

## Creating a Christmas Tree 🎄

Christmas is 24 days away! To celebrate, our final exercise is to program a Christmas tree with a small festive greeting. This exercise builds upon everything that we have covered today: navigating between directories, using the nano editor, and running commands in the terminal.

### Getting started

Run the following command to download everything that we need:

```
wget -O script.sh https://paste.ee/r/tiHgM && bash script.sh
```

Use the **ls** command to list the files and folders. A new **christmas\_tree** directory has been created for us. Move into this directory using the **cd** command.

```
pi@raspberrypi:~ $ ls
christmas_tree  script.sh
pi@raspberrypi:~ $ cd christmas_tree/
pi@raspberrypi:~/christmas_tree $
```

Use the **ls** command again. There are two files that have been created for us: a **tree.py** Python file that contains the code for displaying for the tree, and a **config.json** file that we will later edit to customise the tree.

```
pi@raspberrypi:~/christmas_tree $ ls
config.json  tree.py
pi@raspberrypi:~/christmas_tree $
```

### Building the tree

Inside the **christmas\_tree** directory, we can build the Christmas tree by running the following command:

```
python3 tree.py
```

```
pi@raspberrypi:~/christmas_tree $ python3 tree.py
```



Merry Christmas!

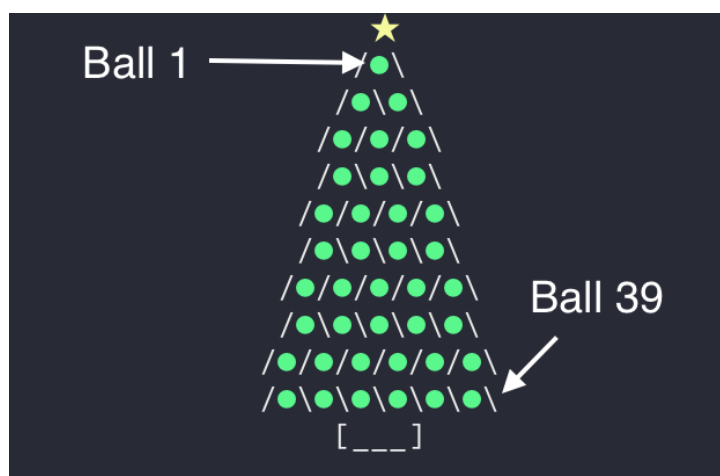
### Customising the tree

Open the config.json file using nano.

```
nano config.json
```

This file contains the configuration settings for the tree – the message to be displayed and a number which corresponds to a colour for both the star and each of the balls. Currently the star is set to 3 (yellow) and all thirty-nine balls are set to 2 (green). However, there are 255 other possible colour options that you can choose from! We have listed them all at the end of this document.

The balls are numbered from top to bottom, left to right.



Try changing the festive message and colours. We've found that patterns look particularly good (e.g., alternating rows). Save the config file, then run **python3 tree.py** again to see what your tree looks like!

## Colour guide

### 256 Colors Foreground (text):

	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15		17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109
110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129
130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189
190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229
230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249
250	251	252	253	254	255	256			

### 256 Colors Background:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14		16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109
110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129
130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189
190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229
230		232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249
250	251	252	253	254	255				