



Astro Pi: Mission Zero

Valmistaudu Mission Zeroon



Askel 1 Sinä teet

Suorita tämä projekti osallistuaksesi Astro Pi Mission Zero -haasteeseen ja koodisi ajetaan avaruudessa Astro Pi -tietokoneella.

Projektisi asettaa kuvan taustaväriksi sen värin, jonka Astro Pi havaitsee. Näin Kansainvälisestä avaruusasemasta (ISS) tulee entistä värikkäämpi aluksella oleville astronauteille. Koodisi käyttää uuden 2. sukupolven Astro Pi -tietokoneen Sense HATin värin kirkkausanturia tämän toteuttamiseksi.

Tässä on esimerkki ohjelmasta, jonka voit tehdä avaruudessa ajettavaksi Astro Pillä.



Sinä tarvitset

Käytät Astro Pi -emulaattoria verkkoselaimessa ohjelman luomiseen. Et tarvitse Astro Pi -tietokonetta.

Astro Pi Mission Zeron vaatimukset

Jos projektisi täyttää kelpoisuusvaatimukset (<https://astro-pi.org/fi/mission-zero/eligibility>), valmis ohjelmasi ajetaan Kansainvälisellä avaruusasemalla! Saat myös erityisen todistuksen, joka näyttää tarkalleen missä ISS oli ohjelmasi ajon aikana.

Opit Astro Pi -tietokoneesta ja sen hallinnasta, mukaan lukien miten:

- Luot väri **muuttujia** kuvassasi käytettäväksi
- Suunnittelet ja näytät kuvan Sense HATissa
- Tunnistat valon värin ISS:llä



Huomautuksia ohjaajille

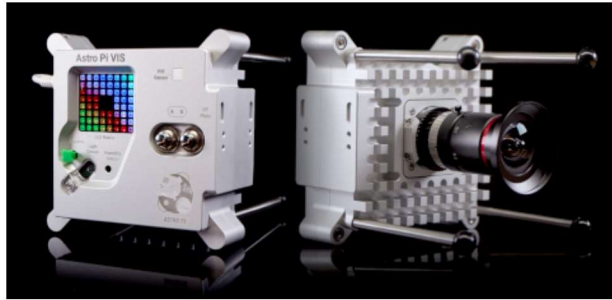
Mission Zero sopii ohjelmoinnin aloittelijoille ja/tai peruskouluikäisillä lapsille, ja voidaan suorittaa kokonaisuudessaan yhden 60 minuutin istunnon aikana millä tahansa tietokoneella, jossa on internet-yhteys. Erikoislaitteistoja tai aikaisempia ohjelmointitaitoja ei tarvita. Kaikki voidaan tehdä verkkoselaimessa.

Muodosta joukkueita, joissa on enintään neljä nuorta, ja anna meidän opastaa heidät kirjoittamaan lyhyt Python-ohjelma, joka tunnistaa ISS:llä olevan värin ja luo sitä käyttävän kuvan.

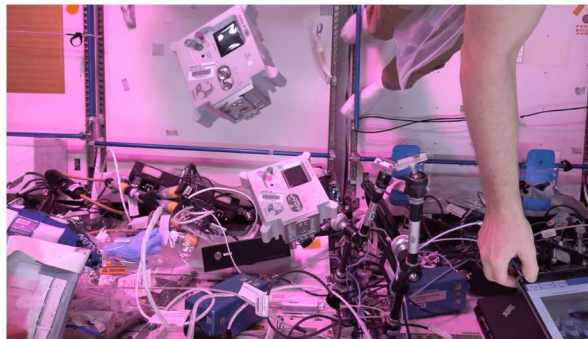
Lue Mission Zeron **viralliset säännöt** (<https://astro-pi.org/fi/mission-zero/guidelines>).

Askel 2 Mikä on Astro Pi?

Astro Pi on Raspberry Pi -tietokone, jonka kotelo on erityisesti suunniteltu avaruusolosuhteisiin.



Astro Pi -tietokoneissa on joukko antureita ja vempaimia, joita voidaan käyttää suorittamaan hienoja tieteellisiä kokeita. Tätä antureiden joukkoa kutsutaan 'Sense HAT':ksi (se tarkoittaa 'Hardware Attached on Top'). Sense HAT antaa Astro Pille kyvyn 'aistia' ja tehdä monenlaisia mittauksia, lämpötilasta liikkeeseen ja tiedon tulostamiseen 8 x 8 LED -matriisia käyttäen. Astro Pissä on myös ohjaussauva ja painikkeita aivan kuten videopelikonsoleissa!

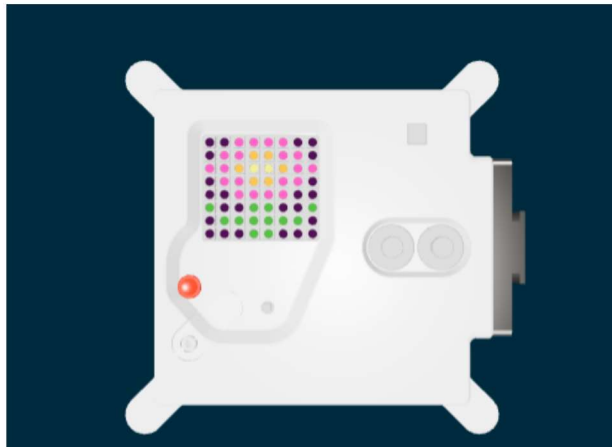


Tätä tehtävää varten käytät Sense HAT -emulaattoria, joka simuloi Astro Pin päätoimintoja verkkoselaimessasi.

Askel 3 Näytä kuva

Astro Pin LED-matriisi voi näyttää värejä. Tässä vaiheessa näytät luontokuvia Astro Pin LED-matriisilla.

LED-matriisi on ruudukko LEDejä, joita voidaan ohjata yhdessä tai erikseen erilaisten valotehosteiden luomiseksi. Sense HATin LED-matriisissa on 64 LEDiä 8 x 8 -ruudukossa. LEDit voidaan ohjelmoida tuottamaan laaja valikoima värejä.



Avaa **Mission Zero -aloitusprojekti** (https://missions.astro-pi.org/fi/mz/code_submissions/new).



Näet, että muutama koodirivi on lisätty sinulle automaattisesti.

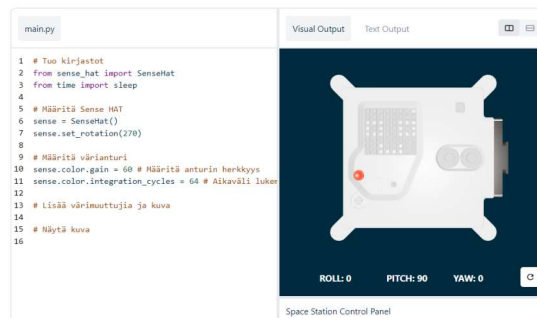
Tämä koodi ottaa yhteyden Astro Pihin, varmistaa Astro Pin LED-näytön olevan oikein päin ja määrittää värianturin. Jätä koodi sinne, koska tulet tarvitsemaan sitä.

main.py

```
# Tuo kirjastot
from sense_hat import SenseHat
from time import sleep

# Määritä Sense HAT
sense = SenseHat()
sense.set_rotation(270)

# Määritä värianturi
sense.color.gain = 60 # Määritä anturin herkkyys
sense.color.integration_cycles = 64 # Aikaväli lukeman ottamiseksi
```



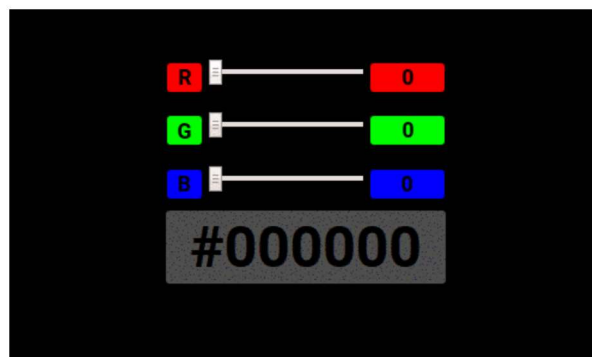
RGB-värit

Värejä voidaan luoda käyttämällä punaista, vihreää ja sinistä eri suhteissa. Voit tutustua RGB-väreihin täältä:



RGB-värit

Kun haluamme esittää väriä tietokoneohjelmassa, voimme tehdä tämän määrittämällä punaisen, sinisen ja vihreän määrät, jotka yhdistettynä muodostavat kyseisen värin. Nämä määrät talletetaan numerona väliltä 0 ja 255.



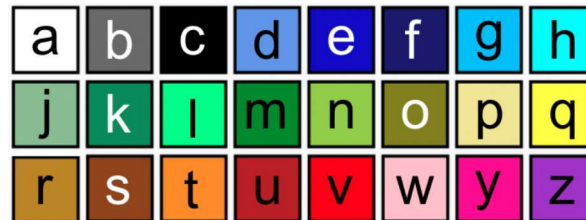
Seuraavassa on taulukko, jossa esitetään joitain väriarvoja:

Punainen Vihreä Sininen Väri

255	0	0	Punainen
0	255	0	Vihreä
0	0	255	Sininen
255	255	0	Keltainen
255	0	255	Purppura
0	255	255	Sinivihreä

Löydät mukavan **värivalitsimen kokeiltavaksi w3schoolsissa** (https://www.w3schools.com/colors/colors_rgb.asp).

LED-matriisi on 8 x 8 -ruudukko. Jokainen ruudukon LED-valo voidaan asettaa eri väriin. Tässä on luettelo muuttujista 24 eri värille. Jokaisella värillä on arvo punaiselle, vihreälle ja siniselle:

**Luettelo värimuuttujista**

main.py

```
# Väripaletti
a = (255, 255, 255) # Valkoinen
b = (105, 105, 105) # Tummanharmaa
c = (0, 0, 0) # Musta
d = (100, 149, 237) # Ruiskukansininen
e = (0, 0, 205) # Tummansininen
f = (25, 25, 112) # Yönsininen
g = (0, 191, 255) # Tummantaivaansininen
h = (0, 255, 255) # Syaani
j = (143, 188, 143) # Tummanmerenvihreä
k = (46, 139, 87) # Merenvihreä
l = (0, 255, 127) # Keväänvihreä
m = (34, 139, 34) # Metsänvihreä
n = (154, 205, 50) # Keltavihreä
o = (128, 128, 0) # Oliivi
p = (240, 230, 140) # Khaki
q = (255, 255, 0) # Keltainen
r = (184, 134, 11) # Kultapuikokeltainen
s = (139, 69, 19) # Satulanruskea
t = (255, 140, 0) # Tummanoranssi
u = (178, 34, 34) # Tulenpunainen
v = (255, 0, 0) # Punainen
w = (255, 192, 203) # Vaaleanpunainen
x = (255, 20, 147) # Tummanvaaleanpunainen
z = (153, 50, 204) # Tummaorkidea
```

Valitse kuva

Valitse: Valitse näytettävä kuva alla olevista vaihtoehdoista. Python tallentaa kuvan tiedot luetteloon. Jokaisen kuvan koodi sisältää käytetyt värimuuttujat ja luettelon.



Sinun on **kopioitava** kaikki valitsemasi kuvan koodi ja sitten **liitettävä** se projektiisi alapuolelle riviä, jolla lukee **# Lisää värimuuttujia ja kuva**.



Kana



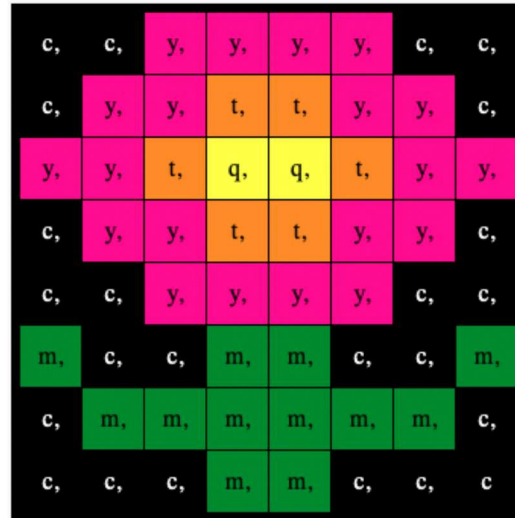
main.py

```
a = (255, 255, 255) # Valkoinen
c = (0, 0, 0) # Musta
e = (0, 0, 205) # Tummansininen
q = (255, 255, 0) # Keltainen
t = (255, 140, 0) # Tummanoranssi
w = (255, 192, 203) # Vaaleanpunainen

kuva = [
    c, c, c, q, q, q, c, c,
    c, c, t, q, e, q, c, c,
    c, c, c, q, q, q, c, c,
    c, w, w, w, w, w, w, c,
    c, w, a, a, a, a, w, c,
    c, w, a, a, a, a, w, c,
    c, c, w, a, a, w, c, c,
    c, c, c, w, w, c, c, c]
```



Kukka



main.py

```

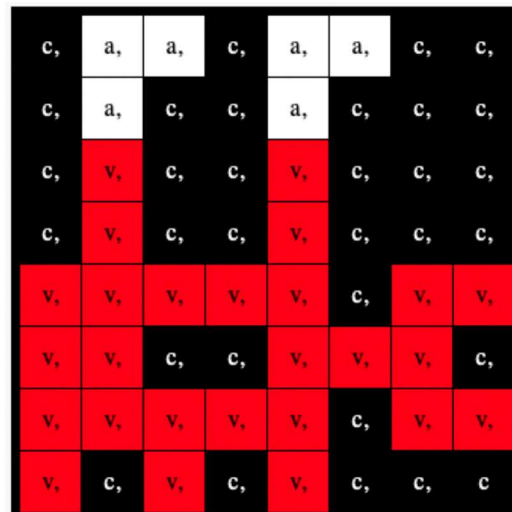
c = (0, 0, 0) # Musta
m = (34, 139, 34) # Metsänvihreä
q = (255, 255, 0) # Keltainen
t = (255, 140, 0) # Tummanoranssi
y = (255, 20, 147) # Tummanvaaleanpunainen

kuva = [
    c, c, y, y, y, y, c, c,
    c, y, y, t, t, y, y, c,
    y, y, t, q, q, t, y, y,
    c, y, y, t, t, y, y, c,
    c, c, y, y, y, y, c, c,
    m, c, c, m, m, c, c, m,
    c, m, m, m, m, m, m, c,
    c, c, c, m, m, c, c, c]

```



Rapu



main.py

```

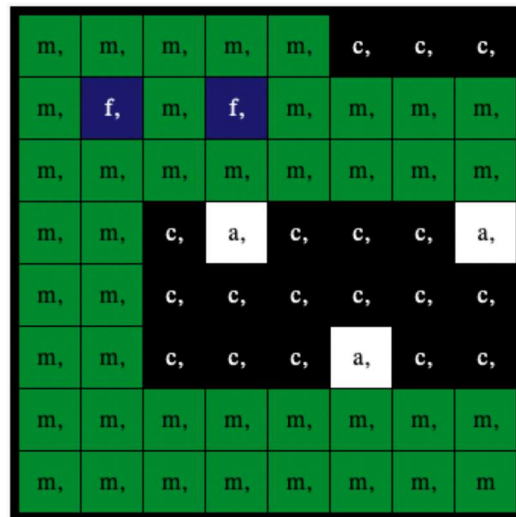
a = (255, 255, 255) # Valkoinen
c = (0, 0, 0) # Musta
v = (255, 0, 0) # Punainen

kuva = [
  c, a, a, c, a, a, c, c,
  c, a, c, c, a, c, c, c,
  c, v, c, c, v, c, c, c,
  c, v, c, c, v, c, c, c,
  v, v, v, v, v, c, v, v,
  v, v, c, c, v, v, v, c,
  v, v, v, v, v, c, v, v,
  v, c, v, c, v, c, c, c]

```



Krokotiili



main.py

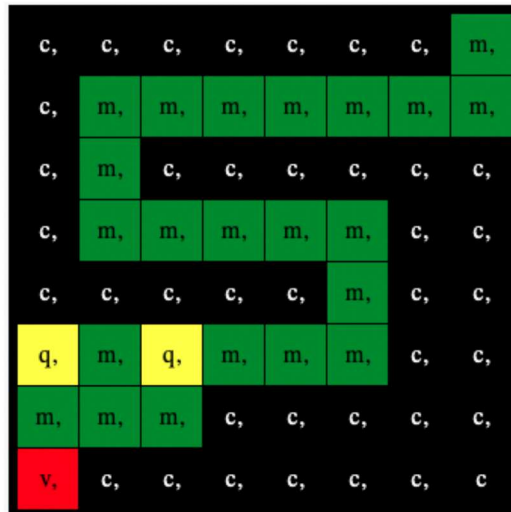
```

a = (255, 255, 255) # Valkoinen
c = (0, 0, 0) # Musta
f = (25, 25, 112) # Yönsininen
m = (34, 139, 34) # Metsänvihreä

kuva = [
  m, m, m, m, m, c, c, c,
  m, f, m, f, m, m, m, m,
  m, m, m, m, m, m, m, m,
  m, m, c, a, c, c, c, a,
  m, m, c, c, c, c, c, c,
  m, m, c, c, c, a, c, c,
  m, m, m, m, m, m, m, m,
  m, m, m, m, m, m, m, m]

```

i Käärme



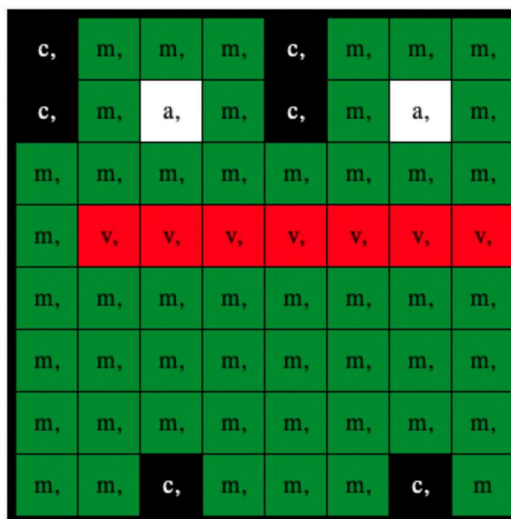
main.py

```

c = (0, 0, 0) # Musta
m = (34, 139, 34) # Metsänvihreä
q = (255, 255, 0) # Keltainen
v = (255, 0, 0) # Punainen

kuva = [
    c, c, c, c, c, c, c, m,
    c, m, m, m, m, m, m, m,
    c, m, c, c, c, c, c, c,
    c, m, m, m, m, m, c, c,
    c, c, c, c, c, m, c, c,
    q, m, q, m, m, m, c, c,
    m, m, m, c, c, c, c, c,
    v, c, c, c, c, c, c, c]
    
```

i Sammakko



main.py

```

c = (0, 0, 0) # Musta
m = (34, 139, 34) # Metsänvihreä
q = (255, 255, 0) # Keltainen
v = (255, 0, 0) # Punainen

kuva = [
  c, m, m, m, c, m, m, m,
  c, m, q, m, c, m, q, m,
  m, m, m, m, m, m, m, m,
  m, v, v, v, v, v, v, v,
  m, m, m, m, m, m, m, m,
  m, m, m, m, m, m, m, m,
  m, m, m, m, m, m, m, m,
  m, m, c, m, m, m, c, m]

```

Etsi: rivi, jossa lukee `# Näytä kuva` ja lisää koodirivi näyttääksesi kuvasi LED-matriisissa:



main.py

```

kuva = [
  c, c, c, q, q, q, c, c,
  c, c, t, q, e, q, c, c,
  c, c, c, q, q, q, c, c,
  c, w, w, w, w, w, w, c,
  c, w, a, a, a, a, w, c,
  c, w, a, a, a, a, w, c,
  c, c, w, a, a, w, c, c,
  c, c, c, w, w, c, c, c]

# Näytä kuva
sense.set_pixels(kuva)

```

Paina **Aja** editorin alaosassa nähdäksesi kuvasi LED-matriisissa.



Vianselvitys



Koodissani on syntaksivirhe:

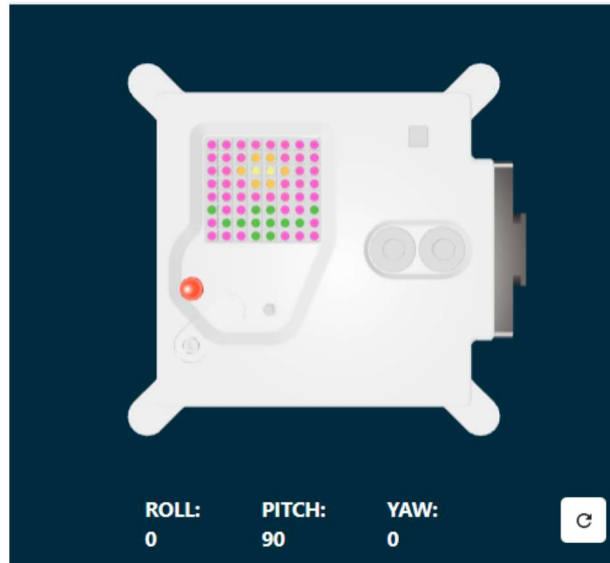
- Tarkista, että koodisi vastaa yllä olevien esimerkkien koodia
- Tarkista, että olet sisentänyt koodin luettelossasi
- Tarkista, että luettelosi ympärillä on `[ja]`
- Tarkista, että jokainen värimuuttuja luettelossa on erotettu pilkulla

Kuvani ei näy:

- Tarkista, että `sense.set_pixels(kuva)` ei ole sisennetty

Askel 4 Tunnista väri

Tässä vaiheessa määrität värin kirkkausanturin ja käytät sitä tunnistaaksesi punaisen, vihreän ja sinisen määrät anturissa. Tätä väriä käytetään sitten valitsemasi kuvan värittämiseen. Astronautti, joka kävelee anturin luo sinisessä paidassa, näkisi erilaisen kuvan kuin punaisessa paidassa oleva astronautti.



Riippumatta valitsemastasi kuvasta, tausta käyttää **c** muuttujaa, joka on asetettu mustaksi.

Käytä värianturia värittääksesi taustasi.



Lisää koodi kuvalistasi eteen saadaksesi värin anturilta ja muuta taustavärisi muuttuja `c` käyttääksesi Sense HATin värianturin tunnistamaa väriä mustan sijasta.

Vinkki: Sinun ei tarvitse kirjoittaa #-alkuisia kommentteja (ne ovat siellä koodin selittämiseksi).

main.py

```
# Lisää värimuuttujia ja kuva

c = (0, 0, 0) # Musta
m = (34, 139, 34) # Metsänvihreä
q = (255, 255, 0) # Keltainen
t = (255, 140, 0) # Tummanoranssi
y = (255, 20, 147) # Tummanvaaleanpunainen

rgb = sense.color # hae väri anturilta
c = (rgb.red, rgb.green, rgb.blue) # käytä tunnistettua väriä

kuva = [
    c, c, y, y, y, y, c, c,
    c, y, y, t, t, y, y, c,
    y, y, t, q, q, t, y, y,
    c, y, y, t, t, y, y, c,
    c, c, y, y, y, y, c, c,
    m, c, c, m, m, c, c, m,
    c, m, m, m, m, m, c,
    c, c, c, m, m, c, c, c]
```

Kokeile: Siirrä värin liikusäädin valitsemaasi väriin ja sitten **aja** koodisi. Taustavärisi muuttuu. Toista tämä kokeilu uudella värillä.



Vinkki: Sinun on napsautettava 'Aja' aina, kun muutat väriä.

Aja ohjelmaasi silmukassa

Astro Pi Mission Zero -ohjelma saa olla ajossa enintään 30 sekuntia. Käytät tämän ajan toistuvasti värianturin tarkastamiseen ja kuvan päivittämiseen.

Koodisi käyttää `for`-silmukkaa ajaakseen 28 kertaa. **Joka** kerta se:

- tunnistaa uusimman värin
- päivittää kuvan taustavärin
- odottaa yhden sekunnin

Etsi sinun `rgb = sense.color` koodirivi.



Lisää koodi sen yläpuolelle määrittääksesi `for`-silmukan `28` toistolle.

main.py

```
for i in range(28):
    rgb = sense.color # hae väri anturilta
    c = (rgb.red, rgb.green, rgb.blue)

    kuva = [
        c, c, y, y, y, y, c, c,
        c, y, y, t, t, y, y, c,
        y, y, t, q, q, t, y, y,
        c, y, y, t, t, y, y, c,
        c, c, y, y, y, y, c, c,
        m, c, c, m, m, c, c, m,
        c, m, m, m, m, m, m, c,
        c, c, c, m, m, c, c, c]
```

Sinun on nyt sisennettävä kaikki `for`-silmukan alapuolella oleva koodi niin, että se sijoittuu `for`-silmukan **sisälle**.



Vinkki: Sisentääksesi useita rivejä, ensin valitse sisennettävät rivit, sitten paina näppäimistösi Tab-näppäintä (yleensä Caps Lock-näppäimen yläpuolella).

main.py

```
2 for i in range(28):
    rgb = sense.color # hae väri anturilta
    c = (rgb.red, rgb.green, rgb.blue)

    kuva = [
        c, c, y, y, y, y, c, c,
        c, y, y, t, t, y, y, c,
        y, y, t, q, q, t, y, y,
        c, y, y, t, t, y, y, c,
        c, c, y, y, y, y, c, c,
        m, c, c, m, m, c, c, m,
        c, m, m, m, m, m, m, c,
        c, c, c, m, m, c, c, c]

# Näytä kuva
17 sense.set_pixels(kuva)
```

Koodisi loppuun lisää yhden sekunnin mittainen `sleep` silmukasi sisälle:



main.py

```
# Näytä kuva

sense.set_pixels(kuva)
sleep(1)
```

Vinkki: Varmista, että tämä koodirivi on sisennettynä `for`-silmukassasi.

Kokeile: Aja koodisi ja vaihda värivalitsinta useita kertoja projektisi suorituksen aikana. Tarkista, että kuvasi päivittyvä havaitun värin mukaisesti seuraavalla suorituskerralla.



Kuvan päivitys lopetetaan silmukan päätyttyä, jotta ohjelman ajo ei kestä yli 30 sekuntia.

Viian selvitys



Koodissani on syntaksivirhe tai ei toimi odotetulla tavalla:

- Tarkista, että koodisi vastaa yllä olevien esimerkkien koodia
- Tarkista, että olet sisentänyt koodin `for`-silmukassasi
- Tarkista, että luettelosi ympärillä on `[ja]`
- Tarkista, että jokainen värimuuttuja luettelossa on erotettu pilkulla

Koodini toimii yli 30 sekuntia:

- Vähennä `for`-silmukan suorituskertojen lukumäärää 28:sta 25:een tai jopa 20:een.
- Lyhennä `sleep`in kestoja 1 sekunnista 0.5 sekuntiin.

Lisää `sense.clear()` koodin loppuun kuvan tyhjentämiseksi silmukan lopussa. Tämä auttaa sinua näkemään, milloin animaatiosi on päättynyt.



Vinkki: Varmista, että **et** sisennä `sense.clear()`-koodiriviä, koska haluat sen suoritettavan vain yhden kerran animaation lopussa.

main.py

```
# Näytä kuva

sense.set_pixels(kuva)
sleep(1)

sense.clear()
```

Kokeile: Aja koodisi uudelleen. Kun projektisi on suoritettu, LED-matriisi tyhjenee kaikkien valojen vaihtuessa mustaksi (sammuvat).




Vianselvitys



LED-matriisi muuttuu mustaksi joka sekunti:

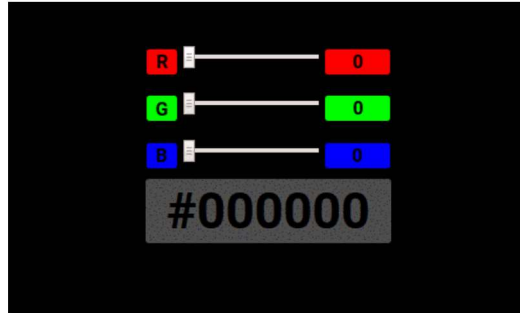
- Tarkista, että et ole sisentänyt `sense.clear()`-koodia `for`-silmukkasi sisälle

Lisää koodia LED-matriisiin tyhjentämiseksi valitsemallasi värillä. Luo muuttuja nimeltä `x` tallentaaksesi uuden värisi. 

Voit sekoittaa oman värisi tai käyttää väriluettelon arvoja luodaksesi uuden `x`-värisi.

RGB-värit

Kun haluamme esittää väriä tietokoneohjelmassa, voimme tehdä tämän määrittämällä punaisen, sinisen ja vihreän määrät, jotka yhdistettynä muodostavat kyseisen värin. Nämä määrät talletetaan numerona väliltä 0 ja 255.

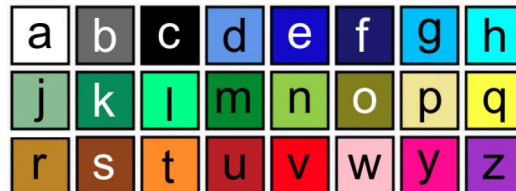


Seuraavassa on taulukko, jossa esitetään joitain väriarvoja:

Punainen	Vihreä	Sininen	Väri
255	0	0	Punainen
0	255	0	Vihreä
0	0	255	Sininen
255	255	0	Keltainen
255	0	255	Purppura
0	255	255	Sinivihreä

Löydät mukavan **värivalitsimen** kokeiltavaksi **w3schoolsissa** (https://www.w3schools.com/colors/colors_rgb.asp).

Luettelo värimuuttujista



main.py

```
# Väripaletti
a = (255, 255, 255) # Valkoinen
b = (105, 105, 105) # Tummanharmaa
c = (0, 0, 0) # Musta
d = (100, 149, 237) # Ruiskukansininen
e = (0, 0, 205) # Tummansininen
f = (25, 25, 112) # Yönsininen
g = (0, 191, 255) # Tummantaivaansininen
h = (0, 255, 255) # Syaani
j = (143, 188, 143) # Tummanmerenvihreä
k = (46, 139, 87) # Merenvihreä
l = (0, 255, 127) # Keväänvihreä
m = (34, 139, 34) # Metsänvihreä
n = (154, 205, 50) # Keltavihreä
o = (128, 128, 0) # Oliivi
p = (240, 230, 140) # Khaki
q = (255, 255, 0) # Keltainen
r = (184, 134, 11) # Kultapuikonkeltainen
s = (139, 69, 19) # Satulanruskea
t = (255, 140, 0) # Tummanoranssi
u = (178, 34, 34) # Tulenpunainen
v = (255, 0, 0) # Punainen
w = (255, 192, 203) # Vaaleanpunainen
v = (255, 20, 147) # Tummanvaaleanpunainen
z = (153, 50, 204) # Tummaorkidea
```

main.py

```
# Näytä kuva

sense.set_pixels(kuva)
sleep(1)

x = (178, 34, 34) # valitse omat punaisen, vihreän ja sinisen arvot väliltä 0 - 255
sense.clear(x)
```

Kokeile: Aja koodisi uudelleen. Kun projektisi on suoritettu, LED-matriisi tyhjenee valittuun väriisi. Voit vaihtaa ja kokeilla väriä niin monta kertaa kuin haluat.





Valmis koodiesimerkki



```
c,  c,  y,  y,  y,  y,  c,  c,
c,  y,  y,  t,  t,  y,  y,  c,
y,  y,  t,  q,  q,  t,  y,  y,
c,  y,  y,  t,  t,  y,  y,  c,
c,  c,  y,  y,  y,  y,  c,  c,
m,  c,  c,  m,  m,  c,  c,  m,
c,  m,  m,  m,  m,  m,  m,  c,
c,  c,  c,  m,  m,  c,  c,  c
```

main.py

```
# Tuo kirjastot
from sense_hat import SenseHat
from time import sleep

# Määritä Sense HAT
sense = SenseHat()
sense.set_rotation(270)

# Määritä värianturi
sense.color.gain = 60 # Määritä anturin herkkyys
sense.color.integration_cycles = 64 # Aikaväli lukeman ottamiseksi

# Lisää värimuuttujia ja kuva

c = (0, 0, 0) # Musta
m = (34, 139, 34) # Metsänvihreä
q = (255, 255, 0) # Keltainen
t = (255, 140, 0) # Tummanoranssi
y = (255, 20, 147) # Tummanvaaleanpunainen

for i in range(28):
    rgb = sense.color # hae väri anturilta
    c = (rgb.red, rgb.green, rgb.blue)

    kuva = [
        c, c, y, y, y, y, c, c,
        c, y, y, t, t, y, y, c,
        y, y, t, q, q, t, y, y,
        c, y, y, t, t, y, y, c,
        c, c, y, y, y, y, c, c,
        m, c, c, m, m, c, c, m,
        c, m, m, m, m, m, c,
        c, c, c, m, m, c, c, c]

# Näytä kuva

sense.set_pixels(kuva)
sleep(1)

x = (178, 34, 34) # valitse omat punaisen, vihreän ja sinisen arvot väliltä 0 - 255
sense.clear(x)
```

Askel 5 Lähetä ohjelmasi

Voit nyt osallistua **Astro Pi Mission Zero** (<https://astro-pi.org/fi/mission-zero>) -haasteeseen kirjoittamalla koodilla.

On olemassa muutamia sääntöjä, joita koodisi on noudatettava, jotta voit lähettää sen suoritettavaksi Kansainvälisellä avaruusasemalla. Jos koodisi noudattaa niitä, säännöt **Sense HAT -emulaattorin** alareunassa syttyvät vihreinä ohjelmaa ajaessasi.

Sinun ohjelmasi täytyy:

1. Toimia ilman virheitä
2. Käyttää värin ja kirkkauden anturia
3. Suoriutua enintään 30 sekunnissa
4. Käyttää LEDejä

Tämä tarkistetaan automaattisesti joka kerta, kun painat 'Aja'.

Varmistathan, että joukkueesi nimi, ohjelma ja kuvat noudattavat **Mission Zeron virallisia sääntöjä**. Muuten ohjelmaasi ei voida ajaa Kansainvälisellä avaruusasemalla.

Vinkki: Kokeile koodiasi muutamalla eri väriasetuksella (valitsimen avulla) varmistaaksesi, että se toimii aina oikein.

Varmista, että työsi noudattaa Mission Zeron **virallisia sääntöjä** (<https://astro-pi.org/fi/mission-zero/guidelines>). Jos se ei noudata sääntöjä, ohjelmaasi ei voida ajaa Kansainvälisellä avaruusasemalla.

Älä sisällytä joukkueesi nimeen tai koodiin mitään seuraavista:

- Kaikki mikä voidaan tulkita laittomaksi, poliittiseksi tai arkaluonteiseksi
- Lippuja, koska niitä voidaan pitää poliittisesti arkaluonteisina
- Kaikki mikä viittaa toisen henkilön haittaamiseen tai vahingoittamiseen
- Henkilötietoja kuten puhelinnumeroita, sosiaalisen median tunnuksia tai sähköpostiosoitteita
- Siveettömiä kuvia
- Erikoismerkkejä tai hymiöitä

- Huonoa kieltä tai kiroilua

Kirjoita luokkahuonekoodisi ja joukkueesi nimi alhaalla olevaan ruutuun – ohjaajasi kertoo sinulle, mikä koodisi on.



SUBMIT YOUR PROGRAM

Once your program is ready, enter your classroom code to continue to the submission form. Mentors need to register for a classroom code at astro-pi.org/mission-zero.



Run your experiment to make sure it meets the criteria

Classroom code

Team name

Your team name can be up to 8 characters long and can only contain alphanumeric characters and underscores. Please note that after you click on 'Add your team', you will not be able to change your team name.

Add your team

Huomautuksia ohjaajille löytyy **Johdanto** (<https://projects.raspberrypi.org/fi-FI/projects/astro-pi-mission-zero/0>)-vaiheesta.

Paina **Lisää joukkueesi** -painiketta syöttääksesi koodisi. Huomioithan, että ohjelmaa ei voi muuttaa enää lähettämisen jälkeen.

Ohjaajasi saa sähköpostin vahvistukseksi ilmoittautumisestasi.

Jos haluat, voit jakaa linkin koodiisi sosiaalisessa mediassa kertoaksesi, että koodia, jonka kirjoitit, käytetään avaruudessa!



Askel 6 Mitä seuraavaksi – lisää Astro Pi -projekteja

Nyt, kun olet suorittanut tehtäväsi, miksi et kokeilisi lisää projekteja Astro Pin muita antureita käyttäen?

Jos tunnet olosi varmaksi, voit osallistua **Mission Space Labiin** (<https://astro-pi.org/missions/space-lab/>)! Muodostakaa joukkue, jossa on kahdesta kuuteen henkilöä, ja työskennelkää yhdessä kuin oikeat avaruustieteilijät ja suunnitelkaa oma kokeenne. Voit saada Astro Pi -pakkauksen tehtäväsi edesauttamiseen, jos ideasi on parhaiden joukossa.

Vaihtoehtoisesti voitte kokeilla joitain muita Astro Pi -projektejamme:

- Opi **lisää Sense HATista** (<https://projects.raspberrypi.org/fi-FI/projects/getting-started-with-the-sense-hat>) ja muista asioista, joita se voi tehdä
- Luo kauniita **satunnaisia kimalluksia** (<https://projects.raspberrypi.org/fi-FI/projects/sense-hat-random-sparkles>) Sense HATin LED-näytöllä
- Luo **Flappy Astronaut** (<https://projects.raspberrypi.org/fi-FI/projects/flappy-astronaut>) peli
- Haasta ystäväsi **marmorilabyrintti** (<https://projects.raspberrypi.org/fi-FI/projects/sense-hat-marble-maze>) -peliin
- Tee uusiksi klassinen peli **Pong** (<https://projects.raspberrypi.org/fi-FI/projects/sense-hat-pong>)

Tämän projektin ovat kääntäneet vapaaehtoiset:

Erno Pentzin

Vapaaehtoisten ansiosta ihmisillä ympäri maailma on mahdollisuus oppia omalla kielellään. Voit auttaa meitä tavoittamaan enemmän ihmisiä vapaaehtoistyönä kääntämällä - lisätietoja osoitteessa rpf.io/translate (rpf.io/translate).

Julkaissut **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) lisenssillä **Creative Commons license** (<https://creativecommons.org/licenses/by-sa/4.0/>).

Näytä projekti ja lisenssi GitHubissa (<https://github.com/RaspberryPiLearning/astro-pi-mission-zero>)