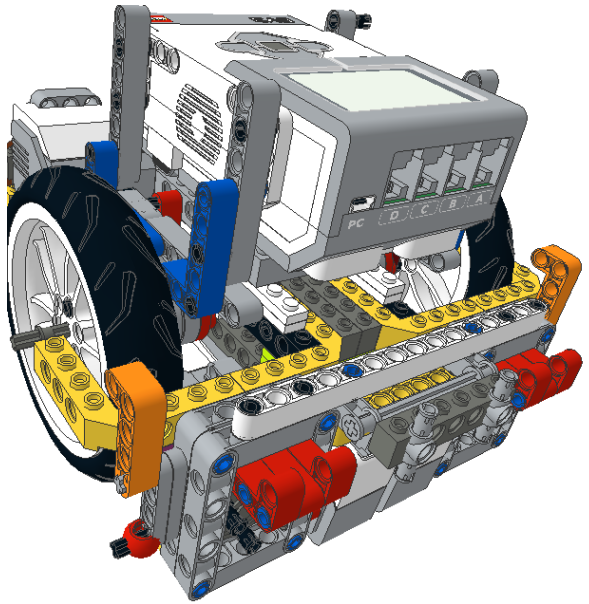
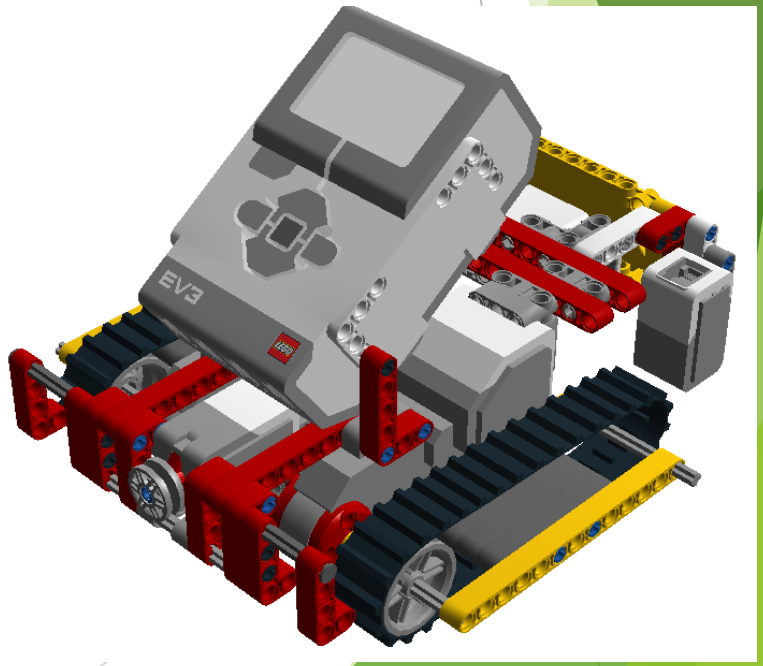
Robot JavaScript

For EV3 Robots





JavaScript Programming

- History of Languages
- Syntax
- Style Guide
- Language Patterns
- Integrated Development Environment (IDE)

History of Programming Languages

- 1st Generation: Machine Language
  CPU operates only on 1's and 0's
  Programmed by using toggle switches

History of Programming Languages

- 1st Generation: Machine Language
  CPU operates only on 1's and 0's
  Programmed by using toggle switches

History of Programming Languages

- 2nd Generation: Assembly Language
  Programmed using 3 or 4 letter codes
  Example: mov, add, sub, push, call, goto

History of Programming Languages

- 2nd Generation:
  Assembly Language
  Programmed using 3 or 4 letter codes

```
_sub:   push   ebp
        mov    ebp, esp
        mov    eax, [ebp+8]
        mov    ecx, [ebp+0Ch]
        lea    eax, [ecx+eax*2]
        pop    ebp
        retn
_main:  push   ebp
        mov    ebp, esp
        push   ecx
        mov    eax, [ebp+0Ch]
        mov    ecx, [eax+4]
        push   ecx
        call   dword ptr ds:__imp__atoi
        add    esp, 4
        mov    [ebp-4], eax
        mov    edx, [ebp-4]
        push   edx
        mov    eax, [ebp+8]
```

History of Programming Languages

- 3rd Generation: Procedural Languages

- Examples:
  - C
  - BASIC (Beginners All-purpose Symbolic Instruction Code)
  - Fortran (Formula Translation)
  - JavaScript

History of Programming Languages

- 3rd Generation: Procedural Languages

  BASIC (1964)

```
10 INPUT "What is your name: ", U$
20 PRINT "Hello "; U$
30 INPUT "How many stars do you want: ", N
40 S$ = ""
50 FOR I = 1 TO N
60 S$ = S$ + "*"
70 NEXT I
80 PRINT S$
90 INPUT "Do you want more stars? ", A$
100 IF LEN(A$) = 0 THEN GOTO 90
110 A$ = LEFT$(A$, 1)
120 IF A$ = "Y" OR A$ = "y" THEN GOTO 30
130 PRINT "Goodbye "; U$
140 END
```

History of Programming Languages

- 3rd Generation: Procedural Languages

C (1973)

```
#pragma config(Sensor, S1,     touchSensor1,   sensorEV3_Touch)
#pragma config(Sensor, S2,     colorSensor,    sensorEV3_Color, modeEV3Color_Color)
#pragma config(Sensor, S3,     colorSensor2,   sensorEV3_Color, modeEV3Color_Color)
#pragma config(Sensor, S4,     ultrasonicSensor, sensorEV3_Ultrasonic)
#pragma config(Motor,  motorA,          valveControl,  tmotorNXT, openLoop, encoder)
#pragma config(Motor,  motorB,          rightMotor,    tmotorEV3_Large, PIDControl, driveLeft, encoder)
#pragma config(Motor,  motorC,          leftMotor,     tmotorEV3_Large, PIDControl, driveRight, encoder)
#pragma config(Motor,  motorD,          valveControl2, tmotorNXT, openLoop, encoder)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

#define BasicPragmasDefined;
#define ColorSensorDefined;
#define UltrasonicSensorDefined;

/*-----------------------------------
  Useful Pragmas

 #pragma config(Sensor, S1,     touchSensor1,   sensorEV3_Touch)
 #pragma config(Sensor, S2,     colorSensor,    sensorEV3_Color)
 #pragma config(Sensor, S3,     ultrasonicSensor, sensorEV3_Gyro)
-----------------------------------*/
#include "Menu2014.c";
#include "Library2014.c";
```

History of Programming Languages

- 3rd Generation: Procedural Languages

C (1973)

```
294  // They are for internal purposes in dealing with the output of the light sensors
295  float getLightPortion() {
296    float result;
297    long numerator, denominator;
298    if((long) SensorRaw[lightSensorB] > tHighLight2) { tHighLight2 = (long) SensorRaw[lightSensorB];
299    if((long) SensorRaw[lightSensorB] < tLowLight2) { tLowLight2 = (long) SensorRaw[lightSensorB]; }
300    if((long) SensorRaw[lightSensorA] > tHighLight) { tHighLight = (long) SensorRaw[lightSensorA]; }
301    if((long) SensorRaw[lightSensorA] < tLowLight) { tLowLight = (long) SensorRaw[lightSensorA]; }
302    if(direction == BACKWARD) {
303      numerator = (long) SensorRaw[lightSensorB] - tLowLight2;
304      denominator = tHighLight2 - tLowLight2;
305    } else {
306      numerator = (long) SensorRaw[lightSensorA] - tLowLight;
307      denominator = tHighLight - tLowLight;
308    }
309    result = (float)  numerator / denominator;
310    if(result < 0) { result=0; }
311    if(result > 1) { result=1; }
312    return result;
313  }
314  float getLightPercent() {
315      return 100 * getLightPortion();
316  }
317  float getLightDiffPortion() {
318
319    float result;
320    long numerator1;
321    long numerator2;
322    long numerator;
```

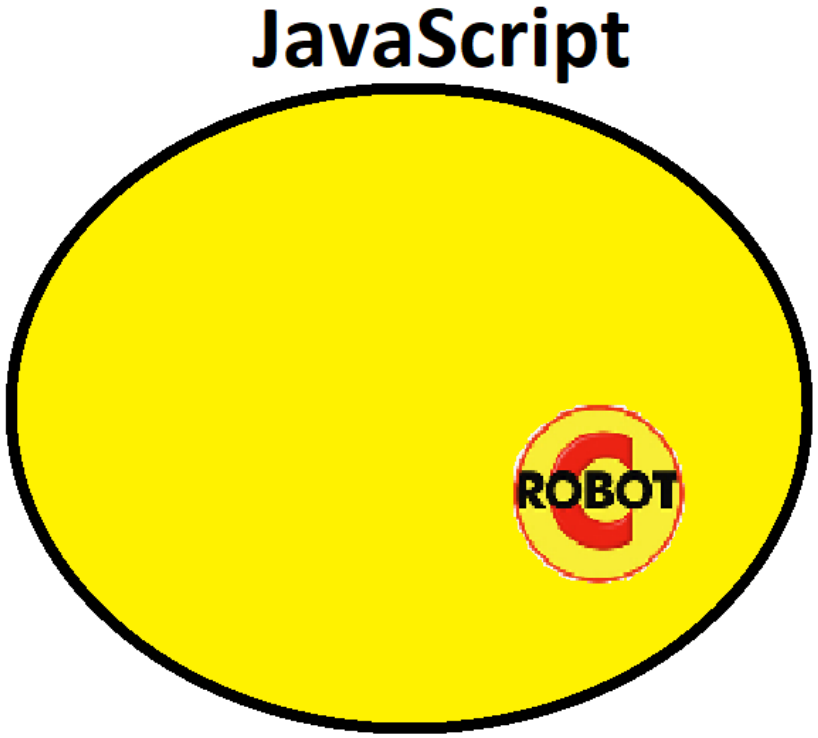
RJS

- 3rd Generation:
  Procedural Languages

- JavaScript (1996)

```
function clearCursor(cValue) {
  clearCircle(10, cValue*15+3, 5)
}
function drawCursor(cValue) {
  circle(10, cValue*15+3, 5)
}
var menu=[]                                  // Initialize an array called menu
menu.push(new menuItem('Item 1', 1))         // Push a new menuItem onto the ar
menu.push(new menuItem('Item 2', 2))         // Push another new menuItem onto
menu.push(new menuItem('Item 3', 3))         // Push another new menuItem onto
menu.push(new menuItem('Item 4', 4))         // Push another new menuItem onto
menu.push(new menuItem('Item 5', 5))         // Push another new menuItem onto
menu.push(new menuItem('Item 6', 6))         // Push another new menuItem onto
function drawMenu() {                        // Draw the menu
  clearScreen()                              // Clear the screen
  for(j=0;j<menu.length;j++) {               // For each menu item
    drawText(20, j*15+15, menu[j].prompt)
  }
}
option = 1
while(true) {
  drawMenu()
  while(true) {
    drawCursor(option)
    theButtonPressed = waitForPress()
    clearCursor(option)
    if ( theButtonPressed==1 ) option-=1
    if ( theButtonPressed==3 ) option+=1
    if ( theButtonPressed==2 ) break
  }
```

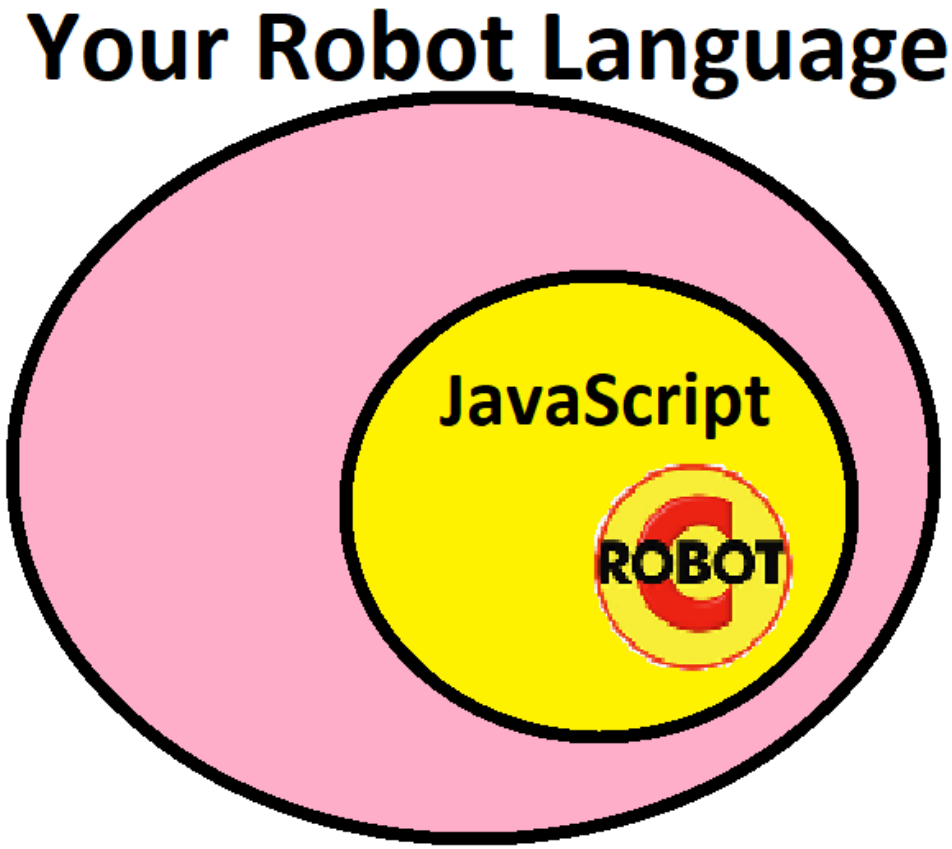
JavaScript is a Superset of C



Robot JavaScript includes syntax, many keywords, and some functions of RobotC.

Robot JavaScript extends the C Language by adding operator overloading, less strict variable typing, JSON objects and event handling.

Your Job this Fall Season:
Create a Superset of JavaScript



You need to extend the Robot JavaScript language by adding your own functions to make the robot perform higher level tasks.

Example of a Superset of Robot JavaScript

These functions were defined by the programmer:

driveForward()
driveBackward()
pointTurnCounterClockwise()
pointTurnClockwise()

JavaScript gives you the ability to create your own functions. To create your own "Superset".

```
function steptopullupbar() {
        driveBackward(1.0625,35)        //back aw
        stopAllMotors()
        pointTurnCounterClockwise(90)// lines up
        syncMotors(B, C,  30)
        sleep(1000)
        while(touchSensorPressed()==false) {

          sleep(50)
        }
        stopAllMotors()
        sleep(1000)
        stopAllMotors()
        driveForward(31.25,30)
        stopAllMotors()
        pointTurnCounterClockwise(75)
        stopAllMotors()
        driveForward(11.5,20)
        stopAllMotors()
        pointTurnClockwise(75)
        stopAllMotors()
        driveForward(6,20)
        // Turn one wheel at a time
        syncMotors( B, C, 30, -100 )
        sleep(1000)
        syncMotors( B, C, 30, 100 )
        sleep(1000)
        // Accelerate the motor on port (B, C,)
        for(i=1;i<20;i++) {
          setMotor(2, 10+i*4)
          //setMotor(3, 10+i*-4)
          sleep(300)
        }
        sleep(3000)
}
```

Organization of Text Documents "Books"

- Book
- Chapters (one or more)
- Paragraphs (one or more)
- Sentences (one or more)
- Words (one or more)

Organization of Text Documents "Computer Programs"

- Program
- Functions (one or more)
- Blocks (one or more)
- Statements (one or more)
- Expressions (one or more)

Similarities between “Books” and “Computer Programs”

- Book
- Chapters (one or more)
- Paragraphs (one or more)
- Sentences (one or more)
- Words (one or more)

- Program
- Functions (one or more)
- Blocks (one or more)
- Statements (one or more)
- Expressions (one or more)

Components of a JavaScript Program

```
20 function drawMenu() {
21   clearScreen()
22   for(j=0;j<menu.length;j++) {
23     drawText(20, j*15+15, menu[j].prompt)
24   }
25 }
26 option = 1
27 while(true) {
28   drawMenu()
29   while(true) {
30     drawCursor(option)
31     theButtonPressed = waitForPress()
32     clearCursor(option)
33     if ( theButtonPressed==1 ) option-=1
34     if ( theButtonPressed==3 ) option+=1
35     if ( theButtonPressed==2 ) break
36   }
37   switch (option) {
38   case 1:
39     alert('Case 1: blah blah blah')
40     break;
41   case 2:
42     alert('Case 2: blah blah blah')
43     break;
44   case 3:
```

Function

Block of Code

Statement

Expression

Block of Code

Block of Code

Assignment Expression

If Statements

Function Expression

String Literal Expressions

Common Expressions

- Literal
  1
  3.14159
  true
- Assignment
  a = 2
  c = 10
  d = a + c
- Mathematical
  4 + 6
  7 * 2
  9 / 3
- Functional
  alert(‘Hello World’)
  drawText(10, 10, ‘Testing’)
  sqrt(9)

Less Common Expressions

- Increment/Decrement
  a++
  b--
- Parenthesized
  (a + 2) * 3
  (ls(2) * 4) + 16
- Assignment Operator
  a += 2
  c += 10
  d *= a

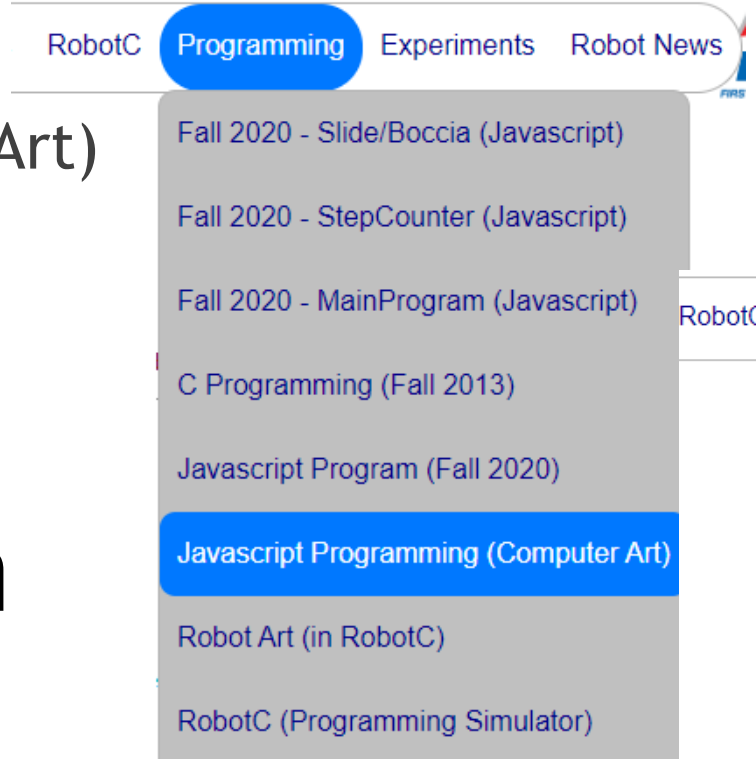
Advanced Expressions

- Ternary Expressions
  a = (b > 19) ? 100 : -100
- Member Index Expressions
  a = motor[1]
  b = sensor[3]
- Member Dot Property
  a = coordinate.x
- Member Dot Method
  a.push(lightSensorPct())

Integrated Development Environments

- RoboCatz Tools that you can use
- JavaScript Programming (Computer Art)
- JavaScript Programming (Robot Simulator)

https://robocatz.com



Integrated Development Environments

- Demonstrate the Robot Simulator
- Demonstrate Computer Art
- Kids use the computers now for Computer Art.