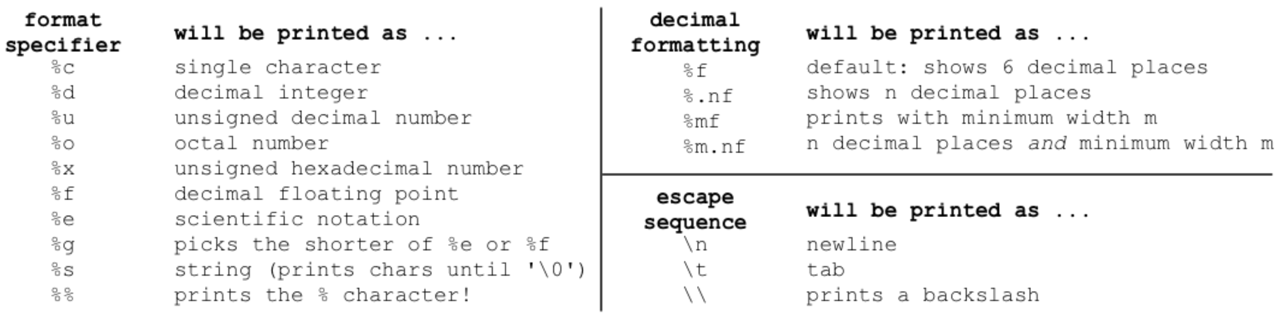
Printing redux

coursera.org/learn/programming-fundamentals/supplement/4F2uM/printing-redux

As we learned in the lesson on printing, C supports printing formatted information via the function printf. Now that we have multiple types, we can explore the various format specifiers, which allow us to print variables of a variety of types. The figure below shows the most common specifiers. You should not try to memorize these (nor really anything in learning to program)—rather, you should know how/where to look up what you need.



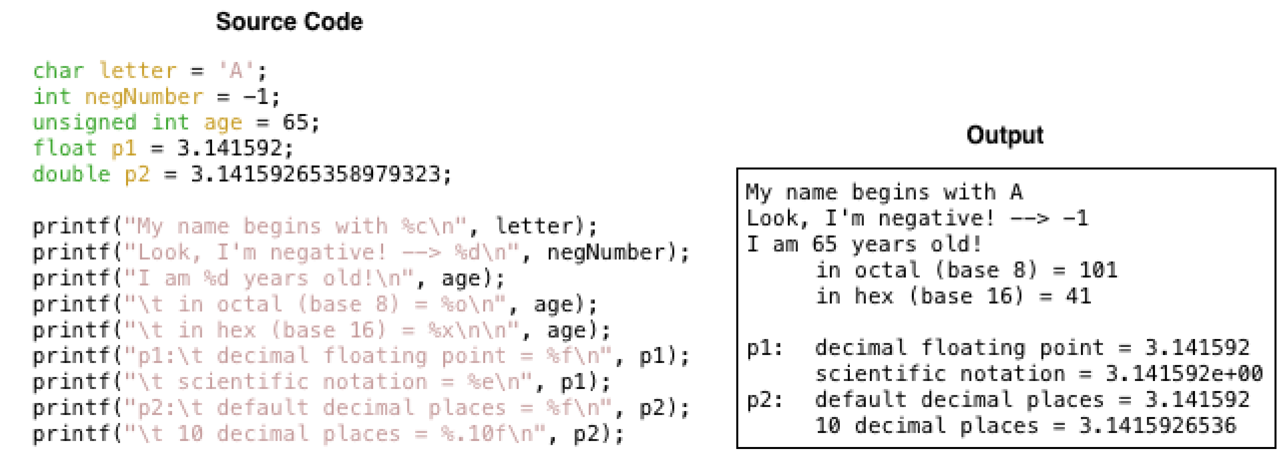
format specifier	will be printed as ...
%c	single character
%d	decimal integer
%u	unsigned decimal number
%o	octal number
%x	unsigned hexadecimal number
%f	decimal floating point
%e	scientific notation
%g	picks the shorter of %e or %f
%s	string (prints chars until '\0')
%%	prints the % character!

decimal formatting	will be printed as ...
%f	default: shows 6 decimal places
%.nf	shows n decimal places
%mf	prints with minimum width m
%m.nf	n decimal places *and* minimum width m

escape sequence	will be printed as ...
\n	newline
\t	tab
\\	prints a backslash

As you use the most common ones often, they will come to you naturally. You can find more format specifiers, as well as more information about these format specifiers in the man page for printf (we will talk about man pages more in the next course).

The figure below shows some examples of these format specifiers being used. Here, the code (shown on the left) declares a few variables, and prints them out using the format specifiers described in the figure above. Note that while we have already discussed hexadecimal (base 16), this example also makes reference to octal—which is base 8.



Source Code

```c
char letter = 'A';
int negNumber = -1;
unsigned int age = 65;
float p1 = 3.141592;
double p2 = 3.14159265358979323;

printf("My name begins with %c\n", letter);
printf("Look, I'm negative! --> %d\n", negNumber);
printf("I am %d years old!\n", age);
printf("\t in octal (base 8) = %o\n", age);
printf("\t in hex (base 16) = %x\n\n", age);
printf("p1:\t decimal floating point = %f\n", p1);
printf("\t scientific notation = %e\n", p1);
printf("p2:\t default decimal places = %f\n", p2);
printf("\t 10 decimal places = %.10f\n", p2);
```

Output

```
My name begins with A
Look, I'm negative! --> -1
I am 65 years old!
     in octal (base 8) = 101
     in hex (base 16) = 41

p1:  decimal floating point = 3.141592
     scientific notation = 3.141592e+00
p2:  default decimal places = 3.141592
     10 decimal places = 3.1415926536
```