

Sample Codes:

Use Outline for easy navigation.

Joint Test- 90 degree rotation and back.

```
new Job({name: "jointTest",  
  
    do_list: [Dexter.set_parameter("MaxSpeed", 20),  
              Dexter.move_all_joints([0, 0, 0, 0, 0]),  
              Dexter.move_all_joints([90, 0, 0, 0, 0]),  
              Dexter.move_all_joints([0, 0, 0, 0, 0]),  
              Dexter.move_all_joints([0, 80, 0, 0, 0]),  
              Dexter.move_all_joints([0, 0, 0, 0, 0]),  
              Dexter.move_all_joints([0, 0, 90, 0, 0]),  
              Dexter.move_all_joints([0, 0, 0, 0, 0]),  
              Dexter.move_all_joints([0, 0, 0, -90, 0]),  
              Dexter.move_all_joints([0, 0, 0, 0, 0])  
  
    ]})
```

Hard-coded job that draws on a whiteboard.

```
new Job({name: "amogus",  
  
    do_list: [Dexter.set_parameter("MaxSpeed", 20),  
              Dexter.move_all_joints([0, 0, 0, 0, 0]),  
              Dexter.move_all_joints([90, -35, 100, 0, 0]),    //xyz  
              Dexter.move_all_joints([90, -33.5, 100, 0, 0]), //upper right eye  
              Dexter.wait_until(5),  
              Dexter.move_all_joints([90, -36, 110, 0, 0]), //bottom right eye  
              Dexter.move_all_joints([80, -36, 109, 0, 0]), //bottom left eye  
              Dexter.move_all_joints([80, -33, 100, 0, 0]), //upper left eye  
              Dexter.move_all_joints([90, -33.5, 100, 0, 0]),  
              Dexter.move_all_joints([90, -40, 100, 0, 0]),  
              Dexter.move_all_joints([95, -33, 95, 0, 0]),  
              Dexter.move_all_joints([95, -32, 97, 0, 0]), //upper right of body  
              Dexter.move_all_joints([95, -41, 123, 0, 0]), //lower right of body  
              Dexter.move_all_joints([85, -42, 122, 0, 0]), //lower left of right leg  
              Dexter.move_all_joints([85, -39, 114, 0, 0]), //upper left of right leg  
              Dexter.move_all_joints([80, -39, 114, 0, 0]), //upper right of left leg  
              Dexter.move_all_joints([80, -41.5, 122.5, 0, 0]), //lower right of left leg  
              Dexter.move_all_joints([69, -39, 120, 0, 0]), //lower left of left leg
```

```

Dexter.move_all_joints([69, -35, 109, 0, 0]),//midpoint up left
Dexter.move_all_joints([69, -30, 96, 0, 0]),//upper left of body
Dexter.move_all_joints([95, -32, 97, 0, 0]),
Dexter.move_all_joints([80, -50, 114, 0, 0]),
Dexter.move_all_joints([0, 0, 0, 0, 0])
}})

```

Follow Me mode Test

```

Function setFollowMe(){
    var ret(MD = [])
    ret(MD.push(make_ins("w", DIFF_FORCE_SPEED_FACTOR_ANGLE,
DEF_SPEED_FACTOR_DIFF))
    ret(MD.push(make_ins("w", DIFF_FORCE_SPEED_FACTOR_ROT,
DEF_SPEED_FACTOR_DIFF))
    ret(MD.push(make_ins("w", PID_P, 0))
    ret(MD.push(make_ins("w", PID_ADDRESS, 3))
    ret(MD.push(make_ins("w", PID_ADDRESS, 4))
    ret(MD.push(make_ins("w", PID_ADDRESS, 0))
    ret(MD.push(make_ins("w", PID_ADDRESS, 1))
    ret(MD.push(make_ins("w", PID_ADDRESS, 2))
    ret(MD.push(make_ins("w", SPEED_FACTORA, DEF_SPEED_FACTOR_A))
    ret(MD.push(make_ins("S", "J1Friction",5 ))
    ret(MD.push(make_ins("S", "J2Friction",5 ))
    ret(MD.push(make_ins("S", "J3Friction",5 ))
    ret(MD.push(make_ins("S", "J4Friction",15 ))
    ret(MD.push(make_ins("S", "J5Friction",15 ))
    ret(MD.push(make_ins("w", 67, 0))
    ret(MD.push(make_ins("w", 68, 0))
    ret(MD.push(make_ins("w", 69, 0)) //Nice
    ret(MD.push(make_ins("w", 70, 0))
    ret(MD.push(make_ins("w", 71, 0))

    ret(MD.push(make_ins("w", 79, 50 ^ 200 ))
    ret(MD.push(make_ins("w", 80, 50 ^ 200 ))
    ret(MD.push(make_ins("w", 81, 50 ^ 200 ))
    ret(MD.push(make_ins("w", 42, 12448 ))
    Return ret(MD
}

```

Palletizing Code [WORK IN PROGRESS]

```

function handle_print_job_1_dialog_input(vals){
  if(vals.clicked_button_value == "Continue"){
    Job.print_job_1.user_data.color = vals.pallets
    Job.print_job_1.user_data.color_status = "ok"
    Job.print_job_1.user_data.stack = vals.stack
    Job.print_job_1.user_data.stack_status = "ok"
  }
  else if(vals.clicked_button_value == "Cancel"){
    Job.print_job_1.user_data.color_status = "cancel"
  }
}

```

```

function print_job_1_dialog(){
  show_window({content:`
    <span style="font-size:18px; font-family:Arial; font-style:normal; color:#000000;
font-weight:200;">
      How many pallets are there?
    <br/>
    Please choose a value from 0-6.</span>
    <br/>
    <span style="font-size:18px; font-family:Comic Sans MS; font-style:normal;
color:#000000; font-weight:200;">
      <input name="pallets" type="range" id="Pallets" list="tickmarks"
        min="0" max="6" oninput="rangeValue.innerText = this.value"/>
      <label for="Pallets"># of pallets</label>
      <datalist id="tickmarks">
        <option value="0" label="0"></option>
        <option value="1" label="1"></option>
        <option value="2" label="2"></option>
        <option value="3" label="3"></option>
        <option value="4" label="4"></option>
        <option value="5" label="5"></option>
        <option value="6" label="6"></option>
      </datalist>

    <br/>
    How many should I stack?</span>
    <br/>
    <span style="font-size:18px; font-family:Comic Sans MS; font-style:normal;
color:#000000; font-weight:200;">
      <input name="stack" type="range" id="Stack" list="tickmarks"
        min="0" max="6" oninput="rangeValue.innerText = this.value"/>
      <label for="stack"># to stack</label>
      <datalist id="tickmarks">

```

```

        <option value="0" label="0"></option>
        <option value="1" label="1"></option>
        <option value="2" label="2"></option>
        <option value="3" label="3"></option>
        <option value="4" label="4"></option>
        <option value="5" label="5"></option>
        <option value="6" label="6"></option>
    </datalist>

```

```

</span>
<br/>
<input type="submit" value="Continue"/>
<input type="submit" value="Cancel"/>
,
title: "How many pallets should I stack!",
background_color: "#B6FFFF",
height: 225,
callback: handle_print_job_1_dialog_input
}
)
}

new Job({name: "print_job_1",
do_list:
[
    Dexter.move_all_joints([0, 45, 90, -45, 0]),
    function(){print_job_1_dialog()},
    Robot.wait_until(function(){return this.user_data.color_status != undefined}),

[j3],
/* function(vals){if(this.user_data.color == 1){
    [j3],
        prompt("Are you sure?","1")
    }},
*/

//Dexter.move_all_joints([0, 0, 0, 0, 0]),
function(){if (this.user_data.color_status == "cancel"){
    return Robot.error("ran out of filament.")
}

    this.user_data.color

```

```
    })  
    Job.print_job_1.start()
```