

# The Discoverability Paradox

**Joe Kissell**

[joeontech.net](http://joeontech.net) • [@joekissell](https://twitter.com/joekissell)

Joe  
ON  
Tech

# Here's the plan.

1. Individual complaining.
2. Group complaining.
3. Individual navel-gazing.
4. Group navel-gazing.
5. A practical discussion about what to do.

# **Individual Complaining: Decreasing discoverability**

MacBook Pro  
start looking at this  
REVISÉD & EXPANDED EDITION

# *The* DESIGN *of* EVERYDAY THINGS

DON  
NORMAN



# What is discoverability?

## Don Norman:

- “Is it possible to even figure out what actions are possible and where and how to perform them?”
- “All operations can be discovered by systematic exploration of menus.”

# What is discoverability?

**My take:**

It's easy to figure out...

- What can I do here?
- How can I accomplish x?
- What is currently happening?

# What is discoverability?

Discoverable  $\neq$  Visible (but it helps)

- Progressive disclosure is useful.
- Context-sensitive controls are often just fine.
- The question is, can users find this thing?



# Degrees of discoverability

- Action always visible on screen or menu
- Action visible on screen or menu when applicable
- Action on contextual menu
- Action on hidden contextual menu
- Action easily findable in Help
- You just have to know.



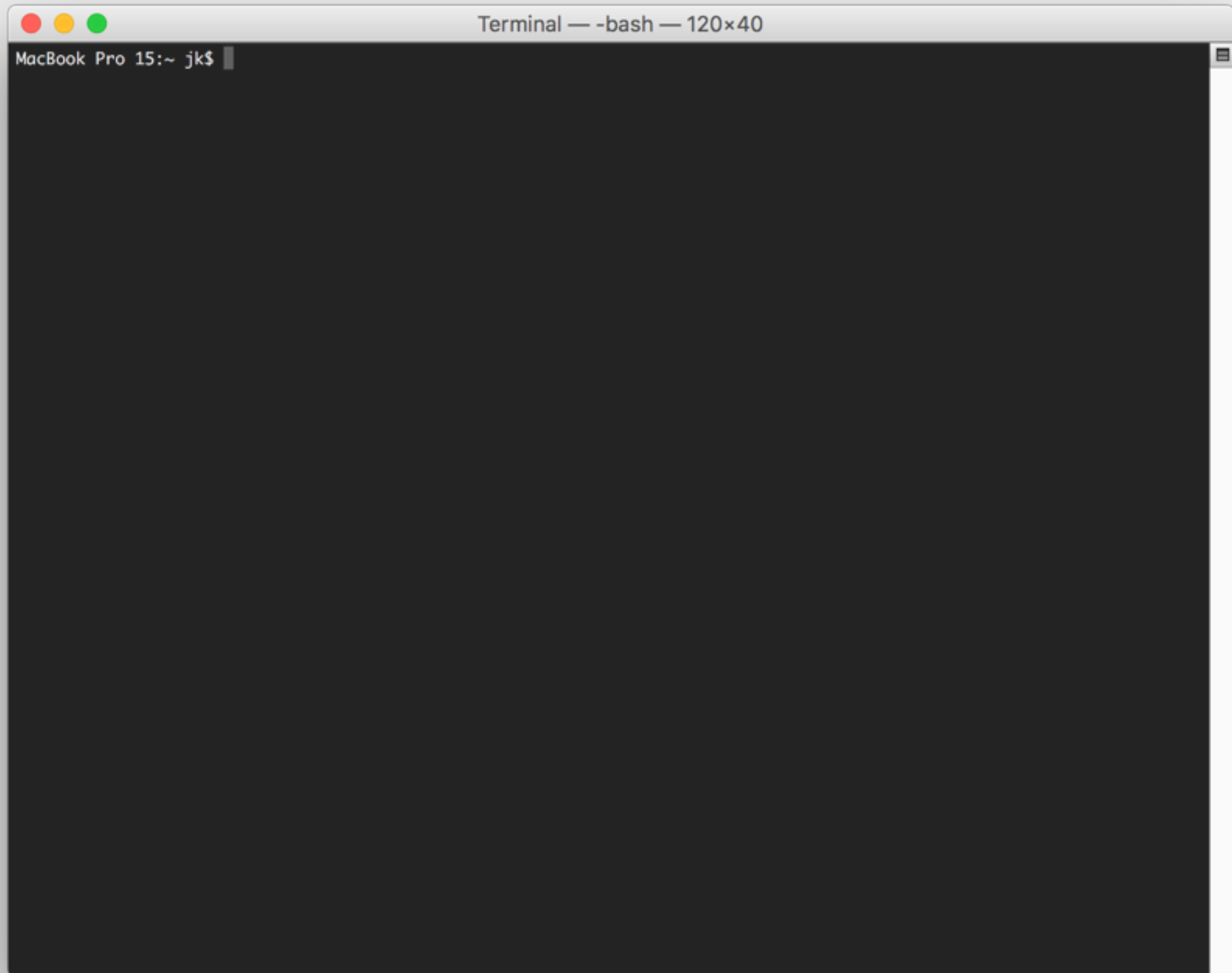
# Why do I care?

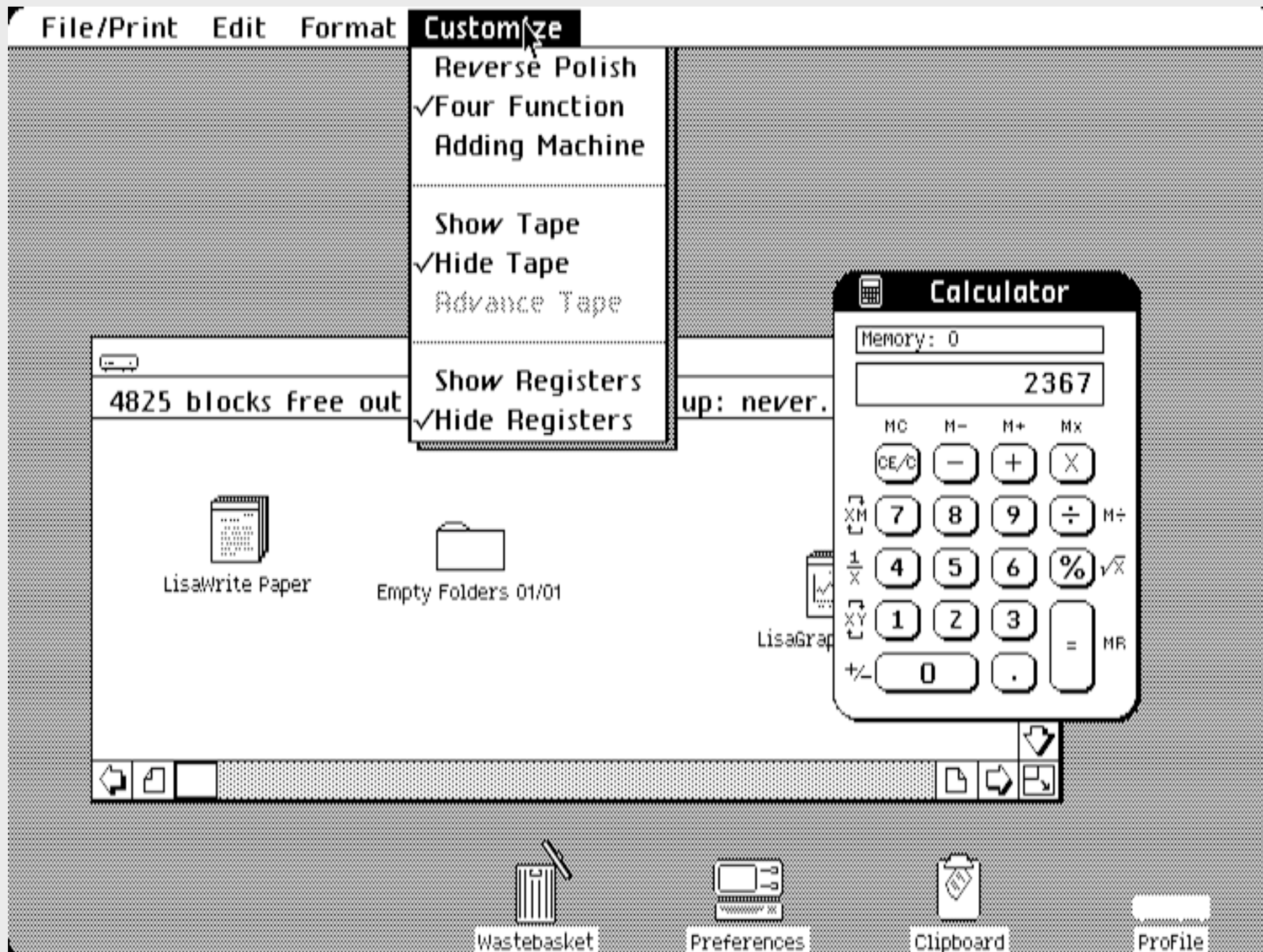
- I have to write about this stuff for consumers, day in and day out.
- My audience is impatient and easily frustrated.
- If I don't address discoverability issues, I hear about it.

# Why should you care?

- **Techs:** Your customers need your help to overcome their confusion.
- **Developers:** You can do the right thing and have happier users who need less tech support.
- **Web designers:** You do want repeat visitors, right?

# Some History





# HIG Statements

People should be able to find all the available features in your app. Don't hide features by failing to make commands available in a menu. Menus present lists of commands so that people can see their choices rather than try to remember command names.

# HIG Statements

Make sure the functionality in your app is easily discoverable. When the usage of your app is obvious, users feel empowered and they can be successful right away. Although you should make help content available in case users need it, users should not feel that they must read a manual before they can begin using your app.



# HIG Contradictions

Always ensure that contextual menu items are also available as menu commands. **A contextual menu** is hidden by default and a user might not know it exists, so it **should never be the only way to access a command.**

# HIG Contradictions

It's a **good idea** to list all important commands in the appropriate app menus so that people can find them easily. It **can be appropriate to omit** infrequently used or power user commands from your app's menus (**instead making them available in a contextual menu**, for example), but be wary of doing this too often. Even experienced users can fail to find commands that are essentially hidden in this way.

# Important moments

- **10.0:** You can hide the Dock.
- **10.7:** Scroll bars hidden by default. Scroll bar arrows are gone altogether.
- **10.7:** ~/Library hidden by default.
- **10.9:** Hover over (or near) stuff to reveal it.
- **10.11:** You can hide the menu bar.

# The iOS effect

By design, features must be maximally discoverable, despite limited screen real estate.

- **Successes:** Most controls in most apps
- **Discoverability weak spots:** The Settings app, most gestures, shake to undo, cut/copy/paste, 3D Touch

# Bad Discoverability

- Unnecessarily hidden UI elements
  - Especially egregious when there's *no other way* to accomplish a task
- *Missing* UI elements
- Undocumented behaviors

# **Demonstrations of Bad Discoverability**

# Group Complaining



# Individual Navel-Gazing

# Virtuous Hiding

- When is it good to limit discoverability?
  - Keeping novice users out of trouble
  - Avoid overwhelming users
  - Progressive disclosure
  - Context sensitivity
- Examples of virtuous hiding

# My Three Theories

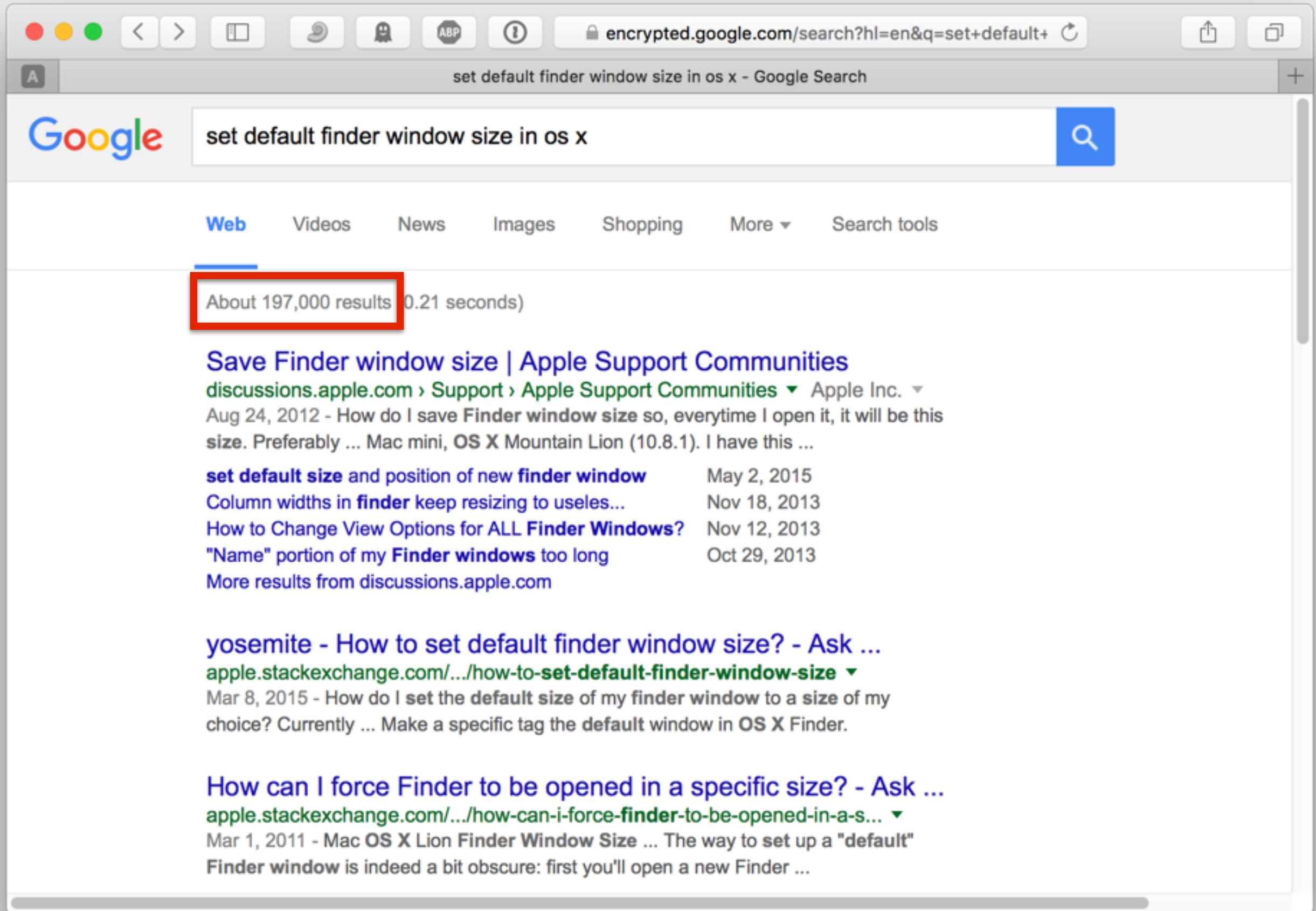
- The veneer of simplicity—  
“Apple products must not seem complicated.”
- The Jony Ive design aesthetic—  
“Avoid the perception of visual clutter.”
- The “treasure hunt” theory—  
“Every product is a magical mystery box.”

 [View interactive transcript](#)



Share this talk and track your influence!

J.J. Abrams traces his love for the unseen mystery -- a passion that's evident in his films and TV shows, including *Lost*, *Star Trek* and the upcoming *Star Wars VII* — back to its magical beginnings.



# The Treasure Hunt

- This isn't necessarily a bad or cynical thing in itself; discovering treasure is delightful!
- But, that's no excuse to frustrate users.

# Group Navel-Gazing



# Practical Discussion

How can we encourage discoverability (in operating systems, apps, and websites) without ruining aesthetics or overburdening developers?