Mixpanel Cheatsheet

A comprehensive cheat sheet covering Mixpanel's core functionalities, tracking methods, user identification, and people properties management. Includes code snippets and examples for quick implementation.

Core Concepts

Initialization

Before using Mixpanel, initialize it with your project token. This is typically done once on page load.

```
mixpanel.init('YOUR_PROJECT_TOKEN');
```

The init method sets up the Mixpanel object and prepares it for tracking data. Replace `'YOUR_PROJECT_TOKEN'` with your actual Mixpanel project token.

Identifying Users

Identifying users is crucial for tracking their behavior across sessions. Use `mixpanel.identify()` to set a unique user ID.

```
mixpanel.identify('user123');
```

Alternatively, you can use `mixpanel.alias()` to connect anonymous users to identified users, typically after registration or login.

```
mixpanel.alias('newUser',
mixpanel.get_distinct_id());
```

Tracking Events

Track user actions using `mixpanel.track()`. Events can include properties providing additional context.

```
mixpanel.track('Product Viewed', {
'product_id': '456', 'product_name': 'Awesome
Gadget' });
```

Customize event tracking with distinct properties to segment and analyze data effectively.

People Properties

Setting User Profile Properties

Use `mixpanel.people.set()` to store user profile information, such as demographics or preferences.

```
mixpanel.people.set({ 'name': 'John Doe',
'age': 30, 'signup_date': new Date() });
```

Setting properties allows for detailed user segmentation and personalized messaging.

Incrementing Numeric Properties

You can increment numeric properties using `mixpanel.people.increment()`. This is useful for tracking metrics like purchase counts or points.

```
mixpanel.people.increment('total_purchases',
1);
```

Incrementing ensures accurate tracking of cumulative metrics over time.

Appending to List Properties

Append values to list properties with `mixpanel.people.append()`. This is great for tracking a user's interests or viewed categories.

```
mixpanel.people.append({ 'viewed_categories':
'electronics' });
```

Appending allows for dynamic tracking of user activities and interests.

Advanced Tracking

Registering Super Properties

Super properties are automatically included with every event. Use `mixpanel.register()` to set these.

```
mixpanel.register({ 'app_version': '1.2.3',
'platform': 'web' });
```

Registering super properties simplifies event tracking and ensures consistent data across events.

Timing Events

Measure the duration of specific processes using `mixpanel.time_event()` before tracking the event.

```
mixpanel.time_event('Image Upload');
// ... (code for image upload) ...
mixpanel.track('Image Upload'); // Duration is
automatically included
```

Timing events provides valuable insights into user experience and performance bottlenecks.

Tracking Revenue

Track revenue generated by users with `mixpanel.people.track_charge()`. This updates the user's total revenue.

```
mixpanel.people.track_charge(49.99, {
'product_id': '789', 'product_name': 'Premium
Access' });
```

Revenue tracking is essential for understanding the monetary value of users.

Engagement

A/B Testing

Mixpanel supports A/B testing. Use event tracking to measure the performance of different variations.

```
mixpanel.track('A/B Test Result', {
'variation': 'control', 'conversion': true });
```

Track user interactions to optimize content and offers.

Sending Push Notifications

Integrate Mixpanel with push notification services to engage users. Track push notification opens and conversions.

```
mixpanel.track('Push Notification Opened', {
'notification_id': '123' });
```

Push notifications can drive user engagement and retention.

Using Funnels

Define funnels in Mixpanel to track user progression through key steps, such as signup or purchase flows.

```
mixpanel.track('Funnel Step 1', { 'user_id':
'user456' });
```

Analyze funnel data to identify drop-off points and improve user flows.