

efl Data Science Course

Introduction

About us: Who we are



Original Mission:

- Investigate and co-shape Digital Finance 2.0
 - Web-based selfservices of customers
- Research was performed in three different Layers:
 - Customers in E-Finance
 - E-Financial Markets and Market Infrastructures
 - IT Infrastructures: Service Systems in E-Finance



New Mission (Since 2019):

- Use expertise in Data Science to deliver cutting edge research in the fields of
 - Financial Services
 - Retail & Marketing
 - Health
 - Law
 - General, cross-sectional research

About us: The efl



Industry - Academic Partnership

Universities



Sponsors



About us: Lecturers



M.Sc. Tino Cestonaro

Research Assistant

Tino Cestonaro joined the team in April 2020. His research focuses on Market Microstructure, Financial Machine Learning,...



M.Sc. Micha Bender

Research Assistant

Micha Bender focuses on empirical financial market research. Furthermore, he is interested in the application of machine...



M.Sc. Johannes Chen

Research Assistant

Johannes Chen studied Business Informatics at the Technical University of Darmstadt. During his studies, he focused on...



**Dr. Benjamin M.
Abdel-Karim**

Alumnus

Since March 2018, Benjamin M. Abdel-Karim is a research assistant of Prof. Dr. Oliver Hinz at the Chair of Information...



Agenda

Day 1 (Python Course) (30.10.2023)

9:00 - 10:30 Uhr

Python Basics
Introduction and Primitive Data Types

10:40 - 12:10 Uhr

Data Structures
Lists, Sets, Dictionaries

13:30 - 15:00 Uhr

Control Structures
Loops (for, while), case distinction
(if, else)

15:15 - 16:45 Uhr

Functions
Structure of Functions and Application

Day 2 (Python Course) (01.11.2023)

9:00 - 10:30 Uhr

Helpful functions for data processing
Libraries: os, re, csv

10:40 - 12:10 Uhr

Data types and data structures
Libraries: numpy, pandas

13:30 - 15:00 Uhr

Data import and visualization
Libraries: csv (cont'd), matplotlib

15:15 - 16:45 Uhr

Outlook: Data Science
Exemplary implementation of a KDD process

Day 3 (Data Science) (03.11.2023)

9:00 - 10:30 Uhr

Introduction to Data Science
Terminology and basic concepts

10:40 - 12:10 Uhr

Working with Data
Preprocessing, explorative data analysis

13:30 - 15:00 Uhr

Data Analysis I
Classification

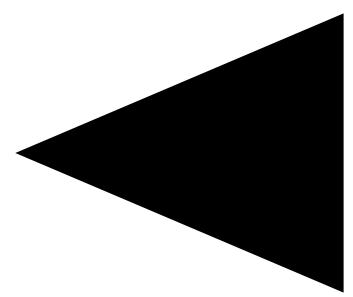
15:15 - 16:45 Uhr

Data Analysis II
Neural Networks



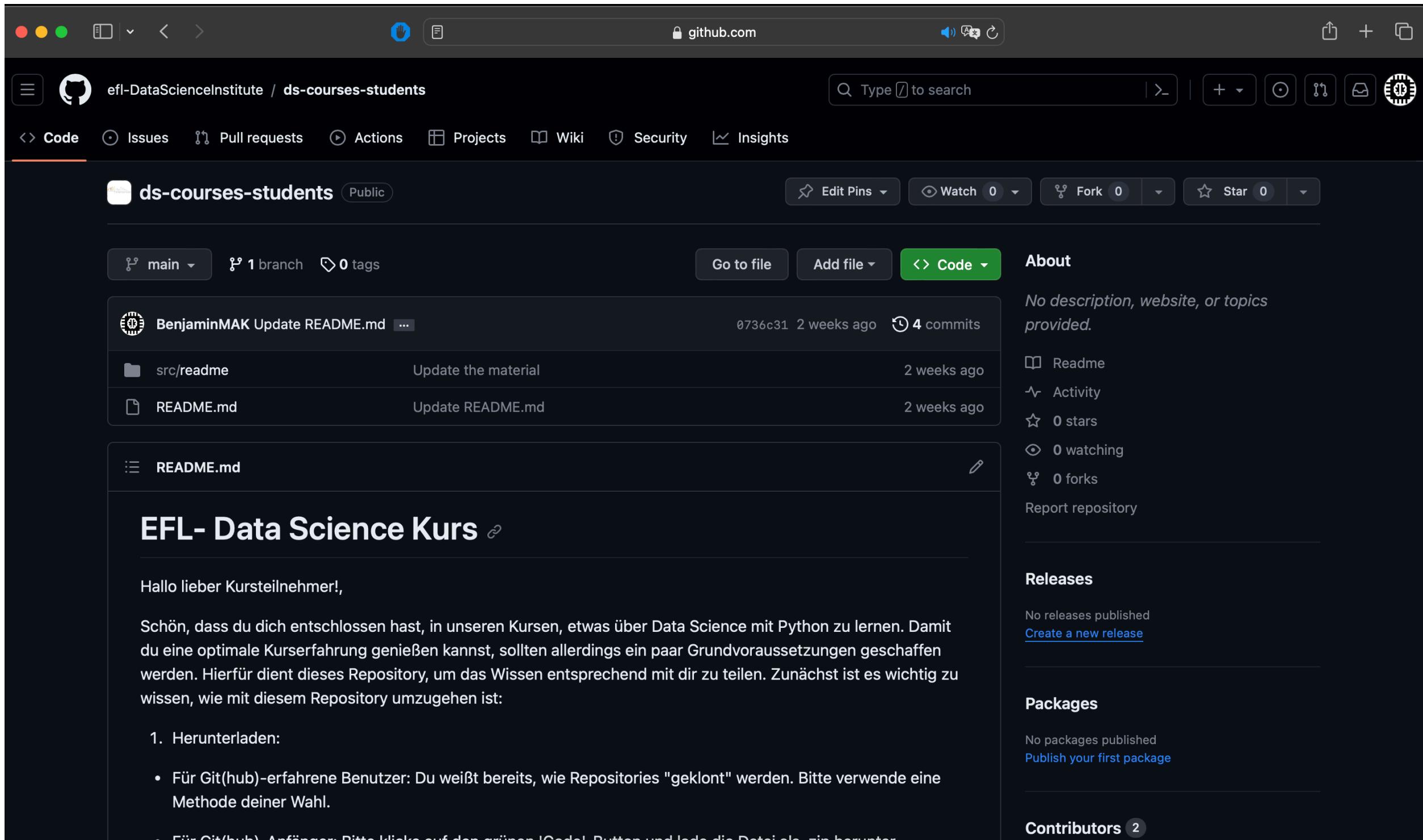
<https://www.eflab.de/teaching>

Certificates



- Completion of tasks from the last class
- Submission: Description of how the tasks were solved
- Formalities:
 - Min. 2 pages (Arial 11, 1.5 Line spacing, 3 CM Correction margin)
 - Code must be delivered separately (code folder)

Course Material?



The screenshot shows a GitHub repository page for 'efl-DataScienceInstitute / ds-courses-students'. The repository is public and contains one branch ('main') and one tag ('0.1'). There are four commits from 'BenjaminMAK' in the last two weeks. The README.md file is open, showing a welcome message for students and instructions for downloading the material. The 'About' section notes that there is no description, website, or topics provided. The 'Releases' section indicates no releases have been published. The 'Contributors' section shows two contributors.

Code

ds-courses-students Public

main · 1 branch · 0 tags

Go to file Add file <> Code

About

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Report repository

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

EFL- Data Science Kurs

Hallo lieber Kursteilnehmer!,

Schön, dass du dich entschieden hast, in unseren Kursen, etwas über Data Science mit Python zu lernen. Damit du eine optimale Kurserfahrung genießen kannst, sollten allerdings ein paar Grundvoraussetzungen geschaffen werden. Hierfür dient dieses Repository, um das Wissen entsprechend mit dir zu teilen. Zunächst ist es wichtig zu wissen, wie mit diesem Repository umzugehen ist:

1. Herunterladen:
 - Für Git(hub)-erfahrene Benutzer: Du weißt bereits, wie Repositories "geklont" werden. Bitte verwende eine Methode deiner Wahl.
 - Für Git(hub)-Anfänger: Bitte klicke auf den grünen 'Code'-Button und lade die Datei als .zip herunter.



<https://github.com/efl-DataScienceInstitute/ds-courses-students>

Why you are here

Kenntnis in Python und anderen Softwareprogrammen werden auf dem heutigen Arbeits- und Praktikumsmarkt oft vorausgesetzt. Um auch über die in den Vorlesung vermittelten Inhalte hinaus etwas über statistische Softwareprogramme zulernen, möchte ich an dem Kurs teilnehmen.

This semester I am taking "Business Informatics" with Professor Dr. Peter Gomber, which brought you to my attention.

Since the importance of Data Science as a tool for business decision making has already been discussed in class, an interest has already been sparked here. Now I hope to gain a deeper understanding through your course.

Auffrischen von Python, neue Einblicke in Software

Bei meiner aktuellen Werksstudentenstelle merke ich immer wieder, wie Entscheidungen sehr oft datengetrieben werden und welche Rolle gut aufgearbeitete Daten spielen. Daher würde ich gerne, um mich beruflich und persönlich weiterzubilden, gerne an dem Kurs teilnehmen.

Why Coding?



NACH ABSCHLUSS IN ...
**INFORMATIK, MATHEMATIK,
 WIRTSCHAFTSINFORMATIK**

TOP BERUFSTITEL*

IT Manager
92.275 €

IT-Projektmanager/in
86.695 €

Senior Developer
78.635 €

Analyst
74.469 €

Software-Architekt/in
74.152 €

TOP BRANCHE

Banken
85.067 €

Chemie- und Erdöl-
 verarbeitende Industrie
84.033 €

Konsum- und
 Gebrauchsgüter
82.586 €

Unternehmensberatung,
 Wirtschaftsprüfung und Recht
82.336 €

Finanzdienstleister
81.331 €



*Berufstitel ohne Management- & Personalverantwortung
 Bruttodurchschnittsgehalt (inkl. variabler Bezüge), Mittelwert

What is coding?

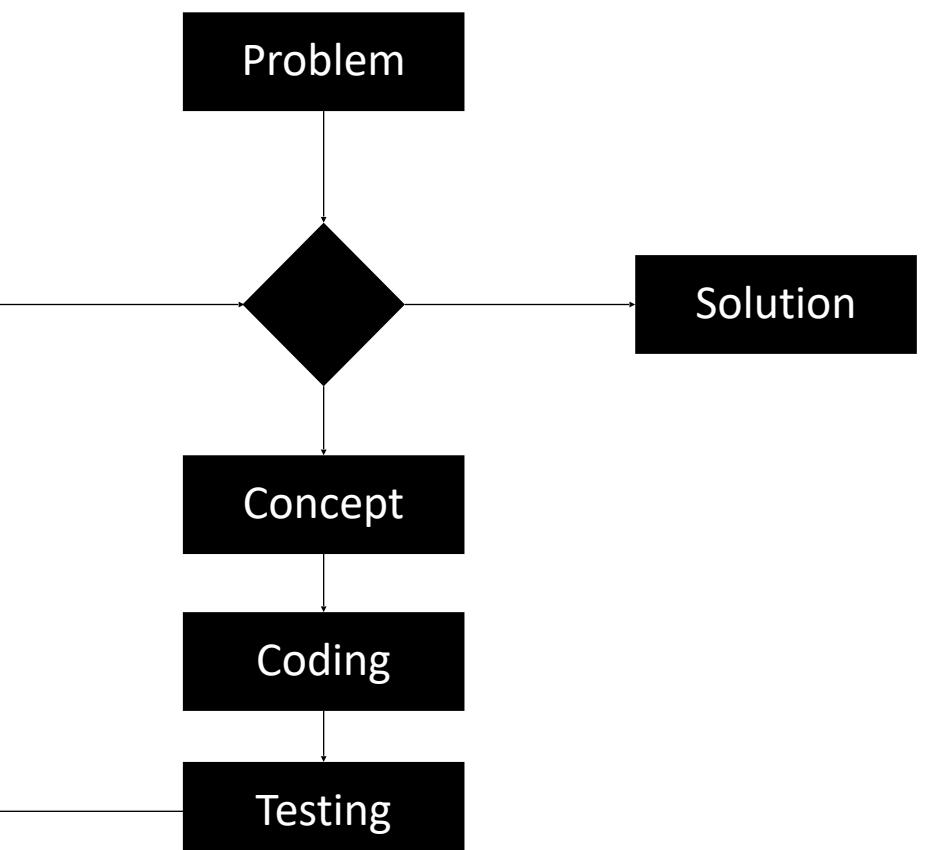


Computer programming

Article [Talk](#)

From Wikipedia, the free encyclopedia

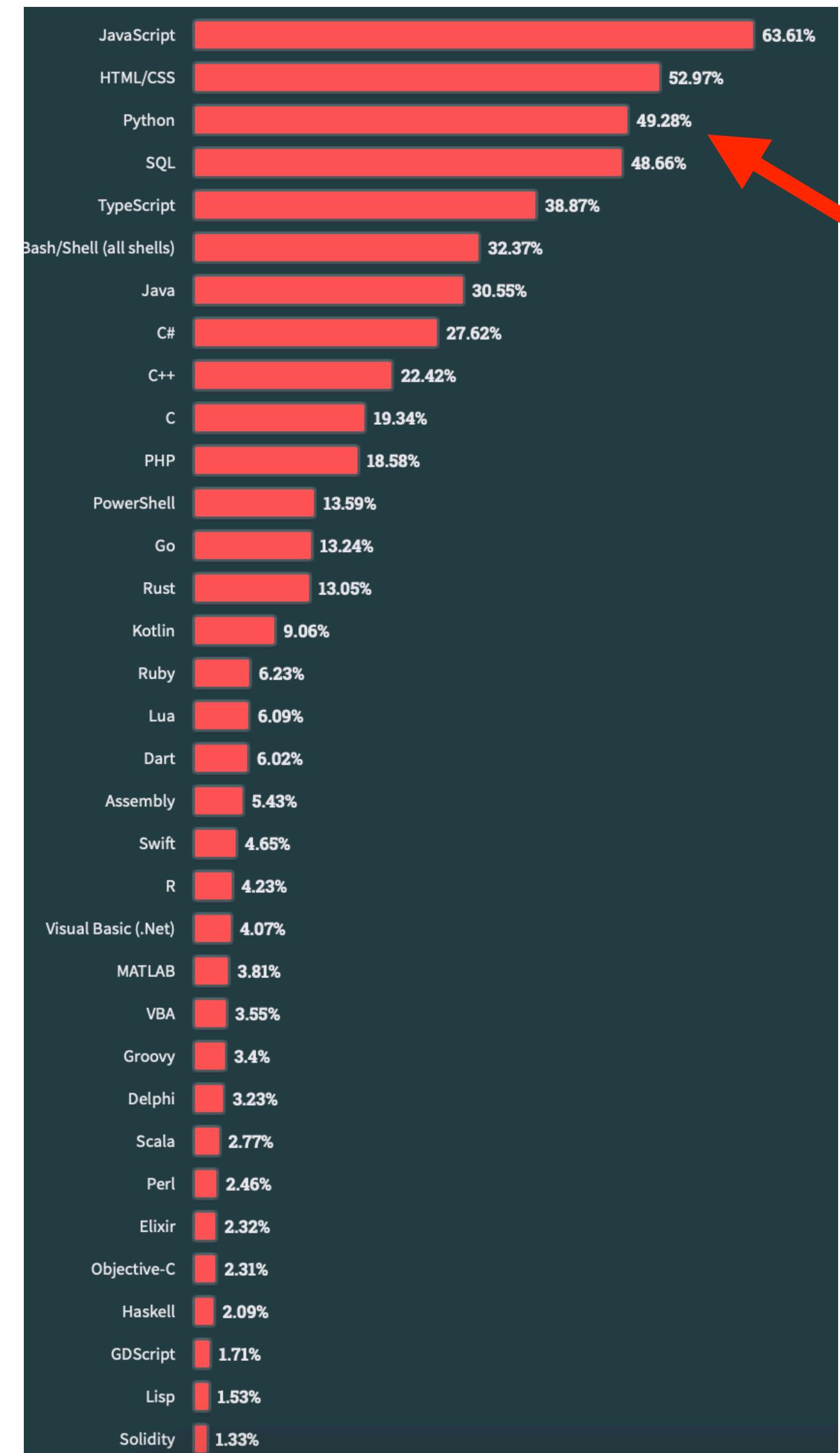
Computer programming or **coding** is the composition of sequences of instructions, called **programs**, that computers can follow to perform tasks.^{[1][2]} It involves designing and implementing **algorithms**, step-by-step specifications of procedures, by writing **code** in one or more **programming languages**. Programmers typically use **high-level programming languages** that are more easily intelligible to humans than **machine code**, which is directly executed by the **central processing unit**. Proficient programming usually requires expertise in several different subjects, including knowledge of the **application domain**, details of programming languages and generic code **libraries**, specialized algorithms, and formal **logic**.



https://en.wikipedia.org/wiki/Computer_programming

Motivation: Why Python?

- Released in 1991 by Guido van Rossum
- With the explosive growth of ‘big data’ in disciplines such as bioinformatics, neuroscience and astronomy, programming know-how is becoming ever more crucial (Perkel 2015, p. 125).



<https://survey.stackoverflow.co/2023/>

Motivation: The Elementary Basics in Python

Pros Python:

- General purpose language
- Higher programming language
- It's simple
- Fast to read
- Structuring by indenting
- No {} or ; => Faster to Code
- Data types are managed dynamically. There is no static type check like in java
- Widespread in science
- Extensive Support Libraries (important data science, math and many more)
- Integration Feature
- Productivity (Many Frameworks such as unit testing)



Ozgur, C., Colliau, T., Rogers, G., & Hughes, Z. (2017). MatLab vs. Python vs. R. Journal of data Science, 15(3), 355-371.
Abdel-Karim, B. M. (2022). Data Science. In Data Science: Best Practices mit Python (pp. 57-62). Wiesbaden: Springer Fachmedien Wiesbaden.

Cons of Python

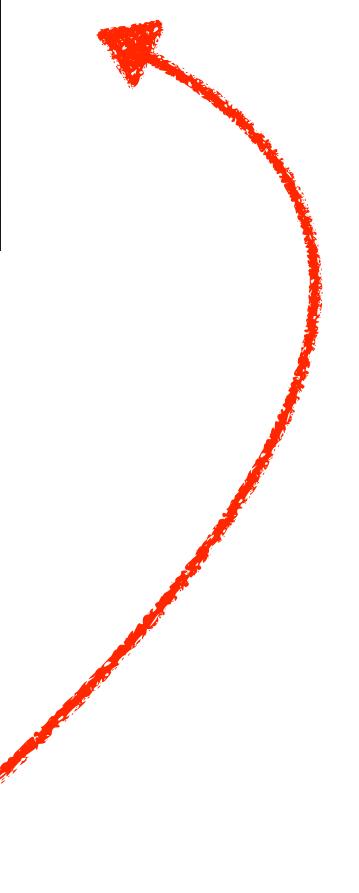
Some Points:

- Python is an interpreted language, this is slow as compared to C/C++
- See Interpreted vs. Compiled programming languages
- Python can have runtime errors (dynamical typing feature)
- Consumes a lot of memory space
- Not easy to test



Small Satellite*

*) <https://www.nbcnews.com/id/wbna43313086>



Not in Python ...

Prechelt, L. (2000). An empirical comparison of c, c++, java, perl, python, rex and tcl. *IEEE Computer*, 33(10), 23-29.

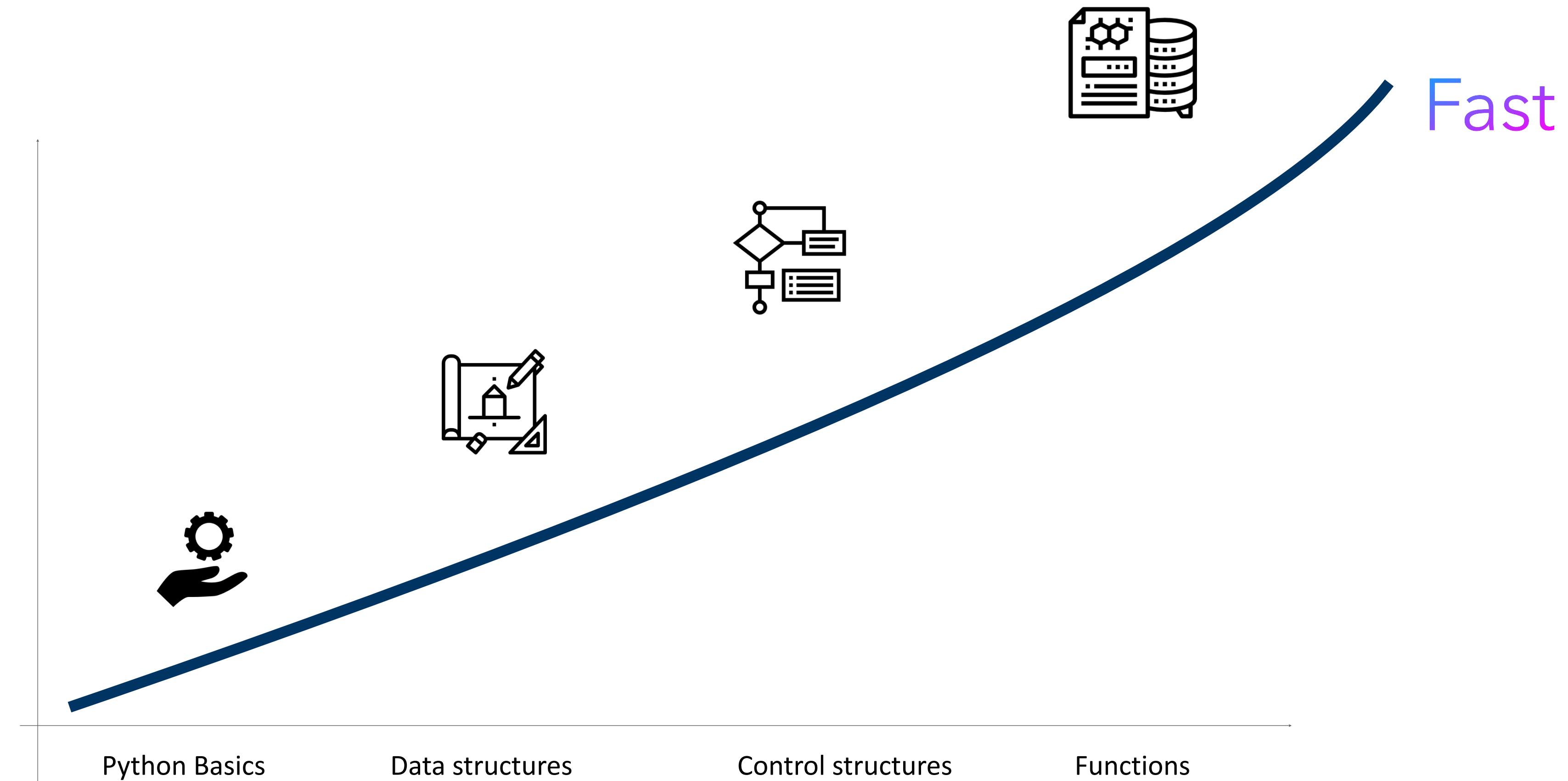
Newhall, T. K. (1999). Performance Measurement of Interpreted, Just-in-Time compiled, and Dynamically Compiled Executions. The University of Wisconsin-Madison.

Agreement

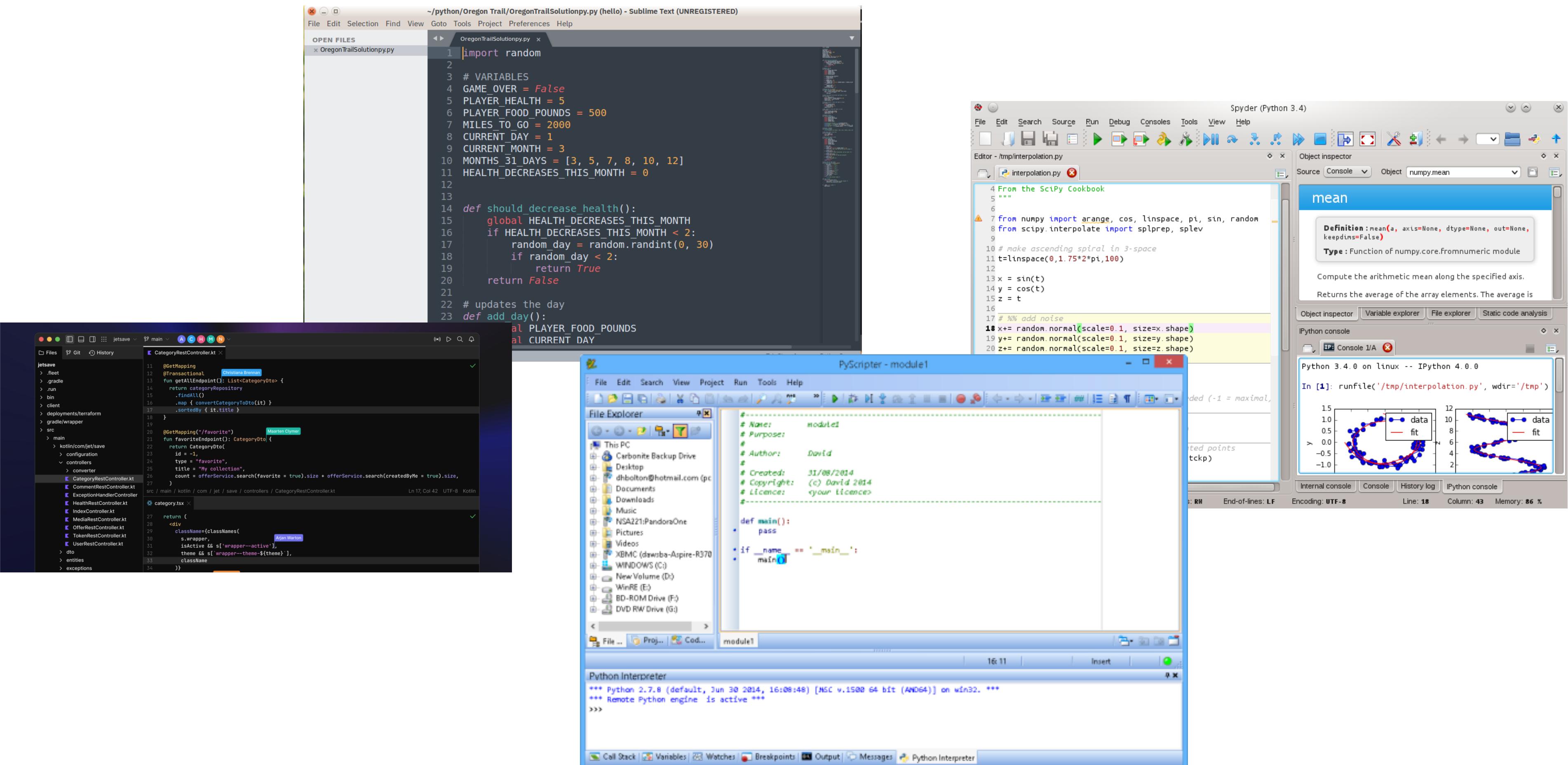


- Introduction to programming => All Questions are allowed!!!
- For beginners
- The module is interactive => Use your computer
- **We develop the solutions together!**
- Please be on time
- There are no dumb questions
- Nobody knows everything
- Copying solutions is plagiarism

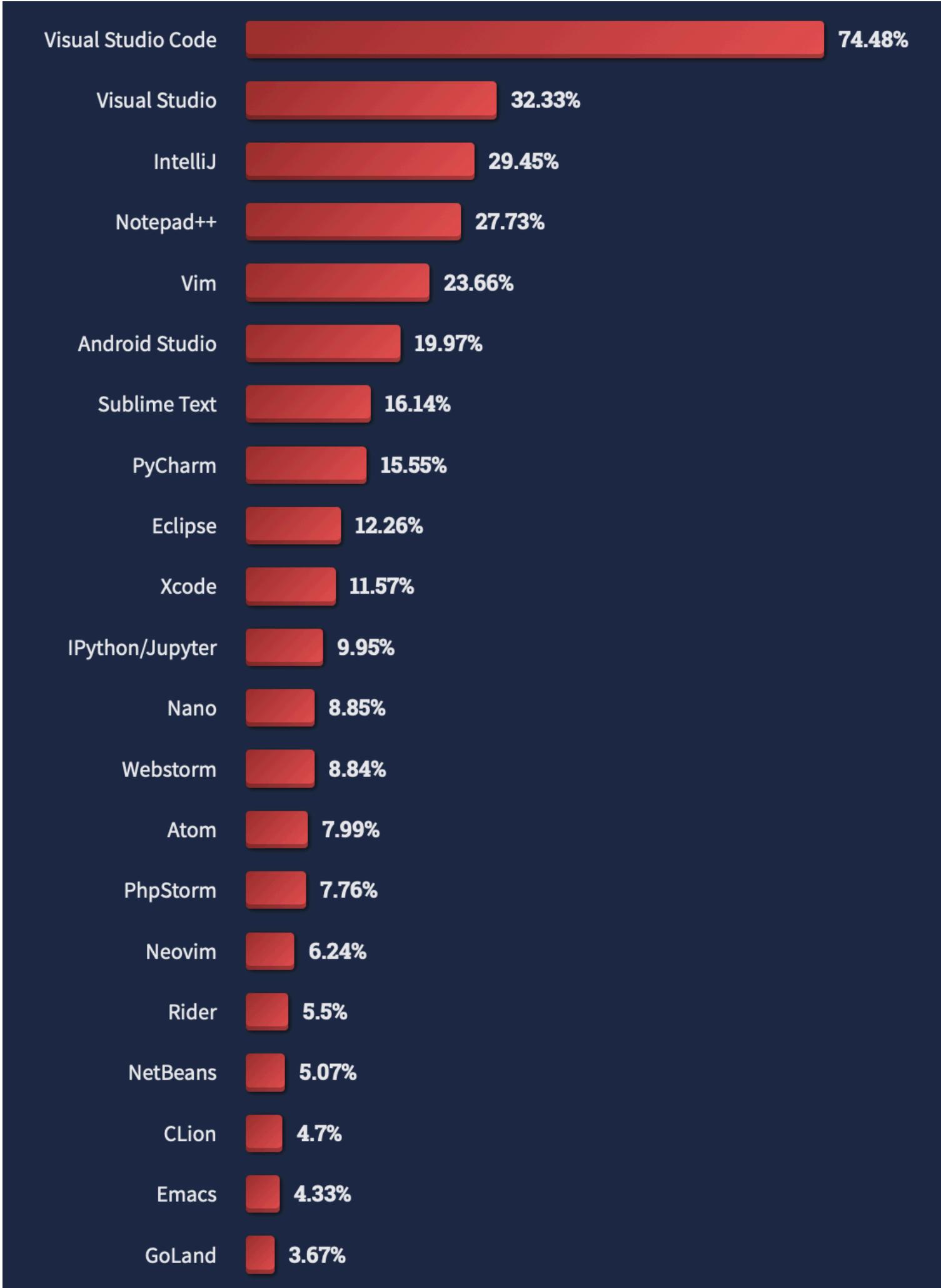
Note: Learning Curve for Today



Integrated Development Environment IDE a Deeper look? The Problem of Choice

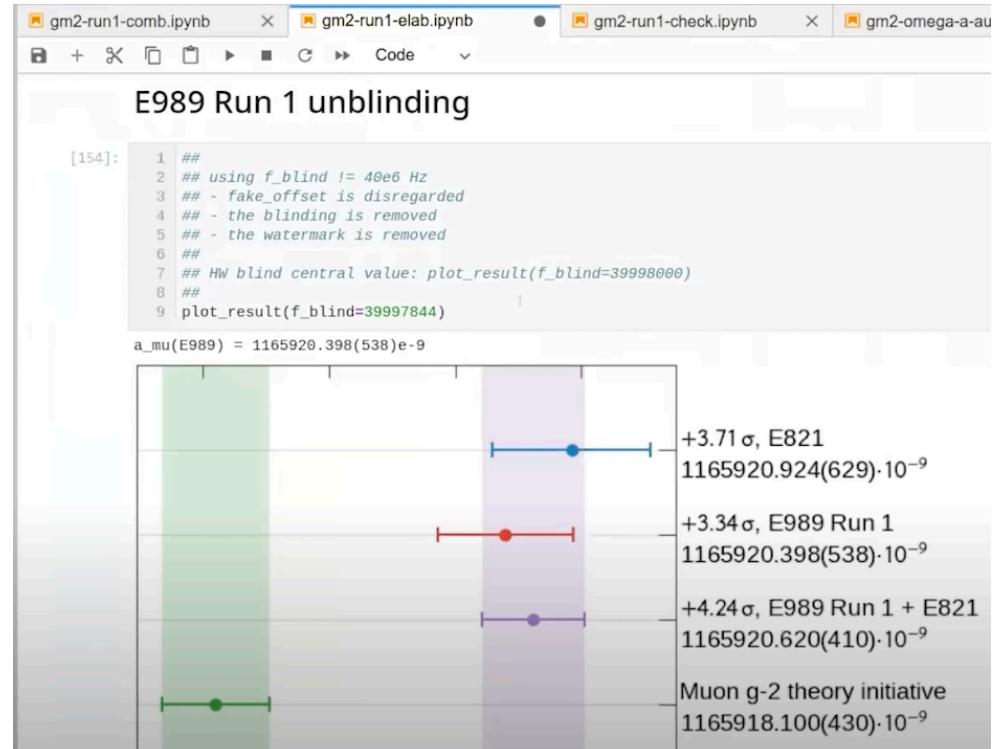


The Problem of Choice



<https://survey.stackoverflow.co/2022/#most-popular-technologies-new-collab-tools-prof>

A Question of Style



E989 Run 1 unblinding

```
[154]: 1 ## using f_blind != 4@e6
2 ## - fake_offset is disregarded
3 ## - the blinding is removed
4 ## - the watermark is removed
5 ## - the HWW blind central value: plot_result(f_blind=39998000)
6 ##
7 ## HWW blind central value: plot_result(f_blind=39997844)
8 #
9 plot_result(f_blind=39997844)
```

a_mu(E989) = 1165920.398(538)e-9

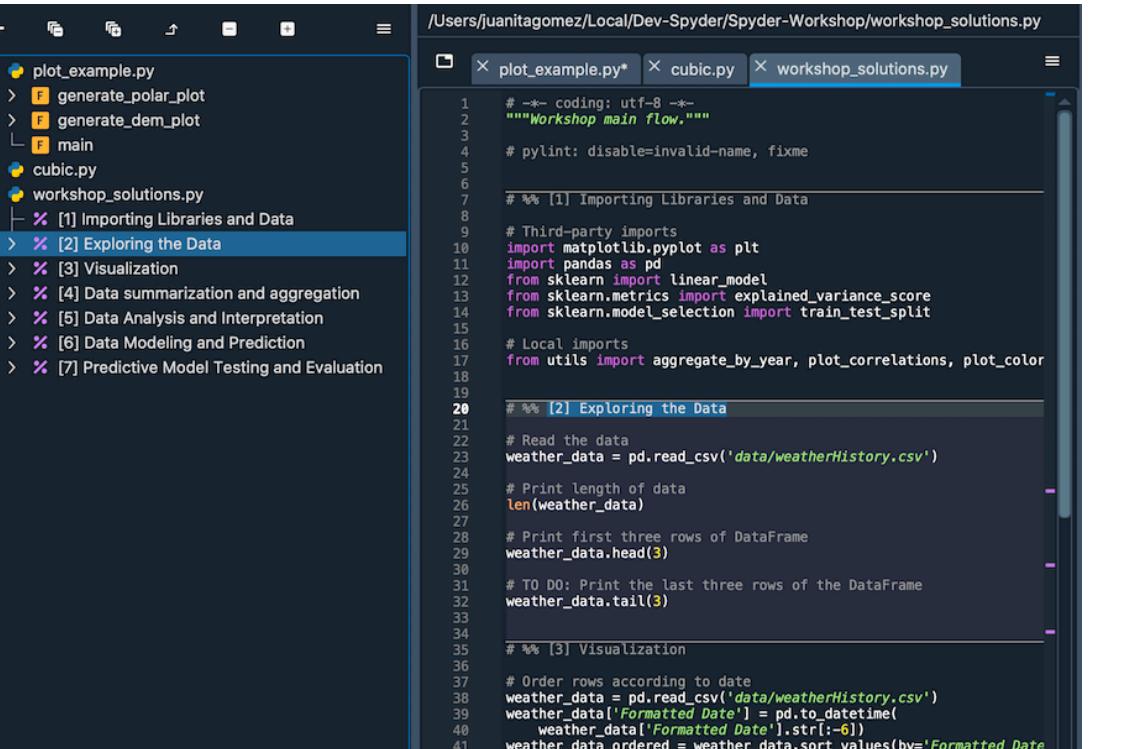
+3.71 σ, E821
1165920.924(629).10⁻⁹

+3.34 σ, E989 Run 1
1165920.398(538).10⁻⁹

+4.24 σ, E989 Run 1 + E821
1165920.620(410).10⁻⁹

Muon g-2 theory initiative
1165918.100(430).10⁻⁹

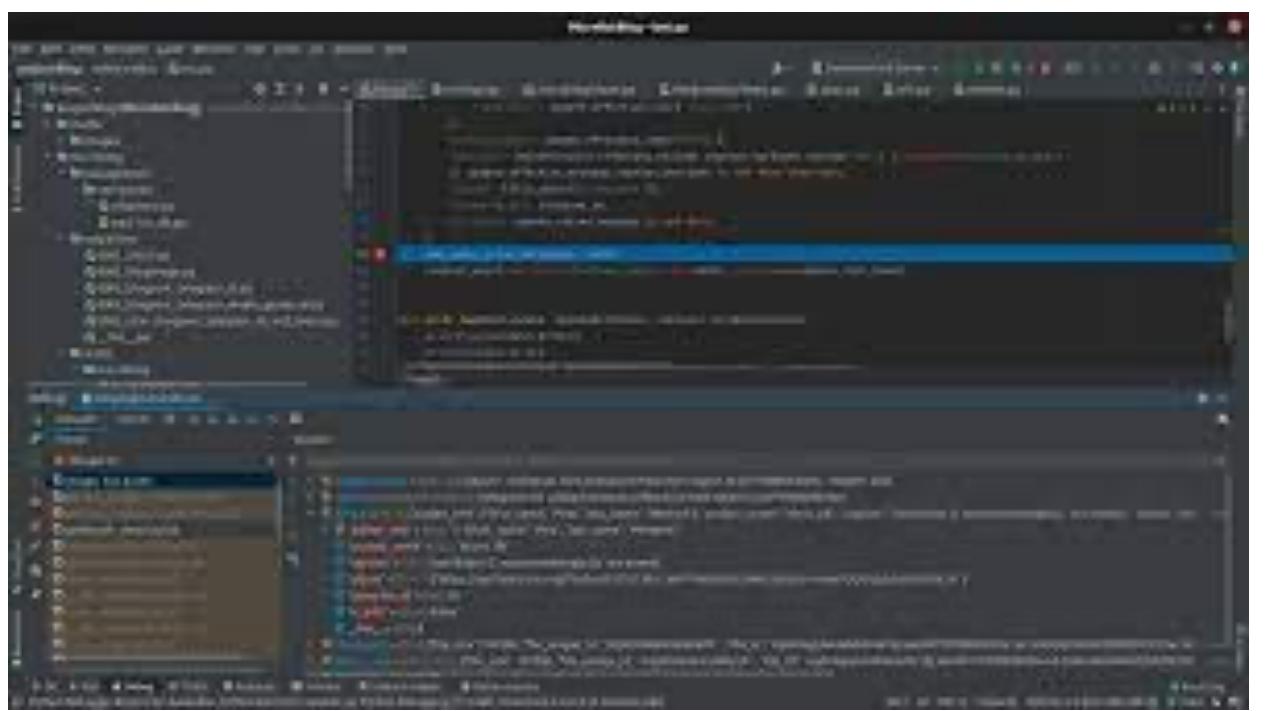
Notebooks (Research)*



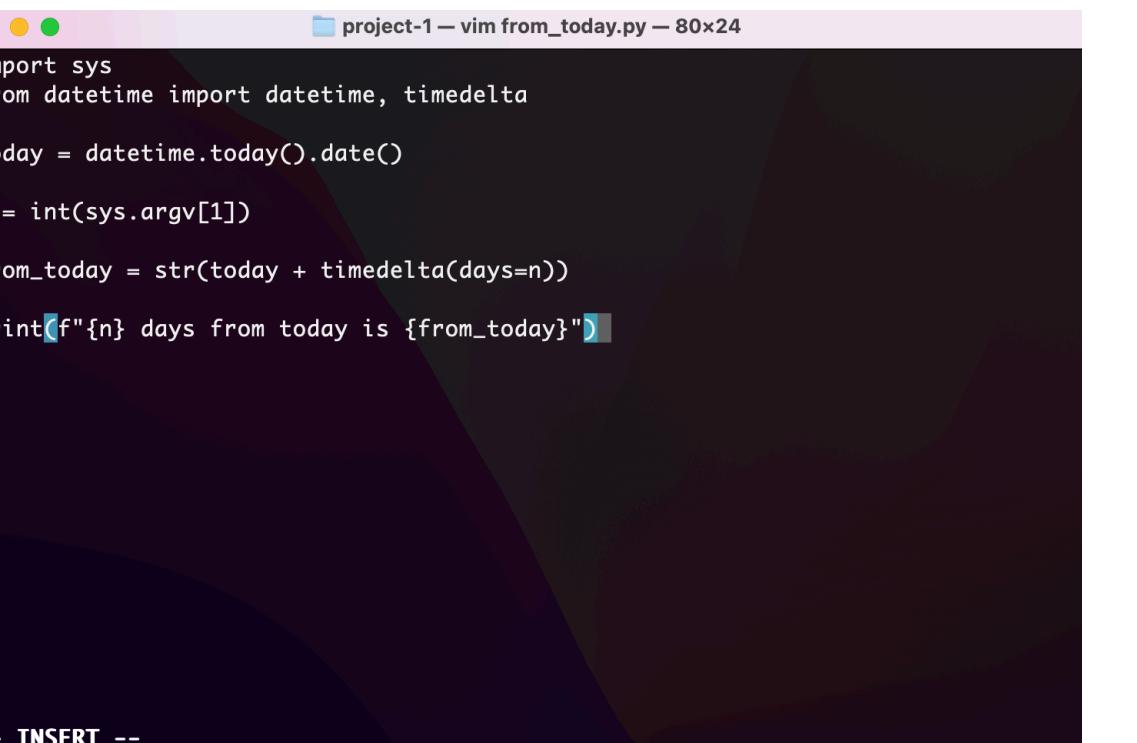
/Users/juanitogomez/Local/Dev-Spyder/Spyder-Workshop/workshop_solutions.py

```
1 #-- coding: utf-8 --#
2 """Workshop main flow."""
3
4 # pylint: disable=invalid-name, fixme
5
6 # %% [1] Importing Libraries and Data
7
8 # Third-party imports
9 import matplotlib.pyplot as plt
10 import pandas as pd
11 from sklearn import linear_model
12 from sklearn import metrics
13 from sklearn import model_selection
14 from sklearn.model_selection import train_test_split
15
16 # Local imports
17 from utils import aggregate_by_year, plot_correlations, plot_color
18
19 # %% [2] Exploring the Data
20
21 # Read the data
22 weather_data = pd.read_csv('data/weatherHistory.csv')
23
24 # Print length of data
25 len(weather_data)
26
27 # Print first three rows of DataFrame
28 weather_data.head(3)
29
30 # TO DO: Print the last three rows of the DataFrame
31 weather_data.tail(3)
32
33
34 # %% [3] Visualization
35
36 # Order rows according to date
37 weather_data = pd.read_csv('data/weatherHistory.csv')
38 weather_data['Formatted Date'] = pd.to_datetime(
39     weather_data['Formatted Date'].str[-6:])
40 weather_data_ordered = weather_data.sort_values(by='Formatted Date')
```

Hybrid #%% (standard cell separator)



Classical IDE Coding



```
project-1 — vim from_today.py — 80x24
import sys
from datetime import datetime, timedelta

today = datetime.today().date()

n = int(sys.argv[1])

from_today = str(today + timedelta(days=n))

print(f"{n} days from today is {from_today}")
```

-- INSERT --

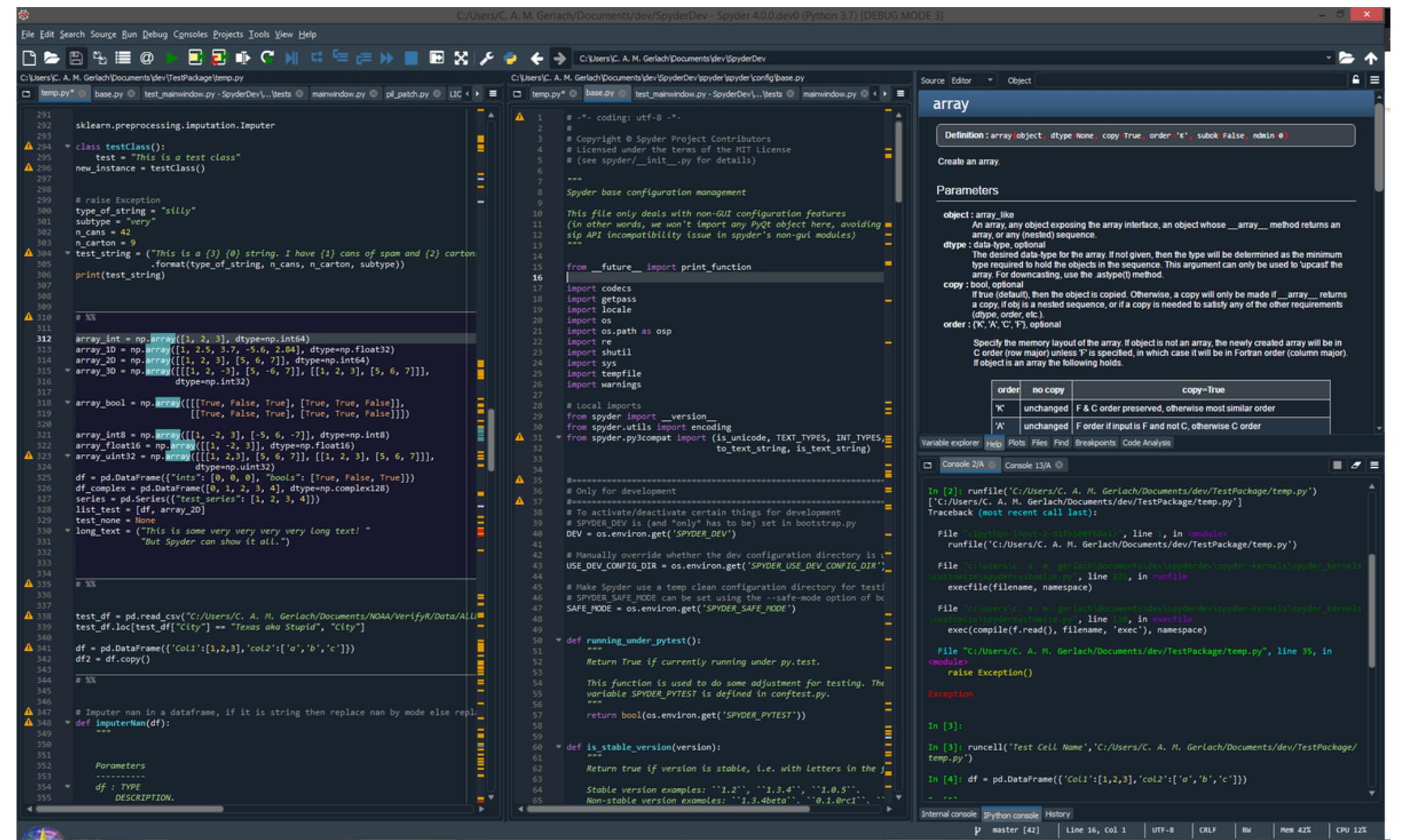
Coding via Terminal (Ultra Mode)

*) Famous experiment: https://www.youtube.com/watch?v=hL2w_FkTae8

What We Use for This Course



SPYDER



The screenshot shows the Spyder 4.0.0.dev0 Python 3.7 [DEBUG MODE 3] interface. The code editor displays a script named `temp.py` containing Python code related to array management. The documentation viewer on the right provides details about the `array` class, including its definition, parameters, and memory layout options. The console window at the bottom shows the output of running the script, including tracebacks and execution results.

```

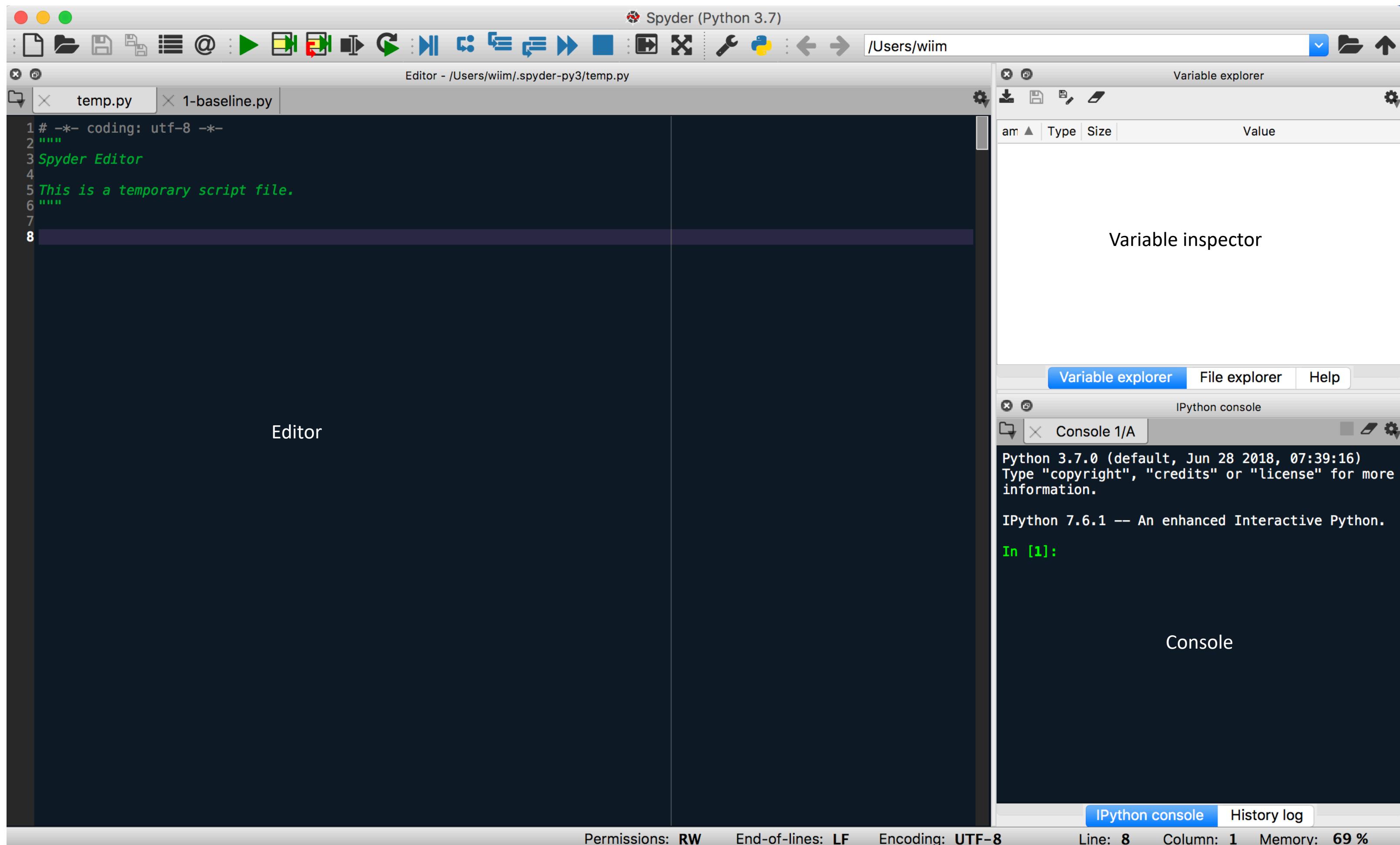
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:/Users/C. A. M. Gerlach/Documents/dev/spyderDev - Spyder 4.0.0.dev0 (Python 3.7) [DEBUG MODE 3]
C:/Users/C. A. M. Gerlach/Documents/dev/spyderDev
temp.py  base.py  test_mainwindow.py - SpyderDev..._Tests mainwindow.py  pil_patch.py  LIC...
temp.py  base.py  test_mainwindow.py - SpyderDev..._Tests mainwindow.py

1 # -*- coding: utf-8 -*-
2
3 # Copyright © Spyder Project Contributors
4 # Licensed under the terms of the MIT License
5 # (see spyder/_init_.py for details)
6
7 """
8 Spyder base configuration management
9
10 This file only deals with non-GUI configuration features
11 (in other words, we won't import any PyQt object here, avoiding
12 SIP API incompatibility issue in spyder's non-gui modules)
13
14
15 from __future__ import print_function
16
17 import codecs
18 import imp
19 import locale
20 import os
21 import os.path as osp
22 import platform
23 import shutil
24 import sys
25 import tempfile
26 import warnings
27
28 # Local imports
29 from spyder import _version
30 from spyder.utils import encoding
31 from spyder.utils import importlib
32
33 #-----#
34 # Only for development
35 # To activate/deactivate certain things for development
36 # SPYDER_DEV is (and "only" has to be) set in bootstrap.py
37 # DEV = os.environ.get('SPYDER_DEV')
38
39 # Manually override whether the dev configuration directory is used
40 USE_DEV_CONFIG_DIR = os.environ.get('SPYDER_USE_DEV_CONFIG_DIR')
41
42 # Make Spyder use a temp clean configuration directory for tests
43 # if SPYDER_DEV is set. This is a temporary mode option of bc
44 SAFE_MODE = os.environ.get('SPYDER_SAFE_MODE')
45
46 #-----#
47
48 def running_under_pytest():
49     """
50     Return True if currently running under py.test.
51
52     This function is used to do some adjustment for testing. The
53     variable SPYDER_PYTEST is defined in conftest.py.
54
55     return bool(os.environ.get('SPYDER_PYTEST'))
56
57
58 def is_stable_version(version):
59     """
60     Return true if version is stable, i.e. with letters in the form
61     '1.2.3'. Non-stable version examples: '1.2', '1.3.4', '1.0.5'.
62     Non-stable version examples: '1.3.4beta', '0.1.0rc1', ...
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
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
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
245
246
247
247
248
248
249
249
250
251
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
99
```

Practice and Questions

- Take your computer and let's get started!
- Create a Folder 'Day 1' on our Desktop
- Launch Spyder and select the folder

In a Nutshell: Integrated Development Environment

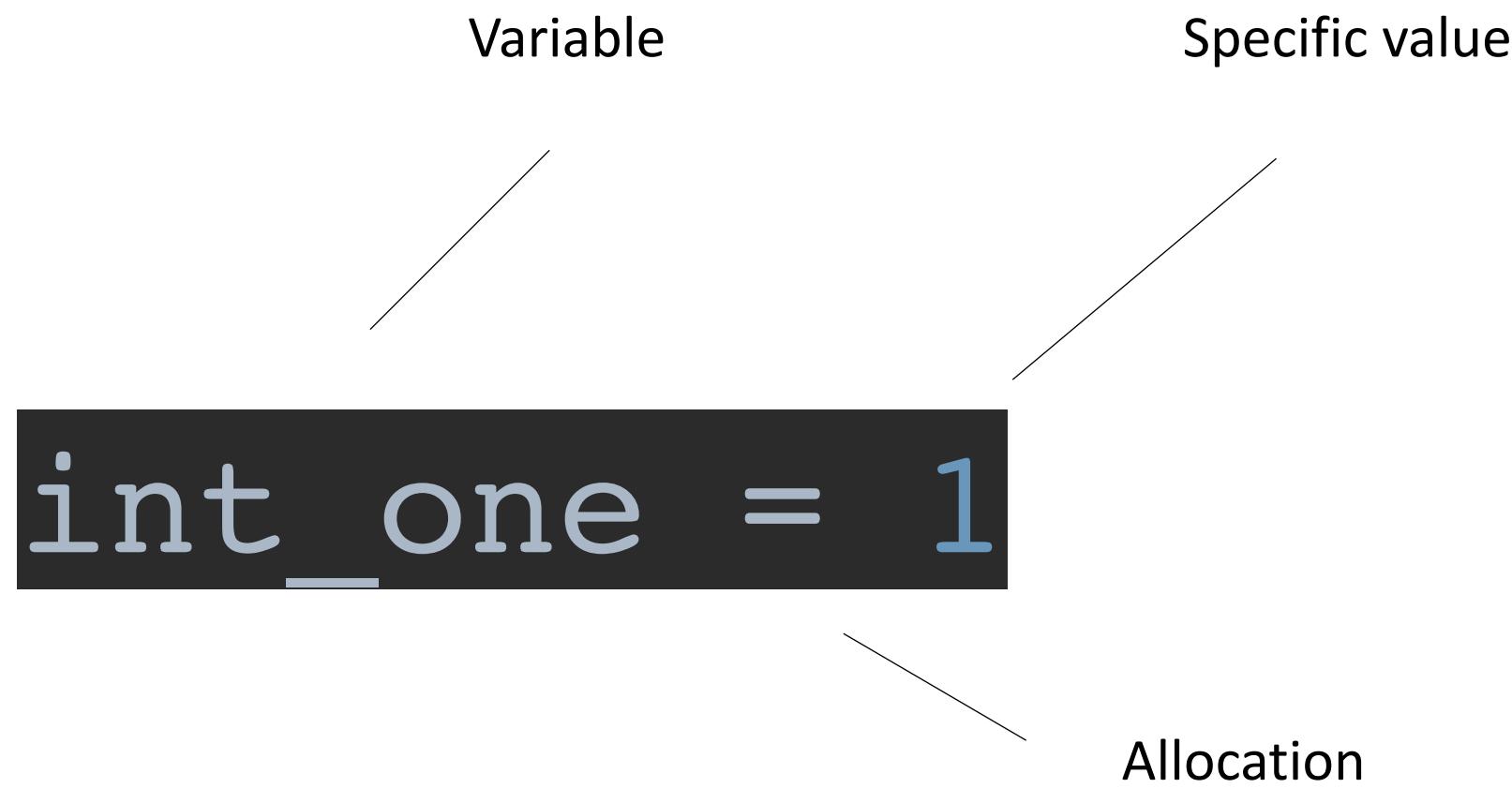


In a Nutshell: Hello World

```
print('Hello World')
```

Abstraction (Computer Science)

The first step - The concept of variable



We assign the value 1 to the variable 'one'.

Now we can continue working with the variable 'one'.

Advantage: We are independent of concrete value

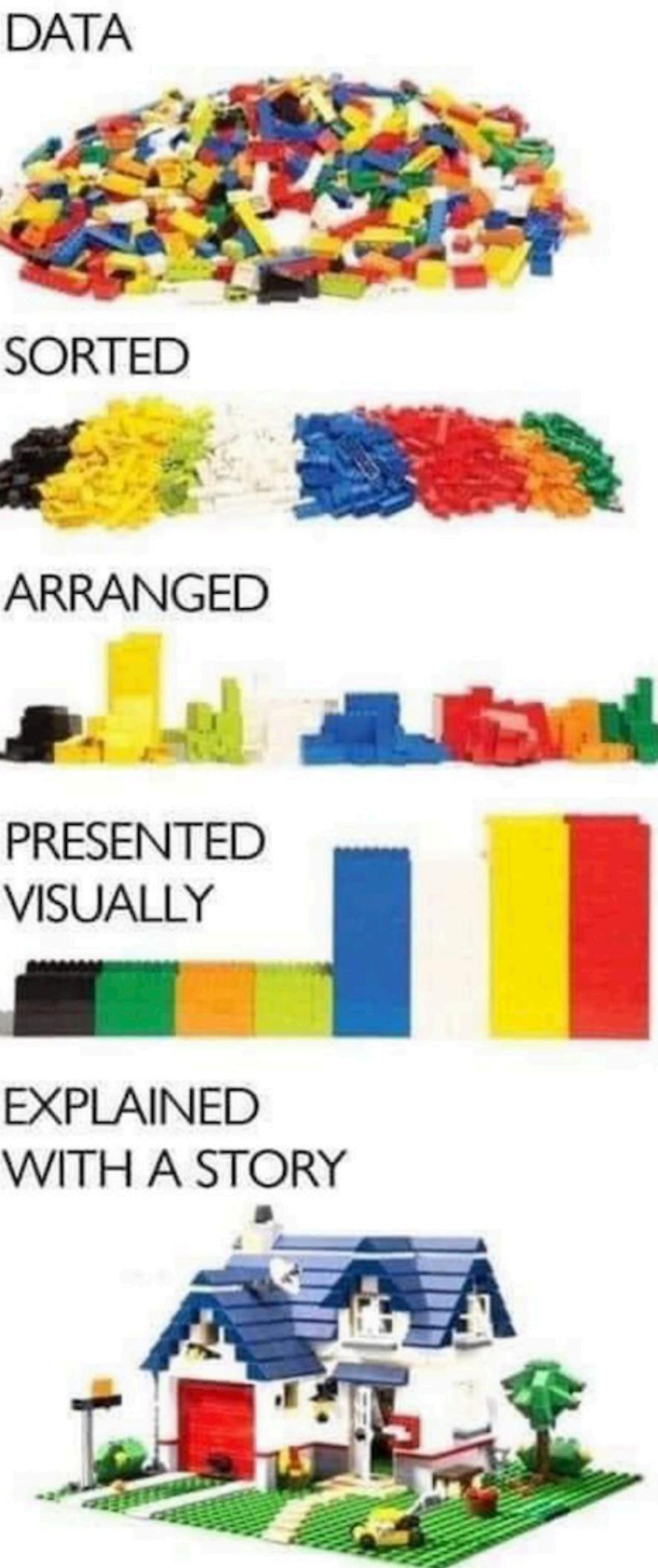
The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context.

– John V. Guttag



Different Building Blocks

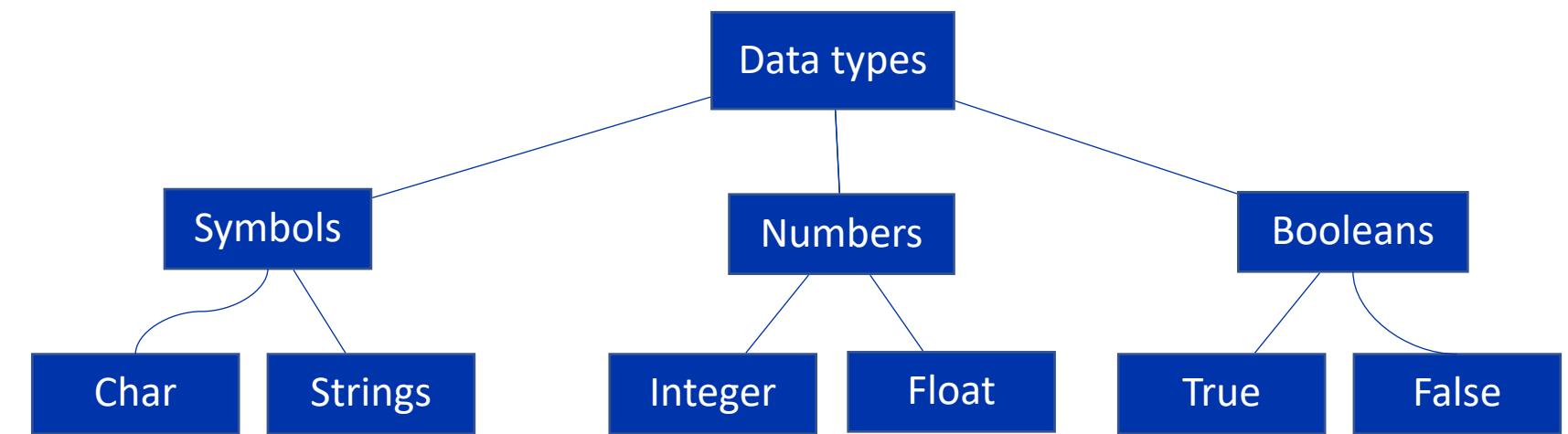
- Integer: int (signed integers) – They are often just called integers or ints, are positive or negative whole numbers with no decimal point.
- Float: Float (floating point real values, double) – Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10.
- Strings are a sequence of chars. We can create them simply by enclosing characters in quotes. “Hello World” is a String! Therefore strings in Python are bytes representing Unicode characters. In Detail: Python does not have a character data type, a single character is simply a string with a length of 1.
- Boolean: We can perform logical operations with True and False in combination with AND and OR.



Primitive Types

- Integer: int (signed integers) – They are often just called integers or ints, are positive or negative whole numbers with no decimal point.
- Float: Float (floating point real values, double) – Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10.
- Strings are a sequence of chars. We can create them simply by enclosing characters in quotes. “Hello World” is a String! Therefore strings in Python are bytes representing Unicode characters. In Detail: Python does not have a character data type, a single character is simply a string with a length of 1.
- Boolean: We can perform logical operations with True and False in combination with AND and OR.

Primitive Data Types: A first overview



* Simplified Illustration

Python Documentation: <https://docs.python.org/3/library/stdtypes.html>

Other classification: Chun, W. (2001). Core python programming (Vol. 1). Prentice Hall Professional.

Documentation

Source Code Documentation

@author: name

@since: first implementation date

@version: date of last update

@source: if you using links etc.

@code: special code note

@param: if special parameter is used or you have to describe.

```
# I am a comment
@author: My Name
@since: 2022-10-03
@update: 2023-10-26
@version: v.0.0.2
print('Hello World')
```

More Information: PEP 8 – Style Guide for Python Code: <https://peps.python.org/pep-0008/>

Naming convention

- Names of attributes, variables, methods start with a small letter
 - may use letters without ß or similar
 - which points to the data type like i, s or l
- This is standard in professional software development.
- Camel Case: Compound words are written in programming language. Every new word is capitalized.

(More PeP8 Style)

- name = Is the name of...
- bscript_main = Simple code file that does something
- CName = Class (later more)
- float_name = Variable that saves a floating point (double)
- int_name = Variable that saves an integer value
- str_name = Variable that saves a string value
- b_name = Boolean for true or false values
- list_name = Object from type list
- method_name = Self-written function

(Camel Case Style)

- Name = Is the name of...
- bScriptName = Simple code file that does something
- CName = Class (later more)
- dName = Variable that saves a floating point (double)
- iName = Variable that saves an integer value
- sName = Variable that saves a string value
- bName = Boolean for true or false values
- LName = Object from type list
- fName = Self-written function

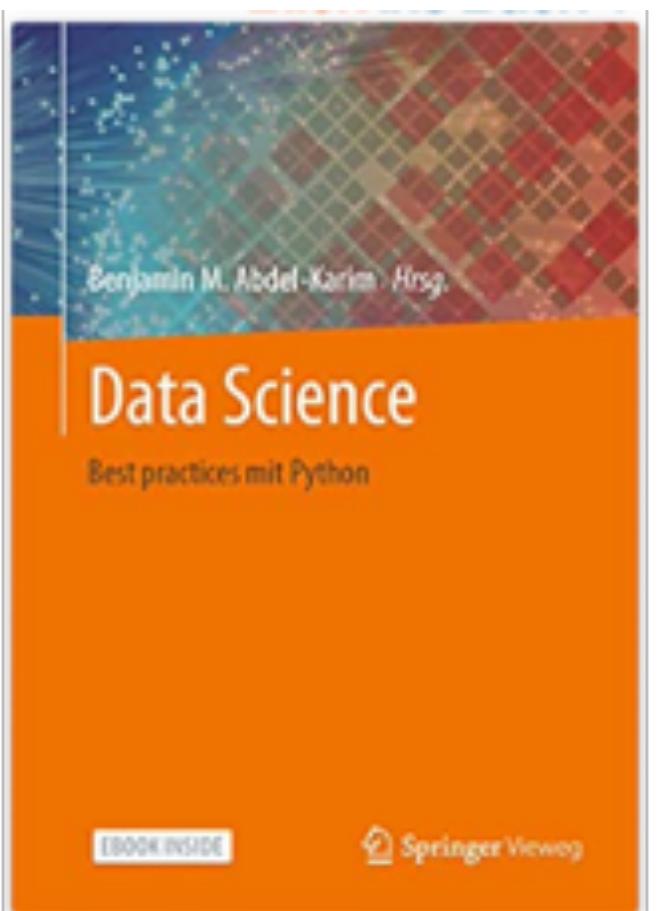
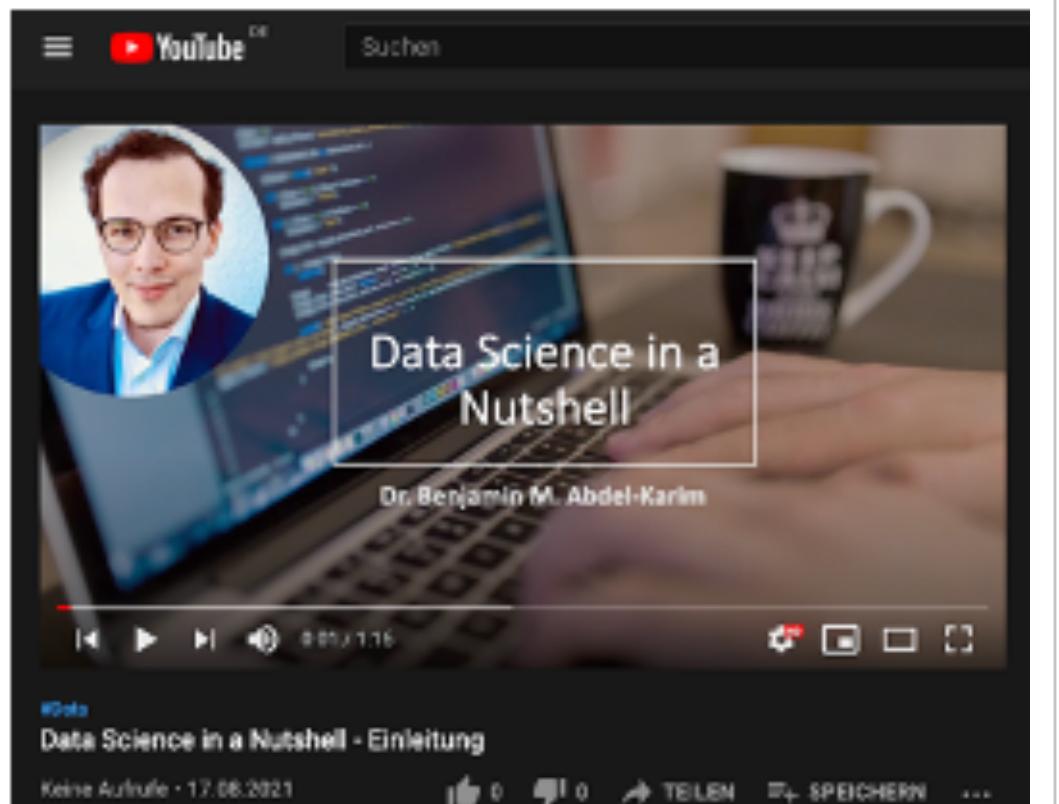
Code is some kind of Art. Therefore...



Jackson Pollock Art

These are all approaches. Find your own style!

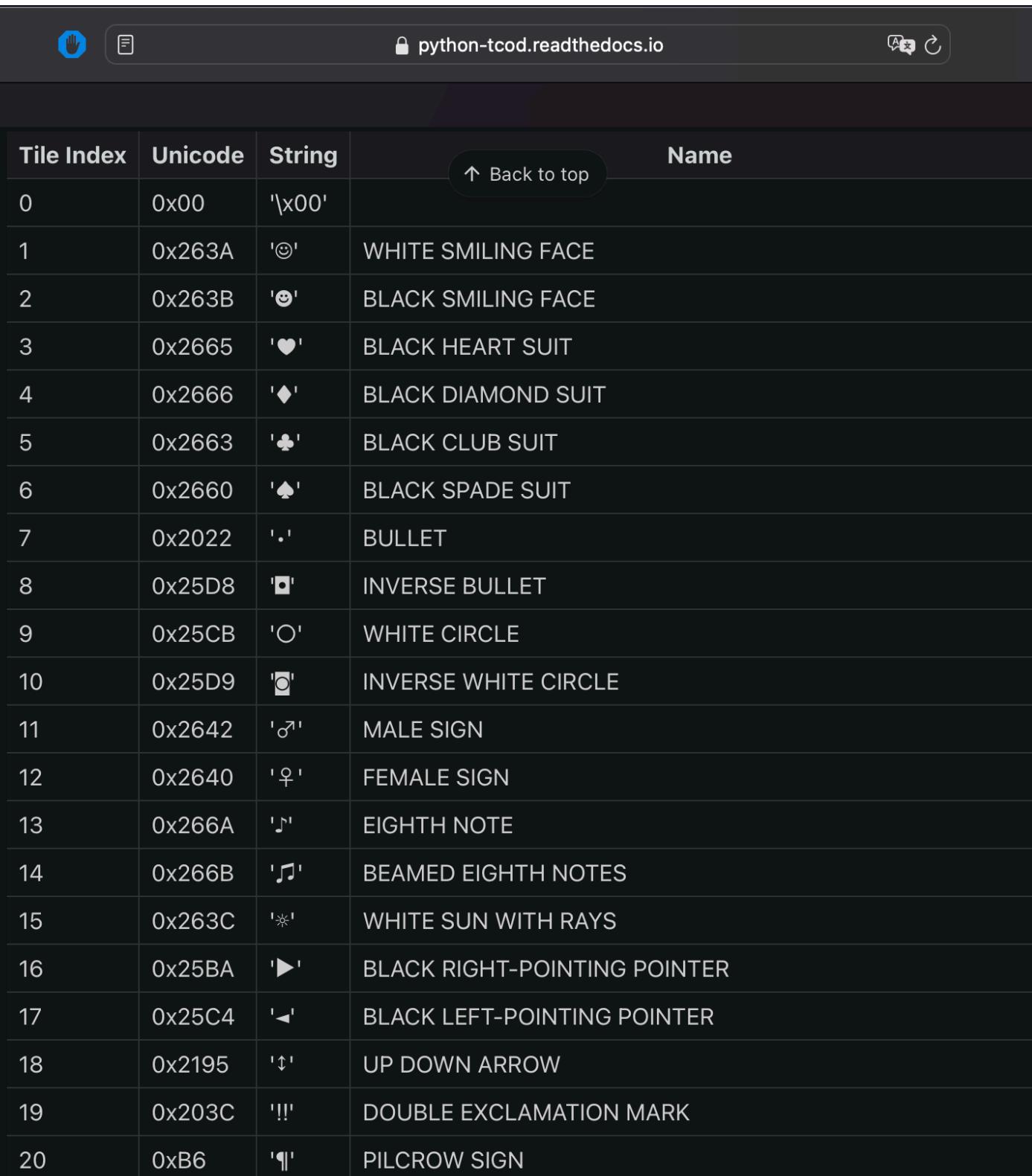
Thank You



Appendix

Unicode Table

`ord(c) inverse chr()`



The screenshot shows a table of Unicode characters from the Python tcod documentation. The table has columns for Tile Index, Unicode, String, and Name. The rows list various characters such as WHITE SMILING FACE, BLACK SMILING FACE, and various card suits.

Tile Index	Unicode	String	Name
0	0x00	'\x00'	
1	0x263A	'☺'	WHITE SMILING FACE
2	0x263B	'☻'	BLACK SMILING FACE
3	0x2665	'♥'	BLACK HEART SUIT
4	0x2666	'♦'	BLACK DIAMOND SUIT
5	0x2663	'♣'	BLACK CLUB SUIT
6	0x2660	'♠'	BLACK SPADE SUIT
7	0x2022	'•'	BULLET
8	0x25D8	'▣'	INVERSE BULLET
9	0x25CB	'○'	WHITE CIRCLE
10	0x25D9	'◐'	INVERSE WHITE CIRCLE
11	0x2642	'♂'	MALE SIGN
12	0x2640	'♀'	FEMALE SIGN
13	0x266A	'♪'	EIGHTH NOTE
14	0x266B	'♫'	BEAMED EIGHTH NOTES
15	0x263C	'☀'	WHITE SUN WITH RAYS
16	0x25BA	'▶'	BLACK RIGHT-POINTING POINTER
17	0x25C4	'◀'	BLACK LEFT-POINTING POINTER
18	0x2195	'↕'	UP DOWN ARROW
19	0x203C	'❗'	DOUBLE EXCLAMATION MARK
20	0xB6	'¶'	PILCROW SIGN

<https://python-tcod.readthedocs.io/en/latest/tcod/charmap-reference.html>