

Coding Basics

Programming for Good 1, Workshop

Today we will talk about coding basics.

What are coding basics?

Coding is the process of designing a program to perform a particular task using a computer.

https://en.wikipedia.org/wiki/Computer_programming

Basics means we are just getting started!

What is so great about coding?

Coding can be used to solve problems like:

- math calculations
- saving or changing data
- making decisions
- making a machine do an action in the real world

How does coding do all of this?

Code is used to make a program.

A program is a sequence of well-defined instructions in a programming language that a computer can do or understand.

Sequence means the instructions are in a certain order.

**Another name for a program is
algorithm.**

<https://en.wikipedia.org/wiki/Algorithm>

What is a computer?

A long time ago, before there were machines to do math, a computer was human! “Computer” was a job for a person that did difficult math problems that required a lot of steps.

Today a computer is a machine that follows instructions from an algorithm.

<https://en.wikipedia.org/wiki/Computer>

**Let's try an example of an algorithm with
my assistant...**

In this example, what was the algorithm?

Who was the coder?

What was the computer?

Another example:

Daisy: Tell me how old you are.

In this example, what was the algorithm?

Who was the coder?

What was the computer?

Another example:

Brownie: Tell me what day it is.

What was the algorithm?

Who was the coder?

What was the computer?

Another example:

Junior: Multiply 2 and 3. Then clap that many times.

Who was the coder?

What was the computer?

What was the algorithm?

How many steps are in the algorithm? Why is the sequence important?

In all these examples I made instructions and gave them to my computers to perform.

Let's do one more example. This time you are all the coders.

Program my sandwich-making robot to make a peanut butter and jelly sandwich.

You tell me the algorithm. I will write them down and give them to my robot.

Remember to use a sequence of well-defined instructions!

My robot understands English, but a modern machine computer does not!

A computer understands machine language.

A computer has many small electronic switches that can be off or on, or have a value of 0 or 1. And that is it! It is called binary.

This says “Hello World.” in binary:

```
01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100
```

It is very hard for people to type instructions into a computer in binary!

So computer scientists have created programming languages to turn easier-to-write instructions into binary.

Some of these languages are still hard to read and write though!

See <https://towardsdatascience.com/hello-world-not-so-easy-in-assembly-23da6644ff0d>.

```

1     .equ    LAST_RAM_WORD,    0x007FFFFC
2     .equ    JTAG_UART_BASE,   0x10001000
3     .equ    DATA_OFFSET,      0
4     .equ    STATUS_OFFSET,    4
5     .equ    WSPACE_MASK,     0xFFFF
6
7     .text
8     .global _start
9     .org    0x00000000
10
11    _start:
12        movia    sp, LAST_RAM_WORD
13        movi    r2, '\n'
14        call    PrintChar
15        movia    r2, MSG
16        call    PrintString
17    _end:
18        br     _end
19
20    PrintChar:
21        subi    sp, sp, 8
22        stw    r3, 4(sp)
23        stw    r4, 0(sp)
24        movia    r3, JTAG_UART_BASE
25    pc_loop:
26        ldwio   r4, STATUS_OFFSET(r3)
27        andhi  r4, r4, WSPACE_MASK
28        beq    r4, r0, pc_loop
29        stwio   r2, DATA_OFFSET(r3)
30        ldw    r3, 4(sp)
31        ldw    r4, 0(sp)
32        addi   sp, sp, 8
33        ret
34
35    PrintString:
36        subi   sp, sp, 12
37        stw    ra, 8(sp)
38        stw    r3, 4(sp)
39        stw    r2, 0(sp)
40        mov    r3, r2
41    ps_loop:
42        ldb    r2, 0(r3)
43        beq    r2, r0, end_ps_loop
44        call   PrintChar
45        addi   r3, r3, 1
46        br     ps_loop
47    end_ps_loop:
48        ldw    ra, 8(sp)
49        ldw    r3, 4(sp)
50        ldw    r2, 0(sp)
51        addi   sp, sp, 12
52        ret
53
54        .org    0x1000
55    MSG:   .asciz   "Hello World\n"
56        .end

```

In order to print “Hello World” in this example, you need to:

Set the memory location on the machine (line 1)

Tell where the program starts in the machine’s memory (line 8)

Tell the computer how to print a character (line 20)

Tell a computer how to print a word (line 35)

In short, you need to know a lot about how the machine actually works to make a program run!

Grace Hopper thought a computer language could be a lot easier to write if it looked like English.

She was a mathematician and computer scientist that worked with some of the first modern computers.



<https://en.wikipedia.org/wiki/Computer>, https://en.wikipedia.org/wiki/Grace_Hopper

She made the first compiler. A compiler turns a programming language that is easy for people to read and write into machine language.

A compiler makes it so that more people could code more quickly and with fewer problems.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. IDSAMPLE.  
ENVIRONMENT DIVISION.  
PROCEDURE DIVISION.  
    DISPLAY 'HELLO WORLD'.  
    STOP RUN.
```

<https://en.wikipedia.org/wiki/Computer>

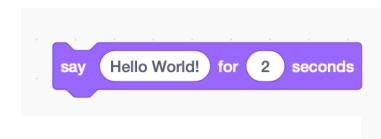
https://en.wikipedia.org/wiki/Grace_Hopper

<https://www.ibmmainframer.com/cobol-tutorial/cobol-hello-world/>

Since Grace's time, programming languages have become easier and easier to write.

```
#include <stdio.h>
void main(){
    printf("Hello World");
}
```

```
console.log('Hello World');
```



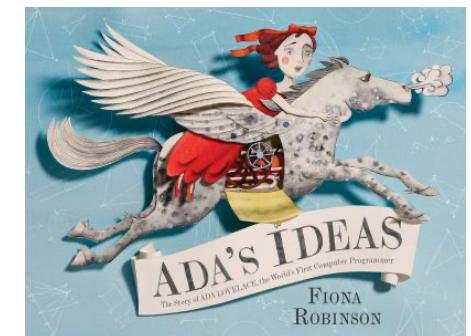
<https://towardsdatascience.com/hello-world-not-so-easy-in-assembly-23da6644ff0d>

There have been a lot of interesting women working on and with computers!



Ada Lovelace wrote one of the first computer programs before the first computer was ever built! She knew computers could be used for more than just difficult calculations. She thought anything with repeating patterns could be coded, like music and pictures. And in fact, that is how computers and cellphones are used today.

Read about Ada in *Ada's ideas : the story of Ada Lovelace, the world's first computer programmer.*



https://en.wikipedia.org/wiki/Ada_Lovelace

Dorothy Vaughan was a mathematician and computer for the government during World War 2. Human computers were needed for calculations for airplanes during the war.

She also worked at NASA as a computer. When NASA started using modern digital computers, Dorothy became a programmer. She worked on computer programs to launch satellites into space.

Read about Dorothy in *Hidden Figures*.



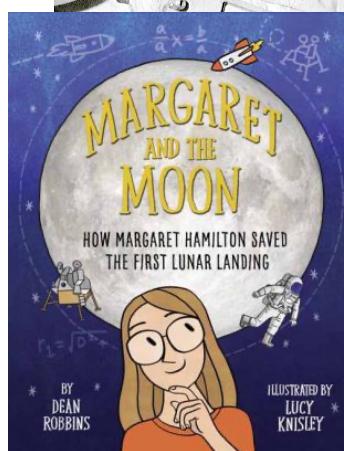
https://en.wikipedia.org/wiki/Dorothy_Vaughan

Margaret Hamilton was a computer scientist and software engineer.

She led a team that designed the programs for the moon landing.

Margaret convinced others that programming was a type of engineering that includes design, error detection and recovery, testing, and human-machine interactions.

Read about her in *Margaret and the moon : how Margaret Hamilton saved the first lunar landing.*



https://en.wikipedia.org/wiki/Dorothy_Vaughan

That's the end!

See you in a couple weeks for part 2!

A final thought from Grace Hopper...



"A ship in port is safe, but that is not what ships are for. Sail out to sea and do new things."

-Grace Hopper
Electrical Engineer