DITA and Markdown

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu

syncro soft

<oXygen/> XML Editor

Agenda

- Content and markup
- Structured authoring
- Markdown
- DITA
- Markdown and DITA inside a documentation project

It all starts with the content

Create a Google account

How to create or set up your Google Account on your mobile phone.

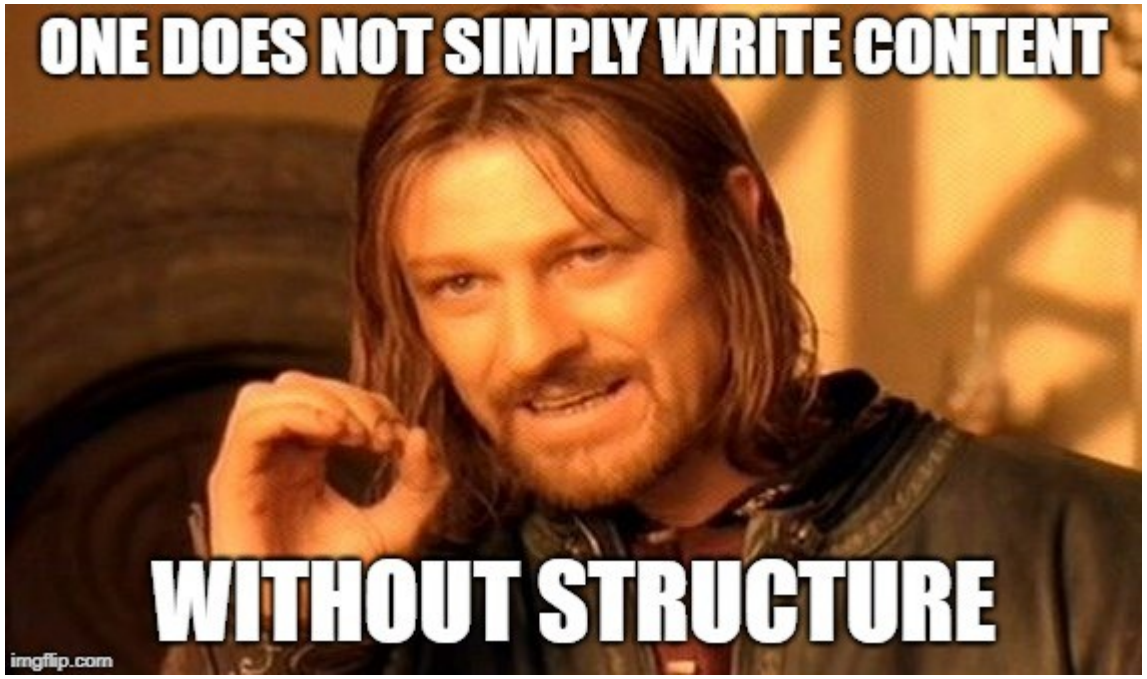
From a Home screen, swipe up to access Apps.

Tap Settings > Accounts

Tap Add account > Google.

A typical task

Content alone is not enough



Why do we need structure?

- Defines the organization/model of content
- Helps us enforce the defined model
- Increases consistency
- Automatic processing
- Faster publishing workflows

What Meaning Lies Beneath

Create a Google account — Title

How to create or set up your Google Account on your mobile phone. — Short description

From a Home screen, swipe up to access Apps.

Tap Settings > Accounts

Tap Add account > Google. — Procedure

Encode it with a Markup Language

Create a Google account	Title
How to create or set up your Google Account on your mobile phone.	Short description
From a Home screen, swipe up to access Apps. Tap Settings > Accounts Tap Add account > Google.	Procedure

Markdown

- Easy to learn
- Minimalistic
- Many authoring tools available
- Publishing tools

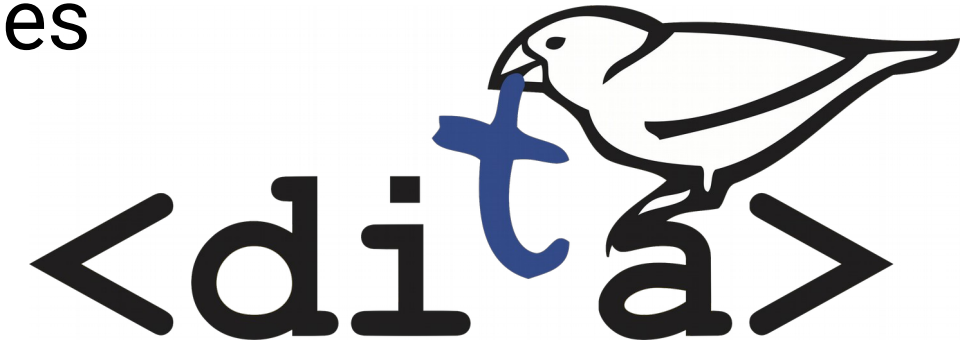
```
Create a Google account
============

How to create or set up your **Google Account** on your mobile phone.

* From a Home screen, swipe up to access Apps.
* Tap **Settings** > **Accounts**
* Tap **Add account** > **Google**.
```

XML

- e(**X**)tensible (**M**)arkup (**L**)anguage
- Define your own tags/markup
- Powerful mechanisms to enforce valid content structure (DTD/Schema/Schematron)

DITA

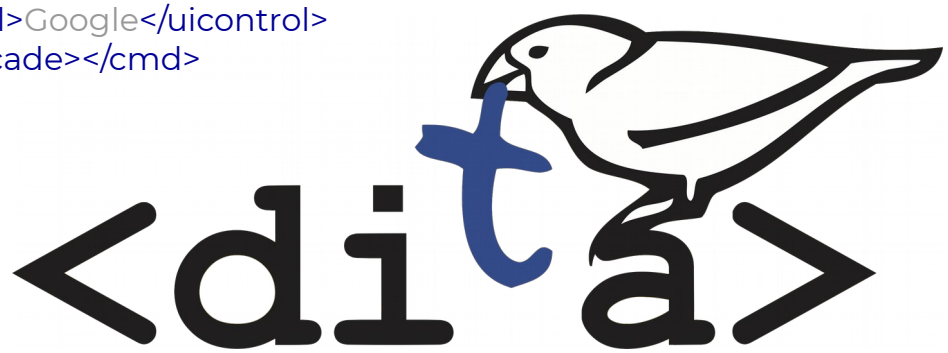
- DITA is an XML-based open **standard** for structuring, developing, managing, and publishing content.
- Semantic markup (separates formatting from content)
- Strong content reuse concepts
- Restrictions and specializations
- Huge ecosystem of publishing choices



Side by side

```
Create a Google account
============

How to create or set up your **Google Account**
on your mobile phone.

1. From a Home screen, swipe up to access Apps.
1. Tap **Settings** > **Accounts**
1. Tap **Add account** > **Google**.
```

```
<task id="create_google_account">
  <title>Create a Google account</title>
  <shortdesc>How to create or set up your <term>Google Account</term>
on your mobile phone.</shortdesc>
  <taskbody>
    <steps>
      <step>
        <cmd>From a Home screen, swipe up to access Apps.</cmd>
      </step>
      <step>
        <cmd>Tap <menucascade>
          <uicontrol>Settings</uicontrol>
          <uicontrol>Accounts</uicontrol>
        </menucascade></cmd>
      </step>
      <step>
        <cmd>Tap <menucascade>
          <uicontrol>Add account</uicontrol>
          <uicontrol>Google</uicontrol>
        </menucascade></cmd>
      </step>
    </steps>
  </taskbody>
</task>
```



Side by side – visual mode

Create a Google account

How to create or set up your **Google Account** on your mobile phone.

1. From a Home screen, swipe up to access Apps.
2. Tap **Settings** > **Accounts**
3. Tap **Add account** > **Google**.

Create a Google account

Short Description: How to create or set up your *Google Account* on your mobile phone.

1. From a Home screen, swipe up to access Apps.
2. Tap **Settings→Accounts**
3. Tap **Add account→Google**

Scenario

- Main documentation project written in DITA
- Contributors (devs) sending content in Markdown

[1] Convert MD to DITA

- MD => HTML => DITA
  - https://pandoc.org/
  - http://dita-ot.sourceforge.net/1.5.2/readme/DITA-h2d-ant.html#h2d-ant
- MD => DITA
  - DITA-OT plugin developed by Jarno Elovirta

    https://github.com/jelovirt/dita-ot-markdown
  - Oxygen Batch Converter plugin

    https://github.com/oxygenxml/oxygen-resources-converter

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - https://en.wikipedia.org/wiki/URL

  - <topicref href="md2dita:/topic.md" format="dita"/>
  - https://github.com/oxygenxml/dita-glass

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - <topicref href="md2dita:/topic.md" format=”dita"/>
  - https://github.com/oxygenxml/dita-glass
- Refer MD files directly in your map
  - <topicref href="tasks/changingtheoil.md" format="markdown"/>
- Seamless publishing

Using specific DITA concepts in MD

- Metadata
- Specialization types
- Titles and document structure
- Image and Key references

https://github.com/jelovirt/dita-ot-markdown/wiki/Syntax-reference

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - <topicref href="md2dita:/topic.md" format=”dita"/>
  - https://github.com/oxygenxml/dita-glass
- Refer MD files directly in your map
  - <topicref href="tasks/changingtheoil.md" format="markdown"/>
  - **<topicref href="tasks/changingtheoil.md" format="mdita"/>**
- Seamless publishing

What is Lightweight DITA?

- LwDITA is a proposed standard for expressing *simplified DITA documents* in *XML, HTML5, and Markdown*.
- The core goals of LwDITA:
  - Provide a simpler DITA experience
  - Provide mappings between *XML, HTML5, and Markdown* that enable individuals to:
    - Author content in the format of their choice
    - Easily exchange and publish content whose source exists in these different markup languages

What is Lightweight DITA?

- LwDITA is a proposed standard for expressing *simplified DITA documents* in *XML, HTML5, and Markdown*.
- The core goals of LwDITA:
  - Provide a simpler DITA experience
  - Provide mappings between *XML, HTML5, and Markdown* that enable individuals to:
    - Author content in the format of their choice
    - Easily exchange and publish content whose source exists in these different markup languages

Advantages

- Single sourcing across DITA and Markdown
- Collaboration on Markdown source
- Use DITA features and publishing options with Markdown
- Use Markdown publishing options
  - https://github.com/jelovirt/dita-ot-markdown#generating-markdown-output
  - Make use of publishing platforms like Jekyll, Vuepress, Mkdocs etc.
    - https://nostalgic-hamilton-b6dae0.netlify.com/

Disadvantages

- Markdown lacks semantics
  - https://github.com/IBM-Cloud/docs-services/tree/staging#using-the-copyright-and-last-updated-header-required
  - https://raw.githubusercontent.com/IBM-Cloud/docs-services/staging/getting_started_template/servicename_task.md

Disadvantages

- Markdown lacks semantics
- Consistency challenges (not a standard, can receive different flavors)
  - https://github.com/IBM-Cloud/docs-services/tree/staging#using-the-copyright-and-last-updated-header-required
  - https://raw.githubusercontent.com/IBM-Cloud/docs-services/staging/getting_started_template/servicename_task.md

Disadvantages

- Markdown lacks semantics
- Consistency challenges (not a standard, can receive different flavors)
- Markdown language restrictions
- No reuse mechanisms
- Review/Collaboration tracking challenges
  - When done on the Markdown source
    - No support for tracking changes
    - Difficult to visualize changes (diffs needed)
  - When done on converted content, like PDF
    - Writer Generate PDF => Devs review on PDF => Writer incorporates Review.
    - Extra overhead to incorporate review into source

Markdown Consistency Challenges

- What can we do?

Markdown Consistency Challenges

- XML/DITA has Schematron
- It does structure checks too

```
<sch:pattern>
    <sch:rule context="topic">
        <sch:assert test="shortdesc">Please add a short description.</sch:assert>
    </sch:rule>
</sch:pattern>
```

Schematron for Markdown

- Markdown syntax maps to a subset of HTML tags
- Apply Schematron on the HTML with back-mapping support

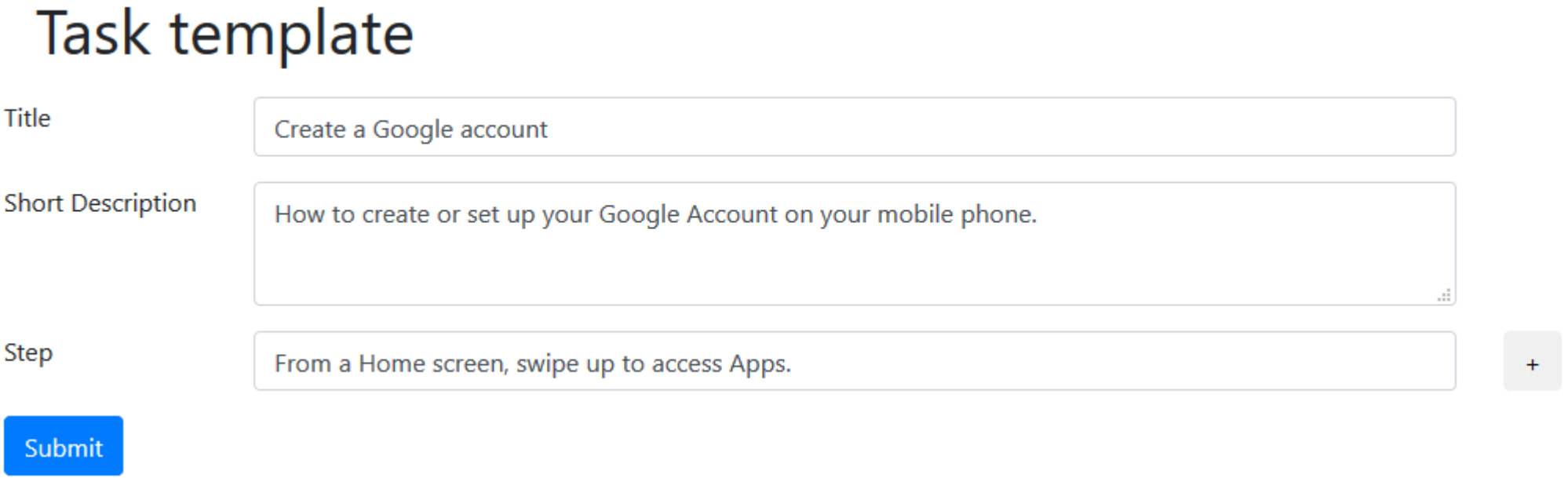
Why do Devs tend to use Markdown?

- It has a low learning curve
- You do things fast
- Don't have time to learn another language
- They don’t need any additional tool installed on their system

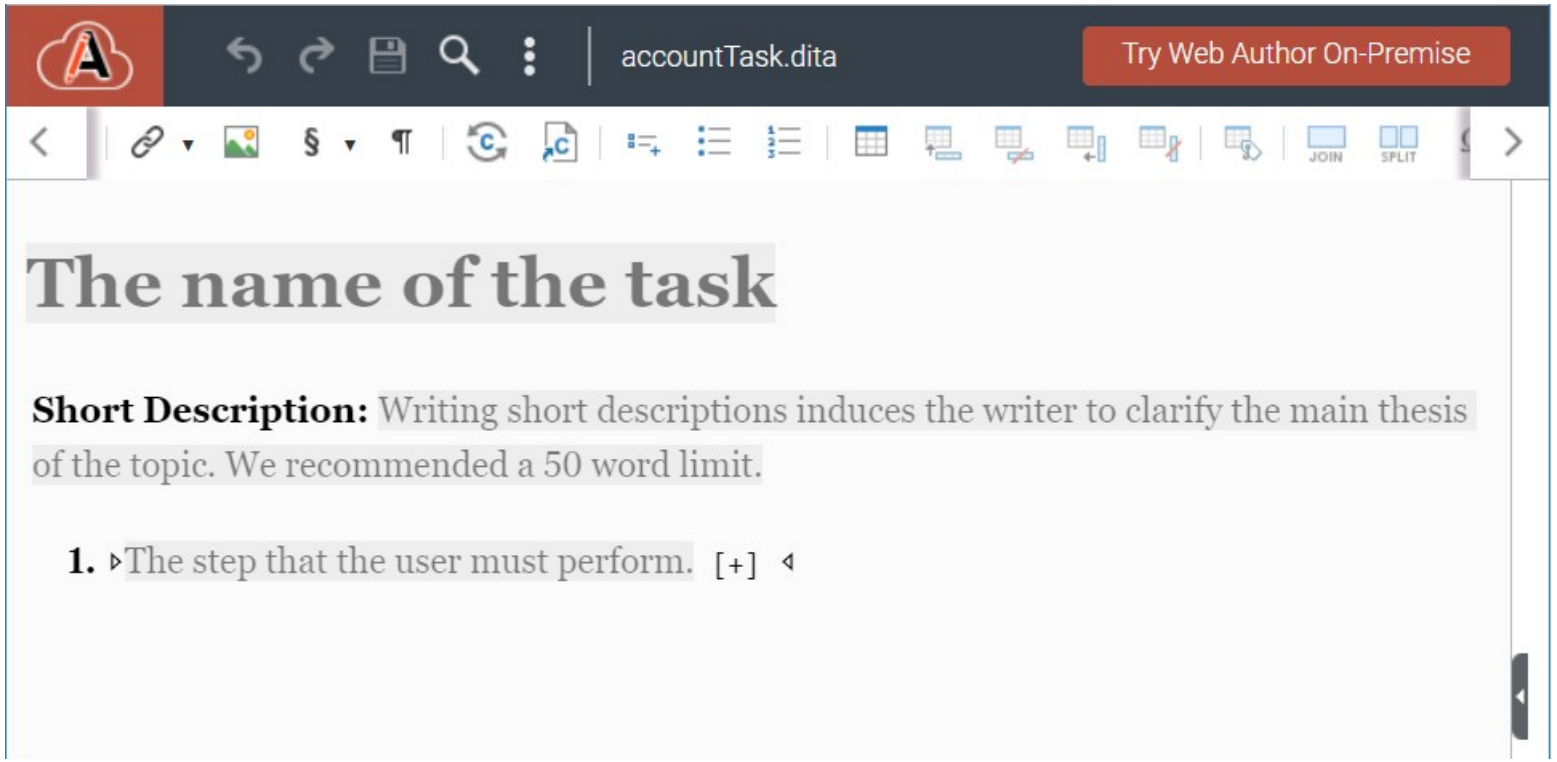
[3] Could Devs write DITA?

- Learn it as you use it through Markdown2DITA controlled conversions. Like a *“Learning assistant”*.
  - Powered by Schematron Quick Fixes
  - https://github.com/oxygenxml/ditaMark
- Give Devs specialized editing enviroments (cloud based):
  - Specialized UI (e.g. HTML form) that generates consistent DITA
  - Specialized Web based DITA editors
    - Oxygen XML Web Author
    - Oxygen Content Fusion

Could Devs write DITA?



Could Devs write DITA?



Conclusion

- There is no one-size-fits-all solution
- Convert if it's a one time thing
- Keep them together and achieve single sourcing
- Consistency/Collaboration might require a switch to DITA

THANK YOU!

Any questions?

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu