

29/Aug/2022

Object Oriented Programming

Student number: S2212942

Email: S2212942@bi.no

1.

→ ~ git:(master) X git clone <https://github.com/BI-DS/GRA-4152.git>

```
Cloning into 'GRA-4152'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 32 (delta 8), reused 27 (delta 3), pack-reused 0
Receiving objects: 100% (32/32), 4.43 KiB | 2.21 MiB/s, done.
Resolving deltas: 100% (8/8), done.
```

→ ~ git:(master) X cd GRA-4152

a.

→ GRA-4152 git:(master) git log --graph --decorate

```
* commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498 (HEAD -> master, origin/master, origin/HEAD)
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Wed Aug 31 09:59:14 2022 +0200

    added honor code

* commit a610fc6f9cdf3dd3e84e42e8e6ac168a73ab3a28
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Mon Aug 29 09:33:29 2022 +0200

    adding 1 async exercise from lecture 1

* commit 0f6036be424d789df67656a3be50cec8848f2d4f
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Mon Aug 29 09:31:42 2022 +0200

    adding 1 async exercise from lecture 1

* commit ae45ae7891f40dab569153b9decaaedc71601ff5
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Mon Aug 29 08:51:21 2022 +0200

    removing git_exercise.py file

* commit 84ed53d35856ac4c6a59a8de7853e2d723b44a39
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Fri Aug 26 08:56:46 2022 +0200

    adding file for git exercise in lecture 2

* commit 37ac00f151ef14e398bf6ab3c3078a799d974775
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Mon Aug 22 11:57:28 2022 +0200

    adding some py files for lecture 1

* commit 0fb7842d8311144c4d1941b8e9d828059e11c500
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Mon Aug 22 08:50:07 2022 +0200

    adding instructions for UML

* commit 20b88515dc1668bed7942359e3ad183f51961f62
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date: Fri Aug 19 08:48:04 2022 +0200

    initial commit

~
~
~
~
[END]
```

b. & c.

→ GRA-4152 git:(master) git log -p -1 --pretty=format:"%h - %an, %ar : %s"

```
71f261f - rogelioandrade, 3 days ago : added honor code
diff --git a/README.md b/README.md
index a7359ae..a404da3 100644
--- a/README.md
+++ b/README.md
@@ -0,3 +0,7 @@
 pip install pylint
 sudo apt install graphviz
 pyreverse -o png <your code>.py

+
+## Honor Code
+You are free to form study groups and may discuss homework in groups. However, each student must write down the solutions and code from scratch independently and must understand the solution well enough. It is a honor code violation to copy, refer to, or look at written or code solutions from a previous year or solutions posted online (inspired by the Stanford Honor Code).
+
+ (END)
```

2.

Firstly, file .gitignore is created in my repository:

→ GRA4152 git:(main) X touch .gitignore

Secondly, foo.py is also created by TextEdit.

Before the foo.py is added to .gitignore, the state of the working directory is displayed as:

→ GRA4152 git:(main) X git status

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        foo.py
```

The foo.py is added to the .gitignore, the state of the working directory is displayed as:

→ GRA4152 git:(main) X echo "/foo.py" >> .gitignore

→ GRA4152 git:(main) X git status

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
```

The .gitignore file allows ignoring a file committed in the past.

3.

Cloning repository from Git hub:

~ git:(master) X git clone <https://github.com/Macjutsu/super.git>

~ git:(master) X cd super

super git:(main) ls

```
CHANGELOG.md Example-Logs Example-MDM Images LICENSE README.md super
```

Modifying the README.md:

→ super git:(main) cat README.md

```
# S.U.P.E.R.M.A.N.

### Software Update Policy Enforcement (with) Recursive Messaging And Notification

S.U.P.E.R.M.A.N. optimizes the macOS software update experience.

by Kevin M. White

## Introduction

S.U.P.E.R.M.A.N. (or just `super`) is an open source script that provides administrators with a comprehensive workflow to encourage and enforce macOS software updates for both Intel and Apple silicon Mac computers. Deployed using a single script, `super` creates a background agent (aka LaunchDaemon) that ensures software updates are applied with the least user interference possible. Further, `super` can also enforce software updates with options for customizable deferrals and deadlines. In other words, `super` makes the macOS update experience better for both users and administrators.

## Learn More

Please visit the [S.U.P.E.R.M.A.N. Wiki](https://github.com/Macjutsu/super/wiki) for detailed documentation!

Detailed update information can be found on both the [Change Log](https://github.com/Macjutsu/super/blob/main/CHANGELOG.md) and [Releases](https://github.com/Macjutsu/super/releases) pages.

You can also join the conversation at the [Mac Admins Foundation Slack](https://www.macadmins.org) in channel [#super](https://macadmins.slack.com/archives/C03LKQ8EN2C).
```

→ super git:(main) echo " I have a dream to be a superman." >> README.md

→ super git:(main) X cat README.md

```
# S.U.P.E.R.M.A.N.

### Software Update Policy Enforcement (with) Recursive Messaging And Notification

S.U.P.E.R.M.A.N. optimizes the macOS software update experience.

by Kevin M. White

## Introduction

S.U.P.E.R.M.A.N. (or just `super`) is an open source script that provides administrators with a comprehensive workflow to encourage and enforce macOS software updates for both Intel and Apple silicon Mac computers. Deployed using a single script, `super` creates a background agent (aka LaunchDaemon) that ensures software updates are applied with the least user interference possible. Further, `super` can also enforce software updates with options for customizable deferrals and deadlines. In other words, `super` makes the macOS update experience better for both users and administrators.

## Learn More

Please visit the [S.U.P.E.R.M.A.N. Wiki](https://github.com/Macjutsu/super/wiki) for detailed documentation!

Detailed update information can be found on both the [Change Log](https://github.com/Macjutsu/super/blob/main/CHANGELOG.md) and [Releases](https://github.com/Macjutsu/super/releases) pages.

You can also join the conversation at the [Mac Admins Foundation Slack](https://www.macadmins.org) in channel [#super](https://macadmins.slack.com/archives/C03LKQ8EN2C).
I have a dream to be a superman.
```

When I type git stash, it shows that:

```
→ super git:(main) X git stash
Saved working directory and index state WIP on main: 1b32831 Update README.md
```

Git log --all --oneline displays the result:

```
05cd8b3 (refs/stash) WIP on main: 1b32831 Update README.md
9919f73 index on main: 1b32831 Update README.md
1b32831 (HEAD -> main, origin/main, origin/HEAD) Update README.md
53af0c5 Update Super-Service-Account-Jamf-Pro-PPPC.mobileconfig
8db60ab (tag: v2.0) 2.0
ad1495a (origin/2.0) 2.0
c9973ff (tag: v2.0-rc1, origin/2.0rc1) 2.0rc1
14c1317 Log Examples
a2883c5 Typo in username handle.
4af07b2 (tag: v2.0-b2, origin/2.0b2) 2.0b2
ce24d27 Update README.md
f543d01 (tag: v2.0-b1) Merge branch '2.0b1'
5275c20 (origin/2.0b1) 2.0b1
e4df654 (origin/1.1.1) 1.1.1
a99cbc2 (tag: v1.1, origin/1.1) 1.1
5ab69f4 (origin/1.1b2) 1.1b2
9a0c040 (origin/1.1b1) 1.1b1
f2ff76f Preference Example Update
2634d20 (tag: v1.0-rc1) Merge pull request #3 from Macjutsu/1.0rc1
c67af0e (origin/1.0rc1) Update README.md
4c8f610 1.0rc1
684d360 1.0rc1
322e1e9 Update com.macjutsu.super.plist
7138243 1.0rc1
8712333 1.0rc1
359486c 1.0rc1
115f4f3 1.0rc1
d8f17b9 1.0rc1
039e9ab 1.0rc1
cc44538 1.0rc1
5a920c7 Merge pull request #2 from Macjutsu/1.0b
75c2560 Create com.macjutsu.super.plist
66c0931 Update super
9222154 Merge pull request #1 from Macjutsu/0.9
ce8e454 0.9
8ef9f4b Initial commit
(END)
```

The stash change is shown at the top of the result.

After that, I run git stash pop to undo what I did with git stash.

Git stash saves the changes that are not committed. This way is effective when working with branches, or on large projects containing enormous coding lines. Git stash saves the uncommitted changes locally, allowing you to make changes, switch branches, and perform other Git operations.

Besides, git stash pop is used to restore unfinished working. Git stash pop will drop the stash if there is no merge conflict.

I have not found any meanings when apply that the senario of using git stash first, then using git stash pop immediatly.

→ super git:(main) git stash list

→ super git:(main) git stash drop stash@{0}

```
stash@{0}: WIP on main: 1b32831 Update README.md
(END)
```

Now, modify a file and stash changes. Make a new modification to the same file, but this time commit those changes.

→ super git:(main) X git stash pop

→ super git:(main) X cat README.md

```
➤ super git:(main) X git stash pop
No stash entries found.
➤ super git:(main) X cat README.md
# S.U.P.E.R.M.A.N.

### Software Update Policy Enforcement (with) Recursive Messaging And Notification

S.U.P.E.R.M.A.N. optimizes the macOS software update experience.

by Kevin M. White

## Introduction

S.U.P.E.R.M.A.N. (or just `super`) is an open source script that provides administrators with a comprehensive workflow to encourage and enforce macOS software updates for both Intel and Apple silicon Mac computers. Deployed using a single script, `super` creates a background agent (aka LaunchDaemon) that ensures software updates are applied with the least user interference possible. Further, `super` can also enforce software updates with options for customizable deferrals and deadlines. In other words, `super` makes the macOS update experience better for both users and administrators.

## Learn More

Please visit the [S.U.P.E.R.M.A.N. Wiki](https://github.com/Macjutsu/super/wiki) for detailed documentation!

Detailed update information can be found on both the [Change Log](https://github.com/Macjutsu/super/blob/main/CHANGELOG.md) and [Releases](https://github.com/Macjutsu/super/releases) pages.

You can also join the conversation at the [Mac Admins Foundation Slack](https://www.macadmins.org) in channel [#super](https://macadmins.slack.com/archives/C03LKQ8EN2C).

not every one can be a superman.
```

After commit changes, use the git stash pop, then open the README.md, the last change is disappeared in README.md

4.

Firstly, To create a new branch “ my_test_branch” in my class_repository:

→ GRA4152 git:(main) git branch "my_test_branch"

Secondly, To Check branches in my repository:

→ GRA4152 git:(main) git branch

Thirdly, To switch between branches:

→ GRA4152 git:(main) git switch my_test_branch

Fourthly, To comment line in README.md, then add and commit changes in the branch my_test_branch:

→ GRA4152 git:(my_test_branch) cat README.md

```
# GRA4152
# This repository is used for course " Object Oriented Programming with Python."
```

→ GRA4152 git:(my_test_branch) echo "We have a deadline on Sunday Sep 4th" >> README.md

- GRA4152 git:(my_test_branch) X git add README.md
- GRA4152 git:(my_test_branch) X git commit -m "Commit changes in README.md "

Finally, to merge my_test_branch into master:

- GRA4152 git:(master) git checkout main
- GRA4152 git:(main) git merge my_test_branch

5.

Fork the class repository, and clone it:

~ git:(master) X git clone <https://github.com/S2212942/GRA-4152.git>

Add a text file and send the pull request to the class repository:

- GRA-4152 git:(master) cat > S2212942.txt
- GRA-4152 git:(master) X git add .
- GRA-4152 git:(master) X git commit -m "add a text file S2212942"
- GRA-4152 git:(master) git push

Finally, the pull request is delivered to the class repository:

