



Navigation

Navigation guides users between different parts of your app.

Your app's structure should be organized according to the content and tasks you want users to see. Focus attention on important destinations by placing them in tabs or side navigation. Alternatively, de-emphasize inessential content by placing it in less prominent locations.

Usage

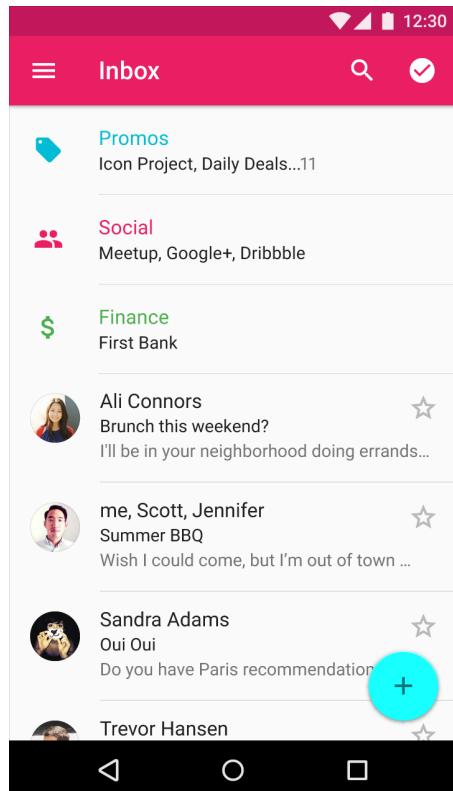
Navigation through your app should be intuitive and predictable. Both new and returning users should be able to figure out how to move through your app with ease.

When users move from one view to the next, they observe scenes containing imagery, actions, and content. These scenes come together to tell a story about the content they contain, leading users down paths that emphasize certain actions.

For information on movements between states, see [navigational transitions](#).

Communicate relationships & context

Group content into logical and relatable chunks that show clear relationships between items.



This image shows relationships between email items by grouping them into these clusters: Promos, Social, Finance.

Illuminate actions

Guide users from one scene to the next. Task: **Task flow path**
that encourage specific choices or free exploration.



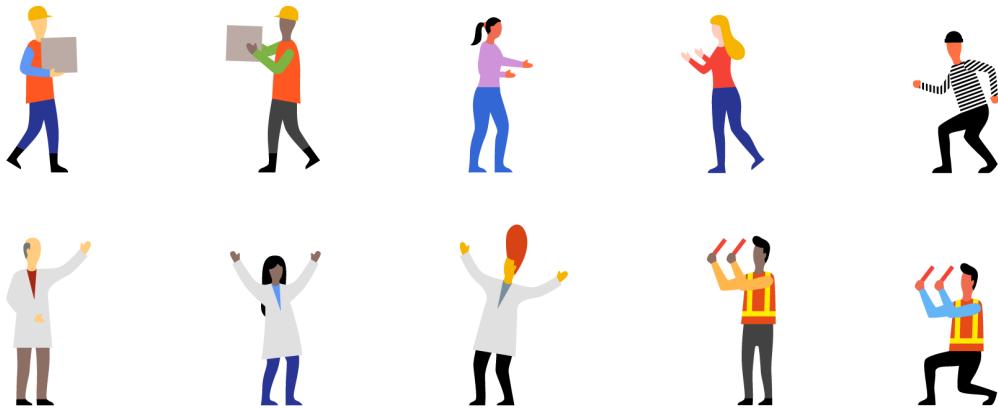
Focus attention

Task: **Task flow path**
app's navigation to pr that encourage specific choices or free exploration.



Use simple design and lightweight structure when user needs are simple.

Defining your navigation



Navigation organizes content so that it's easier to find important destinations in your app. Navigation may include first-time setup, visited places, contain settings, or encourage specific actions.

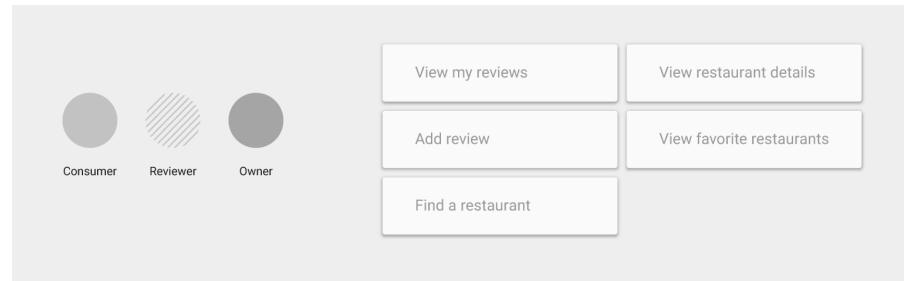
To determine the type of navigation best suited to your app, identify your app's users, typical paths they might take through your app, and actions you want them to perform.

For instance, if your app is a restaurant guide, your users may want to make reservations, post menus, or write reviews. By identifying the goals most users will have, you'll be able to better define a navigation structure that meets their priorities.

Take inventory

Identify your app's users and their potential roles, such as consumer, business owner, or journalist.

Identify the most common tasks they may want to perform.

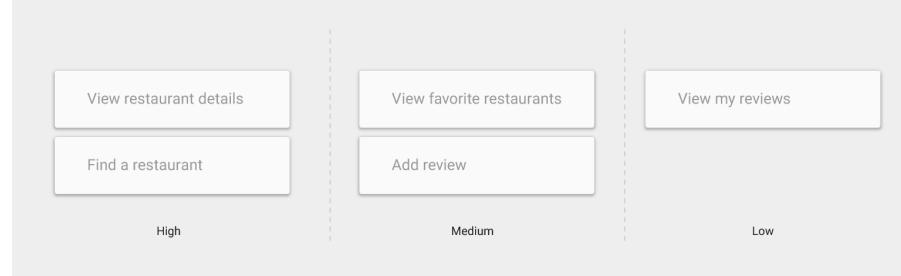


Identify your key users and the use cases they might want to perform.

Prioritize

Assign priority levels of high, medium, or low to common user tasks. Give prominence in the UI to paths and destinations with high priority levels and frequent use.

Adjust priority levels as user needs change over time.



When designing navigation for a restaurant app, the highest priority tasks are restaurant details or finding new restaurants.

Sequence

Identify the different paths that users take through your app and use those paths to define your navigation:

- List frequent destinations prominently in your navigation
- Group related tasks together and use those groupings to structure your navigation



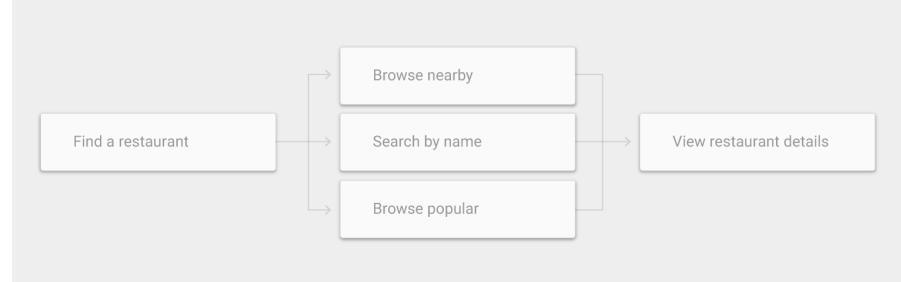
Whether a consumer wants to find a new restaurant or view a favorite one, they can quickly access the destination via the navigation sequence. Make frequently used paths to these destinations (e.g., favorites) easily available.

Deconstruct

Divide complex, broad, or vague use cases into smaller activities.

These smaller tasks may be frequently used, more easily understood, or better meet user goals.

For example, dividing search into smaller activities that differentiate searching by name, location, or popularity may reveal the need to surface those smaller activities as part of navigation.



The broader activity of “Find a restaurant” may actually encapsulate smaller tasks like Browse nearby, Search by name, and Browse popular. These smaller tasks help users to perform searches and should be integrated within your navigation.

Hierarchy

Navigation organizes content by placing it within a hierarchy. Scenes at the top of the hierarchy are the most prominent and accessible, connecting users to scenes lower in the hierarchy.



Home

The **home** scene is the entrance to the app. It introduces the app and its navigation.

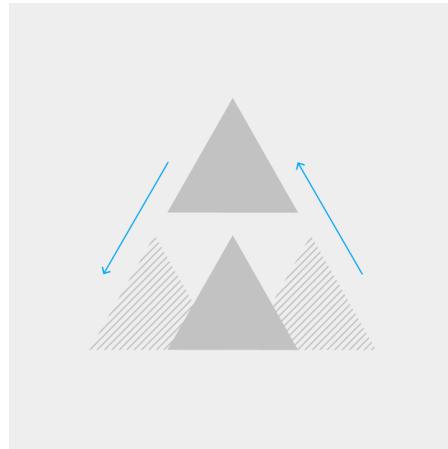
The home scene may vary the content it displays. A **stateful home** contains indications of a user's previous interaction with the app.



Parent and child

A higher level of navigational hierarchy is called a **parent**, and the level (or levels) below are referred to as **children**.

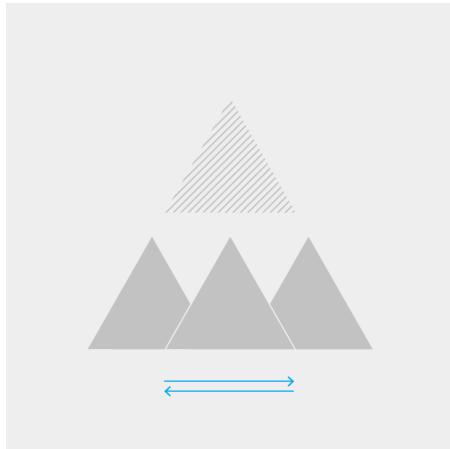
For example, the home screen is the parent to all other screens in an app.



Navigation

When you navigate more deeply into an app, you descend from a higher level of hierarchy to a lower one.

Moving from a parent scene to a child scene is **descending navigation**. Moving from child to parent is **ascending navigation**.



Siblings

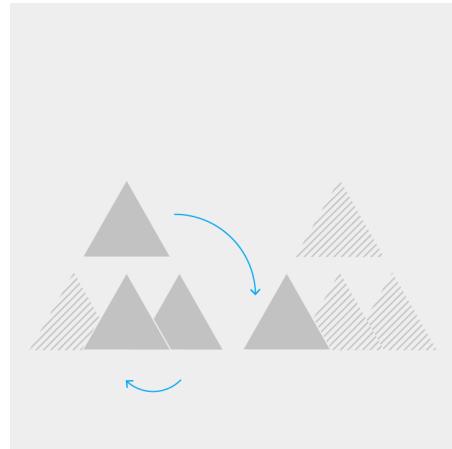
Scenes that have the same parent are called **siblings**. Apps that have multiple use cases, all of equal priority, usually contain many sibling scenes.

Lateral navigation is movement between siblings.



Collections

Collections include multiple items that share the same parent.



Links

Links allow users to move quickly between scenes that aren't navigationally adjacent.

Crosslinks are links that take users anywhere within your app.

External links are links from sources outside your app. They take users directly to scenes deep within your app.

Patterns

Navigation should best accommodate the needs of the majority of your app's users. Apps with complex and deep hierarchies (that require switching between multiple views) have different navigation requirements than apps with a single primary view.

The following navigation patterns are designed to:

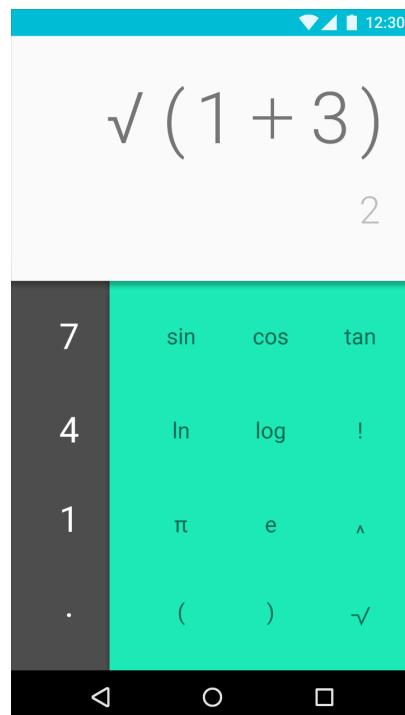
- Promote engaging and responsive layouts
- Anchor navigation and actions
- Highlight important destinations and de-emphasize infrequent ones
- Promote a primary action with the floating action button

Embedded navigation

Apps with simple navigation may embed navigation inside app content, but doing so reduces the available space to display that content.

Recommended for:

- Apps with a strong primary view, and few alternate views
- Apps that perform common tasks in the main view
- Infrequently used apps



Mobile example of an app with embedded navigation. Common tasks are performed in a strong primary view.

Tabs

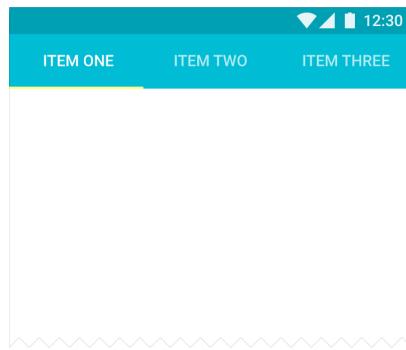
Tabs allow users to quickly move between a small number of equally important views.

Appropriate for these hierarchies:

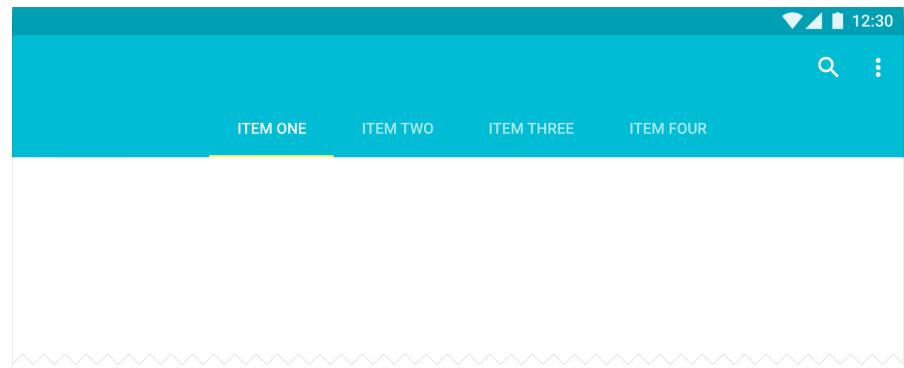
- Parents with embedded child views
- A group of sibling views

Recommended for:

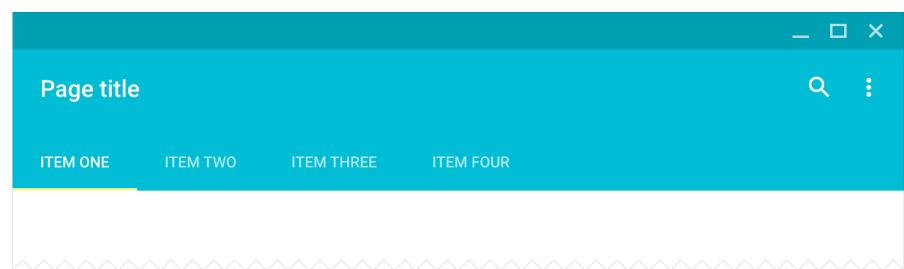
- Frequent switching between views
- Apps with few top-level views
- Promoting awareness of alternate views



Mobile example with three tabs



Tablet example of centered tabs

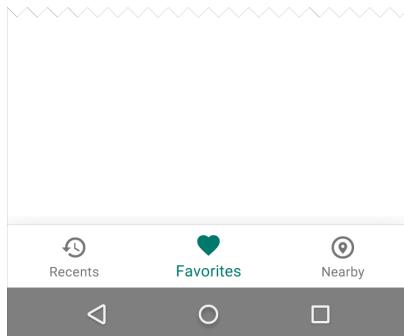


Desktop example of left-aligned fixed tabs with title and extended header

Bottom navigation bar

A **bottom navigation bar**

allow users to quickly move between a small number of top-level views.



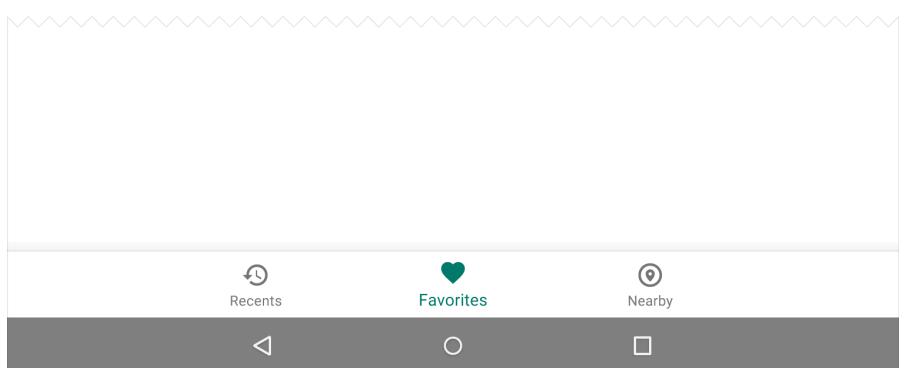
Appropriate for these hierarchies:

- Parents with embedded child views
- A group of parent views

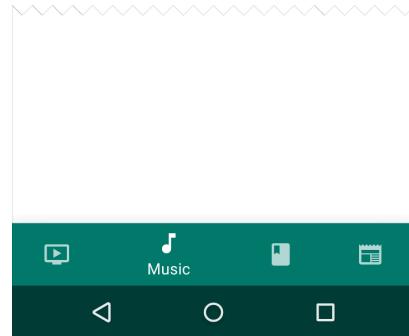
Recommended for:

- Frequent switching between views
- Apps with few top-level views
- Promoting awareness of alternate views
- Mobile devices, as bottom navigation is located in a more ergonomic location

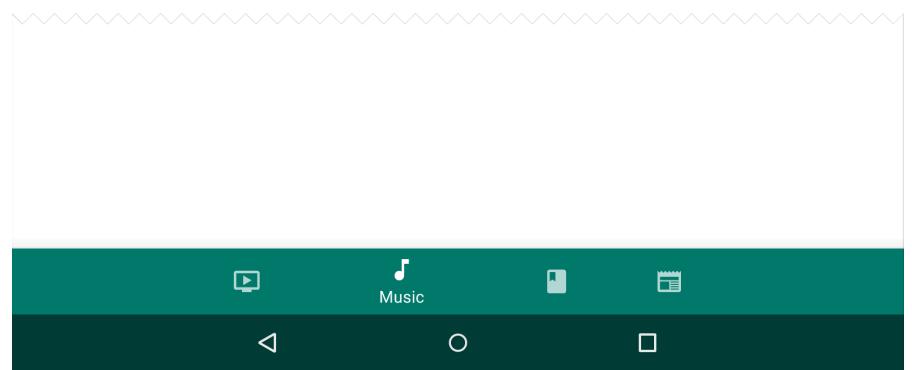
Mobile example with fixed text labels present at all times



Tablet example with text labels present at all times



Mobile example with shifting
text & labels



Tablet example with shifting text & labels

Navigation drawer

When there is insufficient space to support tabs, side navigation is a good alternative. Side navigation can display many navigation targets at once. A drawer remains hidden until invoked by the user.

Apps with a single “home” should list the most frequently accessed destinations at the top of side navigation.

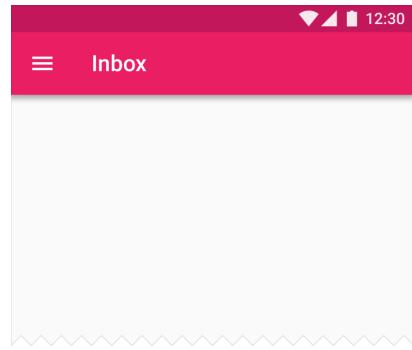
Side navigation may appear either with or without a **navigation drawer**.

Appropriate for these hierarchies:

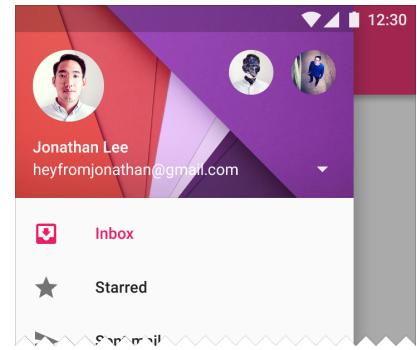
- Lateral navigation
- Parents with siblings or peers

Recommended for:

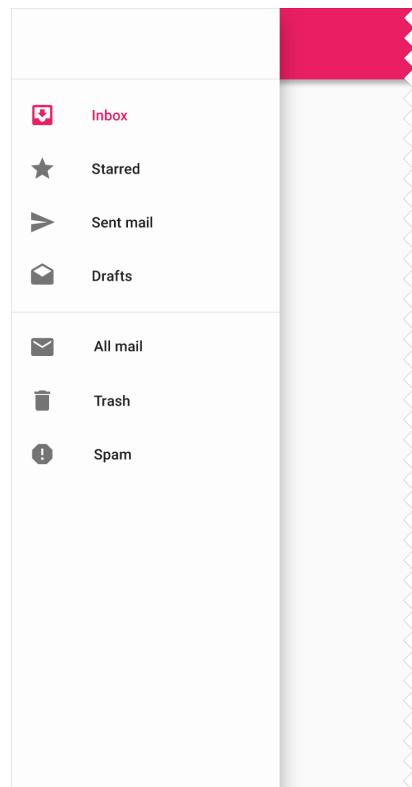
- Apps with many top-level views
- Enabling quick navigation between unrelated views
- Deep navigation structures
- Reducing visibility of infrequent destinations



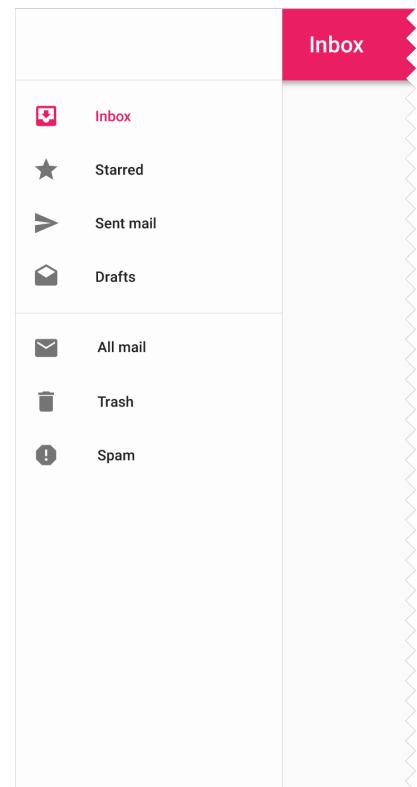
Mobile example of a closed left nav



Mobile example of an open left nav



Tablet example of an open left nav



Desktop example of an open and pinned-in-place left nav

Nested Navigation

When you have multiple levels of navigation, sibling views should be nested underneath their parent.

On desktop, a secondary level of navigation may be nested within the navigation drawer.

Appropriate for these hierarchies:

- Lateral navigation
- Parents with siblings or peers

Recommended for:

- Deep navigation structures with many views
- Enabling quick navigation between unrelated views

The Fortnightly

Americas

World

Africa

Americas

Asia Pacific

Europe

Middle East

U.S.

Politics

Business

Tech

Sports

Arts

Fishing Industry Drops

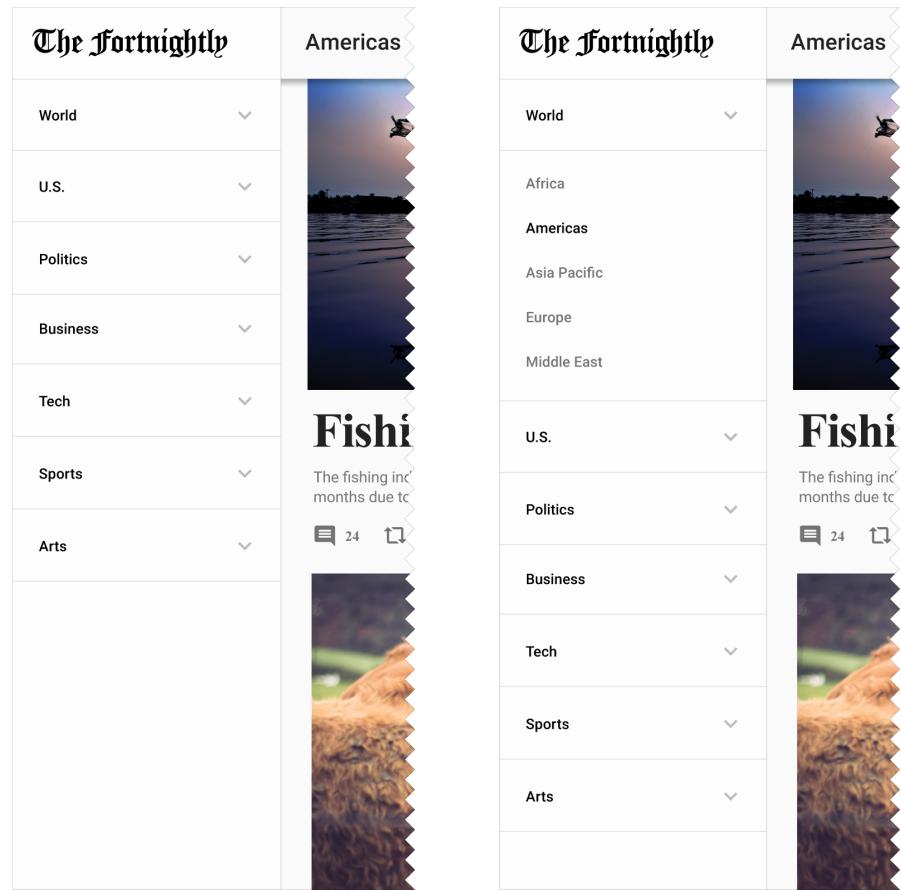
The fishing industry in the Florida Keys has been downturn in recent months due to a catastrophic storm that hit the reg...

24 3

Desktop example of an expanding left nav

Expanding navigation drawer

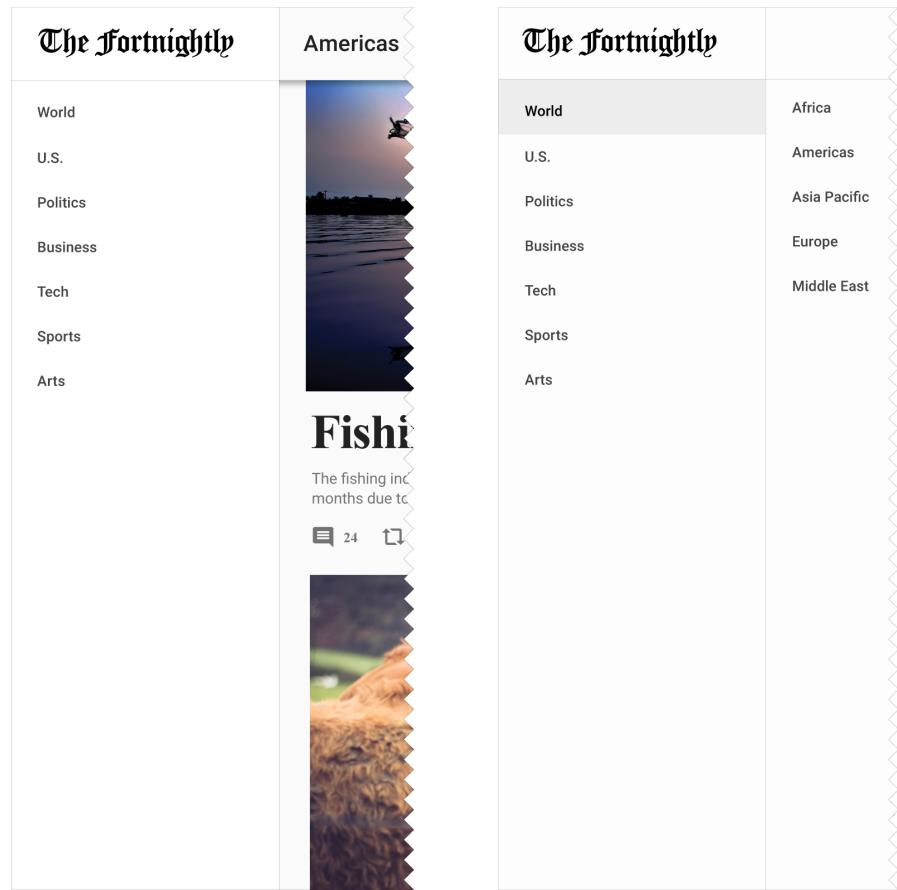
If you have a deep navigational hierarchy, you may expand that hierarchy within the navigation drawer. Upon selecting a level, the level of navigation below is revealed. Selecting a collapsed section expands that level in-line, hiding all levels outside of it.



Desktop example of an expanding left nav

Cascading navigation drawer (Desktop only)

If you have a deep navigational hierarchy, you may use cascading menus to expand navigation drawer content horizontally. Each collection of views is presented in its own panel, and a level closes when a subsection is selected.



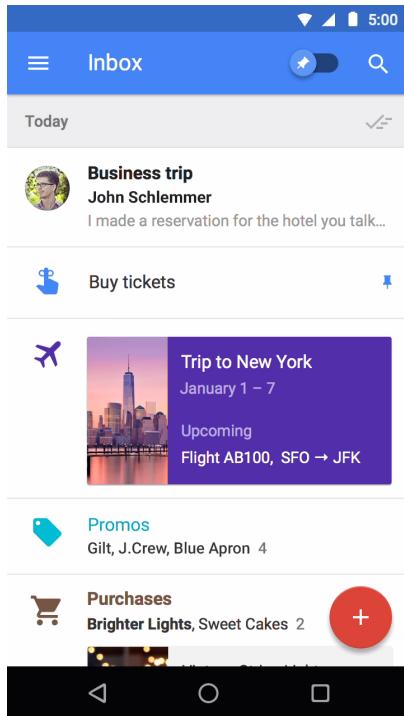
Desktop example of a cascading left nav

Gestural

Gestural navigation allows users to use swipe gestures to navigate between sibling or peer views. Supported gestures include: touching and dragging the screen horizontally (left/right), vertically (up/down), or while zooming in or out.

Recommended for:

- Naturally ordered relationships, such as pages representing consecutive calendar days
- Views with few siblings
- Views with similar content types



When in Inbox, you are able to pull down on a message to return back to the parent view.

See [Navigational transitions](#) for more information on hierarchical journeys between scenes.

Combined patterns

In-context navigation

In-context navigation (used



2016/3/22

with tabs or navigation drawers) allows nimble movement between related sets of data.

In-context navigation integrates into an app's content and is typically less linear in nature. When moving through in-context navigation, navigating to a new scene will display navigational controls specific to that view.

Appropriate for these hierarchies:

- Parents with siblings

Recommended for:

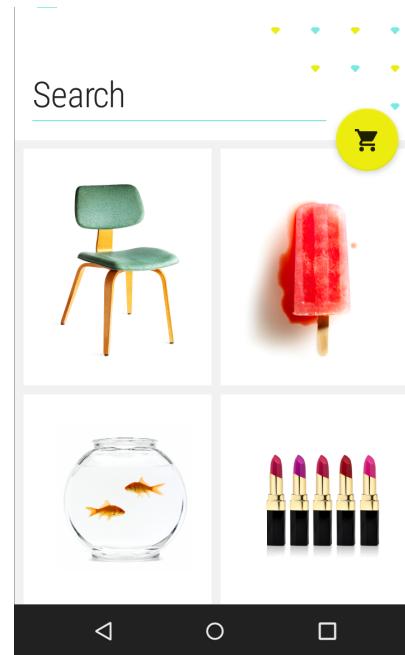
- Showing large sets of child scenes
- Focusing attention on a specific set of tasks
- Apps with a strong primary view, and few alternates

Example uses:

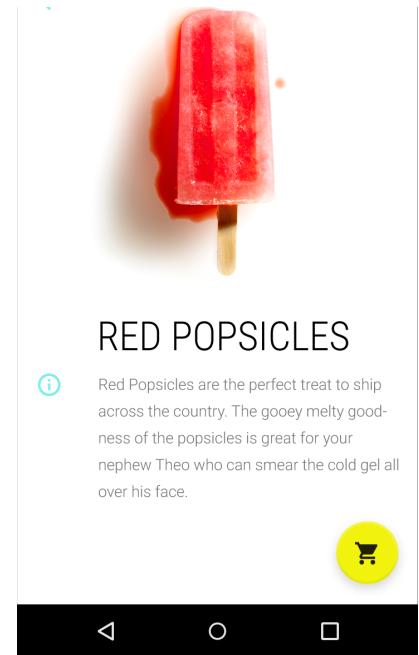
- Link a song to an artist
- Move between recent items and a complete history
- Connect a user's post to their profile page

The up arrow is used to return to the previous level.

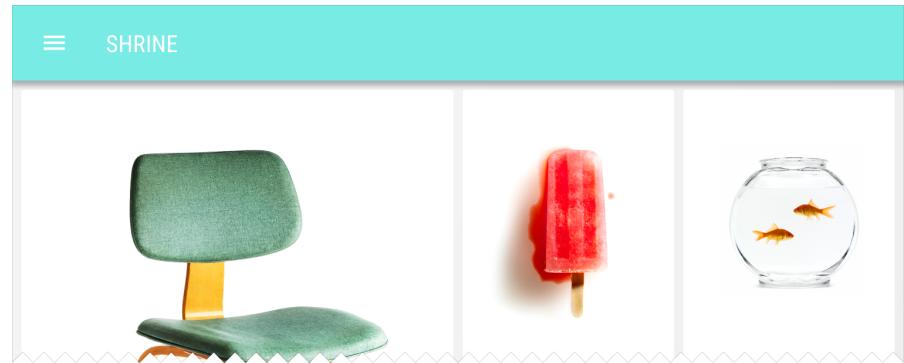
Navigation - Patterns - Google design guidelines



Mobile example of navigation integrated into content



Mobile example of up arrow



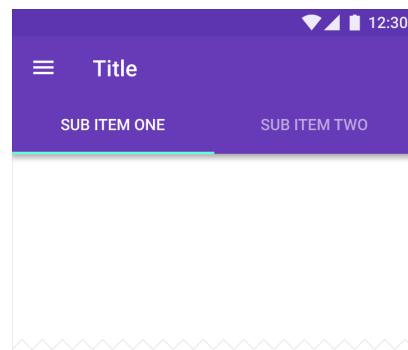
Tablet/Desktop example of navigation integrated into content



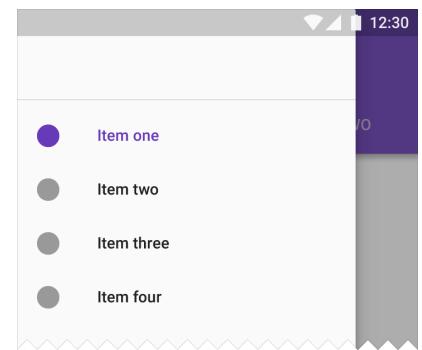
Tablet/Desktop example of an up arrow

Side nav and tab combinations

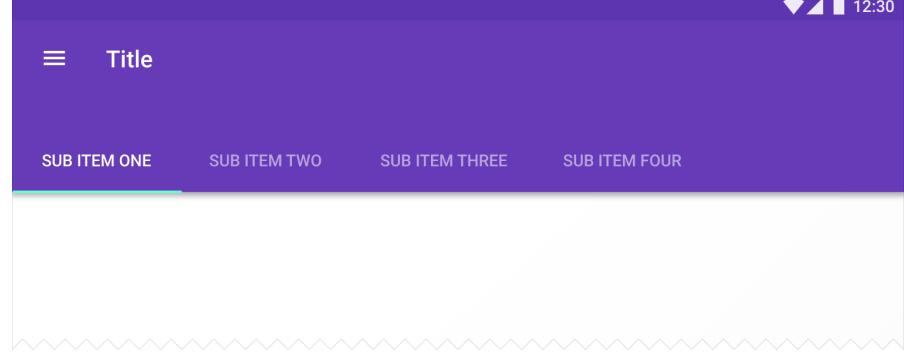
Products with two levels of navigation may pair a left navigation drawer with tabs.



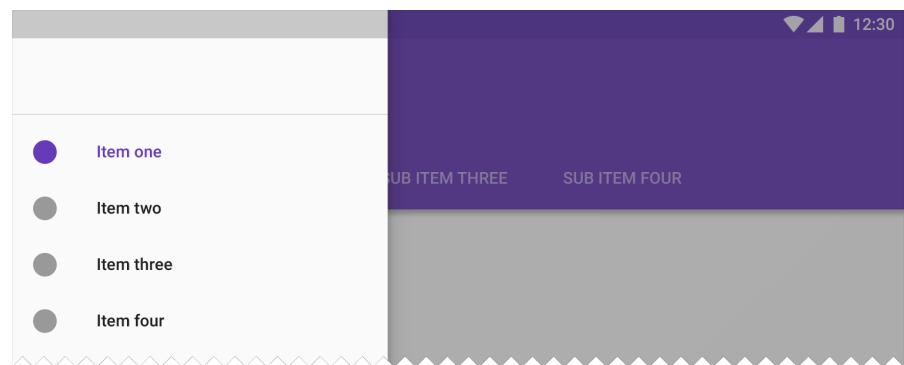
Mobile example using two levels of navigation: a navigation drawer and tabs



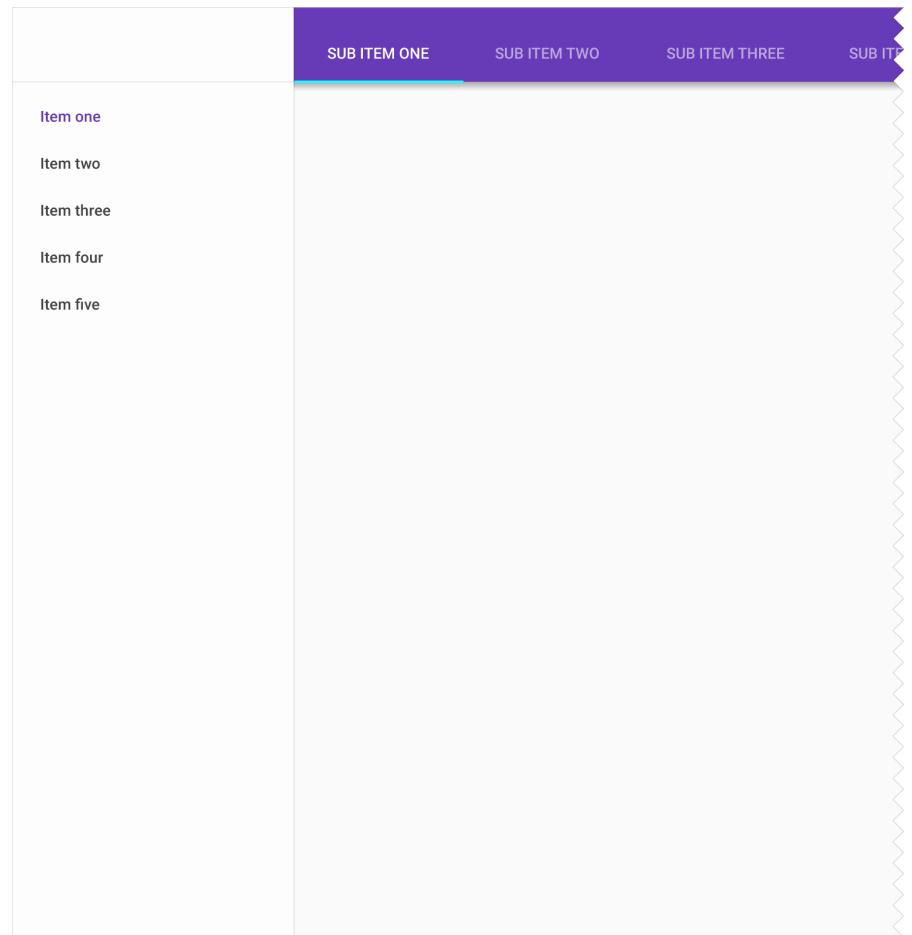
Mobile example of an open left nav



Tablet example using two levels of navigation: a navigation drawer and tabs



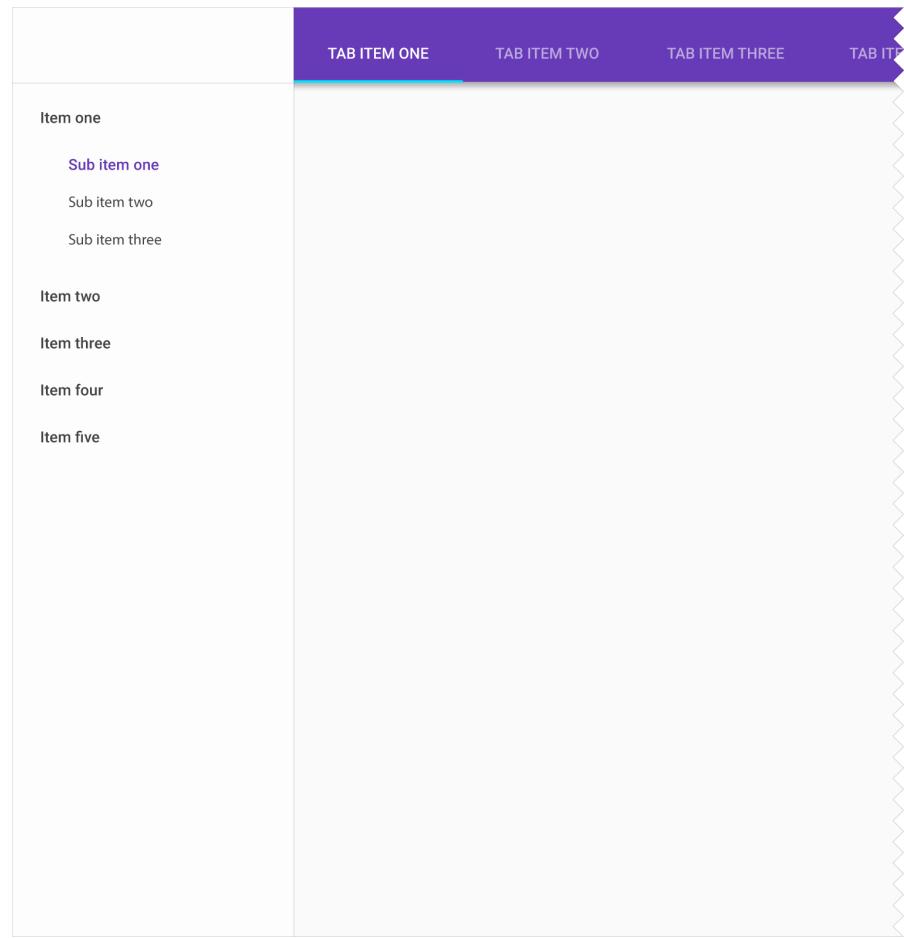
Tablet example of an open left nav in front of tabs



Desktop example using two levels of navigation: a navigation drawer and

Nested navigation drawer + tabs (Desktop only)

If your product has three levels of navigation, pair a navigation drawer with nested navigation and tabs.



Desktop example of a nested left nav with tabs