

Atom: A Hackable Editor for the 21st Century

Jerry Feldman



BLU 2/19/2020



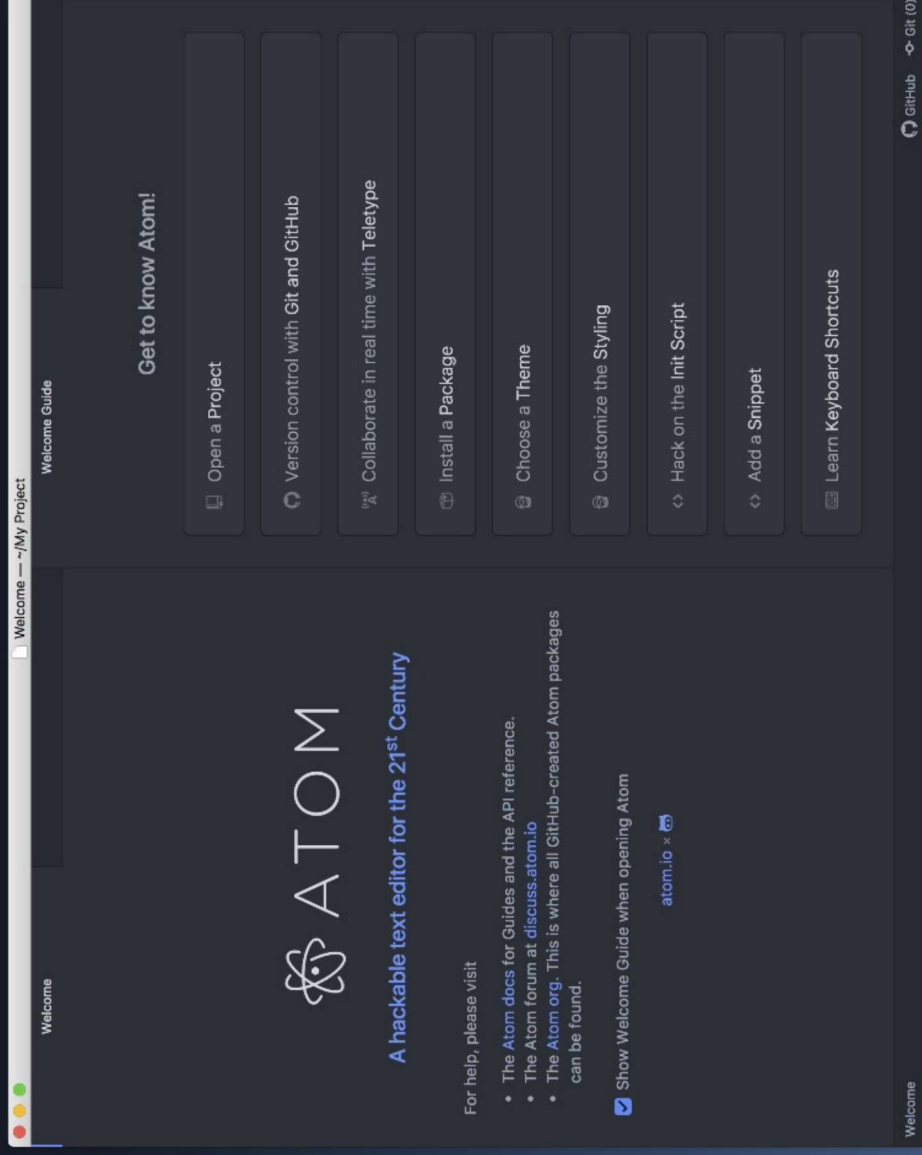
Overview

- First I will display some information from the Atom Flight Manual.
- Next I will display some screen shots with some of the screen elements that are hard to see in the demo
- Third I will demo a couple of Python scripts I have been working on to show many of the useful features.



The Atom Welcome Screen

This is what you will see when you first run Atom. You can bring up the welcome screen at any time



Atom: A Hackable Editor for the 21st Century

The Atom Flight Manual

- The Atom Flight Manual is an extensive guide in how to use atom. This will tell you all you need to know about atom:
- <https://flight-manual.atom.io/>



How to get Atom

- Atom can be downloaded directly from: <https://flight-manual.atom.io/getting-started/s/sections/installing-atom/>
- The flight manual has detailed instructions that can be copied and pasted.



Why I Chose Atom

- As I mentioned in the abstract, I am a long time EMACS user going back over 30 years. I used EMACS mainly for C, C++, and more recently TCL and Python.
- When I moved to Red Hat, I found there were many things that EMACS failed to do with my Python code, such as trim trailing spaces and syntax check.
- Many on my team used Eclipse which I hate.



Why I Chose Atom

- I looked at Geany, but IMHO it was somewhat light weight for me. One of the main features I used in Emacs was to be able to display files in different windows, both horizontally and vertically.
- Pycharm was a better IDE, but to my knowledge it did not do panes.
- Another developer suggested Atom. Atom pretty much did what I wanted.

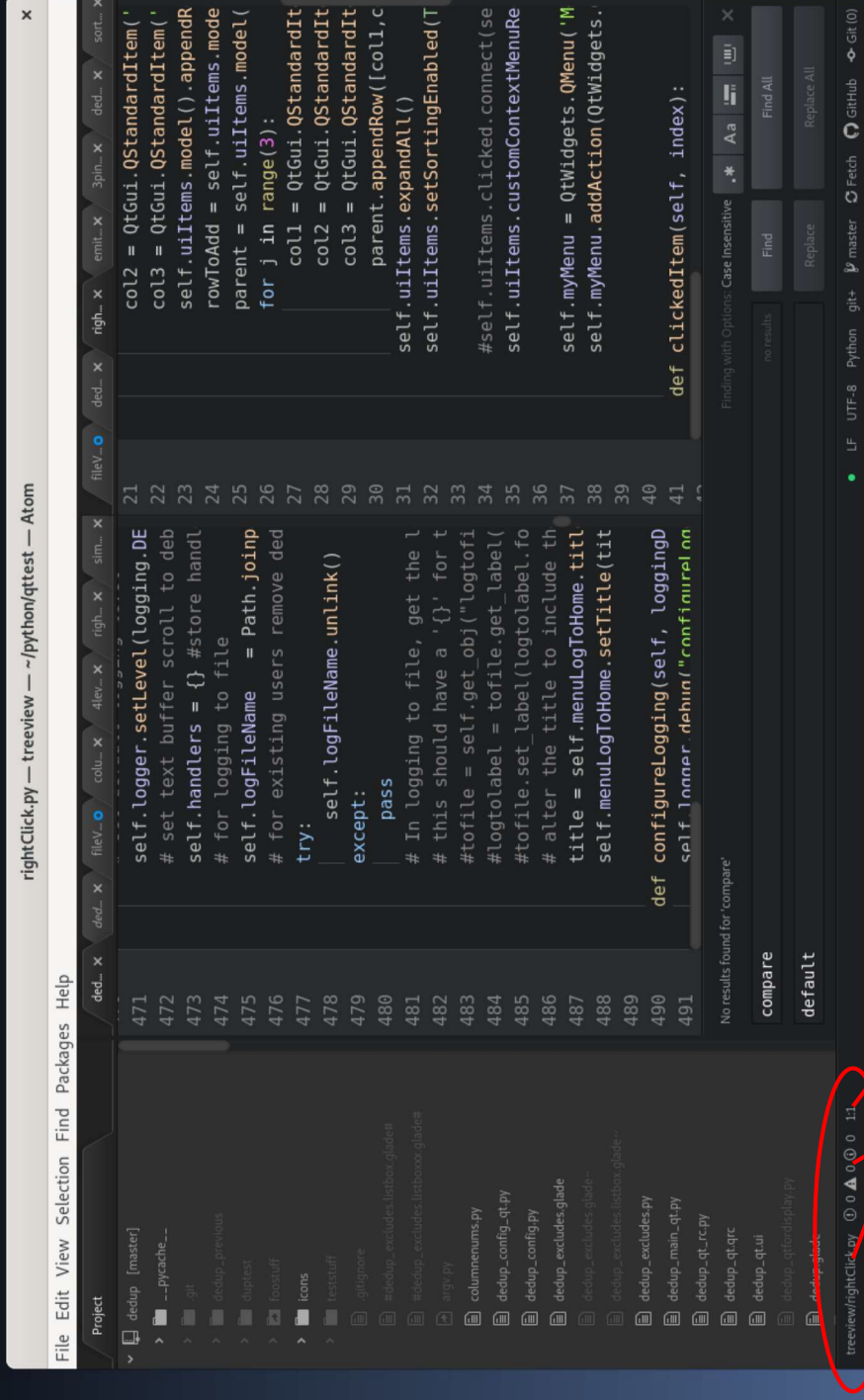


The Features I can't live without

- As I mentioned: Panes (In Atom terminology).
- A built in file manager. Atom, Geany and Pycharm have tree views.
- Different themes.
- Note that text editors can be very personal.

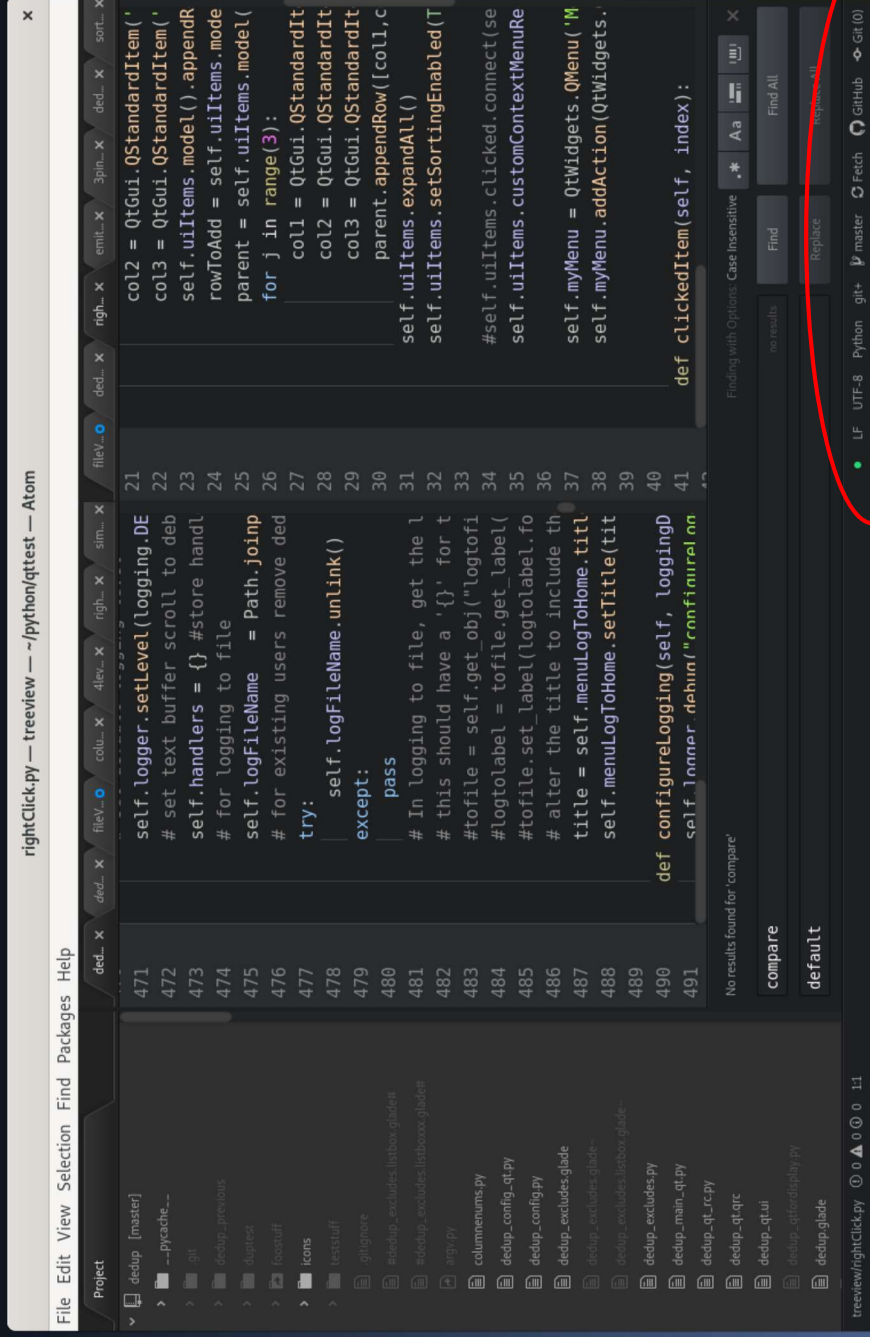


Some Screen Shots



Atom: A Hackable Editor for the 21st Century

Some Screen Shots



LF: Use LF (LineFeed) as end of line.
UTF-8 character set
Python language
Use the git+ package....

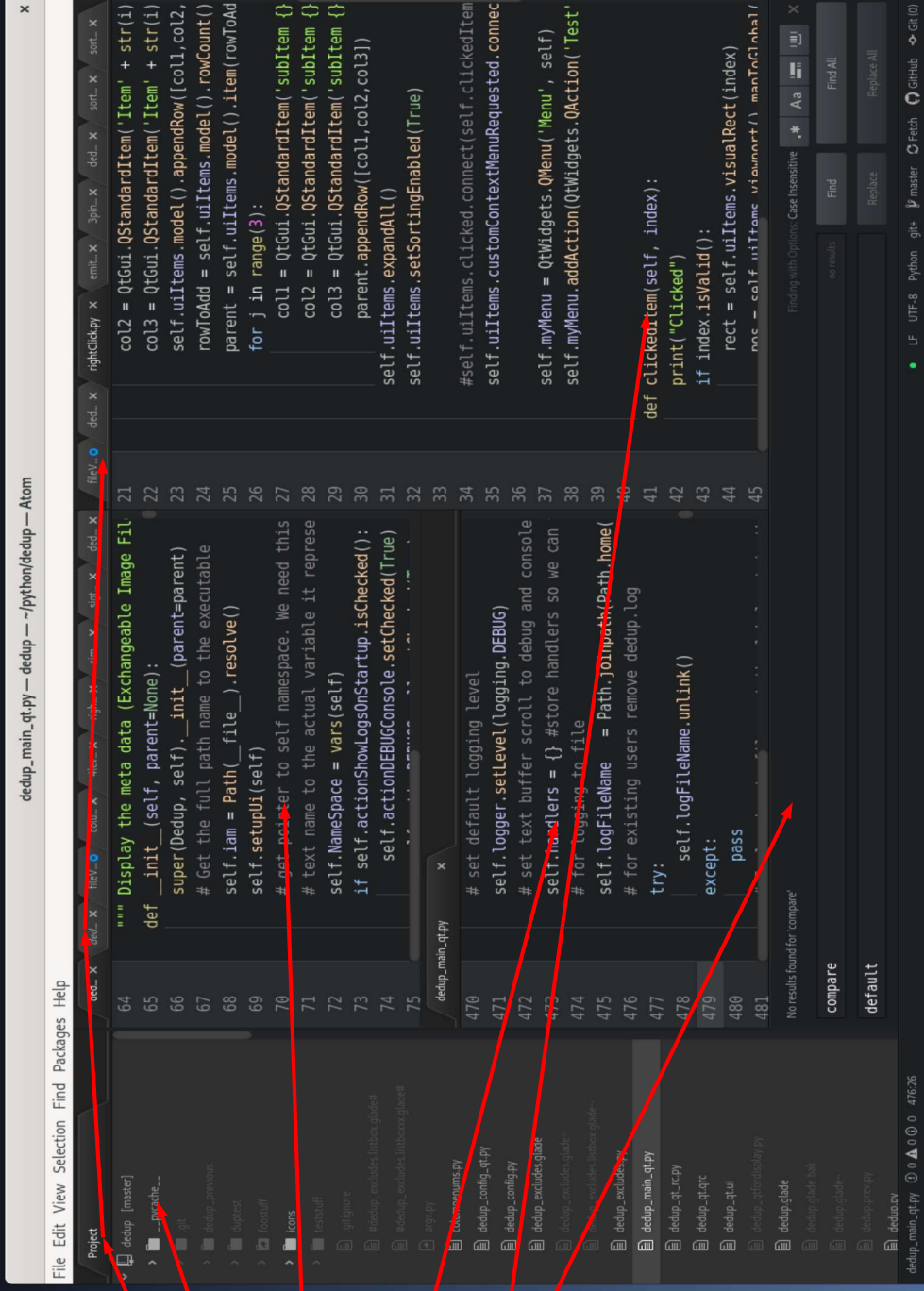


Atom: A Hackable Editor for the 21st Century

Some Screen Shots

4 Panes Visible
Each pane has its
own set of tabs.

Treeview
Top Part of code
Further on in the
same code
yet another
program
Find and Replace

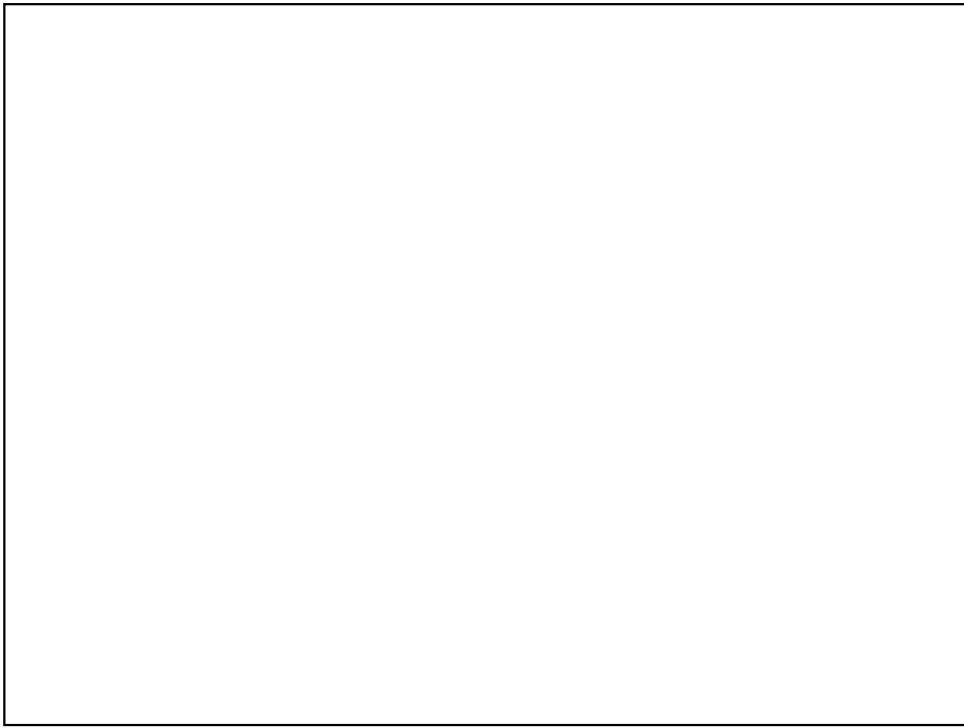


Atom: A Hackable Editor for the 21st Century

Demo

- I'm going to demo some of Atom's features. The code I will demo is exifdata, a project I created as a graphical forensic tool to view the meta data of an image.
- I will also explain each of the menus that are hard to capture in a screen shot.





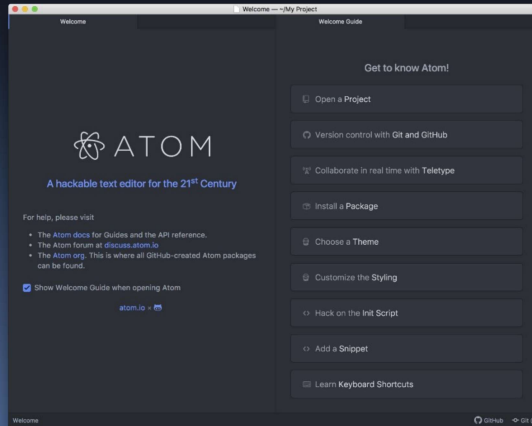
Overview

- First I will display some information from the Atom Flight Manual.
- Next I will display some screen shots with some of the screen elements that are hard to see in the demo
- Third I will demo a couple of Python scripts I have been working on to show many of the useful features.



The Atom Welcome Screen

This is what you will see when you first run Atom. You can bring up the welcome screen at any time



The Atom Flight Manual

- The Atom Flight Manual is an extensive guide in how to use atom. This will tell you all you need to know about atom:
 - <https://flight-manual.atom.io/>



How to get Atom

- Atom can be downloaded directly from:
<https://flight-manual.atom.io/getting-started/sections/installing-atom/>
- The flight manual has detailed instructions that can be copied and pasted.



Why I Chose Atom

- As I mentioned in the abstract, I am a long time EMACS user going back over 30 years. I used EMACS mainly for C, C++, and more recently TCL and Python.
- When I moved to Red Hat, I found there were many things that EMACS failed to do with my Python code, such as trim trailing spaces and syntax check.
- Many on my team used Eclipse which I hate.



Why I Chose Atom

- I looked at Geany, but IMHO it was somewhat light weight for me. One of the main features I used in EMACS was to be able to display files in different windows, both horizontally and vertically.
- Pycharm was a better IDE, but to my knowledge it did not do panes.
- Another developer suggested Atom. Atom pretty much did what I wanted.



The Features I can't live without

- As I mentioned: Panes (In Atom terminology).
- A built in file manager. Atom, Geany and Pycharm have tree views.
- Different themes.
- Note that text editors can be very personal.



Some Screen Shots

```
471 self.logger.setLevel(logging.DEBUG)
472 # set text buffer scroll to deb
473 self.handlers = [] #store handlers
474 # for logging to file
475 self.logFileName = Path.join(
476     # for existing users remove dot
477     self.logFileName.unlink()
478 )
479 except:
480     pass
481 # in logging to file, get the l
482 # this should have a ( ) for t
483 # tofile = self.get_obj("logfile")
484 #logfile = tofile.get_label()
485 #logfile.set_label(logfile_label)
486 # alter the title to include th
487 title = self.menuLogToHome.setTitle(title)
488 self.menuLogToHome.setTitle(title)
489
490 def configureLogging(self, loggingg
491     self.logger.setLevel(loggingg
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
```

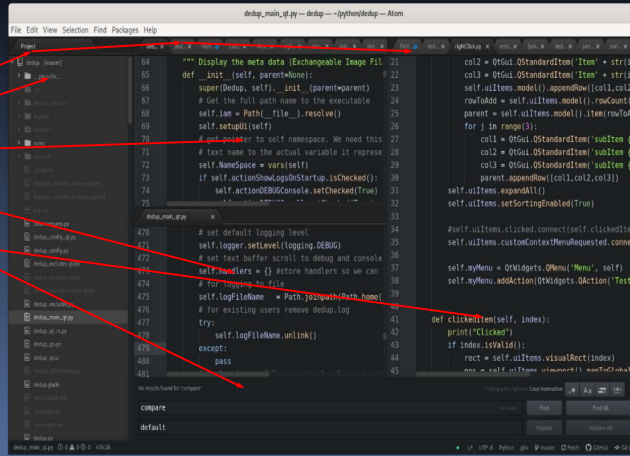
LF; Use LF (LineFeed) as end of line.
UTF-8 character set
Python language
Use the git+ package...



Some Screen Shots

4 Panes Visible
Each pane has its own set of tabs.

Treeview
Top Part of code
Further on in the same code
yet another program
Find and Replace



Atom: A Hackable Editor for the 21st Century



Demo

- I'm going to demo some of Atom's features. The code I will demo is exifdata, a project I created as a graphical forensic tool to view the meta data of an image.
- I will also explain each of the menus that are hard to capture in a screen shot.

