0.1 - numpy

April 11, 2017

```
In [1]: import numpy as np

In [2]: l = list(range(0, 60, 3))

In [3]: l

Out[3]: [0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57

In [4]: a = np.array(l)

In [5]: a

Out[5]: array([ 0,  3,  6,  9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48,
                51, 54, 57])

In [6]: %timeit -n 1 -r 1 sum(l)

1 loop, best of 1: 1.72 µs per loop

In [7]: %timeit -n 1 -r 1 a.sum()

1 loop, best of 1: 42.8 µs per loop

In [8]: l = list(range(0, 10**6, 3))

In [9]: a = np.array(l)

In [10]: l[:15]

Out[10]: [0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42]

In [11]: a[:15]

Out[11]: array([ 0,  3,  6,  9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42])

In [12]: %timeit -n 1 -r 1 sum(l)

1 loop, best of 1: 2.84 ms per loop

In [13]: %timeit -n 1 -r 1 a.sum()

1 loop, best of 1: 479 µs per loop
```

0.1 Operazioni elementwise

```
In [14]: %timeit [el*2 for el in l]

10 loops, best of 3: 19.1 ms per loop


In [15]: %timeit [el*2 for el in a]

10 loops, best of 3: 57 ms per loop


In [16]: %timeit [a[i] for i in range(len(a))]

10 loops, best of 3: 37.5 ms per loop


In [17]: %timeit 2*a

1000 loops, best of 3: 668 µs per loop


In [18]: from math import log

In [19]: log(a)

         ---------------------------------------------------------------------------

         TypeError                                 Traceback (most recent call last)

         <ipython-input-19-1e9003004902> in <module>()
     ----> 1 log(a)


         TypeError: only length-1 arrays can be converted to Python scalars


In [20]: np.log(a)

/home/pietro/.local/lib/python3.5/site-packages/ipykernel/__main__.py:1: RuntimeWar
  if __name__ == '__main__':


Out[20]: array([        -inf,   1.09861229,   1.79175947, ...,  13.81550356,
                13.81550656,  13.81550956])
```

0.2 Due dimensioni

```
In [21]: b = np.random.rand(5, 10)

In [22]: b

Out[22]: array([[ 0.82700212,  0.87674464,  0.5172631 ,  0.52667995,  0.45653736,
                  0.76321588,  0.68724396,  0.19321036,  0.97177778,  0.40453558],
                [ 0.50984837,  0.99411375,  0.03151988,  0.70797914,  0.89868737,
                  0.55849075,  0.8425051 ,  0.52323326,  0.77879962,  0.60651667],
                [ 0.9096573 ,  0.17862387,  0.73654496,  0.51932208,  0.57680968,
                  0.60098555,  0.04970194,  0.78800403,  0.24405052,  0.92455709],
                [ 0.61762449,  0.4045357 ,  0.6680333 ,  0.57566043,  0.57028665,
                  0.38079978,  0.39499248,  0.4383409 ,  0.97847985,  0.13480468],
                [ 0.1495647 ,  0.36092674,  0.72716829,  0.22474578,  0.88016085,
                  0.8450918 ,  0.61955163,  0.48775798,  0.12196898,  0.41122412]])

In [23]: b.dtype

Out[23]: dtype('float64')

In [24]: b[0]

Out[24]: array([ 0.82700212,  0.87674464,  0.5172631 ,  0.52667995,  0.45653736,
                 0.76321588,  0.68724396,  0.19321036,  0.97177778,  0.40453558])

In [25]: b[:,1]

Out[25]: array([ 0.87674464,  0.99411375,  0.17862387,  0.4045357 ,  0.36092674])

In [26]: b[2,4]

Out[26]: 0.57680968494003504

In [27]: b[3,3] = 'a'
         # Non perché numpy non supporti le stringhe, ma semplicemente perché quest


         ---------------------------------------------------------------------------

         ValueError                                Traceback (most recent call last)

         <ipython-input-27-57eaabcf4e52> in <module>()
     ----> 1 b[3,3] = 'a'
           2 # Non perché numpy non supporti le stringhe, ma semplicemente perché qu


         ValueError: could not convert string to float: 'a'


In [28]: b[3,4] = -1
```

```
In [29]: b[:2, 3:-5] = [[1, 2], [3,4]]

In [30]: b[:2, 3:-5]

Out[30]: array([[ 1.,  2.],
                [ 3.,  4.]])

In [31]: c = np.array([[[1,2], [3,4]],[[1,7], [6,4]]])

In [32]: cstr = c.astype(str)

In [33]: c[1,1] = 1500

In [34]: c

Out[34]: array([[[   1,    2],
                 [   3,    4]],

                [[   1,    7],
                 [1500, 1500]]])

In [35]: cstr

Out[35]: array([[['1', '2'],
                 ['3', '4']],

                [['1', '7'],
                 ['6', '4']]],
               dtype='<U21')

In [36]: d = np.array([[1,2], [3,4]])

In [37]: d

Out[37]: array([[1, 2],
                [3, 4]])

In [38]: d.transpose()

Out[38]: array([[1, 3],
                [2, 4]])

In [39]: d.reshape((1,4))

Out[39]: array([[1, 2, 3, 4]])

In [40]: objarr = np.array(['a', 1, None])

In [41]: objarr

Out[41]: array(['a', 1, None], dtype=object)

In [42]: np.array([1,2,3,4], dtype=object)

Out[42]: array([1, 2, 3, 4], dtype=object)
```

0.3 Digressione su slice e getitem

```
In [43]: a[:5]

Out[43]: array([ 0,  3,  6,  9, 12])

In [44]: a[slice(5)]

Out[44]: array([ 0,  3,  6,  9, 12])

In [45]: class Pippo:
             def __getitem__(self, value):
                 print("Mi è stato chiesto", value)
                 return "pippo"

In [46]: p = Pippo()

In [47]: idx = 5
         idx2 = 8

In [48]: a[idx]

Out[48]: 15

In [49]: a[:idx]

Out[49]: array([ 0,  3,  6,  9, 12])

In [50]: a[slice(idx)]

Out[50]: array([ 0,  3,  6,  9, 12])

In [51]: a[idx:idx2]

Out[51]: array([15, 18, 21])

In [52]: a[slice(idx, idx2, 2)]

Out[52]: array([15, 21])

In [53]: l2 = list(range(1000))

In [54]: l2[slice(75, 30, -6)]

Out[54]: [75, 69, 63, 57, 51, 45, 39, 33]

In [55]: l2[75:30:-6]

Out[55]: [75, 69, 63, 57, 51, 45, 39, 33]

In [56]: slice(None)
```

```
Out[56]: slice(None, None, None)

In [57]: l[::]

Out[57]: [0,
          3,
          6,
          9,
          12,
          15,
          18,
          21,
          24,
          27,
          30,
          33,
          36,
          39,
          42,
          45,
          48,
          51,
          54,
          57,
          60,
          63,
          66,
          69,
          72,
          75,
          78,
          81,
          84,
          87,
          90,
          93,
          96,
          99,
          102,
          105,
          108,
          111,
          114,
          117,
          120,
          123,
          126,
          129,
```

132,
135,
138,
141,
144,
147,
150,
153,
156,
159,
162,
165,
168,
171,
174,
177,
180,
183,
186,
189,
192,
195,
198,
201,
204,
207,
210,
213,
216,
219,
222,
225,
228,
231,
234,
237,
240,
243,
246,
249,
252,
255,
258,
261,
264,
267,
270,
273,

276,
279,
282,
285,
288,
291,
294,
297,
300,
303,
306,
309,
312,
315,
318,
321,
324,
327,
330,
333,
336,
339,
342,
345,
348,
351,
354,
357,
360,
363,
366,
369,
372,
375,
378,
381,
384,
387,
390,
393,
396,
399,
402,
405,
408,
411,
414,
417,

420,
423,
426,
429,
432,
435,
438,
441,
444,
447,
450,
453,
456,
459,
462,
465,
468,
471,
474,
477,
480,
483,
486,
489,
492,
495,
498,
501,
504,
507,
510,
513,
516,
519,
522,
525,
528,
531,
534,
537,
540,
543,
546,
549,
552,
555,
558,
561,

564,
567,
570,
573,
576,
579,
582,
585,
588,
591,
594,
597,
600,
603,
606,
609,
612,
615,
618,
621,
624,
627,
630,
633,
636,
639,
642,
645,
648,
651,
654,
657,
660,
663,
666,
669,
672,
675,
678,
681,
684,
687,
690,
693,
696,
699,
702,
705,

708,
711,
714,
717,
720,
723,
726,
729,
732,
735,
738,
741,
744,
747,
750,
753,
756,
759,
762,
765,
768,
771,
774,
777,
780,
783,
786,
789,
792,
795,
798,
801,
804,
807,
810,
813,
816,
819,
822,
825,
828,
831,
834,
837,
840,
843,
846,
849,

852,
855,
858,
861,
864,
867,
870,
873,
876,
879,
882,
885,
888,
891,
894,
897,
900,
903,
906,
909,
912,
915,
918,
921,
924,
927,
930,
933,
936,
939,
942,
945,
948,
951,
954,
957,
960,
963,
966,
969,
972,
975,
978,
981,
984,
987,
990,
993,

```
996,
999,
1002,
1005,
1008,
1011,
1014,
1017,
1020,
1023,
1026,
1029,
1032,
1035,
1038,
1041,
1044,
1047,
1050,
1053,
1056,
1059,
1062,
1065,
1068,
1071,
1074,
1077,
1080,
1083,
1086,
1089,
1092,
1095,
1098,
1101,
1104,
1107,
1110,
1113,
1116,
1119,
1122,
1125,
1128,
1131,
1134,
1137,
```

1140,
1143,
1146,
1149,
1152,
1155,
1158,
1161,
1164,
1167,
1170,
1173,
1176,
1179,
1182,
1185,
1188,
1191,
1194,
1197,
1200,
1203,
1206,
1209,
1212,
1215,
1218,
1221,
1224,
1227,
1230,
1233,
1236,
1239,
1242,
1245,
1248,
1251,
1254,
1257,
1260,
1263,
1266,
1269,
1272,
1275,
1278,
1281,

1284,
1287,
1290,
1293,
1296,
1299,
1302,
1305,
1308,
1311,
1314,
1317,
1320,
1323,
1326,
1329,
1332,
1335,
1338,
1341,
1344,
1347,
1350,
1353,
1356,
1359,
1362,
1365,
1368,
1371,
1374,
1377,
1380,
1383,
1386,
1389,
1392,
1395,
1398,
1401,
1404,
1407,
1410,
1413,
1416,
1419,
1422,
1425,

1428,
1431,
1434,
1437,
1440,
1443,
1446,
1449,
1452,
1455,
1458,
1461,
1464,
1467,
1470,
1473,
1476,
1479,
1482,
1485,
1488,
1491,
1494,
1497,
1500,
1503,
1506,
1509,
1512,
1515,
1518,
1521,
1524,
1527,
1530,
1533,
1536,
1539,
1542,
1545,
1548,
1551,
1554,
1557,
1560,
1563,
1566,
1569,

1572,
1575,
1578,
1581,
1584,
1587,
1590,
1593,
1596,
1599,
1602,
1605,
1608,
1611,
1614,
1617,
1620,
1623,
1626,
1629,
1632,
1635,
1638,
1641,
1644,
1647,
1650,
1653,
1656,
1659,
1662,
1665,
1668,
1671,
1674,
1677,
1680,
1683,
1686,
1689,
1692,
1695,
1698,
1701,
1704,
1707,
1710,
1713,

1716,
1719,
1722,
1725,
1728,
1731,
1734,
1737,
1740,
1743,
1746,
1749,
1752,
1755,
1758,
1761,
1764,
1767,
1770,
1773,
1776,
1779,
1782,
1785,
1788,
1791,
1794,
1797,
1800,
1803,
1806,
1809,
1812,
1815,
1818,
1821,
1824,
1827,
1830,
1833,
1836,
1839,
1842,
1845,
1848,
1851,
1854,
1857,

1860,
1863,
1866,
1869,
1872,
1875,
1878,
1881,
1884,
1887,
1890,
1893,
1896,
1899,
1902,
1905,
1908,
1911,
1914,
1917,
1920,
1923,
1926,
1929,
1932,
1935,
1938,
1941,
1944,
1947,
1950,
1953,
1956,
1959,
1962,
1965,
1968,
1971,
1974,
1977,
1980,
1983,
1986,
1989,
1992,
1995,
1998,
2001,

2004,
2007,
2010,
2013,
2016,
2019,
2022,
2025,
2028,
2031,
2034,
2037,
2040,
2043,
2046,
2049,
2052,
2055,
2058,
2061,
2064,
2067,
2070,
2073,
2076,
2079,
2082,
2085,
2088,
2091,
2094,
2097,
2100,
2103,
2106,
2109,
2112,
2115,
2118,
2121,
2124,
2127,
2130,
2133,
2136,
2139,
2142,
2145,

2148,
2151,
2154,
2157,
2160,
2163,
2166,
2169,
2172,
2175,
2178,
2181,
2184,
2187,
2190,
2193,
2196,
2199,
2202,
2205,
2208,
2211,
2214,
2217,
2220,
2223,
2226,
2229,
2232,
2235,
2238,
2241,
2244,
2247,
2250,
2253,
2256,
2259,
2262,
2265,
2268,
2271,
2274,
2277,
2280,
2283,
2286,
2289,

2292,
2295,
2298,
2301,
2304,
2307,
2310,
2313,
2316,
2319,
2322,
2325,
2328,
2331,
2334,
2337,
2340,
2343,
2346,
2349,
2352,
2355,
2358,
2361,
2364,
2367,
2370,
2373,
2376,
2379,
2382,
2385,
2388,
2391,
2394,
2397,
2400,
2403,
2406,
2409,
2412,
2415,
2418,
2421,
2424,
2427,
2430,
2433,

```
2436,
2439,
2442,
2445,
2448,
2451,
2454,
2457,
2460,
2463,
2466,
2469,
2472,
2475,
2478,
2481,
2484,
2487,
2490,
2493,
2496,
2499,
2502,
2505,
2508,
2511,
2514,
2517,
2520,
2523,
2526,
2529,
2532,
2535,
2538,
2541,
2544,
2547,
2550,
2553,
2556,
2559,
2562,
2565,
2568,
2571,
2574,
2577,
```

2580,
2583,
2586,
2589,
2592,
2595,
2598,
2601,
2604,
2607,
2610,
2613,
2616,
2619,
2622,
2625,
2628,
2631,
2634,
2637,
2640,
2643,
2646,
2649,
2652,
2655,
2658,
2661,
2664,
2667,
2670,
2673,
2676,
2679,
2682,
2685,
2688,
2691,
2694,
2697,
2700,
2703,
2706,
2709,
2712,
2715,
2718,
2721,

2724,
2727,
2730,
2733,
2736,
2739,
2742,
2745,
2748,
2751,
2754,
2757,
2760,
2763,
2766,
2769,
2772,
2775,
2778,
2781,
2784,
2787,
2790,
2793,
2796,
2799,
2802,
2805,
2808,
2811,
2814,
2817,
2820,
2823,
2826,
2829,
2832,
2835,
2838,
2841,
2844,
2847,
2850,
2853,
2856,
2859,
2862,
2865,

```
2868,
2871,
2874,
2877,
2880,
2883,
2886,
2889,
2892,
2895,
2898,
2901,
2904,
2907,
2910,
2913,
2916,
2919,
2922,
2925,
2928,
2931,
2934,
2937,
2940,
2943,
2946,
2949,
2952,
2955,
2958,
2961,
2964,
2967,
2970,
2973,
2976,
2979,
2982,
2985,
2988,
2991,
2994,
2997,
...]
```