



iRobot Education's Create 3

Web Playground for Python

Explore the following code snippets used to program the Create 3 Education Robot in the [Web Playground for Python](#).

<pre>@event(robot.when_bumped, [True, False]) async def when_bumper(robot):</pre>	<p>Robot senses input from the bumper sensors on the left and right side.</p> <p>[True, False]: Left Side of Bumper Pressed [False, True]: Right Side of Bumper Pressed []: Any Bumper Pressed</p>
<pre>@event(robot.when_touched, [True, False]) async def when_touched(robot):</pre>	<p>Robot senses input from the bumper sensors on the left and right side.</p> <p>[True, False]: Button 1 Pressed [False, True]: Button 2 Pressed</p>
<pre>@event(robot.when_play) async def when_play(robot):</pre>	<p>Commands run when Play button is pressed.</p>
<pre>await robot.move(16)</pre>	<p>Robot moves forward specified number of centimeters.</p> <p>Example shows 16cm.</p>
<pre>await robot.arc(Robot.DIR_LEFT, 90, 4) await robot.arc(Robot.DIR_RIGHT, 90, 4)</pre>	<p>Robot drives forward or backward in an arc, either clockwise (RIGHT) or counter-clockwise (LEFT) for a specified number of degrees along a curve of a specified radius.</p> <p>Example #1 shows robot driving counter-clockwise 90° around a curve with a 4cm radius.</p>
<pre>await robot.turn_left(90) await robot.turn_right(90)</pre>	<p>Robot rotates clockwise (RIGHT) or counter-clockwise (LEFT) for a specified number of degrees.</p> <p>Example #1 shows robot driving counter-clockwise 90°</p>
<pre>await robot.navigate_to(16, 16)</pre>	<p>Robot navigates across an invisible grid of centimeter increments. The origin of (0, 0) is automatically set to the robot's starting position, with the x-axis aligning to the robot's center button.</p> <p>Example shows robot driving to the coordinate (16, 16)</p>
<pre>await robot.get_position()</pre>	<p>Robot identifies its current location based on an invisible grid of centimeter increments. The x-axis aligns to the robot's center button.</p> <p>Example will return robot's current XY coordinates.</p>



await robot.reset_navigation()	Command reset robot's navigation grid origin of (0, 0) to current position of robot, with the x-axis aligning to the robot's center button.
await robot.set_lights_rgb(0, 0, 255) await robot.set_lights(Robot.LIGHT_ON, Color(R, G, B)) await robot.set_lights(Robot.LIGHT_SPIN, Color(R, G, B)) await robot.set_lights(Robot.LIGHT_BLINK, Color(R, G, B)) await robot.set_lights(Robot.LIGHT_OFF, Color(R, G, B))	Command controls the light pattern (on, spinning, blinking or off) and RGB color of the LED's in the robot's Ring Light. *Example #1 shows a convenience function with light pattern automatically set to "on," as this is the most commonly used light status. The color of the lights is pure blue.
await robot.set_wheel_speeds(10, -10)	Command sets the speed (in cm/s) and direction of robot's wheels on either side. Example shows robot's left wheel (on the same side as Button 1) driving forward 10 cm/s and right wheel (on the same side as Button 2) driving backward 10 cm/s. The robot will rotate in place, clockwise.
await robot.wait(0.5)	Command specifies a delay in seconds before moving onto the next line. Example tells robot to wait .5 seconds before reading next line.
await robot.play_note(440, 0.25)	Robot will play a musical note of a specified frequency for a specified duration. Example plays the note A4 for .25 seconds.
await robot.stop_sound()	Robot will stop all sounds currently playing.



iRobot Education's Create 3

RGB Color Reference Sheet

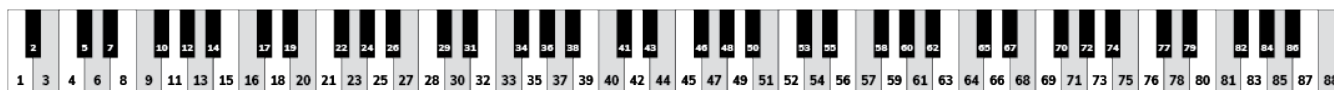
Use these RGB values when setting the color of the LED's in the Ring Light for the Create 3 Education Robot in the [Web Playground for Python](#).

100% Brightness	50% Brightness
(255, 0, 0)	(127, 0, 0)
(255, 64, 0)	(127, 32, 0)
(255, 115, 0)	(127, 57, 0)
(0, 255, 0)	(0, 127, 0)
(101, 197, 181)	(50, 98, 90)
(0, 0, 255)	(0, 0, 127)
(111, 71, 127)	(55, 35, 63)
(255, 255, 255)	(127, 127, 127)
(60, 60, 60)	(0, 0, 0)

iRobot Education's Create 3

Note Frequencies for Python

The following note values are for use in the [Web Playground for Python](#).



Key #	Note	Hz	Constant
1	A ₀	27.50	
2	A ₀ [#] /B ₀ ^b	29.14	
3	B ₀	30.87	
4	C ₁	32	C1
5	C ₁ [#] /D ₁ ^b	34	C1_SHARP
6	D ₁	36	D1
7	D ₁ [#] /E ₁ ^b	38	D1_SHARP
8	E ₁	41	E1
9	F ₁	43	F1
10	F ₁ [#] /G ₁ ^b	46	F1_SHARP
11	G ₁	48	G1
12	G ₁ [#] /A ₁ ^b	51	G1_SHARP
13	A ₁	55	A1
14	A ₁ [#] /B ₁ ^b	58	A1_SHARP
15	B ₁	61	B1
16	C ₂	65	C2
17	C ₂ [#] /D ₂ ^b	69	C2_SHARP
18	D ₂	73	D2
19	D ₂ [#] /E ₂ ^b	77	D2_SHARP
20	E ₂	82	E2
21	F ₂	87	F2
22	F ₂ [#] /G ₂ ^b	92	F2_SHARP
23	G ₂	97	G2
24	G ₂ [#] /A ₂ ^b	103	G2_SHARP
25	A ₂	110	A2
26	A ₂ [#] /B ₂ ^b	116	A2_SHARP
27	B ₂	123	B2
28	C ₃	130	C3
29	C ₃ [#] /D ₃ ^b	138	C3_SHARP
30	D ₃	146	D3

Key #	Note	Hz	Constant
31	D ₃ [#] /E ₃ ^b	155	D3_SHARP
32	E ₃	164	E3
33	F ₃	174	F3
34	F ₃ [#] /G ₃ ^b	184	F3_SHARP
35	G ₃	195	G3
36	G ₃ [#] /A ₃ ^b	207	G3_SHARP
37	A ₃	220	A3
38	A ₃ [#] /B ₃ ^b	233	A3_SHARP
39	B ₃	246	B3
40	C ₄	261	C4
41	C ₄ [#] /D ₄ ^b	277	C4_SHARP
42	D ₄	293	D4
43	D ₄ [#] /E ₄ ^b	311	D4_SHARP
44	E ₄	329	E4
45	F ₄	349	F4
46	F ₄ [#] /G ₄ ^b	369	F4_SHARP
47	G ₄	391	G4
48	G ₄ [#] /A ₄ ^b	415	G4_SHARP
49	A ₄	440	A4
50	A ₄ [#] /B ₄ ^b	466	A4_SHARP
51	B ₄	493	B4
52	C ₅	523	C5
53	C ₅ [#] /D ₅ ^b	554	C5_SHARP
54	D ₅	587	D5
55	D ₅ [#] /E ₅ ^b	622	D5_SHARP
56	E ₅	659	E5
57	F ₅	698	F5
58	F ₅ [#] /G ₅ ^b	739	F5_SHARP
59	G ₅	783	G5
60	G ₅ [#] /A ₅ ^b	830	G5_SHARP

Key #	Note	Hz	Constant
61	A ₅	880	A5
62	A ₅ [#] /B ₅ ^b	932	A5_SHARP
63	B ₅	987	B5
64	C ₆	1046	C6
65	C ₆ [#] /D ₆ ^b	1108	C6_SHARP
66	D ₆	1174	D6
67	D ₆ [#] /E ₆ ^b	1244	D6_SHARP
68	E ₆	1318	E6
69	F ₆	1396	F6
70	F ₆ [#] /G ₆ ^b	1479	F6_SHARP
71	G ₆	1567	G6
72	G ₆ [#] /A ₆ ^b	1661	G6_SHARP
73	A ₆	1760	A6
74	A ₆ [#] /B ₆ ^b	1864	A6_SHARP
75	B ₆	1975	B6
76	C ₇	2093	C7
77	C ₇ [#] /D ₇ ^b	2217	C7_SHARP
78	D ₇	2349	D7
79	D ₇ [#] /E ₇ ^b	2489	D7_SHARP
80	E ₇	2637	E7
81	F ₇	2793	F7
82	F ₇ [#] /G ₇ ^b	2959	F7_SHARP
83	G ₇	3135	G7
84	G ₇ [#] /A ₇ ^b	3322	G7_SHARP
85	A ₇	3520	A7
86	A ₇ [#] /B ₇ ^b	3729	A7_SHARP
87	B ₇	3951	B7
88	C ₈	4186	C8