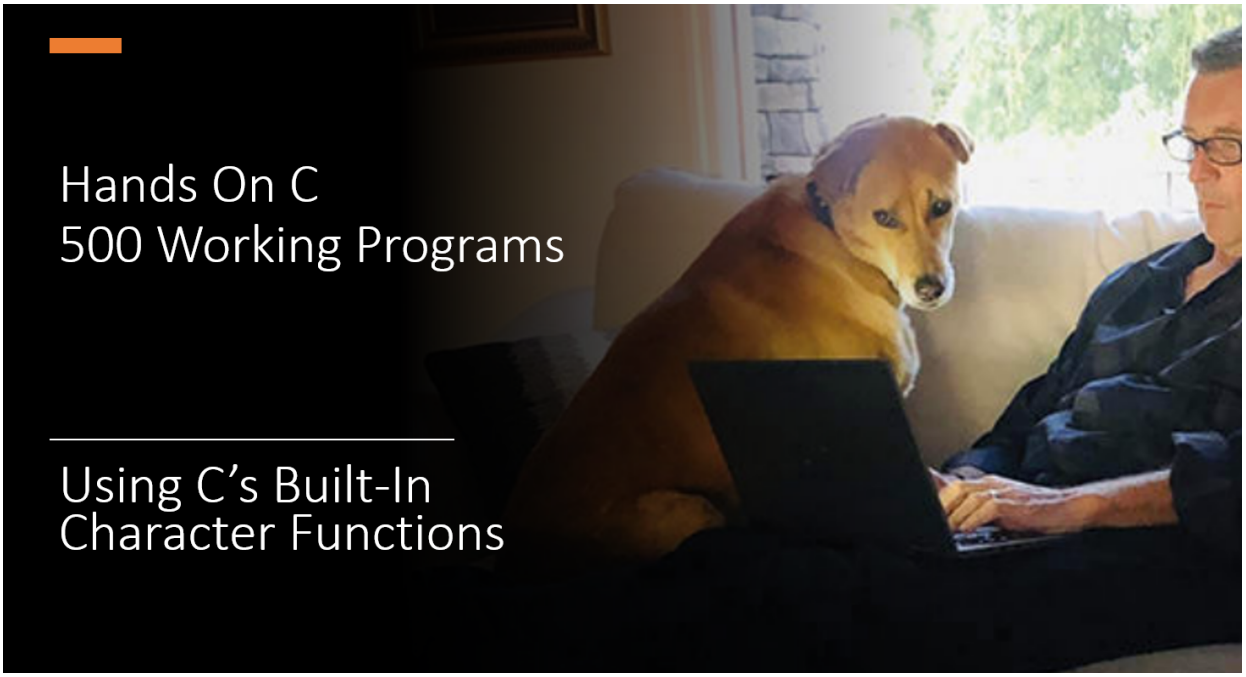

Determining If a Character is a Letter

In [1]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isalpha(string[i]))
       printf("%c is a letter\n", string[i]);
}
```

```
A is a letter
b is a letter
c is a letter
```

Determining if a Character is a Letter or Number

In [2]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isalnum(string[i]))                    // isalpha in some compilers
       printf("%c is alphanumeric\n", string[i]);
}
```

```
A is alphanumeric
b is alphanumeric
c is alphanumeric
1 is alphanumeric
2 is alphanumeric
3 is alphanumeric
```

Determining if a Character is a Digit

In [3]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isdigit(string[i]))
       printf("%c is a digit\n", string[i]);
}
```

```
1 is a digit
2 is a digit
3 is a digit
```

Determining if a Character is an ASCII Value

In [4]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123";

   for (int i = 0; string[i]; i++)
     if (__isascii(string[i]))              // isacii() in some compilers
       printf("%c is ASCII\n", string[i]);

   printf("Letter (value %d) %s an ASCII character\n", 255, (__isascii(255)) ? "i
}
```

```
A is ASCII
b is ASCII
c is ASCII
1 is ASCII
2 is ASCII
3 is ASCII
Letter (value 255) is not an ASCII character
```

Determining if a Character is a Control Character

In [5]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (iscntrl(string[i]))
       printf("%d is a control character\n", (int) string[i]);
}
```

```
9 is a control character
10 is a control character
```

Determining if a Letter is a Printable Character

In [6]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isgraph(string[i]))
       printf("%c is printable\n", string[i]);
}
```

```
A is printable
b is printable
c is printable
1 is printable
2 is printable
3 is printable
```

Determining if a Character is Upper or Lowercase

In [7]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isupper(string[i]))
       printf("%c is uppercase\n", string[i]);
     else if (islower(string[i]))
       printf("%c is lowercase\n", string[i]);
}
```

```
A is uppercase
b is lowercase
c is lowercase
```

Determing if a Character is Punctuation

In [8]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123.,;\t\n";

   for (int i = 0; string[i]; i++)
     if (ispunct(string[i]))
       printf("%c is punctuation\n", string[i]);
}
```

```
. is punctuation
, is punctuation
; is punctuation
```

Determing if a Character is Whitespace

In [9]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123 \t\n";

   for (int i = 0; string[i]; i++)
     if (isspace(string[i]))
       printf("%d is whitespace\n", (int) string[i]);
}
```

```
32 is whitespace
9 is whitespace
10 is whitespace
```

Determining if a Letter is a Hexadecimal Value

In [10]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";

   for (int i = 0; string[i]; i++)
     if (isxdigit(string[i]))
       printf("%c is a hexadecimal\n", string[i]);
}
```

```
A is a hexadecimal
b is a hexadecimal
c is a hexadecimal
1 is a hexadecimal
2 is a hexadecimal
3 is a hexadecimal
```

Converting a Character to Uppercase or Lowercase

In [11]:
```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
   char string[] = "Abc123\t\n";
   char upper[64];
   char lower[64];
   int i = 0;

   for (i = 0; string[i]; i++)
     {
       lower[i] = tolower(string[i]);
       upper[i] = toupper(string[i]);
     }

   lower[i] = '\0';
   upper[i] = '\0';

   printf("Upper %s\nLower %s", upper, lower);
}
```

```
Upper ABC123

Lower abc123
```

What You will Learn Next

```
As you have learned, to change a parameter's value within a C function, you
must pass a pointer (the address) to the variable in memory. In the next
lesson, you will examine C pointer operations in detail.

int value = 0;
int *address = &value;
```