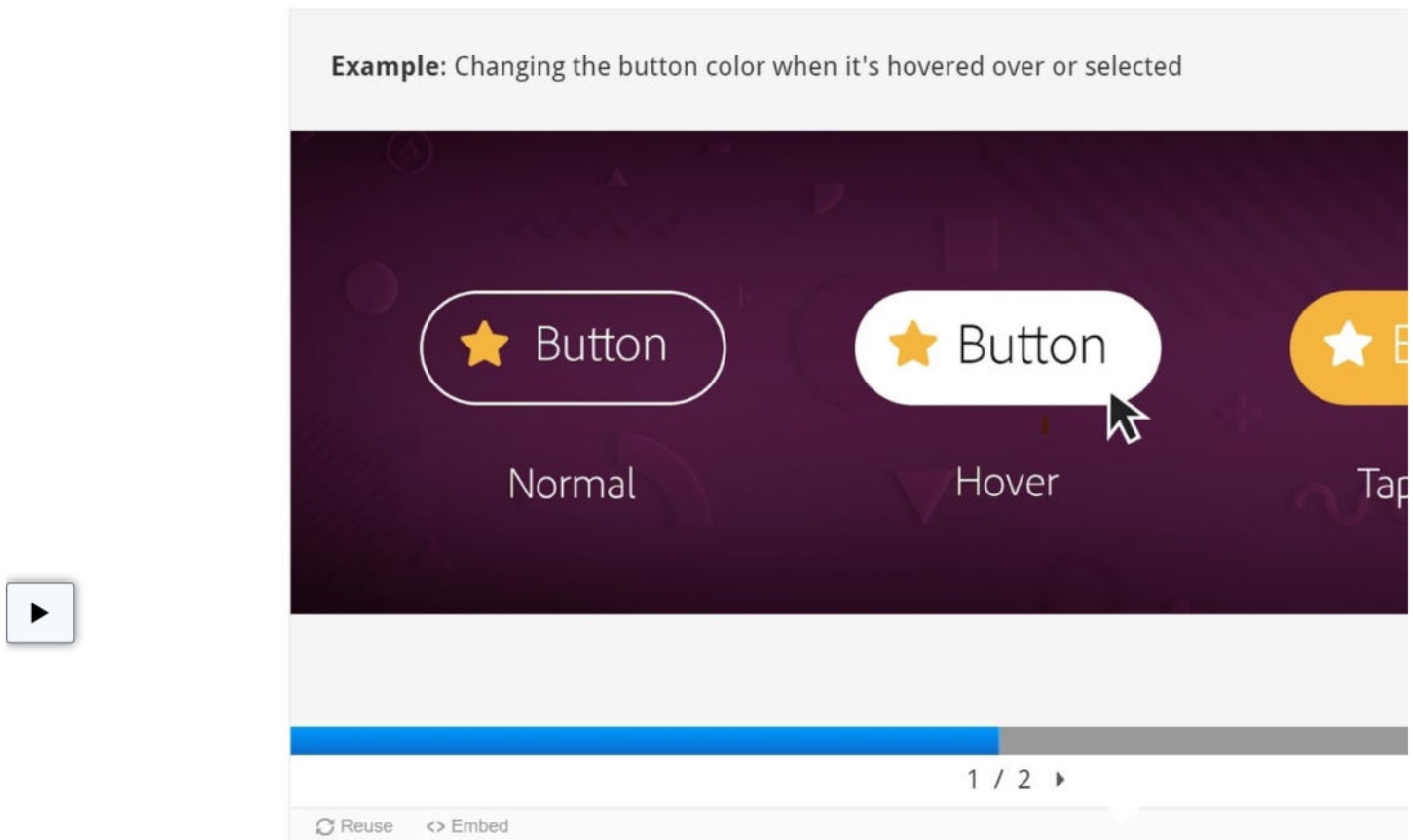


Exploration: Nielsen's Heuristics

How to Complete This Module

This module will teach you about Nielsen's 10 Usability Heuristics. Included with descriptions of this topic are example photos and quizzes. Note that the photos are presented as multiple slides. You can use the arrows or toolbar at the bottom of the photo widget to navigate between photos (shown below).



Additionally, there is an end quiz with multiple-choice questions for you to complete at the end of the module.

Continue reading below to start learning!

Introduction: What are Software Usability Heuristics?

Software usability heuristics are guidelines for making software user interface (UI) designs more usable. Software designers go through heuristics one-by-one when evaluating the usability of a

UI. Software usability heuristics can also guide the creation of a new UI.

An advantage of software usability heuristics is they are an inexpensive way to make a UI more usable: they do not require testing with actual users.

What are Nielsen's Heuristics?

One set of software usability heuristics is Nielsen's Heuristics. Jakob Nielsen, a leader in usability research, published this set of "10 general principles for interaction design" in 1994. Keep reading, below, to learn about each of the heuristics.

Heuristic #1 (of 10): Visibility of system status

Always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time. Allow users to feel in control by providing communication between the system and the user.



Heuristic #2 (of 10): Match between system and real life

Use words, phrases and concepts familiar to the user, rather than system-oriented terms. Do this by following real-world conventions, making information appear in a natural and logical order.



Heuristic #3 (of 10): User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. To help them, support undo and redo and let users handle accidents or changed minds.

► Heuristic #4 (of 10): Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. According to Jakob's Law, people spend most of their time on sites other than yours. Thus, maintaining consistency both within a UI (*internal* consistency) and across UIs in general (*external* consistency) increases learnability of the UI as there are fewer new things for users to learn.

▶ Heuristic #5 (of 10): Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. This can be critical for software where errors may have significant impact on critical parts of people's lives (e.g., medical, financial, etc.).

► heuristic #6 (of 10): Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. Users should not have to remember information from one part of the UI to another. Provide users instructions for using the system that are visible or easily retrievable.

Example of using recognition: Is the [REDACTED]

Example of using Recall: What is the [REDACTED]

► Heuristic #7 (of 10): Flexibility and efficiency of use

Accelerators, unseen/ignorable by the novice user, may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

► Heuristic #8 (of 10): Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. "Communicate, don't decorate."

► Heuristic #9 (of 10): Help users recognize, diagnose, and recover from errors

Express error messages in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. You can even provide an immediate course of action for solving the problem when the error is shown.

Example: "Error 404" vs. "The page you're looking for wasn't found"

Example: "Page not found, click 'here' to return to the previous page"

▶ Heuristic #10 (of 10): Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. To do this, give straightforward, scannable steps for the user and use screenshots when possible.

rap Up

Nielsen's Heuristics are a set of 10 software usability heuristics for evaluating and improving general UI usability.

Quiz

Which is a characteristic of good help and documentation?

- ☐ Provides concrete steps
- ☐ 2000 pages long
- ☐ Mixes instructions for many tasks

 Check



 Reuse  Embed

H-P

References

Jakob Nielsen. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.



Return to Extra Credit

