Strumenti SW

per il corso di Tecnologie Web

Strumenti software

Questa serie di slide riassume brevemente gli strumenti che verranno utilizzati in questo corso.

Alcuni strumenti sono obbligatori,

altri consigliati,

Vi invito a provarli un po tutti e a farvi un’idea vostra….

Python toolchain

Python **è** il linguaggio “principe” di questo corso.

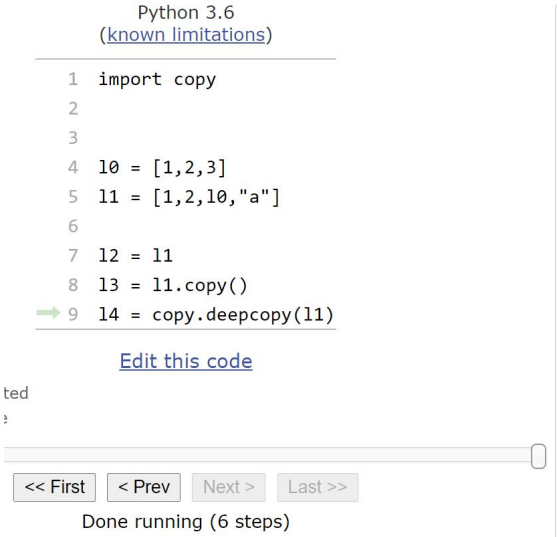
Vi è un altro plico di slide con alcuni link per averne un’installazione di base per tutti i sistemi operativi comunemente utilizzati.

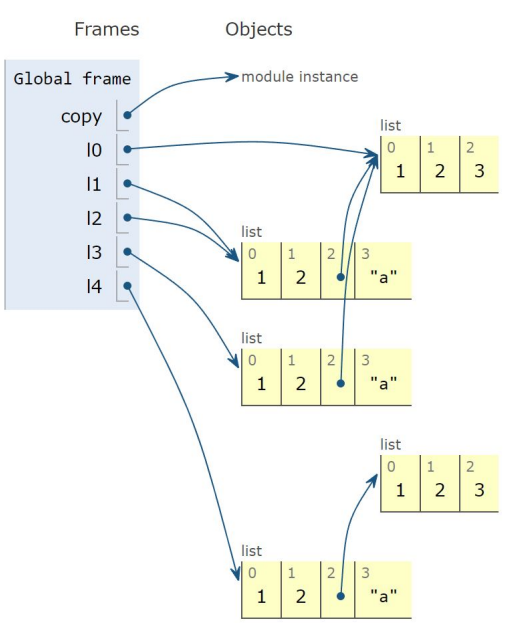
IMPORTANTE: Python ver >= 3.4

Installando python si installano anche i relativi tool per la gestione dei pacchetti, i.e. **pip**

Per “capire meglio” Python

https://pythontutor.com/





pipenv [fortemente consigliato]

It automatically creates and manages a virtualenv for your projects, as well as adds/removes packages from your `Pipfile` *as you install/uninstall packages. It also generates the ever-important* `Pipfile.lock`*, which is used to produce deterministic builds*

In altre parole, pipenv ci permette di:

- “Isolare” i nostri progetti in cartelle appartenenti ambienti virtuali
- Tiene traccia delle **effettive** dipendenze dei nostri progetti
- Permette di distribuire agevolmente i nostri sorgenti
- Permette di installare agevolmente i nostri progetti

IDE (possibile scelta)

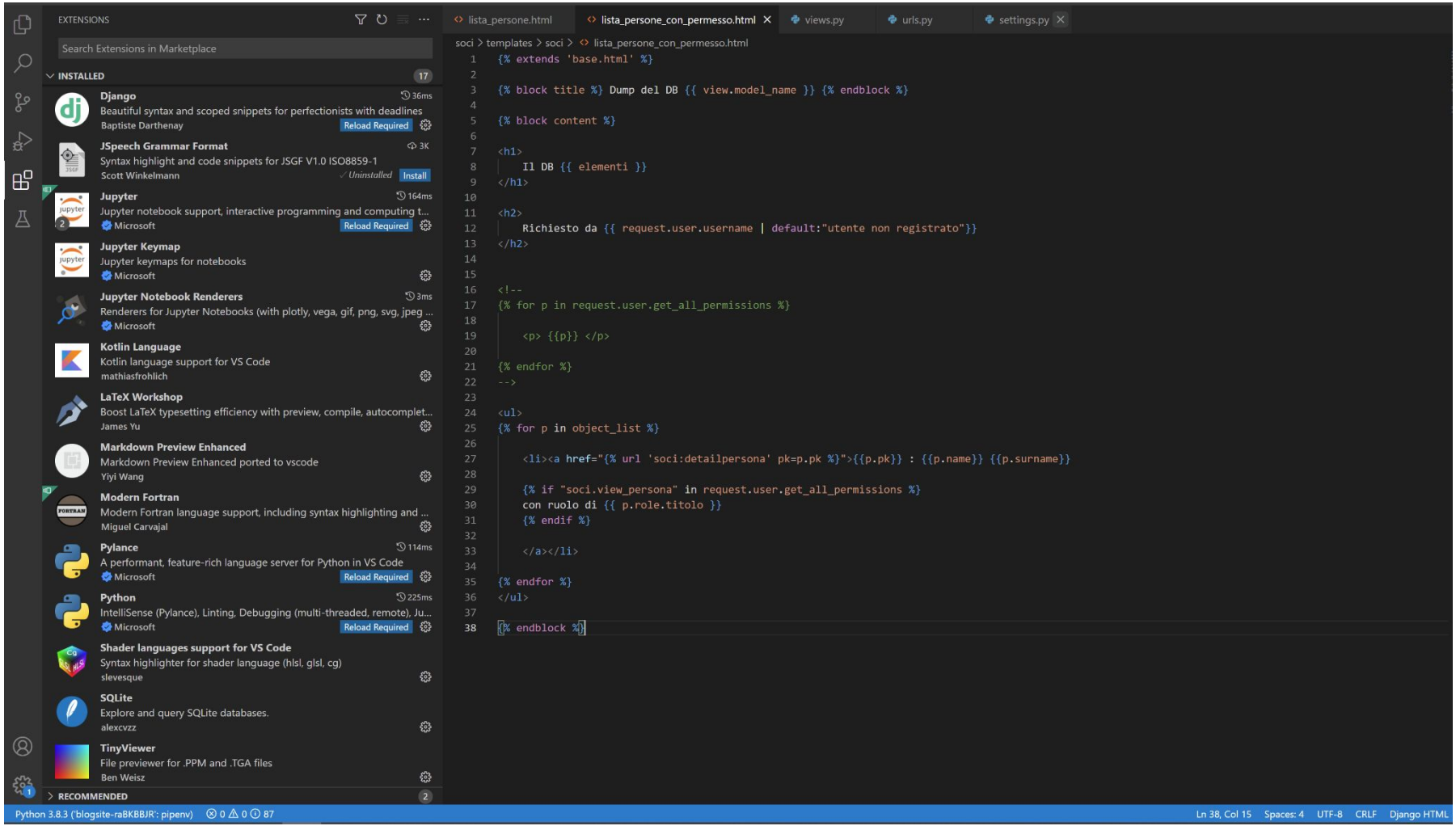
Il già citato PyCharm con versione per studenti.

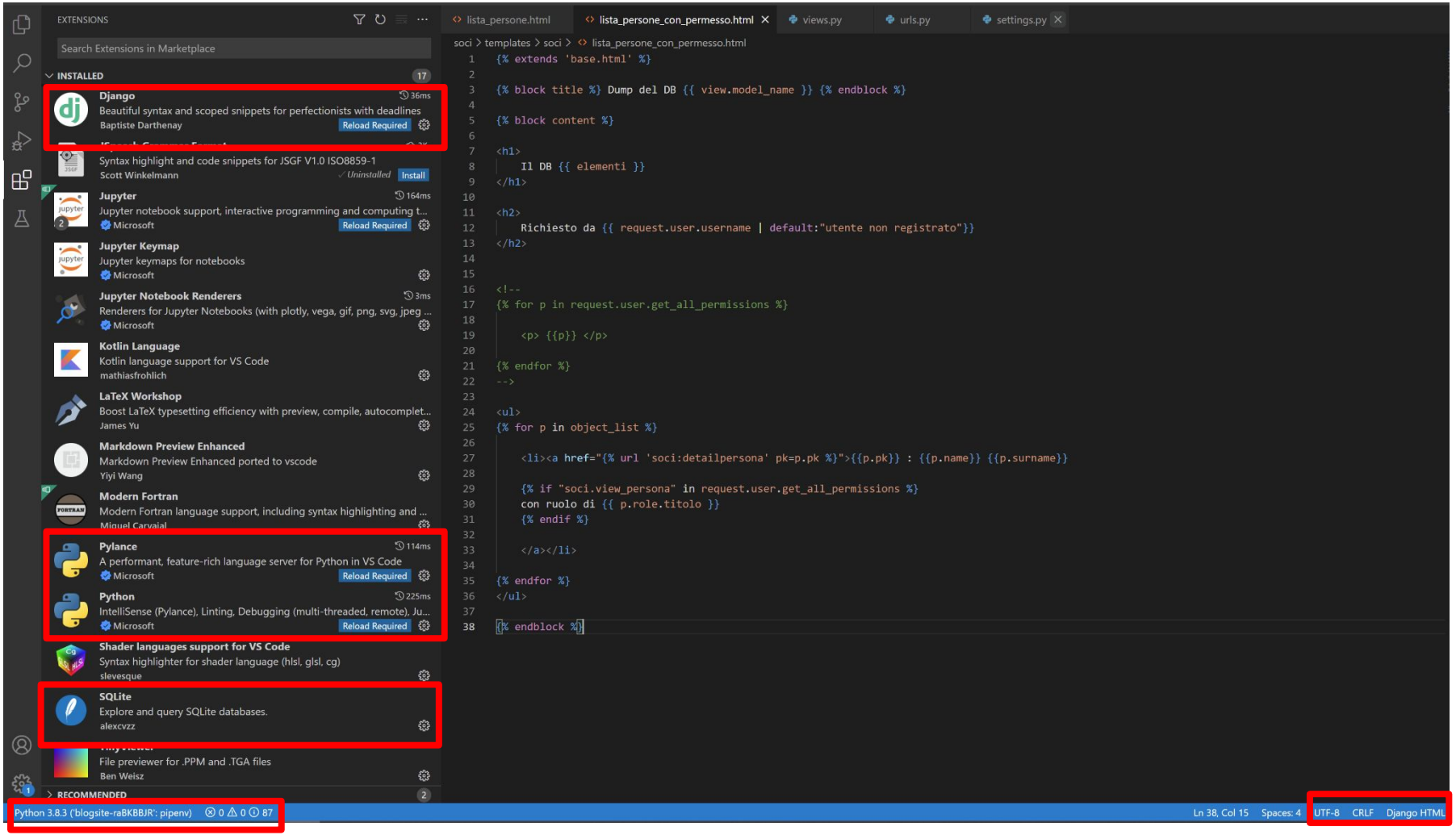
- Un buon IDE per python, ampiamente customizzabile
- Integrazione con DBs
- Integrazione con Django
- Integrazione con git

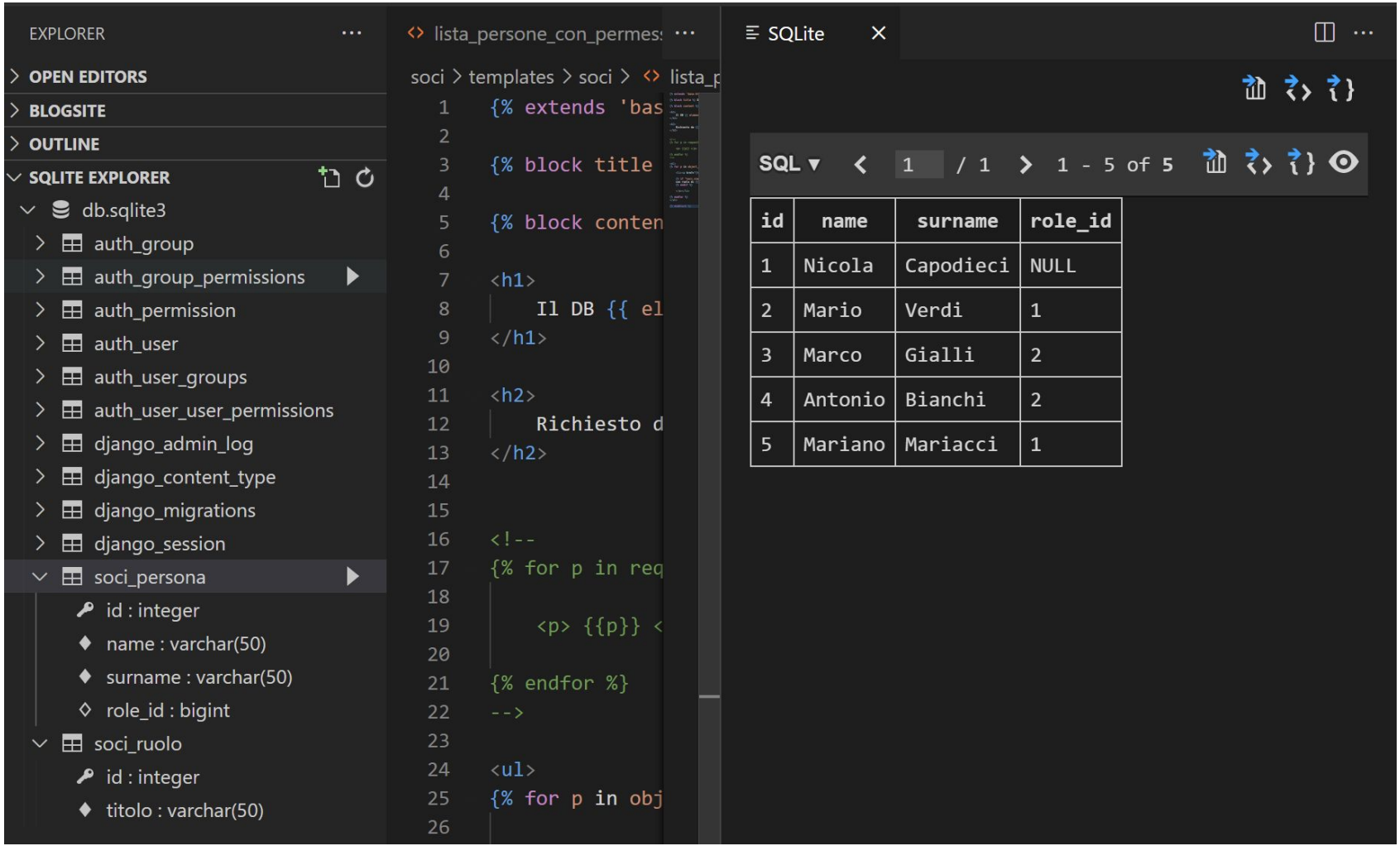
IDE (altra possibile scelta)

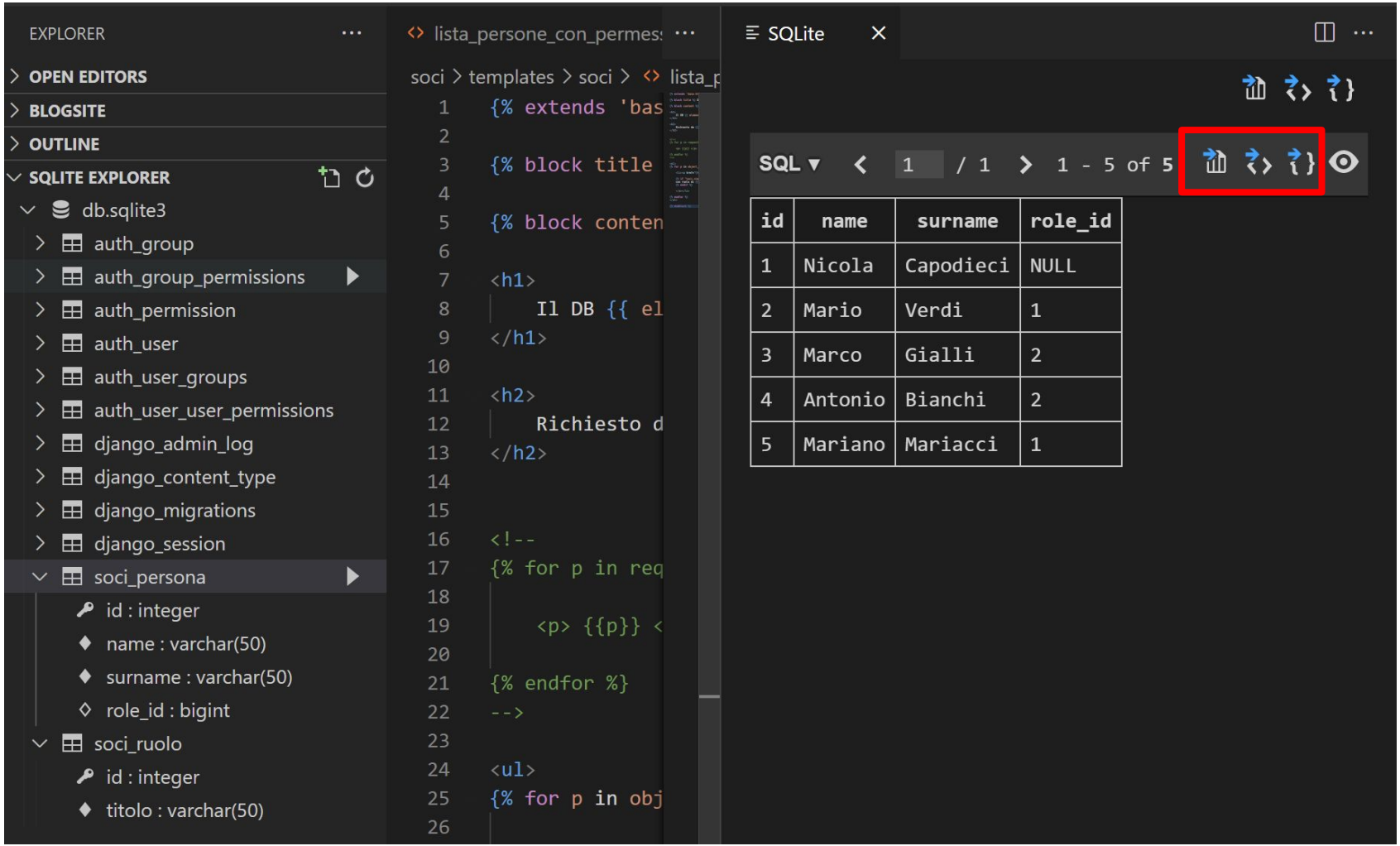
Visual Studio code + cmd line

- Integrazione nativa con git
- Plugin per Python
- Plugin per DTL
- Plugin per accesso a DB
- … altro vedere slides successiva


EXTENSIONS
Search Extensions in Marketplace
INSTALLED 17
Django 36ms
Beautiful syntax and scoped snippets for perfectionists with deadlines
Baptiste Darthenay
Reload Required
JSpeech Grammar Format 3K
Syntax highlight and code snippets for JSGF V1.0 ISO8859-1
Scott Winkelmann
Uninstalled
Install
Jupyter 164ms
Jupyter notebook support, interactive programming and computing t...
Microsoft
Reload Required
Jupyter Keymap
Jupyter keymaps for notebooks
Microsoft
Jupyter Notebook Renderers 3ms
Renderers for Jupyter Notebooks (with plotly, vega, gif, png, svg, jpeg ...
Microsoft
Kotlin Language
Kotlin language support for VS Code
mathiasfrohlich
LaTeX Workshop
Boost LaTeX typesetting efficiency with preview, compile, autocomplet...
James Yu
Markdown Preview Enhanced
Markdown Preview Enhanced ported to vscode
Yiyi Wang
Modern Fortran
Modern Fortran language support, including syntax highlighting and ...
Miguel Carvajal
Pylance 114ms
A performant, feature-rich language server for Python in VS Code
Microsoft
Reload Required
Python 225ms
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Ju...
Microsoft
Reload Required
Shader languages support for VS Code
Syntax highlighter for shader language (hlsl, glsl, cg)
slevesque
SQLite
Explore and query SQLite databases.
alexcvzz
TinyViewer
File previewer for .PPM and .TGA files
Ben Weisz
RECOMMENDED 2
lista_persone.html
lista_persone_con_permesso.html
views.py
urls.py
settings.py
soci > templates > soci > lista_persone_con_permesso.html
1 {% extends 'base.html' %}
2
3 {% block title %} Dump del DB {{ view.model_name }} {% endblock %}
4
5 {% block content %}
6
7 <h1>
8 Il DB {{ elementi }}
9 </h1>
10
11 <h2>
12 Richiesto da {{ request.user.username | default:"utente non registrato"}}
13 </h2>
14
15
16 <!--
17 {% for p in request.user.get_all_permissions %}
18
19 {{p}}
20
21 {% endfor %}
22 -->
23
24
25 {% for p in object_list %}
26
27 {{p.pk}} : {{p.name}} {{p.surname}}
28
29 {% if "soci.view_persona" in request.user.get_all_permissions %}
30 con ruolo di {{ p.role.titolo }}
31 {% endif %}
32
33
34
35 {% endfor %}
36
37
38 {% endblock %}
Python 3.8.3 ('blogsite-raBKBBJR': pipenv) 0 0 87
Ln 38, Col 15 Spaces: 4 UTF-8 CRLF Django HTML


EXTENSIONS
Search Extensions in Marketplace
INSTALLED 17
Django 36ms
Beautiful syntax and scoped snippets for perfectionists with deadlines
Baptiste Darthenay
Reload Required
Syntax highlight and code snippets for JSGF V1.0 ISO8859-1
Scott Winkelmann
Uninstalled
Install
Jupyter 164ms
Jupyter notebook support, interactive programming and computing t...
Microsoft
Reload Required
Jupyter Keymap
Jupyter keymaps for notebooks
Microsoft
Jupyter Notebook Renderers 3ms
Renderers for Jupyter Notebooks (with plotly, vega, gif, png, svg, jpeg ...
Microsoft
Kotlin Language
Kotlin language support for VS Code
mathiasfrohlich
LaTeX Workshop
Boost LaTeX typesetting efficiency with preview, compile, autocomplet...
James Yu
Markdown Preview Enhanced
Markdown Preview Enhanced ported to vscode
Yiyi Wang
Modern Fortran
Modern Fortran language support, including syntax highlighting and ...
Miguel Carvajal
Pylance 114ms
A performant, feature-rich language server for Python in VS Code
Microsoft
Reload Required
Python 225ms
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Ju...
Microsoft
Reload Required
Shader languages support for VS Code
Syntax highlighter for shader language (hlsl, glsl, cg)
slevesque
SQLite
Explore and query SQLite databases.
alexcvzz
File previewer for .PPM and .TGA files
Ben Weisz
RECOMMENDED 2
lista_persone.html
lista_persone_con_permesso.html
views.py
urls.py
settings.py
soci > templates > soci > lista_persone_con_permesso.html
1 {% extends 'base.html' %}
2
3 {% block title %} Dump del DB {{ view.model_name }} {% endblock %}
4
5 {% block content %}
6
7 <h1>
8 Il DB {{ elementi }}
9 </h1>
10
11 <h2>
12 Richiesto da {{ request.user.username | default:"utente non registrato"}}
13 </h2>
14
15
16 <!--
17 {% for p in request.user.get_all_permissions %}
18
19 {{p}}
20
21 {% endfor %}
22 -->
23
24
25 {% for p in object_list %}
26
27 {{p.pk}} : {{p.name}} {{p.surname}}
28
29 {% if "soci.view_persona" in request.user.get_all_permissions %}
30 con ruolo di {{ p.role.titolo }}
31 {% endif %}
32
33
34
35 {% endfor %}
36
37
38 {% endblock %}
Python 3.8.3 ('blogsite-raBKBBJR': pipenv) 0 0 87
Ln 38, Col 15 Spaces: 4 UTF-8 CRLF Django HTML


EXPLORER
OPEN EDITORS
BLOGSITE
OUTLINE
SQLITE EXPLORER
db.sqlite3
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
django_content_type
django_migrations
django_session
soci_persona
id : integer
name : varchar(50)
surname : varchar(50)
role_id : bigint
soci_ruolo
id : integer
titolo : varchar(50)
lista_persone_con_permes
soci > templates > soci > lista_p
1 {% extends 'bas
2
3 {% block title
4
5 {% block conten
6
7 <h1>
8 Il DB {{ el
9 </h1>
10
11 <h2>
12 Richiesto d
13 </h2>
14
15
16 <!--
17 {% for p in req
18
19 <p> {{p}} <
20
21 {% endfor %}
22 -->
23
24
25 {% for p in obj
26
SQLite
SQL
1 / 1
1 - 5 of 5
id name surname role_id
1 Nicola Capodieci NULL
2 Mario Verdi 1
3 Marco Gialli 2
4 Antonio Bianchi 2
5 Mariano Mariacci 1


EXPLORER
OPEN EDITORS
BLOGSITE
OUTLINE
SQLITE EXPLORER
db.sqlite3
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
django_content_type
django_migrations
django_session
soci_persona
id : integer
name : varchar(50)
surname : varchar(50)
role_id : bigint
soci_ruolo
id : integer
titolo : varchar(50)
lista_persone_con_permes
soci > templates > soci > lista_p
1 {% extends 'bas
2
3 {% block title
4
5 {% block conten
6
7 <h1>
8 Il DB {{ el
9 </h1>
10
11 <h2>
12 Richiesto d
13 </h2>
14
15
16 <!--
17 {% for p in req
18
19 <p> {{p}} <
20
21 {% endfor %}
22 -->
23
24
25 {% for p in obj
26
SQLite
SQL
1
/ 1
1 - 5 of 5
id
name
surname
role_id
1
Nicola
Capodieci
NULL
2
Mario
Verdi
1
3
Marco
Gialli
2
4
Antonio
Bianchi
2
5
Mariano
Mariacci
1

Git

Esempi di codice, slides e altro saranno distribuiti tramite repo git.

Git è un **sistema** di controllo di “versioning” creato originalmente da Linus Torvalds nel 2005.

Non è l’unico e non è il primo…

Linus Torvalds on Git (Why Subversion Sucks):

http://courses.ics.hawaii.edu/ics314f16/morea/configuration-management/reading-screencast-torvalds-on-git.html

Come funziona Git

1. **Crea** un "repository" (repo/project) *navigabile tramite un frontend\hosting tool*
2. **Copia** (o **clona**) il repository in ***locale***
3. **Aggiungi** uno o più file al tuo repo locale e tramite un'operazione di "commit" si esegue il salvataggio delle modifiche
4. "**Push**" le tue modifiche al **main branch** (salvataggio in remoto)
5. Tali modifiche possono essere fatte anche tramite un **web frontend**
6. "**Pull**" cioè scarica le modifiche altrui sulla tua macchina locale
7. Crea altri "**branch**" (versioni)
8. Ripeti i passi dal 2/3 al 6 …
9. **Unisci** (merge) diversi branch

Funzionalità avanzate

- Risoluzione (manuale) dei conflitti
- “Blaming”
- CI/CD, testing
- Pull requests
- Moduli
- …

Git frontends

Git è un **sistema**.

Nella sua installazione di base accetta comandi testuali, che vedremo in seguito.

Un frontend di Git è un programma che rende disponibile un’interfaccia utente di solito grafica per agevolare le operazioni su un git repo

Esistono quindi front-end desktop e web hosting tools.

Inoltre la quasi totalità degli IDE comunemente utilizzati si interfaccia “nativamente” a git.

Git Desktop frontends

https://git-scm.com/downloads/guis

Git web hosting tools

https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities

Esempi su gitlab

https://git.hipert.unimore.it/ncapodieci/vkpolybench

https://gitlab.com/cicciodev/fim-tecnologie_web

https://git.hipert.unimore.it/ncapodieci/mobileprogramming

Il git del corso

https://git.hipert.unimore.it/ncapodieci/techweb

MobileProgramming

Project ID: 219

Star 0

45 Commits **1** Branch **0** Tags **112.9 MB** Files **113.1 MB** Storage

master mobileprogramming History Find file Clone

Link is now fixed. Again
ncapodieci authored 2 months ago

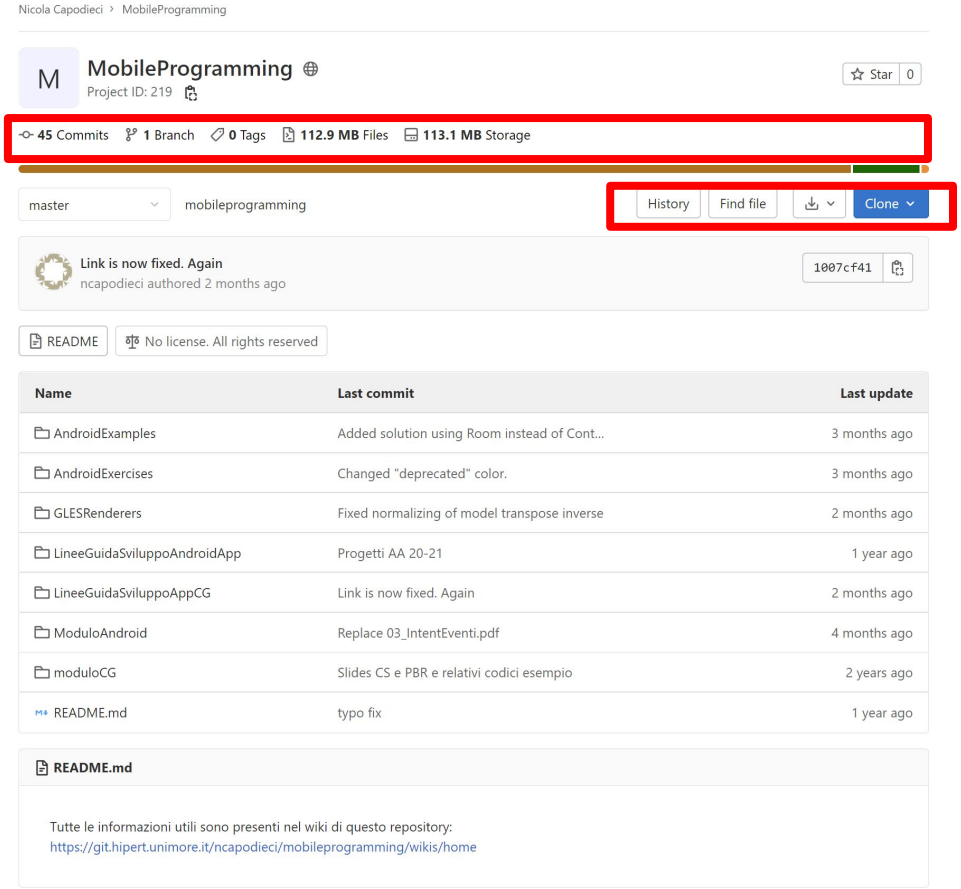
1007cf41

README No license. All rights reserved

Name	Last commit	Last update
AndroidExamples	Added solution using Room instead of Cont...	3 months ago
AndroidExercises	Changed "deprecated" color.	3 months ago
GLESRenderers	Fixed normalizing of model transpose inverse	2 months ago
LineeGuidaSviluppoAndroidApp	Progetti AA 20-21	1 year ago
LineeGuidaSviluppoAppCG	Link is now fixed. Again	2 months ago
ModuloAndroid	Replace 03_IntentEventi.pdf	4 months ago
moduloCG	Slides CS e PBR e relativi codici esempio	2 years ago
README.md	typo fix	1 year ago

README.md

Tutte le informazioni utili sono presenti nel wiki di questo repository:
https://git.hipert.unimore.it/ncapodieci/mobileprogramming/wikis/home



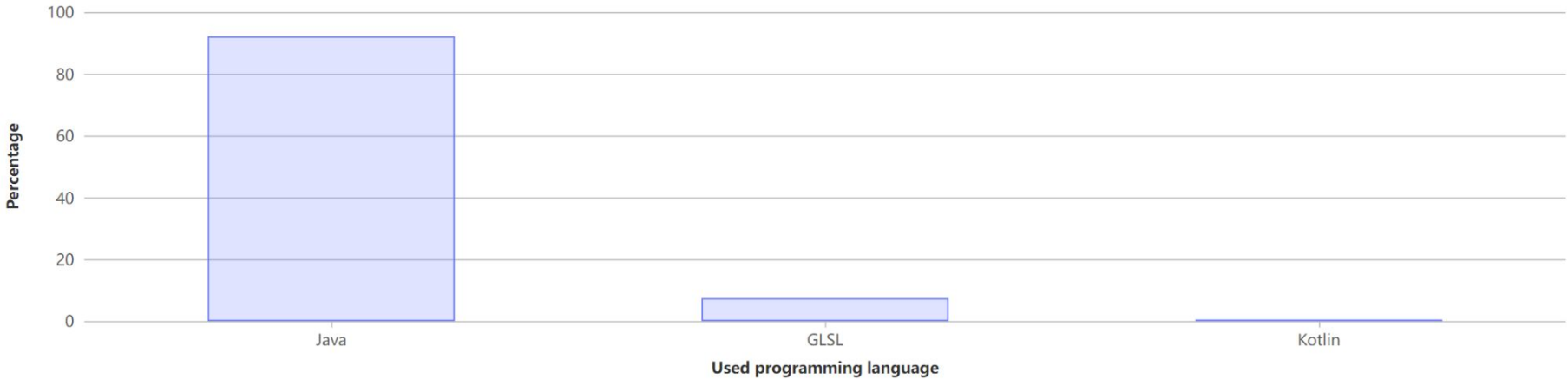
I repo **dovrebbero** avere una licenza

I repo **dovrebbero** avere un readme, e **potrebbero** avere un wiki

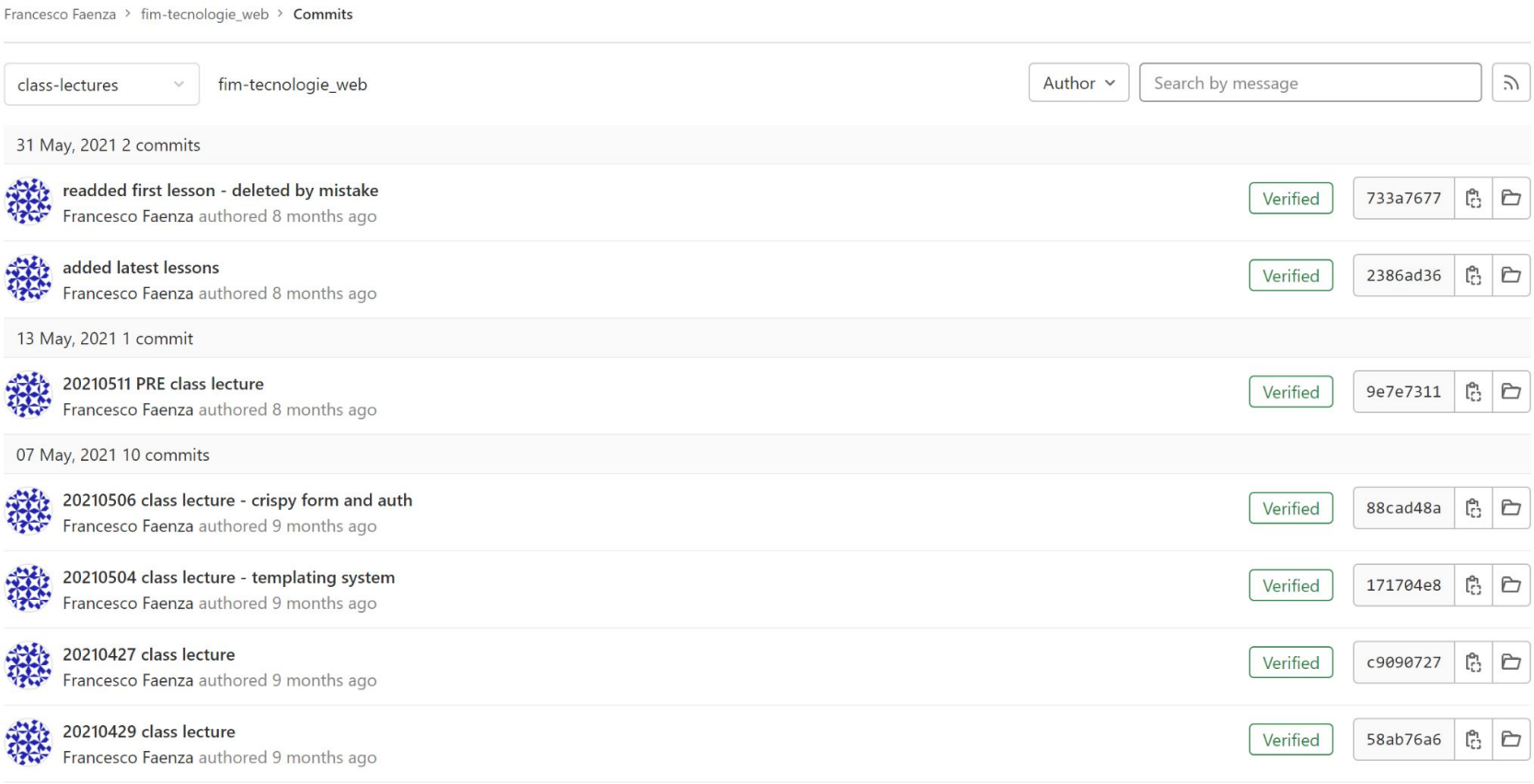
Repository Analytics

Programming languages used in this repository

Measured in bytes of code. Excludes generated and vendored code.



History


Francesco Faenza > fim-tecnologie_web > Commits
class-lectures
fim-tecnologie_web
Author
Search by message
31 May, 2021 2 commits
readded first lesson - deleted by mistake
Francesco Faenza authored 8 months ago
Verified
733a7677
added latest lessons
Francesco Faenza authored 8 months ago
Verified
2386ad36
13 May, 2021 1 commit
20210511 PRE class lecture
Francesco Faenza authored 8 months ago
Verified
9e7e7311
07 May, 2021 10 commits
20210506 class lecture - crispy form and auth
Francesco Faenza authored 9 months ago
Verified
88cad48a
20210504 class lecture - templating system
Francesco Faenza authored 9 months ago
Verified
171704e8
20210427 class lecture
Francesco Faenza authored 9 months ago
Verified
c9090727
20210429 class lecture
Francesco Faenza authored 9 months ago
Verified
58ab76a6

Branches

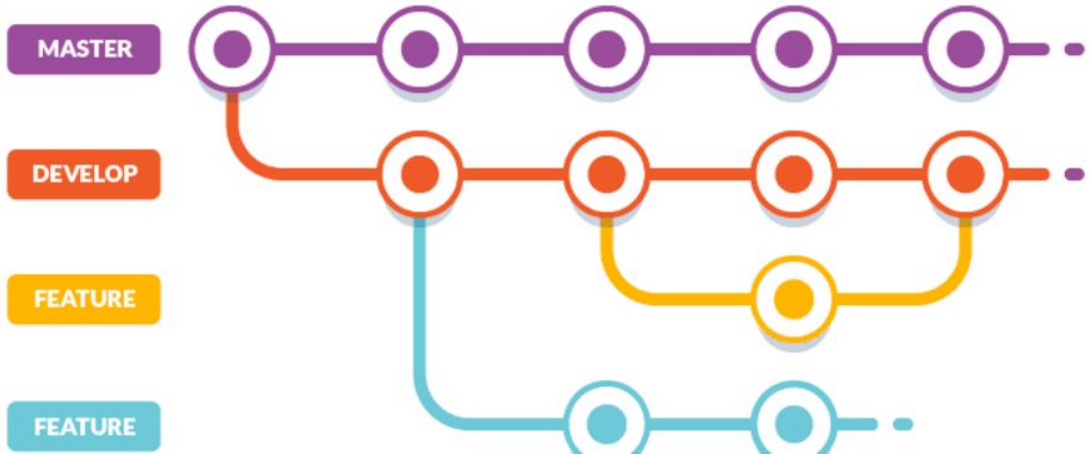
Nicola Capodieci › vkpolybench › Graph

master

You can move around the graph by using the arrow keys.

Git revision | Begin with the selected commit

Aug	19	master	Update README.md: added citing info
Mar	8	1.0	Add .gitattributes file for recognizing glsl shaders
			Update README.md
			high contrast logo lowres added
			Update README.md
			Added logo and android app icon
	6		Merge branch 'vkpolyandroid' into 'master'
		vkpolyandroid	Updated license references
	3		More Updates to README.md
			Update README.md
			Updated README and minor fixes
	2		Clean up and licenses added
Feb	9		Updates to README.md
			Android port is finalized. With tests and results
	8		[ALL]Maybe fixed gradle and tested every benchmark
			[ATAX] Gradle is the worst system ever built.
			[2/3DCONV,2/3MM] Works on android w/ gradle issues
			[2DCONV] Tested and working on Android
	7		glsl to spv bin crosscompile for Android added
			spvBin->shaderModule member function for Android
			Build passes. No tests.
	6		WARM UP RUN support added for all benchmarks
			[3DCONV] Support for WARM UP run



Commit\Push\Pull\

Removed intentional memory leak in Photo\Browser.

parent 8c5bbbf4 master

No related merge requests found

Changes 1

Showing 1 changed file with 6 additions and 5 deletions

Hide whitespace changes | Inline | Side-by-side

AndroidExercises/03_IntEveAdvanced_solutions/es1_FotoBrowser/MainActivity.java

View file @f632f599

```
@@ -28,16 +28,17 @@ public class MainActivity extends Activity implements TextWatcher, View.OnClickL
        setContentView(R.layout.activity_main);
        text = findViewById(R.id.text);
        pulsante = findViewById(R.id.button);
-   }
-
-   @Override
-   protected void onStart(){
-       super.onStart();

        text.addTextChangedListener(this);
        pulsante.setOnClickListener(this);

        im = findViewById(R.id.imageView);
+
+   }
+
+   @Override
+   protected void onStart(){
+       super.onStart();
    }

    @Override
```

Usare repo di altri

Si clonano in locale.

Abbiamo visto i “pulsantini” dell'interfaccia di gitlab.

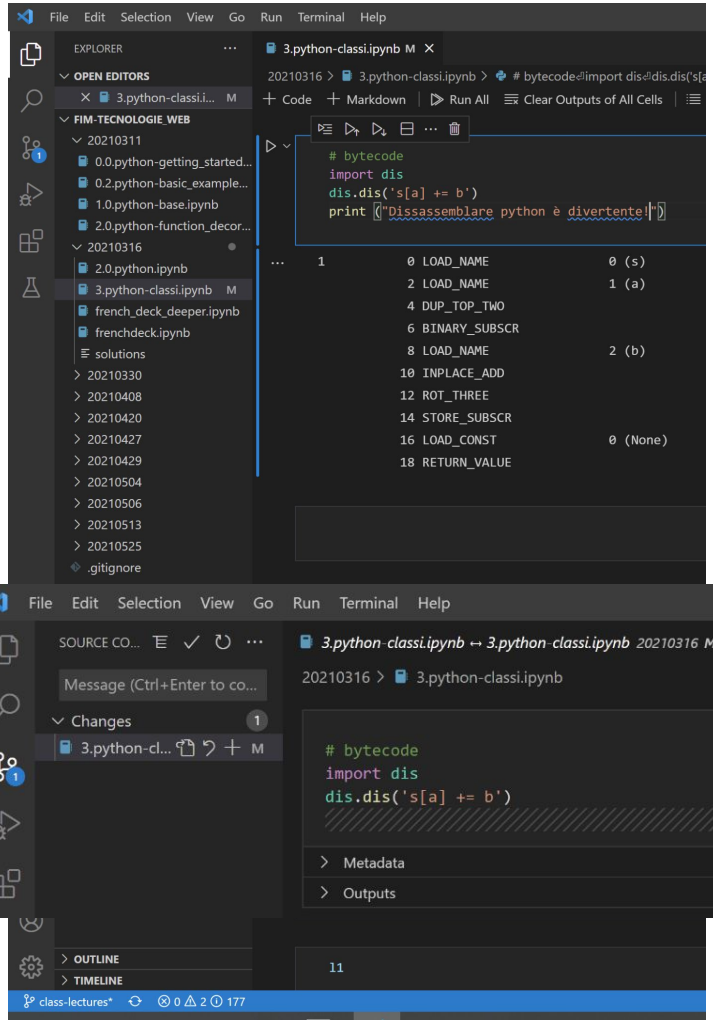
via cmd line: **git clone gitrepoaddress**

git clone https://git.hipert.unimore.it/ncapodieci/vkpolybench.git

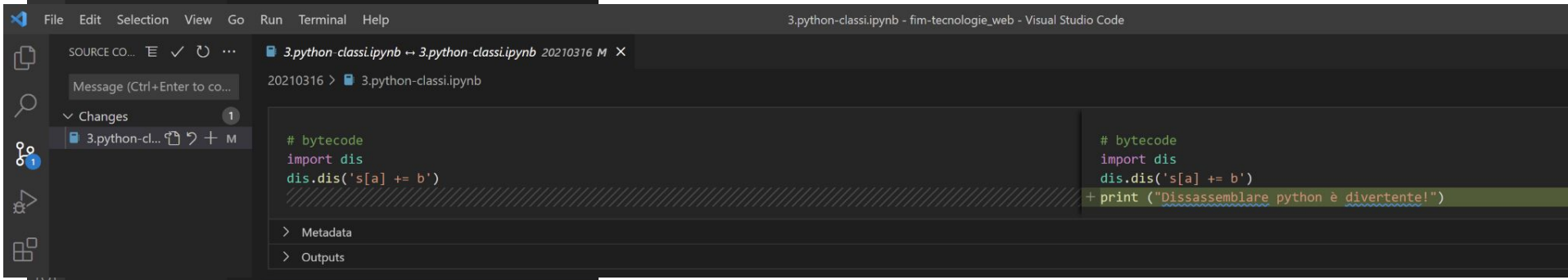
git clone https://gitlab.com/cicciodev/fim-tecnologie_web.git

git clone https://git.hipert.unimore.it/ncapodieci/mobileprogramming

PS C:\Users\Nicola\Desktop\techwb\repos> *git clone https://gitlab.com/cicciodev/fim-tecnologie_web.git*
Cloning into 'fim-tecnologie_web'...
remote: Enumerating objects: 220, done.
remote: Counting objects: 100% (220/220), done.
remote: Compressing objects: 100% (141/141), done.
Receive: Total 220 (delta 90), reused 135 (delta 57), pack-reused 0eceiving objects: 51% (113/220)
Receiving objects: 100% (220/220), 125.95 KiB | 0 bytes/s, done.
Resolving deltas: 100% (90/90), done.
PS C:\Users\Nicola\Desktop\techwb\repos> *cd .\fim-tecnologie_web*
PS C:\Users\Nicola\Desktop\techwb\repos\fim-tecnologie_web> *git branch*
* main
PS C:\Users\Nicola\Desktop\techwb\repos\fim-tecnologie_web> *git branch -a*
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/class-lectures
  remotes/origin/main
  remotes/origin/prod-example
PS C:\Users\Nicola\Desktop\techwb\repos\fim-tecnologie_web> **git checkout class-lectures**
Branch class-lectures set up to track remote branch class-lectures from origin.
Switched to a new branch 'class-lectures'



Integrazione con IDE,
In questo caso VS Code



Crearsi il proprio repo via cmd line

1. Scaricare git
2. Configurare la propria identità
3. Il branch main\master è di norma già creato
4. Creare altri branch
5. fare un commit di prova (magari inserendo un README.md)
6. …

Verifica dell’installazione

git --version

Stabilire la propria identità

git config --global user.name "username"

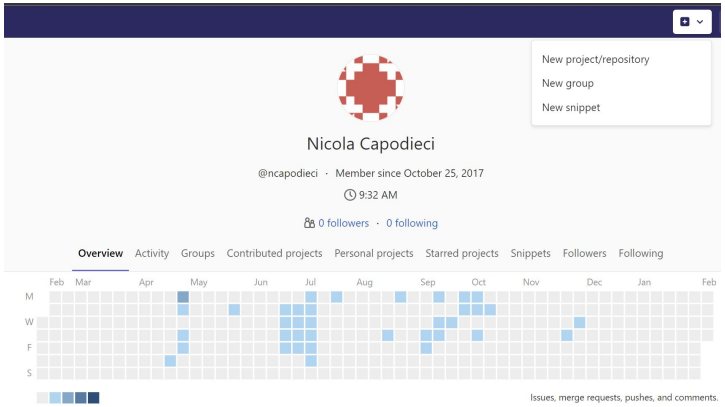
git config --global user.email "nome@email.com"

git config --list

NB: L’accesso ai repo può essere protetto da username & password, ma anche metodi di autenticazione più sofisticati, e.g. SSL

Inizializzare un repo

- Ho codice esistente?
  - cd /path/to/your/existing/code
  - git init
- Vado from scratch?
  - git init <nomerepo>
- Terza opzione: lo creo tramite web hosting tool e lo clono

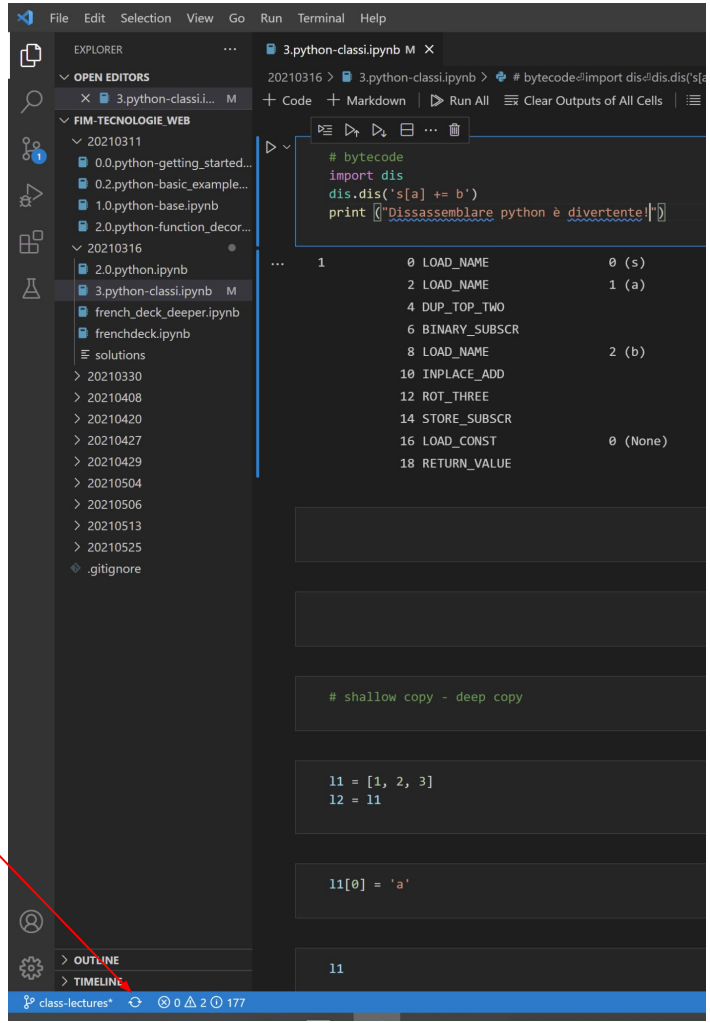


git pull

Pull: scaricamento in locale delle modifiche effettuate da altri sviluppatori

- Assicurasi di avere un git inizializzato (e.g. precedentemente clonato)
- Assicurarsi di aver fatto il checkout del branch di interesse
- git pull
- git pull <remote branch>

Da IDE


File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
3.python-classi.i... M
FIM-TECNOLOGIE_WEB
20210311
0.0.python-getting_started...
0.2.python-basic_example...
1.0.python-base.ipynb
2.0.python-function_decor...
20210316
2.0.python.ipynb
3.python-classi.ipynb M
french_deck_deeper.ipynb
frenchdeck.ipynb
solutions
20210330
20210408
20210420
20210427
20210429
20210504
20210506
20210513
20210525
.gitignore
OUTLINE
TIMELINE
3.python-classi.ipynb M
20210316 > 3.python-classi.ipynb > # bytecode import dis dis.dis('s[a
Code Markdown Run All Clear Outputs of All Cells
# bytecode
import dis
dis.dis('s[a] += b')
print ("Dissassemblare python è divertente!")
1 0 LOAD_NAME 0 (s)
2 LOAD_NAME 1 (a)
4 DUP_TOP_TWO
6 BINARY_SUBSCR
8 LOAD_NAME 2 (b)
10 INPLACE_ADD
12 ROT_THREE
14 STORE_SUBSCR
16 LOAD_CONST 0 (None)
18 RETURN_VALUE
# shallow copy - deep copy
l1 = [1, 2, 3]
l2 = l1
l1[0] = 'a'
l1
class-lectures*
0 2 177

Cosa può andare storto?

Conflitti!

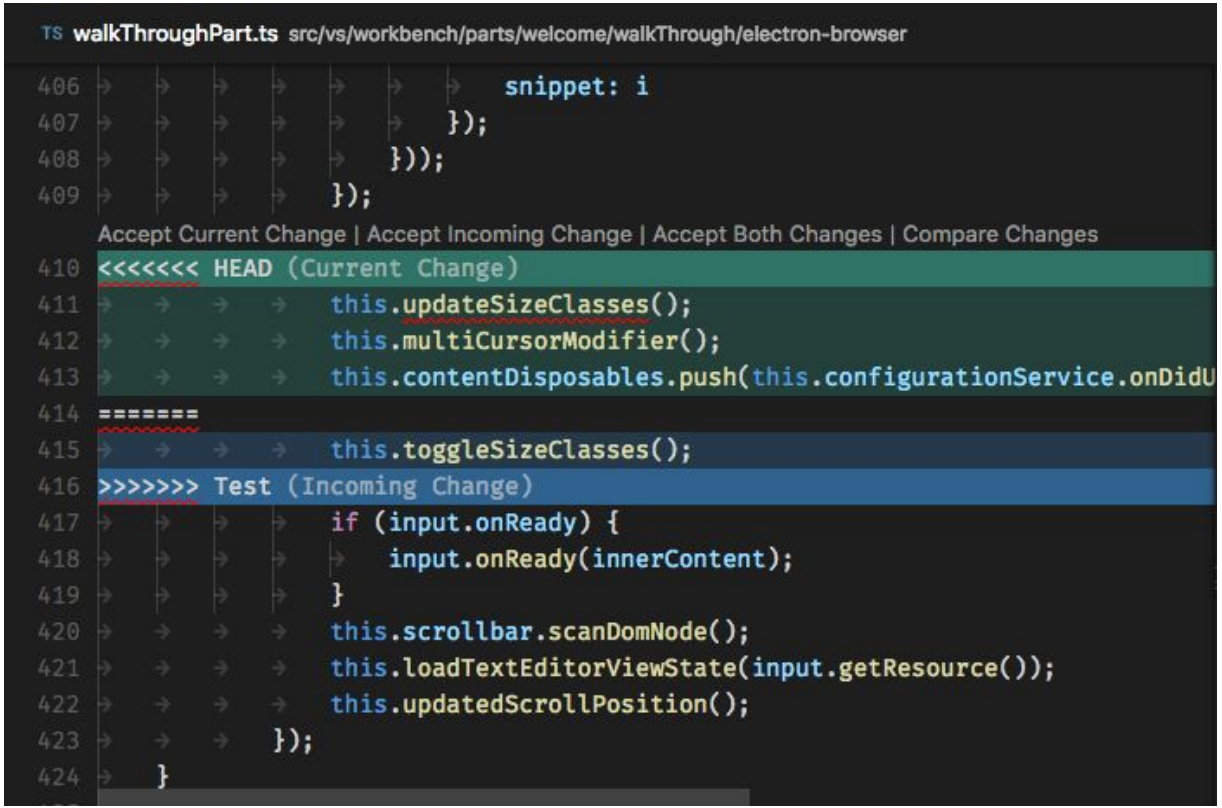
Un po di tempo fa ho clonato un repo in cui diversi sviluppatori hanno accesso.

Comincio ad apportare modifiche in locale al repo

Nel mentre altri sviluppatori eseguono modifiche e le “pushano” sullo stesso branch e negli stessi file…

Faccio un pull…

Via IDE


TS walkThroughPart.ts src/vs/workbench/parts/welcome/walkThrough/electron-browser
406 snippet: i
407 });
408 }));
409 });
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
410 <<<<<<< HEAD (Current Change)
411 this.updateSizeClasses();
412 this.multiCursorModifier();
413 this.contentDisposables.push(this.configurationService.onDidU
414 =======
415 this.toggleSizeClasses();
416 >>>>>>> Test (Incoming Change)
417 if (input.onReady) {
418 input.onReady(innerContent);
419 }
420 this.scrollbar.scanDomNode();
421 this.loadTextEditorViewState(input.getResource());
422 this.updatedScrollPosition();
423 });
424 }

Add/Commit/Save

AKA: “faccio un push” delle mie modifiche in remoto

- Assicurarsi di avere un git inizializzato (e.g. precedentemente clonato)
- Assicurarsi di aver fatto il checkout del branch di interesse
- git pull & eventuale risoluzione di conflitti

- Occorre “selezionare” i file che si intende caricare in remoto
- Occorre operare “commit” per fa sì che le modifiche entrino in uno stato chiamato “staged”. Questo obbliga l’utente ad inserire un messaggio di commit
- Infine siamo pronti per rendere effettive le modifiche in remoto.

Via cmd line

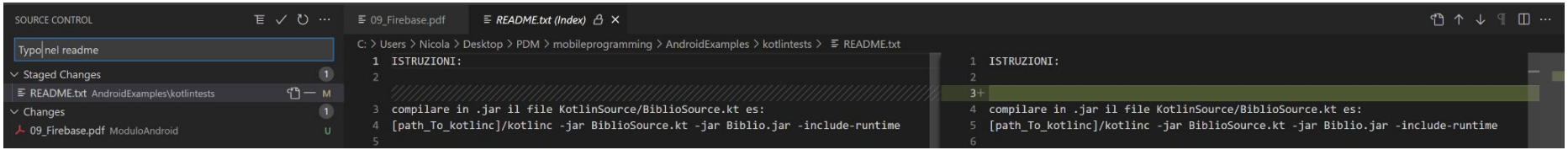
git add [nome file, nomi files, expr, wildcards]

git commit -m “messaggio di commit che dettaglia il cambiamento”

git push

> git add .\README.txt
warning: CRLF will be replaced by LF in AndroidExamples/kotlintests/README.txt.
The file will have its original line endings in your working directory.
> git commit -m "Typo nel README"
[master 031efd4] Typo nel README
 1 file changed, 1 insertion(+)
> git push

Via IDE


SOURCE CONTROL
Typo nel readme
Staged Changes 1
README.txt AndroidExamples\kotlintests M
Changes 1
09_Firebase.pdf ModuloAndroid U
09_Firebase.pdf
README.txt (Index)
C: > Users > Nicola > Desktop > PDM > mobileprogramming > AndroidExamples > kotlintests > README.txt
1 ISTRUZIONI:
2
3 compilare in .jar il file KotlinSource/BiblioSource.kt es:
4 [path_To_kotlinc]/kotlinc -jar BiblioSource.kt -jar Biblio.jar -include-runtime
5
1 ISTRUZIONI:
2
3+
4 compilare in .jar il file KotlinSource/BiblioSource.kt es:
5 [path_To_kotlinc]/kotlinc -jar BiblioSource.kt -jar Biblio.jar -include-runtime
6


OUTLINE
TIMELINE
master*
2↓ 1↑
0
0

Add everything!

Nella directory root del project\repo

git add .

oppure

git add -A

Le successive istruzioni di commit & push riguarderanno TUTTI i file aggiunti\modificati a partire dalla root del progetto.

E’ una buona idea?

Non sempre

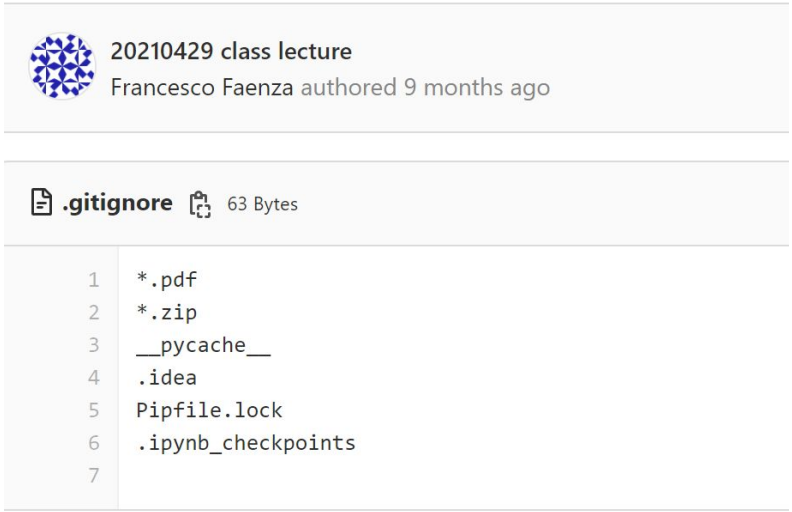
Sicuramente comodo perchè così evito di dovermi ricordare quali e quanti file ho effettivamente modificato.

Del resto però rischio di fare commit\push di file che non dovrebbero essere inviati per ovvi motivi, e.g. binari compilati, file oggetto ed altri “sotto-prodotti” delle operazioni di compilazione\interpretazione…

.gitignore

E’ un file di testo che dettaglia le regole attraverso le quali cartelle, file etc… verranno ignorati dai comandi di tracciamento git.

E’ normalmente posto nella root del progetto



La cosa più importante

Regole per i messaggi di commit

Le sette regole

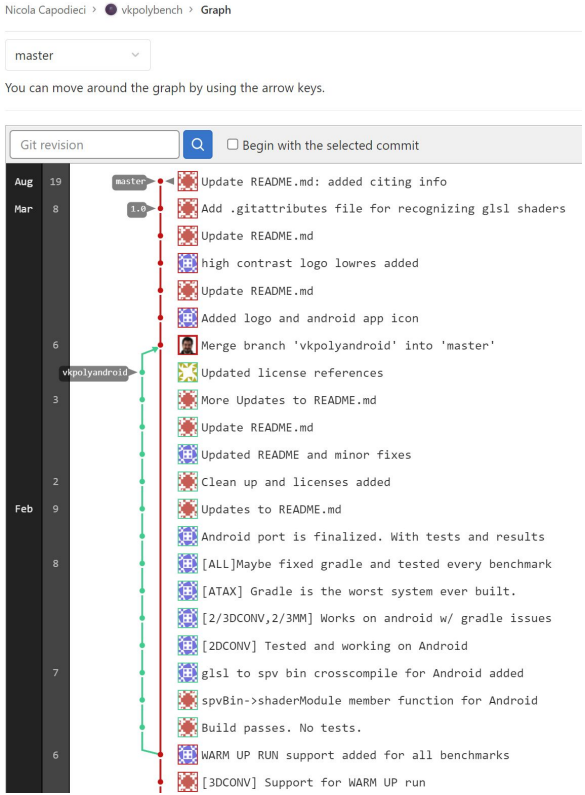
Keep in mind: This has all been said before.

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain what and why vs. how

Regola d’oro generica per i commit

Non accorpare modifiche tra loro non correlate nello stesso commit. **Si usino diversi commit.**

Predicare bene…



Esempio

Si pensi di stare sviluppando una **web app.**

In un unica sessione di coding:

- Modifico la parte di interfacciamento al DB
- Modifico la parte di presentazione HTML\CSS\template vari…
- Modifico script di inizializzazione del mio web server.

Sarebbe opportuno in questo caso separare queste modifiche in tre commit diversi…

Link utili

https://education.github.com/git-cheat-sheet-education.pdf

https://rogerdudler.github.io/git-guide/