

Guide pour l'IDE PyCharm

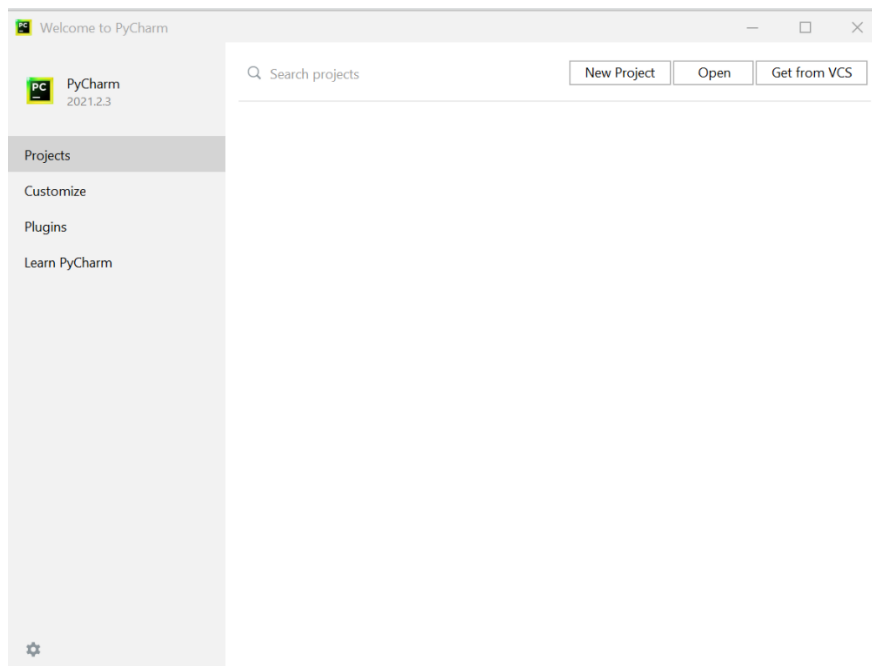
Objectif : découvrir l'environnement de travail PyCharm

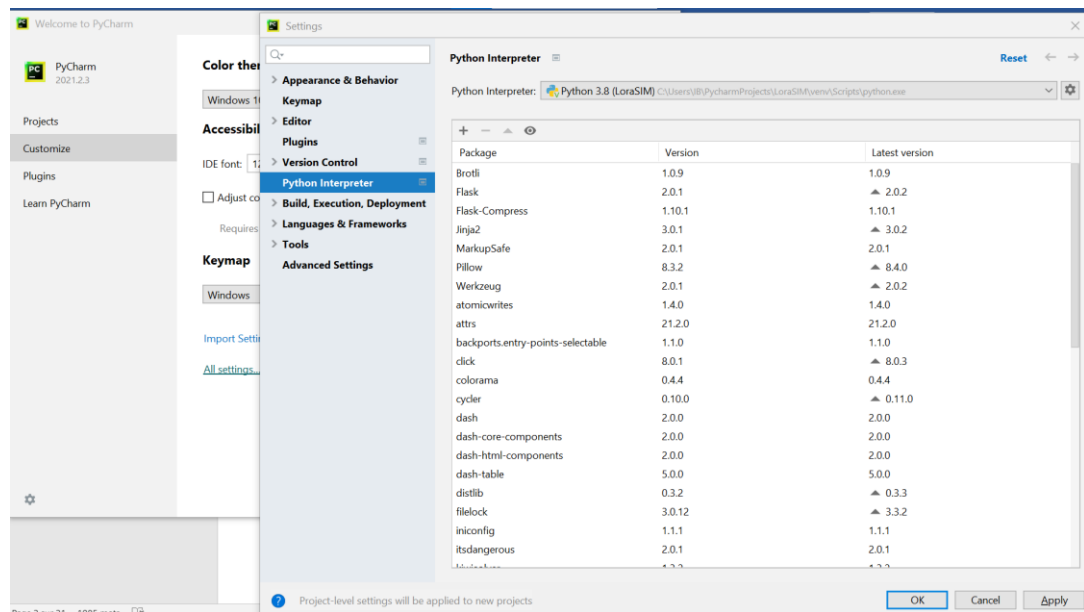
I. Introduction et mise en place de l'IDE

PyCharm est un IDE (Integrated Development Environment). Il s'agit d'un logiciel permettant d'intégrer dans une même fenêtre tous les éléments utiles à la programmation en python à savoir : un éditeur de texte pour écrire des scripts, une console pour exécuter des programmes, ainsi qu'un explorateur de fichiers pour parcourir le projet en cours.

Configuration

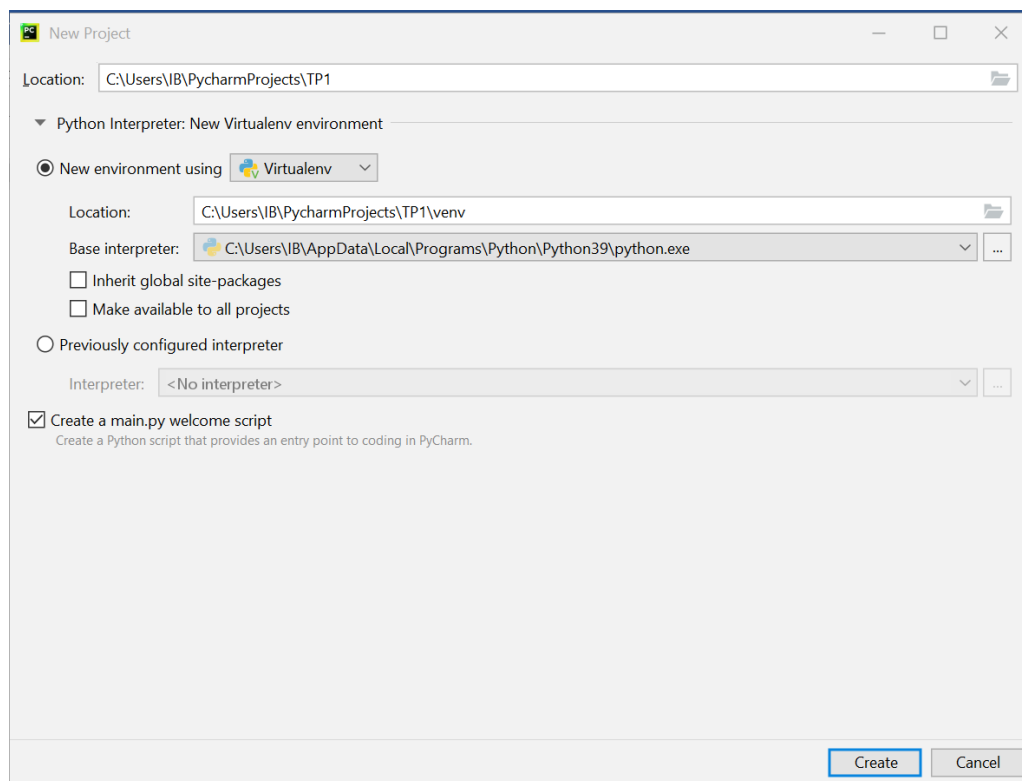
Lors de votre première utilisation de PyCharm, vous pouvez vérifier / modifier quelques paramètres notamment : l'interpréteur à utiliser, le dossier de travail. Pour cela, après avoir lancé l'IDE, aller vers '**Customize**' puis '**All settings**', puis '**Python interpreter**' et sélectionner dans le menu déroulant la version 3.X installée. Il est normal de ne pas observer la même liste dans l'exemple de la figure ci-dessous.



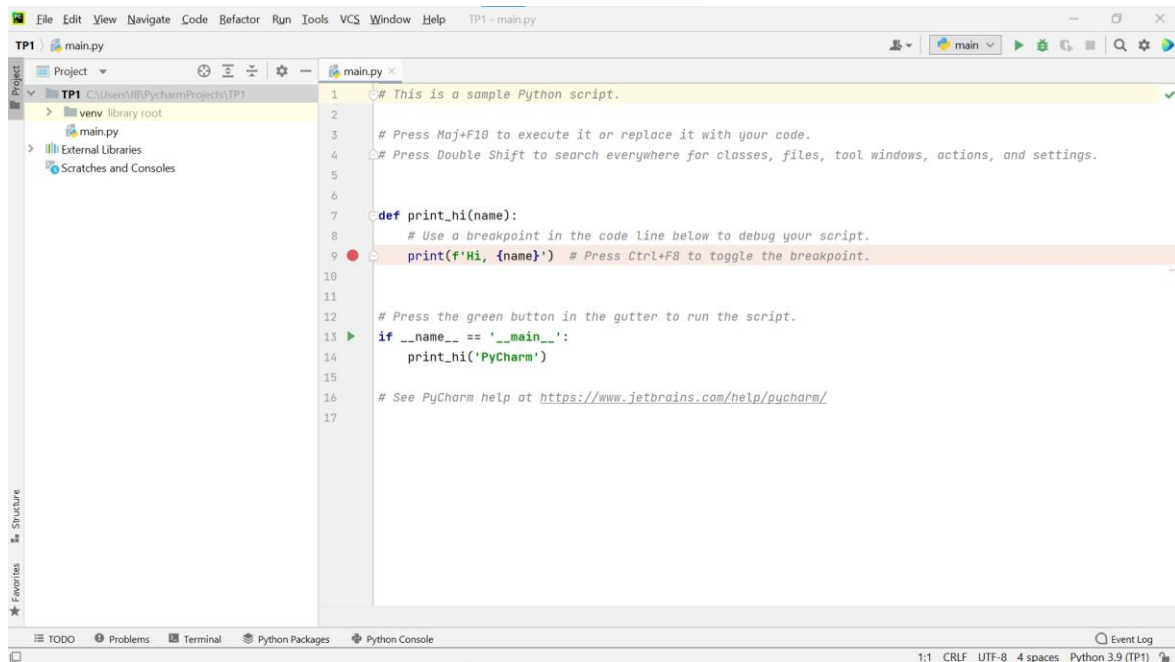


Créer votre projet

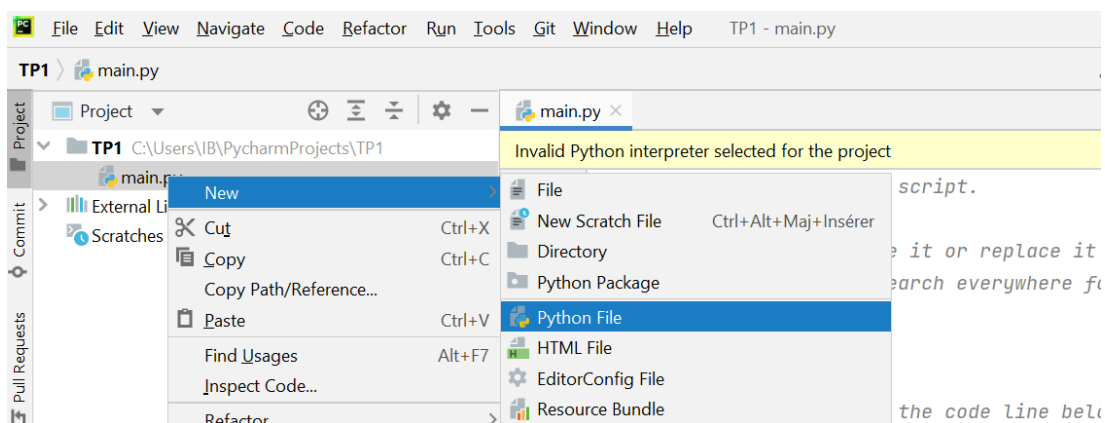
Maintenant vous allez créer votre premier projet, pour cela appuyez sur New Project



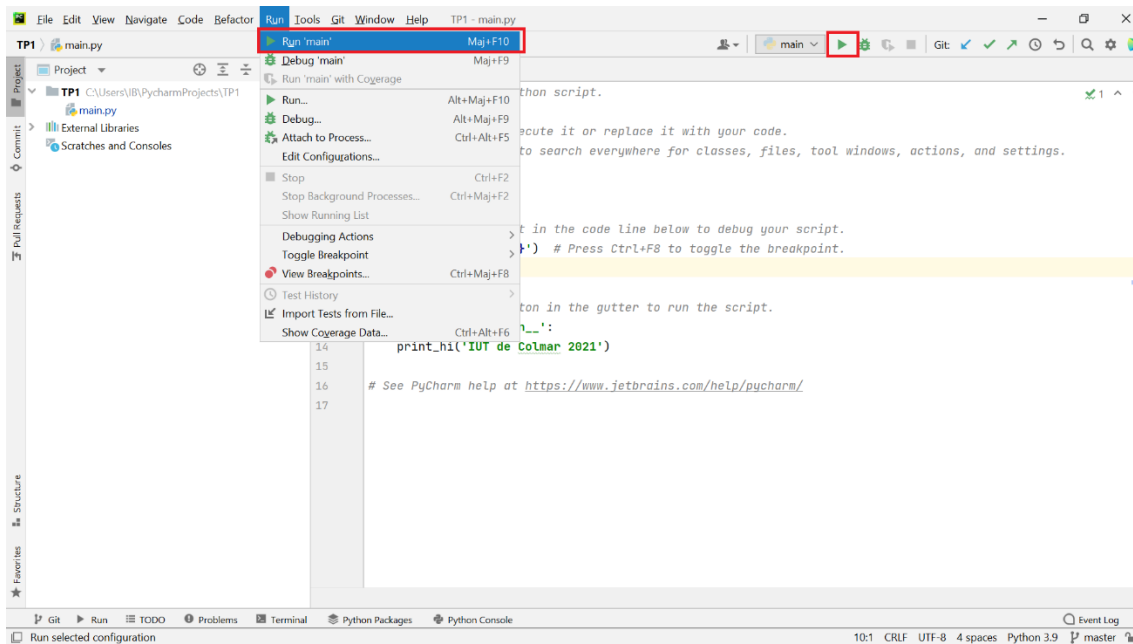
- ✓ Sélectionnez à nouveau le chemin vers votre exécutable python à l'aide de la flèche dans la ligne '**Base interpreter**'. Sélectionnez ensuite le dossier qui va contenir le projet. Créer un dossier avec votre nom et prénom afin que ça soit votre espace de travail. Une fois créer, vous arrivez sur votre projet avec un fichier main.py créer par défaut.



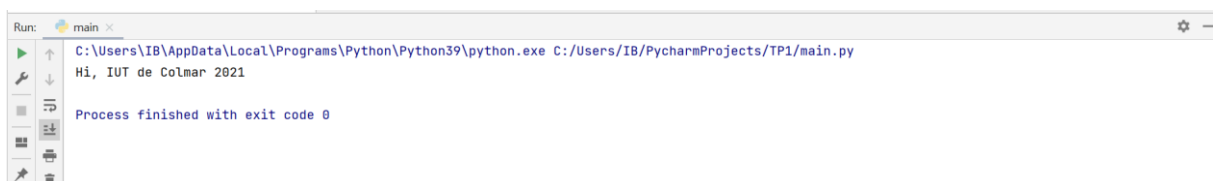
- ✓ Vous pouvez aussi, une fois arrivé sur l'écran principal de PyCharm, créer votre premier script python. Pour cela, en haut à gauche, cliquez sur **File > New... > Python File**. Le fichier sera automatiquement placé dans le dossier de votre projet.



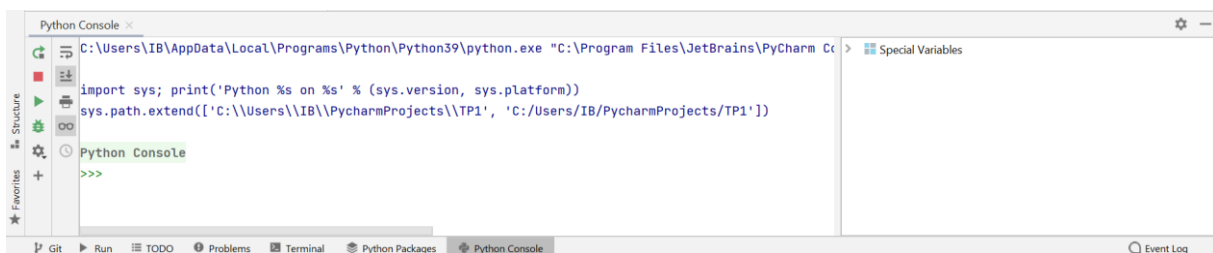
- ✓ Pour exécuter votre script, allez dans l'onglet **Run** puis cliquez sur le fichier à exécuter.



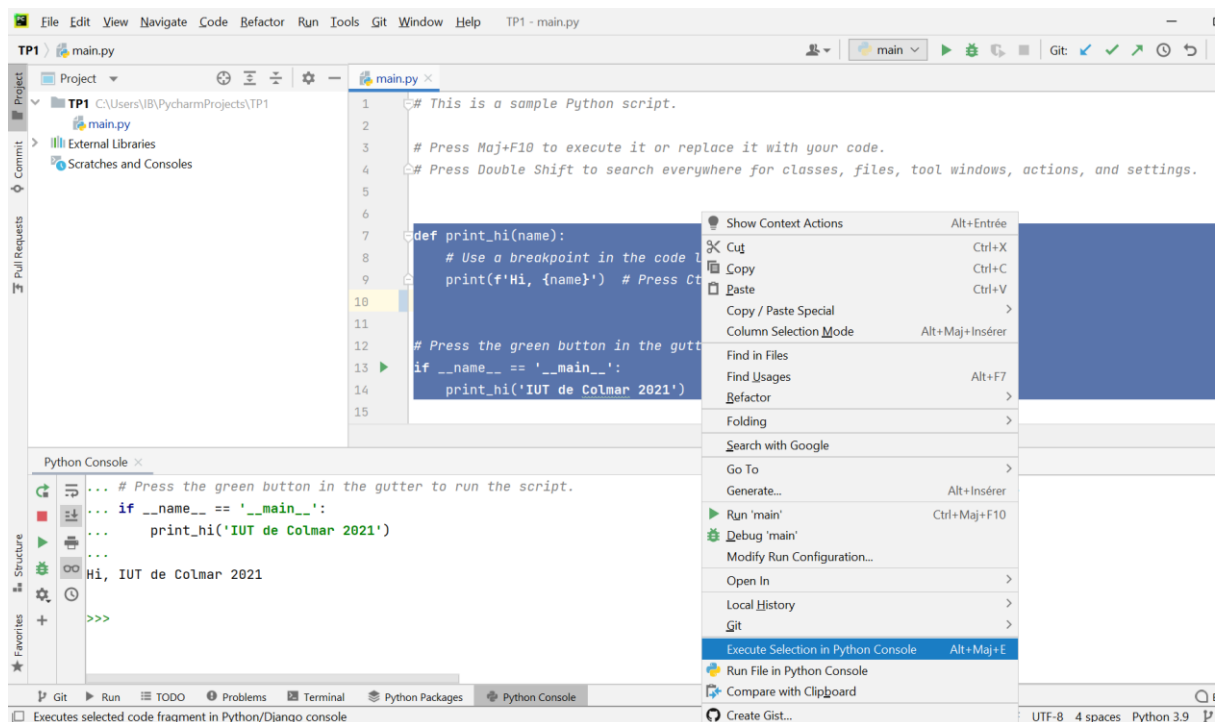
- ✓ Une nouvelle zone apparaît. Il s'agit de la console. C'est ici que vous observerez les résultats de vos scripts. Par défaut, l'output d'un script est renvoyé dans la console Run.



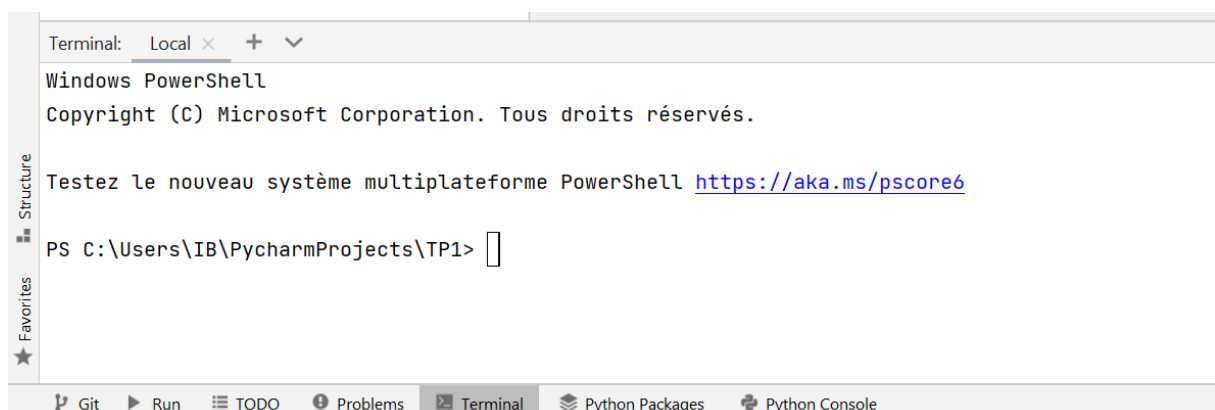
- ✓ Vous pouvez aussi utiliser la console python pour effectuer des tests indépendamment de ce que vous mettez dans vos scripts.



- ✓ Il est possible d'exécuter une partie de votre script dans la console. Pour cela, sélectionnez les lignes à exécuter, faites un clic droit et appuyez sur **Execute Selection in Console**. Faire cela produit le même effet que si vous aviez tapé toutes les instructions sélectionnées dans la console.

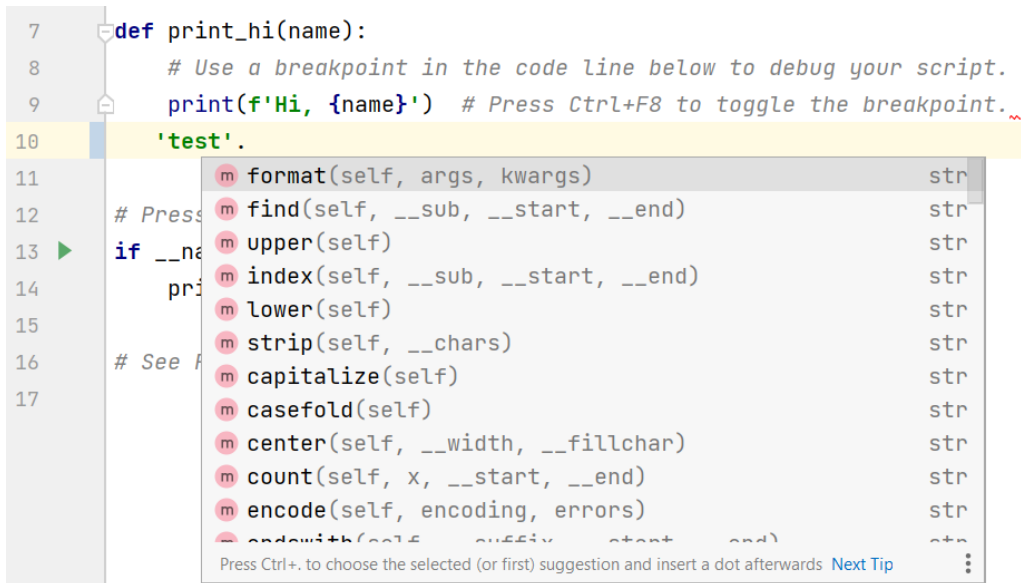


✓ Vous avez aussi accès à un terminal de commande de votre système d'exploitation.

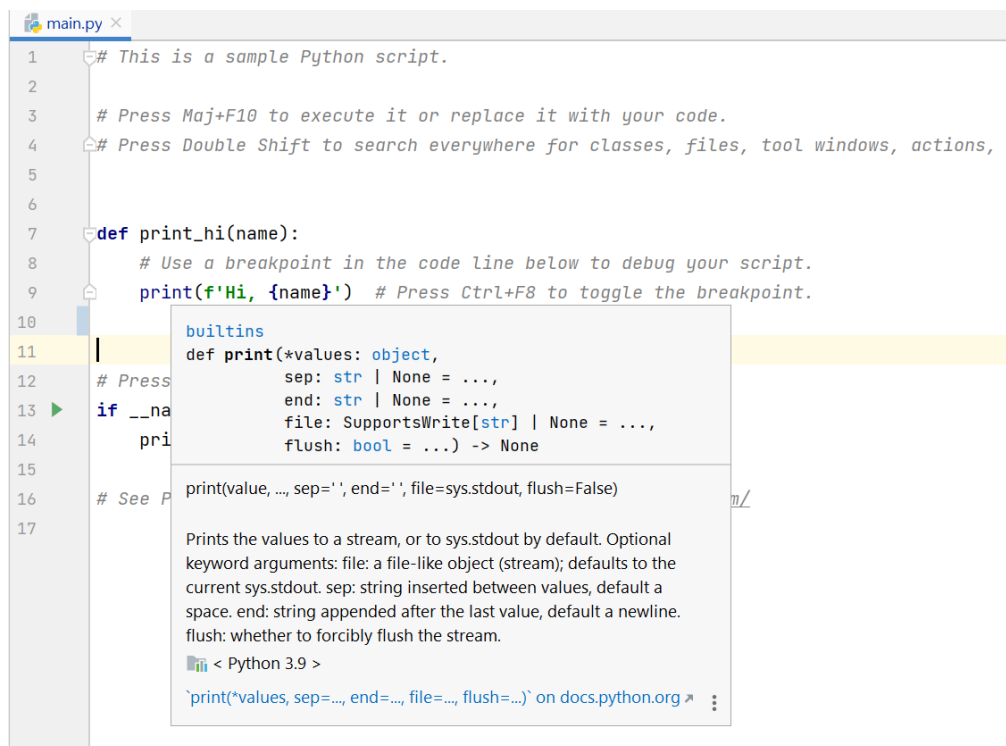


II. Astuces

✓ PyCharm propose un système d'**auto-complétion** qui permet de faciliter l'utilisation de nombreux modules. Lorsque vous écrivez, le logiciel vous propose les différentes possibilités selon ce que vous écrivez. Les différentes variables et fonctions contenues dans un module vous sont aussi proposées. Il vous est donc possible de choisir une des propositions et celle-ci sera écrite automatiquement.



- ✓ PyCharm propose également d'avoir accès à de la documentation concernant une variable ou une fonction. Pour cela, placez votre curseur sur un élément et une fenêtre popup s'affichera avec la documentation. Vous pouvez aussi aller dans l'onglet View > Quick Documentation ou bien avec le raccourci Ctrl+Q :



- ✓ Avec PyCharm vous pouvez également accéder à la définition d'une fonction ou d'une variable. Pour cela, placez votre curseur sur un élément et allez dans **View > Quick Definition** (ou bien avec le raccourci **Ctrl+Maj+I**)

```

1 # This is a sample Python script.
2
3 # Press Maj+F10 to execute it or replace it with your code.
4 # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings
5
6
7 def print_hi(name):
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press
13 if __na
14     pri
15
16 # See P
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

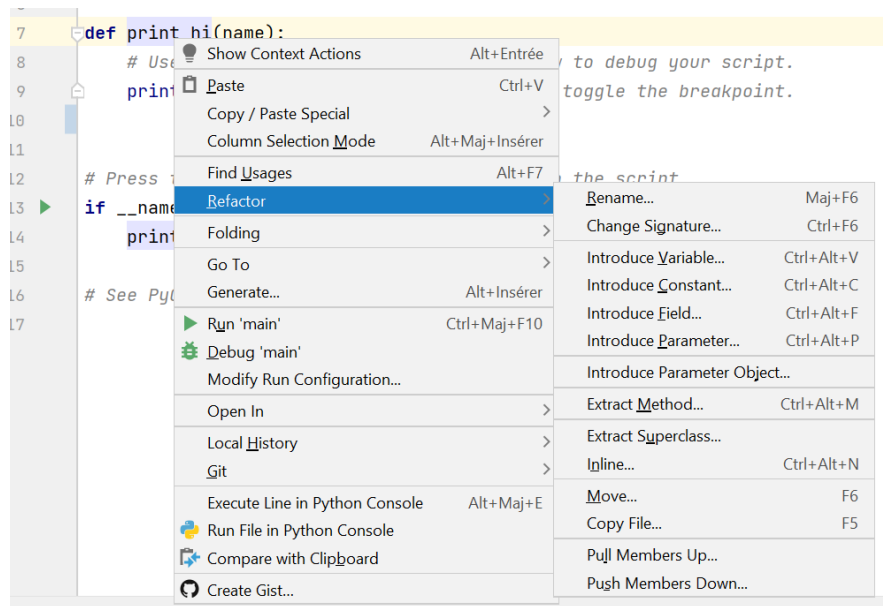
- ✓ Vous pouvez obtenir un accès complet au fichier où l'élément est défini. Pour cela, utilisez le raccourcis **CTRL + Clic gauche** sur Windows (ou **CMD + Clic gauche** sur MAC) en cliquant sur l'élément souhaité. Le fichier contenant la définition de l'élément s'ouvrira dans l'éditeur de texte. Cela est très utile si vous voulez retrouver et changer la valeur d'une variable ou la définition d'une fonction.

```

480 * def pow(*args, **kwargs): # real signature unknown
481     """
482     Equivalent to base**exp with 2 arguments or base**exp % mod with 3 arguments
483
484     Some types, such as ints, are able to use a more efficient algorithm when
485     invoked using the three argument form.
486     """
487     pass
488
489 * def print(self, *args, sep=' ', end='\n', file=None): # known special case of print
490     """
491     print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
492
493     Prints the values to a stream, or to sys.stdout by default.
494     Optional keyword arguments:
495     file: a file-like object (stream); defaults to the current sys.stdout.
496     sep: string inserted between values, default a space.
497     end: string appended after the last value, default a newline.
498     flush: whether to forcibly flush the stream.
499     """
500     pass

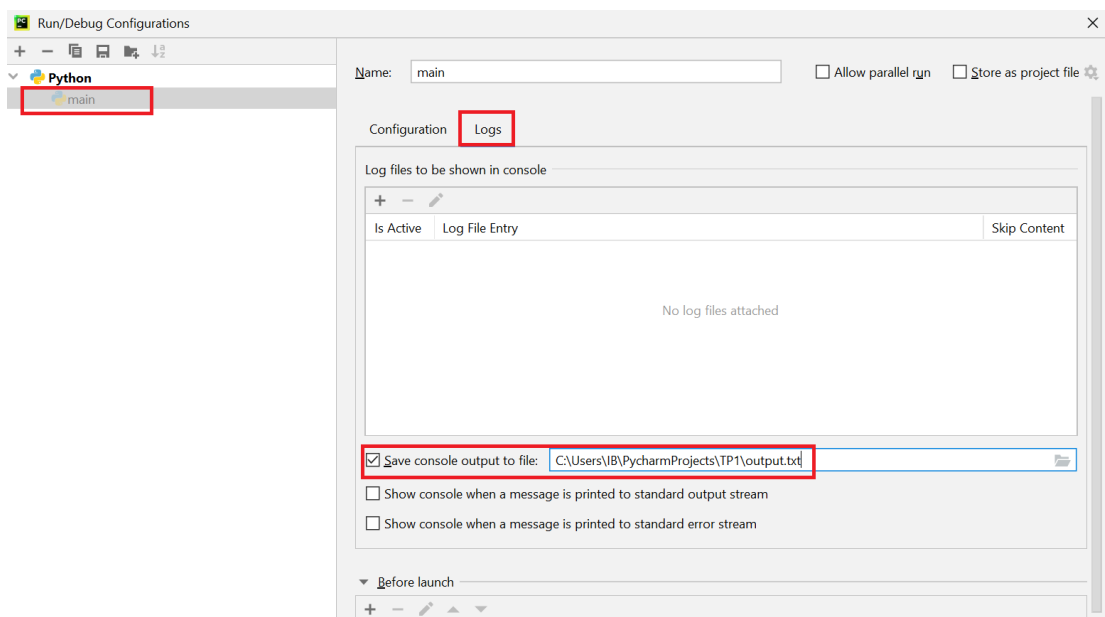
```

- ✓ PyCharm permet d'utiliser des changements appelés '**Refactoring**' qui permettent de modifier facilement votre code. Pour cela, faites un clic droit sur un élément et affichez les Refactors. Selon l'élément sur lequel vous cliquez, différents refactorings seront disponibles. Sur une variable, vous pouvez par exemple la **renommer** : cela la changera dans tous vos scripts.



III. Redirection de l'input et de l'output standard

Par défaut le résultat d'un script (son output) est renvoyé dans la console Run. Il vous est aussi possible de rediriger l'output dans un fichier. Pour cela, allez dans l'onglet **Run > Edit Configuration**. Allez ensuite dans l'onglet **Logs** et cochez la case **Save console output to file** en choisissant le nom du fichier dans lequel vous voulez enregistrer l'output.



Il vous est aussi possible de faire ces redirections en utilisant le terminal de commande.


```
Terminal: Local x + v
PS C:\Users\IB\PycharmProjects\TP1> python main.py > output.txt
PS C:\Users\IB\PycharmProjects\TP1> ls

Répertoire : C:\Users\IB\PycharmProjects\TP1

Mode                LastWriteTime         Length Name
----                -
d-----          07/11/2021    00:03             .idea
-a----          06/11/2021    23:57           553 main.py
-a----          07/11/2021    14:01           50  output.txt
```

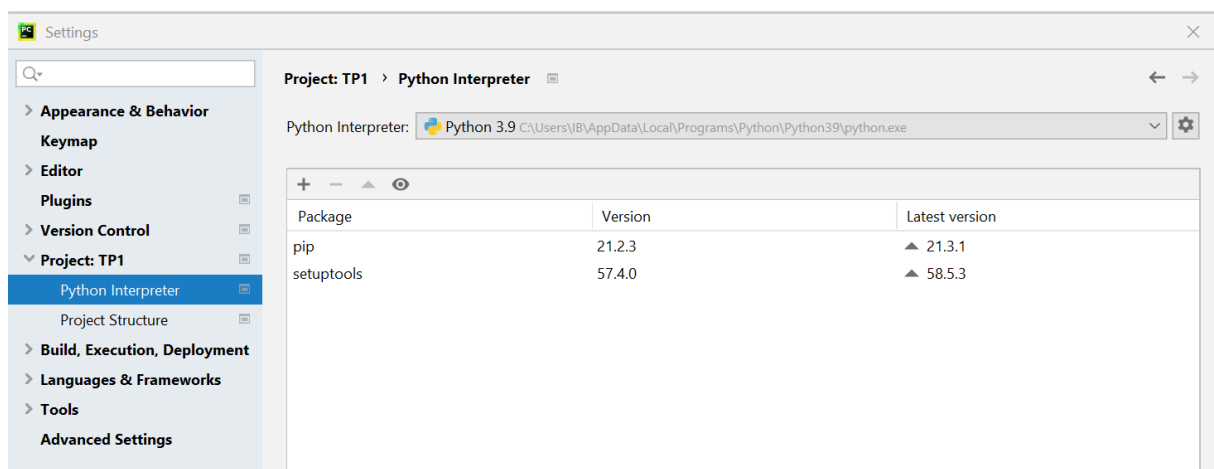
Cette commande permet de dire à la machine d'exécuter **main.py** avec le programme Python, de rediriger l'output sur le fichier output.txt (> **output.txt**).

Noter qu'il est aussi possible de transmettre en input standard un fichier par exemple input.txt avec '< **input.txt**', et aussi de rediriger les erreurs dans un fichier par exemple error.txt avec '2> **error.txt**'.

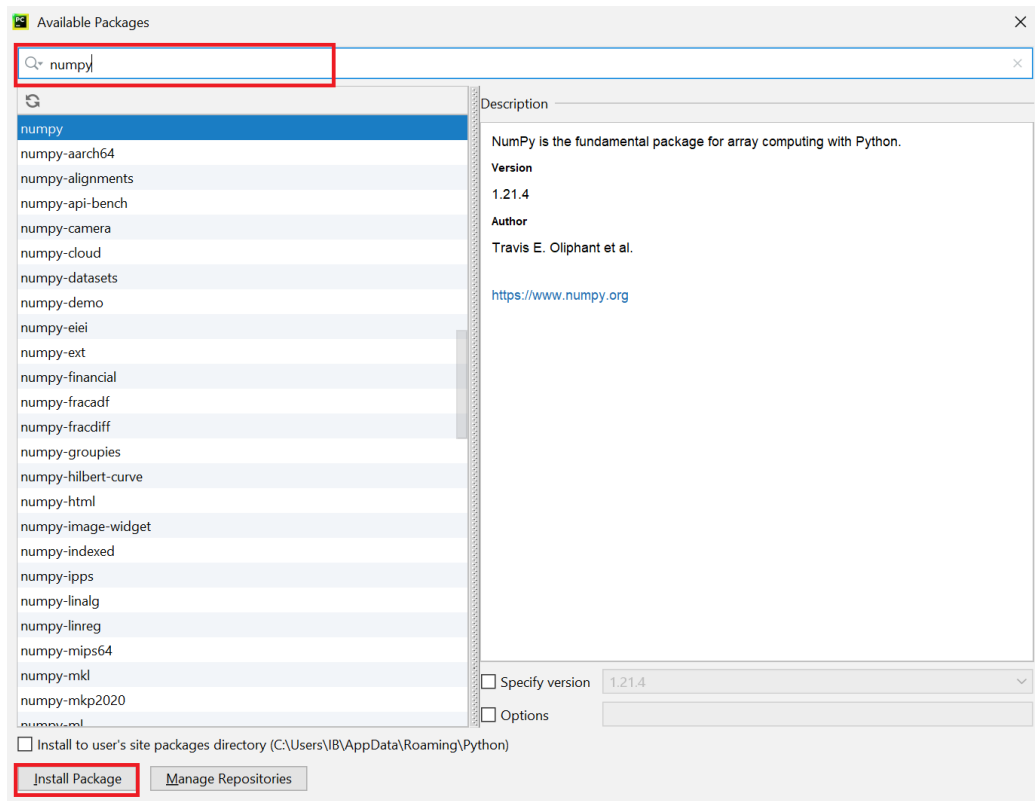
IV. Installer des modules

PyCharm permet d'installer et de gérer des modules python à l'aide du module **pip**. Pour cela, ce dernier doit être installé sur la machine.

Dans PyCharm allez dans le menu **File > Settings**. Allez ensuite dans l'onglet **Project > Project interpreter**. Ici s'affiche la liste des modules déjà installés.



Appuyez sur le **plus** en haut à droite pour ajouter un module. Une fenêtre s'affiche où vous pouvez rechercher un module et l'installer. Vous pouvez aussi spécifier la version du module ainsi que des options si nécessaire.



Si vous travaillez sur un ordinateur de la salle informatique, vous devez indiquer l’option **–user** afin d’installer le module seulement pour l’utilisateur connecté.

Une documentation des commandes disponibles avec pip est accessible à l’adresse <https://pip.pypa.io/en/stable/>

Vous pouvez aussi installer les modules en utilisant le **terminal** de l’ordinateur.

V. Git & Github

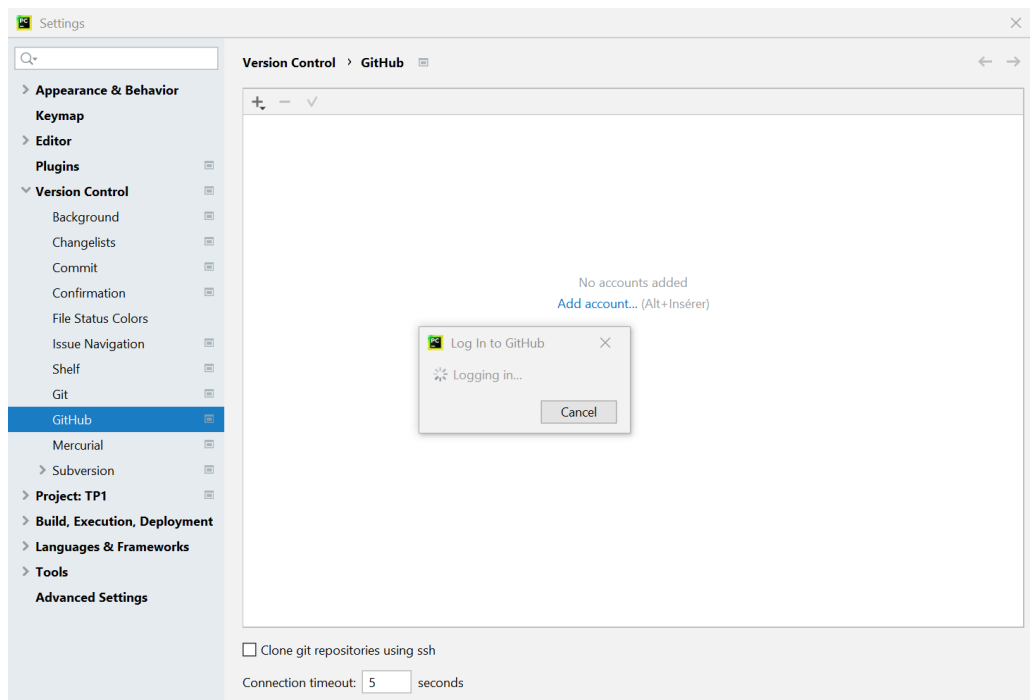
Git est un gestionnaire de versions qui permet aux développeurs de conserver un historique des modifications et des versions de tous leurs fichiers. Vous pouvez l’utiliser pour créer un dépôt local et gérer les versions de vos fichiers. Ainsi, Git permet de garder en mémoire :

- Chaque modification de chaque fichier ;
- Pourquoi elle a eu lieu ;
- Par qui !

GitHub est un service en ligne qui va héberger votre dépôt. Dans ce cas, on parle de **dépôt distant** puisqu’il n’est pas stocké sur votre machine.

PyCharm propose l’utilisation de Git et Github pour le suivi de versions. Dans le cadre des TP de cette ressource (R107), vous allez plutôt utiliser Github pour pouvoir sauvegarder vos codes dans un dépôt distant et de pouvoir les récupérer en cas de besoin. Pour cela il faut connecter votre compte Github (créer-le si vous ne l’avez pas encore, voir la partie VI) avec votre IDE PyCharm.

Commencer par aller dans le menu **File > Settings** puis dans l’onglet **Version Control > GitHub** puis le + associer votre compte GitHub



Vous serez redirigé vers une page d'authentification, accepter et continuer.

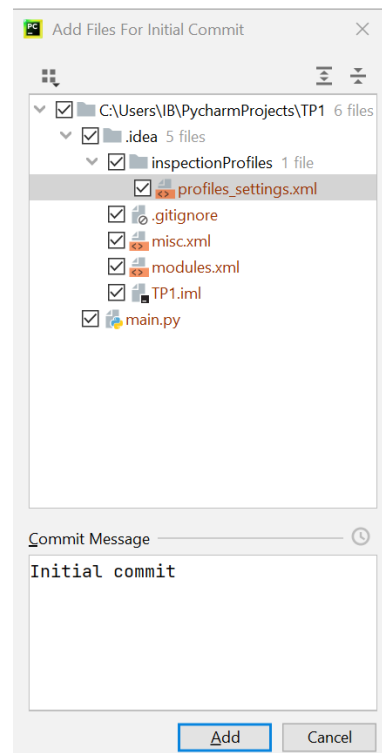
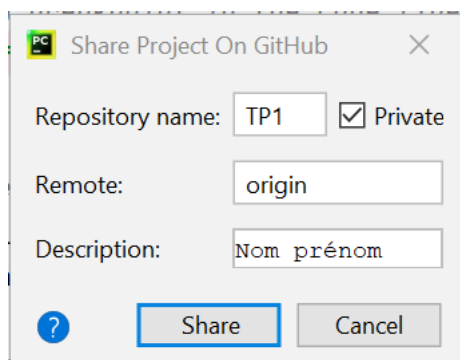
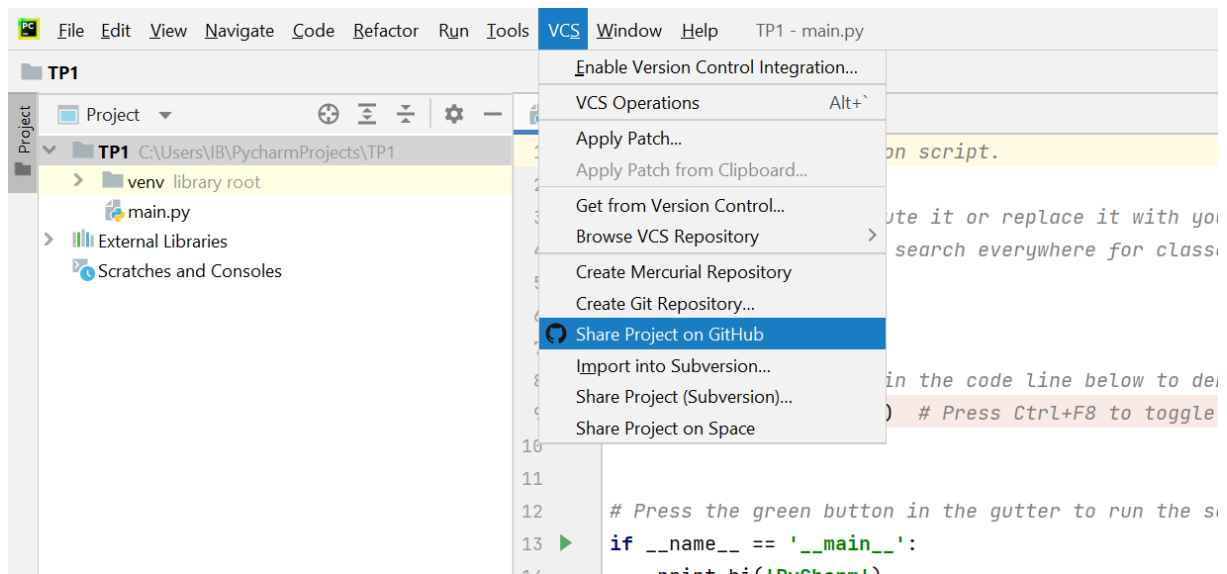


Please continue only if this page is opened from a [JetBrains IDE](#).

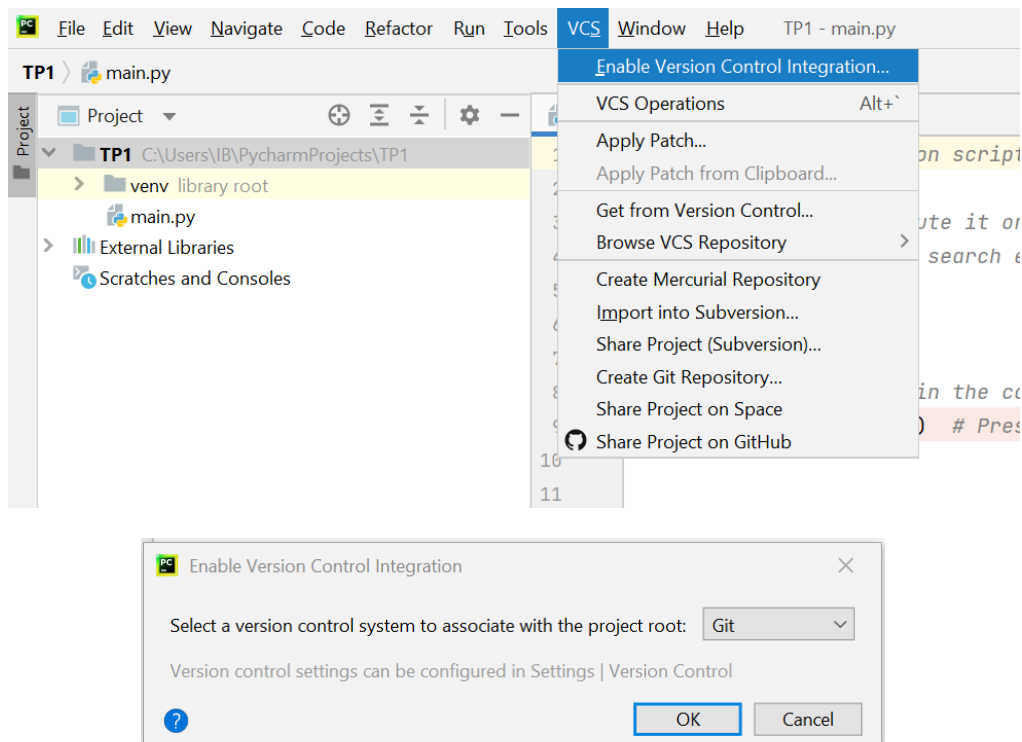
[Authorize In GitHub](#)

- Vous seriez amené à répéter cette étape au début de chaque séance afin d'associer votre compte GitHub à l'IDE et pouvoir récupérer vos codes. Idem à la fin de la séance afin de déconnecter votre compte de l'IDE.

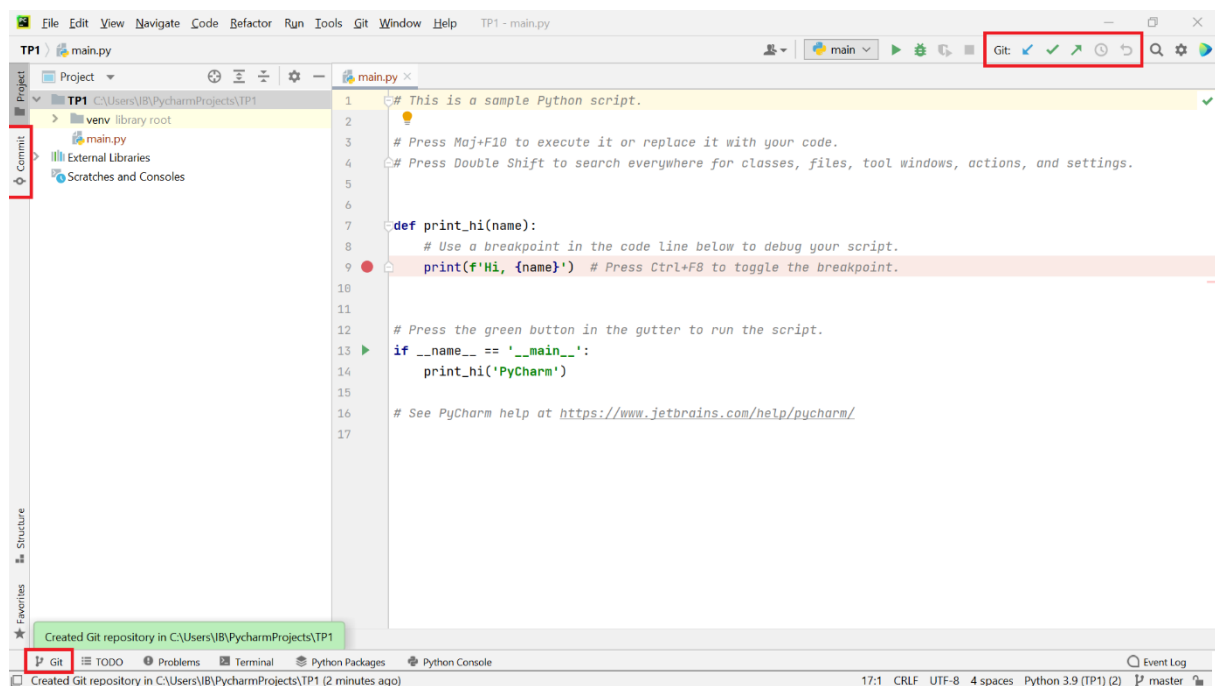
Maintenant vous pouvez charger votre projet sur votre espace GitHub. Pour cela aller dans le menu **VCS > Share Project on GitHub**. Une fenêtre va s'ouvrir en vous demandant le nom du '**Repository**' et la description (renseigner par exemple votre nom et prénom). Cochez aussi '**Private**'. Ensuite vous pouvez sélectionner les éléments de votre projet auxquels vous voulez appliquer un '**Commit**'. Une fois les éléments sélectionnés, dans la partie 'Commit Message' ajouter un commentaire qui décrit la version du projet que vous êtes en train de charger.




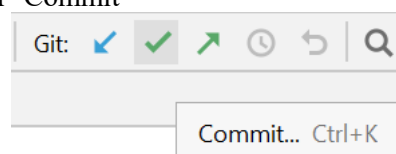
Afin de faciliter le suivi de version et le chargement du code sur GitHub vous pouvez activer le mode suivi de version sur PyCharm. Pour cela aller dans le menu **VCS > Enable Version Control Integration**. Sélectionner après 'Git' comme système de suivi de version.

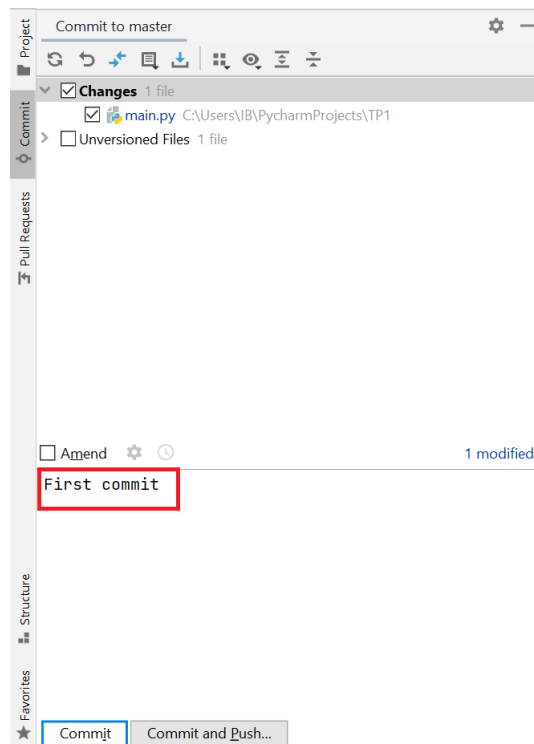


Une fois ces modifications appliquées, des options seront rajoutées au niveau de votre IDE comme illustrer dans la figure suivante :

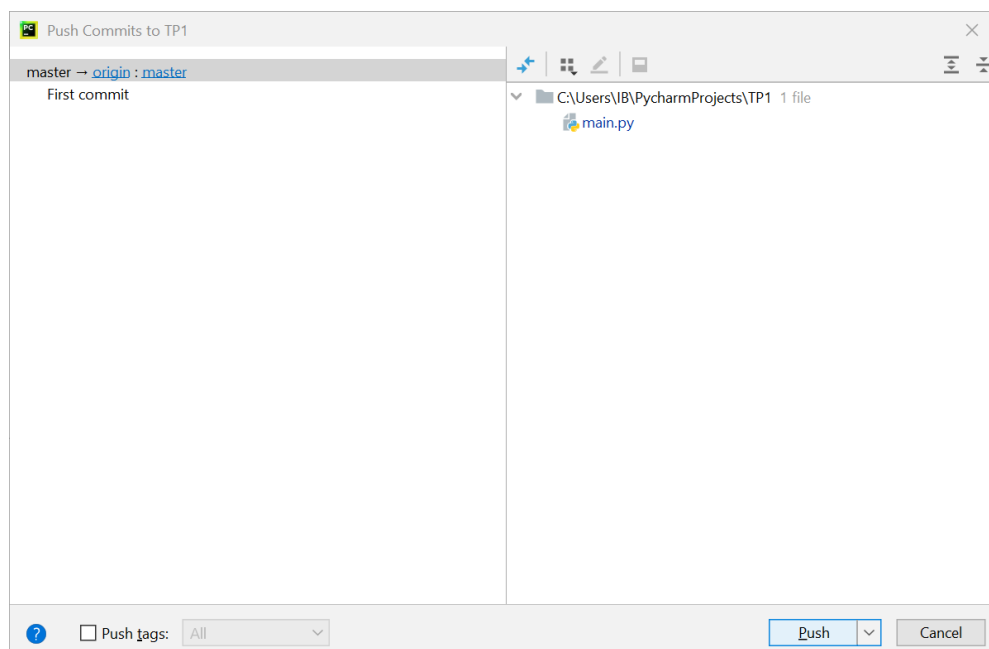
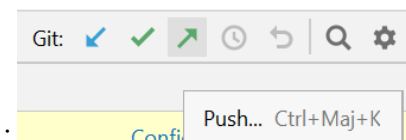


Maintenant vous pouvez faire un 'Commit' directement avec le bouton  puis préciser le commentaire et en fin appuyer sur 'Commit'





Il reste maintenant à recharger les modifications sur votre compte :

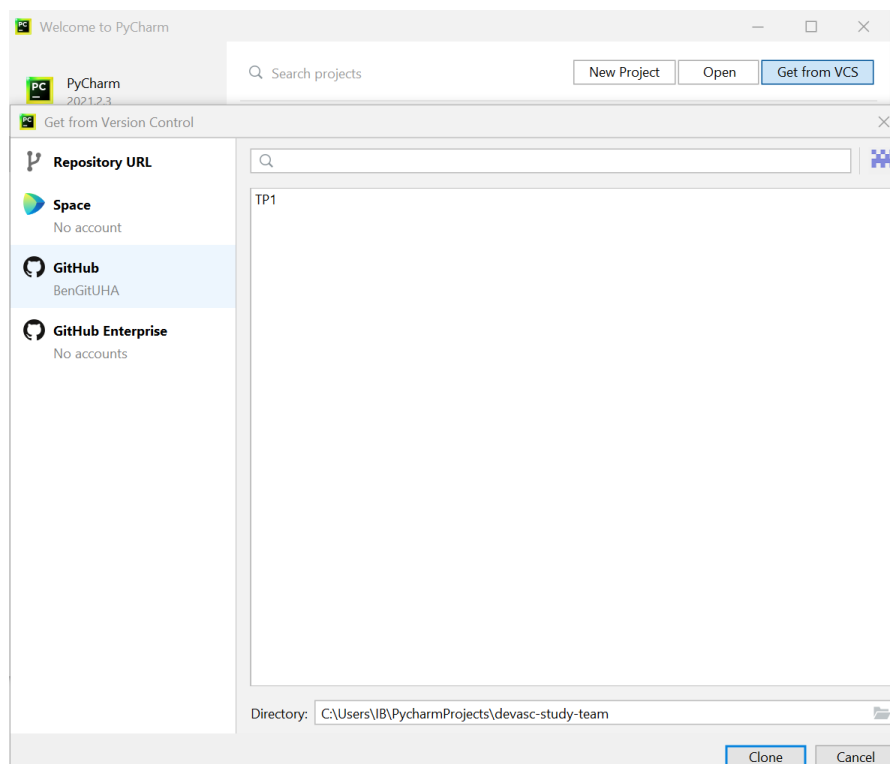


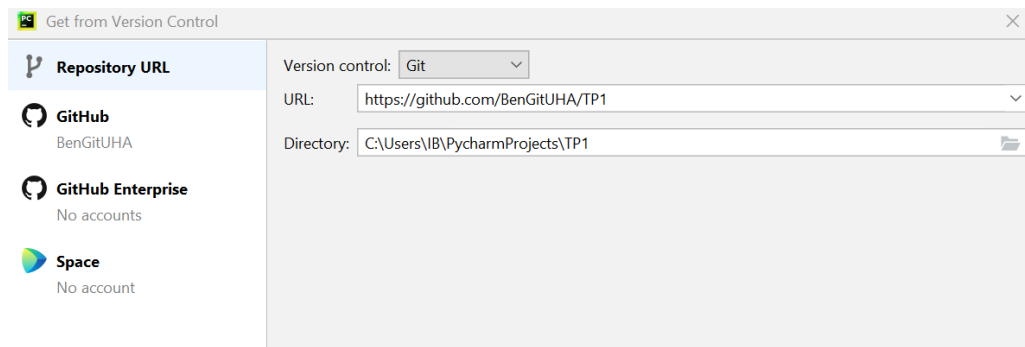
Vous pouvez vérifier la présence de vos modifications sur votre compte :

```
17 lines (12 sloc) | 672 Bytes

1  # This is a sample Python script.
2
3  # Press Maj+F10 to execute it or replace it with your code.
4  # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7  def print_hi(name):
8      # Use a breakpoint in the code line below to debug your script.
9      print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
11     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
12
13 # Press the green button in the gutter to run the script.
14 if __name__ == '__main__':
15     print_hi('IUT de Colmar 2021')
16
17 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

Maintenant si vous devez importer votre projet depuis votre compte GitHub suivez les étapes suivantes : après lancement de l'IDE sélectionner '**Get from VCS**'. Après soit vous précisez l'URL de votre projet, soit vous sélectionnez le projet à charger depuis la liste des projets chargée depuis votre compte GitHub (à condition que vous soyez connectés).





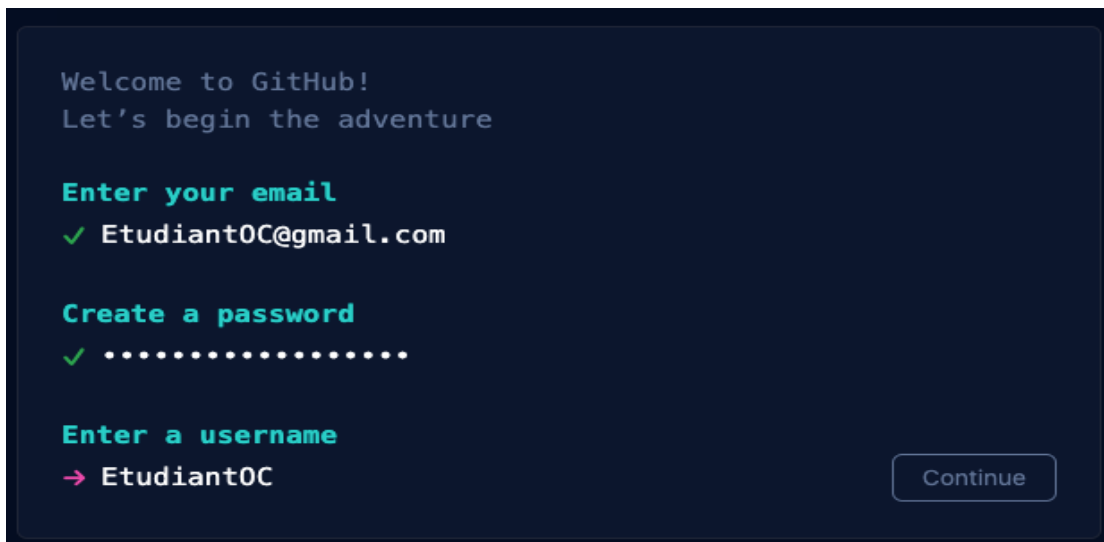
- ✚ Encore une fois, l'utilisation de GitHub, dans le cadre de la ressource R107, se résumera à sauvegarder vos codes sur votre compte et de pouvoir y avoir accès en cas de besoin.

VI. Création de compte sur Github


- ✓ Allez tout simplement sur <https://github.com>.
- ✓ Cliquez sur Sign up.



- ✓ Saisissez une adresse électronique et un mot de passe.
- ✓ Choisissez un nom d'utilisateur qui n'est pas déjà pris.
- ✓ Cliquez sur « Continue ».
- ✓ Un code de vérification vous sera envoyé sur votre adresse e-mail afin de confirmer votre identité.
- ✓ Et voilà, vous êtes à présent inscrit sur GitHub ! Par défaut, GitHub est gratuit.



Note : GitHub fournit presque toutes ses fonctionnalités pour les comptes gratuits, à l'exception de quelques fonctions avancées.

- ✓ En cliquant sur le logo  Octocat (logo en forme de chat) dans le coin supérieur gauche de l'écran, vous accéderez à votre tableau de bord. Vous êtes maintenant prêt à utiliser GitHub.


L'interface repository

- ✓ L'interface Repositories est l'emplacement où vous pourrez créer et retrouver vos dépôts existants.
- ✓ Pour mettre votre projet sur GitHub, vous devez créer un repository (ou dépôt en français) dans lequel il pourra être installé.
- ✓ Cliquez sur le "+" dans le coin supérieur droit, pour faire apparaître l'option New repository.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 Mustafasamara ▾

Repository name *

/ Python ✓

Great repository names are short and memorable. Need inspiration? How about [friendly-octo-spork?](#)

Description (optional)

Python - Programmation - TP - Devoirs

☐  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

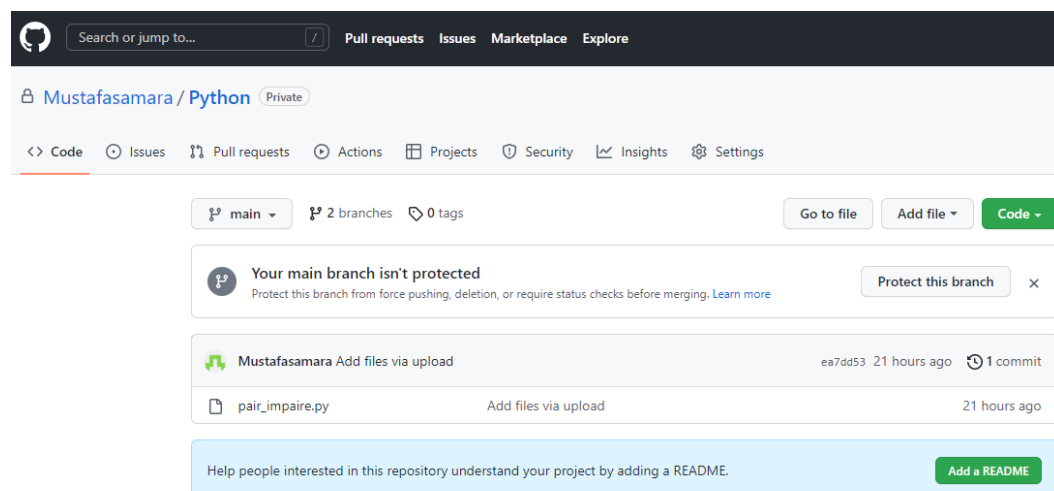
License: None ▾

 You are creating a private repository in your personal account.

Create repository

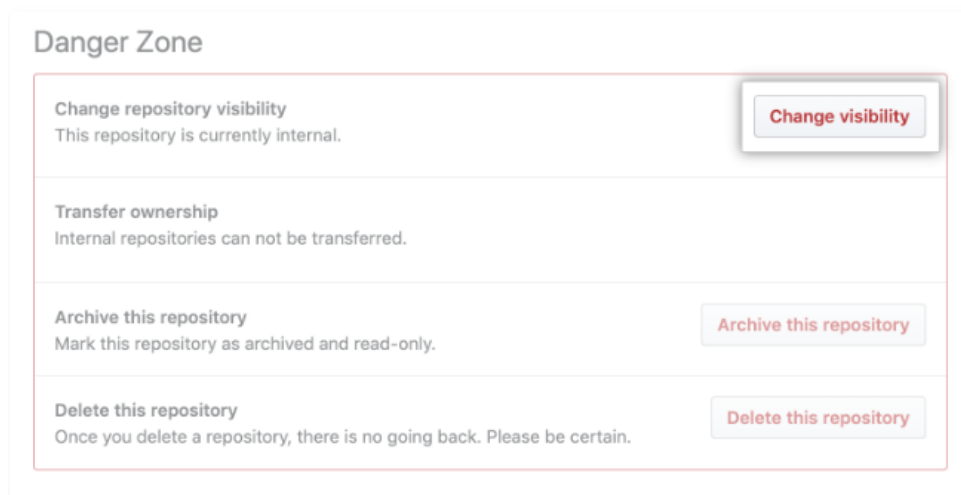
Visibilité du dépôt « Repository »

- ✓ Sous le nom de votre « Repository », cliquez sur « Setting »



The screenshot shows the GitHub interface for a repository named 'Python' owned by 'Mustafasamara'. The repository is marked as 'Private'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. The main content area shows the repository's main branch 'main' with 2 branches and 0 tags. A warning message states 'Your main branch isn't protected' and suggests protecting it. Below this, there is a commit history table showing a commit by 'Mustafasamara' titled 'Add files via upload' with the commit hash 'ea7dd53' and a timestamp of '21 hours ago'. The commit message is 'pair_impair.py'. At the bottom, there is a prompt to 'Add a README' to help people understand the project.

- ✓ Sous « Danger Zone », à droite de « Change repository visibility », click Change visibility.



✓ Vous pouvez aussi gérer l'accès à vos « Repositories »

✓ Cliquez sur « Add people »

