

lec04

September 6, 2021

```
[1]: from datascience import *  
import numpy as np  
  
%matplotlib inline  
import matplotlib.pyplot as plots  
plots.style.use('fivethirtyeight')
```

0.1 Numbers

```
[2]: 30
```

```
[2]: 30
```

```
[3]: 10 * 3      # int
```

```
[3]: 30
```

```
[4]: 10 / 3      # float
```

```
[4]: 3.3333333333333335
```

```
[5]: 10 / 2
```

```
[5]: 5.0
```

```
[6]: 10 ** 3
```

```
[6]: 1000
```

```
[7]: 10 ** 0.5
```

```
[7]: 3.1622776601683795
```

```
[8]: 1234567 ** 890
```

```
[8]: 28057633421073692307938098083108857240301516986438987883806628169801138978537695  
05682553341995361840866283163839674938063847465098181599413516030923830315020297
```

42911448484247280603760163318317739234241955228144609888233311336049893606068941
79810605801695817815170810008874669869201967873657354729766373847906605276561835
29108057677628812897898917415727467403873251697100284318113613171090832770107627
37342929757810045254133113797239439551887974825249762995316918974254668874383801
48429987972224885784827422555902788303262644158870424243493383366855861080147968
43647297854686632059611017792593114133643766176826645471505135284339608286440790
10860511468726144621410670323830462674743032028317050562359359819248218325332322
74733612174594975935145761031393542639102351175529448538352593282406358910892032
47501395301794244132584186874755797831852390704417457044048557568618633160284149
58334035268651200961197578423985926433687839238748279104915691285415971551621470
61662144662300373398237804927743372746758002895806963158194944180047334128552051
55027918058344891850668473780787520539851110601937177777258527635476258052342827
17976518638183497352230313682985283424267955988837199054654777373514287261138310
27427533112349589296021790176094534507880215768899680905585269460028670671554339
37025164603278496325064732080655085967120552122358643425906339419468088785812576
64117039978155519851854222875576409081251478816375887722803147462624998232017323
05137907662110518070715090183942995944962985534250328886395752333084279229620726
91110016410832613254845558533479779137394546533348267260208654147664661662489837
72216646553598447030013019536152672465799924773766411954516636797011896053965370
07282467808412434228659278220297454306877793414948355110483310917084354358106326
77284968867671217350609932555531424276428454845446681148799804144451127195396670
37373315550368844076637611730814523133920451451811418244266972692695846026350474
78231713577595857706653699946779446532469277172620877216169753908252175074082205
10861018634338133464060418295047933927635868626199925788463191380080935701443847
34653274683834944983320418371952042527350846142736326881090181643920752060245579
20694063678216588391212776492275556232735646773013314043396033683541872791705429
54088017099056552277230318062383216542056927422226999811205450549485589246237819
87486963891730354164471435608160246861651214402780862166991796752329884773328099
81888491266604101445168407511211771222187273129172359053373330605565166119243336
61893530040067042505379294125436401184352256775565463142705637306783871882313794
72883142752958315084614942700480232208778835796899226443869299300980988387546551
86461724704958394104925990604557163676374403646301314717501067091324658953716704
12945256073822697903510522211765697295922053593229276419226531331408317896936830
65581587366635519730537870896466820989929134470544427633357871932390417095207432
86438114045442568430257094254743878149739605420186301282011612723290247920117407
45530212579878161378961127918038096648026162029808878747866210894432543002029434
56561313235504377738576383123806095236344979787176162254177745365378408145671994
04197667511928678042692685056365748321996895389867152507426302005234903390940854
25178568698903268320220973567490288420919261018482168538171910000711992076873681
12930759867528550334596016101639183072826032669342820217716011954373680034429777
14810796270769880319759785758956476542215586200929918846870918626136057221442714
90738163190650508558761186844994717152403269557878581184075275936408939256017646
71555930626019625901017042703736107326757314702651062869562112813217251546407641
87429401031139646116816994079509677744134281843552490413347147658274806758998002
35537773385443229622067233523354213261412403375030932576886654499317808172035910
62343948892349772999003413295599161617531530111000313665555246469642827977836547
74061643075250111197830550476648647988184263076041723810736957986883778891934154

```
23645042188927398145864445871680431617772823584658707705057426132797633601281478
88754287170888477613299349811362979107521226913126313510421190192847830966740064
24771452013624333656585787072670181223889068531032647978375503149068263279645898
11537800151348078278689526492266541600168152334174087551992509222975658050902151
57384952244507458873110903655478968028034572004600912050792815263201807869771012
02183681475705991410316733006823483261564972440115748178021870998348803273462106
24556369660018678859929271654108596951193479453034452114779847850504280525490303
49273073184870449732163957954535905140700262906991240051847070751230847749498900
63902493512735614539577671037491050994868594418234991201033599745620953618782199
93785312682108204349115081403854873968693651144816992008787094413654541840750267
94542894065514710210377687792449535458967447120366380097370224026967778461034601
04418242618177635925998705465949871288020208898786236948002475752516614392761570
08847529529237192162577463356503617323132760079688627706444510425779441871297035
02639676885112950172245422031641877707491283823839116265868337163495941554418138
40581542519607672171016321392583121509047776231567889373097135405693067652321656
71618935868039816292110221657560493742908804867952618200515005539316960724088551
53913362630252255734541324938465849237169880509328900471734053281019669288091881
68628324059419208283025684619139040887021476547138780562181170118390947699181687
41678648022940613796829566893389100562532347981860917540264049
```

```
[9]: 10 / 3
```

```
[9]: 3.3333333333333335
```

```
[10]: 75892745.215489247589274985712
```

```
[10]: 75892745.21548925
```

```
[11]: 75892745.215489247589274985712 - 75892745.21548925
```

```
[11]: 0.0
```

```
[12]: (13 ** 0.5) ** 2
```

```
[12]: 12.999999999999998
```

```
[13]: int(10 / 5)
```

```
[13]: 2
```

```
[14]: int(10 / 4)
```

```
[14]: 2
```

```
[15]: float(3)
```

```
[15]: 3.0
```

6 / 4000

0.0015

[illegible]

1.5e-56

`4000000000000000000000000000000000000000 * 1.5e-56`

1.5e-56

1.5e-56

```
File "<ipython-input-22-6d17da6697ca>", line 1
    2x
    ~
SyntaxError: invalid syntax
```

```
round(3.7)
```

```
round(2.00000052345324, 2)
```

10 * 3.0

30.0

0.2 Strings

```
[27]: 'Flavor'
```

```
[27]: 'Flavor'
```

```
[28]: Flavor
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-28-c281a38ccebfb> in <module>  
----> 1 Flavor  
  
NameError: name 'Flavor' is not defined
```

```
[29]: "Flavor"
```

```
[29]: 'Flavor'
```

```
[30]: "Don't always use single quotes"
```

```
[30]: "Don't always use single quotes"
```

```
[31]: 'Don't always use single quotes'
```

```
File "<ipython-input-31-8490ca019922>", line 1  
    'Don't always use single quotes'  
    ~  
SyntaxError: invalid syntax
```

```
[32]: 'straw' + 'berry' # concatenation
```

```
[32]: 'strawberry'
```

```
[33]: 'Chirs' + 'Paul' # spaces aren't added for you
```

```
[33]: 'ChirsPaul'
```

```
[34]: 'Chris' + ' ' + 'Paul'
```

```
[34]: 'Chris Paul'
```

```
[35]: 'ha' * 100
```

```
[36]: 'lo' * 5.5
```

```
[37]: 'ha' + 10
```

```
[38]: int('3')
```

```
[39]: int('3.0')
```

```
[40]: float('3.0')
```

```
[41]: str(3)
```

```
[42]: str(4.5)
```

```
[42]: '4.5'
```

0.3 Types

```
[43]: type(10)
```

```
[43]: int
```

```
[44]: a = 10
```

```
[45]: type(a)
```

```
[45]: int
```

```
[46]: type(4.5)
```

```
[46]: float
```

```
[47]: type('abc')
```

```
[47]: str
```

```
[48]: type(nba)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-48-c7f8962b3941> in <module>  
----> 1 type(nba)  
  
NameError: name 'nba' is not defined
```

```
[49]: type(True)
```

```
[49]: bool
```

```
[50]: type(abs)
```

```
[50]: builtin_function_or_method
```

0.4 Arrays

```
[51]: my_array = make_array(1, 2, 3, 4)
```

```
[52]: my_array
```

```
[52]: array([1, 2, 3, 4], dtype=int64)
```

```
[53]: my_array * 2
```

```
[53]: array([2, 4, 6, 8], dtype=int64)
```

```
[54]: my_array ** 2
```

```
[54]: array([ 1,  4,  9, 16], dtype=int64)
```

```
[55]: my_array + 1
```

```
[55]: array([2, 3, 4, 5], dtype=int64)
```

```
[56]: my_array # array is unchanged, just like when we call show/select/drop on Table
```

```
[56]: array([1, 2, 3, 4], dtype=int64)
```

```
[57]: another = make_array(5, 6, 7, 8)
```

```
[58]: my_array + another
```

```
[58]: array([ 6,  8, 10, 12], dtype=int64)
```

```
[59]: yet_another = make_array(5, 6, 7)
```

```
[60]: my_array + yet_another
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-60-a4a5e45ad569> in <module>  
----> 1 my_array + yet_another  
  
ValueError: operands could not be broadcast together with shapes (4,) (3,)
```

```
[61]: str_array = make_array('ha', 'he', 'ho')
```

```
[62]: str_array * 4
```

```
-----  
UFuncTypeError                            Traceback (most recent call last)  
<ipython-input-62-1f62457f1a5c> in <module>  
----> 1 str_array * 4  
  
UFuncTypeError: ufunc 'multiply' did not contain a loop with signature matching  
→ types (dtype('<U3'), dtype('<U3')) -> dtype('<U3')
```



```
[63]: sum(my_array)
```

```
[63]: 10
```

```
[64]: np.average(my_array)
```

```
[64]: 2.5
```

```
[65]: len(my_array)
```

```
[65]: 4
```

```
[66]: sum(my_array) / len(my_array)
```

```
[66]: 2.5
```

```
[67]: np.average(my_array)
```

```
[67]: 2.5
```

```
[68]: another = make_array(70, 60, 90, 80)
```

```
[69]: my_array + another
```

```
[69]: array([71, 62, 93, 84], dtype=int64)
```

```
[70]: yet_another = make_array(5, 6, 7)
```

```
[71]: my_array + yet_another
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-71-a4a5e45ad569> in <module>
----> 1 my_array + yet_another

ValueError: operands could not be broadcast together with shapes (4,) (3,)
```

```
[72]: tunas = make_array('bluefin', 'albacore', 'jim')
tunas
```

```
[72]: array(['bluefin', 'albacore', 'jim'], dtype='<U8')
```

```
[73]: tunas * 4
```

```
-----
UFuncTypeError                            Traceback (most recent call last)
```

```
<ipython-input-73-c34521c521fa> in <module>
----> 1 tunas * 4
```

```
UFuncTypeError: ufunc 'multiply' did not contain a loop with signature matching
↳ types (dtype('<U8'), dtype('<U8')) -> dtype('<U8')
```

```
[74]: tunas.item(0) # NOTE: indexing starts at 0!
```

```
[74]: 'bluefin'
```

```
[75]: tunas.item(2)
```

```
[75]: 'jim'
```

```
[76]: tunas.item(3)
```

```
-----
IndexError                                Traceback (most recent call last)
<ipython-input-76-5031eceaa018> in <module>
----> 1 tunas.item(3)

IndexError: index 3 is out of bounds for axis 0 with size 3
```

0.5 Columns of Tables are Arrays

```
[77]: nba = Table.read_table('nba_salaries.csv').relabelled(3, 'SALARY')
      warriors = nba.where('TEAM', 'Golden State Warriors')
```

```
[78]: warriors
```

```
[78]: PLAYER          | POSITION | TEAM          | SALARY
      Klay Thompson   | SG      | Golden State Warriors | 15.501
      Draymond Green  | PF      | Golden State Warriors | 14.2609
      Andrew Bogut    | C       | Golden State Warriors | 13.8
      Andre Iguodala  | SF      | Golden State Warriors | 11.7105
      Stephen Curry   | PG      | Golden State Warriors | 11.3708
      Jason Thompson  | PF      | Golden State Warriors | 7.00847
      Shaun Livingston | PG      | Golden State Warriors | 5.54373
      Harrison Barnes | SF      | Golden State Warriors | 3.8734
      Marreese Speights | C       | Golden State Warriors | 3.815
      Leandro Barbosa | SG      | Golden State Warriors | 2.5
      ... (4 rows omitted)
```

```
[79]: warriors.select('SALARY')
```

```
[79]: SALARY
      15.501
      14.2609
      13.8
      11.7105
      11.3708
      7.00847
      5.54373
      3.8734
      3.815
      2.5
      ... (4 rows omitted)
```

```
[80]: np.average(warriors.column('SALARY'))
```

```
[80]: 6.72036692857143
```

```
[81]: warriors.column('SALARY')
```

```
[81]: array([15.501   , 14.26087 , 13.8     , 11.710456, 11.370786,  7.008475,
           5.543725,  3.873398,  3.815    ,  2.5     ,  2.008748,  1.270964,
           1.13196 ,  0.289755])
```

```
[82]: raptors = nba.where('TEAM', 'Toronto Raptors')
```

```
[83]: np.average(warriors.column('SALARY')) - np.average(raptors.column('SALARY'))
```

```
[83]: 2.3278598697479005
```

0.6 Ranges

```
[84]: make_array(0, 1, 2, 3, 4, 5, 6)
```

```
[84]: array([0, 1, 2, 3, 4, 5, 6], dtype=int64)
```

```
[85]: np.arange(7)
```

```
[85]: array([0, 1, 2, 3, 4, 5, 6])
```

```
[86]: np.arange(5, 11)
```

```
[86]: array([ 5,  6,  7,  8,  9, 10])
```

```
[87]: np.arange(0, 20, 2)
```

```
[87]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

```
[88]: np.arange(0, 21, 2)
```

```
[88]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

```
[89]: np.arange(0, 1, 0.1)
```

```
[89]: array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9])
```

```
[ ]:
```