A Few Questions

While loop.

Question: What is a good example of using a while loop where a for loop would make it much more difficult.

```
 1: # Example of why to use a "while" - reverse a string
 2:
 3: vIn = "abcd"
 4: vOut = ""
 5: i = len(vIn)
 6: while ( i > 0 ):
 7:     i = i - 1
 8:     vOut = vOut + vIn[i]
 9:
10: print ( "vOut = ->{}<-".format(vOut ) )
```

or with a different way of offsetting to 0

```
 1: # Example of why to use a "while" - reverse a string
 2:
 3: vIn = "abcd"
 4: vOut = ""
 5: i = len(vIn)-1
 6: while ( i >= 0 ):
 7:     vOut = vOut + vIn[i]
 8:     i = i - 1
 9:
10: print ( "vOut = ->{}<-".format(vOut ) )
```

Formatting.

Question: Is the `{}` a dictionary in the format statement.

```
>>> a = 1.2345
>>> print ( "X Decimal Places {}".format(a) )
>>> print ( "2 Decimal Places {:.2f}".format(a) )
>>> print ( "In Order {} second {} third {}".format( "1st", 2, "last" ) )
```

A better if-else example

Personal Income Tax Calculator. This is not all of taxes. This is just in the case where you have a job and you get a paycheck. Let's say you have $88,000.00 a year in pay. What do the "tax" calculations mean.

First there is a standard deduction. For 2021 this is:

Amount	Description
$12,550	single taxpayers.
$12,550	married taxpayers filing separately.
$18,800	heads of households.
$25,100	married taxpayers filing jointly.

This looks like a table we can turn into an "if"/"else" in python.

```
 1: print ( "1 for single taxpayers. " )
 2: print ( "2 for married taxpayers filing separately.  " )
 3: print ( "3 for heads of households.  " )
 4: print ( "4 for married taxpayers filing jointly. " )
 5:
 6: maritul_status = input()
 7:
 8: standard_decuction = 12550
 9: if maritul_status == "1" :
10:     standard_decuction = 12550
11: elif maritul_status == "2" :
12:     standard_decuction = 12550
13: elif maritul_status == "3" :
14:     standard_decuction = 18800
15: elif maritul_status == "4" :
16:     standard_decuction = 25100
17: else:
18:     print ( "invalid input, should be 1, 2, 3, or 4" )
19:
20: print ( "Standard Deduction = {}".format( standard_decuction ) )
```

The standard deduction is take off of your income before you calculate your taxes. So the $88,000.00 minus $25,100 is: $62900.

This is the amount we use in the 2nd tax calculation.

If you search for "tax tables 2021" you get:

Tax Rate	Taxable Income Bracket	Tax Owed
10%	$0 to $14,200	10% of taxable income
12%	$14,201 to $54,200	$1,420 plus 12% of the amount over $14,200
22%	$54,201 to $86,350	$6,220 plus 22% of the amount over $54,200
24%	$86,351 to $164,900	$13,293 plus 24% of the amount over $86,350

What this table means is that you pay 10% on the first $14,200. Then take that off then pay 12% on the next chunk of money.

Let's implement that.

```
 1: print ( "What is your per year income" )
 2: income_str = input()
 3: income = float(income_str)
 4:
 5: print ( "1 for single taxpayers. " )
 6: print ( "2 for married taxpayers filing separately.  " )
 7: print ( "3 for heads of households.  " )
 8: print ( "4 for married taxpayers filing jointly. " )
 9:
10: maritul_status = input()
11:
12: tax = 0
13: standard_decuction = 12550
14:
15: if maritul_status == "1" or maritul_status == "2" : # Single, Married File Separat
16:     standard_decuction = 12550
17:     income = income - standard_decuction
18:
19:     tax = (10/100) * income
20:     if income >= 9951:
21:         tax = tax + (2/100) * ( income - 14200 )
22:     if income >= 40526:
23:         tax = tax + (10/100) * ( income - 54200 )
24:     if income >= 86376:
25:         tax = tax + (2/100) * ( income - 86350 )
26:     if income >= 164926:
27:         tax = tax + (10/100) * ( income - 164925 )
28:     if income >= 209426:
29:         tax = tax + (3/100) * ( income - 209425 )
30:     if income >= 523601:
31:         tax = tax + (2/100) * ( income - 523600 )
32:
33:
34:
35:
36:
```

```
37: elif maritul_status == "3" :        # Head of Household
38:     standard_decuction = 18800
39:
40:     income = income - standard_decuction
41:
42:     tax = (10/100) * income
43:     if income >= 14201:
44:         tax = tax + (2/100) * (income - 14200)
45:     if income >= 54201:
46:         tax = tax + (10/100) * (income - 54200)
47:     if income >= 86351:
48:         tax = tax + (2/100) * (income - 86350)
49:     if income >= 164901:
50:         tax = tax + (10/100) * (income - 164900)
51:     if income >= 209401:
52:         tax = tax + (3/100) * (income - 209400)
53:     if income >= 523601:
54:         tax = tax + (2/100) * (income - 523501)
55:
56: elif maritul_status == "4" :
57:     standard_decuction = 25100
58:
59:     income = income - standard_decuction
60:
61:     tax = (10/100) * income
62:     if income >= 14201:
63:         tax = tax + (2/100) * (income - 14200)
64:     if income >= 54201:
65:         tax = tax + (10/100) * (income - 54200)
66:     if income >= 86351:
67:         tax = tax + (2/100) * (income - 86350)
68:     if income >= 164901:
69:         tax = tax + (10/100) * (income - 164900)
70:     if income >= 209401:
71:         tax = tax + (3/100) * (income - 209400)
72:     if income >= 523601:
73:         tax = tax + (2/100) * (income - 523501)
74:
75: else:
76:     print ( "invalid input, should be 1, 2, 3, or 4" )
77:
78: print ( "Standard Deduction = {}".format( standard_decuction ) )
79:
80: print ( "total tax for the year = {:.2f}".format(tax) )
81:
82: print ( "What is monthly witholding" )
83: withold_str = input()
84: withold = float(withold_str)
85:
86:
87:
```

```
88: owe = tax - ( 12 * withold )
89: if owe < 0 :
90:     print ( "You get a tax refund of {:.2f}".format(-owe) )
91: elif owe == 0:
92:     print ( "You don't owe any and you don't get a refund" )
93: elif owe > 0:
94:     print ( "Send the IRS: {:.2f}".format(owe) )
```

Import

Most of the time when you build a program you have multiple files. Python deals with this with the "import" statement.

The 2 most commonly used formats are:

```
import file
```

and

```
from file imort function
```

Let's try it (this is the all in one directory version):

```
1: import jane
2:
3: jane.janefunc()
```

```
1: def janefunc():
2:     print ( "jane" )
```

Multiple Directories Version:

```
1: imort x.bob
2:
3: x.bob.bobfunc()
```

```
1: def bobfunc():
2:     print ( "bob" )
```

or we can just import a single function

```
1: from x.bob import bobfunc
2:
3: bobfunc()
```

Copyright

Copyright © University of Wyoming, 2021.