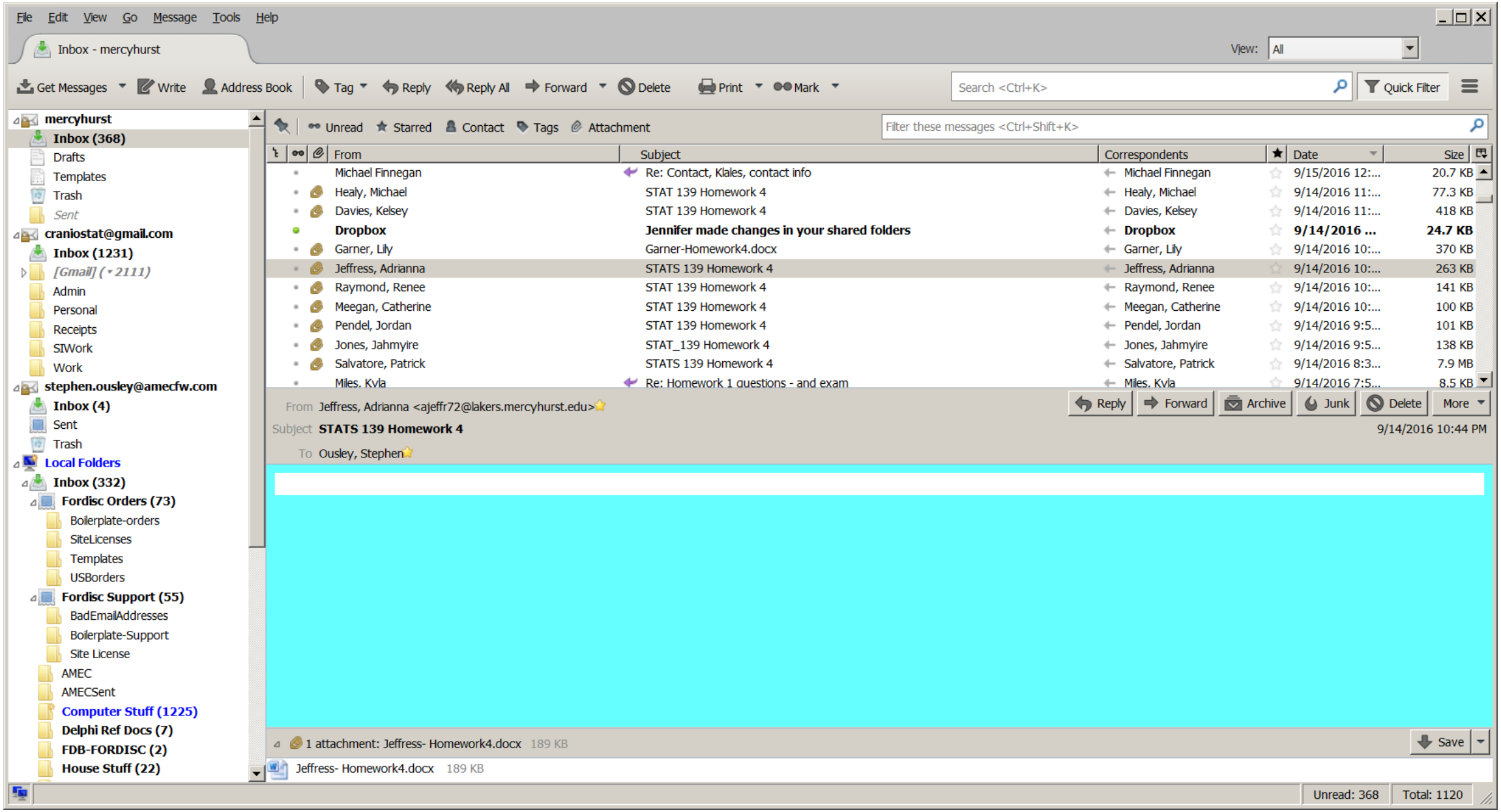
Functions II

Defining our own functions

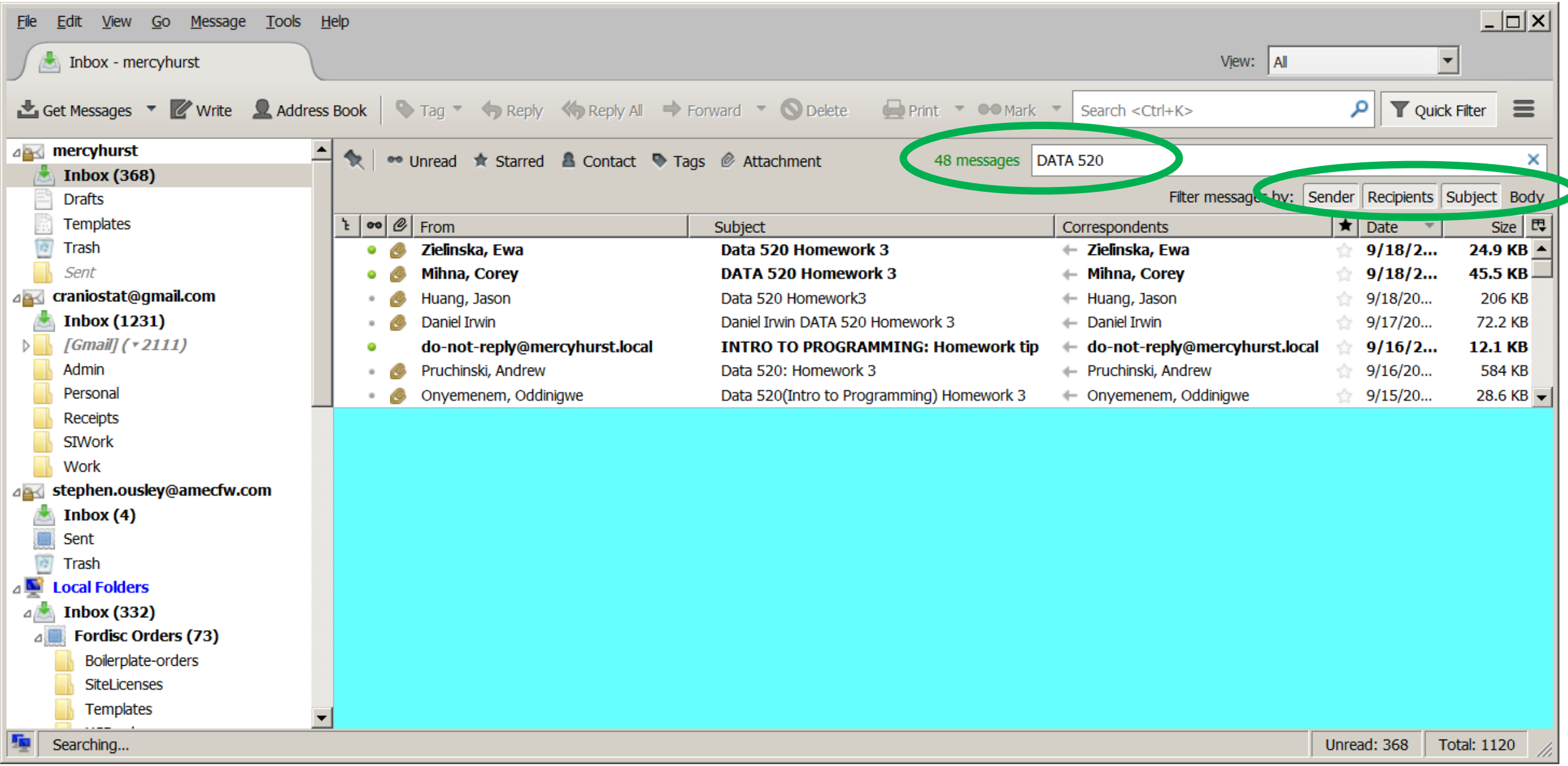
Formatting numbers

* also in ASCII text format (Lec5-Functions-2.txt)

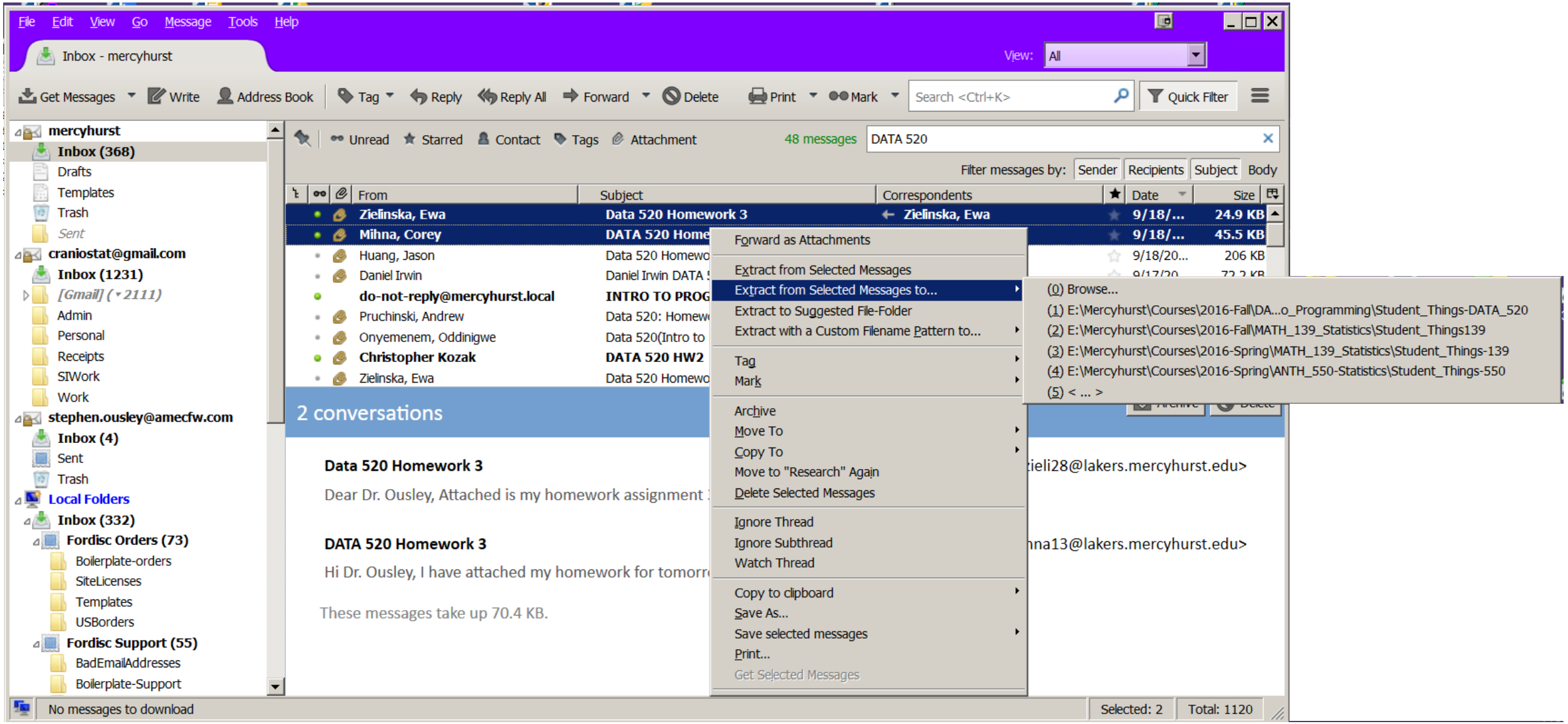
Another Application: Thunderbird



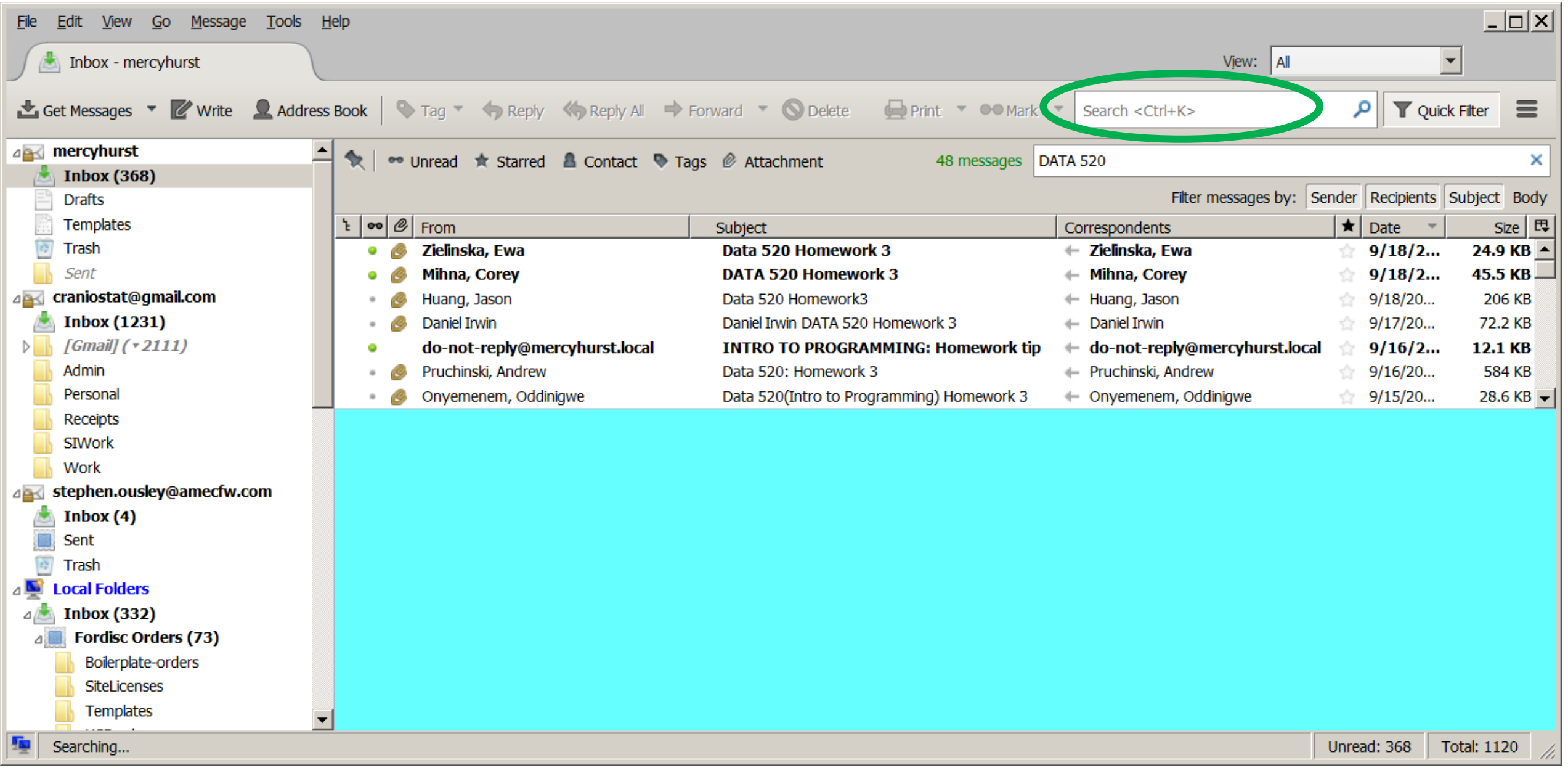
Local filter



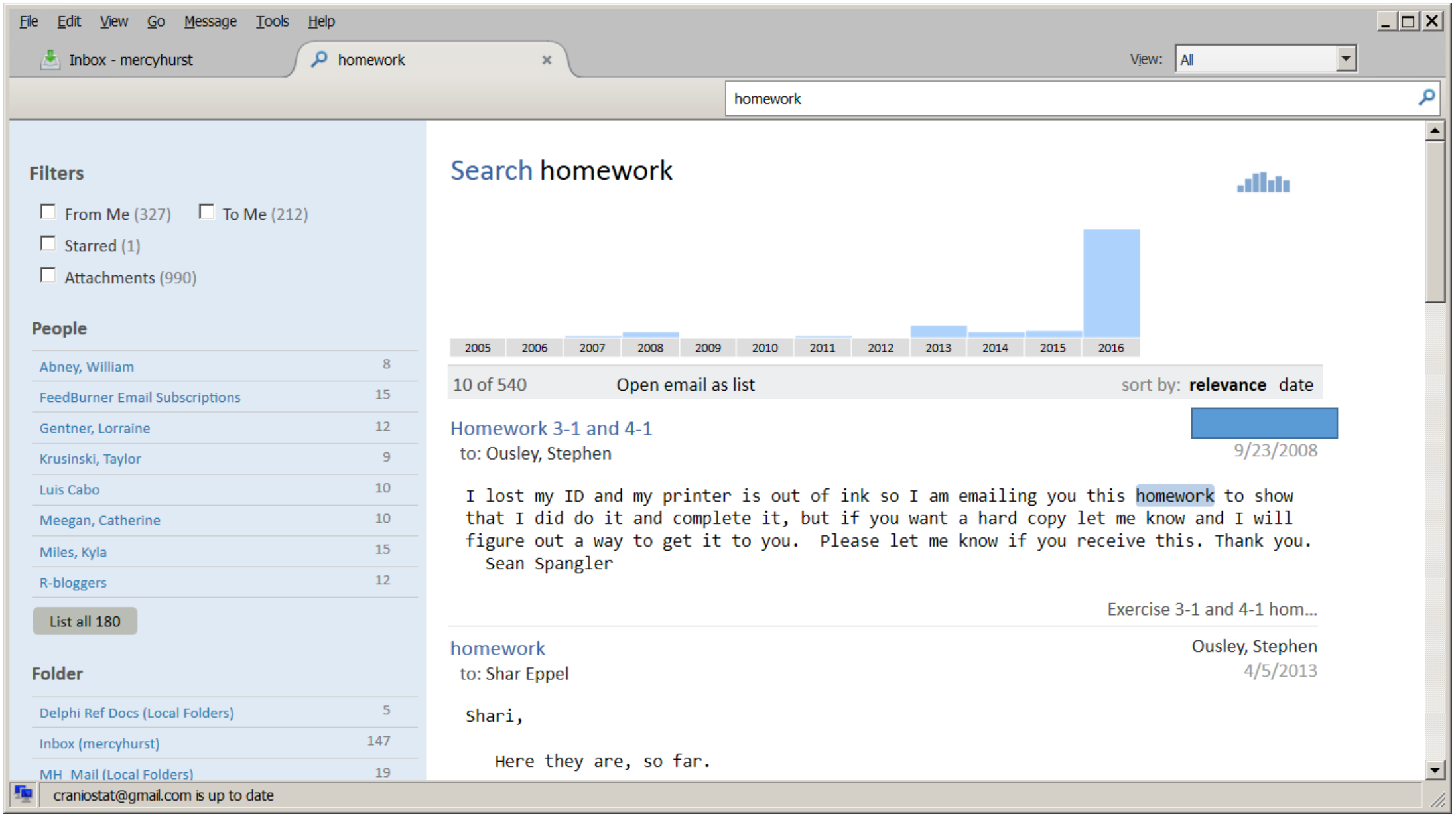
An Add-On:


File Edit View Go Message Tools Help
Inbox - mercyhurst
View: All
Get Messages Write Address Book Tag Reply Reply All Forward Delete Print Mark Search <Ctrl+K> Quick Filter
mercyhurst
Inbox (368)
Drafts
Templates
Trash
Sent
craniostat@gmail.com
Inbox (1231)
[Gmail] (2111)
Admin
Personal
Receipts
SIWork
Work
stephen.ousley@amecfw.com
Inbox (4)
Sent
Trash
Local Folders
Inbox (332)
Fordisc Orders (73)
Boilerplate-orders
SiteLicenses
Templates
USBorders
Fordisc Support (55)
BadEmailAddresses
Boilerplate-Support
Unread Starred Contact Tags Attachment
48 messages
DATA 520
Filter messages by: Sender Recipients Subject Body
From Subject Correspondents Date Size
Zielinska, Ewa Data 520 Homework 3 Zielinska, Ewa 9/18/... 24.9 KB
Mihna, Corey DATA 520 Home 9/18/... 45.5 KB
Huang, Jason Data 520 Homewo 9/18/20... 206 KB
Daniel Irwin Daniel Irwin DATA
do-not-reply@mercyhurst.local INTRO TO PROG
Pruchinski, Andrew Data 520: Homew
Onyemenem, Oddinigwe Data 520(Intro to
Christopher Kozak DATA 520 HW2
Zielinska, Ewa Data 520 Homewo
2 conversations
Data 520 Homework 3
Dear Dr. Ousley, Attached is my homework assignment
ieli28@lakers.mercyhurst.edu>
DATA 520 Homework 3
Hi Dr. Ousley, I have attached my homework for tomorr
nna13@lakers.mercyhurst.edu>
These messages take up 70.4 KB.
Forward as Attachments
Extract from Selected Messages
Extract from Selected Messages to...
Extract to Suggested File-Folder
Extract with a Custom Filename Pattern to...
Tag
Mark
Archive
Move To
Copy To
Move to "Research" Again
Delete Selected Messages
Ignore Thread
Ignore Subthread
Watch Thread
Copy to clipboard
Save As...
Save selected messages
Print...
Get Selected Messages
(0) Browse...
(1) E:\Mercyhurst\Courses\2016-Fall\DA...o_Programming\Student_Things-DATA_520
(2) E:\Mercyhurst\Courses\2016-Fall\MATH_139_Statistics\Student_Things139
(3) E:\Mercyhurst\Courses\2016-Spring\MATH_139_Statistics\Student_Things-139
(4) E:\Mercyhurst\Courses\2016-Spring\ANTH_550-Statistics\Student_Things-550
(5) < ... >
No messages to download
Selected: 2
Total: 1120

Global filter



Global search results



Defining functions

```
# We want to convert to Celsius
# anatomy of a function: def, then indented code, then return
def convert_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5 / 9
```

fahrenheit (inside the function) is a parameter

There are many many metric/English conversion functions

Defining functions more formally (the function design recipe)

```
def convert_to_celsius(fahrenheit):
      """ (number) -> float

    Return the number of Celsius degrees equivalent to fahrenheit degrees.

    >>> convert_to_celsius(75)
    23.8888888888889
    """
    return (fahrenheit - 32.0) * 5.0 / 9.0
```

Defining functions more formally (the function design recipe)

function header

```
def convert_to_celsius(fahrenheit):
    """ (number) -> float
```

docstring starts with three double quotes; provides parameter and return format (type contract)

number = integer or float

description

```
    Return the number of Celsius degrees equivalent to fahrenheit degrees.

    example(s)
    >>> convert_to_celsius(75)
    23.8888888888889
    """
```

body (the code that does something!)

```
    return (fahrenheit - 32.0) * 5.0 / 9.0
```

Defining functions more formally

Days difference macro

```
def days_difference(day1, day2):
    """ (int, int) -> int

    Return the number of days between day1 and day2, which are
    both in the range 1-365 (thus indicating the day of the
    year).

    >>> days_difference(200, 224)
    24
    >>> days_difference(50, 50)
    0
    >>> days_difference(100, 99)
    -1
    """
    return day2 - day1
```

Defining functions more formally

Our function provides help (pg.52)

```
help(days_difference)

Help on function days_difference in module __main__:

days_difference(day1, day2)
    (int, int) -> int

    Return the number of days between day1 and day2, which are both
    in the range 1-365 (thus indicating the day of the year).
    >>> days_difference(200, 224)
    24
    >>> days_difference(50, 50)
    0
    >>> days_difference(100, 99)
    -1
```

Day of the week in the future

Day of the Week	Number
Sunday	1
Monday	2
Tuesday	3
Wednesday	4
Thursday	5
Friday	6
Saturday	7

What day will it be x days from now?

What day of the week will it be, given the current weekday (1 - 7) and given how many days ahead we want to calculate

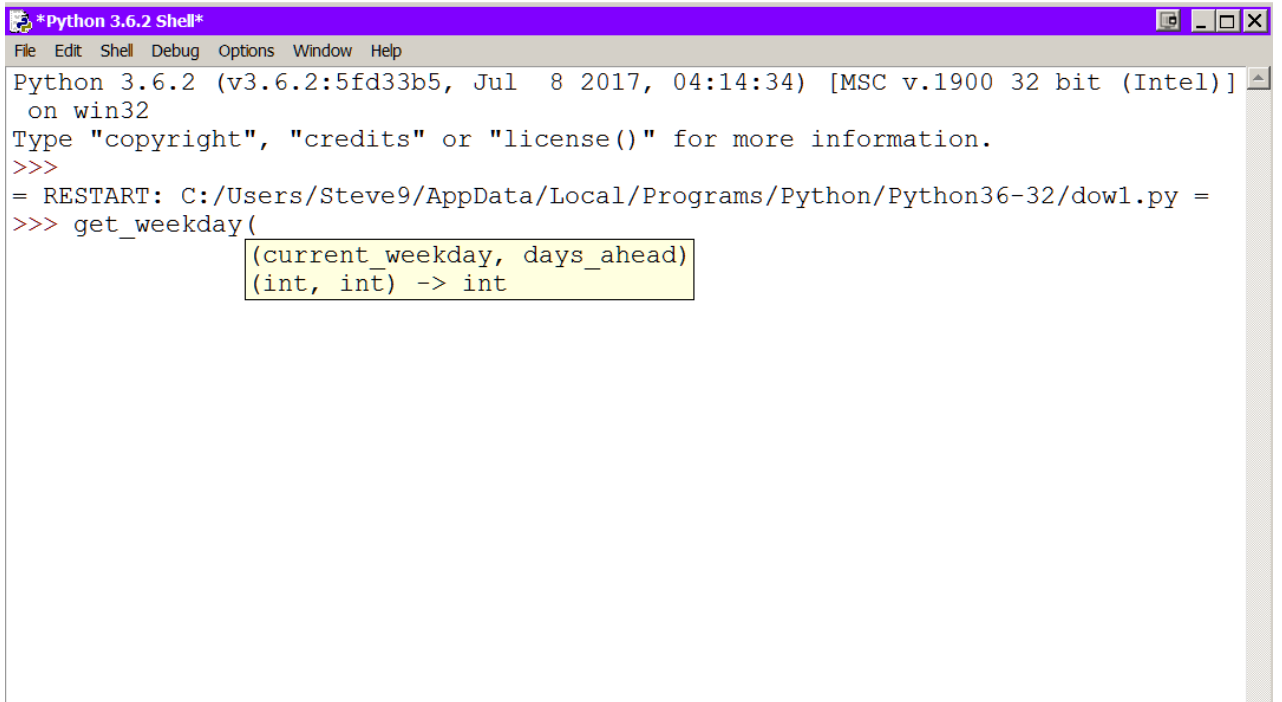
```
def get_weekday(current_weekday, days_ahead): # or day_current, days_ahead
    """ (int, int) -> int

    Return which day of the week it will be days_ahead days from
    current_weekday.

    current_weekday is the current day of the week and is in the
    range 1-7, indicating whether today is Sunday (1), Monday (2),
    ..., Saturday (7).

    days_ahead is the number of days after today.

    >>> get_weekday(3, 1)
    4
    >>> get_weekday(6, 1)
    7
    >>> get_weekday(7, 1)
    1
    >>> get_weekday(1, 0)
    1
    >>> get_weekday(4, 7)
    4
    >>> get_weekday(7, 72)
    2
    """
    return current_weekday + days_ahead % 7
```



What day will it be x days from now?

Test!!

```
>>> get_weekday(3, 1)
4
>>> get_weekday(6, 1)
7
>>> get_weekday(7, 1)
8
```

Oops! What happened!?!?!?

```
return current_weekday + days_ahead % 7
```

What day will it be x days from now?

```
def get_weekday(current_weekday, days_ahead):
    """ (int, int) -> int

    Return which day of the week it will be days_ahead days from
    current_weekday.

    current_weekday is the current day of the week and is in the
    range 1-7, indicating whether today is Sunday (1), Monday (2),
    ..., Saturday (7).

    days_ahead is the number of days after today.

    >>> get_weekday(3, 1)
    4
    >>> get_weekday(6, 1)
    7
    >>> get_weekday(7, 1)
    1
    >>> get_weekday(1, 0)
    1
    >>> get_weekday(4, 7)
    4
    >>> get_weekday(7, 72)
    2
    """
    return (current_weekday + days_ahead) % 7
```

What day will it be x days from now?

Test!!

```
>>> get_weekday(3, 1)
4
>>> get_weekday(7, 1)
1
>>> get_weekday(6, 1)
0
#oops!
```

What day of the week will it be x days from now?

Trick:

c. Subtract, then add 1 to the entire result: (current_weekday + days_ahead - 1) % 7 + 1.

Let's test it again:

```
def get_weekday(current_weekday, days_ahead):
    """ (int, int) -> int

    Return which day of the week it will be days_ahead days from
    current_weekday.

    current_weekday is the current day of the week and is in the
    range 1-7, indicating whether today is Sunday (1), Monday (2),
    ..., Saturday (7).

    days_ahead is the number of days after today.

    >>> get_weekday(3, 1)
    4
    >>> get_weekday(6, 1)
    7
    >>> get_weekday(7, 1)
    1
    >>> get_weekday(1, 0)
    1
    >>> get_weekday(4, 7)
    4
    >>> get_weekday(7, 72)
    2
    """
    return (current_weekday + days_ahead - 1) % 7 + 1
# was:
    return (current_weekday + days_ahead) % 7
```

What day of the week will it be on a certain date?

Limits:

think in terms of the day of the week in numbers (1 to 7)

think in terms of a date as a number (1 to 365)

What day of the week will it be on a certain date?

Limits:

think in terms of the **day** of the week in numbers (1 to 7)

think in terms of a **date** as a number (1 to 365)

all inputs

output: **day** of the week

```
get_birthday_weekday(day of week, current date number, target date ("birthday"))

get_birthday_weekday(5, 123, 284)
```

Function Header:

```
def get_birthday_weekday(current_weekday, current_day, birthday_day):
```

What day of the week will it be on a certain date?

```
# second.py
def get_birthday_weekday(current_weekday, current_day,
    birthday_day):
    """ (int, int, int) -> int

    Return the day of the week it will be on birthday_day,
    given that the day of the week is current_weekday and the
    day of the year is current_day.

    current_weekday is the current day of the week and is in
    the range 1-7, indicating whether today is Sunday (1),
    Monday (2),    , Saturday (7).

    current_day and birthday_day are both in the range 1-365.

    >>> get_birthday_weekday(5, 3, 4)
    6
    >>> get_birthday_weekday(5, 3, 116)
    6
    >>> get_birthday_weekday(6, 116, 3)
    5
    """
    days_diff = days_difference(current_day, birthday_day)
    return get_weekday(current_weekday, days_diff)
```

What day of the week will it be on a certain date?

```
>>>> get_birthday_weekday(5, 3, 14)
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    get_birthday_weekday(5, 3, 4)
  File "C:/Users/Steve9/AppData/Local/Programs/Python/Python36-32/second.py", line 22, in
get_birthday_weekday
    days_diff = days_difference(current_day, birthday_day)
NameError: name 'days_difference' is not defined
```

What happened??!?!?!

What day of the week will it be on a certain date?

What happened??!?!?!

Python "runs" and keeps in memory only what is current and loaded.
- all functions need to be in same file for now
add this code below the other and run:

```
def days_difference(day1, day2):
    """ (int, int) -> int

    Return the number of days between day1 and day2, which are
    both in the range 1-365 (thus indicating the day of the
    year).

    >>> days_difference(200, 224)
    24
    >>> days_difference(50, 50)
    0
    >>> days_difference(100, 99)
    -1
    """
    return day2 - day1
```

What day of the week will it be on a certain date?

```
>>>> get_birthday_weekday(5, 3, 14)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    get_birthday_weekday(5, 3, 4)
  File "C:/Users/Steve9/AppData/Local/Programs/Python/Python36-32/second.py", line 23, in
get_birthday_weekday
    return get_weekday(current_weekday, days_diff)
NameError: name 'get_weekday' is not defined
```

What happened??!?!?!

What day of the week will it be on a certain date?

What happened??!?!?!

Python "runs" and keeps in memory only what is current and loaded.
- ALL functions need to be in same file for now
add this trimmed code to end and run:

```
def get_weekday(current_weekday, days_ahead):
    """ (int, int) -> int

    Return which day of the week it will be days_ahead days from
    current_weekday.

    >>> get_weekday(7, 72)
    2
    """
    return (current_weekday + days_ahead - 1) % 7 + 1
```

What day of the week will it be on a certain date?

```
>>>> get_birthday_weekday(5, 3, 14)

Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    get_birthday_weekday(5, 3, 4)
  File "C:/Users/Steve9/AppData/Local/Programs/Python/Python36-32/second.py", line 23, in
get_birthday_weekday
    return get_weekday(current_weekday, days_diff)
NameError: name 'get_weekday' is not defined
```

What happened??!?!?!

What day of the week will it be on a certain date?

What happened??!?!?!

get_birthday_weekday function last line:

```
return (current_weekday + days_ahead) % 7
```

- any function called within the return statement must appear before the function that calls it

- as a general practice, you can move ALL functions called ahead of the externally called function

cut and paste get_weekday function to top and run

What day of the week will it be on a certain date?

```
get_birthday_weekday(5, 3, 14)
2
```

Success!!!!!!!!!!!!

Current layout in your python file: (works)

```
def days_difference(day1, day2):  # the first two can be switched
def get_weekday(current_weekday, days_ahead):
def get_birthday_weekday(current_weekday, current_day, birthday_day): return calls get_weekday
```

Will not work:

```
def get_weekday(current_weekday, days_ahead):
def get_birthday_weekday(current_weekday, current_day, birthday_day): return calls get_weekday
def days_difference(day1, day2):
```

Returning to the celsius function

```
def convert_to_celsius(fahrenheit):
      """ (number) -> float

    Return the number of Celsius degrees equivalent to fahrenheit degrees.

    >>> convert_to_celsius(75)
    23.8888888888889
    """
    return (fahrenheit - 32.0) * 5.0 / 9.0
```

I hate that output format!!!!

Formatting Numbers - another function

```
format (number, <format>) # format is a string
returns a string
<format>
'.2f' = two decimal places, float
'.0f' = rounded integer
'.6e' = 6 decimal places, scientific format
importantly, you always get a leading zero.
```

```
n = 23.88888888888889
format(n,'.2f')
'23.89'
```

```
>>> format(n,'.2f')
'23.89'
>>> format(n,'.6e')
'2.388889e+01'
>>> format(n,'.0f')
'24'
>>> format(n,'0.2f')
'23.89'
>>> format(0.003,'0.2f')
'0.00'
>>> format(0.035,'0.2f')
```

Returning to the celsius function

```
def convert_to_celsius(fahrenheit):
    """ (number) -> string  # hmmm

    Return the number of Celsius degrees equivalent to fahrenheit degrees.
    with 1 decimal place

    >>> convert_to_celsius(75)
    '23.9'
    """
    return format((fahrenheit - 32.0) * 5.0 / 9.0, '.1f')
```

I like that output format!!!!

Returning to the celsius function

```
def convert_to_celsius(fahrenheit):
    """ (number) -> float  # much better

    Return the number of Celsius degrees equivalent to fahrenheit degrees.
    with 1 decimal place

    >>> convert_to_celsius(75)
    23.9
    """
    return float( format( (fahrenheit - 32.0) * 5.0 / 9.0, '.1f') )
```

I LOVE that output format!!!!

Functions

Built-in Functions				
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

Homework 3 due before class Monday

Write a fahrenheit to celsius program that accepts another argument for number of digits precision

Be sure the output is a number (integer or float).

Due Monday before class