



**FREE eBook**

**LEARNING**

**Documentation**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#documenta  
tion**

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with Documentation.....</b>	<b>2</b>
Versions.....	2
Examples.....	2
About Documentation.....	2
Hello World.....	2
<b>Chapter 2: Example order.....</b>	<b>4</b>
Examples.....	4
Pinned example.....	4
Votes.....	4
<b>Chapter 3: Examples.....</b>	<b>5</b>
Examples.....	5
How to write a good example?.....	5
How can I move an example to another tag?.....	5
<b>Chapter 4: Tables.....</b>	<b>6</b>
Examples.....	6
Linebreaks.....	6
Automatic Linebreaks.....	6
Manual Linebreaks.....	6
<b>Chapter 5: Topic scope.....</b>	<b>8</b>
Introduction.....	8
Examples.....	8
Language feature Topics.....	8
Grab bag Topics.....	8
<b>Perl Tips and Tricks.....</b>	<b>8</b>
<b>Chapter 6: Topics.....</b>	<b>9</b>
Introduction.....	9
Examples.....	9
Creating a Topic.....	9
Remember that in Stack Overflow Documentation, examples are the star of the show!.....	9

Anatomy of a Topic.....	9
The Introductory topic.....	10
<b>Chapter 7: Versions.....</b>	<b>11</b>
Examples.....	11
Version Table.....	11
Multiple Version Tables.....	11
desktop development kit.....	12
mobile development kit.....	12
Versions of a Topic.....	12
Versions of an Example (single version table).....	13
Versions of an Example (multiple version tables).....	13
Available Conditionals.....	14
Multiple Conditionals.....	15
<b>Chapter 8: Voting.....</b>	<b>16</b>
Examples.....	16
You can vote for your own examples.....	16
<b>Credits.....</b>	<b>17</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [documentation](#)

It is an unofficial and free Documentation ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Documentation.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with Documentation

## Versions

Version	Release Date
Private Beta	2016-03-26
1	2016-07-25
2	2016-08-04
3	2016-08-29
4	2016-09-12
5	2016-09-29
6	2016-10-20
7	2016-11-29

## Examples

### About Documentation

#### What can be documented?

- Examples of functions for various languages.
- A brief introduction of each tag.

#### What is the difference between a question and a topic?

- Topics have a broader scope than questions; documentation topics that are asked as a question can be closed because they are too broad.
- Topics should have multiple examples; a request that can be served by a topic with a single example (e.g., one block of code) is probably too narrow.

### Hello World

A hello world topic should consist of the most basic, complete, runnable, starter program available, that may be achieved by the widest amount of versions of the language, framework or system being documented. If necessary, the remarks should include basic instructions on how to prepare and execute the examples.

The topic should generally not require more than one or two examples, except in cases where later versions of the language are not backwards compatible. Remember, the hello world topic will be the starting point for anyone just beginning to learn a language. As such it should demonstrate minimum working examples that a learner can use to achieve a correct program. It is not necessarily a place to demonstrate a language's newest features.

Read [Getting started with Documentation online](https://riptutorial.com/documentation/topic/258/getting-started-with-documentation):

<https://riptutorial.com/documentation/topic/258/getting-started-with-documentation>

---

## Chapter 2: Example order

### Examples

#### Pinned example

One example per topic may be pinned by clicking the pin icon in the bottom left corner of the example. So long as a topic is pinned, it will stay at the top of a topic's example collection.

#### Votes

Aside from pinned examples, it is not possible to enforce the order of examples. Instead, order is naturally selected in descending order of total votes. The most popular example will therefore bubble to the top of the collection of examples.

Read Example order online: <https://riptutorial.com/documentation/topic/5458/example-order>

---

# Chapter 3: Examples

## Examples

### How to write a good example?

Example is the main part of Documentation, it is thus important to provide clear and useful examples.

Good examples are self-contained and succinct. Examples usually would have code for users to understand the topic better.

Refer to the [Help Center](#) for more information.

### How can I move an example to another tag?

1. Click the “Move examples” button
2. Select the examples you want to move
3. Click “Move”
4. Select “Search for a Topic”
5. Paste the URL of the topic in the search box
6. Click “Move Examples”.
7. Click “Submit Drafts” or “Edit Drafts”

Read Examples online: <https://riptutorial.com/documentation/topic/1469/examples>



---

# Chapter 4: Tables

## Examples

### Linebreaks

#### Automatic Linebreaks

Table content is line wrapped automatically if it is too wide. In the following example, text is added to the second column, which causes both the text itself and the column header of the first column to be wrapped.

```
| this is a really wide column header | another header |  
| --- | --- |  
| | |
```

this is a really wide column header	another header
-------------------------------------	----------------

```
| this is a really wide column header | another header |  
| --- | --- |  
| | Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut  
labore et dolore magn |
```

this is a really wide column header	another header
----------------------------------------	----------------

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magn

#### Manual Linebreaks

It's also possible to insert manual breaks as you would in regular text. This can allow other columns to expand even further.

```
| this <br/> is <br/> a <br/> really <br/> wide <br/> column <br/> header | another header |  
| --- | --- |  
| | Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut  
labore et dolore magn |
```

this is a really wide column  header	another header
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magn	

Read Tables online: <https://riptutorial.com/documentation/topic/4983/tables>

---

# Chapter 5: Topic scope

## Introduction

Topics should be made as simple as possible, but not simpler. This one suggests some organizing principles seen on Stack Overflow Documentation.

## Examples

### Language feature Topics

A natural division is simply to make one topic for each language feature. For instance, [Python](#) has:

- [Classes](#)
- [Functions](#)
- [Generators](#)
- [Decorators](#)
- ...

On the downside, features can be quite complicated and it's easy to come up with many more examples than make sense for a single page. [Classes](#) has 16 examples, for instance. Some of those could probably be moved to other feature topics, such as [Inheritance](#), [Properties](#), [Methods](#), etc.

### Grab bag Topics

Some topics are loosely associated examples under a theme, such as [C# 6.0 Features](#) and [Common pitfalls](#). On the whole, this style of topic has been discouraged since examples per topic were [limited](#) as there's no natural place to stop adding content.

It's probably better to move the examples to another Topic (perhaps for [language features](#)). I haven't seen this done, but it's possible these topics could be converted into a meta topic that includes an example or two that consists largely of links to other topics or examples:

---

## Perl Tips and Tricks

- Don't parse XML with a regex! Use [an XML parsing module](#).
- If you need to comment out whole blocks of code, use a [POD comment block](#).
- ...

Read [Topic scope](#) online: <https://riptutorial.com/documentation/topic/8361/topic-scope>

---

# Chapter 6: Topics

## Introduction

Topics organize Examples into an easily digestible page. They should be self-contained and written for the benefit of the reader in a hurry.

## Examples

### Creating a Topic

Use the following steps to create a topic:

1. Click `Create New Topic` on the tag's dashboard.
2. Name the topic.
3. Provide detailed examples in the "Examples" section.
4. If you have more to share, use the "Syntax," "Parameters," and "Remarks" sections.

You have successfully created a topic! The topic will be available to everyone, and everyone can edit it.

## Remember that in Stack Overflow Documentation, examples are the star of the show!

### Anatomy of a Topic

A topic is made up of several sections:

**Title:** What is the subject of the topic. Search existing topics first to make sure you are not repeating content that already exists. Topics already covered elsewhere on Stack Overflow may be deleted.

**Versions:** The versions where the material covered by the topic applies. You may pick from any of the versions listed in the introduction topic, or remove them all to indicate that the topic applies to all versions. See [Versions](#) for more information.

**Introduction (optional):** Explains the organizing principle of the contained examples to both readers and contributors. It resides above the examples and is limited to 500 characters.

**Examples:** Code examples for the topic. The code should be easy to read and well commented. Prose should also be used to explain the concepts needed to understand the code, but try not to be overly verbose.

One example per topic can be "pinned", meaning it floats above the other examples. This can be useful for an extended intro that is too long for the Introduction section.

**Syntax (optional):** It may be appropriate to include a syntax for the topic at hand. This section is always a code block and must be a markdown list:

```
- class.method(required_parameter [, optional])  
- function f(int x) = { code defining what the function does }
```

**Parameters (optional):** When describing a topic as with syntax, it may also be appropriate to include parameter details. Parameters are formatted as a two-column table:

```
| Parameter | Details |  
| ----- | ----- |  
|      yin |     dark |  
|      yang |     light |
```

**Remarks (optional):** Any additional Information regarding the topic that may prove valuable to the end reader. Remarks are freeform.

## The Introductory topic

When a new documentation tag is created, StackOverflow automatically generates an *introductory topic*, acting as a placeholder for the first actual topic.

The introductory topic contains placeholder texts and pointers to point contributors in the right directions for helping expanding on the new documentation tag.

**Please note that introductory topics cannot be deleted. See [this question](#)**

**Read Topics online:** <https://riptutorial.com/documentation/topic/841/topics>

# Chapter 7: Versions

## Examples

### Version Table

Every tag has an overview topic. This topic has a special section named "Versions". In that section, different versions can be defined in a table that has that should

- have at least 2 columns:
  1. the first one should be the name of the version
  2. the last one should be the release date of the version, by which versions will be ordered.

It should be in the `yyyy-MM-dd` format and a valid date.

There can be additional columns between the two to include more information. It makes sense to list versions in chronological order.

This markup:

```
| Version | Additional Information | Release date |  
| --- | ----| ----|  
| 1.R | version names do not have to be numbers | 1980-11-25 |  
| 1 | we have to start somewhere | 1980-11-24 |
```

produces this result:

Version	Additional Information	Release date
1.R	version names do not have to be numbers	1980-11-25
1	we have to start somewhere	1980-11-24

### Multiple Version Tables

For some tags it makes sense to have multiple version tables. There might be different subsets of a programming language or framework available, say for different device types as in this example.

The different tables should be prefaced with a heading.

This markup:

```
## desktop development kit ##
```

```

| Version | Release date |
| --- | ----|
| 1.0 | 1980-12-24 |
| 1 RC | 1980-11-25 |
| 1 beta | 1980-11-24 |

## mobile development kit ##

| Version | Release date |
| --- | ----|
| M 2 | 1980-12-24 |
| M 1.2 | 1980-11-25 |
| M 1 | 1980-11-24 |

```

produces this result:

## desktop development kit

Version	Release date
1.0	1980-12-24
1 RC	1980-11-25
1 beta	1980-11-24

## mobile development kit

Version	Release date
M 2	1980-12-24
M 1.2	1980-11-25
M 1	1980-11-24

### Versions of a Topic

A topic might only apply to certain versions. When creating a new topic, the appropriate versions can be chosen in the title section of the topic.

In the following example .NET 2.0, 4.0, 4.5.1; Compact Framework 3.7 and Micro Framework 4.2 are selected.

**Title**

**Versions**

.NET

1.0
  1.1
  2.0
  3.0
  3.5
  3.5 SP1
  4.0
  4.5
  4.5.1
  4.5.2
  4.6
  4.6.1

Compact Framework

1.0
  2.0
  3.5
  3.7
  3.9

Micro Framework

4.2
  4.3
  4.4

This selection will show up next to the topic title in the list of all topics

## Versions of an Example (single version table)

An example cannot be associated with a version number directly.

Parts of it can however be declared to apply only for certain versions.

```
<!-- if version [eq Java SE 1.3] -->
```

This content is for Java SE 1.3

```
<!-- end version if -->
```

### New Example



Java SE 1.3

This content is for Java SE 1.3

## Versions of an Example (multiple version tables)

If there are many version tables, the title of the desired one has to be included in angle brackets.

```
<!-- if version <Micro Framework> [eq 4.4] -->
```

This content is for Micro Framework 4.4

```
<!-- end version if -->
```



Micro Framework 4.4

This content is for Micro Framework 4.4

## Available Conditionals

The [markdown help](#) states

Available conditionals are `gt`, `gte`, `lt`, `lte`, `eq`, and `neq`.

```
<!-- if version [eq C++03] -->
```

`eq` equal to

```
<!-- end version if -->
```

```
<!-- if version [neq C++03] -->
```

`neq` not equal to

```
<!-- end version if -->
```

```
<!-- if version [gt C++03] -->
```

`gt` greater than

```
<!-- end version if -->
```

```
<!-- if version [gte C++03] -->
```

`gte` greater than equal

```
<!-- end version if -->
```

```
<!-- if version [lt C++03] -->
```

`lt` less than

```
<!-- end version if -->
```

```
<!-- if version [lte C++03] -->
```

`lte` less than equal

```
<!-- end version if -->
```

## New Example



C++03

**eq** equal to

≠ C++03

**neq** not equal to

> C++03

**gt** greater than

≥ C++03

**gte** greater than equal

< C++03

**lt** less than

≤ C++03

**lte** less than equal

## Multiple Conditionals

Multiple conditions can be used.

```
<!-- if version [lte 1.2.0] [gt 1.3.12] [gt 1.4.9] [neq 1.2.0] [eq 1.5.7] -->
```

```
content
```

```
<!-- end version if -->
```

## New Example



```
≤ 1.2.0, > 1.3.12, > 1.4.9, ≠ 1.2.0, 1.5.7
```

```
content
```

Read Versions online: <https://riptutorial.com/documentation/topic/4541/versions>

---

# Chapter 8: Voting

## Examples

### You can vote for your own examples

You can upvote (or downvote, if you want...) examples you created. Although you won't get any reputation for doing so, upvoting your the example will influence the examples' ordering on the page.

Any previous contributors to the examples *will* get [reputation](#).

On the other hand, you might consider waiting for other people to contribute so they can earn reputation from your upvote.

Note that this behavior is not a bug, as indicated in [this meta post](#).

Read [Voting online](#): <https://riptutorial.com/documentation/topic/8334/voting>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with Documentation	<a href="#">Community</a> , <a href="#">Greg Bray</a> , <a href="#">hillary.fraley</a> , <a href="#">Michael Allan</a> , <a href="#">mnoronha</a> , <a href="#">Panda</a> , <a href="#">Stephen Leppik</a> , <a href="#">Toby</a>
2	Example order	<a href="#">4444</a> , <a href="#">Toby</a>
3	Examples	<a href="#">J F</a> , <a href="#">Panda</a>
4	Tables	<a href="#">null</a>
5	Topic scope	<a href="#">Jon Ericson</a>
6	Topics	<a href="#">hillary.fraley</a> , <a href="#">Jon Ericson</a> , <a href="#">Panda</a> , <a href="#">Rick</a> , <a href="#">Stephen Leppik</a> , <a href="#">Toby</a> , <a href="#">xam</a>
7	Versions	<a href="#">null</a>
8	Voting	<a href="#">Jon Ericson</a> , <a href="#">JonasCz</a> , <a href="#">Laurel</a> , <a href="#">Panda</a>