

What IF?

Branching using IF statements

Flow control/Control flow statements

SuperFloatPow

A negative number with a float exponent won't work:

```
>>> pow(2.2,4.4)
```

```
32.111441048903984
```

```
>>> pow(-2.2,4.4)
```

```
(9.92298099798058+30.53979525718783j)
```

```
>>> -2.2**4.4
```

```
-32.111441048903984
```

```
>>> (-2.2)**4.4
```

```
(9.92298099798058+30.53979525718783j)
```

"If both x and y are finite, x is negative, and y is not an integer then pow(x, y) is undefined, and raises ValueError."

- because you are asking for a root of a negative number (not a "real" number - an imaginary one!)

- it is a ValueError

CCleaner (at least version 5.34)

Piriform CCleaner

CCleaner Free v4.19.4867 (64-bit)
MS Windows 7 64-bit SP1
Intel Xeon CPU E3-1270 v3 @ 3.50GHz, 16.0GB RAM, AMD FirePro W5000 (FireGL V) Graphics Adapter

Cleaner
Registry
Tools
Options

Windows Applications

Internet Explorer

- ☒ Temporary Internet Files
- ☒ History
- ☒ Cookies
- ☒ Recently Typed URLs
- ☒ Index.dat files
- ☒ Last Download Location
- ☐ Autocomplete Form History
- ☐ Saved Passwords

Windows Explorer

- ☐ Recent Documents
- ☒ Run (in Start Menu)
- ☒ Search Autocomplete
- ☒ Other Explorer MRUs
- ☒ Thumbnail Cache
- ☒ Taskbar Jump Lists
- ☐ Network Passwords

System

- ☒ Empty Recycle Bin
- ☒ Temporary Files
- ☒ Clipboard
- ☒ Memory Dumps
- ☒ Chkdsk File Fragments
- ☒ Windows Log Files
- ☐ Windows Error Reporting
- ☐ DNS Cache
- ☐ Font Cache
- ☐ Start Menu Shortcuts
- ☐ Desktop Shortcuts

Advanced

- ☐ Windows Event Logs
- ☒ Old Prefetch data
- ☐ Menu Order Cache
- ☐ Trav Notifications Cache

ANALYSIS COMPLETE - (5.838 secs)

1,454 MB to be removed. (Approximate size)

Details of files to be deleted (Note: No files have been deleted yet)


Internet Explorer - Temporary Internet Files	42,979 KB	426 files
Internet Explorer - History	11 KB	2 files
Internet Explorer - Cookies	2 KB	9 files
System - Temporary Files	1,321,748 KB	2,428 files
System - Memory Dumps	9,062 KB	1 files
System - Windows Log Files	7,571 KB	13 files
Firefox - Internet Cache	Skipped	
Firefox - Internet History	Skipped	
Firefox - Session	217 KB	6 files
Google Chrome - Internet Cache	13,596 KB	41 files
Google Chrome - Internet History	165 KB	6 files
Google Chrome - Cookies	3 KB	1 files
Google Chrome - Session	2 KB	2 files
Applications - Adobe Acrobat XI	46,067 KB	125 files
Applications - Adobe Reader 11.0	46,008 KB	124 files
Applications - Office 2013	10 KB	8 files
Multimedia - Adobe Flash Player	1 KB	5 files
Multimedia - Macromedia Shockwave 10	1 KB	1 files
Utilities - Windows Defender	66 KB	1 files
Windows - MS Search	1,024 KB	1 files


Analyze Run Cleaner


Online Help Check for updates...


Ccleaner


<https://www.piriform.com/ccleaner/download>


 Piriform CCleaner


**CCleaner Free** v4.19.4867 (64-bit)
MS Windows 7 64-bit SP1
Intel Xeon CPU E3-1270 v3 @ 3.50GHz, 16.0GB RAM, AMD FirePro W5000 (FireGL V) Graphics Adapter

 Cleaner

 Registry

 Tools

 Options

 Registry Cleaner

☒ Missing Shared DLLs

☒ Unused File Extensions

☒ ActiveX and Class Issues

☒ Type Libraries

☒ Applications

☒ Fonts

☒ Application Paths

☒ Help Files

☒ Installer

☒ Obsolete Software

☒ Run At Startup

☒ Start Menu Ordering

☒ MUI Cache

☒ Sound Events

☒ Windows Services

100%

<input checked="" type="checkbox"/>	Problem	Data	Registry Key
<input checked="" type="checkbox"/>	Unused File Extension	mk	HKCR\mk
<input checked="" type="checkbox"/>	Unused File Extension	res	HKCR\res
<input checked="" type="checkbox"/>	Unused File Extension	.	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.
<input checked="" type="checkbox"/>	Unused File Extension	.adi	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\adi
<input checked="" type="checkbox"/>	Unused File Extension	.DS_Store	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\DS_Store
<input checked="" type="checkbox"/>	Unused File Extension	.dtpart	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\dtpart
<input checked="" type="checkbox"/>	Unused File Extension	.part	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\part
<input checked="" type="checkbox"/>	Unused File Extension	.vec	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\vec
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.3g2	HKCR\ApplianMP.3g2
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.3ga	HKCR\ApplianMP.3ga
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.3gp	HKCR\ApplianMP.3gp
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.3gp2	HKCR\ApplianMP.3gp2
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.3gpp	HKCR\ApplianMP.3gpp
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.669	HKCR\ApplianMP.669
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.a52	HKCR\ApplianMP.a52
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.aac	HKCR\ApplianMP.aac
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.ac3	HKCR\ApplianMP.ac3
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.adt	HKCR\ApplianMP.adt
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.adts	HKCR\ApplianMP.adts
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.aif	HKCR\ApplianMP.aif
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.aifc	HKCR\ApplianMP.aifc
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.aiff	HKCR\ApplianMP.aiff
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.amr	HKCR\ApplianMP.amr
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.amv	HKCR\ApplianMP.amv
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.aob	HKCR\ApplianMP.aob
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.ape	HKCR\ApplianMP.ape
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.asf	HKCR\ApplianMP.asf
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.asx	HKCR\ApplianMP.asx
<input checked="" type="checkbox"/>	Invalid or empty file class	ApplianMP.au	HKCR\ApplianMP.au


Scan for Issues





Fix selected issues...

Online Help

Check for updates...

CCleaner

**CCleaner Free** v4.19.4867 (64-bit)
MS Windows 7 64-bit SP1
Intel Xeon CPU E3-1270 v3 @ 3.50GHz, 16.0GB RAM, AMD FirePro W5000 (FireGL V) Graphics Adapter

Cleaner
Registry
Tools
Options

Uninstall

Startup





























File Finder

System Restore

Drive Wiper

Uninstall

Select a program from the list you want to remove from your computer

Programs to Remove	Publisher	Install Date	Size	Version	
 ADBOU_2.1 version 2.1.041		4/27/2015	5.85 MB	2.1.041	
 Adobe Acrobat Reader DC	Adobe Systems Incorporated	9/5/2017	256 MB	17.012.20098	
 Adobe Acrobat XI Pro	Adobe Systems	9/6/2017	12.2 GB	11.0.22	
 Adobe Creative Cloud	Adobe Systems Incorporated	2/15/2016	298 MB	3.5.1.209	
 Adobe Dreamweaver CC 2015	Adobe Systems Incorporated	8/26/2015	757 MB	16.0.0	
 Adobe Flash Player 27 NPAPI	Adobe Systems Incorporated	9/12/2017	6.17 MB	27.0.0.130	
 Adobe Shockwave Player 12.2	Adobe Systems, Inc.	1/20/2017		12.2.5.195	
 Advantage Data Architect v10.10	iAnywhere, Inc.	10/1/2014	55.6 MB	10.10.0049	
 Advantage ODBC Driver for x86_64 v10.10	Sybase, Inc.	10/5/2015	57.0 MB	10.10.0049	
 Advantage TDataSet Descendant for Delphi/C++Builder v10.10	iAnywhere, Inc.	10/28/2014	54.3 MB	10.10.0017	
 AMD Catalyst Install Manager	Advanced Micro Devices, Inc.	8/1/2014	26.2 MB	8.0.915.0	
 Borland Delphi 6	Borland Software Corporation	10/28/2014	313 MB	6.0	
 CCleaner	Piriform	10/28/2014		4.19	
 ConvertHelper 3.1.1	DownloadHelper	9/14/2015	76.9 MB		
 CutePDF Writer 3.0	Acro Software Inc.	9/22/2014		3.0	
 Dell Backup and Recovery	Dell Inc.	8/1/2014		1.9.2.8	
 Dell Client System Update	Dell Inc.	8/1/2014	33.7 MB	1.3.0	
 Dell Digital Delivery	Dell Products, LP	8/1/2014		2.9.901.0	
 Dell Protected Workspace	Invincea, Inc.	8/1/2014	70.0 MB	2.3.15835	
 DemoForge Mirage Driver for TightVNC 2.0	DemoForge LLC	9/23/2015	1.43 MB	2.0	
 DisplayFusion 8.1.2	Binary Fortress Software	1/11/2017	62.2 MB	8.1.2.0	
 Dropbox	Dropbox, Inc.	9/18/2017		34.4.22	
 Dual Monitor 1.22	Cristi Diaconu	10/13/2014	2.52 MB	1.22.021813	
 EaseUS Partition Master 11.5	EaseUS	8/26/2016	205 MB		
 EditPad Lite 7.3.4	Just Great Software	9/16/2014	15.1 MB	7.3.4	
 Everything 1.3.4.686 (x86)		9/15/2014			
 FileSearchy	Midlinesoft	2/22/2017		1.43	
 FileSearchy	The University of Tennessee	8/17/2014	54.3 MB		

Uninstall

Repair

Rename

Delete


Search


Save to text file...





Online Help

Check for updates...

CCleaner

 Piriform CCleaner

**CCleaner Free** v4.19.4867 (64-bit)
MS Windows 7 64-bit SP1
Intel Xeon CPU E3-1270 v3 @ 3.50GHz, 16.0GB RAM, AMD FirePro W5000 (FireGL V) Graphics Adapter

 Cleaner
 Registry
 Tools
 Options

Uninstall

Startup

File Finder

System Restore

Drive Wiper

System Restore

Manage all your System Restore points (The latest one is disabled for system safety)

Date and Time	Description
9/6/2017 1:31:50 PM	Python 3.6.2 (32-bit)
8/23/2017 4:28:37 PM	Scheduled Checkpoint
5/11/2017 9:37:46 AM	Windows Update
4/18/2017 5:16:46 PM	Windows Update
4/10/2017 7:14:45 PM	Windows Update

Remove

[Online Help](#)[Check for updates...](#)

Boolean logic

From George Boole (1815 - 1864); 1840's: Boolean Algebra

Evaluate a statement as EITHER true OR false

a number is either > 3 , OR ≤ 3

In Python, the type is bool

Boolean Operators

not - highest precedence

and - next

or - lowest

If not true? - must be false, other examples confusing

BUT type in:

```
not true
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#3>", line 1, in <module>
```

```
    not true
```

```
NameError: name 'true' is not defined
```

```
#case-sensitive!
```

```
not True
```

```
False
```

```
not False
```

```
True
```


Boolean Operators

"or" may be a bit better to understand (not "OR")

```
>>> True or True # (IF True or True) then at least one must be True
True
```

```
>>> False or False # (IF False or False) then both are False
False
```

```
>>> True or False # (IF True or False) then at least one must be True
True
```

```
>>> False or True# (IF False or True) then at least one must be True
True
```

"inclusive" or: if EITHER condition met, do something (true)

"exclusive" or: "Chicken OR pasta?" , "Yes." , "Which?"

Boolean Operators

Combinations of and + or: Always use parentheses with OR!

```
>>> b1 = False
```

```
>>> b2 = False
```

```
(b1 and not b2) or (b2 and not b1)
```

```
False
```

```
b1 and not b2 or b2 and not b1
```

```
False
```

```
not b1 and not b2
```

```
True
```

```
not b1 or not b2 or (1/0) # 1/0 is undefined
```

```
True
```

**NOTE: statements are not completely evaluated;
as soon as one answer is clear, it moves on
aka "incomplete evaluation"**

I want to run more examples with multiple lines.

How can we save time typing in multiple lines?

Let's use a file!

Hmm - how?

Boolean Operators

```
b1 = False
```

```
b2 = True
```

```
(b1 and not b2) or (b2 and not b1)
```

Will it run in a file?

Boolean Operators

```
b1 = False
```

```
b2 = True
```

```
(b1 and not b2) or (b2 and not b1)
```

Will it run in a file?

YES!

- but no answer is given.

How to get an answer back?

print! (return if it were a function)

change last line to:

```
print (b1 and not b2) or (b2 and not b1)
```

Boolean Operators

In file:

```
b1 = False
```

```
b2 = True
```

```
print(b1 and not b2) or (b2 and not b1)
```

```
False
```

is that right?

```
# oops, interpreted as:
```

```
# print( (b1 and not b2) ) or (b2 and not b1) # second part is evaluated... and forgotten
```

```
# change to:
```

```
print( (b1 and not b2) or (b2 and not b1) )
```

then run...

Boolean Operators

```
# file code
```

```
b1 = True
```

```
b2 = False
```

```
print((b1 and not b2) or (b2 and not b1))
```

```
True
```

```
# file code
```

```
b1 = True
```

```
b2 = True
```

```
print((b1 and not b2) or (b2 and not b1))
```

```
False
```

Boolean logic

Parentheses are important for logic too!

An Erie winter example:

```
cold = True
```

```
windy = False
```

```
(not cold) and windy
```

```
False
```

```
not (cold and windy)
```

```
True
```

Boolean logic

A Truth Table

cold	windy	cold and windy	cold or windy	(not cold) and windy	not (cold and windy)
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	True
False	False	False	False	False	True

Table 5—Boolean Operators

Boolean logic

More transparent comparisons (not &&):

Symbol	Operation
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Boolean logic

Easy to understand

`3 > 5`

False

`3 < 5`

True

`3 <= 5`

True

with assignment

`x = 3 <= 3`

x

True

`3 == 4` ; because = used for variable assignment (`x = 3`)

False

Boolean logic

be careful with floats

1/3 == 0.33333

False

[illegible]

True

1 / 3

0.3333333333333333

Boolean logic in a function

```
def is_positive(x):  
    """ (number) -> bool  
  
    Return True iff x is positive. # iff = "if and only if"  
  
    >>> is_positive(3)  
    True  
    >>> is_positive(-4.6)  
    False  
    """  
    return x > 0
```

Boolean logic: combinations

`+, /` are evaluated before `<, >`
comparisons before `and, or, not`

```
1 + 3 > 7 : (1 + 3) > 7 #
```

```
x = 5
```

```
y = 10
```

```
z = 20
```

```
x < y and y < z # = (x < y) and (y < z); more readable?
```

```
# combinations with and
```

```
x = 3
```

```
(1 < x) and (x <= 5)
```

```
True
```

Boolean logic: combinations

```
# combos with and
```

```
x = 3
```

```
(1 < x) and (x <= 5)
```

```
True
```

```
# we can "chain" the comparison:
```

```
1 < x <= 5
```

```
True
```

```
3.21 < 3.5 < 3.5000001
```

```
True
```

Boolean logic: combinations

From Gries: a strange example (**don't do this!**)

Most chaining works as you would expect, but:

```
3 < 5 != True
```

```
True
```

```
3 < 5 != False
```

```
True
```

the first one is equivalent to:

```
(3 < 5) and (5 != True)
```

the second is equivalent to:

```
(3 < 5) and (5 != False)
```

Because 5 is neither True nor False

another strange one:

```
not 0
```

```
True
```

Boolean logic: strings

Strings are compared using ASCII key codes

Lowercase letters > Uppercase letters

```
'a' > 'A'    # blame ASCII
```

```
True
```

```
'a' > 'Z'    # wow!
```

```
True
```

```
'abc' < 'abd'
```

```
True
```

```
'abc' < 'abcd'
```

```
True
```


Boolean logic: strings

Python has a built-in string search function: in

```
'Jan' in '01 Jan 1838'
```

True

```
'Feb' in '01 Jan 1838'
```

False

```
'Steve' in 'Stephen' #?
```

```
'ACGCGCGTT'
```

```
' in DNasequence # ????
```

```
date = input('Enter a date in the format DD/MM/YYYY (09/12/2009): ')
```

```
# check for March
```

```
 '/03/' in date
```

Boolean logic: Branching

pH Level	Solution Category
0–4	Strong acid
5–6	Weak acid
7	Neutral
8–9	Weak base
10–14	Strong base

```
ph = float(input('Enter the pH level: '))
```

```
Enter the pH level: 6.0
```

```
if ph < 7.0:
```

```
    print(ph, "is acidic.") # 4 spaces!
```

```
####print
```

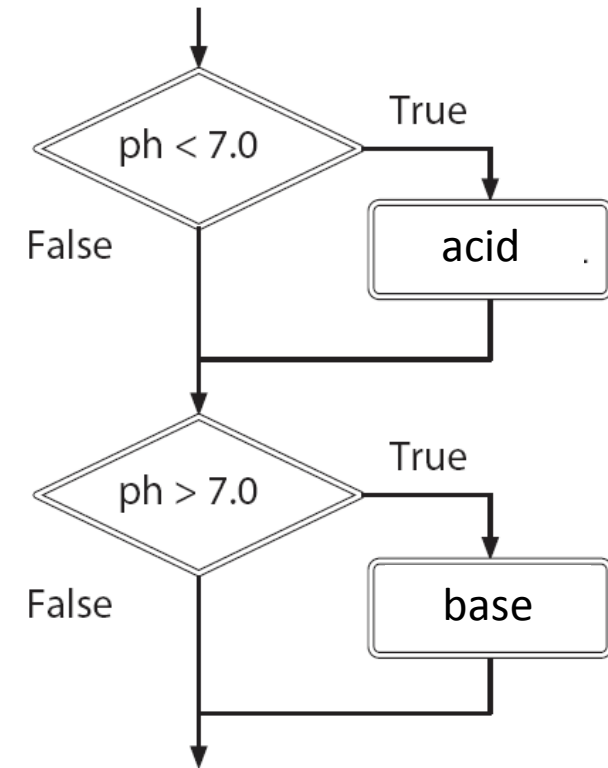
Boolean logic: Branching

```
if ph < 7.0:  
    print(ph, "is acidic.")          # 4 spaces!  
    print('Be careful with that acid!!!') # 4 spaces!
```

new addition to code

```
if ph < 7.0:  
    print(ph, " is acidic.")  
if ph > 7.0:  
    print(ph, " is basic.")
```

problem?



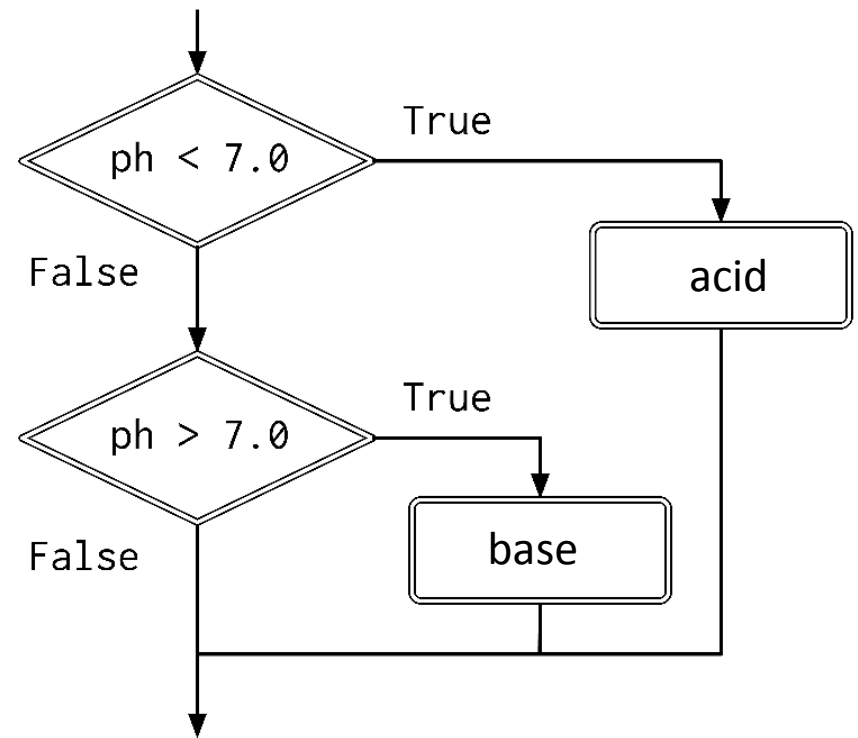
Boolean logic: Branching (elif)

```
# BETTER addition to code: elif (else if)
# if.. (then do this)  else if (then do this)
if ph < 7.0:
    print(ph, " is acidic.")
elif ph > 7.0:
    print(ph, " is basic.")
```

In shell it looks funky:

```
>>> if ph < 7.0: # would be 4 spaces to left
    print(ph, " is acidic.")
elif ph > 7.0:
    print(ph, " is basic.")
```

```
5.0  is acidic.
```



Boolean logic: Branching (elif)

```
# Changes to values within branches can change results
```

```
ph = float(input('Enter the pH level: '))
```

```
if ph < 7.0:
```

```
    ph = 8.0    # for whatever reason, it gets changed if < 7
```

```
if ph > 7.0:    # re-evaluated!
```

```
    print(ph, "is basic.")
```

```
>>> # run it
```

```
Enter the pH level: 6
```

```
8.0 is basic.
```

Boolean logic: Branching (elif)

```
# use elif to change values within branches
ph = float(input('Enter the pH level: '))
Enter the pH level: 6.0
if ph < 7.0:
    ph = 8.0    # for whatever reason, it gets changed if < 7
elif ph > 7.0:    # re-evaluated!
    print(ph, "is basic.")

# nothing gets printed.
```

Boolean logic: Branching (elif and else:)

```
# use else to cover ALL cases
```

```
compound = input('Enter the compound: ')
```

```
# Enter the compound: H2SO4
```

```
if compound == "H2O":
```

```
    print("Water")
```

```
elif compound == "NH3":
```

```
    print("Ammonia")
```

```
elif compound == "CH4":
```

```
    print("Methane")
```

```
else: print("Unknown compound")
```

```
^^^^ notice formatting!!!!
```

```
>>> compound = input('Enter the compound: ')
```

```
Enter the compound: H2SO4
```

```
>>> if compound == "H2O":
```

```
...     print("Water")
```

```
... elif compound == "NH3":
```

```
...     print("Ammonia")
```

```
... elif compound == "CH4":
```

```
...     print("Methane")
```

```
... else:
```

```
...     print("Unknown compound")
```

```
...
```

```
Unknown compound
```

```
>>> if compound == "H2O":
```

```
    print("Water")
```

```
    elif compound == "NH3":
```

```
        print("Ammonia")
```

```
    elif compound == "CH4":
```

```
        print("Methane")
```

```
SyntaxError: invalid syntax
```

```
>>> if compound == "H2O":
```

```
    print("Water")
```

```
elif compound == "NH3":
```

```
    print("Ammonia")
```

```
elif compound == "CH4":
```

```
    print("Methane")
```

```
else: print("Unknown compound")
```

```
Unknown compound
```

Boolean logic: Nested ifs

```
value = input('Enter the pH level: ')
if len(value) > 0:                                # sees if anything entered
    ph = float(value)                             # get value as float
    if ph < 7.0:
        print(ph, "is acidic.")
    elif ph > 7.0:
        print(ph, "is basic.")
    else:
        print(ph, "is neutral.")
else:
    print("No pH value was given!")
```


Boolean logic: Logic power

Risk for heart disease based on BMI and Age

```
if age < 45:
    if bmi < 22.0:
        risk = 'low'
    else:
        risk = 'medium'
else:
    if bmi < 22.0:
        risk = 'medium'
    else:
        risk = 'high'
```

		Age	
		<45	≥45
BMI	<22.0	Low	Medium
	≥22.0	Medium	High

Boolean logic: Logic power

Risk for heart disease based on BMI and Age

```
young = age < 45 # boolean
slim = bmi < 22.0 # boolean
if young:
    if slim:
        risk = 'low'
    else:
        risk = 'medium'
else:
    if slim:
        risk = 'medium'
    else:
        risk = 'high'
```

		Age	
		<45	≥45
BMI	<22.0	Low	Medium
	≥22.0	Medium	High

Boolean logic: Logic power

Risk for heart disease based on BMI and Age

```
young = age < 45 # boolean
slim = bmi < 22.0 # boolean
if young and slim:
    risk = 'low'
elif young and not slim:
    risk = 'medium'
elif not young and slim:
    risk = 'medium'
elif not young and not slim:
    risk = 'high'
```

		Age	
		<45	≥45
BMI	<22.0	Low	Medium
	≥22.0	Medium	High

how can we improve it more?

Boolean logic: Logic power

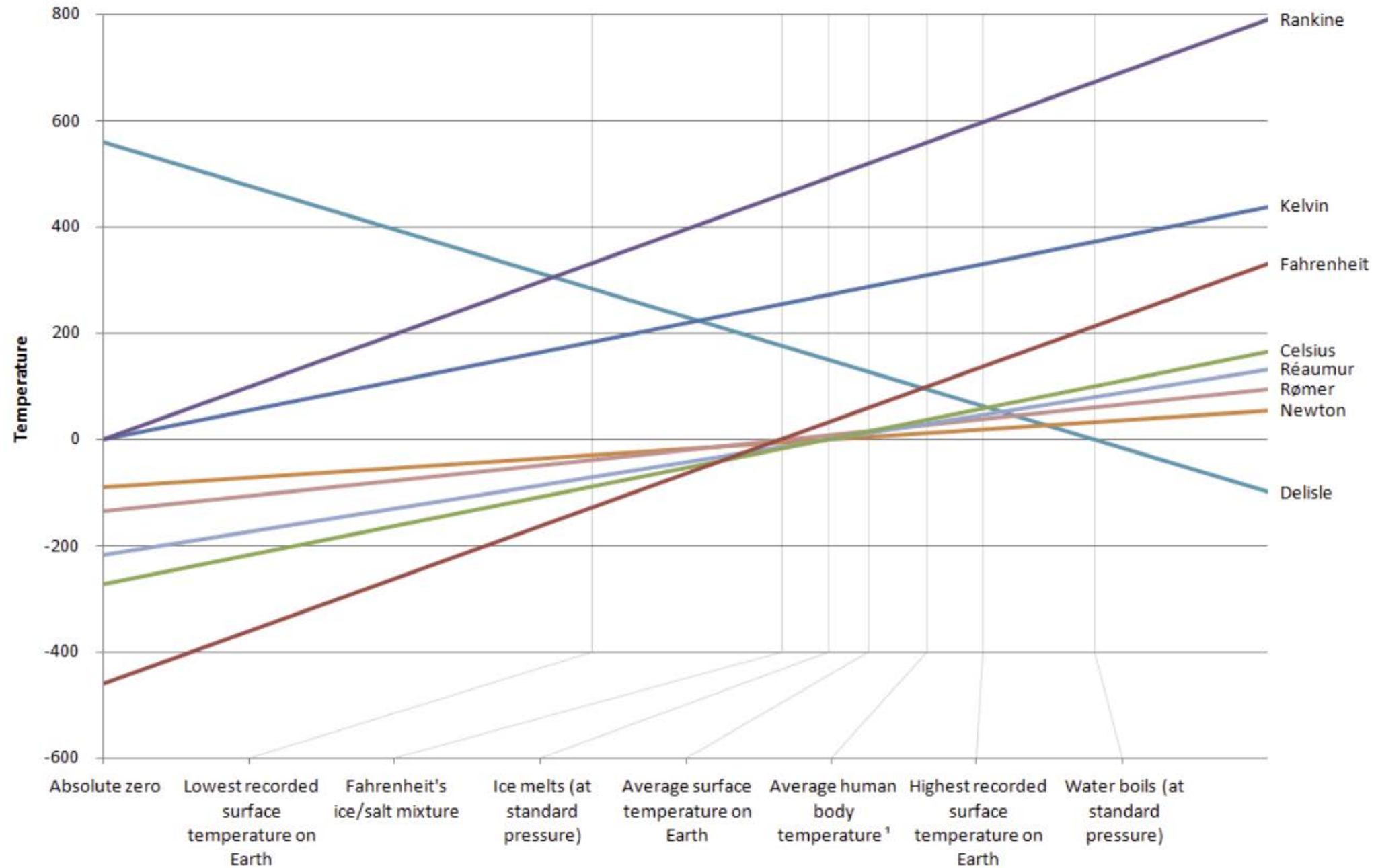
Risk for heart disease based on BMI and Age

```
young = age < 45 # boolean
slim = bmi < 22.0 # boolean
if young and slim:
    risk = 'low'
elif (young and not slim) or (not young and slim):
    risk = 'medium'
elif not young and not slim:
    risk = 'high'
```

		Age	
		<45	≥45
BMI	<22.0	Low	Medium
	≥22.0	Medium	High

Combine based on risk outcome

Comparison of temperature scales



Homework 5 due before class Wednesday

Do 5.6 Exercises 4, 5, **8**, 9

Be sure to read the hint in 8a. You can convert into 8 possible temperature scales from 7 other scales. How many possible conversions would you have to make?

- so always convert to one scale first
- an example of problem simplification