

**week1-assignment.ipynb**

```
1  {
2    "cells": [
3      {
4        "cell_type": "markdown",
5        "metadata": {
6          "colab_type": "text",
7          "id": "mMcWEbFiLKvn"
8        },
9        "source": [
10         "# NumPy Exercises \n",
11         "\n",
12         "Now that we've learned about NumPy let's test your knowledge. We'll start off with a few
13         simple tasks, and then you'll be asked some more complicated questions."
14       ],
15       {
16         "cell_type": "markdown",
17         "metadata": {
18           "colab_type": "text",
19           "id": "DvBjXHM1LKvw"
20         },
21         "source": [
22           "#### Import NumPy as np"
23         ],
24       },
25       {
26         "cell_type": "code",
27         "execution_count": 1,
28         "metadata": {
29           "colab": {},
30           "colab_type": "code",
31           "id": "StInvTamLKv0"
32         },
33         "outputs": [],
34         "source": [
35           "import numpy as np"
36         ],
37       },
38       {
39         "cell_type": "markdown",
40         "metadata": {
41           "colab_type": "text",
42           "id": "qav3VCgyLKv5"
43         },
44         "source": [
45           "#### Create an array of 10 zeros "
46         ],
47       },
48       {
49         "cell_type": "code",
50         "execution_count": 2,
51         "metadata": {
```

```
52     "colab": {},
53     "colab_type": "code",
54     "id": "gLlF23CdLKv6",
55     "outputId": "824bc0c2-05e8-4a2f-b02c-803eb133cb1e"
56 },
57 "outputs": [
58   {
59     "data": {
60       "text/plain": [
61         "array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])"
62       ]
63     },
64     "execution_count": 2,
65     "metadata": {},
66     "output_type": "execute_result"
67   }
68 ],
69 "source": [
70   "np.zeros(10)"
71 ]
72 },
73 {
74   "cell_type": "markdown",
75   "metadata": {
76     "colab_type": "text",
77     "id": "-r7m8k4vLKv_"
78   },
79   "source": [
80     "#### Create an array of 10 ones"
81   ]
82 },
83 {
84   "cell_type": "code",
85   "execution_count": 3,
86   "metadata": {
87     "colab": {},
88     "colab_type": "code",
89     "id": "iwuCBhvcLKwB",
90     "outputId": "2358e05b-b357-4b0e-d4c4-feffc91889469"
91   },
92   "outputs": [
93     {
94       "data": {
95         "text/plain": [
96           "array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])"
97         ]
98       },
99       "execution_count": 3,
100       "metadata": {},
101       "output_type": "execute_result"
102     }
103   ],
104   "source": [
105     "np.ones(10)"
106   ]
107 },
```

```
108 {
109   "cell_type": "markdown",
110   "metadata": {
111     "colab_type": "text",
112     "id": "FIcddv6lLKwO"
113   },
114   "source": [
115     "#### Create an array of 10 fives"
116   ]
117 },
118 {
119   "cell_type": "code",
120   "execution_count": 6,
121   "metadata": {
122     "colab": {},
123     "colab_type": "code",
124     "id": "tXDJ5b8cLKwU",
125     "outputId": "f784a7d9-a26d-4d97-eeee-15e78e84b014"
126   },
127   "outputs": [
128     {
129       "data": {
130         "text/plain": [
131           "array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])"
132         ]
133       },
134       "execution_count": 6,
135       "metadata": {},
136       "output_type": "execute_result"
137     }
138   ],
139   "source": [
140     "np.ones(10)*5"
141   ]
142 },
143 {
144   "cell_type": "markdown",
145   "metadata": {
146     "colab_type": "text",
147     "id": "4UInvEwQLKwg"
148   },
149   "source": [
150     "#### Create an array of the integers from 10 to 50"
151   ]
152 },
153 {
154   "cell_type": "code",
155   "execution_count": 10,
156   "metadata": {
157     "colab": {},
158     "colab_type": "code",
159     "id": "Z9kAbHHiLKwj",
160     "outputId": "1b2d4a50-b3c4-44af-dd3b-643a6546d019"
161   },
162   "outputs": [
163     {
```

```
164     "name": "stdout",
165     "output_type": "stream",
166     "text": [
167         "[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33\n",
168         " 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50]\n"
169     ]
170 }
171 ],
172 "source": [
173     "a=np.arange(10,51)\n",
174     "print(a)"
175 ]
176 },
177 {
178     "cell_type": "markdown",
179     "metadata": {
180         "colab_type": "text",
181         "id": "3ZUIzdDeLKww"
182     },
183     "source": [
184         "#### Create an array of all the even integers from 10 to 50"
185     ]
186 },
187 {
188     "cell_type": "code",
189     "execution_count": 16,
190     "metadata": {
191         "colab": {},
192         "colab_type": "code",
193         "id": "dVVfz73BLKw5",
194         "outputId": "ee6de97d-f84f-4f4b-d6d6-c21c46f59666"
195     },
196     "outputs": [
197         {
198             "name": "stdout",
199             "output_type": "stream",
200             "text": [
201                 "[10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50]\n"
202             ]
203         }
204     ],
205     "source": [
206         "a1=np.arange(10,51,2)\n",
207         "print(a1)"
208     ]
209 },
210 {
211     "cell_type": "markdown",
212     "metadata": {
213         "colab_type": "text",
214         "id": "6e200i5-LKxB"
215     },
216     "source": [
217         "#### Create a 3x3 matrix with values ranging from 0 to 8"
218     ]
219 },
```

```
220 {
221   "cell_type": "code",
222   "execution_count": 34,
223   "metadata": {
224     "colab": {},
225     "colab_type": "code",
226     "id": "60DYRVCxLKxF",
227     "outputId": "0327b8fd-1f75-4da0-ddbc-9193a7cbbd3d"
228   },
229   "outputs": [
230     {
231       "data": {
232         "text/plain": [
233           "array([[0, 1, 2],\n",
234             "       [3, 4, 5],\n",
235             "       [6, 7, 8]])"
236         ]
237       },
238       "execution_count": 34,
239       "metadata": {},
240       "output_type": "execute_result"
241     }
242   ],
243   "source": [
244     "a2=np.arange(0,9).reshape(3,3)\n",
245     "a2"
246   ]
247 },
248 {
249   "cell_type": "markdown",
250   "metadata": {
251     "colab_type": "text",
252     "id": "54wb4NboLKxN"
253   },
254   "source": [
255     "#### Create a 3x3 identity matrix"
256   ]
257 },
258 {
259   "cell_type": "code",
260   "execution_count": 23,
261   "metadata": {
262     "colab": {},
263     "colab_type": "code",
264     "id": "j8JQg-VSLKxT",
265     "outputId": "697e3eed-3266-4f57-90ab-ff5d5c430c7d"
266   },
267   "outputs": [
268     {
269       "data": {
270         "text/plain": [
271           "array([[1., 0., 0.],\n",
272             "       [0., 1., 0.],\n",
273             "       [0., 0., 1.]])"
274         ]
275       },
```

```
276     "execution_count": 23,
277     "metadata": {},
278     "output_type": "execute_result"
279   }
280 ],
281 "source": [
282   "np.eye(3)"
283 ]
284 },
285 {
286   "cell_type": "markdown",
287   "metadata": {
288     "colab_type": "text",
289     "id": "_bpkEUARLKxa"
290   },
291   "source": [
292     "#### Use NumPy to generate a random number between 0 and 1"
293   ]
294 },
295 {
296   "cell_type": "code",
297   "execution_count": 27,
298   "metadata": {
299     "colab": {},
300     "colab_type": "code",
301     "id": "Co4I1qACLKxc",
302     "outputId": "efdd03f8-868e-4eeb-e40f-e75dba4cd14e"
303   },
304   "outputs": [
305     {
306       "data": {
307         "text/plain": [
308           "array([[0.68761205]])"
309         ]
310       },
311       "execution_count": 27,
312       "metadata": {},
313       "output_type": "execute_result"
314     }
315   ],
316   "source": [
317     "x=np.random.rand(1,1)\n",
318     "x"
319   ]
320 },
321 {
322   "cell_type": "markdown",
323   "metadata": {
324     "colab_type": "text",
325     "id": "7Qix8s61LKxp"
326   },
327   "source": [
328     "#### Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution"
329   ]
330 },
```

```

331 {
332   "cell_type": "code",
333   "execution_count": 28,
334   "metadata": {
335     "colab": {},
336     "colab_type": "code",
337     "id": "eUgQBgpQLKxs",
338     "outputId": "04664997-3ff0-421e-b92e-5e12c80321f9"
339   },
340   "outputs": [
341     {
342       "data": {
343         "text/plain": [
344           "array([ 0.40676241,  0.01919772, -0.53207531, -0.36309496,  0.11697544,\n",
345             "         0.88428258,  0.56316995, -0.11828199, -0.96252082, -1.63962687,\n",
346             "-1.1690078 ,  0.64177579, -1.27255107,  0.20594047, -0.32968782,\n",
347             "-0.77043214, -0.6391797 ,  0.14984774, -1.16161564,  0.09739328,\n",
348             "         3.08080732, -0.18831226,  0.99138594,  1.64733949,  0.5664771 ])"
349         ]
350       },
351       "execution_count": 28,
352       "metadata": {},
353       "output_type": "execute_result"
354     }
355   ],
356   "source": [
357     "x=np.random.normal(0,1,25)\n",
358     "x"
359   ]
360 },
361 {
362   "cell_type": "markdown",
363   "metadata": {
364     "colab_type": "text",
365     "id": "-sL9HYlWLKx1"
366   },
367   "source": [
368     "#### Create the following matrix:"
369   ]
370 },
371 {
372   "cell_type": "code",
373   "execution_count": 32,
374   "metadata": {
375     "colab": {},
376     "colab_type": "code",
377     "id": "p2pm5Mm6LKx5",
378     "outputId": "aa76f1ed-5967-4d3d-8061-72f82595c599"
379   },
380   "outputs": [
381     {
382       "name": "stdout",
383       "output_type": "stream",
384       "text": [
385         "[0.    0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13\n",
386         " 0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27\n",

```

```

387     " 0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41\n",
388     " 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55\n",
389     " 0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69\n",
390     " 0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83\n",
391     " 0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97\n",
392     " 0.98 0.99 1.  ]\n",
393     "[0.    0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13\n",
394     " 0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27\n",
395     " 0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41\n",
396     " 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55\n",
397     " 0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69\n",
398     " 0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83\n",
399     " 0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97\n",
400     " 0.98 0.99 1.  ]\n"
401 ]
402 }
403 ],
404 "source": [
405     "x=np.arange(0,1.01,0.01)\n",
406     "print(x)\n",
407     "y=np.linspace(0,1,101)\n",
408     "print(y)\n"
409 ]
410 },
411 {
412     "cell_type": "markdown",
413     "metadata": {
414         "colab_type": "text",
415         "id": "UQzj3r7NLKx_"
416     },
417     "source": [
418         "#### Create an array of 20 linearly spaced points between 0 and 1:"
419     ]
420 },
421 {
422     "cell_type": "code",
423     "execution_count": 33,
424     "metadata": {
425         "colab": {},
426         "colab_type": "code",
427         "id": "AniTWwz9LKyC",
428         "outputId": "dfb092f8-2f1e-446a-99e2-5c6df71631ac"
429     },
430     "outputs": [
431         {
432             "data": {
433                 "text/plain": [
434                     "array([0.          , 0.05263158, 0.10526316, 0.15789474, 0.21052632,\n435                    0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,\n436                    0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,\n437                    0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.          ])"
438                 ]
439             },
440             "execution_count": 33,
441             "metadata": {},
442             "output_type": "execute_result"

```



```
443     }
444   ],
445   "source": [
446     "np.linspace(0,1,20)"
447   ]
448 },
449 {
450   "cell_type": "markdown",
451   "metadata": {
452     "colab_type": "text",
453     "id": "i3vsM3rPLKyK"
454   },
455   "source": [
456     "## Numpy Indexing and Selection\n",
457     "\n",
458     "Now you will be given a few matrices, and be asked to replicate the resulting matrix\n",
459     "outputs:"
460   ],
461   {
462     "cell_type": "code",
463     "execution_count": null,
464     "metadata": {
465       "colab": {},
466       "colab_type": "code",
467       "id": "AM_s5FAvLKyN",
468       "outputId": "b29d6933-31ca-4d45-d4c9-123877d1e7be"
469     },
470     "outputs": [
471       {
472         "data": {
473           "text/plain": [
474             "array([[ 1,  2,  3,  4,  5],\n",
475             "       [ 6,  7,  8,  9, 10],\n",
476             "       [11, 12, 13, 14, 15],\n",
477             "       [16, 17, 18, 19, 20],\n",
478             "       [21, 22, 23, 24, 25]])"
479           ]
480         },
481         "execution_count": 38,
482         "metadata": {
483           "tags": []
484         },
485         "output_type": "execute_result"
486       }
487     ],
488     "source": [
489       "mat = np.arange(1,26).reshape(5,5)\n",
490       "mat"
491     ]
492   },
493   {
494     "cell_type": "code",
495     "execution_count": null,
496     "metadata": {
497       "colab": {},
```

```
498     "colab_type": "code",
499     "id": "rx1UOQ4dLKyX"
500 },
501 "outputs": [],
502 "source": [
503     "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
504     "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
505     "# BE ABLE TO SEE THE OUTPUT ANY MORE"
506 ]
507 },
508 {
509     "cell_type": "code",
510     "execution_count": null,
511     "metadata": {
512         "colab": {},
513         "colab_type": "code",
514         "id": "2BJ3cyl0LKyf",
515         "outputId": "ab55bbee-37f2-4797-bad5-4ed340f4d684"
516     },
517     "outputs": [
518         {
519             "data": {
520                 "text/plain": [
521                     "array([[12, 13, 14, 15],\n",
522                        "       [17, 18, 19, 20],\n",
523                        "       [22, 23, 24, 25]])"
524                 ]
525             },
526             "execution_count": 40,
527             "metadata": {
528                 "tags": []
529             },
530             "output_type": "execute_result"
531         }
532     ],
533     "source": [
534         "mat[2:5,1:5]"
535     ]
536 },
537 {
538     "cell_type": "code",
539     "execution_count": null,
540     "metadata": {
541         "colab": {},
542         "colab_type": "code",
543         "id": "9-U53vNoLKyw"
544     },
545     "outputs": [],
546     "source": [
547         "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
548         "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
549         "# BE ABLE TO SEE THE OUTPUT ANY MORE"
550     ]
551 },
552 {
553     "cell_type": "code",
```

```
554     "execution_count": null,
555     "metadata": {
556         "colab": {},
557         "colab_type": "code",
558         "id": "QJyclCVfLKy6",
559         "outputId": "9762e12a-13fe-4567-fd7a-718402be1edb"
560     },
561     "outputs": [
562         {
563             "data": {
564                 "text/plain": [
565                     "20"
566                 ]
567             },
568             "execution_count": 41,
569             "metadata": {
570                 "tags": []
571             },
572             "output_type": "execute_result"
573         }
574     ],
575     "source": [
576         "mat[3,4]"
577     ]
578 },
579 {
580     "cell_type": "code",
581     "execution_count": null,
582     "metadata": {
583         "colab": {},
584         "colab_type": "code",
585         "id": "E_PuvjL5LKzH"
586     },
587     "outputs": [],
588     "source": [
589         "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
590         "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
591         "# BE ABLE TO SEE THE OUTPUT ANY MORE"
592     ]
593 },
594 {
595     "cell_type": "code",
596     "execution_count": null,
597     "metadata": {
598         "colab": {},
599         "colab_type": "code",
600         "id": "TzFTd7eHLKzY",
601         "outputId": "5aa2b71d-7d91-4bbd-8550-46aa7e62385d"
602     },
603     "outputs": [
604         {
605             "data": {
606                 "text/plain": [
607                     "array([[ 2],\n",
608                     "        [ 7],\n",
609                     "        [12]])"
```

```
610     ]
611   },
612   "execution_count": 42,
613   "metadata": {
614     "tags": []
615   },
616   "output_type": "execute_result"
617 }
618 ],
619 "source": [
620   "mat[0:3,1:2]"
621 ]
622 },
623 {
624   "cell_type": "code",
625   "execution_count": null,
626   "metadata": {
627     "colab": {},
628     "colab_type": "code",
629     "id": "BaybacXxLKze"
630   },
631   "outputs": [],
632   "source": [
633     "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
634     "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
635     "# BE ABLE TO SEE THE OUTPUT ANY MORE"
636   ]
637 },
638 {
639   "cell_type": "code",
640   "execution_count": null,
641   "metadata": {
642     "colab": {},
643     "colab_type": "code",
644     "id": "hvb1pMEFLKz1",
645     "outputId": "95ecd0ff-f24a-4e5f-edae-481966a15a0a"
646   },
647   "outputs": [
648     {
649       "data": {
650         "text/plain": [
651           "array([21, 22, 23, 24, 25])"
652         ]
653       },
654       "execution_count": 46,
655       "metadata": {
656         "tags": []
657       },
658       "output_type": "execute_result"
659     }
660   ],
661   "source": [
662     "mat[4,:]"
663   ]
664 },
665 {
```

```
666     "cell_type": "code",
667     "execution_count": null,
668     "metadata": {
669         "colab": {},
670         "colab_type": "code",
671         "id": "L0SoIZfILKzr"
672     },
673     "outputs": [],
674     "source": [
675         "# WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW\n",
676         "# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T\n",
677         "# BE ABLE TO SEE THE OUTPUT ANY MORE"
678     ]
679 },
680 {
681     "cell_type": "code",
682     "execution_count": null,
683     "metadata": {
684         "colab": {},
685         "colab_type": "code",
686         "id": "h2NmY81aLKzv",
687         "outputId": "ec3006aa-5bd4-43ef-b876-f0475c9a26ce"
688     },
689     "outputs": [
690         {
691             "data": {
692                 "text/plain": [
693                     "array([[16, 17, 18, 19, 20],\n",
694                        "       [21, 22, 23, 24, 25]])"
695                 ]
696             },
697             "execution_count": 49,
698             "metadata": {
699                 "tags": []
700             },
701             "output_type": "execute_result"
702         }
703     ],
704     "source": [
705         "mat[3:5,:]"
706     ]
707 },
708 {
709     "cell_type": "markdown",
710     "metadata": {
711         "colab_type": "text",
712         "id": "40ijP69QLKz6"
713     },
714     "source": [
715         "### Now do the following"
716     ]
717 },
718 {
719     "cell_type": "markdown",
720     "metadata": {
721         "colab_type": "text",
```

```
722     "id": "hJYmPKZ6LKz7"
723   },
724   "source": [
725     "#### Get the sum of all the values in mat"
726   ]
727 },
728 {
729   "cell_type": "code",
730   "execution_count": null,
731   "metadata": {
732     "colab": {},
733     "colab_type": "code",
734     "id": "lgjWcsNHLKz9",
735     "outputId": "a4aebb34-800e-490a-94fa-b2f29e05ec8a"
736   },
737   "outputs": [
738     {
739       "data": {
740         "text/plain": [
741           "325"
742         ]
743       },
744       "execution_count": 50,
745       "metadata": {
746         "tags": []
747       },
748       "output_type": "execute_result"
749     }
750   ],
751   "source": [
752     "mat.sum()"
753   ]
754 },
755 {
756   "cell_type": "markdown",
757   "metadata": {
758     "colab_type": "text",
759     "id": "a0xAVLe8LK0B"
760   },
761   "source": [
762     "#### Get the standard deviation of the values in mat"
763   ]
764 },
765 {
766   "cell_type": "code",
767   "execution_count": null,
768   "metadata": {
769     "colab": {},
770     "colab_type": "code",
771     "id": "1bExi0tvLK0D",
772     "outputId": "cdcf114d-bcd3-494a-afc8-f4b592566334"
773   },
774   "outputs": [
775     {
776       "data": {
777         "text/plain": [
```

```
778         "7.2111025509279782"
779     ],
780 },
781     "execution_count": 51,
782     "metadata": {
783         "tags": []
784     },
785     "output_type": "execute_result"
786 }
787 ],
788 "source": [
789     "mat.std()"
790 ]
791 },
792 {
793     "cell_type": "markdown",
794     "metadata": {
795         "colab_type": "text",
796         "id": "qWvkrLQWLK0H"
797     },
798     "source": [
799         "#### Get the sum of all the columns in mat"
800     ]
801 },
802 {
803     "cell_type": "code",
804     "execution_count": null,
805     "metadata": {
806         "colab": {},
807         "colab_type": "code",
808         "id": "-XFw4SD1LK0J",
809         "outputId": "03cee652-bdb1-411c-a81f-7534525a3d21"
810     },
811     "outputs": [
812         {
813             "data": {
814                 "text/plain": [
815                     "array([55, 60, 65, 70, 75])"
816                 ]
817             },
818             "execution_count": 53,
819             "metadata": {
820                 "tags": []
821             },
822             "output_type": "execute_result"
823         }
824     ],
825     "source": [
826         "sum(mat)"
827     ]
828 },
829 {
830     "cell_type": "markdown",
831     "metadata": {
832         "colab_type": "text",
833         "collapsed": true,
```

```
834     "id": "x75g5hG1LK0N"
835   },
836   "source": []
837 }
838 ],
839 "metadata": {
840   "colab": {
841     "name": "Numpy Exercise .ipynb",
842     "provenance": []
843   },
844   "kernelspec": {
845     "display_name": "Python 3",
846     "language": "python",
847     "name": "python3"
848   },
849   "language_info": {
850     "codemirror_mode": {
851       "name": "ipython",
852       "version": 3
853     },
854     "file_extension": ".py",
855     "mimetype": "text/x-python",
856     "name": "python",
857     "nbconvert_exporter": "python",
858     "pygments_lexer": "ipython3",
859     "version": "3.8.0"
860   }
861 },
862 "nbformat": 4,
863 "nbformat_minor": 1
864 }
865
```