ROBOTC Natural Language - NXT Quick Reference:

<table>
<thead>
<tr>
<th>Robot Type</th>
<th>robotType();</th>
<th>robotType(rembot);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose which robot you are using.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default bot: none.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start Motor</th>
<th>startMotor();</th>
<th>startMotor(motorC, -25);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set a specific motor to a speed.</td>
<td>wait();</td>
<td>wait(0.5);</td>
</tr>
<tr>
<td>Default motor and speed: motorA, 75.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stop Motor</th>
<th>startMotor();</th>
<th>startMotor(motorC, -25);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop a specific motor.</td>
<td>wait();</td>
<td>wait(0.5);</td>
</tr>
<tr>
<td>Default motor: motorA.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wait</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait an amount of time measured in seconds.</td>
<td>wait();</td>
<td>wait(2.7);</td>
</tr>
<tr>
<td>Default time: 1.0.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wait in Milliseconds</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait an amount of time measured in milliseconds.</td>
<td>waitInMilliseconds();</td>
<td>waitInMilliseconds(2700);</td>
</tr>
<tr>
<td>Default time: 1000.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Touch</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for the Touch Sensor to be pressed.</td>
<td>untilTouch();</td>
<td>untilTouch(S4);</td>
</tr>
<tr>
<td>Default sensor port: S1.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Release</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for the Touch Sensor to be released.</td>
<td>untilRelease();</td>
<td>untilRelease(S4);</td>
</tr>
<tr>
<td>Default sensor port: S1.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Bump</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for the Touch Sensor to be pressed in and then released out.</td>
<td>untilBump();</td>
<td>untilBump(S4, 100);</td>
</tr>
<tr>
<td>Default sensor port and delay time: S1, 10.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Button Press</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for a button on the NXT to be pressed.</td>
<td>untilButtonPress();</td>
<td>untilButtonPress(rightBtnNXT);</td>
</tr>
<tr>
<td>Default button: centerBtnNXT.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Sonar - Less Than</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for the Sonar Sensor to read a value in cm less than the threshold.</td>
<td>untilSonarLessThan();</td>
<td>untilSonarLessThan(45, S1);</td>
</tr>
<tr>
<td>Default threshold and sensor port: 30, S4.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Until Sonar - Greater Than</th>
<th>startMotor();</th>
<th>startMotor(motorC, 50);</th>
</tr>
</thead>
<tbody>
<tr>
<td>The robot waits for the Sonar Sensor to read a value in cm greater than the threshold.</td>
<td>untilSonarGreaterThan();</td>
<td>untilSonarGreaterThan(45, S1);</td>
</tr>
<tr>
<td>Default threshold and sensor port: 30, S4.</td>
<td>stopMotor();</td>
<td>stopMotor(motorC);</td>
</tr>
</tbody>
</table>
**Until Dark**
The robot waits for the Light Sensor to read a value less than the threshold.  
*Default threshold and sensor port: 45, S3.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilDark(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilDark(15, S2); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Until Light**
The robot waits for the Light Sensor to read a value greater than the threshold.  
*Default threshold and sensor port: 45, in2.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilLight(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilLight(85, S2); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Until Sound - Less Than**
The robot waits for the Sound Sensor to read a value less than the threshold.  
*Default threshold and sensor port: 50, S2.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilSoundLessThan(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilSoundLessThan(15, S3); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Until Sound - Greater Than**
The robot waits for the Sound Sensor to read a value greater than the threshold.  
*Default threshold and sensor port: 50, S2.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilSoundGreaterThan(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilSoundGreaterThan(85, S3); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Until Rotations**
The robot waits for a motor-encoder to reach a specified number of rotations.  
*Default rotations, encoder: 1.0, motorB.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilRotations(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilRotations(2.75, motorA); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Until Encoder Counts**
The robot waits for a motor-encoder to reach a specified number of encoder counts.  
*Default counts, encoder: 360, motorB.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startMotor(); untilEncoderCounts(); stopMotor();</td>
<td></td>
</tr>
<tr>
<td>startMotor(motorC, 50); untilEncoderCounts(990, motorA); stopMotor(motorC);</td>
<td></td>
</tr>
</tbody>
</table>

**Forward**
The robot drives straight forward.  
*Default speed: 75.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward(); wait(); stop();</td>
<td></td>
</tr>
<tr>
<td>forward(50); wait(2.0); stop();</td>
<td></td>
</tr>
</tbody>
</table>

**Backward**
The robot drives straight backward.  
*Default speed: -75.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backward(); wait(); stop();</td>
<td></td>
</tr>
<tr>
<td>backward(50); wait(2.0); stop();</td>
<td></td>
</tr>
</tbody>
</table>

**Point Turn**
The robot makes a sharp turn in place.  
*Default direction and speed: right, 75.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pointTurn(); wait(); stop();</td>
<td></td>
</tr>
<tr>
<td>pointTurn(left, 50); wait(0.4); stop();</td>
<td></td>
</tr>
</tbody>
</table>

**Swing Turn**
The robot makes a wide turn, activating only one drive motor.  
*Default direction and speed: right, 75.*

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swingTurn(); wait(); stop();</td>
<td></td>
</tr>
<tr>
<td>swingTurn(left, 50); wait(0.75); stop();</td>
<td></td>
</tr>
</tbody>
</table>

**Stop**
The robot halts both driving motors, coming to a stop.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward(); wait(); stop();</td>
<td></td>
</tr>
<tr>
<td>forward(50); wait(2.0); stop();</td>
<td></td>
</tr>
</tbody>
</table>
**Line Track - for Time**
The robot tracks a dark line on a light surface for a specified time in seconds.
*Default time, threshold, sensors: 5.0, 45, S3.*

```c
lineTrackForTime();
stop();
```

```c
lineTrackForTime(7.5, 75, S2);
stop();
```

---

**Line Track - for Rotations**
The robot tracks a dark line on a light surface for a specified distance in rotations.
*Default time, threshold, sensors: 3.0, 45, S3.*

```c
lineTrackForRotations();
stop();
```

```c
lineTrackForRotations(4.75, 75, S3);
stop();
```

---

**Move Straight - for Time**
The robot will use encoders to maintain a straight path for a specified time in seconds.
*Default time, rightEncoder, leftEncoder: 5.0, motorB, motorC.*

```c
moveStraightForTime();
stop();
```

```c
moveStraightForRotations(4.75, motorC, motorA);
stop();
```

---

**Move Straight - for Rotations**
The robot will use encoders to maintain a straight path for a specified distance in encoder rotations.
*Default rotations, rightEncoder, leftEncoder: 1.0, motorB, motorC.*

```c
moveStraightForRotations();
stop();
```

```c
moveStraightForRotations(4.75, motorC, motorA);
stop();
```

---

**Tank Control**
The robot is remote controlled with the right motor mapped to the right joystick and the left motor mapped to the left joystick.
*Default right and left joystick: joy1_y2, joy1_y1.*

```c
while(true)
{
    tankControl();
}
```

```c
while(true)
{
    tankControl(joystick.joy1_x2, joystick.joy1_x1);
}
```

---

**Arcade Control**
The robot is remote controlled with both motors mapped to a single joystick.
*Default vertical and horizontal joysticks: joy1_y2, joy1_y1.*

```c
while(true)
{
    arcadeControl();
}
```

```c
while(true)
{
    arcadeControl(joystick.joy1_y1, joystick.joy1_x1);
}
```