Design Principles And Usability Heuristics

Design Rules

- One of the central problems that must be solved in a user-centered design process is how to provide designers with the ability to determine the usability consequences of their design decisions.
- design rules are required so that a designer can follow in order to increase the usability of the eventual software product
- Classification of design rules:
  - *Rules Authority: Indication of whether or not the rule must be followed in design or whether it is only suggested.*
  - *Rules Generality: Whether the rule can be applied to many design situations or whether it is focussed on a more limited application situation*

Definition of Design rules

Design rules (or usability rules) are rules that a designer can follow in order to increase the usability of the system/product.

They can be in the form of:

- Principles: abstract design rules with high generality and low authority
- Standards: specific design rules, high in authority and limited in application (not generalizable)
- Guidelines : lower in authority and more general in application

Design Rules

- Design rules for interactive systems can be supported by psychological, cognitive, ergonomic, sociological, economic or computational theory, which may or may not have roots in empirical evidence.
- The designer might not always have the relevant background to these domains, hence multidisciplinary teams coming in handy
- The design rules are used to apply the theory in practice. Often a set of design rules will be in conflict with each other, meaning that strict adherence to all of them is impossible.
- The theory underlying the separate design rules can help the designer understand the trade-off for the design that would result in following or disregarding some of the rules.

- Principles are derived from knowledge of the psychological, computational and sociological aspects of the problem domains and are largely independent of the technology; they depend to a much greater extent on a deeper understanding of the human element in the interaction. They can therefore be applied widely but are not so useful for specific design advice.

- Guidelines are less abstract and often more technology oriented, but as they are also general, it is important for a designer to know what theoretical evidence there is to support them.

- A designer will have less of a need to know the underlying theory for applying a standard. However, since standards carry a much higher level of authority, it is more important that the theory underlying them be correct or sound.

Heuristics and 'Golden Rules'

- Well known advocates of user centered design like Schneiderman, Jakob Nielsen and Don Norman have presented sets of heuristics and golden rules.

- While these are inevitably ‘broad-brush’ design rules, which may not be always be applicable to every situation, they do provide a useful checklist or summary of the essence of design advice.

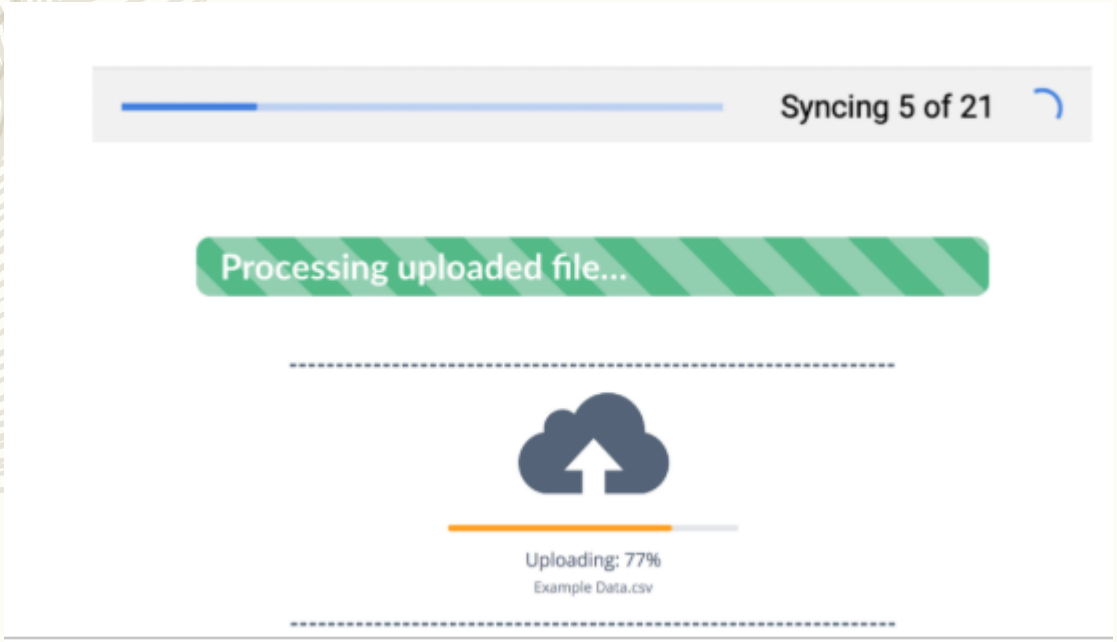
- It is clear that any designer following even these simple rules will produce a better system than one who ignores them.

Usability Heuristics

Jakob Nielsen's 10 Usability Heuristics

– **Visibility of system status**: the system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

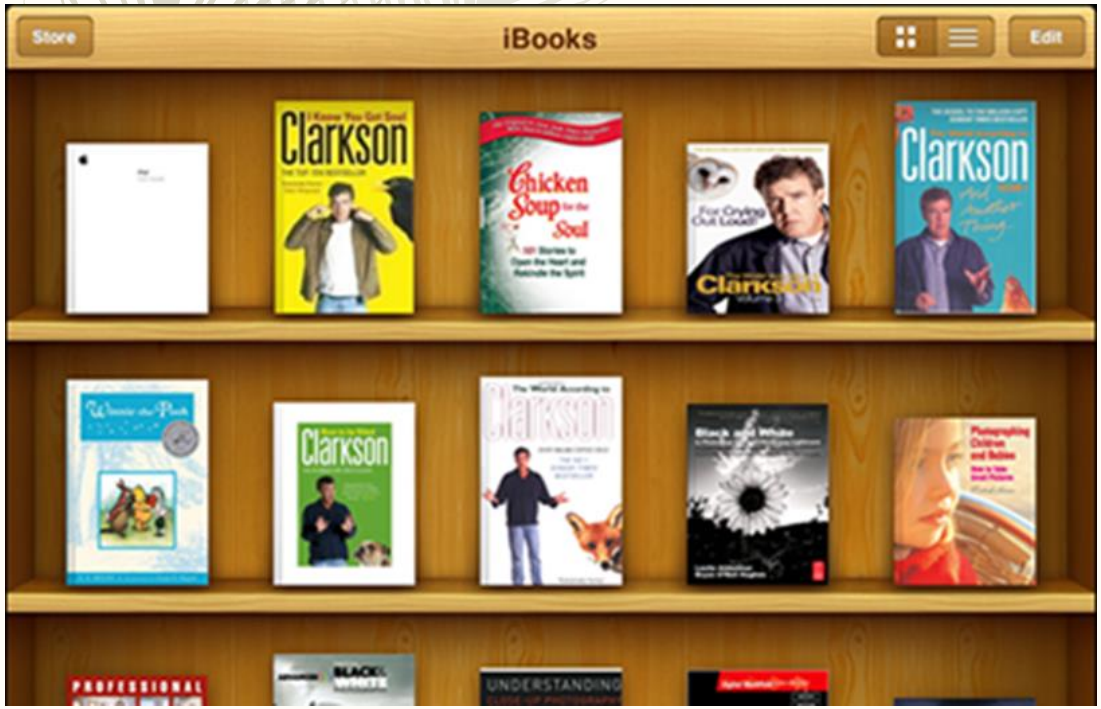
Example: Whenever we try to upload our files into google drive or any other cloud system, we see it's telling us how much the uploading is done or the remaining time by showing percentage (%), number file or a bar. Look at the images below:



Jakob Nielsen's Usability Heuristics for User Interface Design

– **Match between system and the real world**: the system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

iBooks iPad application using the metaphor of wooden book shelf and Using real life metaphor in computer application



Jakob Nielsen's 10 Usability Heuristics

– **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. Support undo and redo.

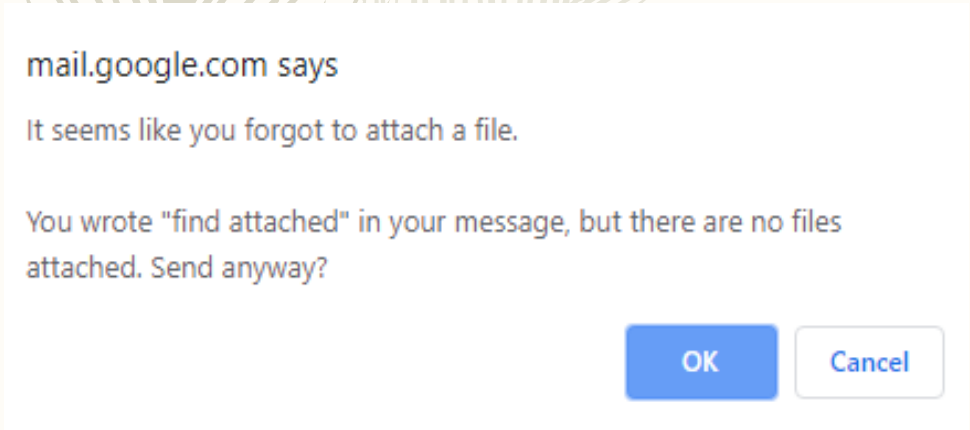
Mozilla suggesting some security tips and handling exception and History in Photoshop helps user in recovering previous steps

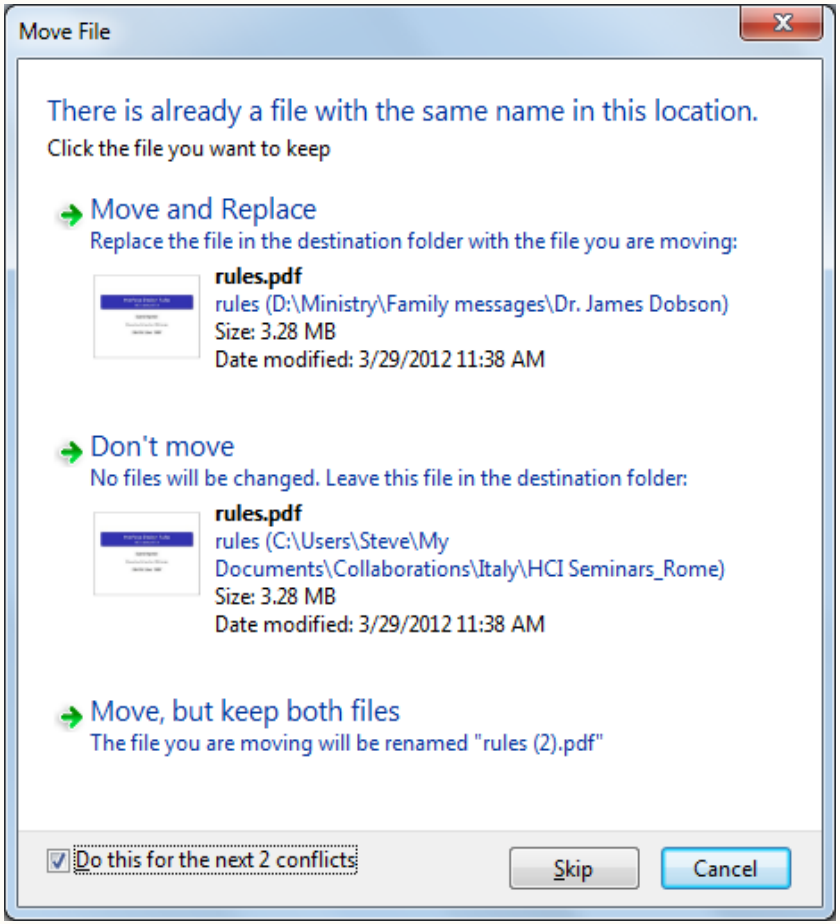
Jakob Nielsen's Usability Heuristics for User Interface Design

– **Consistency and standards**: users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Jakob Nielsen's 10 Usability Heuristics

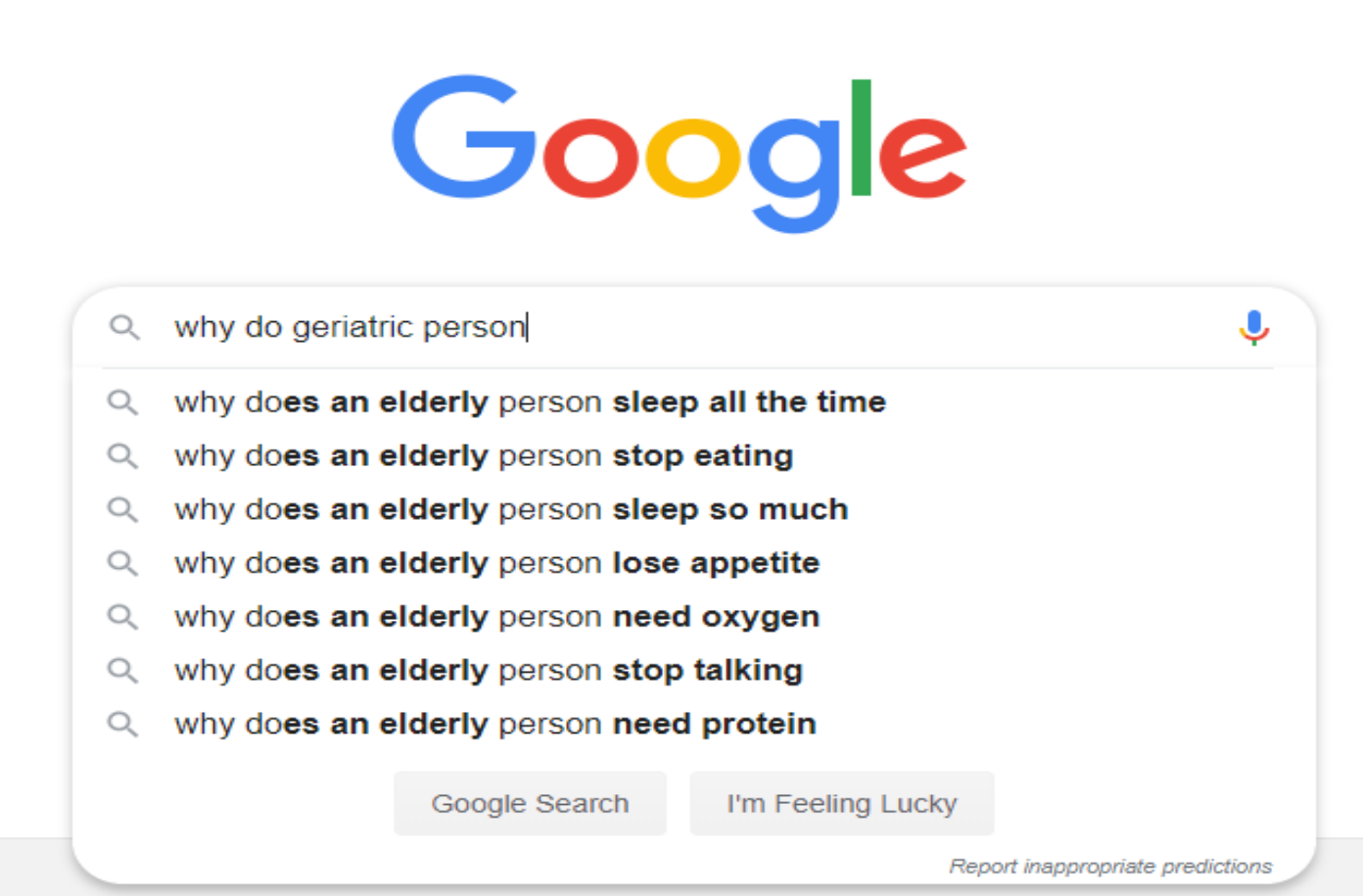
– **Error prevention**: even better than good error messages is a careful design which prevents a problem from occurring in the first place.





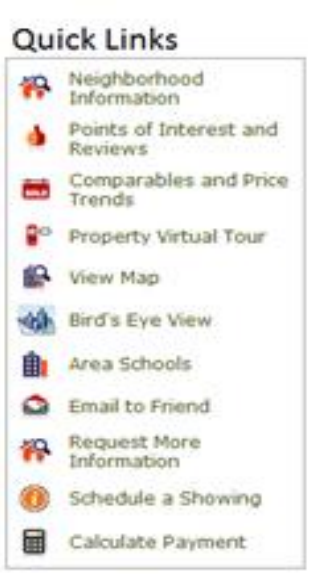
Jakob Nielsen's 10 Usability Heuristics

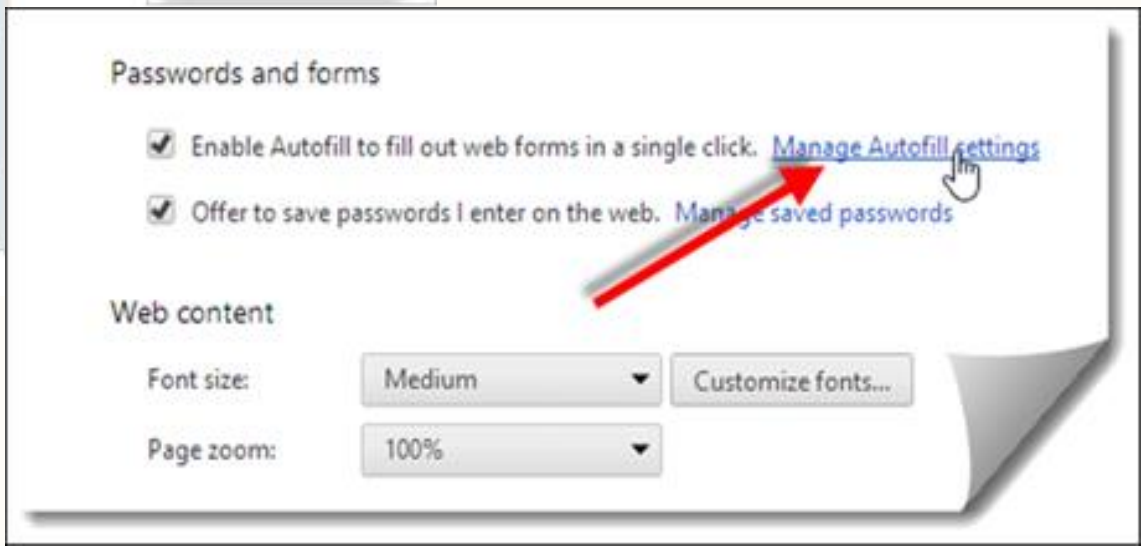
– **Recognition rather than recall**: make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.



Jakob Nielsen's 10 Usability Heuristics

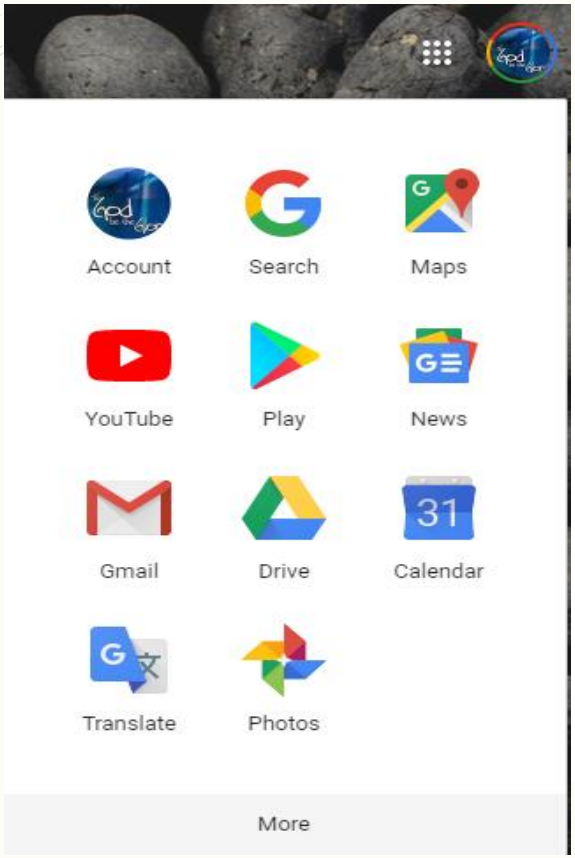
- **Flexibility and efficiency of use**: accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.





Jakob Nielsen's 10 Usability Heuristics

- **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.

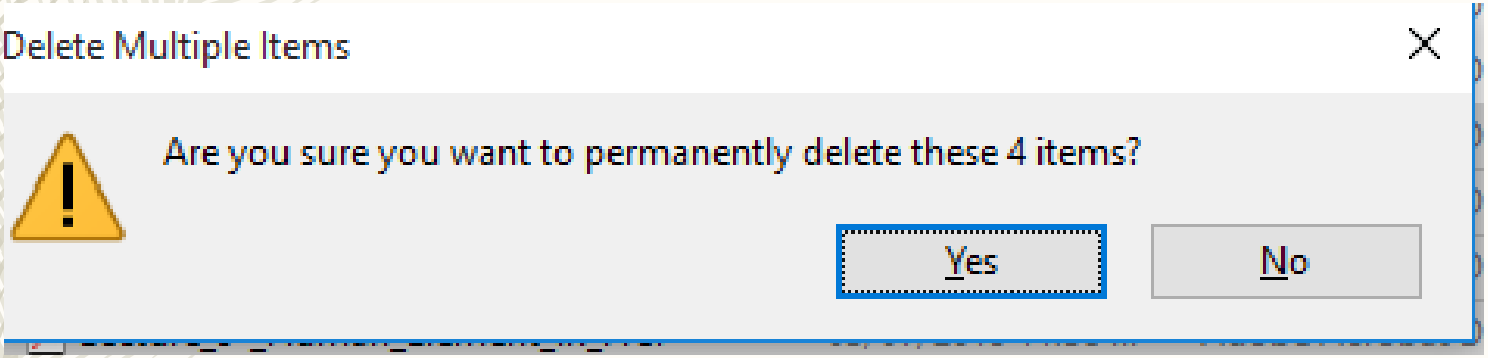


The icons belong to one family.

The design of icons is minimalistic with simple and creative graphics.

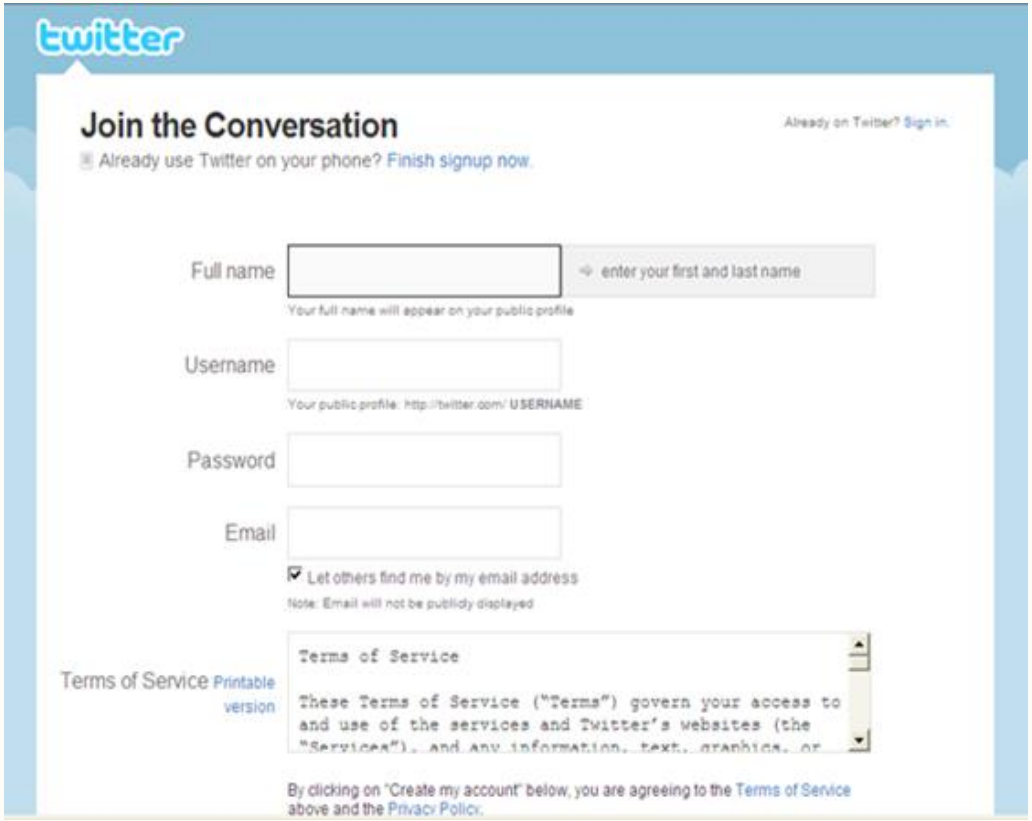
Jakob Nielsen's 10 Usability Heuristics

- **Help users recognize, diagnose, and recover from errors**: error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.



Jakob Nielsen's 10 Usability Heuristics

- **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.
- Contextual help is the best way to help! Also, telling users about the consequences of their actions is also very helpful



Ben Shneirdermann's 8 Golden Rules

- **Strive for consistency**: layout, terminology, command usage, etc.
- **Cater for universal usability**: recognize the requirements of diverse users and technology. For instance add features for novices eg explanations, support expert users e.g. shortcuts.
- **Offer informative feedback**: for every user action, offer relevant feedback and information, keep the user appropriately informed, human-computer interaction.
- **Design dialogs to yield closure**: help the user know when they have completed a task.
- **Offer error prevention and simple error handling**: prevention and (clear and informative guidance to) recovery; error management.
- **Permit easy reversal of actions**: to relieve anxiety and encourage exploration, because the user knows s/he can always go back to previous states.
- **Support internal locus of control**: make the user feel that s/he is in control of the system, which responds to his/her instructions/commands.
- **Reduce short-term memory load**: make menus and UI elements/items visible, easily available/retrievable, ...

Donald Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in the head*.

- People work better when the knowledge they need to do a task is available externally – either explicitly or through the constraints imposed by the environment.
- Experts also need to be able to internalize regular tasks to increase their efficiency.
- So systems should provide the necessary knowledge within the environment and their operation should be transparent to support the user in building an appropriate mental model of what is going on.

2. Simplify the structure of tasks.

- Tasks need to be simple in order to avoid complex problem solving and excessive memory load.
- There are a number of ways to simplify the structure of tasks.
  - One is to provide mental aids to help the user keep track of stages in a more complex task.
  - Another is to use technology to provide the user with more information about the task and better feedback.
  - A third approach is to automate the task or part of it, as long as this does not detract from the user's experience.
  - The final approach to simplification is to change the nature of the task so that it becomes something more simple.
- In all of this, it is important not to take control away from the user.

3. Make things visible

- ❑ Bridge the gulfs of execution and evaluation.

- ❑ The interface should make clear what the system can do and how this is achieved, and should enable the user to see clearly the effect of their actions on the system.

4. Get the mappings right.

- ❑ User intentions should map clearly onto system controls and systems.
- ❑ So it should be clear what does what and by how much.

  Controls, sliders and dials should reflect the task –so a small movement has a small effect and a large movement a large effect.

5. Exploit the power of constraints, both natural and artificial.

- ❑ Constraints are things in the world that make it impossible to do anything but the correct action in the correct way.

  A simple example is a jigsaw puzzle, where the pieces only fit together in one way.

6. *Design for error*.

- To err is human, so anticipate the errors the user could make and design recovery into the system.

7. *When all else fails, standardize*.

- If there are no natural mappings then arbitrary mappings should be standardized so that users only have to learn them once.

  It is this standardization principle that enables drivers to get into a new car and drive it with very little difficulty – key controls are standardized.

  Occasionally one might switch on the indicator lights instead of the windscreen wipers, but the critical controls (accelerator, brake, clutch, steering) are always the same.