Use the math library

We have synonyms for prototype. We also say that it's the declaration. We also say that it is the header. And we also say that it is the signature of the function. So the prototype is the type that is returned by the function, the name of the function, and the type of the parameters for this function.

I will learn how to use the math library, meaning that we will use a library that contains the definitions of all the basic math functions that were written by very skilled developers. And they were compiled. And we have some binaries installed in the system. We have to find those binaries and use those binaries and kind of link them to our program so that we use their definition.

I want to use the definition for the pow function that was written by a developer. And I want to use the binary that was compiled from this definition. let's try to find where is the definition of all those math libraries. Usually it is installed in the system folder called /usr/lib for libraries.

Usually the name of the library starts with lib, l-i-b, and then some kind of name. So I'm going down but we have a lot of libraries here. But here is one of interest. In the middle of nowhere, we have libm.a And this is the math library for the C programming language. So we have found the binary that contains the definition of pow. And we want to use it in the program.

Because we know it's in /usr/lib, and the name is libm.a, well, we have to add a specific option and arguments to the gcc build command here.

```
gcc name_of_program.c absolute_pass_of_library -o executable_file
```

We'll write the absolute path of the library, so /usr/lib/libm.a. So sometimes the library is called differently. But it's always called libm. But sometimes you will have something else as an extension, maybe not .a. We have many libraries with a different extension here-- .so,.so.8 et cetera.

We have used the pow function that is defined in this library just here, the math library. And we know that the prototype is this one. There is a specific folder that contains all the headers of all the libraries that are installed on the system. And that folder is /usr/include. We have math.h, and all the files are ending with .h What does it mean? Well, it means header. So those files contain all the signature, the prototypes, the declarations, the headers of all the functions of all the libraries. So we

We can replace the prototype of a function defined in a library, for example defined in the math library with #include math.h.

And the consequence of writing that is that the compiler will first copy the content of math.h that is in /usr/include because this is the default folder for all the headers. And it will copy the content of this file to that very specific place location in my program. And after doing that, it will compile my program. That means that at the top of the program I will have the prototype of the pow function because it was copied.

So we know that all the headers are in /usr/include, and you have all the .h files here with all the prototypes. And we know that the definitions are in /usr/lib

So to get to the definition, I have to add here before the -o the absolute path of the library.

We can get the library in the gcc command line more or less automatically by using relative path.

So it is simply an option that is called -l for libraries. And you will have to replace the whole path here by deleting /usr/lib/ And you will have to delete the three first letters, l-i-b. And you will have only to get the characters after lib. So for the math library, it's only one character. It's m. And you will have to get rid of the extensions, so .a And you will have to get rid of any space between the l correcter here. So basically, what is staying is -lm without any space in between the - and the l and the m.

If I would like to use another library like libxkbfile.so, for example, I would replace m by skbfile. This is a longer name.

So I know how to search, how to add a library by using #include and the name of the file that contains all the headers. I know that you have to link the definition with a specific option here. Either you use the absolute path of the library or you use -l and then only the name without lib and without the extension. And that is how you use an external library that contains the function definitions of another developer.

Use multiple libraries in C