Introduction to Programming: Python

Formatting Strings with Format Modifiers

The syntax for formatting strings is a little cryptic at times, but once you get used to it, it's not too bad, and it is quite useful. The method that Python uses to format strings is similar to that of Java, C and C++.

To format a string we separate the way our variables are to be displayed from the actual values, then the values get inserted the way we want them. e.g.

```
name = "Bob"
bill = 45.1
print ("Hi %s, your bill comes to $%.2f" % (name, bill))
```

String Formatting Syntax:
<format string> % <values>

<format string> - in this example "Hi %s, your bill comes to $%.2f". Most of the string is displayed exactly as we type it. %s and %.2f are called "conversion specifiers". When we run this line the conversion specifiers are replaced by values from the value list (in order) and displayed as specified.

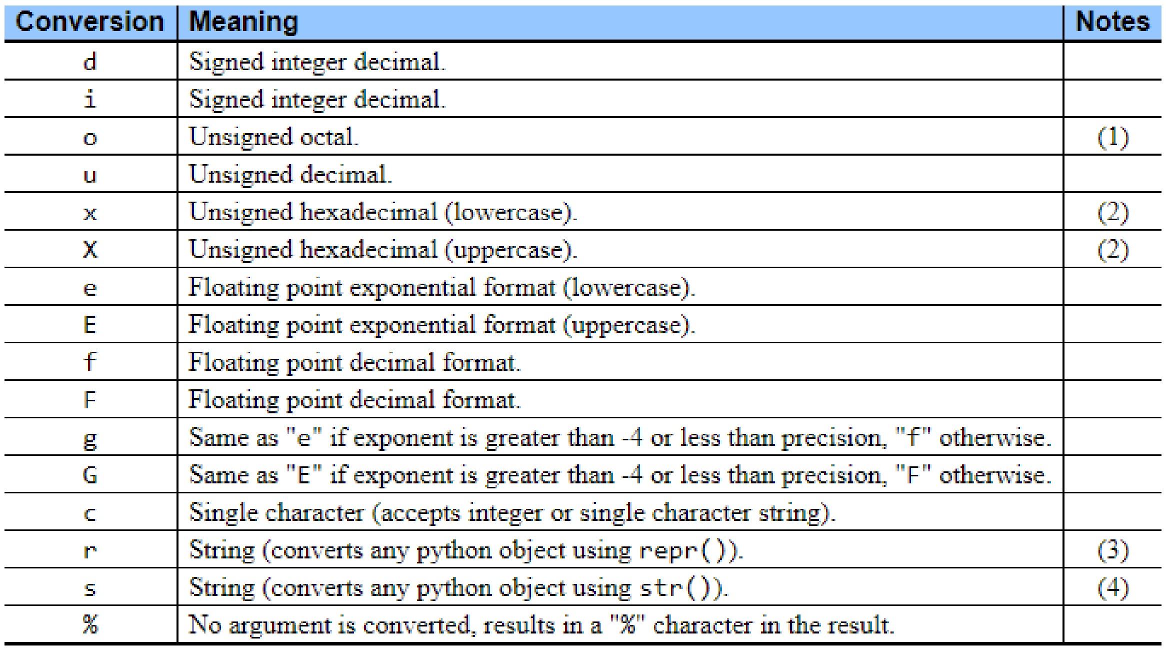
<conversion specifiers> - A conversion specifier (format modifiers) always starts with a "%", and you need to specify what type of value you want to display.

%s - string   %i - integer   %f - float

You can also specify options about how you want the value displayed, like

- how big of a field to display the value in. - width modifier
- how many decimal places for float values
- whether the value should be left-aligned or right-aligned.

Field spacing – The field spacing is the total number of spaces used to display your value, including the ones used by your value. E.g.



Conversion	Meaning	Notes
d	Signed integer decimal.	
i	Signed integer decimal.	
o	Unsigned octal.	(1)
u	Unsigned decimal.	
x	Unsigned hexadecimal (lowercase).	(2)
X	Unsigned hexadecimal (uppercase).	(2)
e	Floating point exponential format (lowercase).	
E	Floating point exponential format (uppercase).	
f	Floating point decimal format.	
F	Floating point decimal format.	
g	Same as "e" if exponent is greater than -4 or less than precision, "f" otherwise.	
G	Same as "E" if exponent is greater than -4 or less than precision, "F" otherwise.	
c	Single character (accepts integer or single character string).	
r	String (converts any python object using `repr()`).	(3)
s	String (converts any python object using `str()`).	(4)
%	No argument is converted, results in a "%" character in the result.	

```
a = 12.3456
n = 300
s = "Joe"
%10f        ---12.3456  - note the decimal takes up one space
%10s        -----Joe
%10i        -------300
%10.2f-----12.35  - note that it will round to 2 decimals
%-10.2f     12.35-----  - "-" will left align
%2i         300               - making the field smaller than the
                                number of characters will not chop off
                                characters
```

Exercise #5

1) A job pays $10.25 an hour. Write a program that asks the user to input how many hours they worked and calculate their pay. Your output should be as follows:

A student works for ? hours at $10.25/hour. Their pay is $?.

Note: All money is to two decimal places.

2) Get the coordinates for two points (four inputs) from the user and compute the distance between the two points. Output your answer rounded to two decimal places.

3) Write a program that asks the user to enter five test marks. If the test was out of 60, display the average for each test as a percentage (to one decimal place) and find the average of all the tests.

For example:

```
Mark        Percent
55           91.7
41           68.3
17           28.3
35           58.3
60          100.0
```

The average for the test is: 66.7%
Note: Mark and percent are both right justified.

4) Write a simple cash register program that asks for the product name and cost for three products. The program should then calculate HST (13%) on the product and add the tax to the total price. An example of the output is as follows:

```
TAB Gift Shop Receipt
------------------------------
glove             12.89
```

```
toque          18.99
scarf          20.00
-------------------------------
HST (13%)       6.74
-------------------------------
TOTAL          58.62
```

Note: Product Name is left justified and no more than 10 characters (do not need to check, just won't be)
Cost, HST and Total are all right justified.