

**QAProcess (/QAProcess):** [DevelopersGuide \(/DevelopersGuide\)](#) | [Review Status \(/Review%20Status\)](#) | [PackageProposalProcess \(/PackageProposalProcess\)](#) | [PackageDocumentation \(/PackageDocumentation\)](#) | [APIReviewProcess \(/APIReviewProcess\)](#) | [DocReviewProcess \(/DocReviewProcess\)](#) | [CodeReviewProcess \(/CodeReviewProcess\)](#) | [StackDocumentation \(/StackDocumentation\)](#) | [StackVersionPolicy \(/StackVersionPolicy\)](#) | [AutomatedTesting \(/AutomatedTesting\)](#) | [StackReleaseProcess \(/StackReleaseProcess\)](#) | [WritingTutorials \(/WritingTutorials\)](#) | [Graveyard \(/Graveyard\)](#)

**Note:** Before writing a tutorial take the time to review the ROS tutorials ([/ROS/Tutorials](#)).

💡 Please ask about problems and questions regarding this tutorial on [answers.ros.org](http://answers.ros.org) (<http://answers.ros.org>). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

# How to Write a Tutorial

**Description:** This tutorial covers useful template and macros for writing tutorials, along with example tutorials that are available for guidance on [ros.org \(/Documentation\)](#)

**Keywords:** teaching, tutorials, writing

**Tutorial Level:** BEGINNER

## Sommaire

1. How to Start
2. Creating a New Tutorial
3. Filling in the Template Header
  1. Note
  2. Title
  3. Description
  4. Next
  5. Level
  6. Keywords
4. Useful Macros
  1. Parsers
  2. CodeRef
  3. TutorialChain
  4. IncludeCSTemplate
  5. Embed Tutorials
5. Example Tutorials

## 1. How to Start

To start writing tutorials, go to the stack or package page where you plan on writing the tutorial. If there are no tutorials yet go to the non-existent [/Tutorials](#) wiki page.

After you open the page you will be redirected to:

## This page does not exist yet. What type of page are you trying to create?

⚠ Please see [EditingTheWiki](#) for guidelines on how our wiki is organized and tips on creating new pages.

### ROS package or Stack

If you're creating the initial page for a stack or package, please give it the page the same name as the stack or package itself, and use one of the following templates:

- [StackTemplate](#) - Template this new page to hold usage documentation for a stack
- [PackageTemplate](#) - Template this new page to hold usage documentation for a package

If you're creating additional documentation for a package or stack, feel free to structure them however you like, but please keep them within the namespace of your package (e.g. "ros.org/wiki/my\_package/more\_details") [Create new empty page](#)

### Tutorials

To keep tutorials organized, each stack and package has a link to tutorials at the bottom of the page. If you got here via that link please start a list of tutorials with this template:

- [TutorialIndexTemplate](#)

Once you save that page, it will have a "Create Tutorial" button that you can use to create new tutorial pages which will be automatically indexed and easier for users to find.

Click on the TutorialIndexTemplate (/TutorialIndexTemplate) link, and save the suggested preview without changes. This will create a new page for creating tutorials as shown below:

Create a new tutorial:

## Existing Tutorials

No results found.

This page is useful for creating tutorials and will display the list of current tutorials. In the sidebar of the stack or package page there will be a link created for tutorials as shown below:

#### Package Links:

[Code API](#)  
[Tutorials](#)  
[Troubleshooting](#)  
[Reviews](#) (unreviewed)  
[Dependency Tree](#)

#### Stack Links:

[Tutorials](#)  
[Troubleshooting](#)  
[Change List](#)  
[Roadmap](#)  
[Reviews](#) (unreviewed)

## 2. Creating a New Tutorial

In the create a new tutorial box enter a page name for your new tutorial and click the button. This will create a new page using the TutorialTemplate (/TutorialTemplate) which looks something like (some ! are added in front of the keys so that the header for this page is generated correctly):

Afficher/masquer les numéros de lignes

```
#####
##FILL ME IN
#####
## for a custom note with links:
## !note =
## for the canned note of "This tutorial assumes that you have completed the
## previous tutorials:" just add the links
## !note.0=
## descriptive title for the tutorial
## !title =
## multi-line description to be displayed in search
## !description =
## the next tutorial description (optional)
## !next =
## links to next tutorial (optional)
## !next.0.link=
## !next.1.link=
## what level user is this tutorial for
## !level= (BeginnerCategory, IntermediateCategory, AdvancedCategory)
## !keywords =
#####

!<<IncludeCSTemplate(TutorialCSHeaderTemplate)>>
```

Note that until you save the page at least once, the preview button does not seem to be working properly.

## 3. Filling in the Template Header

The above template is used to auto generate a header similar to the one at the top of this page. Now let's look at the different parts of the template and how they are used.

### 3.1 Note

Note generates the blue box at the top of the page with reminders or information for a someone about to follow your tutorial.

This filling text in here will generate a completely custom note that looks like below

```
note= This is my awesome text
```

**Note:** This is my awesome text.

Otherwise you can use:

```
note.0=[ [ROS/Tutorials|ROS tutorials]]
note.1=[ [Documentation| ros.org]]
```

This will add the link to the end of the standard note, *This tutorial assumes that you have completed the previous tutorials:* similar to below:

**Note:** This tutorial assumes that you have completed the previous tutorials: ROS tutorials (/ROS/Tutorials), ros.org (/Documentation).

Note that if you want to place links in your custom text for the note they should go in-line rather than in note.0, note.1, etc. This will keep "This tutorial assumes that you have completed the previous tutorials:" from appearing undesired.

## 3.2 Title

This is a descriptive title that will be displayed at the top of the page and also be the title of the tutorial displayed in the tutorial index page.

```
!title= My Awesome Tutorial
```

## 3.3 Description

This is a description that will be displayed at the top of the page and also be the description of the tutorial displayed in the tutorial index page.

```
description= My awesome tutorial is amazingly awesome and cool, you won't even believe it, do it now!
```

This will look like:

**Description:** My awesome tutorial is amazingly awesome and cool, you won't even believe it, do it now!

## 3.4 Next

This is used for linking to any next tutorials that might come after your tutorial, these fields are optional and will display nothing if not filled.

```
next=Here are the next awesome tutorials
next.0.link=[ROS/Tutorials|ROS tutorials]]
next.1.link=[actionlib_tutorials/Tutorials|actionlib tutorials]]
```

This will look like:

**Next Tutorial:** Here are the next awesome tutorials ROS tutorials (/ROS/Tutorials) actionlib tutorials (/actionlib\_tutorials/Tutorials).

OR you can:

```
next.0.link=[actionlib_tutorials/Tutorials|Actionlib tutorials]]
```

Which will look like:

**Next Tutorial:** Actionlib tutorials (/actionlib\_tutorials/Tutorials).

## 3.5 Level

This key serves a dual purpose, it let's the user know what level tutorial they are about to complete and is used as a category to search over for the tutorial index page.

```
level=AdvancedCategory
```

This will look like:

**Tutorial Level:** ADVANCED

## 3.6 Keywords

Here you can add keywords to help users search for your tutorials.

```
keywords= great, cool, awesome
```

This will look like:

**Keywords:** great, cool, awesome

## 4. Useful Macros

Here are some useful macros for helping you write tutorials.

### 4.1 Parsers

Currently the following parsers are installed on the wiki for your usage:

**C++**

example:

```
{ {{
#!cplusplus
#include <ros/ros.h>
int main(int argc, char** argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ...
}}
```

result:

Afficher/masquer les numéros de lignes

```
1 #include <ros/ros.h>
2 int main(int argc, char** argv)
3 {
4     ros::init(argc, argv, "talker");
5     ros::NodeHandle n;
6     ...
```

**latex**

example:

```

{{{
#!latex
$$\overline{\overline{J}} \cdot \overline{v} = -\overline{k}$$
}}}
```

result:

$$\overline{\overline{J}} \dot{\overline{v}} = -\overline{k}$$

**python**

example:

```

{{{
#!python
#!/usr/bin/env python
import roslib; roslib.load_manifest('beginner_tutorials')
import rospy
from std_msgs.msg import String
def talker():
    pub = rospy.Publisher('chatter', String)
    rospy.init_node('talker', anonymous=True)
    ...
}}}
```

result:

Afficher/masquer les numéros de lignes

```

1 #!/usr/bin/env python
2 import roslib; roslib.load_manifest('beginner_tutorials')
3 import rospy
4 from std_msgs.msg import String
5 def talker():
6     pub = rospy.Publisher('chatter', String)
7     rospy.init_node('talker', anonymous=True)
8     ...
```

## 4.2 CodeRef

In a tutorial, you want to reference a code block that you have created earlier in the tutorial you can use CodeRef.

```

{{{
#!cplusplus block=blockname
code....
is super
and awesome
}}}
```

You can then use the CodeRef macro to refer to it

```
<<CodeRef(blockname,1,2)>>
```

This then looks like:

Afficher/masquer les numéros de lignes

```
1 code....  
2 is super
```

This works with python:

Afficher/masquer les numéros de lignes

```
1 #!/usr/bin/env python  
2 import roslib; roslib.load_manifest('beginner_tutorials')
```

Note that `#!/cplusplus` or `#!/python` line needs to be in a new line (ie. new line after three beginning parentheses).

You have to save (not just preview) before the new code block becomes part of the system.

## 4.3 TutorialChain

The TutorialChain (/TutorialChain) macro can be used to generate a list of Tutorials. It will walk the next links in each Tutorial.


```
<<TutorialChain(WritingTutorials)>>
```

This shows the title and description of the tutorial chain starting with the tutorial WritingTutorials.

### 1. How to Write a Tutorial (/WritingTutorials)

This tutorial covers useful template and macros for writing tutorials, along with example tutorials that are available for guidance on [ros.org](http://ros.org) (/Documentation)

## 4.4 IncludeCSTemplate

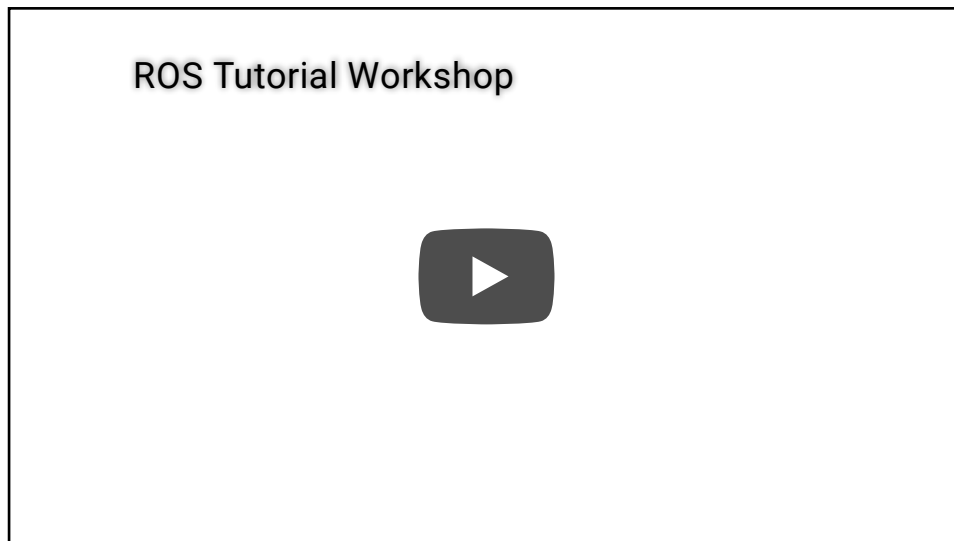
This macro takes the keys at the top of the page and turns them into  clearsilver (<http://clearsilver.net>) variables that can be operated on using the clearsilver templating commands. In the case of tutorials the keys are used in the TutorialCSHeaderTemplate.

```
!<<IncludeCSTemplate(TutorialCSHeaderTemplate)>>
```

## 4.5 Embed Tutorials

You can embed videos using the Youtube and Vimeo macros

```
<<Youtube(Q5KC-trrw_o)>>
```



## 5. Example Tutorials

Here is a list of good tutorials to base your tutorials off of:

- ROS Tutorials (</ROS/Tutorials>)
- Driver Tutorials (</ROS/DriverTutorials>)
- actionlib tutorials ([/actionlib\\_tutorials](/actionlib_tutorials))

Except where otherwise noted, the

ROS wiki is licensed under the

Creative Commons Attribution 3.0

(<http://creativecommons.org/licenses/by/3.0/>)

Wiki: WritingTutorials (dernière édition le 2017-12-21 07:09:55 par Bluetronics (</Bluetronics>))

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)