

# Live code examples from lecture

These map to the lecture slides and should be read in conjunction.

## string interpolation

```
In [1]: "%s my interpolated string" % "things"
```

```
Out[1]: 'things my interpolated string'
```

```
In [2]: val_to_insert = "Jon"  
        "Hi I am %s" % val_to_insert
```

```
Out[2]: 'Hi I am Jon'
```

## Dynamic typing

```
In [6]: 1.+(3)
```

```
Out[6]: 4.0
```

```
In [7]: "hello " +("world")
```

```
Out[7]: 'hello world'
```

```
In [8]: first_arg = 1  
        first_arg + 4
```

```
Out[8]: 5
```

```
In [9]: first_arg = "hello"  
        first_arg + "world"
```

```
Out[9]: 'helloworld'
```

```
In [10]: imaginary_num = 4 + 7j  
         imaginary_num.real
```

```
Out[10]: 4.0
```

```
In [11]: imaginary_num  
         imaginary_num.imag
```

```
Out[11]: 7.0
```

## Lists and "list-like" data types

```
In [12]: my_list = [1,2,3]  
         my_list
```

```
Out[12]: [1, 2, 3]
```

```
In [13]: my_list = [1,2,"hello"]  
         my_list
```

```
Out[13]: [1, 2, 'hello']
```

```
In [14]: my_list.append("world")  
         my_list
```

```
Out[14]: [1, 2, 'hello', 'world']
```

```
In [15]: print my_list[1]  
         print my_list[0]  
         print my_list[3]  
         print my_list[-1]
```

```
2  
1  
world  
world
```

```
In [16]: a,b,c = (1,2,3)  
         print a  
         print b  
         print c
```

```
1  
2  
3
```

```
In [17]: tup = (1,2,3)  
         a = tup[0]  
         a
```

```
Out[17]: 1
```

```
In [19]: print my_list[0]  
my_list
```

```
1
```

```
Out[19]: [1, 2, 'hello', 'world']
```

```
In [20]: my_reference_num = (1,2,3,4,5)  
my_reference_num
```

```
Out[20]: (1, 2, 3, 4, 5)
```

```
In [21]: my_string = "jon"  
my_string[0]
```

```
Out[21]: 'j'
```

```
In [22]: my_char_tuple = ('j','o','n')  
my_char_tuple[0]
```

```
Out[22]: 'j'
```

```
In [23]: print sorted(my_string)  
print sorted(my_list)  
print sorted(my_char_tuple)
```

```
['j', 'n', 'o']  
[1, 2, 'hello', 'world']  
['j', 'n', 'o']
```

```
In [24]: # unicode  
u'sdfsd'
```

```
Out[24]: u'sdfsd'
```

## Control Flow

```
In [25]: for c in my_string:  
        print c
```

```
j  
o  
n
```

```
In [27]: x = 0
        while x <= 2:
            print "Hello"
            x += 1
```

Hello

Hello

Hello

```
In [28]: for word in ['jon','mary','sue','frank']:
        print "Hello I am " + word
```

Hello I am jon

Hello I am mary

Hello I am sue

Hello I am frank

```
In [29]: range(3, 13)
```

```
Out[29]: [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

## Exercise

```
In [ ]: for number in range(12, 5000):
        if (number % 7 == 0) and (number % 5 != 0):
            print number
```

```
In [35]: results = []
```

```
In [36]: for number in range(12, 5000):
        if (number % 7 == 0) and (number % 5 != 0):
            results.append(number)
```

```
In [38]: # print first 10 elements (we will cover this next lesson)
        results[0:10]
```

```
Out[38]: [14, 21, 28, 42, 49, 56, 63, 77, 84, 91]
```

```
In [39]: type(results)
        ','.join(["hello", "class", "im","jon"])
```

```
Out[39]: 'hello,class,im,jon'
```

```
In [40]: results = []
```

```
for number in range(12, 5000):  
    if (number % 7 == 0) and (number % 5 != 0):  
        results.append(str(number))
```

```
In [41]: ','.join(results)
```

```
Out[41]: '14,21,28,42,49,56,63,77,84,91,98,112,119,126,133,147,154,161,168,182,189,196,203,217,224,231,238,252,259,266,273,287,294,301,308,322,329,336,343,357,364,371,378,392,399,406,413,427,434,441,448,462,469,476,483,497,504,511,518,532,539,546,553,567,574,581,588,602,609,616,623,637,644,651,658,672,679,686,693,707,714,721,728,742,749,756,763,777,784,791,798,812,819,826,833,847,854,861,868,882,889,896,903,917,924,931,938,952,959,966,973,987,994,1001,1008,1022,1029,1036,1043,1057,1064,1071,1078,1092,1099,1106,1113,1127,1134,1141,1148,1162,1169,1176,1183,1197,1204,1211,1218,1232,1239,1246,1253,1267,1274,1281,1288,1302,1309,1316,1323,1337,1344,1351,1358,1372,1379,1386,1393,1407,1414,1421,1428,1442,1449,1456,1463,1477,1484,1491,1498,1512,1519,1526,1533,1547,1554,1561,1568,1582,1589,1596,1603,1617,1624,1631,1638,1652,1659,1666,1673,1687,1694,1701,1708,1722,1729,1736,1743,1757,1764,1771,1778,1792,1799,1806,1813,1827,1834,1841,1848,1862,1869,1876,1883,1897,1904,1911,1918,1932,1939,1946,1953,1967,1974,1981,1988,2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,2142,2149,2156,2163,2177,2184,2191,2198,2212,2219,2226,2233,2247,2254,2261,2268,2282,2289,2296,2303,2317,2324,2331,2338,2352,2359,2366,2373,2387,2394,2401,2408,2422,2429,2436,2443,2457,2464,2471,2478,2492,2499,2506,2513,2527,2534,2541,2548,2562,2569,2576,2583,2597,2604,2611,2618,2632,2639,2646,2653,2667,2674,2681,2688,2702,2709,2716,2723,2737,2744,2751,2758,2772,2779,2786,2793,2807,2814,2821,2828,2842,2849,2856,2863,2877,2884,2891,2898,2912,2919,2926,2933,2947,2954,2961,2968,2982,2989,2996,3003,3017,3024,3031,3038,3052,3059,3066,3073,3087,3094,3101,3108,3122,3129,3136,3143,3157,3164,3171,3178,3192,3199,3206,3213,3227,3234,3241,3248,3262,3269,3276,3283,3297,3304,3311,3318,3332,3339,3346,3353,3367,3374,3381,3388,3402,3409,3416,3423,3437,3444,3451,3458,3472,3479,3486,3493,3507,3514,3521,3528,3542,3549,3556,3563,3577,3584,3591,3598,3612,3619,3626,3633,3647,3654,3661,3668,3682,3689,3696,3703,3717,3724,3731,3738,3752,3759,3766,3773,3787,3794,3801,3808,3822,3829,3836,3843,3857,3864,3871,3878,3892,3899,3906,3913,3927,3934,3941,3948,3962,3969,3976,3983,3997,4004,4011,4018,4032,4039,4046,4053,4067,4074,4081,4088,4102,4109,4116,4123,4137,4144,4151,4158,4172,4179,4186,4193,4207,4214,4221,4228,4242,4249,4256,4263,4277,4284,4291,4298,4312,4319,4326,4333,4347,4354,4361,4368,4382,4389,4396,4403,4417,4424,4431,4438,4452,4459,4466,4473,4487,4494,4501,4508,4522,4529,4536,4543,4557,4564,4571,4578,4592,4599,4606,4613,4627,4634,4641,4648,4662,4669,4676,4683,4697,4704,4711,4718,4732,4739,4746,4753,4767,4774,4781,4788,4802,4809,4816,4823,4837,4844,4851,4858,4872,4879,4886,4893,4907,4914,4921,4928,4942,4949,4956,4963,4977,4984,4991,4998'
```

## File objects

```
In [44]: with open('/Users/jonathandinu/Repositories/workshopper/data-engineering-101/text/article0.txt', 'r') as f:
          print type(f)
          print f.readlines()
```

```
<type 'file'>
```

```

[comp.sys.ibm.pc.hardware\new, ': >point of view, why does SCSI ha
ve an advantage when it comes to multi-\new, ': >>tasking? Data is
data, and it could be anywhere on the drive. Can\new, ': >>SCSI fin
d it faster? can it get it off the drive and into the computer\new,
': >>faster? Does it have a better cache system? I thought SCSI w
as good at\new, ': >>managing a data bus when multiple devices are a
ttached. If we are\new, ': >>only talking about a single drive, exp
lain why SCSI is inherently\new, ': >>faster at managing data from a
hard drive.\new, '\new, ': >IDE: Integrated Device Electronics \new,
': > currently the most common standard, and is mainly used for med
ium sized \new, ': > drives. Can have more than one hard drive. Asyn
chronous Transfer: ~5MB/s max.\new, '\new, ": Why don't you start wit
h the spec-sheet of the ISA bus first?\new, ": You can quote SCSI sp
ecs till you're blue in the face, but if they\new, ": exceed the ISA
bus capability, then what's the point?\new, '\new, 'Who said ISA was
necessary? EISA or VLB are the only interfaces worth\new, "investin
g thousands of dollars (e.g. a new pc's worth of money ) in .\new,
'\new, ': Whosays IDE is limited to 5 megas/sec? What about VLB-ID
E? Does anyone\new, ': know how they perform?\new, '\new, "You didn't
read to carefully. VLB-IDE uses the same connection mechanism\new,
"as standard IDE. If transfer rate is limited by IDE, whether i
t's\new, 'interfaced to ISA, EISA or VLB matters not.\new, '\new, ':
>So at its LOWEST setting SCSI-2 interface in Asynchronous SCSI-1 m
ode AVERAGES \new, ': >the through put MAXIMUM of IDE in asynchronou
s mode. In full SCSI-2 mode\new, ': >it blows poor IDE out the wind
ow, down the street, and into the garbage can.\new, ': As implimente
d on what system? \new, '\new, 'On mine, for one thing. SCSI blows
IDE out of the water, hands down. If\new, "IDE has better throughpu
t, why isn't it used on workstations and file\new, 'servers? \new,
'\new, ': >The problem becomes can the drive mechanisim keep up with
those through put\new, ': >rates and THAT is where the bottleneck an
d cost of SCSI-2 comes from. NOT\new, ': >the interface itself but
more and more from drive mechanisims to use the\new, ': >SCSI-2 thro
ugh put. \new, '\new, ': Given the original question (SCSI used only
as a single hard drive\new, ': controller), is it then necessary to
get a SCSI drive that will do\new, ': at least 5, maybe 10 megas/sec
for the SCSI choice to make any sence?\new, ': What does a 200-400 m
eg 5 megas/sec SCSI drive cost?\new, '\new, "No, that's the nice thing
-- on a multitasking OS, SCSI can use both drives\new, "at once.
I've got unix loaded on one of my pcs (along with windogs) and the
OS can only use one of the two IDE drives at one time. It's pretty
ugly.\new, '\new, 'I just bought at Quantum 240 for my mac at home.
I paid $369 for it. I\new, "haven't seen IDE drives cheaper.\new,

```

'\n', ": The original CGA cart back in '84 was \$300. I think the original EGA card\n", ': (or VGA?) was \$800. SCSI has stood relatively alone in not coming down\n', ": in price, mainly because we're talking about PC's and not Sun's or Sparc\n", ': or SGI or (name your favorite unix workstation). That is, after millions\n', ': of PC buying decisions over the years, SCSI has had plenty of time to\n', ': come down in price.\n', '\n', "No, actually, we're talking about SCSI being expensive simply because\n", 'nobody did a common interface for the PC. If they had a common (read:\n', 'easily implemented) method of adding scsi to a PC (like as in a Sun or\n', "Mac), then you'd find SCSI the connection medium of choice.\n", '\n', ": I won't argue that the SCSI standard makes for a good, well implemented\n", ': data highway, but I still want to know why it intrinsically better\n', ": (than IDE, on an ISA bus) when it comes to multi-tasking OS's when\n", ': managing data from a single SCSI hard drive.\n', '\n', 'On a single drive, SCSI is more expensive. But, you bought your PC for\n', "expandability, so, you'd want to add more drives or whatever. The\n", 'following are why I find SCSI intrinsically better than IDE:\n', '\n', 'A (partial?) list:\n', '\t1. You can add many different types of devices and access them\n', '\tconcurrently.\n', '\t2. A SCSI device works on many different machines (I have a mac\n', '\tand a PC at home and moving hard drives between them is VERY nice\n', '\twith SCSI -- hook them up and away they go)\n', '\t3. SCSI devices work together better than IDE devices. For\n', '\tinstance, recently, I added an older connor 100 meg IDE to a maxtor\n', '\t212 meg IDE. The connor \*MUST\* be setup as the slave. It will\n', '\twork no other way. On SCSI, you set the address, check the\n', '\ttermination, plug it in, and away it goes.\n', '\t4. I have a problem with IDE's mutual exclusion - I notice that\n', '\tthe time it takes to switch from accessing drive c: to drive d: is\n', '\tquite long as compared to the time it takes to switch from drive c:\n', '\tto d: on a SCSI system. Under a multitasking OS, this is very\n', '\tnoticable, as many things can be going on at once.\n', '\n', "One neat thing that I've noticed lately (a fringe benefit) has been the\n", 'ability to add older (almost dead) drives as storage on a SCSI system with\n", "little problem -- we've got a bunch of almost dead 20 meg drives that I've\n", "added to my PC. I've now got the interface full, but, it does allow me to\n", 'have 4 20 meg drives, 1 240 meg drive, 1 tape drive, and 1 105 meg drive\n', 'all on the same card. \n', '\n', 'Simply put, SCSI is handier than IDE. No mysterious jumpers to figure out.\n', '\n', 'Greg.']

```
In [45]: # close file handle
         f
```

```
Out[45]: <closed file '/Users/jonathandinu/Repositories/workshopper/data-engineering-101/text/article0.txt', mode 'r' at 0x105c58780>
```

```
In [46]: for line in f:
         f
```

```
-----
-----
ValueError                                Traceback (most recent ca
ll last)
<ipython-input-46-8bb76d78f093> in <module>()
----> 1 for line in f:
      2     f

ValueError: I/O operation on closed file
```

```
In [48]: with open("/Users/jonathandinu/Repositories/workshopper/data-engine
         ering-101/text/article0.txt", 'r') as f:
         for line in f:
         print line
```

```
comp.sys.ibm.pc.hardware
```

```
: >>point of view, why does SCSI have an advantage when it comes to
multi-
```

```
: >>tasking? Data is data, and it could be anywhere on the drive.
Can
```

```
: >>SCSI find it faster? can it get it off the drive and into the
computer
```

```
: >>faster? Does it have a better cache system? I thought SCSI wa
s good at
```

```
: >>managing a data bus when multiple devices are attached. If we
are
```

```
: >>only talking about a single drive, explain why SCSI is inherent
ly
```

```
: >>faster at managing data from a hard drive.
```

```
: >IDE: Integrated Device Electronics
```

```
: > currently the most common standard, and is mainly used for medi
um sized
```

```
: > drives. Can have more than one hard drive. Asynchronous Transfe
r: ~5MB/s max.
```



: Why don't you start with the spec-sheet of the ISA bus first?

: You can quote SCSI specs till you're blue in the face, but if they

: exceed the ISA bus capability, then what's the point?

Who said ISA was necessary? EISA or VLB are the only interfaces worth

investing thousands of dollars (e.g. a new pc's worth of money ) in .

: Who says IDE is limited to 5 megs/sec? What about VLB-IDE? Does anyone

: know how they perform?

You didn't read to carefully. VLB-IDE uses the same connection mechanism

as standard IDE. If transfer rate is limited by IDE, whether it's interfaced to ISA, EISA or VLB matters not.

: >So at its LOWEST setting SCSI-2 interface in Asynchronous SCSI-1 mode AVERAGES

: >the through put MAXIMUM of IDE in asynchronous mode. In full SCSI-2 mode

: >it blows poor IDE out the window, down the street, and into the garbage can.

: As implemented on what system?

On mine, for one thing. SCSI blows IDE out of the water, hands down. If

IDE has better throughput, why isn't it used on workstations and file

servers?

: >The problem becomes can the drive mechanism keep up with those through put

: >rates and THAT is where the bottleneck and cost of SCSI-2 comes from. NOT

: >the interface itself but more and more from drive mechanisms to use the

: >SCSI-2 through put.

: Given the original question (SCSI used only as a single hard drive

: controller), is it then necessary to get a SCSI drive that will do

: at least 5, maybe 10 meg/sec for the SCSI choice to make any sense?

: What does a 200-400 meg 5 meg/sec SCSI drive cost?

No, that's the nice thing -- on a multitasking OS, SCSI can use both drives

at once. I've got unix loaded on one of my pcs (along with windows) and the OS can only use one of the two IDE drives at one time. It's pretty ugly.

I just bought a Quantum 240 for my mac at home. I paid \$369 for it. I

haven't seen IDE drives cheaper.

: The original CGA card back in '84 was \$300. I think the original EGA card

: (or VGA?) was \$800. SCSI has stood relatively alone in not coming down

: in price, mainly because we're talking about PC's and not Sun's or Sparc

: or SGI or (name your favorite unix workstation). That is, after

millions

: of PC buying decisions over the years, SCSI has had plenty of time to

: come down in price.

No, actually, we're talking about SCSI being expensive simply because

nobody did a common interface for the PC. If they had a common (read:

easily implemented) method of adding scsi to a PC (like as in a Sun or

Mac), then you'd find SCSI the connection medium of choice.

: I won't argue that the SCSI standard makes for a good, well implemented

: data highway, but I still want to know why it intrinsically better

: (than IDE, on an ISA bus) when it comes to multi-tasking OS's when

: managing data from a single SCSI hard drive.

On a single drive, SCSI is more expensive. But, you bought your PC for

expandibility, so, you'd want to add more drives or whatever. The

following are why I find SCSI intrinsically better than IDE:

A (partial?) list:

1. You can add many different types of devices and access them

concurrently.

2. A SCSI device works on many different machines (I have a mac

and a PC at home and moving hard drives between them is VERY nice

with SCSI -- hook them up and away they go)

3. SCSI devices work together better than IDE devices. For

instance, recently, I added an older Connor 100 meg IDE to a Maxtor

212 meg IDE. The Connor *\*MUST\** be setup as the slave. It will

work no other way. On SCSI, you set the address, check the termination, plug it in, and away it goes.

4. I have a problem with IDE's mutual exclusion - I notice that

the time it takes to switch from accessing drive c: to drive d: is

quite long as compared to the time it takes to switch from drive c:

to d: on a SCSI system. Under a multitasking OS, this is very

noticeable, as many things can be going on at once.

One neat thing that I've noticed lately (a fringe benefit) has been the

ability to add older (almost dead) drives as storage on a SCSI system with

little problem -- we've got a bunch of almost dead 20 meg drives that I've

added to my PC. I've now got the interface full, but, it does allow me to

have 4 20 meg drives, 1 240 meg drive, 1 tape drive, and 1 105 meg drive

all on the same card.

Simply put, SCSI is handier than IDE. No mysterious jumpers to figure

ure out.

Greg.