

# Functions II

Defining our own functions

Formatting numbers

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

# Another Application: Thunderbird

The screenshot displays the Mozilla Thunderbird email client interface. The window title is "Inbox - mercyhurst". The menu bar includes File, Edit, View, Go, Message, Tools, and Help. The toolbar contains buttons for Get Messages, Write, Address Book, Tag, Reply, Reply All, Forward, Delete, Print, and Mark. A search bar is located on the right side of the toolbar.

The left sidebar shows the folder structure:

- 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
      - Site License
    - AMEC
    - AMECSent
    - Computer Stuff (1225)
    - Delphi Ref Docs (7)
    - FDB-FORDISC (2)
    - House Stuff (22)

The main pane displays a list of messages. The selected message is from Jeffress, Adrianna, with the subject "STATS 139 Homework 4". The email content is mostly redacted with a large cyan box. The status bar at the bottom shows "Unread: 368" and "Total: 1120".

From	Subject	Correspondents	Date	Size
Michael Finnegan	Re: Contact, Kales, contact info	Michael Finnegan	9/15/2016 12:...	20.7 KB
Healy, Michael	STAT 139 Homework 4	Healy, Michael	9/14/2016 11:...	77.3 KB
Davies, Kelsey	STAT 139 Homework 4	Davies, Kelsey	9/14/2016 11:...	418 KB
<b>Dropbox</b>	<b>Jennifer made changes in your shared folders</b>	<b>Dropbox</b>	<b>9/14/2016 ...</b>	<b>24.7 KB</b>
Garner, Lily	Garner-Homework4.docx	Garner, Lily	9/14/2016 10:...	370 KB
Jeffress, Adrianna	STATS 139 Homework 4	Jeffress, Adrianna	9/14/2016 10:...	263 KB
Raymond, Renee	STAT 139 Homework 4	Raymond, Renee	9/14/2016 10:...	141 KB
Meegan, Catherine	STAT 139 Homework 4	Meegan, Catherine	9/14/2016 10:...	100 KB
Pendel, Jordan	STAT 139 Homework 4	Pendel, Jordan	9/14/2016 9:5...	101 KB
Jones, Jahmyire	STAT_139 Homework 4	Jones, Jahmyire	9/14/2016 9:5...	138 KB
Salvatore, Patrick	STATS 139 Homework 4	Salvatore, Patrick	9/14/2016 8:3...	7.9 MB
Miles. Kvla	Re: Homework 1 questions - and exam	Miles. Kvla	9/14/2016 7:5...	8.5 KB

From: Jeffress, Adrianna <ajeffr72@lakers.mercyhurst.edu>  
Subject: **STATS 139 Homework 4**  
To: Ousley, Stephen

1 attachment: Jeffress- Homework4.docx 189 KB

Jeffress- Homework4.docx 189 KB

Unread: 368 Total: 1120

# Local filter

The screenshot shows an email client interface with a local filter applied. The filter is named "DATA 520" and is currently showing 48 messages. The filter is applied to the "Inbox" of the "mercyhurst" account. The filter criteria are set to "Sender" and "Recipients". The filtered messages are listed in the main pane, showing the "From", "Subject", "Correspondents", "Date", and "Size" columns. The messages are sorted by date, with the most recent at the top. The messages are all related to "DATA 520 Homework 3".

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

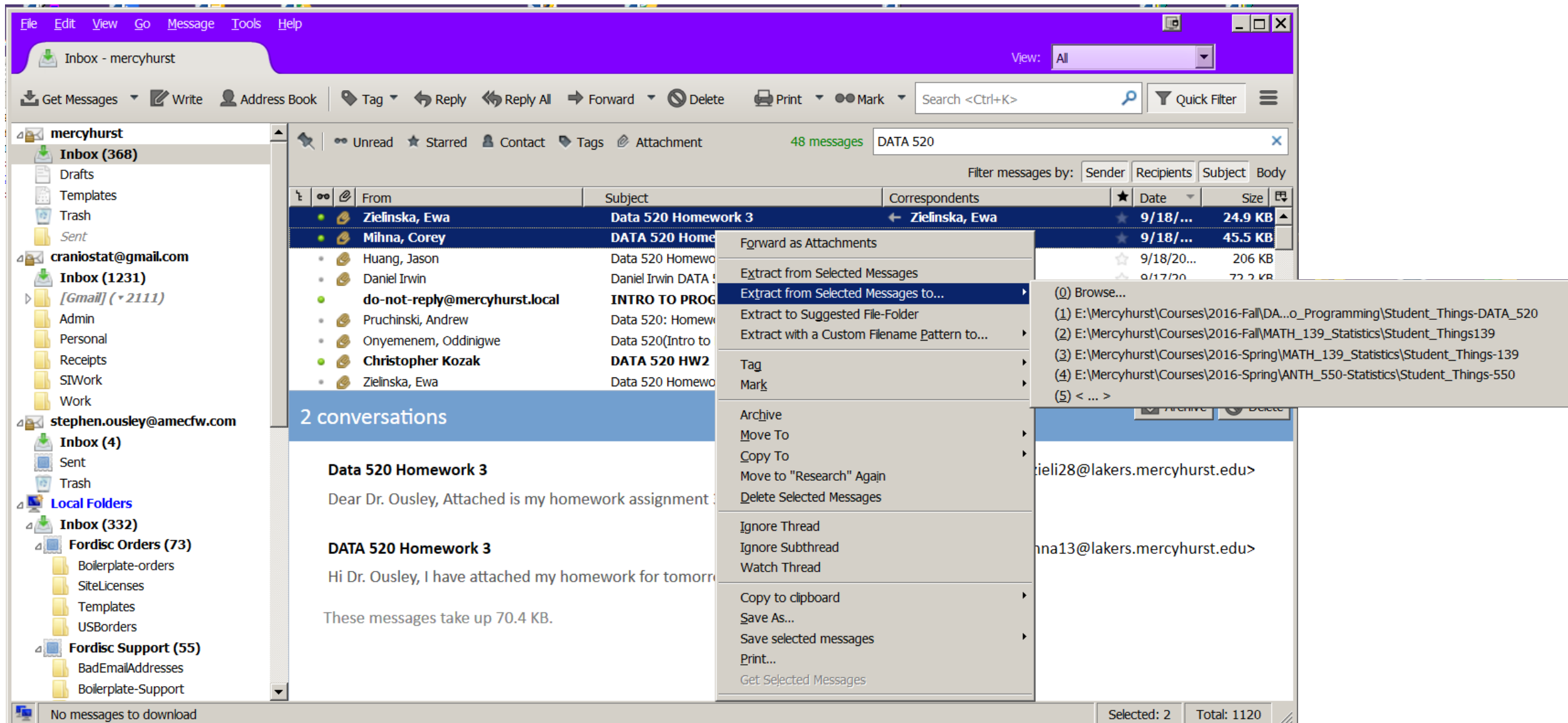
Unread Starred Contact Tags Attachment 48 messages DATA 520

Filter messages by: Sender Recipients Subject Body

From	Subject	Correspondents	Date	Size
Zielinska, Ewa	Data 520 Homework 3	Zielinska, Ewa	9/18/2...	24.9 KB
Mihna, Corey	DATA 520 Homework 3	Mihna, Corey	9/18/2...	45.5 KB
Huang, Jason	Data 520 Homework3	Huang, Jason	9/18/20...	206 KB
Daniel Irwin	Daniel Irwin DATA 520 Homework 3	Daniel Irwin	9/17/20...	72.2 KB
do-not-reply@mercyhurst.local	INTRO TO PROGRAMMING: Homework tip	do-not-reply@mercyhurst.local	9/16/2...	12.1 KB
Pruchinski, Andrew	Data 520: Homework 3	Pruchinski, Andrew	9/16/20...	584 KB
Onyemenem, Oddinigwe	Data 520(Intro to Programming) Homework 3	Onyemenem, Oddinigwe	9/15/20...	28.6 KB

Searching... Unread: 368 Total: 1120

# An Add-On:



# Global filter

The screenshot shows an email client interface with a global filter applied. The filter is "DATA 520", which is highlighted with a green circle. The filter is applied to the "Inbox (368)" of the "mercyhurst" account. The filter results show 48 messages. The messages are listed in a table with columns: From, Subject, Correspondents, Date, and Size. The messages are filtered by the subject "DATA 520".

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

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	9/18/2...	24.9 KB
Mihna, Corey	DATA 520 Homework 3	Mihna, Corey	9/18/2...	45.5 KB
Huang, Jason	Data 520 Homework3	Huang, Jason	9/18/20...	206 KB
Daniel Irwin	Daniel Irwin DATA 520 Homework 3	Daniel Irwin	9/17/20...	72.2 KB
do-not-reply@mercyhurst.local	INTRO TO PROGRAMMING: Homework tip	do-not-reply@mercyhurst.local	9/16/2...	12.1 KB
Pruchinski, Andrew	Data 520: Homework 3	Pruchinski, Andrew	9/16/20...	584 KB
Onyemenem, Oddinigwe	Data 520(Intro to Programming) Homework 3	Onyemenem, Oddinigwe	9/15/20...	28.6 KB

Searching... Unread: 368 Total: 1120

# Global search results

File Edit View Go Message Tools Help

Inbox - mercyhurst homework

View: All

homework

Filters

☐ From Me (327)

☐ To Me (212)

☐ Starred (1)

☐ Attachments (990)

People


Abney, William	8
FeedBurner Email Subscriptions	15
Gentner, Lorraine	12
Krusinski, Taylor	9
Luis Cabo	10
Meegan, Catherine	10
Miles, Kyla	15
R-bloggers	12

List all 180

Folder

Delphi Ref Docs (Local Folders)	5
Inbox (mercyhurst)	147
MH Mail (Local Folders)	19

Search homework



2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016

10 of 540 Open email as list sort by: relevance date

Homework 3-1 and 4-1

to: Ousley, Stephen

9/23/2008

I lost my ID and my printer is out of ink so I am emailing you this homework to show that I did do it and complete it, but if you want a hard copy let me know and I will figure out a way to get it to you. Please let me know if you receive this. Thank you.  
Sean Spangler

Exercise 3-1 and 4-1 hom...

homework

to: Shar Eppel

Ousley, Stephen

4/5/2013

Shari,

Here they are, so far.

craniostat@gmail.com is up to date

# 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)
```

```
23.888888888888889
```

```
"""
```

```
    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
```

```
        number = integer or float
```

docstring starts with three double quotes;

provides parameter and return format (type contract)

## description

Return the number of Celsius degrees equivalent to fahrenheit degrees.

## example(s)

```
>>> convert_to_celsius(75)
```

```
23.888888888888889
```

```
"""
```

## 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)
```

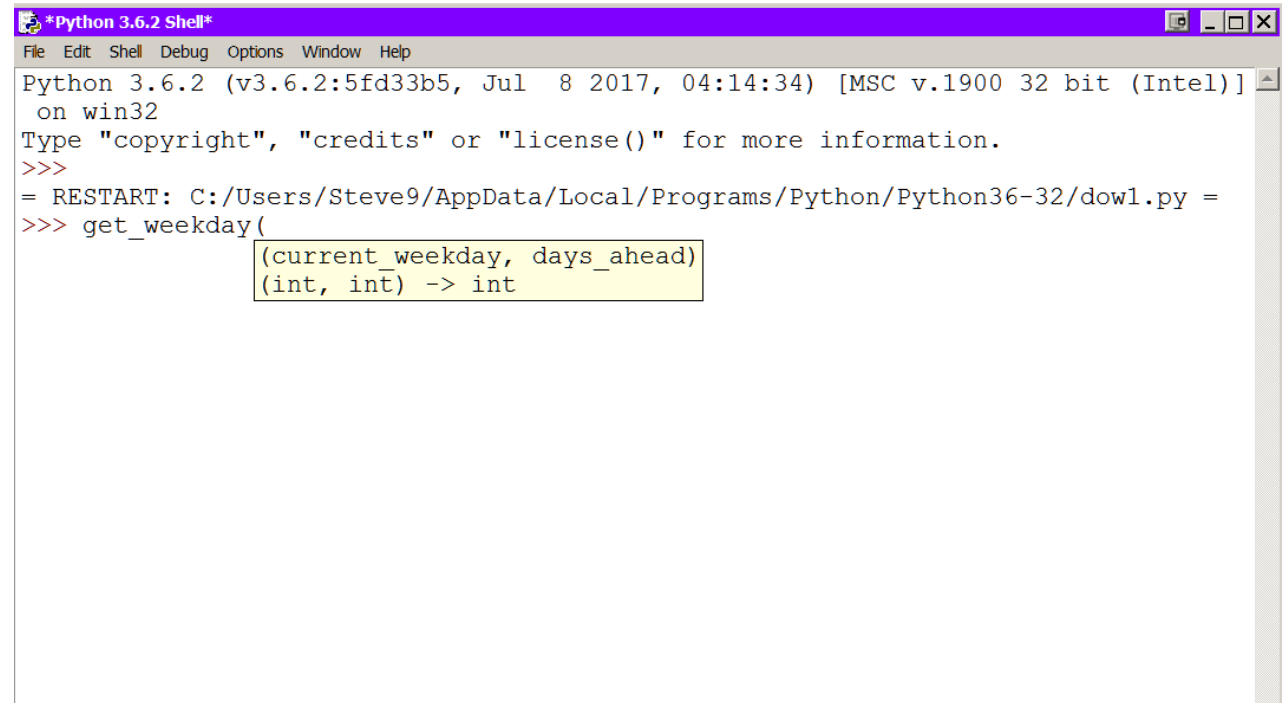
```
4
```

```
>>> get_weekday(7, 72)
```

```
2
```

```
"""
```

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



```
*Python 3.6.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Steve9/AppData/Local/Programs/Python/Python36-32/dow1.py =
>>> get_weekday(
    (current_weekday, days_ahead)
    (int, int) -> int
```

# 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:  $(\text{current\_weekday} + \text{days\_ahead} - 1) \% 7 + 1$ .

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)
```

```
23.888888888888889
```

**I hate that output format!!!!**

```
"""
```

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

# 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 = 23.888888888888889
```

```
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)    I like that output format!!!!
```

```
'23.9'
```

```
"""
```

```
    return format((fahrenheit - 32.0) * 5.0 / 9.0, '.1f')
```

# 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)    I LOVE that output format!!!!
```

```
23.9
```

```
"""
```

```
return float( format( (fahrenheit - 32.0) * 5.0 / 9.0, '.1f') )
```

# Functions

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



# 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