

# efl Data Science Course

## Introduction

# Lecturers



**M.Sc. Nicolas Pfeuffer**

*Research Assistant*

Nicolas Pfeuffer studied Business Informatics at the Goethe-Universität Frankfurt (M.Sc.). During his master's program,...



**M.A. Timo Schäfer**

*Research Assistant*

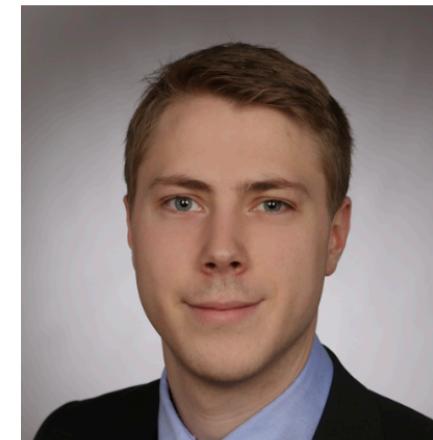
Timo Schäfer received his Master in Banking&Finance and Data Science from the University of Zurich in February 2018 and...



**M.Sc. Benjamin M.  
Abdel-Karim**

*Research Assistant*

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



**M.Sc. Jens Lausen**

*Research Assistant*

Jens Lausen received a Bachelor's degree in Management and Economics and a Master's degree in Management from Johannes...



# The efl



## Industry-academic partnership

### Universities



### Sponsors



# The efl



## 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

## Day 1 (Python Course) (05.10.2020)

**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) (07.10.2020)

**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) (9.10.2020)

**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

Day 1 (Python Course) (05.10.2020)
<b>9:00 - 10:30 Uhr</b> <b>Python Basics</b> Introduction and Primitive Data Types
<b>10:40 - 12:10 Uhr</b> <b>Data Structures</b> Lists, Sets, Dictionaries
<b>13:30 - 15:00 Uhr</b> <b>Control Structures</b> Loops (for, while), case distinction (if, else)
<b>15:15 - 16:45 Uhr</b> <b>Functions</b> Structure of Functions and Application

Day 2 (Python Course) (07.10.2020)
<b>9:00 - 10:30 Uhr</b> <b>Helpful functions for data processing</b> Libraries: os, re, csv
<b>10:40 - 12:10 Uhr</b> <b>Data types and data structures</b> Libraries: numpy, pandas
<b>13:30 - 15:00 Uhr</b> <b>Data import and visualization</b> Libraries: csv (cont'd), matplotlib
<b>15:15 - 16:45 Uhr</b> <b>Outlook: Data Science</b> Exemplary implementation of a KDD process

Day 3 (Data Science) (9.10.2020)
<b>9:00 - 10:30 Uhr</b> <b>Introduction to Data Science</b> Terminology and basic concepts
<b>10:40 - 12:10 Uhr</b> <b>Working with Data</b> Preprocessing, explorative data analysis
<b>13:30 - 15:00 Uhr</b> <b>Data Analysis I</b> Classification
<b>15:15 - 16:45 Uhr</b> <b>Data Analysis II</b> Neural Networks



efl -  
the Data Science Institute

CERTIFICATE OF  
PARTICIPATION

**INTRODUCTION TO  
PYTHON FOR DATA  
SCIENCE**

OCT · 09 · 2019

# Course Material?

The screenshot shows a GitHub repository page. At the top, there's a banner with the text "Learn Git and GitHub without any code!" and a "Read the guide" button. To the right of the banner is a blue callout box containing the text "You can download the course material as .zip file". Below the banner, the repository details are shown: owner "efl-the-data-science-institute", repository name "ds-courses", 3 watches, 0 stars, 1 fork, and 4 commits. The repository description states: "This is the repository for the material of the data science courses organized by the efl - the Data Science Institute." At the bottom of the screenshot, the URL <https://github.com/efl-the-data-science-institute/ds-courses> is provided.

# Why you are here

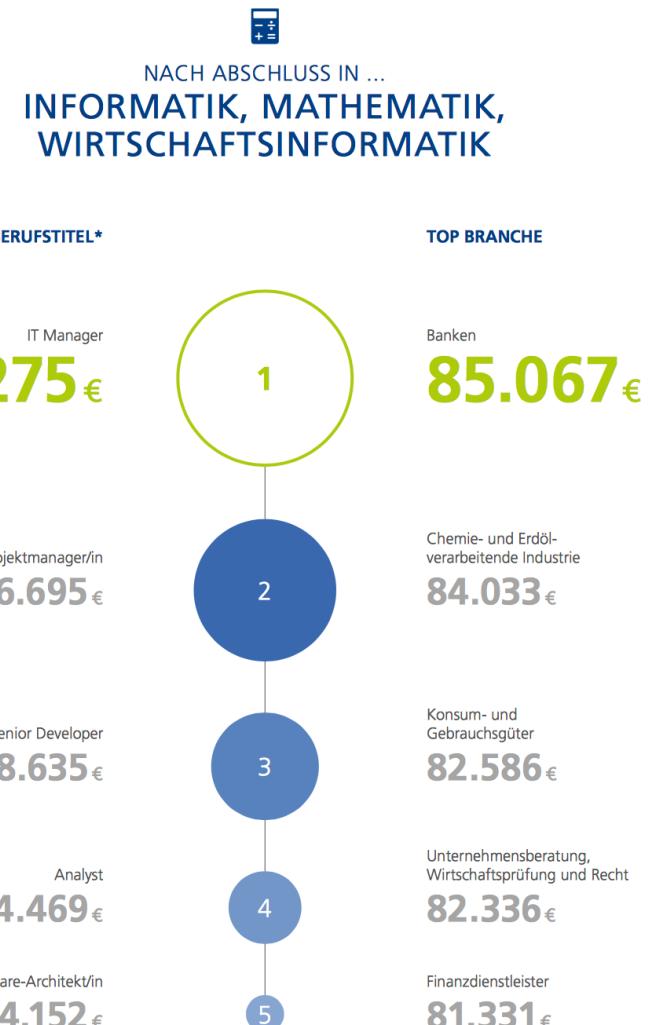
Kenntnis in Python und anderen Softwareprogrammen werden auf dem heutigen **Arbeits- und Praktikumsmarkt oft vorausgesetzt**. Um auch, über die in den Vorlesungen vermittelten Inhalte hinaus, etwas über statistische Softwareprogramme zu lernen, möchte ich an dem Kurs teilnehmen.

Ich möchte mich in diesem Bereich fortbilden, um bei einer Bachelor-Arbeit Daten auszuwerten und zu visualisieren. Darüberhinaus möchte ich wahrscheinlich **einen quantitativ-orientierten Master belegen und halte es daher für sinnvoll einen ersten Python-Kurs zu belegen.**

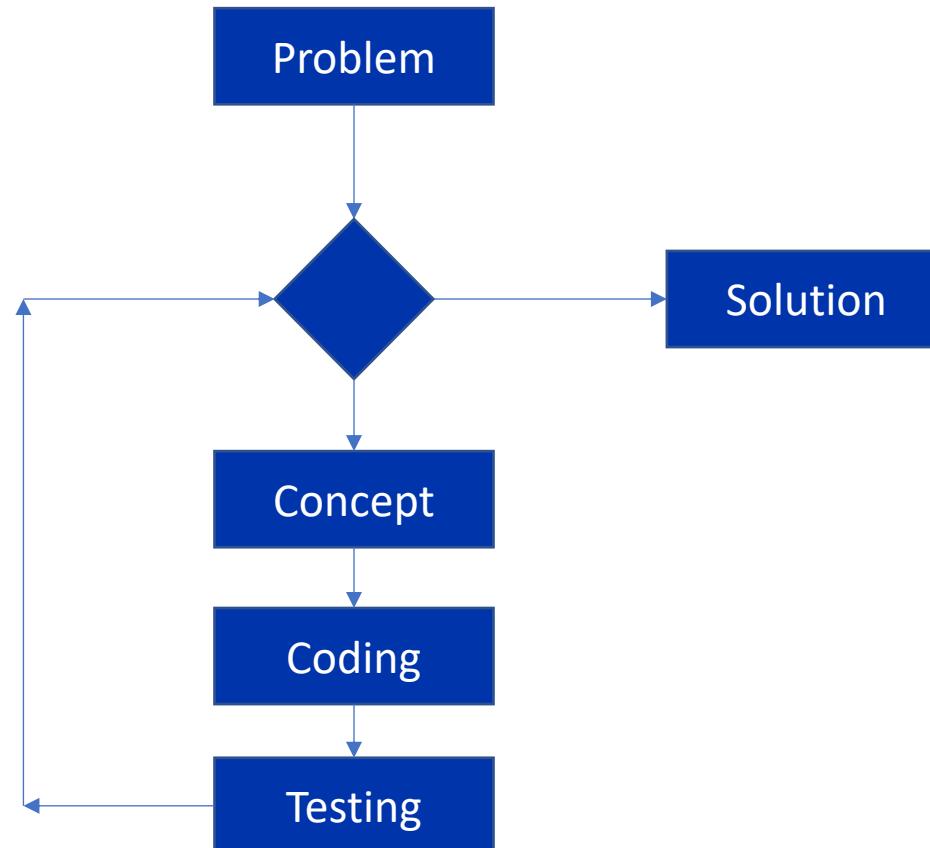
Auffrischen von Python, neue Einblicke in Software

*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 an dem Kurs teilnehmen, um mich beruflich weiterzubilden.*

# Why Coding?



# What is coding?



# 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).



[2]



[3]



[6]



[4]



[8]



[7]

# The elementary basics in Python

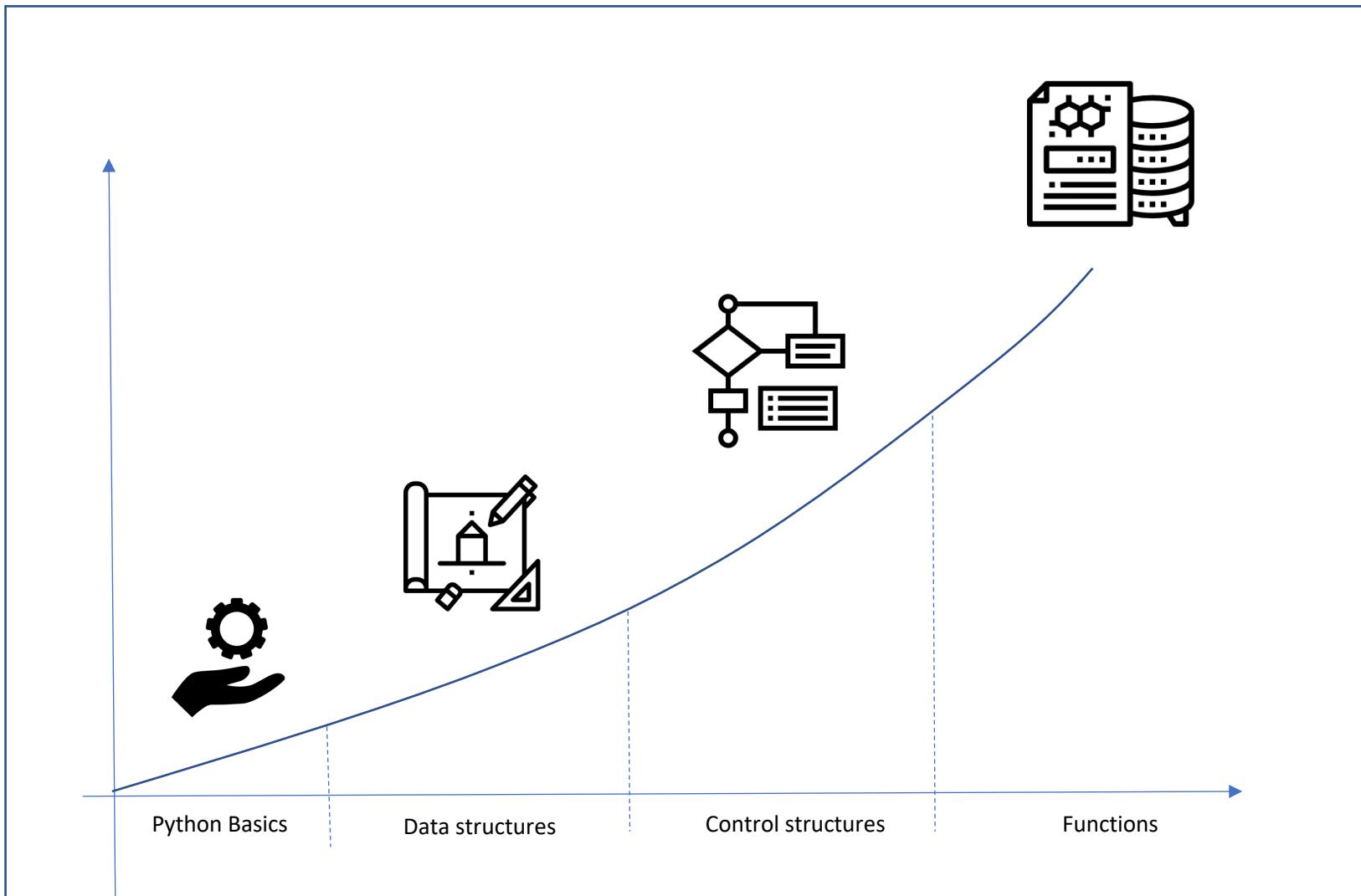
- Philosophy and difference to conventional programming languages
  - Higher programming language
  - It's simple
  - Fast to read
  - Structuring by indenting
  - No {} or ; => Faster to Code
  - Data types are managed dynamically. There is no static type check like in java
  - Widespread in science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

# Introduction goals

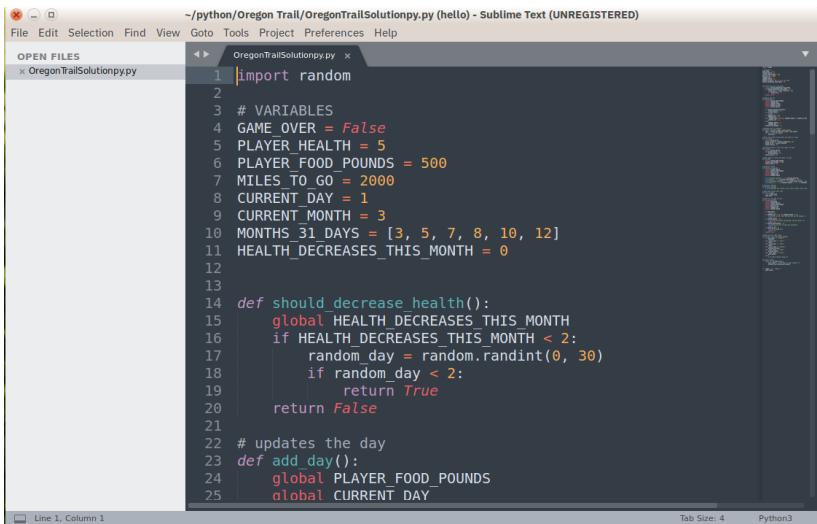


- Introduction to programming
- For beginners
- The module is interactive! Use your computer
- **We develop the solutions together!**
- Please be on time
- There are no dumb questions
- Nobody knows everything
- Copying solutions is plagiarism

# Learning Curve for Today

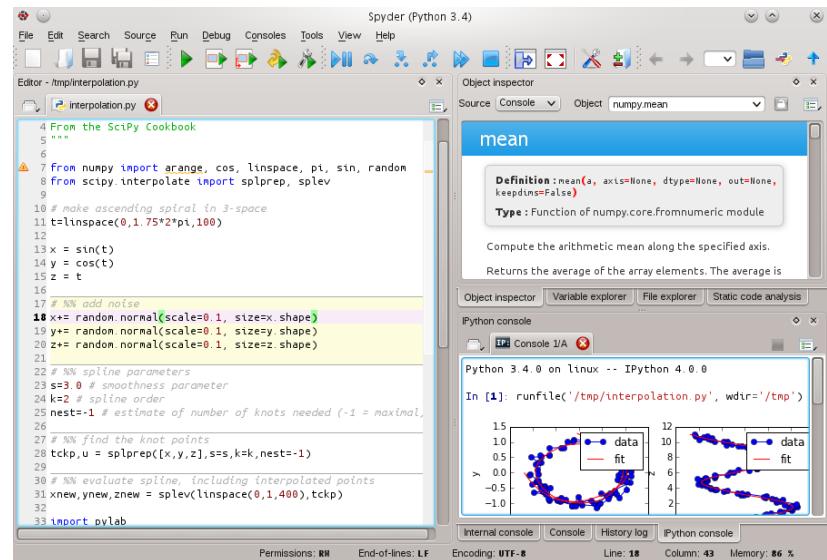


# IDE a deeper look? The Problem of Choice



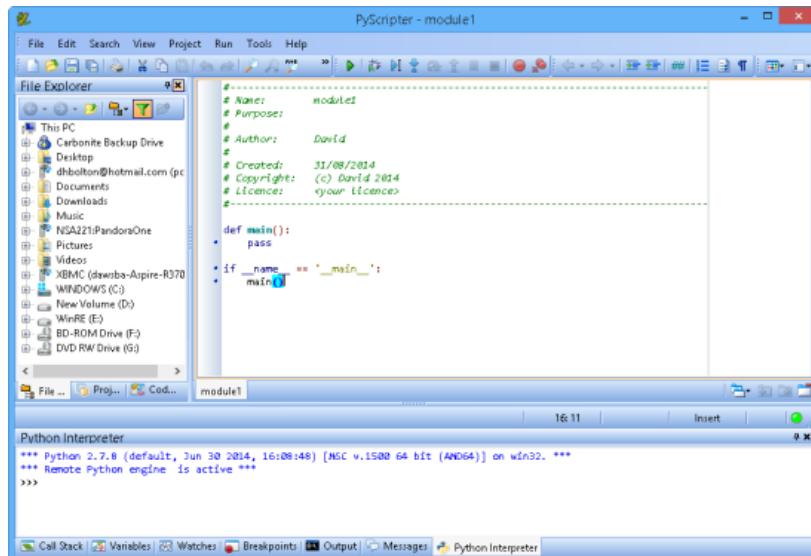
A screenshot of Sublime Text showing a Python script named `OregonTrailSolution.py`. The code defines variables like `GAME_OVER`, `PLAYER_HEALTH`, and `PLAYER_FOOD_POUNDS`. It includes a function `should_decrease_health` that checks if the player's health decreases this month. The script ends with a `def add_day` function.

```
1 import random
2
3 # VARIABLES
4 GAME_OVER = False
5 PLAYER_HEALTH = 5
6 PLAYER_FOOD_POUNDS = 500
7 MILES_TO_GO = 2000
8 CURRENT_DAY = 1
9 CURRENT_MONTH = 3
10 MONTHS_31_DAYS = [3, 5, 7, 8, 10, 12]
11 HEALTH_DECREASES_THIS_MONTH = 0
12
13
14 def should_decrease_health():
15     global HEALTH_DECREASES_THIS_MONTH
16     if HEALTH_DECREASES_THIS_MONTH < 2:
17         random_day = random.randint(0, 30)
18         if random_day < 2:
19             return True
20     return False
21
22 # updates the day
23 def add_day():
24     global PLAYER_FOOD_POUNDS
25     global CURRENT_DAY
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
```



A screenshot of Spyder showing a Python script named `interpolation.py`. The code uses NumPy and SciPy to generate a 3D spiral dataset and fit a spline to it. The IPython console shows the execution of the script and displays two plots: a scatter plot of the data points and a plot showing the fitted spline.

```
4 From the SciPy Cookbook
5 ***
6
7 from numpy import arange, cos, linspace, pi, sin, random
8 from scipy.interpolate import splprep, splev
9
10 # make ascending spiral in 3-space
11 t=linspace(0,1.75*2*pi,100)
12
13 x = sin(t)
14 y = cos(t)
15 z = t
16
17 # %% add noise
18 x+= random.normal(scale=0.1, size=x.shape)
19 y+= random.normal(scale=0.1, size=y.shape)
20 z+= random.normal(scale=0.1, size=z.shape)
21
22 # %% spline parameters
23 s=3.0 # smoothness parameter
24 k=2 # spline order
25 nest=-1 # estimate of number of knots needed (-1 = maximal,
26 # %% find the knot points
27 tkcp,u = splprep([x,y,z],s=s,k=k,nest=-1)
28
29
30 # %% evaluate spline, including interpolated points
31 xnew,ynew,znew = splev(linspace(0,1.400),tkcp)
32
33 import pylab
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
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
```



A screenshot of PyScripter showing a Python script named `module1`. The code includes a module header with author information and a main function that prints "main". Below the editor is a Python Interpreter window showing the Python version and a remote engine status.

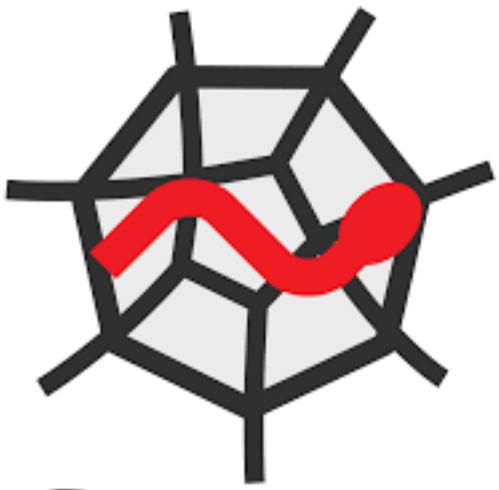
```
# Name: modules
# Purpose:
# Author: David
# Created: 31/08/2014
# Copyright: (c) David 2014
# Licence: your Licence

def main():
    pass

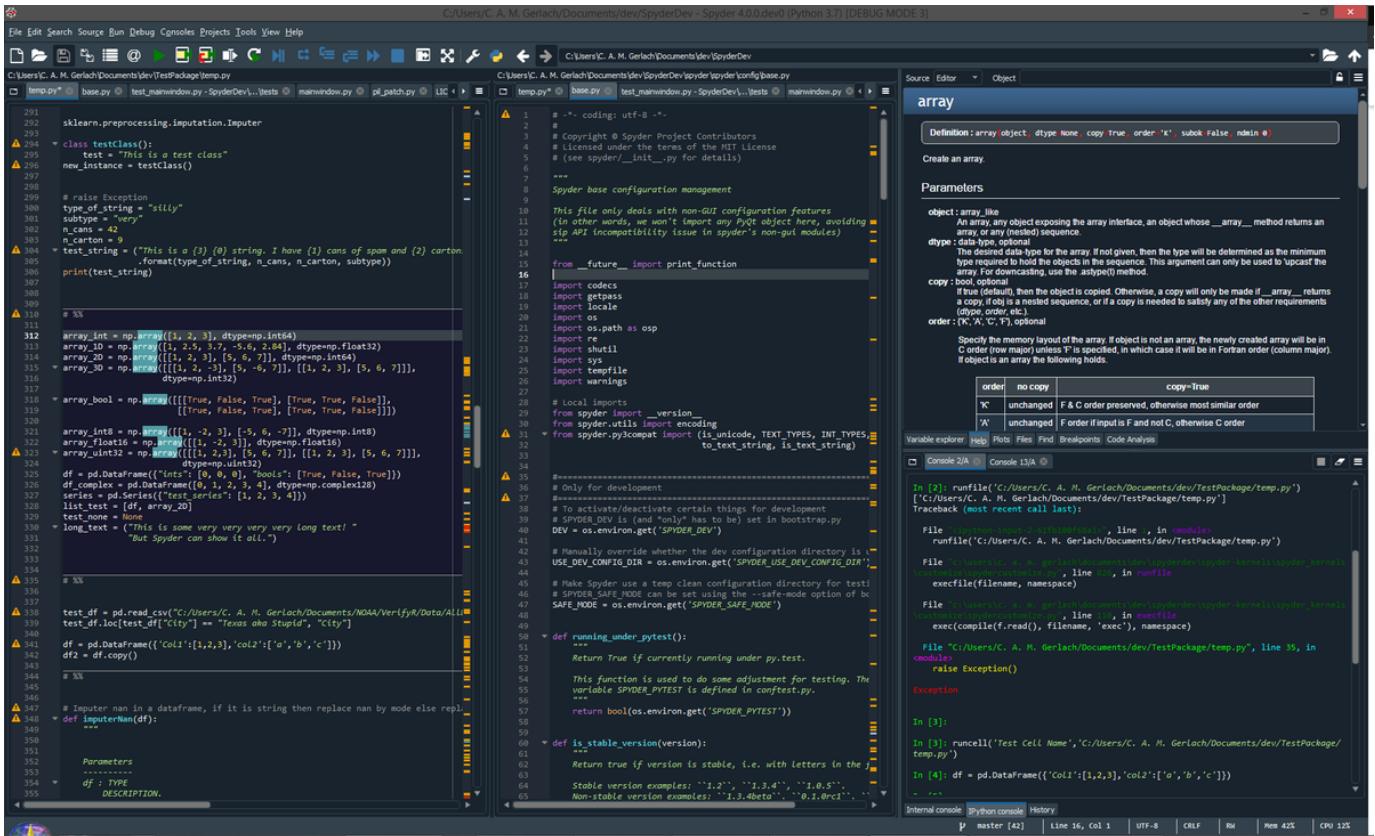
if __name__ == '__main__':
    main()

Python Interpreter
*** Python 2.7.8 (default, Jun 30 2014, 16:08:48) [MSC v.1500 64 bit (AMD64)] on win32 ***
*** Remote Python engine is active ***
>>>
```

# What We Use for This Course



# SPYDER



The screenshot shows the Spyder IDE interface. On the left is a code editor with Python code. The code includes imports from numpy, pandas, and scikit-learn, and demonstrates various array operations like reshaping and concatenation. In the center is a documentation sidebar for the `array` class, showing its definition, parameters, and detailed description. On the right is a console window displaying the output of running a script named `temp.py`.

```
File Edit Search Source Run Debug Cnsoles Projects Tools View Help
C:/Users/C. A. M. Gerlach/Documents/dev/TestPackage/temp.py
C:/Users/C. A. M. Gerlach/Documents/dev/SpyderDev.../Tests mainwindow.py pil_patch.py UIQ ... temp.py @ base.py @ test_mainwindow.py - SpyderDev.../Tests mainwindow.py ...
C:/Users/C. A. M. Gerlach/Documents/dev/SpyderDev spyder spyder config/base.py
C:/Users/C. A. M. Gerlach/Documents/dev/SpyderDev spyder spyder config/base.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 issues in spyder's non-gui modules)
13 """
14
15 from __future__ import print_function
16
17 import codecs
18 import getpass
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
32 # Only for development
33 # To activate developer settings for development
34 # SPYDER_DEV is (and *only* has to be) set in bootstrap-py
35 DEV = os.environ.get('SPYDER_DEV')
36
37 # Manually override whether the dev configuration directory is
38 USE_DEV_CONFIG_DIR = os.environ.get('SPYDER_USE_DEV_CONFIG_DIR')
39
40 # Make Spyder use a temp clean configuration directory for tests
41 # SPYDER_SAFE_MODE can be set using the --safe-mode option of b2
42 SAFE_MODE = os.environ.get('SPYDER_SAFE_MODE')
43
44 def running_under_pytest():
45     """
46     Return True if currently running under py.test.
47
48     This function is used to do some adjustments for testing. The
49     variable SPYDER_PYTEST is defined in conf/test.py
50     """
51     return bool(os.environ.get('SPYDER_PYTEST'))
52
53 def is_stable_version(version):
54     """
55     Return True if version is stable, i.e. with letters in the j
56     Stable version examples: "1.2", "1.3.4", "1.0.5"
57     Non-stable version examples: "1.3.4beta", "0.1.0rc1", ...
58     """
59
60
61
62
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
139
140
141
142
143
144
145
146
147
148
149
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
197
198
199
199
200
201
202
203
204
205
206
207
207
208
209
209
210
211
212
213
214
215
215
216
217
217
218
218
219
219
220
221
222
223
223
224
224
225
225
226
226
227
227
228
228
229
229
229
229
230
230
231
231
232
232
233
233
234
234
235
235
236
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
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
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1
```

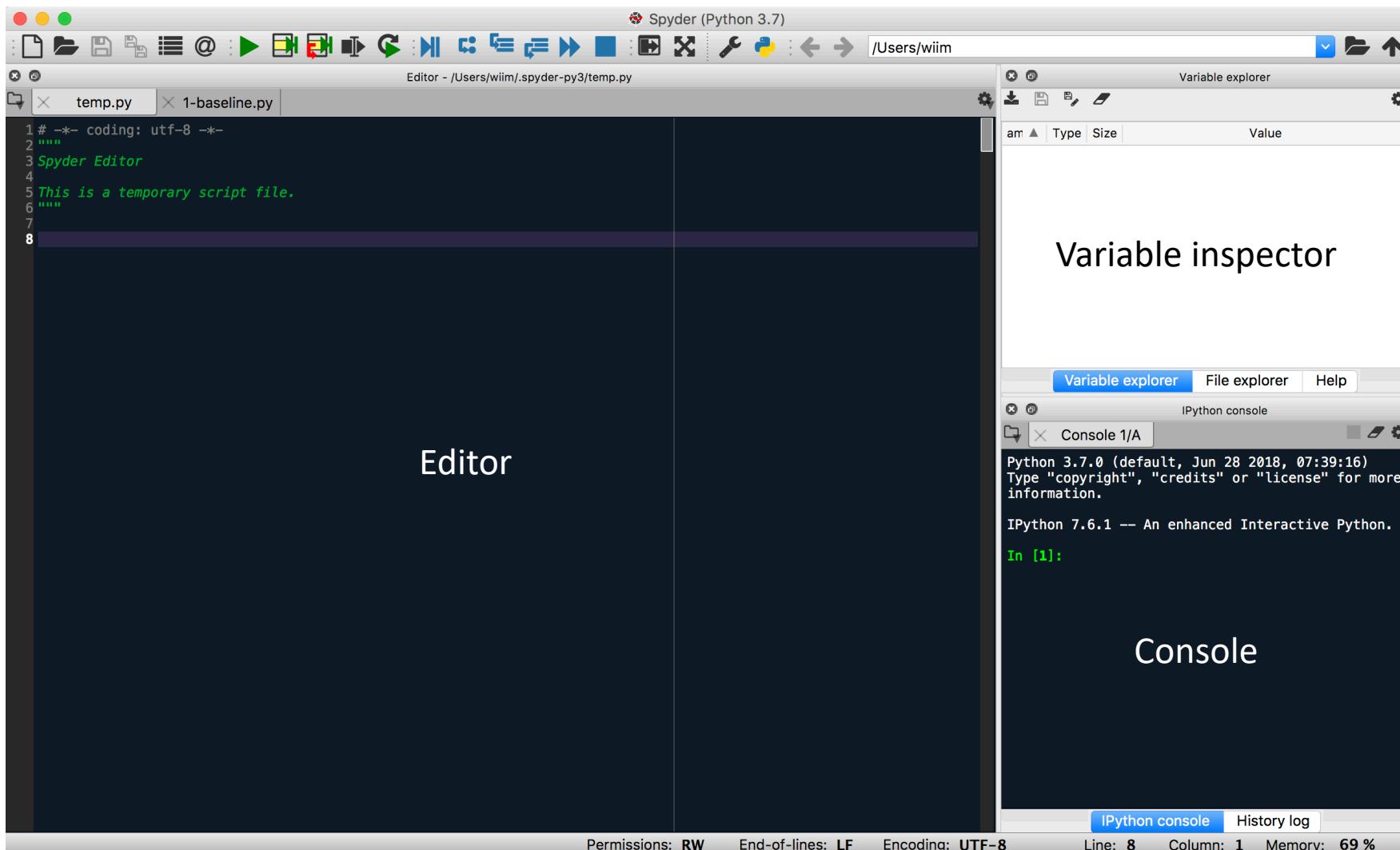
# Practice and Questions

**Take your computer and let's get started!**

Create a Folder 'Day 1' on our Desktop

Launch Spyder and select the folder

# Our Tool



# Hello World

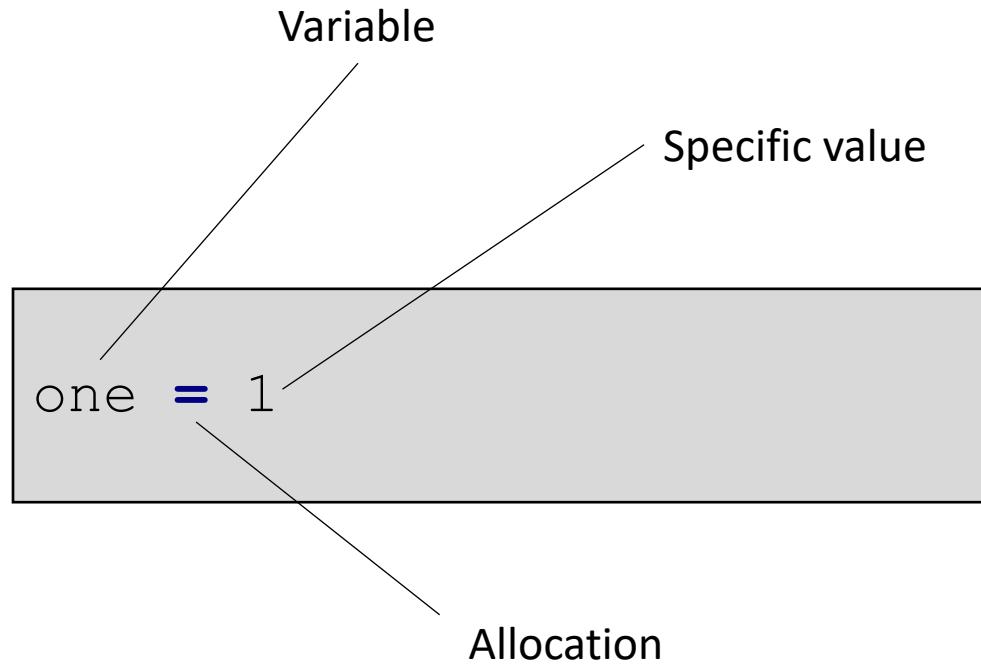
- Hello World
- Let's try our first example!
- Display "Hello World" on the console

# 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



# Numbers in Python



## Integer:

- int (signed integer) – They are often just called integers or ints, are positive or negative whole numbers with no decimal point.

```
iOne = 1
```

## Integer:

- Create two variables 'iOne' and 'iTTwo'.
- Assign the values 2 and 5 to each of these variables.
- Perform the following operations with the variables:
  - iMulti is the multiplication of the two variables.
  - iAdd is the addition of the two variables.
  - iSub is the subtraction of the two variables. Started with iOne.
  - iDiff is the division of iOne by iTTwo.

# Integer Action

```
iMulti = iOne * iTwo
print(iMulti)
```

```
#iAdd is the addition of the two variables.
iAdd = iOne + iTwo
print(iAdd)
```

```
# iSub is the subtraction of the two variables. Started with iOne.
iSub = iOne - iTwo
print(iSub)
```

```
# iDiff is the division of iOne by iTwo.
iDiff = iOne / iTwo
print(iDiff)
```

# Introduction to Primitive Data Types

## 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.
- Let's see how we can use float in python code.

```
dNumber = 1.0
```

## Integer:

- Create two variable 'dOne' and 'dTwo'. Assign the values 2.5 and 5.5 to each of these variables.
- Perform the following operations with the variables:
  - dMulti is the multiplication of the two variables.
  - dAdd is the addition of the two variables.
  - dSub is the subtraction of the two variables. Start with dOne.
  - dDiff is the division of dOne by dTwo.

# Float Action

```
# dMulti is the multiplication of the two variables.  
dMulti = dOne * dTwo  
print(dMulti)
```

```
#dAdd is the addition of the two variables.  
dAdd = dOne + dTwo  
print(dAdd)
```

```
# dSub is the subtraction of the two variables. Start with dOne.  
dSub = dOne - dTwo  
print(dSub)
```

```
# dDiff is the division of dOne by dTwo.  
dDiff = dOne / dTwo  
print(dDiff)
```



## Symbols in Python



# Chars

## Char:

- Chars are symbols!
- Char is short for character.
- All characters in all languages can be represented. This representation is in the **Unicode** format.
- Unicode is a computer encoding methodology that assigns a unique number for every character. It doesn't matter what language or computer platform it's on.
- Lets look at a new script!

Unicode code point	character	UTF-8 (dec.)	name
U+0000		0	<<control>
U+0001	1	1	<<control>
U+0002	2	2	<<control>
U+0003	3	3	<<control>
U+0004	4	4	<<control>
U+0005	5	5	<<control>
U+0006	6	6	<<control>
U+0007	7	7	<<control>
U+0008	8	8	<<control>
U+0009	9	9	<<control>
U+000A	10	10	<<control>
U+000B	11	11	<<control>
U+000C	12	12	<<control>
U+000D	13	13	<<control>
U+000E	14	14	<<control>
U+000F	15	15	<<control>
U+0010	16	16	<<control>
U+0011	17	17	<<control>
U+0012	18	18	<<control>
U+0013	19	19	<<control>
U+0014	20	20	<<control>
U+0015	21	21	<<control>
U+0016	22	22	<<control>
U+0017	23	23	<<control>
U+0018	24	24	<<control>
U+0019	25	25	<<control>
U+001A	26	26	<<control>
U+001B	27	27	<<control>
U+001C	28	28	<<control>
U+001D	29	29	<<control>
U+001E	30	30	<<control>
U+001F	31	31	<<control>
U+0020	32	32	SPACE
U+0021	!	33	EXCLAMATION MARK
U+0022	*	34	QUOTATION MARK
U+0023	#	35	NUMBER SIGN
U+0024	\$	36	DOLLAR SIGN
U+0025	%	37	PERCENT SIGN
U+0026	&	38	AMPERSAND
U+0027	'	39	APOSTROPHE
U+0028	(	40	LEFT PARENTHESIS
U+0029	)	41	RIGHT PARENTHESIS
U+002A	*	42	ASTERISK
U+002B	+	43	PLUS SIGN
U+002C	,	44	COMMA
U+002D	-	45	HYPHEN-MINUS
U+002E	.	46	FULL STOP
U+002F	/	47	SOLIDUS
U+0030	0	48	DIGIT ZERO
U+0031	1	49	DIGIT ONE
U+0032	2	50	DIGIT TWO
U+0033	3	51	DIGIT THREE
U+0034	4	52	DIGIT FOUR
U+0035	5	53	DIGIT FIVE
U+0036	6	54	DIGIT SIX
U+0037	7	55	DIGIT SEVEN
U+0038	8	56	DIGIT EIGHT
U+0039	9	57	DIGIT NINE
U+003A	:	58	COLON

# A unique number for every character.

- Try to get the unique number of ‘A’ as Unicode.

# A unique number for every character.

- Try to get the unique number of ‘A’ as Unicode.

```
print(ord(char))
```

# From Chars to Strings

- 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.

# Create two string variables

- Create two string variables:
  - `sWordOne = 'I Love'`
  - `sWordTwo = 'data'`
- Let's try to build a sentence with these two variables

# Concatenated string

```
# Concatenated string
sWordOne = 'I Love'
sWordTwo = 'data'
sStatement = sWordOne + ' ' + sWordTwo
print(sStatement)
```

# Built-in String Methods

- Python has a set of built-in methods that you can use on strings.
- How often does the word “love” appear in this sentence?

Note:

All string methods return new values.  
They do not change the original string.

**Outlook:** What are functions? This question we will discuss later in detail!

Here are just a few useful string operations for now!

# Built-in String Methods – Count()

```
iCountSubStrings = sStatement.count('Love')  
print(iCountSubStrings)
```

# Lower Case

- I would like to write everything in lower case
- Do I have to rewrite everything now?
- No, thanks to Built-in Methods

# Built-in String Methods – lower()

```
sStatementLower = sStatement.lower()  
print(sStatementLower)
```

$\circ$	$\neg_3 p$	$p$	$\neg_3^* p$	$\wedge_3$	$W$	$U$	$\lambda$
W	F	W	F	W	W	U	F
U	U	U	F	U	U	U	F
F	W	F	W	F	F	F	F

$\vee_3$	$W$	$U$	$F$
W	W	W	W
U	W	U	U
F	W	U	F

$\rightarrow_3$	$W$	$U$	$F$
W	W	U	F
U	W	U	U
F	W	W	W

	$W$	$U$	$F$
W	U	F	
U	U		
W			W

	$W$	$U$
W	W	U
U	F	
F		

Boolean



# Boolean and logical operators

- Boolean variable
- In computer science, the Boolean data type is a data type that has one of two possible values
- They are used to represent truth values (false or true).
- They are helpful for logical operations.

```
bBoolean = True
```

# Logical operations

- We can perform logical operations with True and False
- Let's try it: Execute all combinations with True and False with **and-operation**
- Tip:  $2^2$  possible combinations

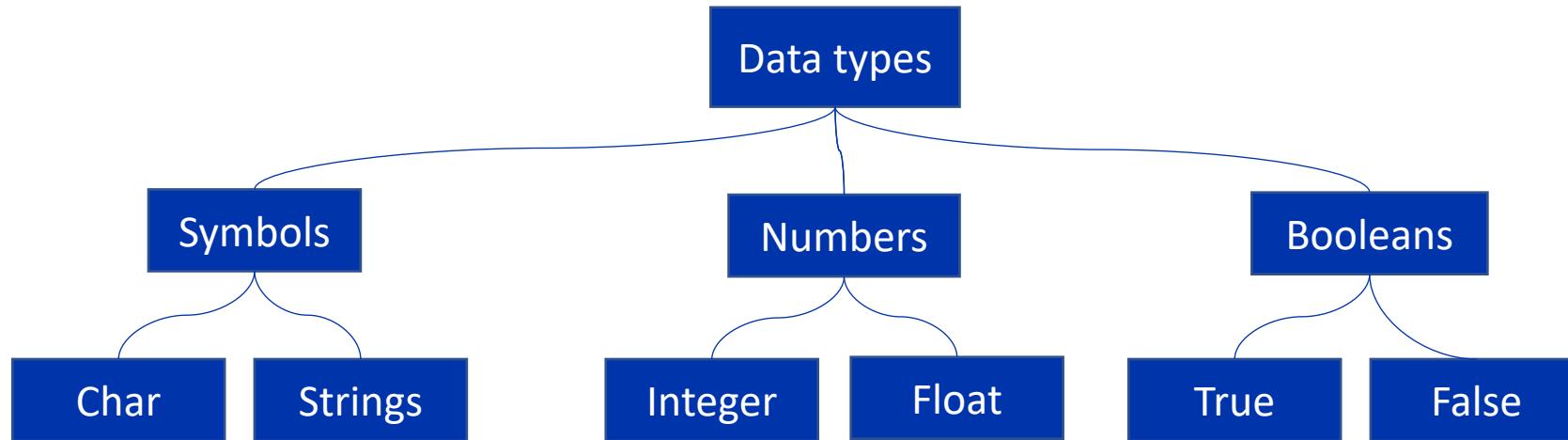
# Logical operations

```
bTrue = True  
bFalse = False  
print(bTrue)  
print('True and True is:', bTrue and bTrue)  
print('False and False is:', bFalse and bFalse)  
print('True and False is:', bTrue and bFalse)  
print('True and False is:', bTrue and bFalse)
```

\* For the sake of completeness: or is also a operator for True and False.

# Introduction to Primitive Data Types

- Systematization of primitive data types





# Formalities



- Source Code Documentation
- @author: name
- @since: first implementation date
- @version: date of last update
- @source: if you use links etc.
- @code: special code note
- @param: if special parameter is used, you have to describe.

```
# I am a comment
@author: unknown
@since: 20190929
@version: beta1
print('Hello World')
```

# 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
  - 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

# Naming convention and comments

- 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.

<https://www.python.org/dev/peps/pep-0008/>

# Naming convention, but python ...

▲ for everything related to Python's style guide: i'd recommend you read [PEP8](#).

175 To answer your question:

▼ Function names should be lowercase, with words separated by underscores as necessary to improve readability.

[12]



<https://www.python.org/dev/peps/pep-0008/>

# Code is some kind of Art. Therefore...

But we are using the presented naming convention!

These are all approaches. Find your own style!