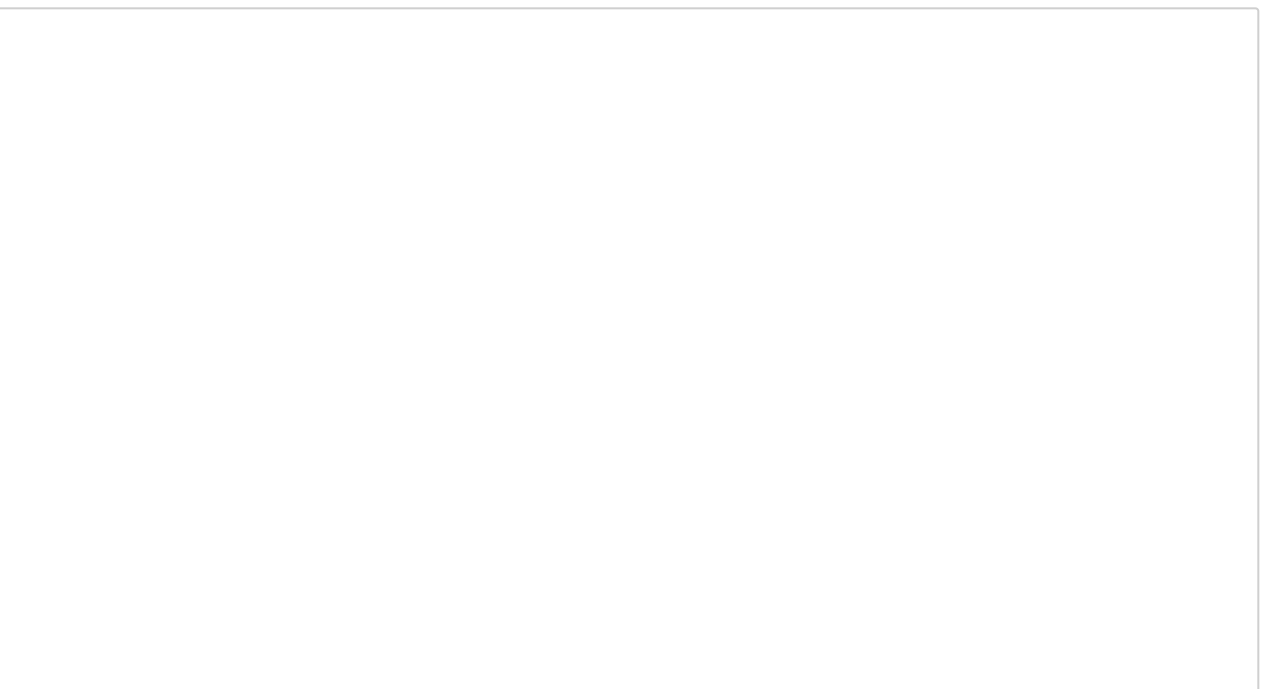
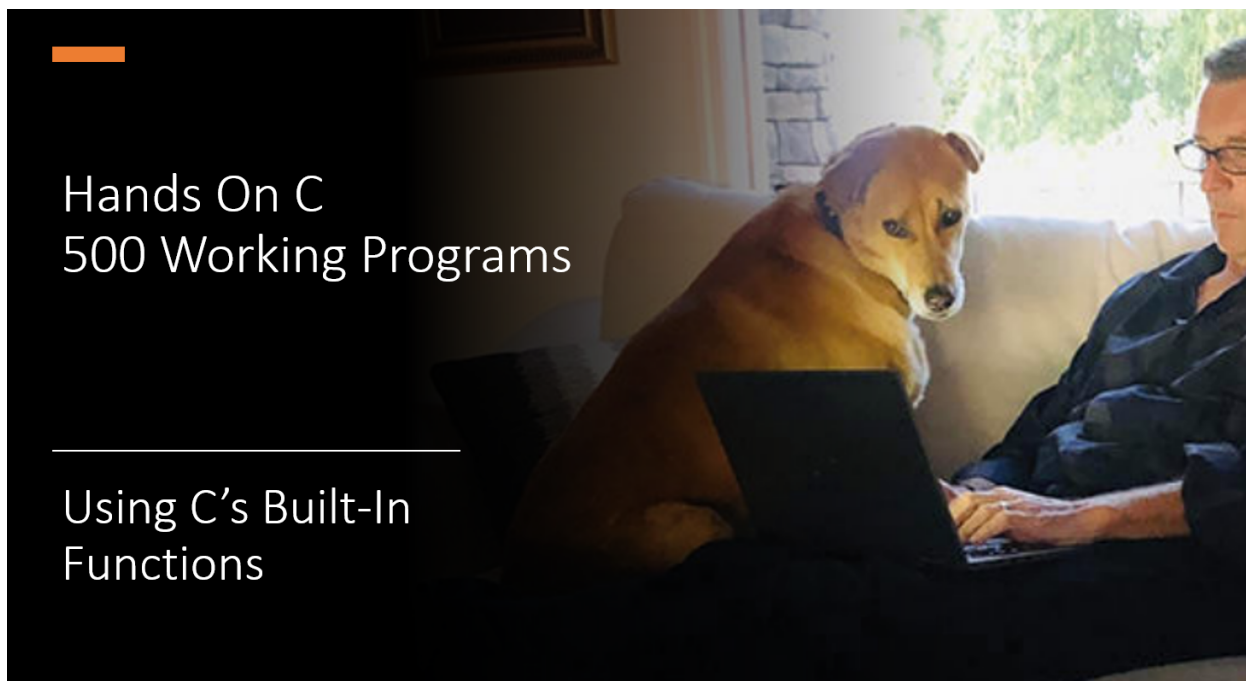




Revisiting Functions

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

int main(void)
{
    printf("Hello, world");
}
```

Hello, world

Getting the Absolute Value of an Integer Expression

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

int main(void)
{
    printf("The absolute value of %d is %d\n", 1, abs(1));
    printf("The absolute value of %d is %d\n", -1, abs(-1));
    printf("The absolute value of %d is %d\n", 0, abs(0));
    printf("The absolute value of %d is %d\n", 3-5, abs(3-5));
}
```

```
The absolute value of 1 is 1
The absolute value of -1 is 1
The absolute value of 0 is 0
The absolute value of -2 is 2
```

Math Functions in math.h

Math Function	Purpose
sin(radians)	Returns the sine of angle in radians
cos(radians)	Returns the cosine of an angle in radians
tan(radians)	Returns the tangent of an angle in radians
asin(radians)	Returns the arcsine of an angle in radians
atan(radians)	Returns the arctangent of an angle in radians
acos(radians)	Returns the arccosine of an angle in radians
cosh(radians)	Returns the hyperbolic cosine of an angle in radians
sinh(radians)	Returns the hyperbolic sine of an angle in radians
tanh(radians)	Returns the hyperbolic tangent of an angle in radians
cabs(complex)	Returns the absolute value of a complex number

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

int main(void)
{
    float radians = 0.5;

    printf("sine of %f is %f\n", radians, sin(0.5));
    printf("cosine of %f is %f\n", radians, cos(0.5));
    printf("tangent of %f is %f\n", radians, tan(0.5));
}
```

```
sine of 0.500000 is 0.479426
cosine of 0.500000 is 0.877583
tangent of 0.500000 is 0.546302
```

Performing an Integer Division

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

int main(void)
{
    div_t result;

    result = div(20, 7);

    printf("20 divided by 7 is %d remainder %d\n", result.quot, result.rem);
}
```

20 divided by 7 is 2 remainder 6

Determining the Value of e to the x

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

int main(void)
{
    printf("The result of e(%f) is %f\n", 0.1, exp(0.1));
}
```

The result of e(0.100000) is 1.105171

Determining the Absolute Value of a Floating-Point Number

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

int main(void)
{
    double value = -3.14;

    printf("Absolute value of %f is %f\n", value, fabs(value) );
}
```

Absolute value of -3.140000 is 3.140000

Determining the Remainder of a Floating-Point Division

20.5 fmod 3.0 => 2.5 (6.0 Remainder 2.5)

```
In [7]: #include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main(void)
{
    double numerator = 20.5;
    double denominator = 3.0;

    printf("%f mod %f is %f", numerator, denominator, fmod(numerator, denominator))
}
```

```
/tmp/tmp2wkqijm.out: symbol lookup error: /tmp/tmpctobl2mh.out: undefined symb
ol: fmod
[C kernel] Executable exited with code 127
```

Calculating x times 2 raised to e

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

int main(void)
{
    double x = 1.25;
    double mantissa;
    int exponent;

    mantissa = frexp(x, &exponent);
    printf("x is %f the mantissa is %f the exponent is %d\n", x, mantissa, exponent);
    double value = mantissa * pow(2.0, exponent);
    printf("The value is %f\n", value);
}
```

x is 1.250000 the mantissa is 0.625000 the exponent is 1

/tmp/tmp2wkqijm.out: symbol lookup error: /tmp/tmpbltiext_.out: undefined symbol: pow
[C kernel] Executable exited with code 127

Determining the Natural Log of an Expression

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

int main(void)
{
    printf("The natural log of 512.0 is %f", log(512.0));
}
```

The natural log of 512.0 is 6.238325

Determining the Result of Log10x

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

int main(void)
{
    printf("The log10x of 10 is %f\n", log10(10.0));
    printf("The log10x of 100 is %f\n", log10(100.0));
    printf("The log10x of 1000 is %f\n", log10(1000.0));
}
```

```
The log10x of 10 is 1.000000
The log10x of 100 is 2.000000
The log10x of 1000 is 3.000000
```

Breaking a value of Type Double into Whole and Real Parts

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

int main(void)
{
    double value = 5.4321;
    double whole, fraction;

    fraction = modf(value, &whole);

    printf("Value %f Whole %f Fraction %f\n", value, whole, fraction);
}
```

Value 5.432100 Whole 5.000000 Fraction 0.432100

Calculating the Result of 10 Raised to x

```
In [12]: #include <stdio.h>
#include <math.h>

int main(void)
{
    printf("10 raised to -1 is %f\n", pow10(-1));
    printf("10 raised to 0 is %f\n", pow10(0));
    printf("10 raised to 1 is %f\n", pow10(1));
    printf("10 raised to 2 is %f\n", pow10(2));
}
```

```
/tmp/tmpumdb2rto.c: In function 'main':
/tmp/tmpumdb2rto.c:6:37: warning: implicit declaration of function 'pow10'; did
you mean 'powl'? [-Wimplicit-function-declaration]
    printf("10 raised to -1 is %f\n", pow10(-1));
                                   ^~~~~~
                                   powl
/tmp/tmpumdb2rto.c:6:31: warning: format '%f' expects argument of type 'double',
but argument 2 has type 'int' [-Wformat=]
    printf("10 raised to -1 is %f\n", pow10(-1));
                                   ~^      ~~~~~~
                                   %d
/tmp/tmpumdb2rto.c:7:30: warning: format '%f' expects argument of type 'double',
but argument 2 has type 'int' [-Wformat=]
    printf("10 raised to 0 is %f\n", pow10(0));
                                   ~^      ~~~~~~
                                   %d
/tmp/tmpumdb2rto.c:8:30: warning: format '%f' expects argument of type 'double',
but argument 2 has type 'int' [-Wformat=]
    printf("10 raised to 1 is %f\n", pow10(1));
                                   ~^      ~~~~~~
                                   %d
/tmp/tmpumdb2rto.c:9:30: warning: format '%f' expects argument of type 'double',
but argument 2 has type 'int' [-Wformat=]
    printf("10 raised to 2 is %f\n", pow10(2));
                                   ~^      ~~~~~~
                                   %d
/tmp/tmp2wkqijm.out: symbol lookup error: /tmp/tmpjfoj_02i.out: undefined symbol:
pow10
[C kernel] Executable exited with code 127
```

Generating a Random Number

```
In [13]: #include <stdio.h>
#include <stdlib.h>

int main(void)
{
    for (int i = 0; i < 5; i++)
        printf("%d\n", rand());
}
```

```
1804289383
846930886
1681692777
1714636915
1957747793
```

```
In [14]: #include <stdio.h>
#include <stdlib.h>

int main(void)
{
    srand(0);

    for (int i = 0; i < 5; i++)
        printf("%d\n", rand());
}
```

```
1804289383
846930886
1681692777
1714636915
1957747793
```

```
In [15]: #include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    time_t systemTime = 0;    // seed with seconds since 1/1/1970
    srand(time(&systemTime));

    for (int i = 0; i < 5; i++)
        printf("%d\n", rand());
}
```

```
1645315998
1366626377
1133947694
14122165
1939924854
```

Calculating the Square Root

```
In [16]: #include <stdio.h>
#include <math.h>

int main(void)
{
    printf("Square root of 25 is %f\n", sqrt(25));
    printf("Square root of 100 is %f\n", sqrt(100));
    printf("Square root of 3 is %f\n", sqrt(3));
}
```

```
Square root of 25 is 5.000000
Square root of 100 is 10.000000
Square root of 3 is 1.732051
```

Getting Time in Seconds Since 1/1/1970

```
In [17]: #include <stdio.h>
#include <time.h>

int main(void)
{
    time_t seconds;
    seconds = time(0);

    printf("Seconds since 1/1/1970 %ld", seconds);
}
```

```
Seconds since 1/1/1970 1612135626
```

Displaying the Current Date

```
In [18]: #include <stdio.h>
#include <time.h>

int main(void)
{
    time_t seconds;
    seconds = time(0);

    printf("Current date and time %s\n", ctime(&seconds));
}
```

Current date and time Sun Jan 31 23:27:06 2021

Delaying a Specific Number of Seconds

```
In [*]: #include <stdio.h>
#include <time.h>

int main(void)
{
    time_t startTime, endTime;

    printf("Starting delay\n");
    startTime = time(0);

    do {
        endTime = time(0);
    } while (endTime - startTime < 5);

    printf("Done.");
}
```

Starting delay

Getting Date and Time Components

```
struct tm
{
    int tm_sec;
    int tm_min;
    int tm_hour;
    int tm_mday;
    int tm_mon;    // 0 thru 11
    int tm_year;   // year - 1900
    int tm_wday;   // Sun 0 - Sat 6
    int tm_yday;   // 1 - 365 day of year
    int tm_isdst;  // 1 if daylight savings 0 otherwise
}
```

```
In [*]: #include <stdio.h>
#include <time.h>

int main(void)
{
    struct tm *currentDate;
    time_t seconds;

    time(&seconds);

    currentDate = localtime(&seconds);
    printf("Date: %d-%d-%d\n", currentDate->tm_mon+1, currentDate->tm_mday, currentDate->tm_year+1900);
    printf("Time: %d:%02d\n", currentDate->tm_hour, currentDate->tm_min);
}
```

See the TZ environment entry to set timezone

```
In [*]: #include <stdio.h>
#include <time.h>

int main(void)
{
    struct tm *currentDate;
    time_t seconds;

    time(&seconds);

    int timezone = -7*60*60; // offset local difference from GMT
    seconds += timezone;

    currentDate = localtime(&seconds);
    printf("Date: %d-%d-%d\n", currentDate->tm_mon+1, currentDate->tm_mday, currentDate->tm_year+1900);
    printf("Time: %d:%02d\n", currentDate->tm_hour, currentDate->tm_min);
}
```

Greenwich Mean Time

```
In [*]: #include <stdio.h>
#include <time.h>

int main(void)
{
    struct tm *currentDate;
    time_t seconds;

    time(&seconds);

    currentDate = gmtime(&seconds);
    printf("Date: %d-%d-%d\n", currentDate->tm_mon+1, currentDate->tm_mday, current
    printf("Time: %d:%d\n", currentDate->tm_hour, currentDate->tm_min);
}
```

Converting a Date to Seconds Since Midnight 1/1/1970

```
In [*]: #include <stdio.h>
#include <time.h>

int main(void)
{
    struct tm currentDate;
    time_t seconds;

    currentDate.tm_mday = 22;
    currentDate.tm_mon = 1;
    currentDate.tm_year = 2021 - 1900;
    currentDate.tm_hour = 12;
    currentDate.tm_min = 30;

    seconds = mktime(&currentDate);
    printf("Seconds %ld\n", seconds);
}
```

What You will Learn Next

C programs make extensive use of character strings such as "Hello, world." You have learned that C stores strings as an array using the '\0' character to mark the last character in the string.

```
printf("Hello, world");
```