

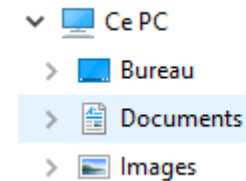
Fonctionnement de la séquence

- Le fichier que vous êtes en train de lire va vous permettre d'apprendre les bases de python à votre rythme.
- Cette séquence donnera lieu à plusieurs notes dont une évaluant votre capacité à respecter les consignes écrites et orales.
- Pour faciliter votre travail comme votre évaluation, vous allez commencer par ouvrir l'explorateur de fichier, vous placer dans votre répertoire document créer un répertoire SNT (si ça n'a pas déjà été fait) et à l'intérieur créer un répertoire python. (si ça ne vous semble pas évident, n'ayez crainte dans la diapo suivante il y aura une explication.
- Durant la séquence vous stockerez vos programmes et vos fichiers Word contenant vos pense bêtes et vos proposition pour les évaluations.
- Il est vivement conseillé de faire une copie de ce répertoire dans votre clé USB au cas où. Vous êtes l'unique responsable de vos fichier et de l'organisation de votre compte sur les ordinateurs du lycée.

Tuto : Créer un répertoire/dossier



- Vous trouverez l'icône de l'explorateur de fichier sur la barre en bas de votre écran. Cliquer dessus.
- Vous trouverez le répertoire document à gauche de votre écran. Cliquer dessus.
- Vous allez voir apparaître le contenu actuel de votre répertoire document. Pour créer un répertoire, vous faites un click droit (bouton droit de la souris) et vous choisissez l'option « Nouveau » puis « Dossier »
- Vous verrez apparaître un nouveau répertoire, avec un texte surligné en bleu disant « nouveau dossier » remplacez ce texte par « Python ».
- Si vous voulez changer le nom d'un répertoire , vous cliquez une ou deux fois sur le nom actuel jusqu'à ce que celui-ci soit surligné en bleu, et là vous pouvez effectuer votre modification.



Tuto : faire une copie d'écran

- Vous allez devoir vous faire un pense bête où vous écrirez tout ce que vous avez appris durant la séance. Ce pense bête vous permet de réviser, mais aussi de retrouver rapidement l'information quand vous l'avez oubliée sans avoir à parcourir ce diaporama (qui est plutôt long)
- Vous pouvez noter l'information en toute lettre, mais vous pouvez aussi copier des morceaux d'écran et les coller dans votre fiche.
- Etape 1
- Chercher sur votre clavier la touche « imprim écran » et appuyer dessus. (le contenu complet de ce qui est présenté sur l'écran est maintenant en mémoire)
- Lancer le programme paint (on clique sur l'icone en bas à gauche de la page) et on sélectionne le programme paint (son icone est une palette de peinture)
- Copier le contenu de la mémoire dans paint (on appuie simultanément sur les touches « Ctrl » et « V »)
- On sélectionne la zone de l'image nous intéressant en traçant autour de celle-ci un cadre avec notre souris. Puis on copie cette zone encadrée en mémoire (on appuie simultanément sur les touches « Ctrl » et « C »)
- Dans le traitement de texte on choisira où on veut mettre notre image puis on la collera avec Ctrl+V

PRISE EN MAIN DE PYTHON

Python est un langage de programmation , pour l'utiliser il nous faut utiliser une des nombreux environnements de travail le contenant.

Sur LorDi

Il vous faudra télécharger le fichier « PortablePython_3.2.5.1.exe » sur le site www.dimension-k.com en suivant le [lien](#), puis il vous faudra l'exécuter ce qui provoquera l'installation vers le répertoire de votre choix (ce qui prendra encore un peu de temps).

Alternative :

Lancer la suite de logiciel MCNL

Dans le menu choisir : Maths , puis à droite cliquer sur + sous Python et choisir Spyder (très lent)

Il apparaît alors un fenêtre séparée en plusieurs parties dont :

Une partie édition de programme

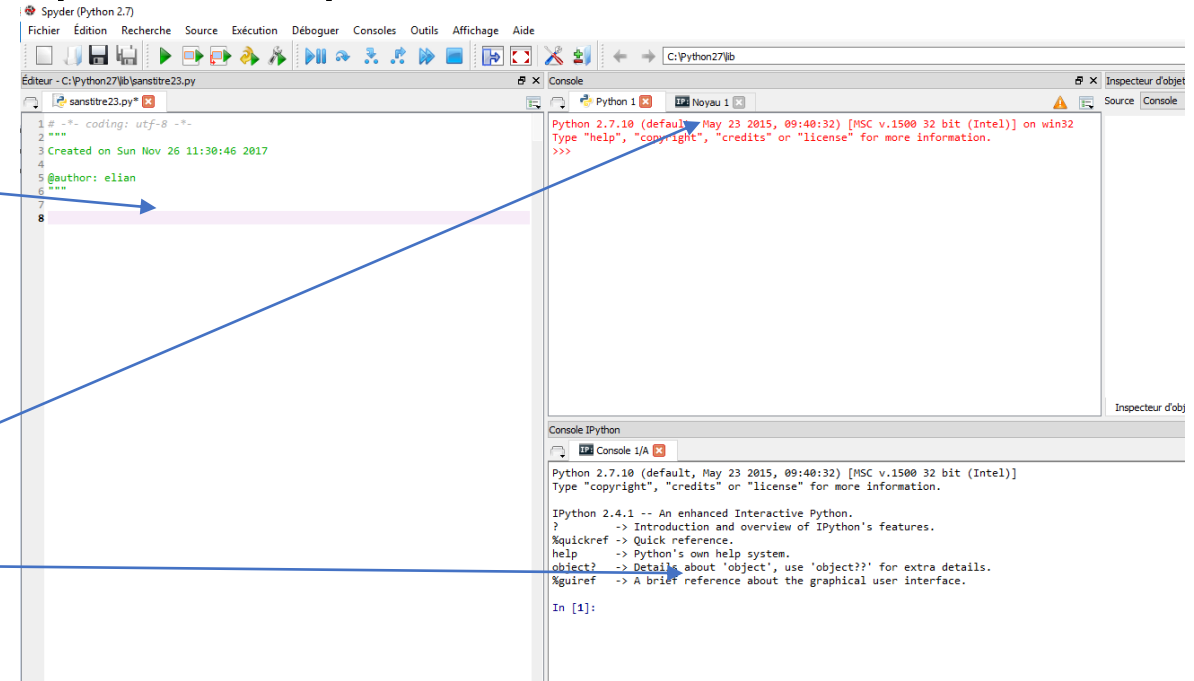
Cette partie permet d'éditer des programmes (à utiliser quand ils sont longs)

Une partie console

(il y en a deux ici) à utiliser pour

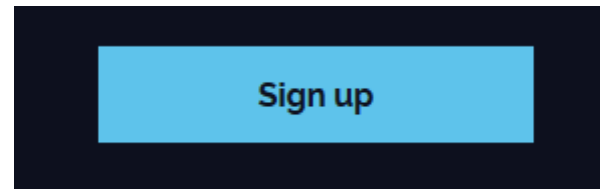
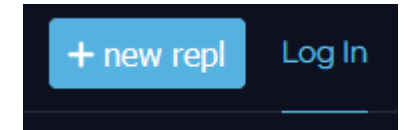
effectuer des calculs, c'est ici que sera écrit les traitements de vos algorithmes (réponses de vos programmes)

Avec la version installé, pyscripter vous fournira une interface équivalente



Utiliser Repl.it (1/3)

- À l'aide d'un navigateur internet, on se rend à l'adresse : repl.it
- Dans la partie supérieure de l'écran on a deux boutons :
- + new repl veut dire lancer une nouvelle session
- Log in : permet de se connecter (ce qui n'est possible que si on a un compte)
- Au milieu de l'écran il y a un bouton permettant de créer un compte :





KERGOT777

silencio2046@hotmail.com

••••••••|

SHOW



I'm a teacher

or log in

Sign up

By continuing, you agree to Repl.it's [Terms of Service](#) and [Privacy Policy](#), and to receiving emails with updates.

▼ My Profile



KERGOT777



First Name

Last Name

School or company name



Add a Bio

0 / 140 characters

Save

Utiliser Repl.it (2/3)

Processus d'inscription :

Première page

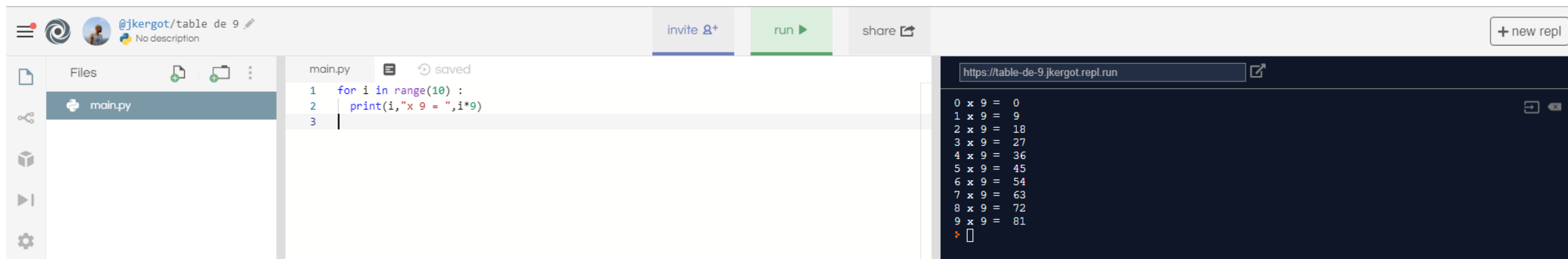
- Après avoir appuyé sur « Sign up »
- Pour username j'ai choisi KERGOT777 qui sera mon alias
- Sur le site, chaque utilisateur a un alias unique
- Email : mon adresse email
- Password : votre mot de passe (vous le noterez dans votre téléphone ou un autre endroit sûr, histoire de le retrouver en cas d'oubli.
- En tant qu'élève il ne faut pas cocher la case « I'm a teacher »

Seconde page (ou en cliquant sur votre username, puis sur (edit your profile))

- Après vous devez donner votre Prénom (First Name) et votre Nom de famille (Last Name)
- Pour la rubrique School or company name, commencez par taper Lycee Albert Einstein et rapidement on vous proposera une liste d'établissement et vous pourrez choisir votre lycée dans la liste.

Utiliser Repl.it (3/3)

- Après avoir choisi l'option : « + new repl », choisi comme langage Python, et choisi le nom du projet, on obtient un écran comme celui de la capture suivante :



- On peut y voir 3 zones :
 - Colonne de droite : la console. C'est un environnement où l'on peut tester nos commandes python, et on peut aussi étudier les conséquences du lancement d'un programme.
 - Colonne centrale : l'éditeur de programme, on y écrit des lignes de codes qui seront compilées et lancées quand on appuiera sur le bouton vert « Run »
 - Colonne de gauche, la gestion de l'environnement, on y trouve main.py le programme principal, mais on peut ajouter des fichiers supplémentaires, c'est de là qu'on pourra commander l'intégration de bibliothèques supplémentaires dans le noyau python (ce qui nous donnera le droit de les appeler dans notre programme)
- Notre travail s'enregistre en temps réel, et si l'on travaille à partir de notre compte on pourra toujours retrouver ce que l'on a programmé en cherchant « My Repls » dans le menu (on accède à celui-ci en cliquant sur l'icone en haut à gauche présentant trois lignes horizontales superposées).

I : LES NOMBRES

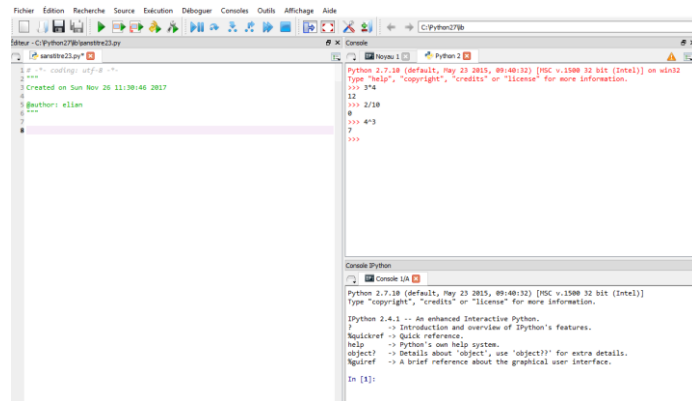
Dans la console on peut effectuer toute sorte de calculs.

Effectuez les calculs suivants : 3×4 , $2/10$, $2.7/10$, 4^3

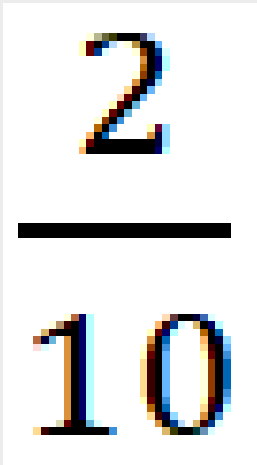
Vérifiez les résultats affichés par la console.

Rappel : sur la calculatrice pour calculer 4^3 , on Tape 4^3

Que pouvez vous observer ?



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Nov 25 11:30:46 2017
4 @author: ellan
5 """
6
7
8
9
10
11
12
13
14
15
16
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
```


$$\frac{2}{10}$$

Pour la division :

Python n'est pas un logiciel de mathématiques,

Dans **sa version 2**, lorsqu'on manipule des nombres entiers, il retournera des nombres entiers.

Pour que Python prenne un entier pour un réel, il suffit de rajouter un point à la suite du nombre :

2 est un entier alors que 2.0 ou 2. est considéré comme un réel.

Pour éviter cela il faut taper 2/10. (mettre un point à la fin de 2/10)

ou taper float(2)/float(10) (plus lourd)

Remarque : dans la version 3 la question ne se pose même pas

Pour la puissance:

On remarque qu'il a ajouté 4 et 3, pour éviter cela il suffit de taper

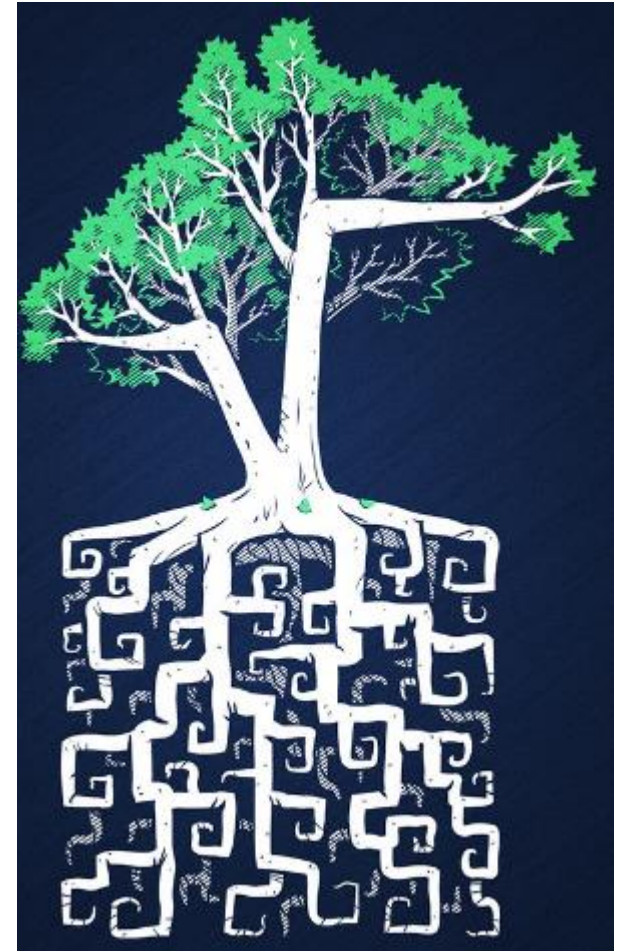
`4**3`

Pour faire un calcul utilisant une racine on peut se rappeler qu'en anglais racine carrée se dit « square root », deux mots qui seront abrégés sur Géogébra comme dans la plus part des langages de programmation par sqrt.

En python la racine carrée sera aussi effectuée avec :
`sqrt(nombre dont on veut la racine)`

Tentez d'effectuer $\sqrt{3}$.

Que se passe-t-il ?



La console nous a affiché le message suivant :

```
>>> sqrt(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
>>> |
```

Autrement dit, il ne reconnaît pas la fonction `sqrt()`.

Contradiction ?

Non, en fait la fonction `sqrt()` n'étant utilisée que dans des contextes assez rares, elle existe mais n'est pas accessible de manière directe, elle est rangée dans une bibliothèque « `math` » qu'il faudra activer.

```
>>> from math import sqrt
>>> sqrt(3)
1.7320508075688772
>>>
```

L'utilisation de librairies permet d'enrichir le langage Python d'une foule de fonctions plus ou moins utiles (gestion de la carte son, des différents cœurs du processeur, dessin 3D, animations, ...)

Charger toutes ces fonctions alourdirait les programmes : ça les rend lents et gourmands en ressources (mémoire vive et espace disque) c'est pour cela qu'on préfère les ranger dans des groupes de fonctions appelées librairies et quand on en a besoin on peut charger en mémoire (rendre utilisable) toutes les fonctions ou on peut comme dans l'exemple précédent charger une seule fonction de la librairie.

Astuce : pour charger toutes les fonctions on peut écrire :

from math import * (* veut dire tout)

Les librairies les plus utilisées sont

Math : pour ce qui est du calcul

Numpy : pour ce qui est des graphiques, listes...

Matplotlib et pylab

Random : pour tout ce qui concerne les probabilités

PIL: traitement des images

Joblib : calcul parallèle

Sympy : pour ce qui est du calcul formel (équations,...)

Et pour vous cette année : Turle qui permet d'apprendre en s'amusant

Variables

En programmation on est amené à enregistrer le résultat de nos calculs ou des valeurs importante dans des variables.

Pour stocker la valeur 5 dans la variable « nombre » il suffit d'écrire :

`nombre = 5`

Si on tape `nombre*3` l'ordinateur nous répondra : 15

Pour stocker le résultat de l'opération `7-9x3` dans la variable « number » :

`number = 7-9x3`

Variables

- Prévoir ce que python nous écrira à la fin de chaque ligne.

```
A = 17
```

```
B=2*A
```

```
A = A-5
```

```
print(A)
```

```
print(B)
```

Conclusion : l'assignation est statique (tant qu'on ne demande pas de manière explicite le changement de la valeur d'une variable celle-ci restera égale à elle même)

Exercices

1/Expliquer en détails ce qu'il se passe ci-contre

```
In [47]: p=10
```

```
In [48]: q=10.
```

```
In [49]: p==q
```

```
Out[49]: True
```

```
In [50]: p is q
```

```
Out[50]: False
```

```
In [51]: type(p)
```

```
Out[51]: int
```

```
In [52]: type(q)
```

```
Out[52]: float
```

2/Effectuer sur la console les calculs suivants de telle sorte que les résultats soient donnés sous forme décimale

calcul 1 : $\frac{\sqrt{3}}{4}$

calcul 2: $\frac{135}{124}$

calcul 3: $\left(\frac{5}{3}\right)^3$

II Les TEXTES

Si on veut taper bonjour dans la console, celle-ci croit qu'on fait référence à une variable s'appelant bonjour et donc affichera un message d'erreur si une telle variable n'a pas été définie au préalable.

```
Python 2.7.10 (default, May 23 2015, 09:40:32) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> bonjour
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'bonjour' is not defined
>>>
```

Pour éviter le message d'erreur, il faut savoir qu'un texte, autrement dit une chaîne de caractères se met toujours entre guillemets.

Donc il faut taper "bonjour",

L'opérateur + appliqué à des chaînes de caractères les concatène (les colle à la suite l'une de l'autre)

Questions :

1) avec cette logique devinez ce que Python vous affichera si vous écrivez "Bonjour"*2

2) Prévoir le retour à la fin de la séquence suivante :

mot = " les gens"

"bonjour "*2+mot

3) en considérant la capture d'écran suivante déduire une règle concernant la multiplication d'une chaîne de caractère par un nombre.

```
In [15]: b*1.5
Traceback (most recent call last):

File "<ipython-input-15-ca2237451620>", line 1, in <module>
    b*1.5
TypeError: can't multiply sequence by non-int of type 'float'
```

4) A quoi sert la fonction len?

```
In [20]: machaine="moi tarzan"

In [21]: len(machaine)
Out[21]: 10

In [22]: machaine1="jane"

In [23]: len(machaine1)
Out[23]: 4
```

- Pour accéder à un caractère d'une chaîne, il faudra faire suivre le nom de la chaîne par un crochet contenant l'index du caractère.

Voir l'exemple à droite :

```
In [43]: texte="Je suis heureux d'apprendre Python et de progresser"
In [44]: texte[3]
Out[44]: 's'

In [45]: texte[0]
Out[45]: 'J'

In [46]: texte[2]
Out[46]: ' '
```

On remarquera que l'index est légèrement décalé par rapport à ce qu'on pouvait penser

En fait pour les chaînes de caractères et les listes le premier élément a pour index 0, le suivant a pour index 1 ainsi de suite.

On peut prendre plusieurs caractères d'une chaîne à la fois

5) Après avoir testé les commandes suivantes :

texte[3:7] texte[8:] texte[:13]

Vous indiquerez à quoi servent les commandes suivantes :

chaîne[i:j] chaîne[i:] chaîne[:j]

6) On donne la capture d'écran,
En déduire l'utilité de la méthode `count`

Une méthode est une sorte de fonction un peu étrange.

Elle agit sur un objet et son action est généralement en relation avec des paramètres.

Dans ce qui précède : `texte.count("u")` l'objet est la chaîne de caractère s'appelant **texte**

Le paramètre c'est le caractère "u"

```
In [24]: texte="Je suis heureux d'apprendre Python et de progresser"
```

```
In [25]: texte.count("eu")
```

```
Out[25]: 2
```

```
In [26]: texte.count("u")
```

```
Out[26]: 3
```

```
In [27]: texte.count("p")
```

```
Out[27]: 3
```

7) Copier le texte suivant et stockez le dans une variable nommée **Brassens** les premières strophes de « la mauvaise réputation »:

Au village, sans prétention, J'ai mauvaise réputation ; Que je me démène ou je reste coi, Je pass' pour un je-ne-sais-quoi. Je ne fais pourtant de tort à personne, en suivant mon ch'min de petit bonhomme ; Mais les brav's gens n'aiment pas que l'on suive une autre route qu'eux... Non, les brav's gens n'aiment pas que l'on suive une autre route qu'eux... Tout le monde médite de moi, sauf les muets, ça va de soi. Le jour du quatorze-Juillet, je reste dans mon lit douillet ; la musique qui marche au pas, cela ne me regarde pas. Je ne fais pourtant de tort à personne, en n'écouter pas le clairon qui sonne ; mais les braves gens n'aiment pas que l'on suive une autre route qu'eux... non les braves gens n'aiment pas que l'on suive une autre route qu'eux... Tout le monde me montre du doigt, sauf les manchots, ça va de soi.

Comptez le nombre d'occurrence du mot "gens"

8) Petite synthèse

En reprenant la variable `brassens` indiquez le nombre de fois où apparaît le 44^{ème} caractère de ce texte.

III Les listes

C'est une structure très importante en Python

Elle est délimitée par des crochets

C'est en fait une collection ordonnée de différents objets, qui peuvent avoir un type différent (texte, nombre). Chaque objet est séparé du suivant par une virgule.

Exercices

1) Ci-contre on a la séquence suivante

Que faut il taper si on veut renvoyer :

- Le premier objet de la liste ?
- L'avant dernier objet de la liste?

2) Quelles sont les différences
avec les chaînes de caractères?

```
In [64]: maliste=[1,2,'lol',3.14,"e"]
```

```
In [65]: maliste[1]
```

```
Out[65]: 2
```

```
In [68]: maliste[-1]
```

```
Out[68]: 'e'
```

```
In [69]: len(maliste)
```

```
Out[69]: 5
```

```
In [70]: liste2=["soupe",1,2.41,"pas bon"]
```

```
In [71]: maliste+liste2
```

```
Out[71]: [1, 2, 'lol', 3.14, 'e', 'soupe', 1, 2.41, 'pas bon']
```

```
In [72]: maliste.count(1)
```

```
Out[72]: 1
```

3) A quoi sert la fonction append?

Est-ce que cette fonction peut permettre d'insérer entre le 2 et le 3 le mot « coucou »? Pourquoi?

4) compléter la commande [77], pour insérer « coucou » entre le 2 et le 3

Même chose pour insérer le mot « coucou » avant le 1

5) On est arrivé en commande 82, on veut enlever le mot « coucou » qu'il y a entre le 2 et 3, compléter la commande 83, est ce que ça marche ?

Il existe une autre fonction pour enlever des éléments de la liste. Explorer le fonctionnement de la fonction del , par exemple « del liste[4] »

```
In [74]: liste=[1,2,3]
In [75]: liste.append(6)
In [76]: print(liste)
[1, 2, 3, 6]
```

```
In [76]: print(liste)
[1, 2, 3, 6]
```

```
In [77]: liste.insert(|
```

Arguments

insert(index, object)

```
In [82]: print(liste)
['coucou', 1, 2, 'coucou', 3, 6]
```

```
In [83]: liste.remove(|
```

Arguments

remove(value)

6) Expliquer en détails, ce qu'il se passe entre les commandes 103 et 106.

Expliquer en détails, ce qu'il se passe

Entre les commandes 107 et 110.

Y a-t-il des différences de traitement entre les objets de type nombre et les objets de type liste?

```
In [103]: c=5
In [104]: d=c
In [105]: d=d+1
In [106]: print(c,d)
(5, 6)
In [107]: c=[1,2,3]
In [108]: d=c
In [109]: d.append(2019)
In [110]: print(c,d)
([1, 2, 3, 2019], [1, 2, 3, 2019])
```

```
In [114]: c=[1,2,3]
In [115]: d=c[:]
In [116]: d.append(2019)
In [117]: print(c,d)
([1, 2, 3], [1, 2, 3, 2019])
```

A gauche on a résolu le problème précédent, expliquer la commande 115.

Quelques listes de base

7) en python 2 range(n) renvoie tous les entiers de 1 à n-1.

```
In [34]: print(range(10))  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

pour avoir le même résultat sous python 3, on utilisera la commande list(...) qui convertira l'objet obtenu en liste utilisable.

Qu'obtiendra t on en tapant : list(range(7))

8) Range peut prendre 1,2 ou trois paramètres.

Après avoir testé range(5,10) , range(20,52) , range(20,52,2) et range(20,52,5)

Expliquer à quoi correspond range(i,j) , range(i,j,k)

Approche alternative :

en important la librairie numpy,

la liste arange(a,b,c) permet de générer

la liste des nombres entre a et b-c, avec

un pas de c

```
In [38]: from numpy import *
```

```
In [39]: print(arange(10,15,0.1))
```

```
[ 10.   10.1  10.2  10.3  10.4  10.5  10.6  10.7  10.8  10.9  11.   11.1  
 11.2  11.3  11.4  11.5  11.6  11.7  11.8  11.9  12.   12.1  12.2  12.3  
 12.4  12.5  12.6  12.7  12.8  12.9  13.   13.1  13.2  13.3  13.4  13.5  
 13.6  13.7  13.8  13.9  14.   14.1  14.2  14.3  14.4  14.5  14.6  14.7  
 14.8  14.9]
```

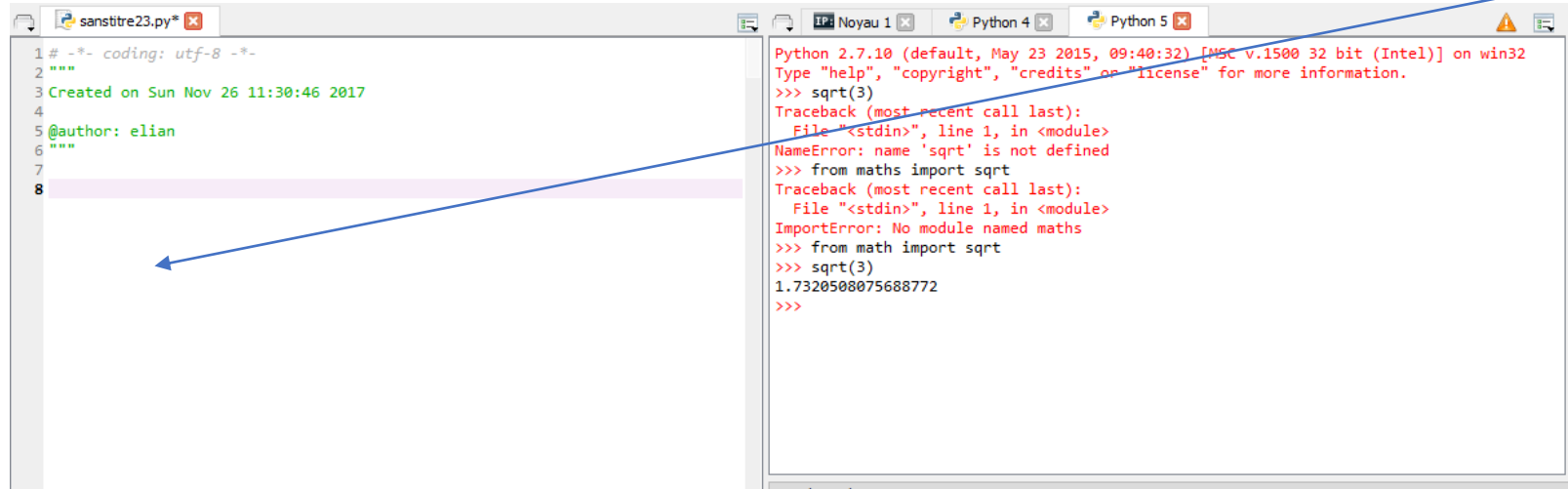
9) Comment créer la liste suivante : [-5,-4.8,-4.6,-4.4, ... , 17.6] ?

10) proposer une liste des entiers de 13 à 955

11) proposer une liste des nombres décimaux arrondis au dixième allant de 147 à 3201

IV Les programmes

Si on veut construire un programme on va utiliser la partie script



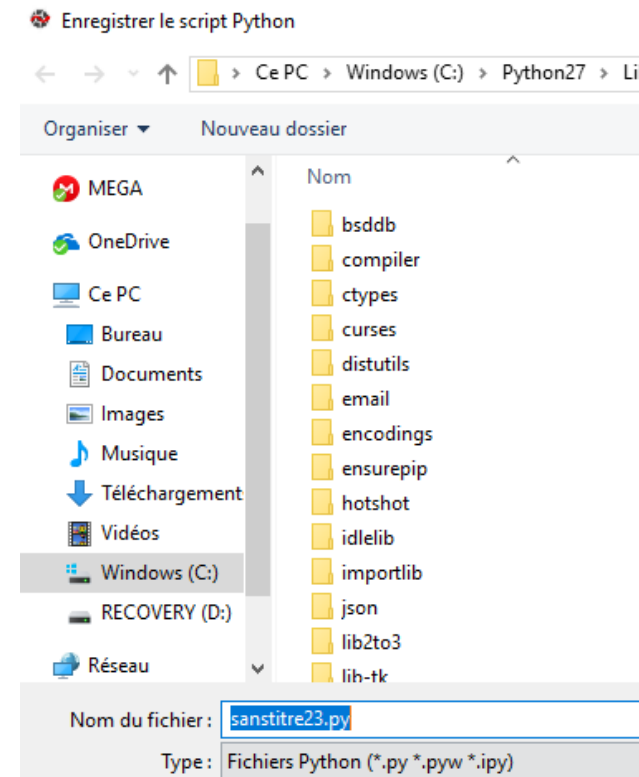
Une fois que le programme est écrit, on l'enregistre, puis on clique sur le bouton Play (flèche verte) présent dans la bande supérieure (s'il vous pose des questions, appuyez juste sur OK).

L'exécution (question, affichage) se fait dans la console.

En cas de problème on peut interrompre l'exécution du programme en appuyant sur le carré bleu.

Pour faire fonctionner le programme (script) il suffit de taper sur la flèche d'exécution le logiciel propose d'enregistrer le script. Pour pouvoir vous y retrouver, il est conseillé de créer un répertoire où vous allez stocker tous les programmes fait durant la séquence

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Nov 26 11:30:46 2017
4
5 @author: elian
6 """
7 x=input("valeur de x")
8 from math import*
9 y=(sqrt(x)+4)**2
10 print(y)
11
```



Ton premier programme

Généralement un programme est composé de trois types d'instructions :

- **Entrée** : on va récupérer de l'information extérieure à travers le clavier, la souris, l'écran (quand il est tactile), la caméra, le micro.
- **Traitement** : on utilise des formules et des instructions sur les informations récupérée avant de le rendre.
- **Sortie** : on va offrir à l'utilisateur une production au travers de l'écran, des enceintes...

Exemple :

```
nom = input(" quel est ton nom ? ")
```

```
phrase = " Bonjour " +nom+ ", j'espere que tu  
vas bien "
```

```
print(phrase)
```

Copier/coller ce programme et vérifiez son bon fonctionnement.

Exercice 1 : A toi de jouer : *dans fichier, sélectionne nouveau , pour créer un nouveau programme*

Ecris un script (programme) demandant à une personne son nom, son prénom, puis le parfum de sa glace préférée et qui en retour affichera une phrase du type « Bonjour, Julien Kergot, ton parfum de glace préféré est chocolat »

Programme utile en mathématiques

Le professeur voudrait que les élèves de sa classe tracent la courbe de la fonction f qui à tout réel positif x associe le nombre : $(\sqrt{x} + 4)^2$

Pour commencer il demande à ses élèves de remplir un tableau de valeurs.

Cette situation est un brin répétitive, et l'informatique peut nous aider à remplir le tableau plus rapidement.

Dans un premier temps crée un programme qui va demander à l'utilisateur la valeur de x et qui lui donnera en retour la valeur de $f(x)$

Remarque : tu pourras t'aider de ce qui a été vu sur les opérations de base et la fonction `sqrt()` découverte précédemment.

*Correction : voir deux diapos plus haut ! M. Bellante a choisi d'utiliser `import *` plutôt que `import sqrt`, avec ce choix il peut utiliser n'importe quelle fonction de la librairie `math`.*

Exercices

```
8 x=input('lettre')
9 texte=input("texte")
10 n=texte.count("x")
11 print(n)
```

2) On a envie de faire un programme comptant le nombre de fois où une lettre est répétée dans un texte.

Un élève propose le script ci-dessus.

Recopiez le et testez le avec les deux configurations suivantes :

Lettre : « e » et texte : « xavier est gentil »

Lettre : « w » et le texte « le rappeur xxxtentacion est mort en juin 2018 à l'âge de 20 ans»

Quelle réponse obtenez vous pour les deux cas ?

Quelle lettre est ce que le programme a compté (au lieu de « e » et « w »)?

Déterminez où est l'erreur et corrigez la.

$$\begin{array}{r|l} 247 & 22 \\ -22 & \\ \hline 27 & 11 \\ -22 & \\ \hline 5 & \end{array}$$

3) ci contre vous avez la division de 247 (dividende) par 22 (le diviseur) qui nous donne un reste de 5 et un quotient de 11.

Dans la console taper : `int(247/22)` puis `247%22`

Créer un programme demandant à l'utilisateur un dividende et un diviseur et qui en retour donnera la valeur du quotient et du reste.

Structure conditionnelle : si alors sinon

Exemple : dire d'un triangle dont on connaît les trois mesures s'il est rectangle ou non.

On sait que si le carré de la plus grande mesure est égale à la somme des carrés des mesures des deux autres côtés alors le triangle est rectangle (réciproque du théorème de Pythagore) et sinon il n'est pas rectangle (contraposée du théorème de Pythagore)

On se propose d'écrire un programme demandant à l'utilisateur les trois mesures du triangle et de tester l'égalité citée plus haut, et suivant si elle est vraie ou non de dire si le triangle est rectangle ou non.

On va donc être amené à faire un test, et si celui-ci est concluant d'exécuter certaines instructions et dans le cas contraire d'en exécuter d'autres.

Voici le script du programme répondant à la consigne de la diapositive précédente.

Décortiquons le :

pour les trois premières lignes,

on remarquera qu'en plus de

input on utilise la fonction float

elle convertit le nombre proposé par l'utilisateur en réel.

```
1 a=float(input("le plus grand côté"))
2 b=float(input("autre côté"))
3 c=float(input("dernier côté"))
4 if a**2==b**2+c**2:
5     print("le triangle est rectangle")
6 else :
7     print("le triangle n'est pas rectangle")
```

À la ligne 4 on pose notre test. « si » se traduit en anglais par « if ».

On remarquera qu'entre les deux membres il y a un double égal : ==

Nous, on veut tester l'égalité, or en Python un signe « = » sert à assigner une valeur à une variable, donc la convention pour tester une égalité sera le double égal : ==

À la fin de l'égalité on a écrit « : », ces deux points servent à introduire un bloc d'informations qui ne sera exécuté que si ce qui précède les deux points est vrai.

Une fois qu'on a écrit les deux points et qu'on appuie sur « entrer » l'éditeur de programme va à la ligne et « indente » le texte (ce qui veut dire qu'il provoque un décalage automatique de quatre espaces vers la droite)

Tant que le texte est décalé on est dans le bloc qui ne s'exécutera que si la condition est vraie,

si on se ramène au bord gauche l'instruction s'exécutera dans tous les cas.

À la ligne 6 on est revenu contre le bord gauche donc elle s'exécute même si la condition n'est pas réalisée d'ailleurs l'instruction est « else » ce qui veut dire « sinon », et elle indique que le bloc qui va suivre ne s'exécutera justement que si la condition de la ligne 4 n'est pas réalisée.

```
1 a=float(input("le plus grand côté"))
2 b=float(input("autre côté"))
3 c=float(input("dernier côté"))
4 if a**2==b**2+c**2:
5     print("le triangle est rectangle")
6 else :
7     print("le triangle n'est pas rectangle")
```

On peut tester autre chose que des égalités.

Pour des conditions complexes il faudra généralement décomposer en plusieurs conditions simples.

Deux exemples :

- Si on veut savoir si $x \in [8; 12]$, il faut d'abord l'exprimer sous forme d'inégalités : $8 \leq x \leq 12$, puis décomposer cet encadrement : il revient à

dire que x vérifie : $8 \leq x$ et $x \leq 12$ sur python on écrira donc `if 8<=x and x<=12 :`

- Si on veut savoir si n'est pas dans $[8; 12]$ on peut utiliser deux approches :

Ne pas être dans $[8; 12]$ est le contraire d'être dans $[8; 12]$ donc en Python on écrira :

`if not (8<=x and x<=12) :`

Ou

ne pas être dans $[8; 12]$ revient à être soit plus petit que 8 ou strictement plus grand que 12, donc on écrira : `if x<8 or x>12 :`

Nom	Python
est égal à	<code>a == b</code>
n'est pas égal à	<code>a != b</code>
est plus grand que	<code>a > b</code>
est plus petit que	<code>a < b</code>
est plus grand ou égal à	<code>a >= b</code>
est plus petit ou égal à	<code>a <= b</code>
ET logique	<code>a and b</code>
OU logique	<code>a or b</code>
négation (NOT)	<code>not a</code>

Exercices

4) On sait que la distance d'arrêt sur sol sec d'un véhicule roulant à la vitesse v (en km/h) est la somme de la distance parcourue durant le temps de réaction ($0,3 \times v$) et de la distance de freinage ($0,0075 \times v^2$)

Créer un programme demandant à l'utilisateur la vitesse du véhicule et la distance séparant le véhicule de l'obstacle, et qui en retour affichera après un court calcul et une comparaison « désolé mais ton véhicule a embouti l'obstacle » ou « tu as réussi à éviter l'obstacle mais reste vigilant! ».

```
In [30]: from random import randint
```

5) La librairie random permet d'importer la fonction randint

```
In [31]: randint(1,6)  
Out[31]: 4
```

Si on tape « randint(1,6) », l'ordinateur permet de ressortir un nombre entier aléatoire entre 1 et 6. Bref cela permet de jouer au dé.

On veut programmer un assistant pour jouer aux petits chevaux. A chaque tour le joueur lance un dé à six faces. Si son cheval est dans l'écurie il ne peut l'en sortir que si le dé affiche un 1 ou un 6. Si le cheval est déjà sorti de l'écurie on avance du nombre de case affiché par le dé.

Le programme simule le lancé d'un dé, affiche le résultat obtenu et affichera « si ton cheval est dans l'écurie tu as le droit de le sortir » ou « si ton cheval est dans l'écurie, désolé mon coco mais il va y rester un tour de plus ! »

6) Un jeu de cartes a 32 cartes (autrement dit il y a quatre familles, et on a retiré les cartes 2, 3, 4, 5 et 6.

Construire un programme permettant de simuler une partie de ce jeu. Pour plus de commodité on numérote les 32 cartes, les 8 premières sont les piques, les huit suivantes des cœurs, puis des trèfles, et enfin des carreaux. Elles sont rangées dans l'ordre suivant : as, 7, 8, 9, 10, valet, dame, roi. Ainsi la carte numéro 2 est un 7 de pique, la carte numéro 8 est un as de cœur, ...

Version facile : Le programme doit simuler un tirage et dire si on a gagné (on a tiré un as) ou si on a perdu (dans le cas contraire).

Version un peu plus difficile : le programme doit aussi indiquer la famille et la nature de la carte

Astuce : on remarquera que si on fait une division euclidienne du numéro de carte par 8, les cartes de même nature donnent le même reste (par exemple les as sont en position 1, 9, 17 et 25, et quand on divise ces nombres par 8, on obtient à chaque fois un reste de 1.

LES BOUCLES POUR

```
In [52]: for k in range(10):
...:     print('salut')
...:     print('aurevoir')
...:
salut
salut
salut
salut
salut
salut
salut
salut
salut
salut
aurevoir
```

Une boucle pour permet de recommencer des instructions, un nombre déterminé de fois

Un exemple

Dans l'algorithme ci-contre l'action que l'on recommence 10 fois est d'afficher le texte 'salut'.

En fait, il y a 10 même action numérotées de 0 à 9

qui se renouvelle. Attention l'instruction qui se renouvelle est mise en retrait sous le k de 'for k in range(10)'

Une fois que les commandes de la boucle sont terminées, le programme passe à la suite des instructions. L'instruction afficher 'aurevoir' est écrite sous le for, donc elle s'effectuera après cette boucle

```
In [53]: for k in range(10):  
....:     print('salut')  
....:     print('au revoir')  
....:
```

```
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir  
salut  
au revoir
```

Un autre exemple: dans cet exemple, comme les instructions `print('salut')` et `print('au revoir')` sont écrites l'une après l'autre, sous le 'k' de '`for k in range(10)`'. ON a donc 10 affichages successifs de 'salut' et 'au revoir'

Exercices

7) Construire un algorithme permettant de simuler 10 lancers d'un dé à six faces (à chaque lancer le programme écrira le résultat du lancer)

8) Reprendre le programme précédent pour qu'à chaque lancer il affiche « gagné » (si on a tiré un 6) ou « perdu » (pour les autres tirages).

9) Analyser le programme ci-contre

```
for k in range(4):  
    L=L+[randint(1,6)]  
print(L)
```

 et indiquer ce qu'on peut s'attendre à avoir en sortie. A quoi pourrait servir un tel programme?

10) Proposer un algorithme permettant de jouer 5000 fois au dé, il doit permettre d'afficher le nombre de fois où on a gagné (obtenu 6)

11) Proposer un programme permettant de simuler 1 tirage du loto (c'est-à-dire 5 nombres pris entre 1 et 49)

Version 1 : on s'autorise à tirer plusieurs fois le même nombre

Fonctionnement interne de la boucle for

On peut avoir l'impression que les boucles précédentes se contentent de répéter 10 fois le bloc qui est indenté.

C'est le cas, mais la situation est plus riche que ça.

`range(10)` correspond comme on a pu le voir précédemment à une sorte de liste allant de 0 à 9.

Ce que la boucle fait, c'est qu'à chaque tour la variable « `i` » va parcourir la liste et prendre une valeur différente : 0 puis 1 puis 2 ... puis 9.

Si on écrit le code ci-contre

```
for i in range(10) :  
    print(i)
```

, on obtiendra la sortie ci-contre.

```
>>>  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

On n'est pas obligé d'utiliser la fonction `range` pour faire une boucle « `for` », n'importe quelle liste fera l'affaire.

Exemple : tapez le script ci-dessous et observez l'affichage quand on l'exécute.

```
mois=0  
for i in ["janvier", "février", "mars"] :  
    mois=mois+1  
    phrase=i+" est le mois n°"+mois+" de l'année"  
    print(phrase)
```

Exercices

12) Reprendre le programme 7 (deux diapos plus haut) pour qu'à chaque lancer il fasse une phrase du type : « le lancer n°5 vaut 4 »

13) En vous inspirant des propriétés de la fonction range, afficher successivement les nombres entre 500 et 563 en allant de 3 en 3 (c'est-à-dire 500 puis 503 puis 506 puis ...)

14) Reprendre l'exercice « programme utile en mathématiques » qui s'intéresse à la fonction $(\sqrt{x} + 4)^2$ de telle sorte qu'il fasse plus que donner l'image d'un nombre unique (il demandera un x minimum, un x maximum et un pas, puis avec une boucle il va afficher des lignes de la forme « $f(0)=16$ », « $f(1)=25$ », etc)

15) Dans une équipe de volley féminin les joueuses ont pour nom : Alice, Nawel, Lou, Alexandra, Candice et Shaïna. Faire un programme affichant consécutivement 6 phrases écrites sur le modèle suivant : « Alice a 5 lettres dans son prénom »

rappel : pour déterminer le nombre de lettre dans le mot stocké dans la variable nom on écrit : `len(nom)`

Palindromes

Un palindrome est un mot qui se lit de la même manière qu'on aille de la droite vers la gauche ou de la gauche vers la droite. Quelques exemples connus : rotor, xanax, kayak

Le but de l'exercice est d'écrire un programme permettant de dire si un mot est un palindrome. Comme ça n'a rien d'une évidence, on vous donne pour vous guider un algorithme commenté :

Algorithme

Demander à l'utilisateur **mot**

On détermine **n** le nombre de lettres dans le mot.

On pose **palin**=True

I prendra ses valeurs entre 0 et n

Si la lettre de rang **i** n'est pas la même que la lettre en position symétrique

Alors **palin** prend la valeur False

(pas de Sinon)

On affiche le contenu de la variable **palin**

commentaire

mot est le mot dont ton cherche à savoir s'il est un palindrome.

Voir dernière ligne de la diapo précédente

la variable **palin** dit si le mot est un palindrome et on considère que jusqu'à preuve du contraire, le mot en est un.

on utilise une boucle pour parcourir les lettres du mot les unes après les autres

il suffit qu'il y ait un couple de lettres non symétriques pour que mot ne soit pas un palindrome

si les deux lettres sont identiques, il n'y a rien à faire, ça ne change en rien ce qu'on a déterminé précédemment

Une fois que la boucle est terminée

1) Donner pour un rang **i** d'une lettre , le rang de la lettre symétrique en fonction de **i** et de **n** (le nombre de lettre du mot)

Petit exemple pour y voir plus clair sur les indices des lettres.

« vietnamien » est un mot de 10 lettres, elles seront numérotées de 0 à 9, ainsi mot[1] vaut « i »

La position symétrique de la lettre 0 est 9, la position symétrique de la lettre 1 sera 8 etc

2) Traduire l'algorithme sous la forme d'un programme en Python

Boucles tant que (while)

On rappelle que les instructions déclarées dans la boucle tant que, se feront tant que la condition reste vraie. Une fois que celle-ci n'est plus vrai, on passe à l'instruction suivante

Souvent, on utilise la boucle tant que, pour parvenir à une fin bien précise, dans l'esprit suivant

Tant que je rate j'essaie encore

C'est une boucle non bornée, en effet cette boucle ne peut jamais s'arrêter. Et oui on peut toujours rater le test

```
In [1]: t=5
```

```
In [2]: while t<10:  
....:     t=t+1  
....:
```

```
In [3]: print (t)  
10
```

Algorithme 1

Dans l'algorithme 1 :t au départ vaut 5,le but est d'obtenir Une valeur de t plus grande ou égale à 10, Tant que $t < 10$; l'instruction est de remplacer la valeur de t par $t+1$, ce qui fait donc augmenter t. de ce fait on a toute les chances pour réussir à être plus grand ou égal à 10


```
In [4]: t=5
```

```
In [5]: while t>6:  
...:     t=t+1  
...:
```

```
In [6]: print(t)  
5
```

Algorithme 2

Le but est d'être inférieure ou égal à 6, or t est au départ égal à 5.

Donc comme la condition d'entrée dans la boucle tant que, n'est pas vérifiée, la valeur de t ne bouge pas. On passe donc à l'instruction suivante, c'est-à-dire écrire t qui vaut 5

Petite devinette

Expliquer ce qui se passe dans cet affichage

```
In [8]: while t<10:  
...:     print('boujour')  
...:     |  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
boujour  
Traceback (most recent call last):  
  
File "<ipython-input-8-1dfa3bde9ddb>", line 2, in <module>  
    print('boujour')  
  
File "C:\Python27\lib\site-packages\IPython\kernel\zmq\iostream.py", line 202, in write  
    string = string.decode(self.encoding, 'replace')  
  
File "C:\Python27\lib\encodings\utf_8.py", line 16, in decode  
    return codecs.utf_8_decode(input, errors, True)  
  
KeyboardInterrupt
```

Exercices

16) Reprendre l'exercice « programme utile en mathématiques » (déjà repris dans la section « for ») qui s'intéresse à la fonction $(\sqrt{x} + 4)^2$ de telle sorte qu'il fasse plus que donner l'image d'un nombre unique (il demandera un x minimum, un x maximum et un pas, puis avec une boucle il va afficher des lignes de la forme « f(0)=16 », « f(1)=25 », etc) **Cette fois ci, on n'utilisera pas la fonction range.**

17) Créer un programme de devinette. Il fonctionnera de la manière suivante : le programme choisit aléatoirement un nombre entier entre 1 et 100. le programme demande alors de deviner la valeur du nombre. Tant que la proposition n'est pas bonne, le programme affichera « trop petit » ou « trop grand » selon le cas rencontré, puis reposera la question. Une fois que l'on aura obtenu la bonne réponse, le programme affichera : « bravo, tu as gagné en coups » (vous utiliserez un compteur, qui va augmenter de 1 à chaque fois que l'utilisateur fera une proposition, et qui permettra donc de compléter les de la phrase de félicitations.)

18) Reprendre l'exercice de tirage de numéros du loto, cette fois ci on n'aura pas le droit de tirer deux fois le même nombre.

V Les fonctions

Une fonction est un procédé dépendant ou non d'une variable, ou de plusieurs variables permettant de retourner un calcul ou autre

Un premier exemple

C'est l'exemple classique mathématique qui à toute valeur x on fait correspondre le réel $f(x) = 2x + 1$.

Evidemment , $f(2) = 2 \times 2 + 1 = 5$

```
In [9]: def f(x):  
...:     return 2*x+1  
...:  
  
In [10]: print(f(2))  
5
```

Un deuxième exemple

Ici la fonction permet de générer un nombre x de parties demandées par l'utilisateur . A l'issue de ces x parties la fonction retourne le nombre de fois où on a gagné

```
In [14]: def f(x):  
...:     L=[]  
...:     for k in range(x):  
...:         if randint(1,6)==6:  
...:             L=L+['gagné']  
...:         else:  
...:             L=L+['perdu']  
...:     c=L.count('gagné')  
...:     return c  
...:
```

```
In [15]: f(10)  
Out[15]: 1
```

```
In [16]: f(10)  
Out[16]: 1
```

```
In [17]: f(10)  
Out[17]: 1
```

```
In [18]: f(20)  
Out[18]: 4
```

```
In [19]: f(20)  
Out[19]: 7
```

```
In [20]: f(30)  
Out[20]: 4
```

Exercice

- 1) Proposer la fonction `lotoGen(x)` permettant de générer un nombre x de tirages du loto. (la sortie sera donc une liste contenant x listes de 5 éléments)
- 2) Créer la fonction `palin(mot)` qui détermine si un mot est un palindrome ou non, elle retournera alors la valeur `True` ou `False`
- 3) Créer la fonction `summ(min, max)` qui détermine la somme de tous les entiers compris entre les entiers `min` et `max`.
- 4) Créer la fonction `facto(nombre)` pour déterminer la factorielle d'un nombre. (pour information : la factorielle de 5 se note $5!$ et elle vaut $5 \times 4 \times 3 \times 2 \times 1 = 120$, et de manière générale $n! = n(n - 1)(n - 2) \dots 3 \times 2 \times 1$)