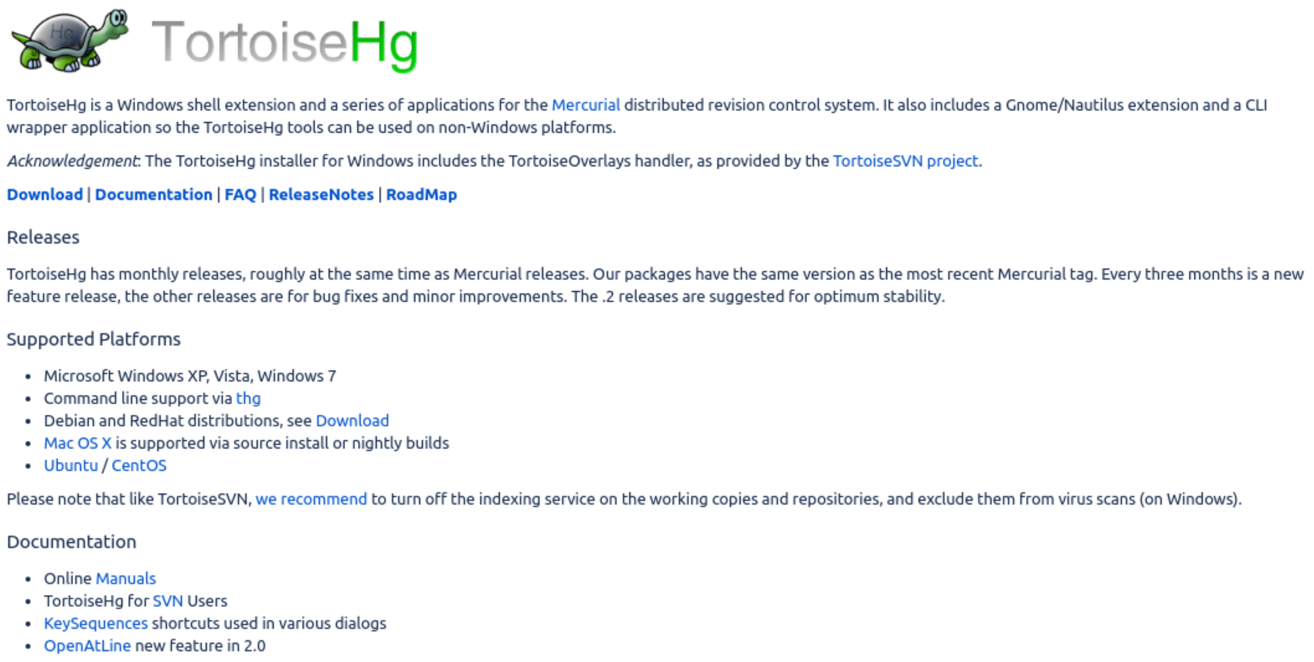
Wikis

Bitbucket service[in form of git repo] when you make a project you get a wiki, it's considered as a project homepage, this homepage you can edit, clone it such as any other files in your project repo
example of wikis homepage



TortoiseHg is a Windows shell extension and a series of applications for the Mercurial distributed revision control system. It also includes a Gnome/Nautilus extension and a CLI wrapper application so the TortoiseHg tools can be used on non-Windows platforms.

Acknowledgement: The TortoiseHg installer for Windows includes the TortoiseOverlays handler, as provided by the TortoiseSVN project.

Download | **Documentation** | **FAQ** | **ReleaseNotes** | **RoadMap**

Releases

TortoiseHg has monthly releases, roughly at the same time as Mercurial releases. Our packages have the same version as the most recent Mercurial tag. Every three months is a new feature release, the other releases are for bug fixes and minor improvements. The .2 releases are suggested for optimum stability.

Supported Platforms

- Microsoft Windows XP, Vista, Windows 7
- Command line support via thg
- Debian and RedHat distributions, see Download
- Mac OS X is supported via source install or nightly builds
- Ubuntu / CentOS

Please note that like TortoiseSVN, we recommend to turn off the indexing service on the working copies and repositories, and exclude them from virus scans (on Windows).

Documentation

- Online Manuals
- TortoiseHg for SVN Users
- KeySequences shortcuts used in various dialogs
- OpenAtLine new feature in 2.0

supported markup lang creole, markdown, etc..
there's bitbuckt special markup that could be used also, in some case bitbucket macros can be used also to do the same job
I think that best markup is markdown lang. Popular, used in most git remote repos
example of wiki the README.md [md stands for markdown] file

YOU CAN FIND TUTORIAL ON Markdown Language Syntax HERE: *https://www.markdowntutorial.com/*

takes about 30 minutes to try all features you can use
also there's a cheat sheet to use it whenever you need

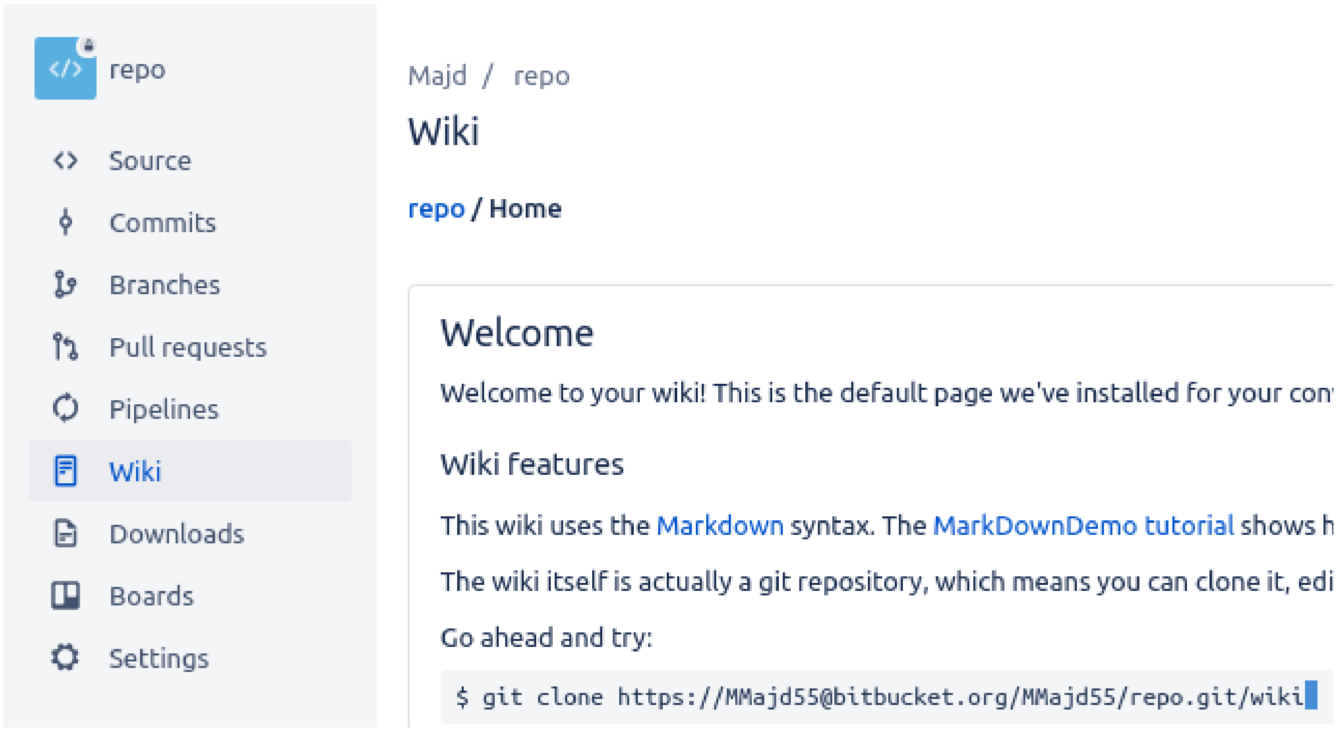
What wikis could be used for:

-Online Documentation
-Used to Manage Weekly Work and Team Member Work [In Hompage]
-Multiply Wikis Could Be used To Document Project History E.G. What Have Been Added To There Project In Which Week and Whose Done it [Would be easier than searching the could]
-Previous History Sheet Could be used to Grade Each Team Member Effort and Productivity
-Used to Make Announcement to Team Members
-Used to Track What Has Been Done on The Project
-Used to Embedded Codes To Open Discussion About Them Without Adding Them to The Repo

---

Enable a Wiki

From the repository, click **Settings**.
Click **Wiki** in the settings navigation.
Select Public Wiki or Private Wiki from the Wiki settings page. A private Wiki is only visible to people who have permission to see it. Anyone can view, edit, or create pages for a public Wiki.
Click Save.
You'll now see the Wiki link in the left panel. When you click it, the Wiki welcome Home page appears by default.



Welcome

Welcome to your wiki! This is the default page we've installed for your con

Wiki features

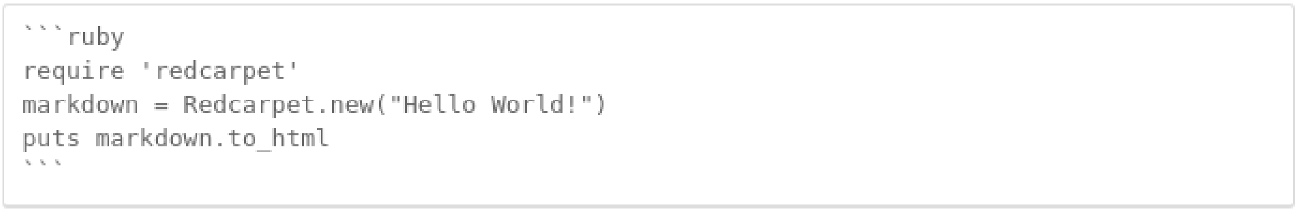
This wiki uses the Markdown syntax. The MarkDownDemo tutorial shows h

The wiki itself is actually a git repository, which means you can clone it, edi

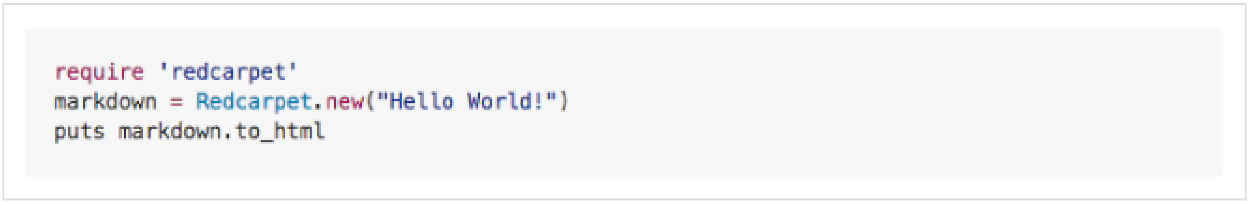
Go ahead and try:

```
$ git clone https://MMajd55@bitbucket.org/MMajd55/repo.git/wiki
```

Notice that we could embedded could into wikis files with highlights such as



~~~
```ruby
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```
~~~



```
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
```

Github link that contains basic writing and formatting syntax of markdown

https://help.github.com/articles/basic-writing-and-formatting-syntax/

MARKDOWN SYNTAX

Markdown is a way to style text on the web. You control the display of the document; formatting words as bold or italic, adding images, and creating lists are just a few of the things we can do with Markdown. Mostly, Markdown is just regular text with a few non-alphabetic characters thrown in, like # or *.

HEADERS

```
# This is an <h1> tag
## This is an <h2> tag
###### This is an <h6> tag
```

EMPHASIS

```
*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

LISTS

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
2. Item 2
3. Item 3
   * Item 3a
   * Item 3b
```

IMAGES

```
![GitHub Logo](/images/logo.png)

Format: ![Alt Text](url)
```

LINKS

```
http://github.com - automatic!

[GitHub](http://github.com)
```

BLOCKQUOTES

```
As Grace Hopper said:

> I've always been more interested
> in the future than in the past.
```

As Grace Hopper said:

> I've always been more interested
> in the future than in the past.

BACKSLASH ESCAPES

Markdown allows you to use backslash escapes to generate literal characters which would otherwise have special meaning in Markdown's formatting syntax.

```
\*literal asterisks\*
```

literal asterisks

Markdown provides backslash escapes for the following characters:

\	backslash	()	parentheses
`	backtick	#	hash mark
*	asterisk	+	plus sign
_	underscore	-	minus sign (hyphen)
{}	curly braces	.	dot
[]	square brackets	!	exclamation mark

GITHUB FLAVORED MARKDOWN

GitHub.com uses its own version of the Markdown syntax, GFM, that provides an additional set of useful features, many of which make it easier to work with content on GitHub.com.

USERNAME @MENTIONS

Typing an `@` symbol, followed by a username, will notify that person to come and view the comment. This is called an "@mention", because you're mentioning the individual. You can also @mention teams within an organization.

ISSUE REFERENCES

Any number that refers to an Issue or Pull Request will be automatically converted into a link.

```
#1
github-flavored-markdown#1
defunkt/github-flavored-markdown#1
```

EMOJI

To see a list of every image we support, check out **www.emoji-cheat-sheet.com**

```
GitHub supports emoji!
:+1: :sparkles: :camel: :tada:
:rocket: :metal: :octocat:
```

GitHub supports emoji!
👍 ✨ 🐫 🎉 🚀 🤘

FENCED CODE BLOCKS

Markdown coverts text with four leading spaces into a code block; with GFM you can wrap your code with ` ``` ` to create a code block without the leading spaces. Add an optional language identifier and your code will get syntax highlighting.

~~~
```javascript
function test() {
 console.log("look ma', no spaces");
}
```
~~~

```javascript
function test() {
  console.log("look ma', no spaces");
}
```

TASK LISTS

```
- [x] this is a complete item
- [ ] this is an incomplete item
- [x] @mentions, #refs, [links](),
**formatting**, and <del>tags</del>
supported
- [x] list syntax required (any
unordered or ordered list
supported)
```

- [x] this is a complete item
- [ ] this is an incomplete item
- [x] @mentions, #refs, links, **formatting**, and ~~tags~~ supported
- [x] list syntax required (any unordered or ordered list supported)

TABLES

You can create tables by assembling a list of words and dividing them with hyphens `-` (for the first row), and then separating each column with a pipe `|`:

```
First Header | Second Header
------------ | -------------
Content cell 1 | Content cell 2
Content column 1 | Content column 2
```

First Header	Second Header
Content cell 1	Content cell 2
Content column 1	Content column 2

JAVA - DOCUMENTATION COMMENTS

The Java language supports three types of comments −

Sr.No.	Comment & Description
1	**/* text */** The compiler ignores everything from /* to */.
2	**//text** The compiler ignores everything from // to the end of the line.
3	**/** documentation */** This is a documentation comment and in general its called **doc comment**. The **JDK javadoc** tool uses *doc comments* when preparing automatically generated documentation.

This chapter is all about explaining Javadoc. We will see how we can make use of Javadoc to generate useful documentation for Java code.

What is Javadoc?

Javadoc is a tool which comes with JDK and it is used for generating Java code documentation in HTML format from Java source code, which requires documentation in a predefined format.

Following is a simple example where the lines inside /*….*/ are Java multi-line comments. Similarly, the line which preceeds // is Java single-line comment.

Example

```
/**
* The HelloWorld program implements an application that
* simply displays "Hello World!" to the standard output.
*
* @author  Zara Ali
* @version 1.0
* @since   2014-03-31
*/
public class HelloWorld {

   public static void main(String[] args) {
      /* Prints Hello, World! on standard output.
      System.out.println("Hello World!");
   }
}
```

You can include required HTML tags inside the description part. For instance, the following example makes use of <h1>....</h1> for heading and <p> has been used for creating paragraph break −

Example

```
/**
* <h1>Hello, World!</h1>
* The HelloWorld program implements an application that
* simply displays "Hello World!" to the standard output.
* <p>
* Giving proper comments in your program makes it more
* user friendly and it is assumed as a high quality code.
*
*
* @author  Zara Ali
* @version 1.0
* @since   2014-03-31
*/
public class HelloWorld {

   public static void main(String[] args) {
      /* Prints Hello, World! on standard output.
      System.out.println("Hello World!");
   }
}
```

The javadoc Tags

The javadoc tool recognizes the following tags −

Tag	Description	Syntax
@author	Adds the author of a class.	@author name-text
{@code}	Displays text in code font without interpreting the text as HTML markup or nested javadoc tags.	{@code text}
{@docRoot}	Represents the relative path to the generated document's root directory from any generated page.	{@docRoot}
@deprecated	Adds a comment indicating that this API should no longer be used.	@deprecated deprecatedtext
@exception	Adds a **Throws** subheading to the generated documentation, with the classname and description text.	@exception class-name description
{@inheritDoc}	Inherits a comment from the **nearest** inheritable class or implementable interface.	Inherits a comment from the immediate surperclass.

{@link}	Inserts an in-line link with the visible text label that points to the documentation for the specified package, class, or member name of a referenced class.	{@link package.class#member label}
{@linkplain}	Identical to {@link}, except the link's label is displayed in plain text than code font.	{@linkplain package.class#member label}
@param	Adds a parameter with the specified parameter-name followed by the specified description to the "Parameters" section.	@param parameter-name description
@return	Adds a "Returns" section with the description text.	@return description
@see	Adds a "See Also" heading with a link or text entry that points to reference.	@see reference
@serial	Used in the doc comment for a default serializable field.	@serial field-description | include | exclude
@serialData	Documents the data written by the writeObject or writeExternal methods.	@serialData data-description
@serialField	Documents an ObjectStreamField component.	@serialField field-name field-type field-description
@since	Adds a "Since" heading with the specified since-text to the generated documentation.	@since release
@throws	The @throws and @exception tags are synonyms.	@throws class-name description
{@value}	When {@value} is used in the doc comment of a static field, it displays the value of that constant.	{@value package.class#field}
@version	Adds a "Version" subheading with the specified version-text to the generated docs when the -version option is used.	@version version-text

Example

Following program uses few of the important tags available for documentation comments. You can make use of other tags based on your requirements.

The documentation about the AddNum class will be produced in HTML file AddNum.html but at the same time a master file with a name index.html will also be created.

```
import java.io.*;

/**
```

```
* <h1>Add Two Numbers!</h1>
* The AddNum program implements an application that
* simply adds two given integer numbers and Prints
* the output on the screen.
* <p>
* <b>Note:</b> Giving proper comments in your program makes it more
* user friendly and it is assumed as a high quality code.
*
* @author  Zara Ali
* @version 1.0
* @since   2014-03-31
*/
public class AddNum {
   /**
   * This method is used to add two integers. This is
   * a the simplest form of a class method, just to
   * show the usage of various javadoc Tags.
   * @param numA This is the first paramter to addNum method
   * @param numB  This is the second parameter to addNum method
   * @return int This returns sum of numA and numB.
   */
   public int addNum(int numA, int numB) {
      return numA + numB;
   }

   /**
   * This is the main method which makes use of addNum method.
   * @param args Unused.
   * @return Nothing.
   * @exception IOException On input error.
   * @see IOException
   */

   public static void main(String args[]) throws IOException {
      AddNum obj = new AddNum();
      int sum = obj.addNum(10, 20);

      System.out.println("Sum of 10 and 20 is :" + sum);
   }
}
```

Now, process the above AddNum.java file using javadoc utility as follows −

```
$ javadoc AddNum.java
Loading source file AddNum.java...
Constructing Javadoc information...
Standard Doclet version 1.7.0_51
Building tree for all the packages and classes...
Generating /AddNum.html...
AddNum.java:36: warning - @return tag cannot be used in method with void return type.
Generating /package-frame.html...
Generating /package-summary.html...
Generating /package-tree.html...
Generating /constant-values.html...
Building index for all the packages and classes...
Generating /overview-tree.html...
Generating /index-all.html...
Generating /deprecated-list.html...
Building index for all classes...
Generating /allclasses-frame.html...
Generating /allclasses-noframe.html...
Generating /index.html...
Generating /help-doc.html...
1 warning
$
```

You can check all the generated documentation here − AddNum. If you are using JDK 1.7 then javadoc does not generate a great **stylesheet.css**, so we suggest to download and use standard stylesheet from https://docs.oracle.com/javase/7/docs/api/stylesheet.css