

Software Documentation → Is a form of writing for both print and online media that supports the efficient and effective use of software in its intended environment.

The forms of software documentation:

- ⇒ Writing to Teach (Tutorial).
- ⇒ Writing to guide (Procedures).
- ⇒ Writing to Support (References).

The process of software documentation:

- ⇒ Analyzing your users
- ⇒ Planning and writing your documentation
- ⇒ Getting useful reviews
- ⇒ Conducting usability tests
- ⇒ Editing and Fine tuning

The tools of software documentation:

- ⇒ Using Graphics
- ⇒ Getting the language right
- ⇒ Layout pages and screen
- ⇒ Designing indexes

A manual should help users **solve problems** in their workplace

Cross references in manual and **hypertext links** in online system can help maintain the user's sense of control over the documentation .

Semantic Orientation in page design means you arrange the elements of the page meaningfully according to elements of **the job the user needs to perform**.

Routine Task → Repeatable tasks that are easily represented by conventional procedures .

Complex task → Require the user to apply knowledge that is not easily codified in step by step procedure, it comes from years of experience.

Task oriented manuals encourages users to identify and get **helps from others**. Some software has some features that encourage group or team work.

User communities can help provide candidates for this kind of user-involved document development.

The **good manual** or help system has many features that make it succeed.

Task Orientation → A design strategy for software documentation that attempts to increase user knowledge of and application of a program by **integrating the software with the user's work environment**.

User → Is a person who operates a computer.

The Default User → a person who needs to learn about menu functions and commands.

The Task-Oriented User → A person who uses software for workplace ends.

Novic Users :

- ⇒ Require tutorial documents,
- ⇒ teach basic functions

Intermediate Users:

- ⇒ Require procedural documentation.
- ⇒ The intension of which is to help them during actual use of the program in their workplace.

Advanced Users:

- ⇒ Require **Reference** documentation .
- ⇒ The intension of which is to further their understanding about how the program operates

A computer **tutorial** → an interactive software program created as a learning tool.

Tutorials help people learn new skills by using a step-by-step process that ensures the user is following along

From the task list **guidelines** to select your tasks:

- ⇒ **Central** to job performance
- ⇒ **Essential** for efficient software use
- ⇒ **Performed** frequently

the Right Type of Tutorial:⇒ **The Guided Tour**

- It presents an **Overview** of the program features to a user unfamiliar with them.
- It focuses on the entire program capabilities and user actions like main screens and useful commands.
- **TOURISM** → short journey around a building or place with a person who tells you about what you.
- **COMPUTING** → an explanation of how to use a website or a piece of software that you read on a computer screen by clicking on a series of buttons or links

⇒ **Demonstration**

- → design a demonstration when you want to illustrate some specific parts of a program , perhaps for a specific user
- passively (**negatively**)
- no interaction

⇒ **The Quick Start**

- → it is for **experienced to advanced** users with domain knowledge who want to get going with a program.
- It involves → **interaction** with the program itself, and **rarely** uses examples.
- It consists of one page or folded cards that explain how to start the program and list of commands.

⇒ The Guided Exploration

- contain instructions for the user to “try out” commands which encourage exploration of the program

⇒ The Instruction Manual

- A manual usually accompanying a technical device and explaining how to install or operate it.

⇒ Offer Highly Specific Instructions

- instruction or lessons should focus on a specific scenario or problem that the user would recognize.

⇒ Pace the tutorial → 10-12 minutes

⇒ Try Your Tutorial:

- If you do not have a real user try to mock-up the situation with someone of similar background as the user.

⇒ the basic elements of tutorials:

- Intention to Teach
- Selectivity in Choosing Material
- Stand Alone Design

⇒ Two design approaches need to be examined:

- The Elaborative Approach, It includes elements of a good storytelling, describe scenario carefully and slowly. → using this approach with novice users.
- The Minimalist Approach:
 - Users jump the gun
 - Users will skip info
 - Users like to lead

⇒ the principles of this approach

- Focus on real task and activities
- Slash the verbiage
- Encourage Exploration
- Support Error Recognition and Recovery

Guidance Information → also known as step-by-step instruction or procedures, makes up the heart of all task oriented documentation system.

Procedures consist of how-to-do-it explanations, but also require how-it-works and why-it-works overviews.

Information you need to provide in an effective procedures:

⇒ Introduction

⇒ Screen shows the user result of actions.

⇒ Tips help the user find ways to use the program efficiently.

⇒ Tables help the user decide what options to exercise for this step.

⇒ Elaboration.

If a procedure becomes extremely long, see whether you can logically break it up into two shorter procedures.

a procedure is a step by step series of commands for accomplishing a meaningful operation with a software program.

A **rich** procedure needs more visuals, more explanations, more options, describe more results.

A **sparse** procedure on the other hand require only the repeating of the steps in the task description.

a **rich** procedure will **contain** the following explanations:

- ⇒ **What happens**
- ⇒ **Suggested response**
- ⇒ **Screens**
- ⇒ **Cautions and warnings.**
- ⇒ **Tips**
- ⇒ **Tables**
- ⇒ **References**

The Appropriate Instructional Format:

- ⇒ **Standard Format** → contains steps, notes, screen shots, and other elements left justified in one or two columns in a sequential order from first to last
 - Advantages
 - of this format are its recognizability, its ease of flow from one page to another, the ability to easily re-number tasks, and the easy to see steps.
- ⇒ **Prose Format** → Puts steps in sentences and paragraph forms making it look and feel more conversational over Standard format's command approach.
 - Advantages
 - of prose are the relaxed tone, saves space, clarifies simple, basic steps, accommodates experienced users.
- ⇒ **Parallel Format** → shows a screen with the fields empty and parallels the field names in the steps that follow.
 - Advantages
 - of parallel format are the organizational benefits, great for complicated screens and dialog boxes.
- ⇒ **Embedded Help** → is the name for "interactive assistance" found in most programs today.

The basic idea of rhythm of expositions lies in the action/response pattern.

Evaluative test → that after you finish the procedure, you have an **actual user**, or **prototype of the user**, or **yourself as a last resort** to perform the steps.

All procedural documentation answers the user's simple question " how do you use the program?

Procedures and tutorials differ greatly.

A procedure **expands** the user's focus, a tutorial **contracts** it.

In the **support level (reference)** the user **define the task and goes** to the documentation to get an essential info needed to perform the task.

What Kind of Info User Need Guidance → **Installation**, **maintaining** and repairing systems.

Scenario → a small story or narrative in a setting, it tells or reminds the user what the task will allow him or her to accomplish in a working setting.

Steps → steps tell the user what to do, it tells the tools to use and the action to take with the tool.

Tables → follow these guideline when using tables:

- ⇒ keep table simple.
- ⇒ cite the table in the text.
- ⇒ use descriptive.
- ⇒ use visual cues

Screens:

- ⇒ Show the partial result of a procedure.
- ⇒ Show the final result of the procedure.
- ⇒ Show dialog box
- ⇒ Show toolbars
- ⇒ Show menus

Reference documentation, or support documentation, are the look-up and help parts of a manual. They should be organized in a task-oriented manner, not just alphabetically.

Elements of a Reference Structure:

- ⇒ Function Entry, tells what the function does.
- ⇒ Declaration, shows how to use the function.
- ⇒ Remarks.
- ⇒ Edge Bleed.
- ⇒ See Also, helps the user see interrelationship among entries.
- ⇒ Examples.
- ⇒ Tips.

The Right Form of Reference

- ⇒ Usual forms of reference, quick reference, along with the procedures in a user's guide.
- ⇒ Special forms or reference:
 - Appendices

It contains info that relevant and useful, but not essential to all users. If you examine an appendix in software manual you will find:

- ⇒ Error messages.
- ⇒ Filenames and extensions.
- ⇒ Troubleshooting tips.
- ⇒ ASCII charts showing.

Readme Files → text documents containing important initial information, including installation details or tips.

Innovative Forms → documentation that are presented in special formats, such as foldouts, posters, and flip-cards.

Keyboard Templates and Short Forms (job aids) → it consist of very brief reminder that attach to key-board. Usually limited to defining keys, they can stick to the keyboard or overlap the keys.

Commands → the instructions used to work with a program.

Glossary → terms used in the manual.

What to Include in A Single Reference Entry?

- ⇒ **Conceptual Information** explains the term and its function.
- ⇒ **Structural Information** explains how the term relates to other terms.
- ⇒ **Technical Information** describes the programming information related to the command.

Pattern → **Topics** included in patterns of reference entries include

- ⇒ **Definition.**
- ⇒ **Explanation.**
- ⇒ **Example.**
- ⇒ **Step-by-Step.**
- ⇒ **Warning/Cautions.**

Organize the Reference Section → **Alphabetical Organization**

Drawback → alphabetical order it does little to support the task orientation of your manual.

Menu-by-Menu → you set up your reference section by menu, according to how the user sees them in the program.

- ⇒ advantage:
 - task orientation

Context-Sensitive → you can organize your help section according to the context within which the user ask for help.

The analysis of users consists of inquiry into the following:

- ⇒ Work motivations.
- ⇒ Computer Experience.
- ⇒ user community.
- ⇒ learning preference.
- ⇒ Usage pattern

Anticipate transfer of learning → It is also important to build up a repository of knowledge about users, which includes all of the small facts, attitudes, artifacts, interactions, and values that guide them in their job duties.

resources for learning about the job duties → including occupational guides, industry-specific guides, placement services, or company job descriptions.

How to conduct an interview

How to observe

Questionnaires are also valuable in that they allow you to gather information from a variety of users, increase the chance of identifying unique concerns, and identify wide patterns of use.

Involve users in all phases of the project:

- ⇒ many benefits
 - Increased accuracy
 - More appropriate information
 - Increased usability
 - Improved relationships

Identify Document goals → The clearer your objectives, the better the chance that you will achieve them.

Computer experience → Novice, Experienced, and Expert.

Usually novice users has a negative attitude toward new software, because they do not see the value of using program to accomplish their tasks.

People relied on their own colleagues and IS staff more than the manuals. (user community).

User group: a group of people who used a certain program or sub program.

Learning preference:

- ⇒ Instructor.
- ⇒ Manuals.
- ⇒ Computer.

Usage Pattern:

- ⇒ Regular, daily, incremental learning.
- ⇒ Intermittent, frequently and voluntarily, learned and forgotten, requires online help.
- ⇒ Casual, little or no formal training, library searches, searches for information in encyclopedia.

But often projects are created in teams. Both development teams and writing teams work to develop software documentation.

The development (“cross-functional”) team develops the entire project and usually includes professionals with varied backgrounds and skills.

Task sheet → It gives you the ability to tick off your accomplishments as you work.

The design plan → content specifications

alpha draft → The first complete document, include all material such as text, graphics, indexes, and other associated materials.

Write a final draft → the end result should be a camera ready document or online help that is ready for distribution with the program.

Conduct a field evaluation → The field evaluation, done by users and operators of the program, allows you to judge how well your product fits the needs of the intended user.

There are two main kinds of projects

- ⇒ A stand alone project
- ⇒ A development project

Functions of the documentation plan

- ⇒ Managerial
- ⇒ Persuasive

The documentation Process → The goal of the process is to tailor the manual and the online help to the user's need with great precision.

In software development there are three main methodologies in place:

- ⇒ The waterfall method.
- ⇒ The rapid development method -prototyping.
- ⇒ The object modeling method.

Documentation Plan → a written document describing a manual or help system for a software program containing specifications for the document and a plan for creating it.

Strategies to follow

There are Two parts of a Documentation Plan:

- ⇒ Project Plan → how you will produce your manual, the schedule, resources, time estimates.
- ⇒ Design Plan → what your manual will contain, and what they will look like.

Online help Development Process:

- ⇒ Identify & list topics
- ⇒ Determine the interconnected elements
- ⇒ Decide what software capability to use
- ⇒ Select a development method to construct the system

Benefits of Online help for writer:

- ⇒ Save paper.
- ⇒ Update easily.

Drawbacks of Online help for writer:

- ⇒ Take up disk space.
- ⇒ Require reformatting of print.

Work in the Drop-Dead mode → This means if you dropped dead one day, somebody else could walk in and take over the project.

the type of the review :

- ⇒ User reviews: Reviews by the actual intended users of the document.
- ⇒ Management reviews: Reviews by managers and supervisors associated with your documentation project.
- ⇒ Technical reviews: Reviews by programmers and developers of the software.
- ⇒ Client reviews
- ⇒ Subject-matter expert reviews

Establish reviews schedule:

- ⇒ Sequential Circulation → involves making only one copy of the document and then passes it to the next person to review it.
- ⇒ Simultaneous Circulation → involves making a copy for all of the reviewers and receiving every ones review back.

The purpose of reviews:

- ⇒ Communication function → help communicate with people associated with your project.
- ⇒ Management function → help you manage your project.
- ⇒ Quality assurance function → help you maintain the quality of your product.

The earlier the stage of the process review the more explanation we will have to provide about the background of the project.

The technical reviews will focus on the accuracy and conformance with the company policies.

مادة الفاينال

Decide when to test → can test any time during the nine stages of the document development

Design phase → predictive test to test the suitability of design specs and production goals, high degree of flexibility.

Writing and draft phase → remedial test immediate change and re-test to the document, moderate degree of flexibility.

Field evaluation → evaluative test changes have to wait for next release.

Tie testing to document Goals → may be you do not have a time to test all your documents, concentrate on how to apply the program to the user's workplace

A test point → an issue or feature of a document that might interfere with the efficient and effective application of a program to a user's work activities.

Test tasks → with high chance of user failure or high cost.

Test complex task → one of a kind, high abstract or technical task.

choose type of test :

1 → **Performance** Tests test whether users can → **successfully complete** a given procedure.

2 → **Understandability** Tests test whether users can → **provide evidence** of what they have learned.

3 → **Read-and-Locate** Tests (can they find it test) test how **effectively** users can **locate** a given topic of information in a documentation set

Objectivity → that you try to set up the test in a way that you don't **prejudice the outcomes too much**

The **tester** → the **person who administers** the test, arranges the meeting with users, sets up the test situation, records the test activities, and so on.

The **evaluator** → **the person taking the usability test**.

Do a **pilot testing**, **Test the test** → is a way of **reviewing your test** to see if your testing material with gather the kind of info you want them to do.

Instructions → do the guidelines you give to **evaluators** are enough to test correctly?

Terminology → are the terms **easy to understand**?

Timing → can the user perform the test in the **allocated time**?

Testing → resembles **reviewing**

Review → produce **comments**

The substantive edits (development edit):

→ **language edit** → **clarifying** the expression of ideas in a document

→ **Information edit** → ensuring that all **elements** of a report work together and in the right order

The copy edits:

→ **Format edit** → all the parts conform with **acceptable format**

→ **Mechanical style edit** → all **abbreviations**, **capitalization** are consistence with a specified in house style

Proofreading → last **stage of edit**, making sure all the parts match each other

managerial editing → should attend **meetings**, edit documents

substantive editing → should **check document** as they are being developed and advises the writers on how to organize the contents of the document.

Copiediting → done after **the document complete**.

Proofreading → to double **check things**.

flip test → **ten second per manual**, layout of pages, overall look.

Skimming → **six to ten pages per minute**, spelling, punctuation.

reading selectively → two to three minutes per page, grammar, complete sentences.

reading analytically → five pages per hour, missing information, technical inaccuracies, paragraph organization.

the long look → one to two minutes per page, omission in title pages.

Novice editors → see their work as making document conform to style guideline or looking for mistakes in grammar.

Expert ones → they do this and more as a level of detail, sentence structure, and language.

Take a constructive attitude → People often see editors as grammar police.

Consult standard style guides → when controversies occur you need to consult a general style guide

Headers → help users locate the information within the context of the entire manual.

Introduction → helps users see the workplace application of function.

Heading → help the users see the hierarchical structure of information.

Icon → helps user recognize key information.

cuing pattern (italics, bold) → helps users recognize key information.

numbered steps → create clear areas for commands

lists and tables → helps users decide how to apply program functions to workplace tasks and give users a sense of control.

page numbers → helps users navigate among abstract concepts.

Color → helps users recognize task name.

Introduction → helps users apply functions to workplace tasks.

Icons → helps users identify screen elements quickly.

cuing (bold) → helps users recognize keystrokes easily.

hypertext links → helps users access related commands and tools.

Try out ideas on users, involve users in the process → eliminate alternatives that would not work for them.

The Design Problem → The problem of software design results from each reader needing to apply the system to a multitude of tasks.

أتوقع سؤال توصيل

No one carefully reads more than 2 sentences at a time. → Solution: Make paragraphs short. Include tables and lists whenever possible.

Most users begin using the table of contents before they read the manual. → Solution: Make table of contents complete. Use abbreviated, complete and **chapter-by-chapter** table of contents.

Most users go to the manual or help only after they have failed to perform tasks.

➔ Solution: Describe error recovery clearly and completely

Most readers do not read instruction first. ➔ Solution: Replace introduction with information about users needs, special documents features, or helpful routing information.

Most readers do not read any sections in its entirety. ➔ Solution: Tell users which section to go for particular tasks/problems.

Navigation ➔ Navigational aids are elements of a document that tell the reader **where to go next** for what kind of information.

Cross reference ➔ **point** to other sections or chapters with related info.

Layering ➔ is having **two versions** of information on the page at once

Indexes and TOC ➔ They are the two most **important** user **tracking and navigation** devices.

Cuing ➔ it refers to the technique of **including visual patterns** to make a certain kind of information memorable.

non scrolling regions ➔ appear on top of screen and stay there.

keyword and whole text searches ➔ **search box**.

links and jumps ➔ allows **user to go from one** topic to a related topic.

expanded text ➔ also called “**stretch text**” allows you to embed more detail into a topic so that the user can click on the expanded text to view the detail.

Indexes ➔ show an alphabetical **view of all the important topics and terminology** used in a help system.

pop ups ➔ to handle glossaries the user just has to click on **the term to see its definition**.

context sensitivity ➔ **the ability of a help system to present info** based on the current state of the program.

Histories ➔ history buttons allow user to **trace their steps**, easily go back to previous topics.

Grid lines ➔ **lines drawn** where page and column margin would fall.

Margins ➔ **spaces between text and page edge**.

Columns ➔ **spaces between the gridlines marking columns**

Gutters ➔ **space between columns**.

Baseline, grid line ➔ at the bottom of the text and graphics area that **define the bottom margin**.

degree of modularity ➔ means breaking the information into chunks of **text and graphic units** to make them easier for the user to digest.

in online help → can overcome the modularity problem by pop ups, and hypertext.

degree of structure → which means that we place the info on the page according to patterns, with certain kinds of info only in certain places.

Two column formats → allow the reader to distinguish between the guidance information and support information.

One column format → this will arrange both graphics and texts in the middle of the page.

left margin → which rule the page, so to speak, because most of the items on the page use the left margin as a starting place.

Columns → newspaper column (snake text) or table columns (discrete item)

headers & footers → contains product name, version number.

Screens → full and partial.

Rules → lines of varying width and length

Pagination → sequential and modular(2-10)

Windows screen format → it contains a non-scrolling region, usually it uses one column format.

Manual pages format → it consists of a handy format for dumping print documentation, it has no left margin.

Multiple window management → don't remove or destroy all the traces of user's work- allow the help screen to cover part of the screen, avoid window clutter.

Graphics → use simple graphics.

Screen grids → use narrow margins, less indentation

line spacing → single space will do it.

page size → the smaller the page the smaller the size and the less dense font.

Media → pages allow you to use smaller, more detailed fonts, whereas screens allow more limited range of fonts.

user expectations → the designer should pay attention to what kinds of type users see regularly in software documents.

Headings → to help reader locate important info. easily distinguished from the body text.

Hints, notes, and cautions → they must read easily and should catch the reader's eye.

User input, computer output → writers usually change the font of input and output messages from that of body text.

Tables and lists → make the tables different in indentation and column layout.