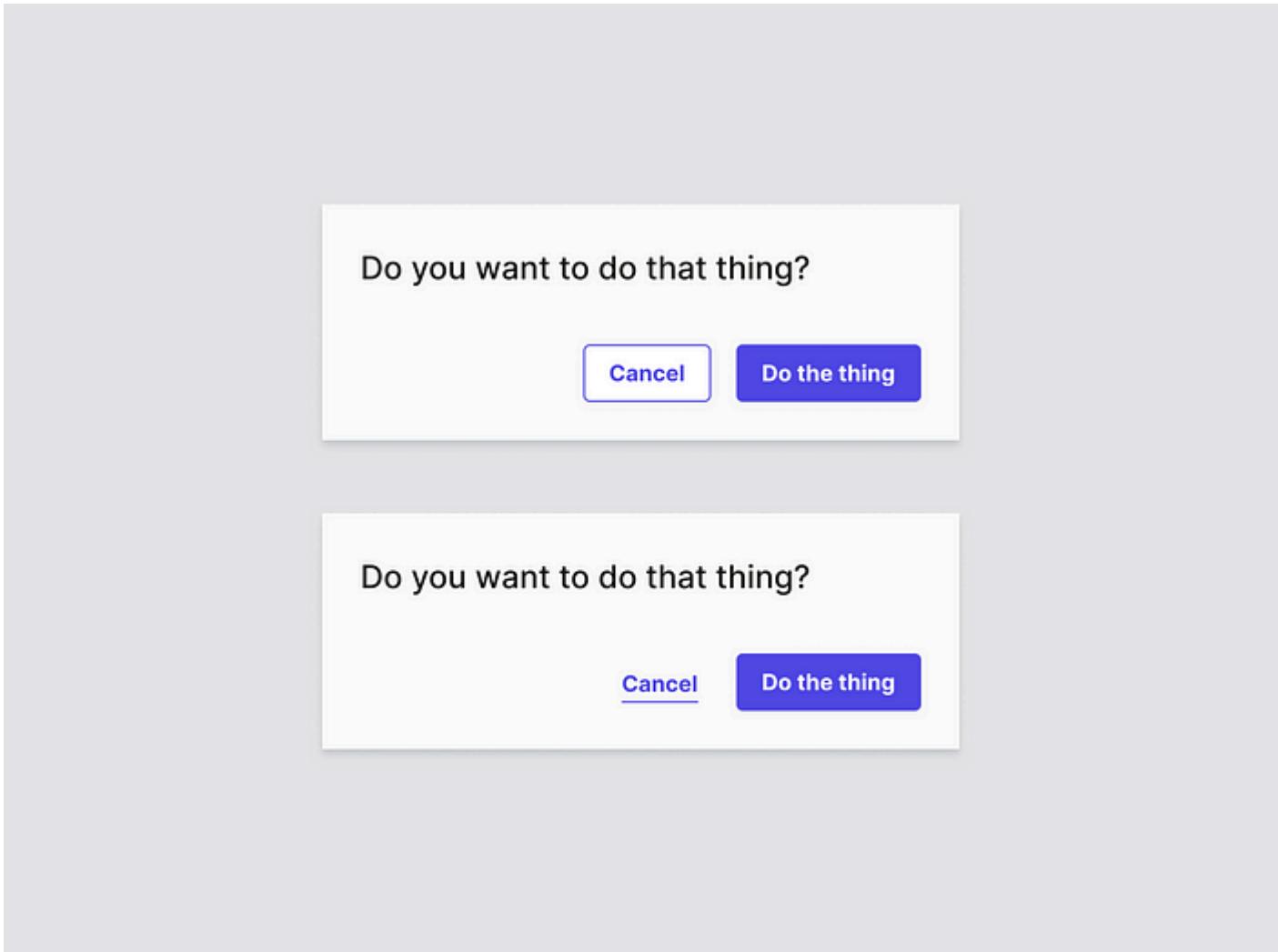


[< Go to the original](#)

Should "cancel" be a button or a link?

Which one is UX best practice?

**Karim Maassen**[Follow](#)

UX Collective a11y-light ~10 min read · March 21, 2022 (Updated: June 29, 2023) · Free: No

A common type of dialogue that is often presented when interacting with a web app, is the "cancel" option. Usually when the user is filling out a form, or manipulating data (e.g. inside a side panel, detail page or modal view). One would typically see it next to a primary action, presented as a link. More and more however, this cancel option is presented as a button. Is this good or bad practice, and why?

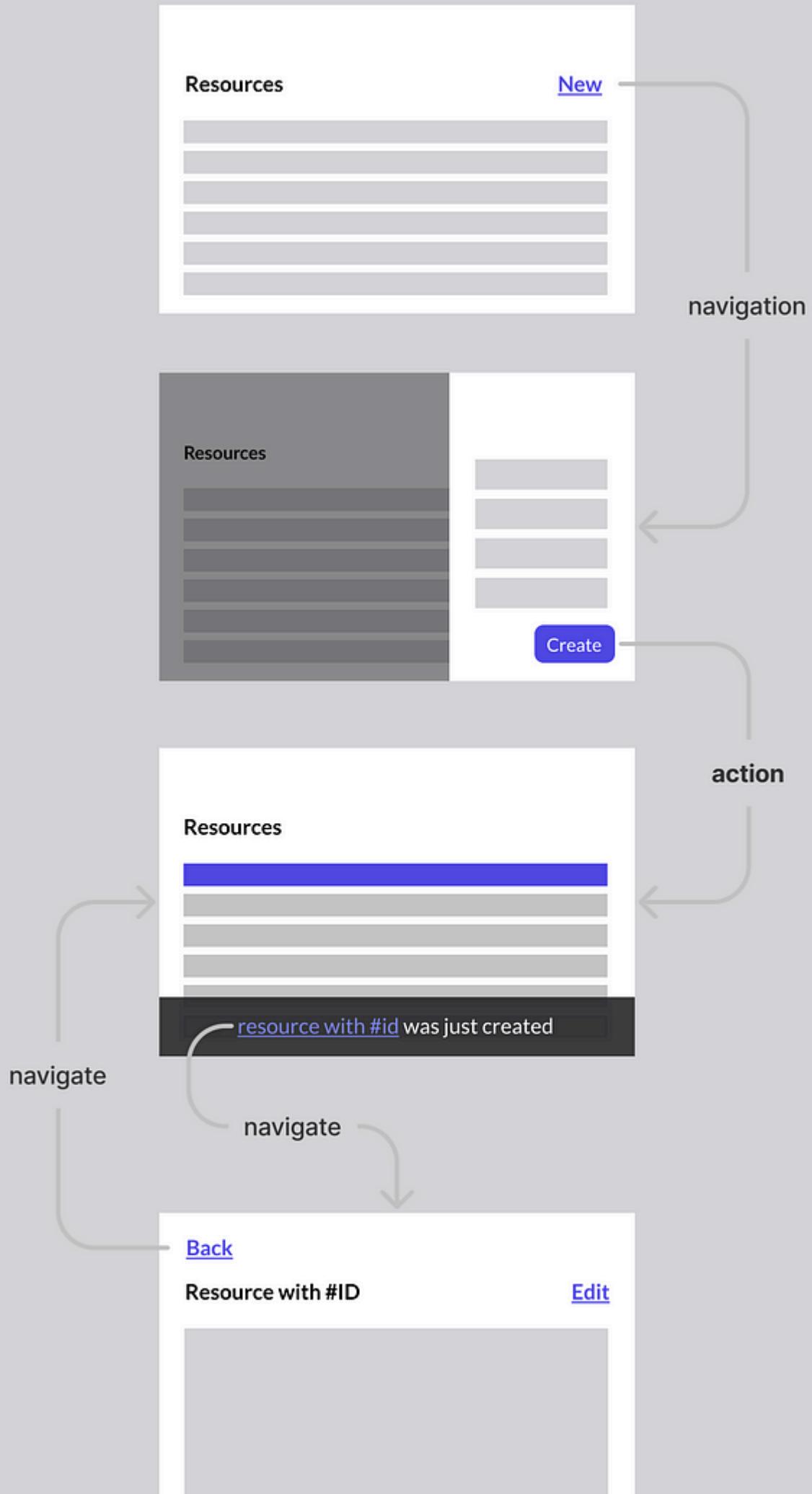
Navigation vs. Action

When looking at a web app, a user typically has to deal with two types of interaction: navigation and actions. Both types of interactions are usually represented in a different but distinct way.

1. Navigation is a link.
2. Action is a button.

Why is this? Links stem from the early days of the world wide web. It's what it makes it be a web (and also world-wide). Buttons are a form of skeuomorphism that translates the physical world of flipping switches and pressing buttons to do stuff, into the digital world of the internet. And thus, they are used differently:

Actions are manipulation of data the user is managing through the interface of the web app, while navigation is a state change of the interface the user is, well, navigating through. In other words, links (or navigation) don't lead to changes of the underlying data, but can lead to the presentation of new pages, or a different presentation of the current page the user is navigating in (i.e. change the view). So when we look at it this way, we are basically saying that navigation is a state change of the UI and an action is data manipulation. In other words, going from one state to the other is the same as *navigating* from one state to the other. I'd like to note that actions (buttons) *can* lead to navigation, but it's not a requirement.



UI Design How one would visualize a link and button is of course a design style decision, but typically links are blue underlined text and buttons are boxes with text inside of them. Both links and buttons can contain icons to emphasize the intended action or navigation (Relying on icons only is discouraged, as this would make a user needlessly think about said action or navigation, but that's a different topic to discuss).

Information Architecture Obviously, having your web application adhere to a certain information architecture which allows for this behavior, is beneficial to the user experience of your solution. That said, understanding the principles of how an application is representing its data, is key to understanding the difference between data manipulation and state changes of your user interface.

Let's take the example of a fruit shop, which sells fruit. This shop could have a list of all fruits they sell. This "list of resources" would be the "list of fruits" for this specific shop:

Fruits (10)

[+ Add Fruit](#)

Type ^	Color ▾	Weight (g) ▾	Origin ▾
Apple	Red	110	Netherlands
Apple	Green	115	Belgium
Apple	Green	95	Netherlands
Banana	Yellow	190	Colombia
Banana	Yellow	210	Costa Rica
Banana	Yellow	185	Colombia
Lemon	Yellow	110	Netherlands
Lemon	Yellow	85	Spain
Lemon	Yellow	75	Spain
Lemon	Yellow	90	Italy

A user can decide to do the following:

- Add a new fruit to the list.
- Filter or sort the list of fruits
- Go to the detail view of a specific fruit from the list
- Go to the detail view of the origin of a specific fruit from the list

Let's find out how these user interactions would look like when applying our logic to it.

Applying this logic to a typical user interface

Add new fruit When wanting to add a new fruit to the list, the user clicks the "add fruit" link, which would lead to a state change of the UI (i.e. a form to fill out the new fruit appears).

It's merely a state change that shows the view to create a new fruit. Hence, it's navigation. By doing this, it sends the message to the user they are not doing data manipulation directly when clicking on "add new fruit" but can expect a new view to perform said task. Therefore, I advice against the FAB or a button in general for this interaction.

Fruits (10)

+ Add Fruit

Type ^	Color	Weight (g)	Origin
Apple	Red	110	Netherlands
Apple	Green	115	Belgium
Apple	Green	95	Netherlands
Banana	Yellow	190	Colombia
Banana	Yellow	210	Costa Rica
Banana	Yellow	185	Colombia
Lemon	Yellow	110	Netherlands
Lemon	Yellow	85	Spain
Lemon	Yellow	75	Spain
Lemon	Yellow	90	Italy

Fruits (10)

Type ^	Color	Weight (g)	Origin
Apple	Red	110	Netherlands
Apple	Green	115	Belgium
Apple	Green	95	Netherlands
Banana	Yellow	190	Colombia
Banana	Yellow	210	Costa Rica
Banana	Yellow	185	Colombia
Lemon	Yellow	110	Netherlands
Lemon	Yellow	85	Spain
Lemon	Yellow	75	Spain
Lemon	Yellow	90	Italy

+

Buttons instead of links where navigation is the default interaction. Thanks, Material UI.

The screenshot shows a list of fruits on the left and an 'Add Fruit' modal on the right.

Fruits (10) List:

Type	Color	Weight (g)
Apple	Red	100
Apple	Green	100
Apple	Green	100
Banana	Yellow	100
Banana	Yellow	100
Banana	Yellow	100
Lemon	Yellow	100

Add Fruit Modal:

- Type: Apple
- Weight (g): 0
- Origin: Belgium
- Buttons: Cancel (+ Add Fruit)

"Add Fruit" here indeed is an action. It confirms the intention of the user and sends a data change request to the server.

This interaction actually brings up an interesting discussion: what will the user see when having added the new fruit? Will they go back to the list of fruits, or will they go to the newly added fruit detail page? This really depends on what is important for the user and is something that needs to be determined through user research. Both options are valid, but **both are navigation**, albeit triggered by the performed action. (remember I said that actions *can* lead to navigation?)

Ignoring the "cancel" interaction for a second and let's dive into the other interactions a user can do, so that we can build up to the final answer:

Filter & Sort When going through the list of fruits, the user can decide to narrow their list by applying a filter. They decide to apply the "apples" filter and sort the

has been manipulated at all. All fruits in the list still exists. None were added or altered. The only thing the user did, was applying a different view, or, changing the state of the UI. This means, filters or sorting would warrant a link instead of a button:

Fruits (10)

[+ Add Fruit](#)[All Types](#) [All Colors](#) [All Weights](#) [All Origins](#)

Type ^	Color	Weight (g)	Origin
Apple	Red	110	Netherlands
Apple	Green	115	Belgium
Apple	Green	95	Netherlands
Banana	Yellow	190	Colombia
Banana	Yellow	210	Costa Rica
Banana	Yellow	185	Colombia
Lemon	Yellow	110	Netherlands
Lemon	Yellow	85	Spain
Lemon	Yellow	75	Spain

Fruits (3/10)

[+ Add Fruit](#)[Apples](#) [All Colors](#) [All Weights](#) [All Origins](#)

Type ^	Color	Weight (g)	Origin
Apple	Red	110	Netherlands
Apple	Green	115	Belgium
Apple	Green	95	Netherlands

Filtering and Sorting

Clearing a filter This also makes it easier to understand that a user interaction that denotes "clear all filters" or "reset sorting" should not be a button (i.e. it's not an action), but warrants a link (i.e. it's a state change of the UI, or navigating back to the preview state of the UI) as no data has been altered:

Fruits (2/10) [+ Add Fruit](#)

[Apples](#) [Green](#) [All Weights](#) [All Origins](#) [Clear all filters](#)

Type ^	Color	Weight (g)	Origin
Apple	Green	115	Belgium
Apple	Green	95	Netherlands

Clear are filters as a link (it's a UI state change and not actual data manipulation), denoted in "destructive" red to warn the user.

Going back to the preview page As said earlier, it's a valid interaction to have the user automatically navigate to the newly added fruit page after the action has been performed. But what if the user wants to go back to the previous page (list of fruits)? Well, this of course depends heavily on how this is presented in the UI, but typically one would expect the user to have access to either one of these options (or both):

1. Back navigation
2. Breadcrumb

Realizing the fact that in both cases, the user would still just be *navigating* back to a previous state of the UI (and thus, not undoing the performed action on the data, nor doing any new type of data manipulation), both options would be links:

[← Back to all Fruits](#)

Apple

 [Edit](#)

detail information

[Home](#) / [Fruits](#) / Apple (ID:100)

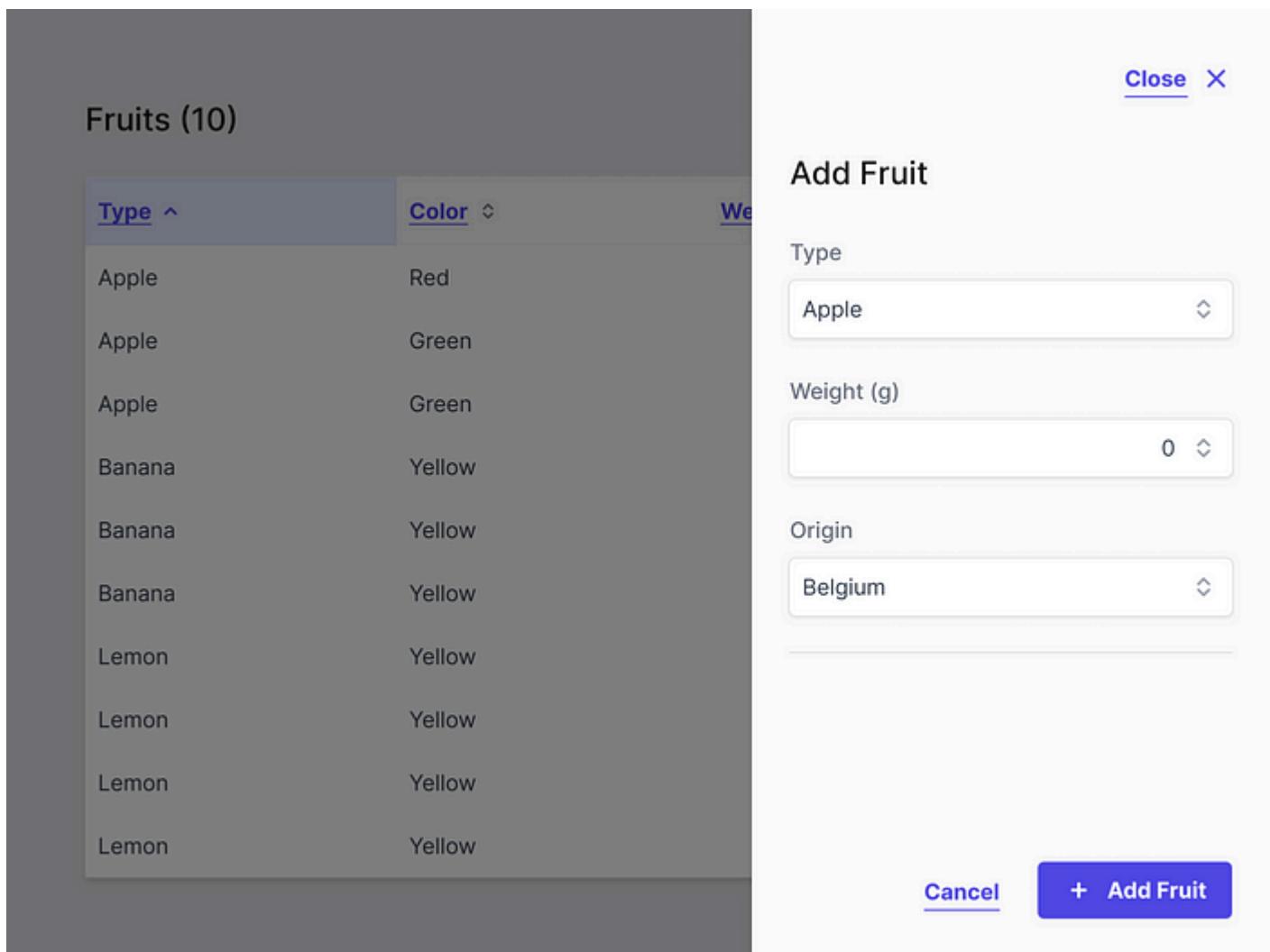
Apple

 [Edit](#)

detail information

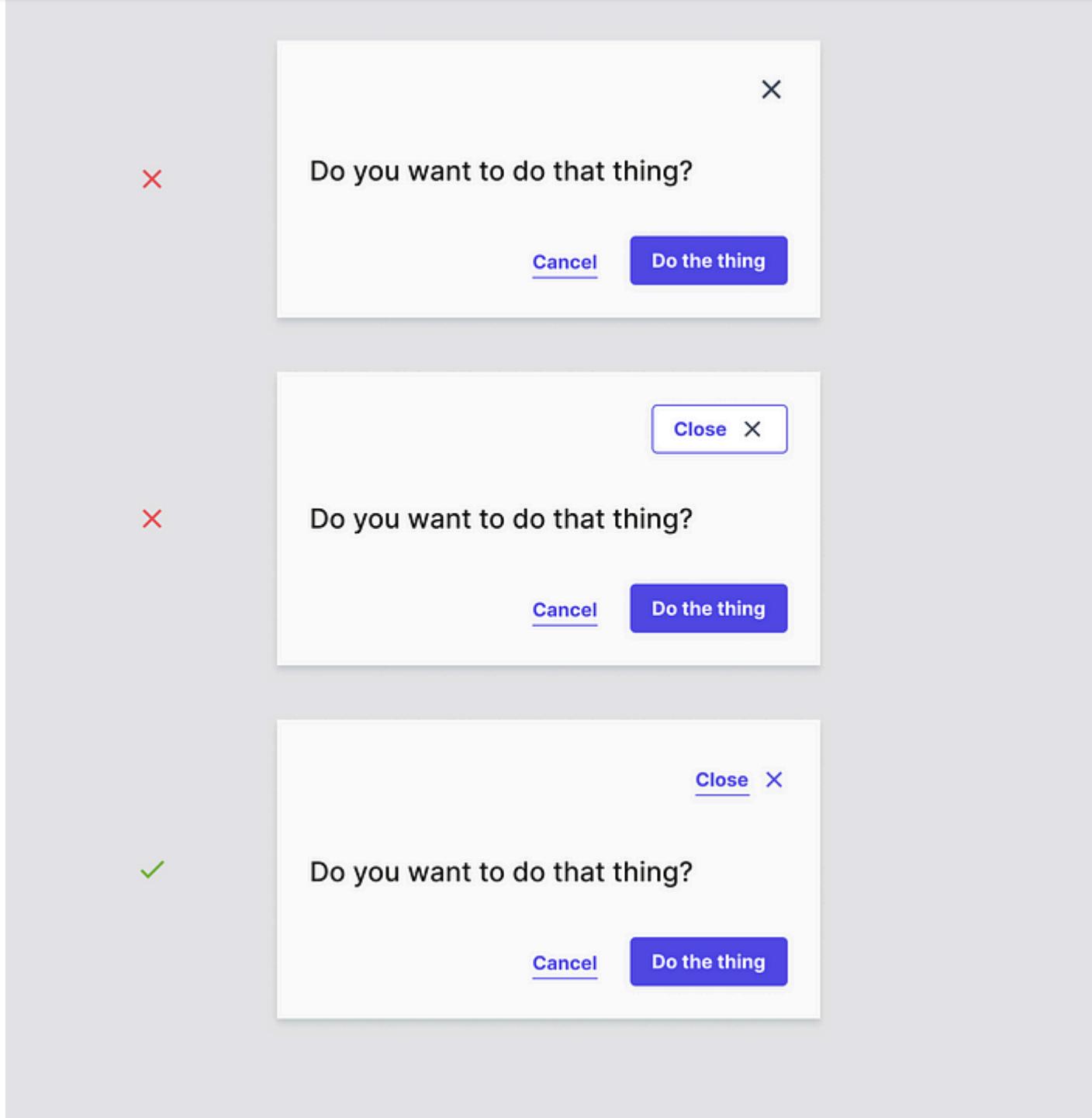
Now, a third option would also typically exist; the back button of the browser. Yes, this is a button, not a link. I will address the OS elephant in the room later in this article.

Closing the side panel And thus this brings us one step closer to the main question of this article. What if the user wants to close the side panel of the "add new fruit" view? Well, typically this would be done by either clicking outside of the view (dismissing it), or clicking a "close" option. Again, asking the question what is it the user is actually doing when "closing" the side panel, we come to the conclusion they are basically *aborting the action* they are performing. In other words, they are navigating back to the previous state of the UI. This means, it warrants a link, and not a button:



Close is a link, as it's a UI state change.

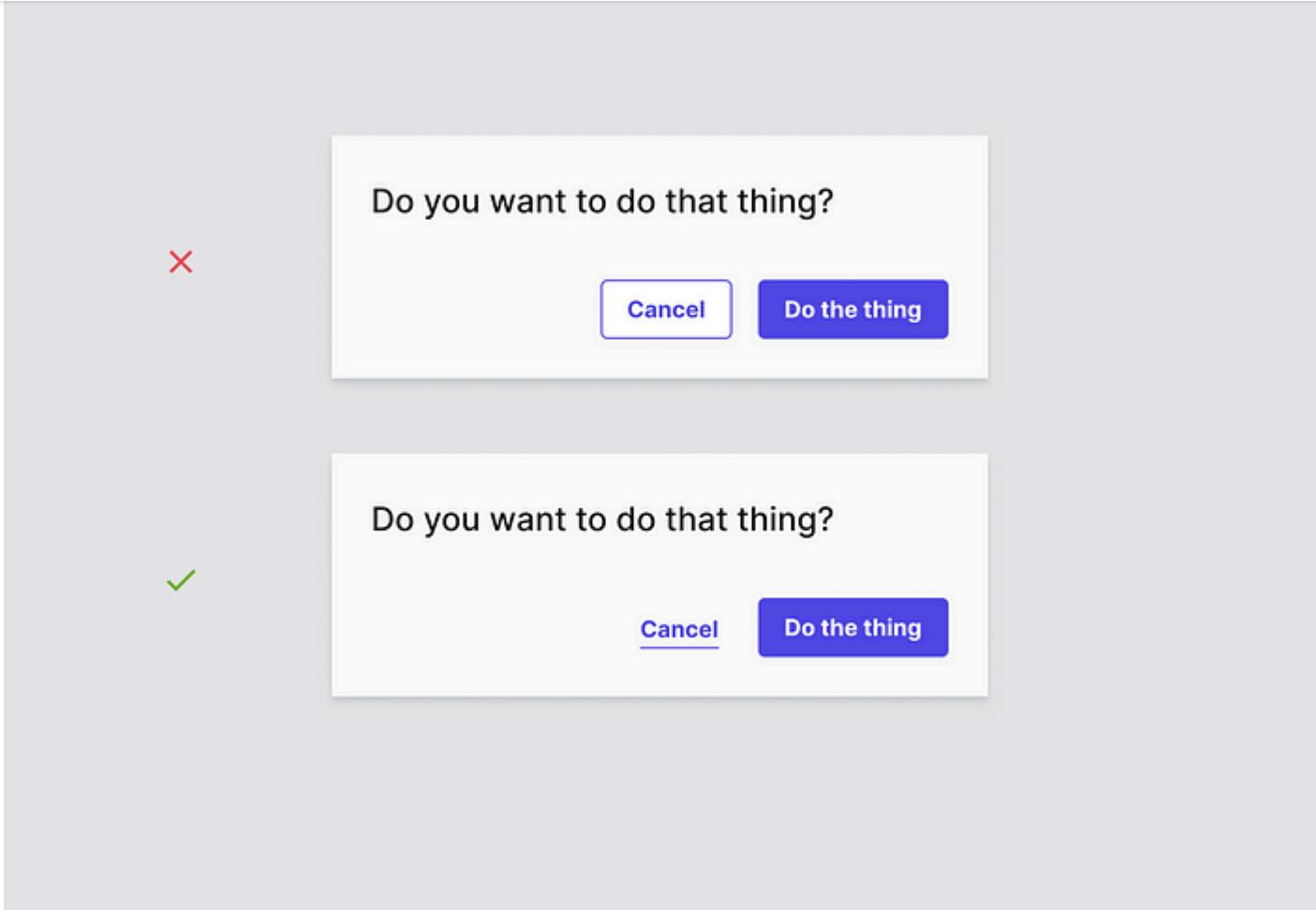
The same can be said about closing any view:



Closing is not an action; it's a UI state change.

Cancel the action

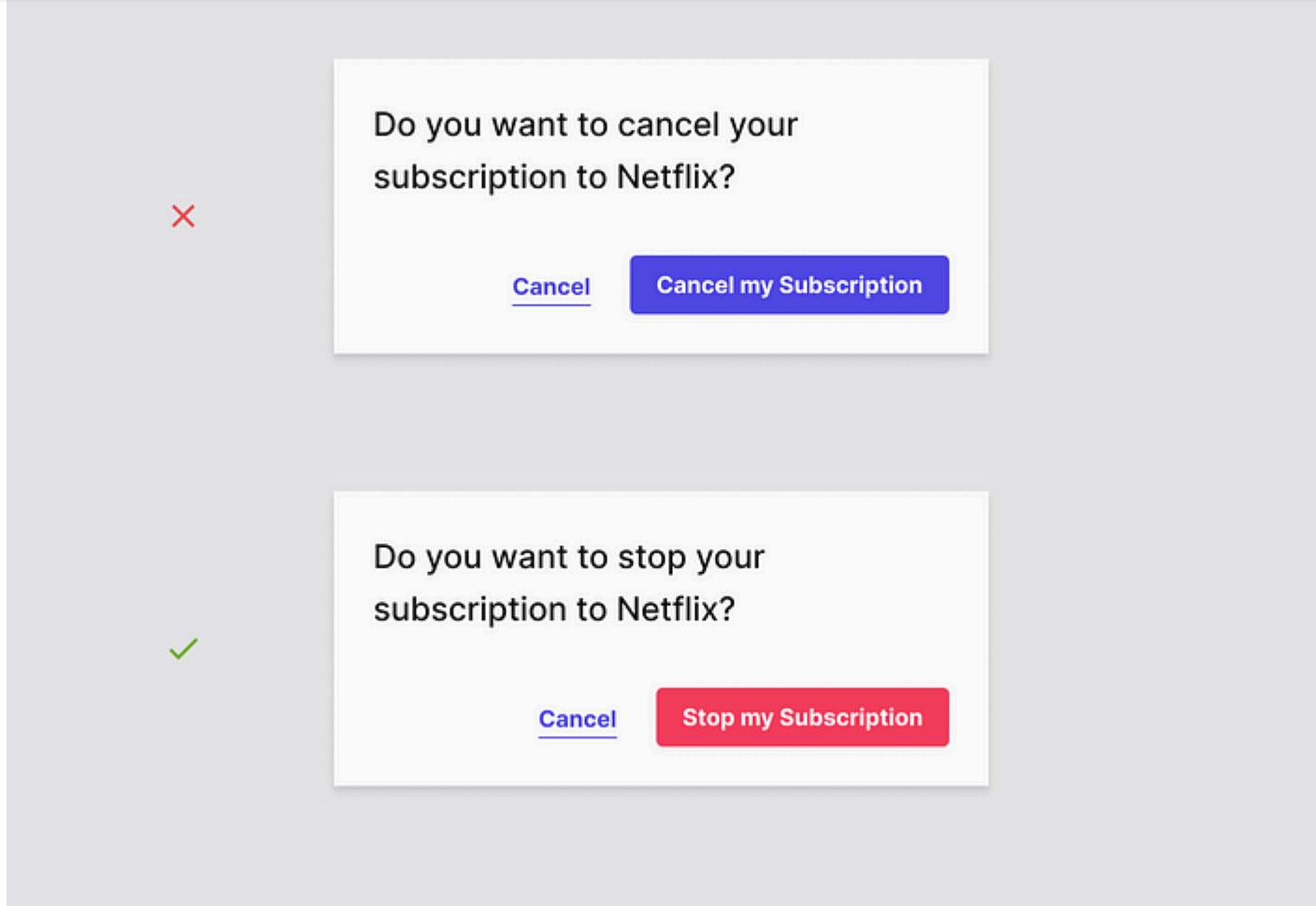
Whether it be inside a side panel, or inside a pop-up (modal view), or maybe even just on a full page form (edit a fruit on the fruit detail page), whenever a user wants to "cancel" the action(s) they are trying to perform, they are basically doing the exact same thing as described in the "Closing the side panel" section; they are aborting the action and thus navigating back to the previous state of the UI. Thus, it requires a link, and not a button.



The answer to this article's question.

"But cancelling is like an action, no?" As you by now might expect, to this, I state a hard "no"! A user cancelling an action is quite the opposite of performing an action; it's the "noping out" of an action. In all the cases described above, cancelling would entail navigating back to the previous state of the UI.

"But what about cancelling my Netflix subscription?" This is an interesting exception, but is simply explained when understanding that in this case, "cancelling" is the primary action. A user can still abort the primary action through a cancel link. Of course, it's probably best not to call it "cancel" and "cancel":



Confusion is confusing.

"**But what about iOS, or Windows or MacOS?**" Sure enough, as addressed earlier, the actual user interface of your operating system basically only uses buttons. For pretty much everything. This completely breaks anything I just wrote. However, you have to realize that a web application is not the same thing as an operating system. It's best to avoid the uncanny valley of UI. Users are confused by mimicking an actual OS, plus one would ignore the power of web UI conventions users are familiar with, which would break usablilty. Let the OS be the OS and the web be the web.

It's also why responsive design is not the same thing as building a native mobile application. They are two kinds of different UI.

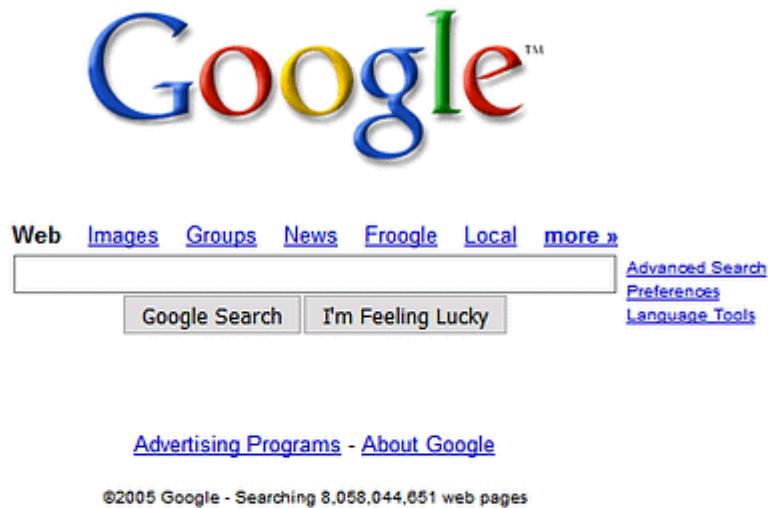
"**What about the main menu? That certainly is navigation?**" Yes, very true. They are indeed navigable items and would only lead to a state change. Therefore, one would expect links instead of buttons. And in reality, the way of visualizing main menu items as a row of buttons has become somewhat of a standard. Users

However, back in the days of ye old interwebs, it was quite common to represent main menu items as links too! (because, you know, they are navigation). And to this day, those items are still links in code, and not buttons, when written in semantically correct HTML.

The screenshot shows the Digg homepage with a dark blue header bar containing the Digg logo, 'News', 'Videos', 'Podcasts', 'Join Digg', 'Login', and a search bar. Below the header is a green navigation bar with 'AllTopics' and various category tabs: Technology, Science, World&Business, Sports, Entertainment, and Gaming. A 'Submit a New Story' button is also in this bar. The main content area is titled 'All News' and lists several news stories with their titles, digg counts, submitter names, submission times, and URLs. Each story has a 'digg it' button, a comment count, and links to 'Blog It', 'Email It', 'Topic', 'Bury', and 'More...'. To the right of the stories is a sidebar titled 'What's Digg?' which explains the user-powered content model. Below the sidebar is a 'Top 10 In All Topics' section listing ten stories with their digg counts and titles. At the bottom of the sidebar is a 'Digg Labs' section with a brief description of the service.

Diggs	Title	Submitted By	Time Ago	URL
80	Last chance to join open source movie project, before phase 3	mattbrighton	made popular 18 minutes ago	(blogs.zdnet.com)
206	Maybe a contract-free iPhone is NOT out of the question	SillyMamma	submitted, made popular 48 minutes ago	(www.tuaw.com)
409	PirateBay proprietor raided by Swedish police!	taylorbulver	submitted, made popular 54 minutes ago	(www.anakata.hack.se)
417	Just add water - students invent alcohol powder	QueenBeeCassi	submitted, made popular 1 hour 8 min ago	(www.reuters.com)
376	How much does a NY cabbage make? How about drug dealers or sex shop owners?	mooji	submitted, made popular 1 hour 9 min ago	(nymag.com)
506	What Would Happen If the Post Office Had Competition?	treemarketer	submitted, made popular 1 hour 48 min ago	(www.lewrockwell.com)
352	How To Fix "24"	1KrazyKorean	submitted, made popular 1 hour 48 min ago	(nomorequo.blogspot.com)

Aah, the old digg. Links everywhere! Even in the main navigation.



Google's main navigation used to be hyperlinks.

Empowering the user by being consistent When we look at the difference between buttons and links, we can easily see that there is a fundamental difference when presented both interactions:

These imply actions directly done on the item in the row:

Fruits (10)

[+ Add Fruit](#)

Type ^	Color	Weight (g)	Origin	Actions
Apple	Red	110	Netherlands	<button>Buy</button>
Apple	Green	115	Belgium	<button>Buy</button>
Apple	Green	95	Netherlands	<button>Buy</button>
Banana	Yellow	190	Colombia	<button>Buy</button>
Banana	Yellow	210	Costa Rica	<button>Buy</button>
Banana	Yellow	185	Colombia	<button>Buy</button>
Lemon	Yellow	110	Netherlands	<button>Buy</button>
Lemon	Yellow	85	Spain	<button>Buy</button>
Lemon	Yellow	75	Spain	<button>Buy</button>
Lemon	Yellow	90	Italy	<button>Buy</button>



Fruits (10)

[+ Add Fruit](#)

Type ^	Color	Weight (g)	Origin	Actions
Apple	Red	110	Netherlands	<button>Buy</button>
Apple	Green	115	Belgium	<button>Bought</button>
Apple	Green	95	Netherlands	<button>Buy</button>
Banana	Yellow	190	Colombia	<button>Buy</button>
Banana	Yellow	210	Costa Rica	<button>Buy</button>
Banana	Yellow	185	Colombia	<button>Buy</button>
Lemon	Yellow	110	Netherlands	<button>Buy</button>
Lemon	Yellow	85	Spain	<button>Buy</button>
Lemon	Yellow	75	Spain	<button>Buy</button>
Lemon	Yellow	90	Italy	<button>Buy</button>

Fruits (10)

[+ Add Fruit](#)

Type ^	Color ^	Weight (g) ^	Origin ^	Actions
Apple	Red	110	Netherlands	Buy →
Apple	Green	115	Belgium	Buy →
Apple	Green	95	Netherlands	Buy →
Banana	Yellow	190	Colombia	Buy →
Banana	Yellow	210	Costa Rica	Buy →
Banana	Yellow	185	Colombia	Buy →
Lemon	Yellow	110	Netherlands	Buy →
Lemon	Yellow	85	Spain	Buy →
Lemon	Yellow	75	Spain	Buy →
Lemon	Yellow	90	Italy	Buy →



Fruits (10)

[Close X](#)

Type ^	Color ^	Weight (g) ^
Apple	Red	110
Apple	Green	115
Apple	Green	95
Banana	Yellow	190
Banana	Yellow	210
Banana	Yellow	185
Lemon	Yellow	110
Lemon	Yellow	85
Lemon	Yellow	75
Lemon	Yellow	90

Buy Fruit

Type

Apple

Weight (g)

115

Origin

Belgium

Shipping Address

My Shop

[Cancel](#)[Buy](#)

Conclusion

The subject of a cancel link or button has been asked previously. I think UX movement's article about this subject almost has it correct: the cancel is indeed not an action, but it should also not be a button. (I also think that a grey button looks more like a disabled button, but that's a different subject). Luke Wroblewski's almost famous blog post comes close to the same conclusion as I do (On top of that it also addresses the position of said UI).

To summarize The cancel option is similar to aborting an action, and therefore similar to navigating back to the previous state of the UI. Navigation is represented in web UI as hyperlinks, and thus **the cancel option should be a hyperlink**. As with everything, this too can (and should!) be debated. I definitely can see the visual appeal to using a button instead of a link. And as with the main navigation convention, things can change over time.

Feel free to ping me if you want. I can be found on Twitter.

Follow-up article

Why “Cancel” should be a link, and not a button

Part 2: your feedback. About SPAs, hyperlinks, dark patterns, and accessibility.

uxdesign.cc