

LINKSFOUNDATION.COM

Applied Data Science Project

L10 – Version Control

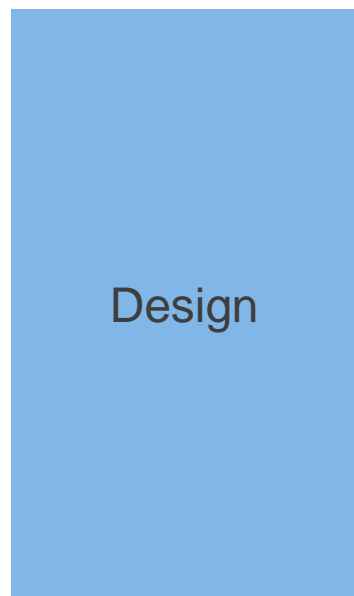


**Politecnico
di Torino**

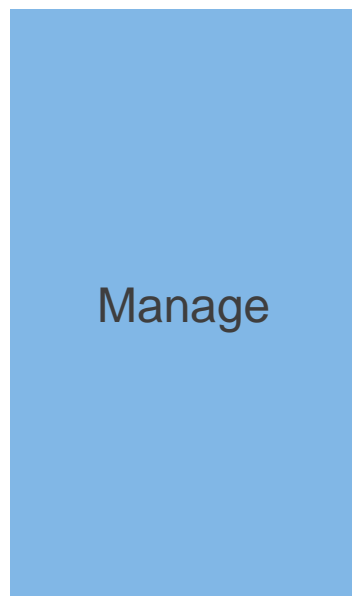


e l i s
European Laboratory for Learning and Intelligent Systems

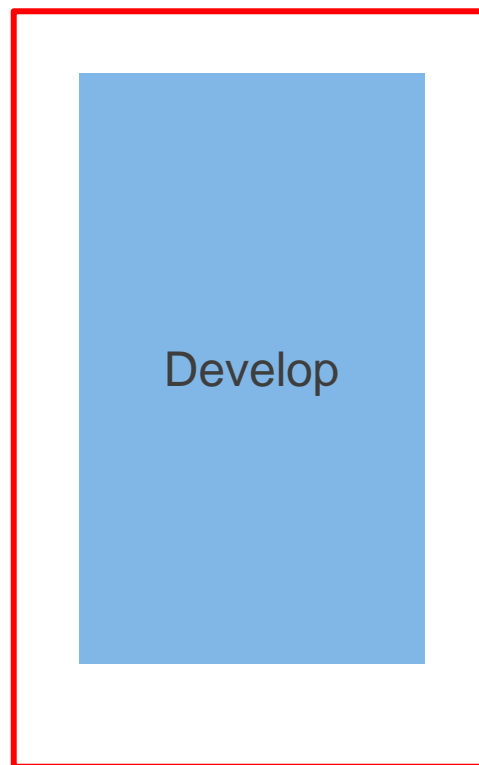
Pillars



Design



Manage



Develop



Communicate

Version control

- History of the development
 - what has been done 2 days ago
 - what a team member contributed to
 - restore a past version that resulted to be more robust than the last one
- Shared space for collaborators
- Monitor development branches and derive forks to be utilized for spinoff projects
- Package the development into tags (releases) that answer project milestones

Version control



<https://colab.research.google.com>

Colaboratory natively stores different development versions each labeled either automatically by Colaboratory or defined by the team manually



<https://github.com>

An application that manages local repositories and synchronises with remote repositories utilizing the git protocol. It also offers an intuitive web dashboard for the supervision of the project and analytics

Two different needs



<https://colab.research.google.com>

When the project is about **one notebook**, this is the favorite option in an initial phase (lean option)



<https://github.com>

When the project grows in terms of **files and modules** that cannot stay in just a notebook

Version control



Simple Sentiment Analysis.ipynb

File Edit View Insert Runtime Tools Help



List of differences

← Revision history

☐ Raw source ☐ Inline diff ☒ Show output

```
1 ## Preparing Data
2
3 One of the main concepts of TorchText is the `Field`. These define ho
4
5 The parameters of a `Field` specify how the data should be processed.
6
7 We use the `TEXT` field to define how the review should be processed,
8
9 Our `TEXT` field has `tokenize='spacy'` as an argument. This defines
10
11 `LABEL` is defined by a `LabelField`, a special subset of the `Field`
12
13 For more on `Fields`, go [here](https://github.com/pytorch/text/blob/
14
15 We also set the random seeds for reproducibility.
```

Code cell <undefined>

[code]

```
1 import torch
2 from torchtext import data
3
4 SEED = 1234
5
6 torch.manual_seed(SEED)
7 torch.backends.cudnn.deterministic = True
8
9 TEXT =
10
11 LABEL = data.Label
```

Text cell <undefined>

[markdown]

```
1 Another handy feature of TorchText is that it has support for common
2
3 The following code automatically downloads the IMDB dataset and split
```

Code cell <undefined>

[code]

```
1 from torchtext import datasets
```

```
1 ## Preparing Data
2
3 One of the main concepts of TorchText is the `Field`. These define ho
4
5 The parameters of a `Field` specify how the data should be processed.
6
7 We use the `TEXT` field to define how the review should be processed,
8
9 Our `TEXT` field has `tokenize='spacy'` as an argument. This defines
10
11 `LABEL` is defined by a `LabelField`, a special subset of the `Field`
12
13 For more on `Fields`, go [here](https://github.com/pytorch/text/blob/
14
15 We also set the random seeds for reproducibility.
```

Code cell <undefined>

[code]

```
1 import torch
2 from torchtext.legacy import data
3
4 SEED = 1234
5
6 torch.manual_seed(SEED)
7 torch.backends.cudnn.deterministic = True
```

Text cell <undefined>

[markdown]

```
1 Another handy feature of TorchText is that it has support for common
2
3 The following code automatically downloads the IMDB dataset and split
```

Code cell <undefined>

[code]

```
1 from torchtext.legacy import datasets
```

List of differences between the versions indicated in the right panel with the checkboxes (left vs right)

<input type="radio"/>	<input checked="" type="radio"/>	Mar 12, 2021 2:25 PM bentrevett update to torchtext 0.9	⋮
<input checked="" type="radio"/>	<input type="radio"/>	Feb 17, 2021 1:52 PM bentrevett updated tutorials + readme with latest versions of libs.	⋮
<input type="radio"/>	<input type="radio"/>	Sep 16, 2019 12:32 PM bentrevett reran all notebooks with latest pytorch and torchtext to ensure still working	⋮
<input type="radio"/>	<input type="radio"/>	Apr 10, 2019 11:27 AM bentrevett added model.eval() in predict sentiment functions (#31)	⋮
<input type="radio"/>	<input type="radio"/>	Apr 1, 2019 5:08 PM bentrevett mentioned how notebook 2 will introduce packed padded sequences	⋮
<input type="radio"/>	<input type="radio"/>	Mar 29, 2019 5:57 PM bentrevett lots of formatting changes	⋮
<input type="radio"/>	<input type="radio"/>	Mar 21, 2019 11:48 PM bentrevett added parameter count and epoch timer functions to all notebooks. also added ...	⋮
<input type="radio"/>	<input type="radio"/>	Mar 21, 2019 6:22 PM bentrevett fixed out of glove vector initialization and missing generate bigrams functio...	⋮
<input type="radio"/>	<input type="radio"/>	Mar 10, 2019 3:45 PM bentrevett changed imgur urls to own assets	⋮

on.com

Version control



Simple Sentiment Analysis.ipynb

File

Edit

View

Insert

Runtime

Tools

Help



← Revision history

☐ Raw source ☐ Inline diff ☒ Show output

Pinned version
Simple Sentiment Analysis.ipynb

Text cell <mkroypBnSzZ2>

[markdown]

1 # Simple Sentiment Analysis

2

3 In this series we'll be building a machine

4

5 In this first notebook, we'll start very s

6

7 ### Introduction

8

9 We'll be using a **recurrent neural networ

10

11 $h_t = \text{text}\{\text{RNN}\}(x_t, h_{t-1})$

12

13 Once we have our final hidden state, h_T

14

15 Below shows an example sentence, with the

16

17 [!\[\]\(https://github.com/bentrevett/pytorch-](https://github.com/bentrevett/pytorch-)

18

19 **Note:** some layers and steps have been

Text cell <CwTZS1Y4SzZ8>

[markdown]

1 ## Preparing Data

2

3 One of the main concepts of TorchText is t

4

5 The parameters of a `Field` specify how t

6

7 We use the `TEXT` field to define how the

Tue Oct 05 2021 10:06:03 GMT+0200 (Central European S
Simple Sentiment Analysis.ipynb

Text cell <mkroypBnSzZ2>

[markdown]

1 # Simple Sentiment Analysis

2

3 In this series we'll be building a machine

4

5 In this first notebook, we'll start very s

6

7 ### Introduction

8

9 We'll be using a **recurrent neural networ

10

11 $h_t = \text{text}\{\text{RNN}\}(x_t, h_{t-1})$

12

13 Once we have our final hidden state, h_T

14

15 Below shows an example sentence, with the

16

17 [!\[\]\(https://github.com/bentrevett/pytorch-](https://github.com/bentrevett/pytorch-)

18

19 **Note:** some layers and steps have been

Text cell <CwTZS1Y4SzZ8>

[markdown]

1 ## Preparing Data

2

3 One of the main concepts of TorchText is t

4

5 The parameters of a `Field` specify how t

6

7 We use the `TEXT` field to define how the

Version pinning

☐ Only show named versions

Oct 5, 2021 10:06 AM
Giuseppe Rizzo

Pinned version
Oct 5, 2021 9:38 AM
Giuseppe Rizzo

Pinned version
Oct 5, 2021 9:38 AM
Giuseppe Rizzo

Version control



[adsp-polito](#) / [adsp-polito.github.io](#) Public

Unwatch 1 Star 0 Fork 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

main 1 branch 0 tags Go to file Add file Code

giusepperizzo Update README.md ✓ e05b240 4 days ago 🕒 17 commits

L1 - ADSP - Intro.pdf	Add files via upload	5 days ago
L2 - ADSP - Model & Data-centric pr...	Add files via upload	5 days ago
L3 - ADSP - Foundation models.pdf	Add files via upload	5 days ago
L4 - ADSP - SGDs and data science ...	Add files via upload	5 days ago
L5 - ADSP - Pillars.pdf	Add files via upload	5 days ago
L6 - ADSP - 10 practical tips.pdf	Add files via upload	5 days ago
L7 - ADSP - Project design tools.pdf	Add files via upload	4 days ago
README.md	Update README.md	4 days ago
_config.yml	Set theme jekyll-theme-tactile	21 days ago
didi_s_project_fd.drawio	Add files via upload	4 days ago
didi_s_sentiment_classifier_fd.drawio	Add files via upload	4 days ago

About

No description, website, or topics provided.

[Readme](#)

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Environments 1

[github-pages](#) Active

Version Control System



Repository is a shared folder where there are saved files necessary for the configuration, development and execution of a project

Workspace is a local folder that developers utilize to work

Both repository and workspace are part of a Version Control System (VCS) that defines the shared development and resolves conflicts

VCS typologies



local: it is a simple database that tracks changes to files under version

centralized: it is a repository stored on a server, while clients access to individual files

distributed: clients have an integral copy of a repository



VCS typologies



local: it is a simple database that tracks changes to files under version

centralized: it is a repository stored on a server, while clients access to individual files

distributed: clients have an integral copy of a repository

Git and GitHub

How a project looks like



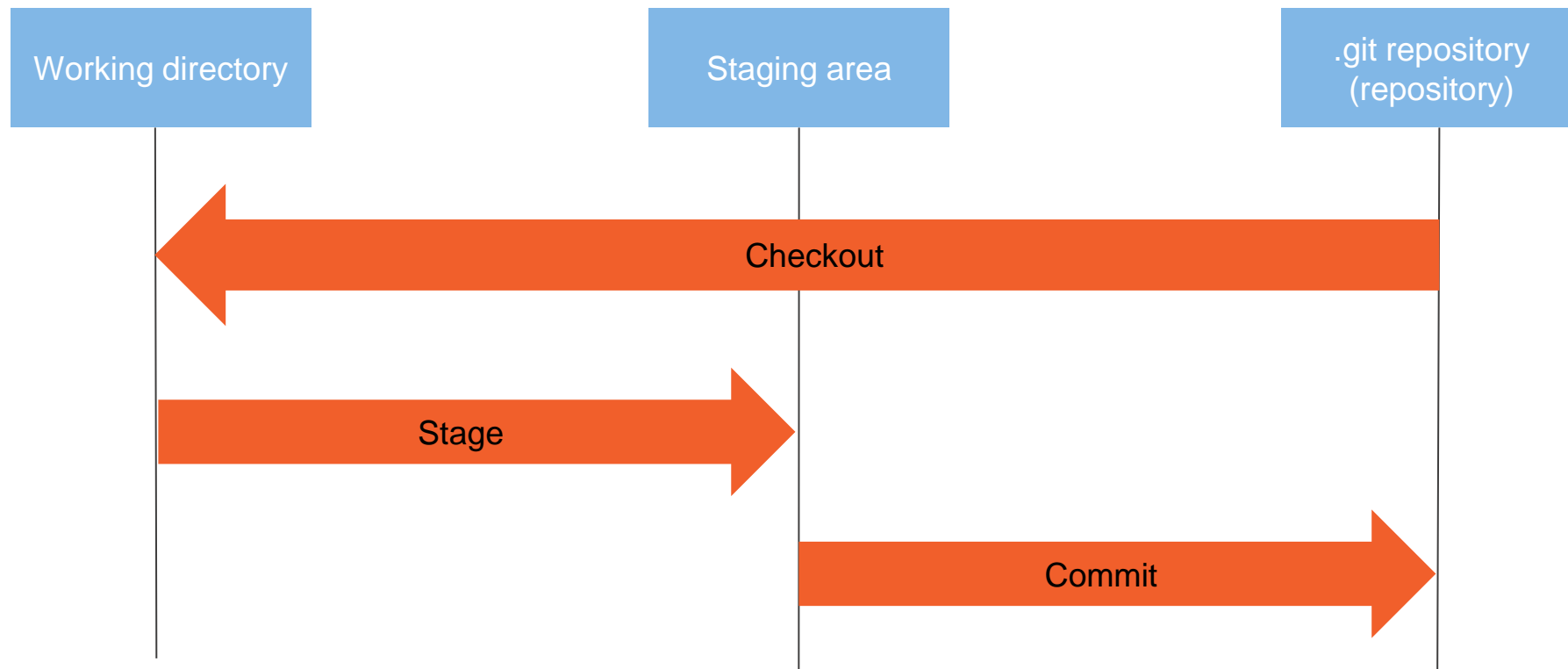
`.git` folder: it is managed automatically and contains the access information to the repository. This folder is created once there is a clone operation from the remote repository

whole working folder: it contains a project version checkout. Files of this folder are computed starting from the indexes present in `.git`. Those files can be modified by a user locally

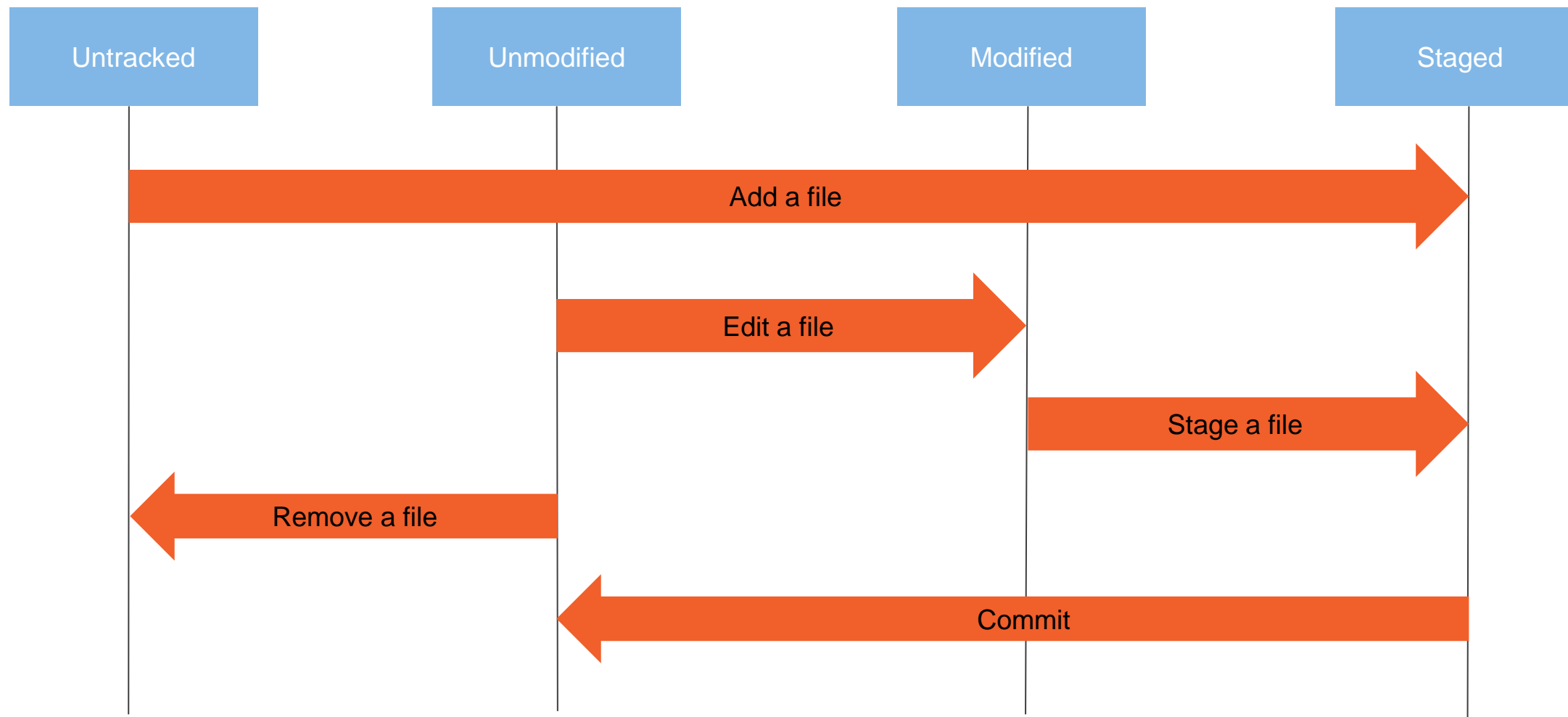
stage area: it saves all files that will be included in the request to update the remote repository with the local changes. Files that are modified locally, but not stages, will not generate a request of change remotely



Flow



Flow



Basic commands



- git status: it gets the status of the files in the workspace
- git add: it stages a file
- git diff: it shows differences between the workspace copy and the one in the repository
- git commit: it does a commit to the workspace
- git mv: it moves a file from a folder to another and the change has an impact to the repository
- git rm: it removes a file with an impact remotely
- git log: it visualizes a log
- git init: it initializes an empty repository
- git clone: it clones a remote repository
- git pull: it downloads the changes done in the remote repository
- git push: it pushes the changes staged to the remote repository

git clone



remote address of the repository we aim to clone locally

```
$ git clone git@github.com:adsp-polito/adsp-polito.github.io.git
Cloning into 'adsp-polito.github.io'...
remote: Enumerating objects: 56, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 56 (delta 24), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (56/56), 5.17 MiB | 8.01 MiB/s, done.
Resolving deltas: 100% (24/24), done.
```

git status



```
$ 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)

L8 - ADSP - AgileSwDev.pdf

a new file is in the workspace but not in the repository

nothing added to commit but untracked files present (use "git add" to track)



git add



```
$ git add L8\ -\ ADSP\ -\ AgileSwDev.pdf
```

add a new file in the workspace

```
$ git status
```

On branch main

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

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: L8 - ADSP - AgileSwDev.pdf

a new file is staged



git commit



```
$ git commit L8\ -\ ADSP\ -\ AgileSwDev.pdf -m "add slides of the lecture  
made by prof. Marco Torchiano"  
[main dd5e042] add slides of the lecture made by prof. Marco Torchiano  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 L8 - ADSP - AgileSwDev.pdf
```

```
$ git status
```

On branch main

```
Your branch is ahead of 'origin/main' by 1 commit.  
(use "git push" to publish your local commits)
```

our workspace is ahead to the
remote repository

nothing to commit, working tree clean



git push



```
$ git push
```

```
Enumerating objects: 4, done.
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 8 threads
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 1.12 MiB | 9.83 MiB/s, done.
```

```
Total 3 (delta 1), reused 0 (delta 0)
```

```
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```

```
To github.com:adsp-polito/adsp-polito.github.io.git  
e05b240..dd5e042 main -> main
```

the change is propagated remotely

```
$ git status
```

```
On branch main
```

```
Your branch is ahead of 'origin/main' by 1 commit.
```

```
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```


git pull



```
$ git pull
```

```
remote: Enumerating objects: 4, done.
```

```
remote: Counting objects: 100% (4/4), done.
```

```
remote: Compressing objects: 100% (3/3), done.
```

```
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
```

```
Unpacking objects: 100% (3/3), 7.31 MiB | 5.94 MiB/s, done.
```

```
From github.com:adsp-polito/adsp-polito.github.io
```

```
dd5e042..f2c6801 main -> origin/main
```

```
Updating dd5e042..f2c6801
```

```
Fast-forward
```

```
L9 - ADSP - Scrum.pdf | Bin 0 -> 9285900 bytes
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 L9 - ADSP - Scrum.pdf
```

a new file present in the remote repository is also added to the local workspace





Thank you for your attention.

Questions?

CONTACTS

Giuseppe Rizzo

Program Manager (LINKS Foundation) and
Adjunct Professor (Politecnico di Torino)

giuseppe.rizzo@polito.it



FONDAZIONE LINKS
Via Pier Carlo Boggio 61 | 10138 Torino
P. +39 011 22 76 150
LINKSFOUNDATION.COM