랜 추천 받습니다.

감사합니다.