Documentation for Software

Olivier Hallot
The Document Foundation
Optimistic developer: "This is a common sense feature that doesn't need documentation"

End user:
The vision on learning process
User Bad or Poor Documentation

- Induce user resistance
- Bad documentation costs money!
Documentation is a love letter you write to yourself in the future
There are 4 documentation functions:
- With different writing style.
- For different objectives.
- Under different usage circumstances
- Under different formats
1- TUTORIALS

- Learning oriented, practical knowledge.
- Allows user to start using the software.
- It is a lesson, the author is the instructor.
- Example:
  - Cash flow in LibreOffice spreadsheets
  - Master documents for book assembly
2- Guides and How-to's

- Task oriented
- Shows how to solve a problem
- Step by step
- Examples
  - How to create a pivot table
  - How to print 3 pages
2- Guides and How-to's

Estante do LibreOffice

O LibreOffice é uma suite de escritório livre, de código aberto e completa. Funciona em computadores Windows, Linux, e macOS.
2- Guides and How-to's

Method 2

1) Position the cursor in the paragraph you want to be at the start of the next page. Right-click and choose **Paragraph > Paragraph** in the context menu.

2) On the *Text Flow* page of the Paragraph dialog (Figure 6), in the *Breaks* section, select **Insert** and **With page style**. In the *Style* drop-down list, select the page style for the next page. Do not select the *Page number* option. Click **OK** to insert the page break.

*Figure 6: Inserting a manual page break using the Paragraph dialog*

```
<table>
<thead>
<tr>
<th>Breaks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert</strong></td>
</tr>
<tr>
<td><strong>Type:</strong> Page</td>
</tr>
<tr>
<td><strong>Position:</strong> Before</td>
</tr>
<tr>
<td><strong>With page style:</strong> Default Page Style</td>
</tr>
<tr>
<td><strong>Page number:</strong> 1</td>
</tr>
</tbody>
</table>
```

**Tip**

See “Numbering the first page something other than 1” below for information on the use of the page number field in these dialogs.
2- Writing How-to's

- Use short, simple, easy-to-understand words and sentences. Be concise and clear. Ideally:
  - Paragraphs should be no more the six sentences long.
  - Sentences should only contain a maximum of 20 words, with the occasional sentence up to 25 words.
- Write in active voice or the imperative, using passive voice only when necessary or appropriate. For example, Do this, then do that. A case of appropriate passive voice is when the focus of the sentence is on the receiver of an action whose doer is obvious or is not important. For example, The File dialog is displayed.
2- Writing How-to's

- Use the present tense, using future tense only when necessary or appropriate. Try to make your descriptions timeless. For example, write The File dialog is displayed, rather than The File dialog will be displayed. Use future tense only when one event is necessarily later than another. For example, If you use styles, your documents will be easier to maintain.

- Avoid the third person, for example phrase the user. Use second person (you) or the imperative.

- Avoid over-using you. For example, instead of saying if you want to have table headers repeat on a new page, you need to do yyy, say to have table headers repeat..., do yyy.
2- Writing How-to's

- When appropriate (as in instructions), **use the imperative**. For example, instead of you should not use slang, say do not use slang.

- In circumstances where the only alternative to you is the passive voice, use you. For example, instead of in this window xxx can be done, say in this window you can do xxx.

- Use **gender-neutral language**, but don’t use awkward phrases like he or she to do so. See page 8 for some examples.

- Avoid using the **possessive apostrophe** (for example Peter’s). Rewrite the sentence to remove the need for a possessive apostrophe.

- Do not use contractions of words (for example: don’t becomes do not; won’t becomes will not, and so on).
Exercise: Explain Copy & Paste text
Oriented for understanding
Explanation prose
Tells about fundamentals, context and theory.
Examples:
- The Fourier Transform in Calc
- Index keys in database tables.
4-REFERENCES

- Information oriented
- Describes the engines
- Precise and complete
- Example
  - API documentation
  - Calc functions reference
4- References

createCursorByRange()

```java
com::sun::star::sheet::XSheetCellCursor createCursorByRange ( [in] com::sun::star::sheet::XSheetCellRange  aRange )
```

creates a cell cursor to travel in the given range context.

**Parameters**

- **aRange** the cell range for the cursor.

**See also**

- com::sun::star::sheet::SheetCellCursor

The documentation for this interface was generated from the following file:

- com/sun/star/sheet/XSpreadsheet.idl
Consequences

- Each documentation function has its own writing style and media format.
- When not making the right distinction, the documentation is much more hard to maintain.
  - By ignoring the right category of which the text belongs.
  - In the organization and maintenance of the documentation.
  - Mixed and confuse text, no clear objective. What do we want to document?
<table>
<thead>
<tr>
<th>Practical Steps</th>
<th>To Study</th>
<th>At work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutorial</td>
<td></td>
<td>How-To's Guides</td>
</tr>
<tr>
<td>Theoretical Knowledge</td>
<td>Explanations</td>
<td>References</td>
</tr>
<tr>
<td>TUTORIAL</td>
<td>GUIDES</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>Learning oriented</td>
<td>Problem oriented</td>
<td></td>
</tr>
<tr>
<td>Multimedia e-Learning</td>
<td>Multimedia e-learning</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPLANATION</td>
<td>REFERENCE</td>
<td></td>
</tr>
<tr>
<td>Understanding oriented</td>
<td>Information oriented</td>
<td></td>
</tr>
<tr>
<td>Multimedia E-learning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

- User Oriented
  - Tutorial
  - How-to’s and User Guides
  - Explanations

- Developer oriented
  - References
Conclusions

- Undocumented software is an **unknown software**, worse, an **ignored software**.
- Lack of documentation is a software industry issue.
- The software solution cost is (TCO):
  - **COST = PRODUCT + CULTURE** (→ Documentation)
- We know programmers don't like to document.
- Often seen as **professional immaturity**.
Thanks

- Daniele Proccida
  - https://www.divio.com/blog/documentation/
Keep Documenting !!!