

Wireshell KISS-C Tutorials

My website, www.wireshell.site11.com, was designed to be an inspiration and a learning center for many young adults across the globe. Wireshell teaches students how to program in KISS-C, a programming language developed by KIPR and used in Botball to program CBC robots. These tutorials not only focus on basic programming like movements and exporting words to the CBC screen, but they also teach students more advanced and complex programming, such as while loops.

KISS-C Basics

The first tutorial is the very basics of KISS-C programming. This includes a “printf” statement and shows the students how to use the simulator. The “printf” statement is used to export words and variables to the CBC screen for the programmer's reference. These statements can prove useful when you are trying to detect a bug in the program. Sometimes, to find where the error in a program is, people put a reference point at every step. The simulator is also a very important tool in KISS-C. Rather than run your robot every single time, sometimes you can run it in the simulator. Students have to keep in mind though, that the robot movement in the simulator, and the real world will always be different.

Variables

Students will be taught variables in lesson two. Variables are useful in copying data, and interpreting data from the CBC sensors. Variables can also be used with while loops, to make a process execute as many times as defined, without having to re-write the command one-hundred times or so.

Useful Ways To Program

In lesson three, sleep, define, and comments will be learned. It is very important for students to use comments in everyday programming, in case someone wants to see what their code does, or so the programmer doesn't lose track of what code corresponds to what movement. Students will also learn how to properly use the sleep command. The sleep command is useful when lifting servos, making the create move, or making the robot wait for a given amount of time. Defines are also useful. Defines work as variables, but on a global scope. This way, you can define a constant speed and use it throughout the program.

CBC Motors and Servos

Lesson four contains instructions on how to program CBC motors and servos. This, along with programming create motors, is the most important part of KISS-C programming. Motors and servos are used to move the robot and/or objects, such as Botguy.

Common Create Commands

The fifth KISS-C programming tutorial focuses on common create commands, such as turning and driving forward. Students will first learn how to establish a connection to the create through the CBC. From there, they will learn the command to make it drive straight, then they will learn how to turn. During the tutorial, it will be made clear that it is important to put a “create_stop()” command at the end of every driving sentence, and that a “sleep()” command will be necessary to define the number of seconds the create should move.

Sensors

In lesson six the students will learn how to interpret data from the analog and digital sensors in the CBC. After learning the subject, students will be able to successfully display what a sensor reads on the CBC screen. This is not very useful yet, but students will learn that sensor data can be used to make decisions on what to do next.

If, Else

In lesson seven, students will learn that if, else if, and else decisions can be used to make decisions upon whether a statement is true or not. Students will also learn how to use their variables from the sensors to automatically tell the robot in which direction to move, or what task to perform next.

Relational Operators

The seventh tutorial focuses on relational operators. Used in if statements and while loops, relational operators can say what to do if the variable is less, more than, or equal to a given number or variable.

Logical Operators

Lesson eight teaches students about logical operators. Logical operators are used in if statements and while loops as well, but instead of comparing a number to a number, students will learn that they can tell whether to perform the same operation if two things are true, or if at least one thing is true.

While and Do While Loops

Lesson nine is on while and do while loops. They are similar to if statements, students will be taught while and do while loops execute a process until the condition is not true.

Increment and Decrement

In lesson ten, increment and decrement are used to add to and subtract numbers from variables. This can be useful in while loops to perform tasks as many times as necessary, as opposed to having to write them over and over again, such as in a slow servo program.

For Loops

Although for loops are not really included in the KISS-C syntax, I still thought it a good idea to inform the students how to use them. I would like to make a request that they be included in the next KISS-C version. The benefit of for loops is that they do the same thing as while loops: executing a task a certain number of times, but are less error prone because they have a certain number of lines.

Functions

Functions are useful because they provide the programmer an opportunity to not have to not write out a piece of code every time they want to run it in different places in their program. Functions act as shortcuts for whenever someone wants to preform an area of code, which is time consuming to write again. The programmer simply has to put their process within curly brackets and give it a name. Whenever they want to run what is in these curly brackets, they just have to write that name out again and the program refers to the block of code. This is why functions are useful.

Vision Tracking

In addition, in the vision tracking tutorial, students will learn how to train the CBC camera to track and follow a specific object. This could prove useful in Botball games where Botguy has to be found in a random location on the board, or it can be used to sort luggage in this year's game board.

All the tutorials above teach students who yearn to learn how to program KISS-C robots. These tutorials will save the team valuable time by teaching this language to new students. The students can log on at home and begin their programming. I designed this website, and these tutorials specifically for this reason, so enjoy!

Kipr's Software Tutorials

KISS-C Tutorials

- » [Lesson 1: Printf, Simulator](#)
- » [Lesson 2: Variables](#)
- » [Lesson 3: Sleep, Define, Comments](#)
- » [Lesson 4: CBC Motor and Servo Commands](#)
- » [Lesson 5: Common Create Commands](#)
- » [Lesson 6: Analog and Digital Sensors](#)
- » [Lesson 7: Else If](#)
- » [Lesson 8: Relational Operators](#)
- » [Lesson 9: Logical Operators](#)
- » [Lesson 10: While, Do While Loops](#)
- » [Lesson 11: Increment and Decrement](#)
- » [Lesson 12: For Loops](#)
- » [Lesson 13: Functions](#)
- » [Lesson 14: Global and Local Scope](#)
- » [Lesson 15: Vision](#)
- » [Lesson 16: Parallel Processes](#)
- » [Lesson 17: Mutual Exclusions](#)
- » [Programs](#)