

Inhoudsopgave

1	Configuratie	1.1
2	Licht, Actie!	1.2
3	Je eigen circuit naaien	1.3
4	Je eigen code schrijven	1.4
5	Kleur en licht	1.5
6	Knipperend licht	1.6
7	Animeer!	1.7
8	Regenboog magie	1.8

Waarschuwing: In dit project wordt gewerkt met knipperende lichten wat gevolgen zou kunnen hebben voor mensen met lichtgevoelige epilepsie.

- 1 In dit project naai je LED lampjes op een t-shirt en programmeer je ze om te knipperen en van kleur te veranderen!
- 2 De LEDs worden aangestuurd door de Adafruit Flora. Je kunt ook de Adafruit Gemma, LilyPad Arduino of LilyPad Arduino USB gebruiken. Het kan dan wel voorkomen dat de code dan aangepast moet worden (bijv. de output pin en de configuratie van het board in Arduino IDE).
 - **NB:** De Gemma werkt niet met Linux, en ook niet op een USB 3.0 poort. Zorg dan voor een USB 2.0 poort of connectie.
- 3 Download en installeer Arduino IDE via dojo.soy/wear2-arduino-ide. Open het programma als het geïnstalleerd is. Er volgen nog wat extra zaken die nodig zijn voor dit project.
- 4 Open in het menu **Bestand** de **Voorkeuren**. Zet bij **Additionele Bordenbeheer URLS** het volgende neer en klik dan op OK.

```
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
```
- 5 Ga naar het **Hulpmiddelen** menu, dan naar **Board "Adafruit Flora"** en dan naar **Bordenbeheer**. Bij **Type** klik je op **Bijgedragen**. Installeer **Adafruit AVR boards by Adafruit**. Klik daarna op "Sluiten".
- 6 Sluit Arduino IDE af en start het opnieuw. Bij Boards zou je nu moeten zien: **Adafruit Flora**, **Adafruit Gemma**, **LilyPad Arduino** en **LilyPad Arduino USB**. Selecteer het board dat je gaat gebruiken.
- 7 Ga naar het **Schets** menu, dan naar **Bibliotheek gebruiken**. Klik op **Bibliotheken beheren** en in het zoekvenster typ je `neopixel`. Installeer **Adafruit NeoPixel by Adafruit**. Klik daarna op "Sluiten".
- 8 Je hebt de volgende materialen nodig voor dit project
 - Adafruit Flora of Gemma en een USB kabel
 - Acht NeoPixels
 - Geleidend draad
 - Drie krokodillenklemmen (je kunt ook stukjes geleidend draad gebruiken, maar krokodillenklemmen werken makkelijk voor het testen)
 - Optioneel: een batterijhouder en knoopbatterij zodat je je project kunt dragen zonder aan de computer vast te zitten!
 - Naald en schaar
 - Een t-shirt
 - Blanco nagellak
 - Optioneel: een borduurring om het naaien van je circuit makkelijker te maken

1 Voordat je begint is het handig om je NeoPixels te testen. Open Arduino IDE. Kijk nog even na welk board je gebruikt. In de voorbeelden gebruiken we *Adafruit Flora*.

2 Ga naar **Bestand**, selecteer **Voorbeelden** en zoek **Adafruit NeoPixel** (let op: deze kan helemaal onderaan staan!) en kies **strandtest**.

3 Een nieuw scherm wordt geopend. Het codebestand heet een **schets** in Arduino IDE. Zoek deze coderegel:

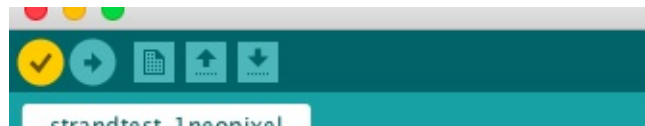
```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(60, PIN, NEO_GRB + NEO_KHZ800);
```

4 Verander het eerste getal naar **1**. Nu moet de code er zo uitzien:

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
```

5 Klik op **Bestand** > **Opslaan als**. Typ een naam voor jouw schets en sla het op.

6 Boven in je **schets** klik je op **Verifiëren**. Onderin je scherm moet komen te staan "Compileren voltooid" wat betekent dat de code juist is **gecompileerd** (zo niet, dan zie je hier de foutmeldingen. Je moet de fouten dan gaan verbeteren).



7 Nu gaan we uploaden! Koppel je Flora aan de computer. Druk op de **reset** knop op de Flora en dan **meteen**, terwijl het rode lichtje knippert, klik je op het pijlcoon om de code naar je Flora te uploaden. Het rode lichtje moet knipperen, gevolgd door twee oranje lichtjes op de Flora. Als het klaar is, zou de tekst "Upload voltooid" te zien moeten zijn onderin je schets.

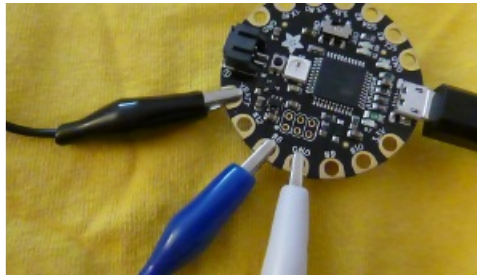


- Soms lukt het uploaden niet meteen. Zorg dat het juiste board geselecteerd is, dat je USB kabel werkt en goed gekoppeld zit. Dan gaat het alleen nog om timing! Je krijgt het vanzelf voor elkaar.

8 Koppel Flora los van je computer (de aan/uit knop op een Flora kan gebruikt worden als deze is aangesloten op een batterijhouder, maar werkt niet als de Flora via USB aan een computer gekoppeld is).

- Koppel Flora los of zet het uit voordat je andere onderdelen vast- of loskoppelt zodat er niets beschadigt!

- 9 Klem drie krokodillenklemmen aan de **GND**, **#6** en **VBATT** pinnen.



- 10 Pak een NeoPixel en verbind de **GND** draad aan de - pin. Verbind de **#6** aan de **data in** pin: dit is de pin met de pijl die naar binnen wijst op de NeoPixel. Verbind tenslotte de **VBATT** aan de + pin.



- 11 Klaar? Zet Flora weer aan en zie hoe je NeoPixel oplicht en in meerdere kleuren knippert!

- 12 Test al je NeoPixels door ze aan te sluiten zoals beschreven bij stap 10. Onthoud dat je Flora **uitzet** voor de de draden vast- of loskoppelt!

- 13 Als je de NeoPixels getest hebt, verander je de code naar het aantal pixels dat je gaat gebruiken. Ik gebruik er acht:

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(8, PIN, NEO_GRB + NEO_KHZ800);
```

- 14 Klik op **Verifieer** en **upload** de nieuwe code naar je Flora. Nu gaan we een NeoPixel circuit maken!

1 Verzamel je NeoPixels en pak textielkrijt of een viltstift. Leg een t-shirt (of een ander stuk stof dat je voor je project gebruikt) op een vlakke ondergrond. Leg de pixels in de vorm die jij wilt. Ik ga een lachend gezichtje maken! Andere ideeën zijn:

- o Een rechte lijn
- o Een hart (daar kun je 8 NeoPixels goed voor gebruiken)
- o Nep "knopen" aan de voorkant van een t-shirt

2 Bepaal waar je Flora moet komen en kies een NeoPixel waar je Flora aan wilt koppelen: deze ga je er als eerste opnaaien. De rest wordt door een keten aan elkaar gekoppeld. Je zou een ononderbroken lijn moeten kunnen volgen met je vinger: zodanig dat de lijn zich nergens kruist (kortsluiting!).

3 Teken de buiten- en binnenkant met textielkrijt en markeer waar elke NeoPixel komt.



4 Klaar om te gaan naaien? Als je alles gemarkeerd hebt, leg de spullen dan aan de kant en pak naald en geleidedraad. Als je pixels dicht bij elkaar liggen dan zou 20 cm draad genoeg moeten zijn. Als je een borduurring hebt dan kan dat het naaien een stuk makkelijker maken.

5 Eerst ga je de **data** lijn naaien. Leg de eerste pixel op zijn plek met de pijltjes in de richting van waar de tweede pixel moet komen. Zet de eerste pixel vast met de pin met de pijl die **weg** wijst van de LED in het midden. Dit is de **output** pin.

- o Zet drie of vier steken door de pin om hem goed aan de stof vast te zetten.

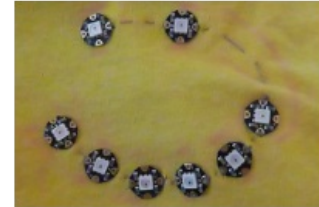
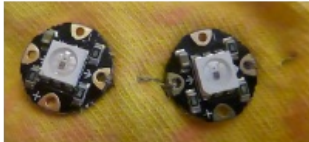
6 Gebruik een rijgsteek tot je bij de plek komt waar je tweede pixel komt. Pak dan die tweede pixel en leg het op de juiste plek, met de pijltjes wijzend naar de plek van de derde pixel.

7 Naai de tweede pixel vast via de **input** pin (onthoud: dit is de pin waar de pijl **naar** de LED in het midden wijst). Zet de draad vast met een paar steken aan de achterkant van de stof en knip het restant af.

- o Tip: gebruik blanco nagellak om de draadeinden vast te zetten; zo zal het niet gaan rafelen en kunnen losse draadjes geen kortsluiting veroorzaken.

8 Gebruik een *nieuw stuk geleidedraad*, verbind de **output** pin van de tweede pixel aan de **input** pin van de derde pixel. Ga zo door tot alle pixels aan elkaar zitten via hun **data** pins. Gebruik steeds een nieuwe draad tussen de pixels.

- De keten stopt bij de laatste pixel; je zet niets vast aan diens **output** pin.



9 Nu ga je alle **negatieve** pins in de keten verbinden, gebruik hiervoor een stuk draad van 50-100 cm. Naai een paar strakke steken door elke - pin van elke pixel. Begin bij de eerste en eindig bij de laatste. gebruik een rijgsteek tussen de pixels.

- Zorg dat deze draad nergens de **data** draad raakt!

10 Gebruik een lang stuk draad om alle **+** pinnen van de pixels te verbinden, net zoals je de - pinnen in stap 9 verbonden hebt.

11 Tot slot leg je de Flora op je t-shirt (zorg dat Flora *niet* aanstaat!). Gebruik drie verschillende stukken geleidedraad. Verbind de **#6** pin aan de **input**, de **GND** pin aan de - en de **VBATT** pin aan de + pin van de eerste NeoPixel. Gebruik weer een rijgsteek. Zorg dat geen van de draden elkaar raken. Je kunt andere pins van de Flora met gewoon naaigaren vastzetten op de stof om het steviger op zijn plek te zetten.



12 Het moment van de waarheid: tijd om je Flora aan te zetten! Schrik niet als niet alle pixels oplichten. Dat kan veroorzaakt worden door:

- Kortsluiting: raken draden elkaar? Raakt iets van metaal de stof of het circuit? Is de stof nat?
- Losse verbindingen: de steken aan elke pin moeten stevig vastzitten voor een goede verbinding.
- Juiste code ge-upload: heeft je code het juiste aantal pixels? Is de code geverifieerd en ge-upload zonder foutmeldingen?

1 In Arduino IDE, klik op **Bestand > Nieuw**. Je ziet een nieuwe **sketch** die er zo uit moet zien:

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Elke regel die start met `//` is een **comment** (= commentaar). Comments worden door de computer genegeerd. Ze zijn handig om notities te maken voor jezelf of anderen.

2 Ga naar **Schets > Bibliotheek gebruiken** en selecteer **Adafruit NeoPixel**. Je zou nu deze coderegel boven in je schets moeten zien: `#include <Adafruit_NeoPixel.h>`. Klik aan het eind van die regel en klik een paar keer op Enter om een paar blanco regels in te voegen.

3 Onder deze nieuwe coderegel typ je `#define PIXELS_PIN 6`. Je vertelt nu welke pin van de Flora gebruikt gaat worden voor **data** (instructies). Dit is de pin die de **data** pinnen van de pixels verbindt: pin nummer 6.

4 Daaronder typ je `#define NUM_PIXELS 8`. Dit is het aantal NeoPixels dat je gebruikt. Als je een ander aantal dan 8 gebruikt, typ dan dat getal.

5 Tenslotte typ je deze regel:

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_PIXELS, PIXELS_PIN, NEO_GRB + NEO_KHZ800);
```

6 Binnen de **setup** functie voeg je de volgende twee coderegels toe:

```
void setup() {  
    // put your setup code here, to run once:  
  
    strip.begin();  
    strip.show();  
  
}
```

- o de code in de **setup** functie wordt uitgevoerd als de Flora opstart.

7 Na `strip.show();` klik je op Enter en typ je vervolgens deze twee regels:

```
strip.setPixelColor(0, strip.Color(0, 0, 255));  
strip.show();
```

8 Klik op **Verifiëren** om je code te compileren en op fouten na te kijken. Als er fouten zijn, verbeter ze dan. Meestal zie je staan in welke regel de fout zit. Zorg dat je precies typt wat er in deze kaarten staat!

9 Koppel Flora aan je computer en voer je code uit. Klik op de **reset** knop op de Flora en dan op de **Upload** knop. Wat gebeurt er als het uploaden klaar is?

10 Hopelijk licht de eerste pixel blauw op. Nu gaan we de volgende doen! Boven de tweede `strip.show();` typ je de volgende twee regels:

```
strip.setPixelColor(1, strip.Color(0, 0, 255));  
strip.setPixelColor(2, strip.Color(0, 0, 255));
```

11 De **setup** functie zou er nu zo uit moeten zien:

```
void setup() {  
    // put your setup code here, to run once:  
    strip.begin();  
    strip.show();  
    strip.setPixelColor(0, strip.Color(0, 0, 255));  
    strip.setPixelColor(1, strip.Color(0, 0, 255));  
    strip.setPixelColor(2, strip.Color(0, 0, 255));  
    strip.show();  
}
```

Snap je wat deze code doet?

12 Verifieer en upload je code. Nu zouden de eerste drie pixels blauw moeten oplichten. Schrijf zelf de coderegels voor de andere pixels!



- 1 Je gaat nu je eigen **function** (= functie) schrijven. **Functions** houden je code netjes. Onderaan de schets, klik **na** de `}` (dus buiten de `loop` functie) en klik een paar keer op Enter. typ nu de volgende code:

```
void lightAll() {  
  
    strip.setPixelColor(0, strip.Color(0, 0, 255));  
    strip.setPixelColor(1, strip.Color(0, 0, 255));  
    strip.setPixelColor(2, strip.Color(0, 0, 255));  
    strip.setPixelColor(3, strip.Color(0, 0, 255));  
    strip.setPixelColor(4, strip.Color(0, 0, 255));  
    strip.setPixelColor(5, strip.Color(0, 0, 255));  
    strip.setPixelColor(6, strip.Color(0, 0, 255));  
    strip.setPixelColor(7, strip.Color(0, 0, 255));  
  
    strip.show();  
  
}
```

Alle code binnen een **function** wordt binnen **accolades** `{ }` gezet.

- 2 Verander nu je **setup** code zodat het er zo uit ziet:

```
void setup() {  
  
    // put your setup code here, to run once:  
  
    strip.begin();  
    strip.show();  
  
    lightAll();  
  
}
```

De laatste regel **roept** de function die je gemaakt hebt **aan**. Het vertelt de functie om te beginnen.

- 3 Verifieer en upload je schets naar Flora. Lichten alle pixels blauw op?

- Je zult vast gezien hebben dat het eerste getal in de regel `strip.setPixelColor(0, strip.Color(0, 0, 255));` bepaalt welke pixel oplicht. Zie je dat de eerste pixel nul is in plaats van één? Dus als je acht pixels hebt is je laatste getal `7`.

- 4 Verander de tweede regel van `lightAll` functie van

```
strip.setPixelColor(1, strip.Color(0, 0, 255));
```

naar

```
strip.setPixelColor(1, strip.Color(255, 0, 0));
```

Verifieer en upload je code naar Flora. Zie je het verschil?

5 Op een computer worden kleuren gemaakt door de drie **primaire kleuren van licht**, *rood*, *groen* en *blauw* te mengen. Je gebruikt getallen van 0 tot 255 om te vertellen hoeveel van elke kleur gemengd moet worden. Dus de code `strip.Color(0, 0, 255)` maakt *blauw* omdat de waarden van rood en groen beide nul zijn.

- Welke kleur geeft `strip.Color(0, 255, 0)` denk je? Probeer het uit!

6 Hier zijn nog wat kleuren die handig zijn om te weten:

```
void lightAll() {
    strip.setPixelColor(0, strip.Color(0, 0, 255)); // blue
    strip.setPixelColor(1, strip.Color(255, 0, 0)); // red
    strip.setPixelColor(2, strip.Color(0, 255, 0)); // green
    strip.setPixelColor(3, strip.Color(255, 0, 255)); // magenta
    strip.setPixelColor(4, strip.Color(255, 255, 255)); // white
    strip.setPixelColor(5, strip.Color(255, 255, 0)); // yellow
    strip.setPixelColor(6, strip.Color(0, 255, 255)); // cyan
    strip.setPixelColor(7, strip.Color(255, 127, 0)); // orange
    strip.show();
}
```

7 Experimenteer met de getallen voor verschillende kleuren. Wat denk je dat er gebeurt als je de waarde op 0 zet voor alle kleuren `strip.Color(0, 0, 0)` ?

8 Zie je al sterretjes?! NeoPixels zijn echt FEL, niet waar! Gelukkig kun je de felheid van de pixels veranderen met deze code: `strip.setBrightness(10);` . Voeg het toe aan de *setup* functie, tussen de regels `strip.begin();` en `strip.show();` . Net als met kleuren kun je getallen tussen 0 en 255 gebruiken.

9 Het kan zijn dat de kleuren er aan het eind van de keten niet zo goed uitzien. Dat komt omdat het circuit aan kracht verliest door **weerstand** in de draad. Je kunt meer kracht toevoegen door een extra draad te naaien langs zowel de **negatieve** als **positieve** lijnen in je circuit.

1 Schrijf deze nieuwe functie na je eerste:

```
void lightAllOneColour(uint32_t c) {
    strip.setPixelColor(0, c);
    strip.setPixelColor(1, c);
    strip.setPixelColor(2, c);
    strip.setPixelColor(3, c);
    strip.setPixelColor(4, c);
    strip.setPixelColor(5, c);
    strip.setPixelColor(6, c);
    strip.setPixelColor(7, c);
    strip.show();
}
```

- Deze functie neemt een **parameter**: dat is het stukje binnen de ronde haakjes (). Het is wat extra informatie die je aan de functie geeft als je het aanroept.

2 Nu ga je een functie schrijven die **loop** (= lus) heet in plaats van **setup**. Klik binnen de **loop** functie en voeg code toe zodat het er zo uit ziet:

```
void loop() {
    lightAllOneColour(strip.Color(0, 0, 255));
    delay(200);
    lightAllOneColour(strip.Color(0, 0, 0));
    delay(200);
}
```

Zie je hoe je een kleur als **parameter** in je functie zet? Dit is de kleur die gebruikt wordt in plaats van `c` op elke regel van je functie `lightAllOneColour`. Dit betekent dat je dezelfde functie kunt gebruiken om pixels elke kleur te laten zijn, zelfs om ze allemaal uit te zetten!

3 Verwijder de regel `lightAll();` in de **setup** functie. Verifieer en upload de code.

- Als Flora start, loopt het alle code binnen de **setup** functie eerst door en blijft dan de **loop** functie eeuwig herhalen!

4 Wat denk je dat de **delay** (= uitstel) functie doet? Zet daar eens verschillende waarden in de **parameter**, bijvoorbeeld

`delay(50);` or `delay(1000);`. Vergeet niet om je code te verifiëren en uploaden om je aanpassingen te testen!

5

Is het je opgevallen dat de kleur (0, 0, 0) alle pixels uitzet? Probeer de volgende code uit op je Flora:

```
void loop() {  
  lightAllOneColour(strip.Color(255, 0, 255));  
  delay(500);  
  lightAllOneColour(strip.Color(0, 0, 0));  
  delay(500);  
  lightAllOneColour(strip.Color(255, 127, 0));  
  delay(500);  
  lightAllOneColour(strip.Color(0, 0, 0));  
  delay(500);  
}
```

Voer nu dezelfde code uit zonder de "uit" kleur:

```
void loop() {  
  lightAllOneColour(strip.Color(255, 0, 255));  
  delay(500);  
  lightAllOneColour(strip.Color(255, 127, 0));  
  delay(500);  
}
```

Zie je het verschil?

6

Probeer je eigen regels te schrijven door de code in de **loop** functie te veranderen! Je kunt zoveel delays en calls aan je **lightAllOneColour** toevoegen als je zelf wilt. Experimenteer met langere en kortere delays en verschillende waarden voor de kleurparameters.

- o onthoud: de hele reeks wordt continu herhaald.

1 Maak de volgende nieuwe functie aan het eind van je schets:

```
void animateOneColour(uint32_t c, uint8_t wait) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
    }
}
```

- o Zie je dat deze functie **twee parameters** heeft? Je gebruikt de tweede pas later.

2 Verwijder (of **comment out!**) de code in de `loop` functie en schrijf een nieuwe aanroep voor je nieuwe functie:

```
void loop() {
    animateOneColour(strip.Color(0, 0, 255), 100);
}
```

Zie je hoe je twee **parameters** binnen de haakjes zet? De tweede wordt nu nog niet gebruikt, maar de code zal niet compileren als je geen waarde voor alle **parameters** invult als je de functie aanroept.

3 Verifieer en upload je code. Wat zie je? Nu hoefde je maar **één regel** te schrijven die `strip.setPixelColor` aanriep en alle pixels aanzette.

4 Binnen je nieuwe functie, zie je dat er nog een paar **acolades** staan met wat code? Dit paar behoort toe aan een **for loop** (= voor lus; maar niet de `loop` functie!), en niet aan een functie. Het ziet er zo uit:

```
for(uint16_t i=0; i<strip.numPixels(); i++) {

}
```

De code hierboven kijkt na hoeveel pixels er in je keten zitten en voert dan de code binnen de accolades dat aantal keer uit. *Dit is er nou zo handig aan: **_de waarde van i begint bij nul en verandert elke keer met één, dus elke keer dat de regel `strip.setPixelColour (i, c);` uitgevoerd wordt, bepaalt het de kleur van de _volgende pixel!***

- 5 Nu gaan we wat met de tweede **parameter** doen! In je nieuwe `animateOneColour` functie voeg je de volgende regel toe na `strip.show();`

```
delay(wait);
```

Zorg ervoor dat de nieuwe regel **vóór** de `}` staat, dus binnen de loop. Je code zou er nu zo uit moeten zien:

```
void animateOneColour(uint32_t c, uint8_t wait) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
    }
}
```

Je gebruikt nu geen getal voor de delay, maar de tweede **parameter** van je functie. Dit betekent dat je verschillende waarden voor de `delay` kunt kiezen als je de functie aanroept.

- 6 Voeg nog een aanroep toe aan je functie binnen `loop`, om de lichtjes uit en aan te zetten:

```
void loop() {
    animateOneColour(strip.Color(0, 0, 255), 100);
    animateOneColour(strip.Color(0, 0, 0), 100);
}
```

- 7 Verifieer je code weer en upload de schets naar je Flora. Nu heb je een gave geanimeerde reeks!

- 8 Je hoeft de pixels natuurlijk niet uit te zetten. Waarom niet een reeks kleuren achter elkaar animeren?

```
void loop() {
    animateOneColour(strip.Color(255, 127, 0), 100);
    animateOneColour(strip.Color(255, 0, 255), 100);
    animateOneColour(strip.Color(0, 255, 255), 100);
}
```

Voeg zoveel kleuren toe als je wilt. Verander ook de waarden van de tweede **parameter** van `100` naar iets anders en zie hoe je animatie sneller of langzamer gaat!

- 1 Voeg de volgende nieuwe functie toe aan het eind van je schets. Geen zorgen, je hoeft het niet meteen te begrijpen! het komt uit de voorbeeldschets die je eerder hebt gebruikt.

```
uint32_t Wheel(byte WheelPos) {  
    WheelPos = 255 - WheelPos;  
    if(WheelPos < 85) {  
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);  
    }  
    if(WheelPos < 170) {  
        WheelPos -= 85;  
        return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);  
    }  
    WheelPos -= 170;  
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);  
}
```

Deze functie laat je elk getal tussen 0 en 255 kiezen en mengt een kleur voor je.

- 2 Voeg nu deze nieuwe functie toe. Zie je de **for loop** daarin?

```
void lightAllRainbow() {  
    for(uint16_t i=0; i<strip.numPixels(); i++) {  
        strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) & 255));  
        strip.show();  
    }  
}
```

Hier zit ook wat wiskunde in! Het kiest een fraaie selectie van kleuren uit de hele regenboog.

- 3 Nu hoef je de functie alleen nog maar **aan te roepen**. Verander de `loop` functie zodat deze coderegel erin zit. Verifieer en upload dan de schets om een prachtige regenboog aan kleuren te zien.

```
void loop() {  
    lightAllRainbow();  
}
```

Je hoeft geen **parameters** in te voeren omdat de nieuwe functie de kleuren voor je uitzoekt!

- 4 En nog een delay toevoegen? Schrijf een nieuwe functie die lijkt op die hierboven maar met een delay toegevoegd in de loop zodat het animeert:

```
void animateRainbow(uint8_t wait) {  
    for(uint16_t i=0; i<strip.numPixels(); i++) {  
        strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) & 255));  
        strip.show();  
        delay(wait);  
    }  
}
```

- 5 Verander de functieaanroep in de `loop` functie en voeg een tweede regel toe om je andere animatiefunctie aan te roepen:

```
void loop() {  
    animateRainbow(100);  
    animateOneColour(strip.Color(0, 0, 0), 100);  
}
```

Probeer het uit op de Flora!

- 6 Combineer verschillende aanroepen in de `animateRainbow` functie en je andere functies. Jouw fantasie is eindeloos! Je kunt heel veel gave dingen doen met de trucjes die je geleerd hebt met kleuren, loops en delays. Als je meer voorbeelden wilt, kijk je in de **strandtest** schets die je gebruikt hebt om de NeoPixels te testen.



- 7 Als je van plan bent om je project te dragen, kun je een batterijhouder gebruiken. Een houder met 3xAA, 3xAAA of een knoopbatterij werkt goed op de Flora. Kijk op dojo.soy/wear2-flora-power voor meer informatie. Gebruik je niet de Flora maar een ander board kijk dan na wat die nodig hebben.