Practical Deployment with Git and Ant
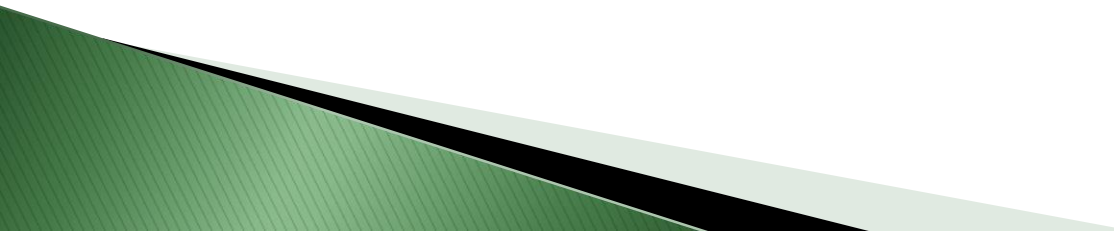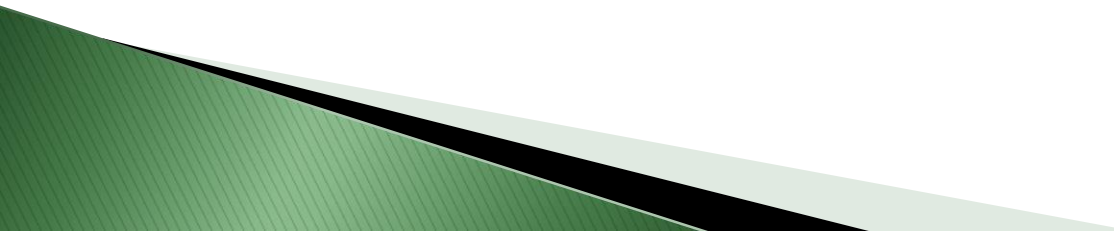
Anant Pradhan
@blitheutopian

What is Git?

Distributed revision control and source code management (SCM) system with an emphasis on speed.
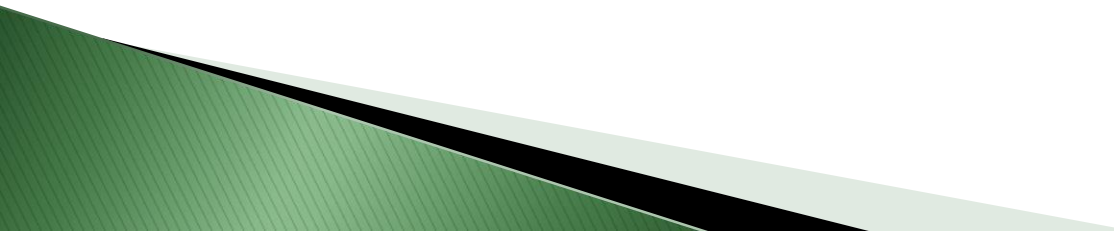
Features

- Distributed development

- Strong support for non-linear development

- Compatibility with existing systems/protocols

- Efficient handling of large projects

- Cryptographic authentication of history

Git's Major Benefits Over Subversion

- Distributed Nature

- Access Control

- Branch Handling

- Performance (Speed of Operation)

- Smaller Space Requirements

Subversion's Major Benefits Over Git

- Mature User Interfaces

- Single Repository

- Partial Checkout/Bandwidth Requirements

- Shorter and Predictable Revision Numbers

- Less restrictive (in some ways)

Git Demo - Committing

- Commands for committing and reverting files
  - git add .
    - adds file contents of all uncommited files to your staging area before you can commit them)
  - git commit -m '[commit message]'
    - commits the file with the commit message specified
  - git status
    - shows you the current status of the repository
  - git revert HEAD
    - reverts the last commit

Git Demo - Branching

- Commands for branching and merging files
  - git branch (lists all available branches)
  - git branch [branchname] (creates a new branch)
  - git checkout [branchname] (switches to that branch)
  - git branch -d [branchname] (deletes a branch)
  - git merge [branchname] (merges branch with current branch)
 - git log --oneline (shows brief commit history of a branch)

Git Demo – Pull/Push

- Commands for pulling/pushing files to/from remote repository
  - Default remote-name: 'origin'
  - git fetch [remote-name]

    fetches any new work that has been pushed to that server since you cloned
  - git pull [remote-name] [branchname]

    merges remote branch with local branch
  - git push [remote-name] [branchname]
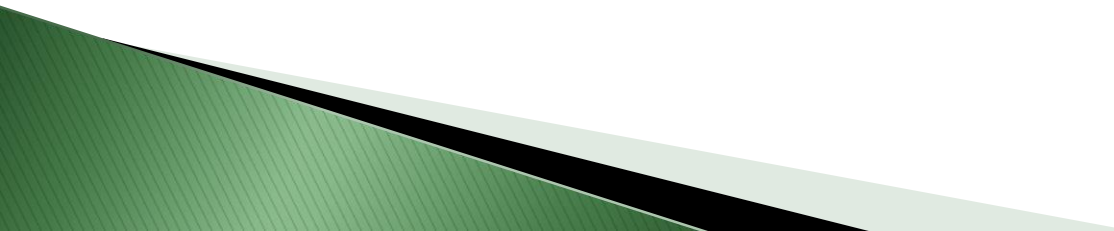
    merges local branch with remote branch
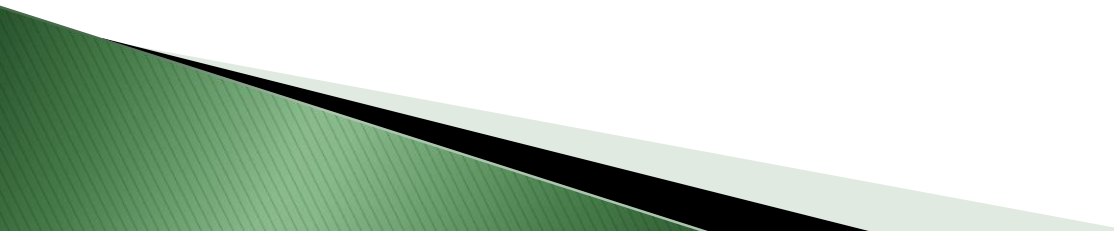
What is Ant?

"Another Neat Tool"

Java based build tool for automating software build processes. Open Source Apache project.
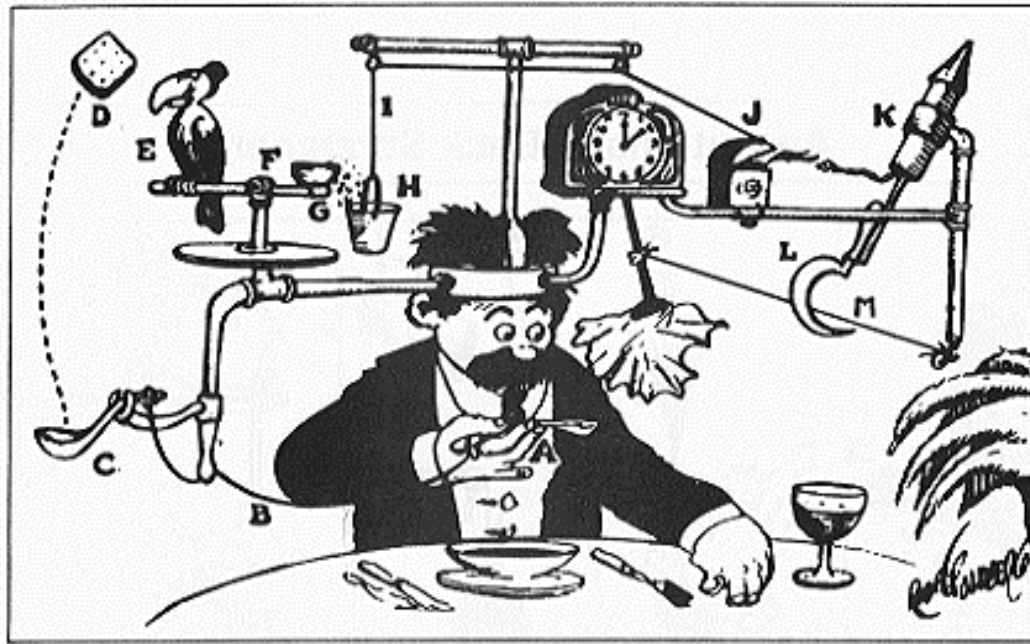
Features

- Platform neutral
- Can perform platform specific tasks
- Written using plain XML
- Big list of predefined tasks
- Interface to develop custom tasks
- Integration with free and commercial IDEs
- Great at automating complicated repetitive tasks

Limitations

- Not a scripting language

- Fairly verbose

- Large projects can get unmanageable

- Limited fault handling and no state persistence

Ant Syntax

▸ build.xml

▸ Target (task or function)

<**target** name="helloWorld" **depends**="helloHumans" >

  <echo> Hello World! </echo>

</target>

▸ Property (variable)

<**property** name="test_dir" value="tests" />

<**property** name=" test_dir"> tests </property>

Ant Syntax

- Antcall (function call)

  <**antcall** target="compress_css" />

- Macrodef (macro definition)

  <**macrodef** name="git">
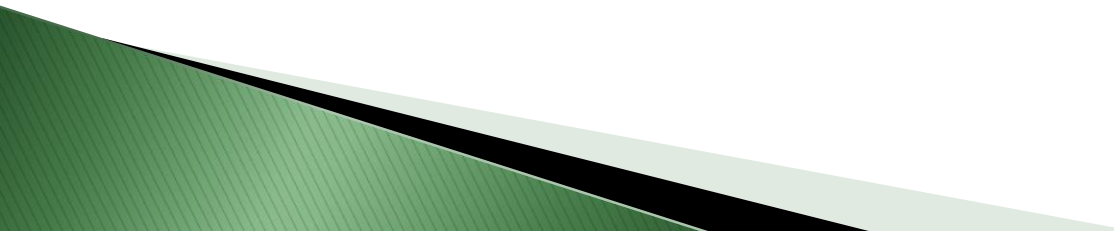
- Exec (system call execution)

  <**exec** executable = "git" dir = "@{dir}">

  </exec>

Ant Demo

- Setting up and running Ant "Hello World"

- Combining CSS/JS files

- Compressing CSS/JS files

Deployment Demo

References

- http://en.wikipedia.org/wiki/Git_(software)

- https://help.github.com/articles/set-up-git

- http://gitref.org/

- http://davidwalsh.name/git-commands

- https://git.wiki.kernel.org/index.php/GitSvnComparison

- http://anant5.com/2013/02/getting_started_with_git/

- http://ant.apache.org/manual/

- http://en.wikipedia.org/wiki/Apache_Ant

- http://www.tutorialspoint.com/ant/ant_quick_guide.htm

- http://net.tutsplus.com/tutorials/other/automate-your-projects-with-apache-ant/

Thank you

Questions?

Contact Info:
- anant@anant5.com
- http://anant5.com
- @blitheutopian