

c s 5 t o d a y

This would make me hungry...
but I ate breakfast this morning!



Computing to the max

The not-so-subtle art of singling out the best (and worst) of anything...

a comparison comparison

```
'm+ms' < [0, 42] < [4, 2]
[0, 'm+ms'] < [4, 'coffee']
```

> or <

Computing with language

- What's in an essay, anyway?
- Battle-tested ciphers & how to break them...

Last hw?

N-step sleepwalking?
Turtle graphics??
Artistic renderings!!!

This week!

Hw #3 due next Monday...

pr0: Are we *The Matrix*?

pr1: Lab...

pr2: Caesar, sorting, & more!

No Reading

to the max



Want the highest price?

```
max([10.06, 10.195, 10.025, 9.9, 9.875, 9.925])
```

GME

What if the dates are in there, as well?

GMEdate

```
max([10.06, '2002-02-13'], [10.195, '2002-02-14'])
```

```
max(['2002-02-13', 10.06], ['2002-02-14', 10.195])
```

dateGME

Mudd's max?

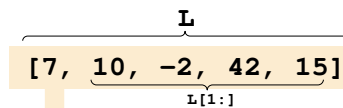
MSt

```
MSt = ['Harvey', 'Mudd', 'College', 'seeks', 'to', 'educate', 'engineers',
'scientists', 'and', 'mathematicians', 'well-versed', 'in', 'all', 'of', 'these',
'areas', 'and', 'in', 'the', 'humanities', 'and', 'the', 'social', 'sciences', 'so',
'that', 'they', 'may', 'assume', 'leadership', 'in', 'their', 'fields', 'with', 'a',
'clear', 'understanding', 'of', 'the', 'impact', 'of', 'their', 'work', 'on', 'society']
```

Or Mudd's min?

min (MSt)

max (MSt)



recursive max

L = ['aliens', 'zap', 'hazy', 'code']

```
def max(L):
```

```
    """Returns the maximum element from L.
    Argument: L, a nonempty list
    """
```

```
    if len(L) < 2:
        return L[0] # only 1 elem.
```

I lovemaxrest!

```
    maxOfRest = max(L[1:]) # max of the rest
```

```
    if L[0] > maxOfRest:
        return L[0] # either L[0]
```

```
    else:
        return maxOfRest # or maxOfRest!
```

max with scrabble-score

L = ['aliens', 'zap', 'hazy', 'code']

Which element has the highest scrabble score?

```
def maxSS(L):
```

```
    """Returns L's highest scrabble-scoring
    element. (Argument: L, a nonempty list)
    """
```

```
    if len(L) < 2:
        return L[0] # only 1 elem.
```

```
    maxOfRest = maxSS(L[1:]) # rest's max
```

```
    if L[0] > maxOfRest:
        return L[0] # either L[0]
```

```
    else:
        return maxOfRest # or maxOfRest!
```

Spacy!
I like it!

A more *comprehensive* solution: LoL

```
L = ['aliens', 'zap', 'hazy', 'code']
```

```
def maxSS(L):
    """Returns L's max-scrabble-score word.
    """
    LoL = [[sScore(w), w] for w in L]

    bestpair = max(LoL)

    return bestpair[1]
```



Quiz

```
L = ['aliens', 'zap', 'hazy', 'code']
def maxlen(L):
    LoL = [[len(s), s] for s in L]
    bstpr = max(LoL)
    return bstpr[1]
```

1. What is LoL? [6, 'aliens'], [3, 'zap'], [4, 'hazy'], [4, 'code']

2. What is bstpr? [6, 'aliens']

3. What is returned? 'aliens'

Try this on the back page first!

Extra!
Change exactly three characters in this code so that 3 is returned.

```
L = [30, 40, 50]
def bestnumb(L):
    """Returns the number in L closest to 42."""
    LoL = [[abs(x - 42), x] for x in L]
    bestpair = min(LoL)
    return bestpair[1]
```

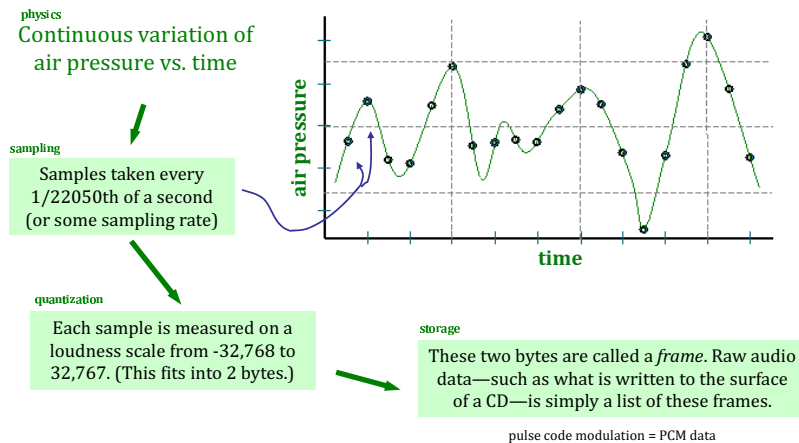
Hint: Python has `abs(x)` built-in

```
L = [3, 4, 5, 7, 6, 7]
def mostnumb(L):
    """Returns the most frequent item in L."""
    LoL = [[count(e, L), e] for e in L]
    best_pair = max(LoL)
    return best_pair[1]
```

Hint: Define a helper function!

```
def count(e, L):
    LC = [1 for x in L \
          if x == e]
    return sum(LC)
```

Lab3 ~ Sound



some examples...

Caesar Cipher: **encipher**

encipher(s, n)

What is it doing?

```
encipher('I <3 Latin', 0) returns 'I <3 Latin'
encipher('I <3 Latin', 1) returns 'J <3 Mbujo'
encipher('I <3 Latin', 2) returns 'K <3 Ncvkp'
encipher('I <3 Latin', 3) returns 'L <3 Odwlq'
encipher('I <3 Latin', 4) returns 'M <3 Pexmr'
encipher('I <3 Latin', 5) returns 'N <3 Qfyms'
...
encipher('I <3 Latin', 25) returns 'H <3 Kzshm'
```

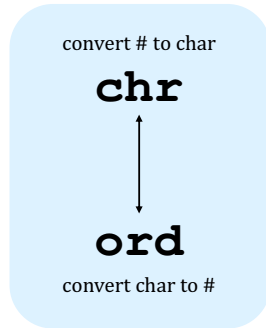
ASCII

American Standard Code for Information Interchange

Binary	Dec	Hex	Glyph
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1

1 byte
8 bits

The SAME bits represent an integer or a string, depending on type: `int` or `str`



ASCII and Unicode

convert # to char

chr

↕

ord

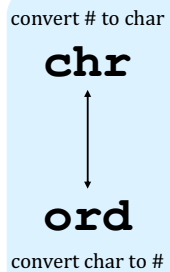
convert char to #

Binary	Dec	Hex	Glyph	Bin	Dec	Hex	Glyph	Bin	Dec	Hex	Glyph
0010 0000	32	20	(blank) (␣)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q

Julius, 9/25 This is why `'CS'` < `'clear'` !

ASCII and Unicode

chr value	abcdefghijklmnopqrstuvwxyz	ord value
	97 99 101 103 105 107 109 111 113 115 117 119 122	
chr value	ABCDEFGHIJKLMNOPQRSTUVWXYZ	ord value
	65 67 69 71 73 75 77 79 81 83 85 87 90	



What is `ord('U') // 2`?

What is `chr(ord('i') + 13)`?

What is `chr(ord('W') + 13)`?

how do we wrap?

Writing Rot13

```

any single character, c
def rot13(c):
    """Rotates c by 13 chars, "wrapping" as needed.
    NON-LETTERS don't change!
    """
    if 'a' <= c <= 'z':
        if ord(c) + 13 <= ord('z'):
            return chr(ord(c) + 13)
        else:
            return chr( )
    elif 'A' <= c <= 'Z':
        # uppercase test!
        return chr( )
    else:
        return c

```

(0) What do these tests do?

(1) What code will "wrap" to the alphabet's other side?

(2) How will uppercase change? Try noting only the code differences...

(3) What if c is not a letter at all?

Extra: How would you rotate n places, instead of 13?



Caesar

Caesar Cipher: **encipher**



Brutus

```
>>> encipher('Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.', 25)  s1
'Aycqyp agnfcq? G npcdcq Aycqyp qyjyb.'

>>> encipher('Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.', 15)
'Qosgof qwdvsf? W dfstsf Qosgof gozor.'

>>> encipher('Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.', 4)
'Fdhvdu flskhu? L suhihu Fdhvdu vdodg.'

>>> encipher('Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.', 1)
'Caesar cipher? I prefer Caesar salad.'

-----

>>> encipher('Hu lkbjhapvu pz doha ylthpuz hmaly dl mvynla '\
'lclyfaopun dl ohcl slhyulk.', 19)
'An education is what remains after we forget everything we
have learned.'  s2
```

Quiz

Name(s)

```
L = ['aliens', 'zap', 'hazy', 'code']  
def maxlen(L):  
    LoL = [[len(s), s] for s in L]
```

1. What is LoL? here is a start: LoL is [[6, 'aliens'], [3, 'zap'], _____, _____]

```
    bstpr = max(LoL)
```

2. What is bstpr?

```
    return bstpr[1]
```

3. What is returned?

Extra!

Change exactly three characters in this code so that 3 is returned.

```
L = [30, 40, 50]
```

Use the LoL method to write these two functions

```
def bestnumb(L):  
    """Returns the # in L closest to 42."""  
    LoL = [  
        bestpair =  
    return bestpair[1]
```

Hint: Python has `abs(x)` built-in

```
L = [3, 4, 5, 7, 6, 7]
```

```
def mostnumb(L):  
    """Returns the most frequent item in L."""  
    LoL = [  
    best_pair =  
    return best_pair[1]
```

Hint: Define a helper function!

```
def count(e, L):  
    """Return # of e's in L"""
```