Introductie

- Inleiding
- Bracket Editor
- Atom Editor
- Phoenix Editor
- Stap 0, Starting up
- Stap 1, het stoplicht
- Stap 2, de lampen
- Stap 3, halo-effect
- Stap 4, eerst oranje
- Stap 5, automatisch van oranje naar rood
- Stap 6, oranje flash
- Challenge, voetgangerslicht

Inleiding

Deze 'module' is een voorbeeld van hoe modules er in de praktijk uit zien.

Deze module slaat een paar stappen over en het is dus heel goed mogelijk dat je (nog) niet alles goed begrijpt.

Lees goed en probeer een beetje te begrijpen wat je doet. Volg de stappen nauwkeurig en test elk stapje goed. Ga niet verder als een stap niet goed werkt.

Wat gaan we doen?

We gaan eerst een code-editor installeren. Dit is een speciale editor (een soort MS Word) waarmee je code kan maken.

Dan gaan we wat kleine dingetjes goed zetten op onze computer, we beginnen met een leeg code document en zorgen dat we die met de editor én met de browser kunnen openen.

Dan gaan we echt coderen, we gaan een stoplicht maken. Dat doen we in stapjes waarbij het steeds een beetje lastigere wordt.

Eerst zijn de stapjes heel klein en wordt precies uitgelegd wat je moet doen. Je krijgt code en er wordt precies verteld waar en hoe je die code moet gebruiken.

In de vervolgstapjes wordt de uitleg steeds minder.

Uiteindelijk kom je bij de challenge en daar wordt alleen iets gevraagd en worden er geen stapjes meer beschreven.

Wat gaan we maken?

We gaan een stoplicht maken.

Eerst maken we een eenvoudig stoplicht dat we zelf helemaal kunnen bedienen, maar in stappen 4, 5 en 6 gaan we het stoplicht ook van een soort algoritme voorzien.

Geen ervaring?

Heb je geen ervaring, dan is het al heel knap als je stap 3 af krijgt. Je hebt dan een stoplicht dat je met de hand kan bedienen.

Heb je een beetje ervaring dan kan je misschien stap 4 of 5 ook doen? Hiermee maak je een soort algoritme waarmee het stoplicht werkt.

Wel ervaring?

Het ligt er natuurlijk aan hoe veel ervaring je hebt, maar misschien kan jij de challenge wel uitvoeren?

In de challenge ga je twee stoplichten maken die aan elkaar zijn gekoppeld.

Let wel, begin niet met de challenge, maar voer eerst alle stappen daarvoor uit.

Bracket Editor

Introductie

Om te kunnen coderen hebben we een editor nodig. Met de editor kunnen we de code intypen en aanpassen.

Voor deze introductie maken we gebruik van de light *weigth code* editor Brackets.

Download

Download Brackets en installeer de editor.

Kies tijdens het installeren kies je voor de standaard ("*typical*") installatie.

image-1714079311423.png image-1714079373886.png

Installatie klaar? Start dan Brackets op!

Atom Editor

Introductie

Om te kunnen coderen hebben we een editor nodig. Met de editor kunne we de code intypen en aanpassen.

Voor deze introductie maken we gebruik van de light weigth code editor *Atom*.

Download

Download Atom en installeer de editor (alternatieve downloads: hier en hier)

Configuratie

Start Atom op en ga naar **menu file** dan **settings** en kies dan in de linker kolom **Keybindings**.

Klik op your keymap file (bovenaan).

image-1713608185734.png

Er wordt een bestand geopend. Kopieer de volgende tekst daar in:

```
'atom-text-editor':
  'ctrl-alt-f': 'editor:auto-indent'
```

De tekst kan gewoon onderaan bestand worden toegevoegd.

Je kan het bestand en de settings tabjes nu sluiten. Je bent klaar voor het echte werk!

Uitleg

Code is beter leesbaar als we de code op een bepaalde manier 'uitlijnen'. Uitlijnen is het netjes vormgeven van code.

Met de configuratie optie die we net hebben ingesteld kunnen we de uitlijning automatisch door de code-editor laten uitvoeren.

Later zullen we zien hoe dit werkt!

--

Phoenix Editor

Wat is een code editor?

Stel je voor: je wilt een verhaal schrijven. Dan gebruik je **Word** of **Google Docs** om netjes tekst te typen.

Een **code editor** is eigenlijk hetzelfde, maar dan speciaal gemaakt om **computerprogramma's te schrijven**.

Dus kort gezegd: een code editor is het "*Word voor programmeurs*", gemaakt om programma's en websites te bouwen.

Phoenix

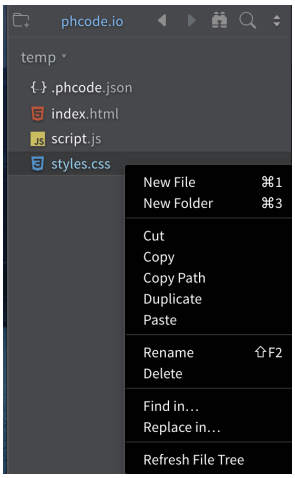
Je kunt de Phoenix editor downloaden of online gebruiken.

Wij gebruiken de online editor.

- Ga naar https://phoenixcode.dev
- Klik op**File - StartProject** en kies **HTML5 project**.
- Kies **Select Folder** (dit is de plaats waar jouw code komt te staan)
- Geef de browser toestemming om bestanden te kunnen lezen (**view files**).
- Druk op **Create Project**

Verwijder de standaard bestanden

Je aan de linker kant van het scherm een overizhct van (4) bestanden. Dit is een soort voorbeeld project. Je kan de bestanden weggooien. **Rechter-muis-delete**



Maak nieuw bestand

Als alle bestanden weg zijn dan maak je met **rechter-muis-new fil**e Noem het bestand *stoplicht.html.*

Ga snel verder met stap 0.

--

Stap 0, Starting up

In de code-editor (Atom, Brackets of VCS) maak je een nieuw bestand aan (file, new *of* new file).

In Phoenix heb je al een nieu leeg bestand gemaakt.

Je ziet een lege pagina.

Dit is de code waar we mee gaan starten.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Simple Traffic Light Simulator</title>

<style>
  <!-- hier wordt de vormgeving met CSS gecodeerd -->
</style>

<script>
   <!-- hier wordt het algoritme, de code van de pagaina gecodeerd -->
</script>

</head>

<!-- de rest van de pagina is HTML, hier begint de body -->
<body class='main'>
  lege pagina
</body>
</html>
```

1. Kopieer de bovenstaande code in de lege pagina in de editor.
2. Bewaar (**file**, **save**) de pagina op bij voorkeur je *Desktop* (*Bureaublad*) en geef het de naam ***stoplicht.htm***
   Nadat je het bestand op de desktop hebt bewaard, zie je ook kleurtjes in de editor.

   Gebuik je **Phoenix** dan heb je al een bestand gemaakt, je moet de nieuwe inhoud nog wel **saven**.

3. Ga naar je *Desktop* en dubbel klik op het nieuwe bestand.

   Gebruik je **Phoenix** dan zie gelijk het resultaat aan de rechterkant.

Als het goed is zie je de lege pagina in je browser.

Drie gedeeltes

Kijk goed naar de code, je hebt drie gedeeltes:

1. Het **CSS gedeelte** voor de vormgeving.
2. Het **script gedeelte** voor de 'echte' code.
3. De **body** waar, het stoplicht wordt geplaatst.

Zie je deze gedeeltes die nu nog bijna helemaal leeg zijn?

?Tip

In windows op je desktop zie je nadat je het bestand hebt opgeslagen een bestand met de naam stoplicht.

Zie je alleen *stoplicht* en niet *stoplicht.htm*?

Als dat zo is, dan kan je dat verhelpen (in Windows 11) door in de file explorer de optie file name extensions aan te klikken:

image-1713688844137.png

Testen

Probeer de tekst '*lege pagina*' te veranderen in een eigen tekst en controleer of je het resultaat in de browser ziet!

Als dit werkt dan kun je verder met de volgende stap.

--

Stap 1, het stoplicht

Aan het einde van deze stap heb je een leeg stoplicht, zeg maar een stoplicht zonder lampen erin.

We beginnen met de algemene set-up.

In de `<body>` `</body` >heb je de tekst (in de vorige stap) aangepast.

Vervang de (aangepaste) tekst in de `body` door deze code.

```
<div class="traffic-light">
    <div id='red' class="lamp"></div>
    <div id='orange' class="lamp"></div>
    <div id='green' class="lamp"></div>
    <div id='text' class="text">-</div>
</div>
```

We hebben nu drie lampen gemaakt en daaronder plaatsen we een tekst.

Als je dit hebt uitgevoerd en je test dit, dan zie je nog **niets** dat komt, omdat we de vormgeving nog moeten maken.

Tussen de `<style>` en `</style>` tags, plaatsen we de volgende CSS-code voor de vormgeving.

```
.main {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 800px;
    background-color:DarkGray;
}
.traffic-light {
    display: flex;
    flex-direction: column;
    width: 100px;
    padding: 20px;
    background-color: black;
    border: 2px solid #666;
    border-radius: 10px;
```

```
    }
    .lamp {
        width: 50px;
        height: 50px;
        margin: 10px auto;
        background-color:darkgray ;
        border-radius: 50%;
        transition: background-color 0.8s;
    }
    .text {
      color:white;
      margin: 10px auto;
      font-family: fantasy;
    }
```

Uitleg

Je ziet telkens een woord met een punt ervoor. Dit zijn class definities.

Als je in de code kijkt dan zie je ergens *class='lamp'* staan. De CSS-code die we net hebben geplaatst en die .lamp heet heeft betrekking op de HTML-code met *class='lamp'*. Dus *class='lamp'* is gekoppeld met de .lamp die in de `style` staat.

Testen

Als het goed is zie je nu een soort stoplicht zonder lampen op je pagina.

image-1713610847009.png

Ga alleen door als je dit verkeerslicht ziet.

--

Stap 2, de lampen

Aan het eind van deze stap heb je een stoplicht met de drie lampen, rood, oranje, en groen.

Rode licht

Om de lamp te laten branden, plaatsen we code in het `<script>` `</script>` gedeelte.

```
function switchLight(color) {
    if (color === 'red') {
      document.getElementById("red").style.backgroundColor="red";
      document.getElementById("orange").style.backgroundColor="darkgray ";
      document.getElementById("green").style.backgroundColor="darkgray ";
      document.getElementById("text").innerHTML="STOP";
    }
}
```

Er gebeurt nu nog niets, maar we hebben nu een functie *switchLight* gemaakt.

Deze functie zet het rode licht aan. Nu moeten we de functie koppelen met het licht zodat als we op het licht klikken deze functie wordt aangeroepen.

We veranderen nu de regel waar het rode licht staat zodat als we op het rode licht klikken de functie wordt aangeroepen.

De HTML-code voor het rode licht wordt nu (het *onclick*-gedeelte komt er dus bij).

```
<div id='red' class="lamp" onclick="switchLight('red')"></div>
```

Je past de `div` van de rode lamp dus aan.

We het licht nu veranderd zodat wanneer je op het licht klikt (onclick), de nieuwe functie die we hebben gemaakt wordt aangeroepen.

image-1713612642744.png

Testen

Controleer of als je o het rode licht klikt, het rode licht inderdaad 'aan' gaat.

Ga verder als het werkt.

Oranje licht.

Voor het oranje licht breiden we de code in het script gedeelte uit met:

```
if (color === 'orange') {
      document.getElementById("red").style.backgroundColor="darkgray ";
      document.getElementById("orange").style.backgroundColor="orange";
      document.getElementById("green").style.backgroundColor="darkgray ";
      document.getElementById("text").innerHTML="-";
}
```

Let op dat deze if **onder** de bestaande if en **in** de functie moet staan.

Let hierbij op de { en }.

Nu kunnen we in de editor op CTRL-ALT=F(alleen in Atom) drukken. De Atom editor zorgt voor de juiste *identation* (vormgeving van de code) zodat je makkelijker kan controleren of je alle { en } goed hebt staan.

Plaats in de HTML-code bij het oranje licht ook een verwijzing naar de functie, net als bij het rode licht.

```
<div id='orange' class="lamp" onclick="switchLight('orange')"></div>
```

image-1713612618172.png

Testen

Test of het oranje en rode licht nu werken. Ga naar je browser en klik op het rode en oranje licht. Lichten beide lichten op? Top, dan kan je verder naar de laatste stap, het groen licht.

Groen licht

We hebben het rode- en groene licht gemaakt. Deze werken.

Probeer nu zelf het groene licht werkend te krijgen!

De tekst die onder het groene licht moet verschijnen is 'GO'.

image-1713612670765.png

Testen

Test of al je lichten werken en ga verder als dat zo is.

--

Stap 3, halo-effect

Aan het eind van deze stap hebben alle drie de lampen een halo-effect.

Halo effect

Om meer een gevoel te krijgen dat er echt een lamp brandt, voegen we bij de lamp die aan is een zogenaamd halo-effect toe.

Dat ziet er als volgt uit:

image-1713615183169.png	image-1713615142704.png
Zonder halo effect	**Met halo effect**

Om dit te maken voegen we eerst een class toe aan de styles (tussen `<style>` en `</style>`):

```
.glow {
	box-shadow: 0 0 20px 8px rgba(165,165,165,0.8);
}
```

Via de code in `<script>` `</script>` kunnen we de style toepassen.

Halo-effect aan

Om het halo-effect aan te zetten voegen we de class toe aan het licht met de code:

```
document.getElementById("red").classList.add('glow');
```

Halo-effect uit

En om het halo-effect uit te zetten halen we de class weer weg met de code:

```
document.getElementById("red").classList.remove('glow');
```

Probeer nu zelf te bepalen waar in de code je het halo-effect aan en uit moet zetten. Doe dat eerst alleen voor het rode licht.

Testen

Als het rode licht aan is, dan zie je het halo-effect en als het rode licht uit is dan is het halo-effect weg.

Je kunt op twee manieren het rode licht uit zetten: door op het oranje- of groene licht te drukken. Test of in beide gevallen het halo-effect uit gaat.

Als dat werkt dan kun je door naar de volgende stap.

Halo effect op oranje en groene licht

Breidt de code nu uit en maak een halo-effect op het groen licht en daarna op het oranje licht.

Testen

Test goed of alles werkt, klik alle lampen aan en uit en controleer of in alle gevallen het halo-effect aan en uitschakelt.

Werkt alles?

Gefeliciteerd, je hebt deel 1 van de opdracht volbracht!

--

Stap 4, eerst oranje

Als je deze stap hebt gedaan dan kun je niet in één keer van groen naar rood. Staat het licht op groen en je drukt op rood dan zal het oranje licht aan gaan.

Als we het licht op groen hebben staan en we drukken op rood dan gaat het licht op rood.

Maar eigenlijk mag dit niet. Als je van groen naar rood gaat dan moet het licht eerst naar oranje!

De opdracht is nu om als we op rood drukken een controle te maken of welk licht aan is. Is het groen licht aan dan gebeurt er niets.

Onthoud de kleur

Om dit te kunnen doen, moeten we eerst in de code 'onthouden' welke lamp aan is.

Dit gaan we onthouden in een variabele.

Helemaal bovenaan in de code tussen (<script> en </script>) plaatsen we de volgende code.

```
var colorOn;
```

Hiermee maken we een variabele waarin we 'onthouden' welke lamp aan is.

Nu moeten we zorgen dat de juiste waarde in variabele wordt gezet.

In het if-blokje van green, zetten we

```
colorOn = 'green';
```

In het rode blokje zetten we natuurlijk

```
 colorOn = 'red'
```

En in het orange blokje zetten we

```
colorOn='orange'
```

Nu wordt telkens als er een lamp wordt aangezet de kleur van de lamp onthouden in de variabele colorOn.

Controle bij rood

In het de code vlak onder de regel:

```
if (color === 'red') {
```

gaan we nu controleren of het groene licht aan is.

Dat doen we met de volgende code:

```
if ( colorOn == 'green' ){
        return;
}
```

Wat hier staat is dat als het groene licht aan is, we de functie verlaten en verder niets meer doen. Het rode licht gaat dan dus niet meer aan.

Testen

Dus we kunnen op een lamp drukken en dan gaat de lamp aan, behalve als de groen lamp aan is en we drukken op de rode lamp dan gebeurt er niets. Dit mag immers niet omdat we eerst op oranje moeten drukken.

Rood via oranje

Als het stoplicht op groen staat en we drukken op rood dan gebeurt er nu niets. Maar we zouden in dit geval ook het licht op oranje kunnen zetten.

Dus als het licht op groen staat en je drukt op rood dan gaat het oranje licht aan.

Wat gebeurt er nu als we van groen naar rood willen gaan? Dan wordt er een *return* uitgevoerd en dat betekent dat er niets gebeurt. In plaats van de *return* zouden we ook het licht op oranje kunnen zetten en daarna de functie stoppen via een *return*.

Kan jij bedenken hoe je dit zou moeten doen?

> Kijk eens wat er gebeurt als je op het licht klikt? Dan wordt de functie aangeroepen. Vlak voor s de *return* zou je dus ook eerst het licht op oranje kunnen zetten door op de juiste manier de functie kunnen aanroepen.

Testen

Alle lichten gaan aan als je op het licht drukt, behalve als je op het rode licht drukt, terwijl het groene licht aan staat dan zal het oranje licht gaan branden.

--

Stap 5, automatisch van oranje naar rood

Deze stap is een uitbreiding op de volgende stap. Druk je vanuit groen op rood dan zal het oranje licht aan gaan. Dit hadden we al. Maar nu zal na 1 seconde het oranje licht uit gaan en vanzelf het rode licht aan gaan.

Als we het licht op groen staat en we drukken op rood dan gaat het oranje licht aan. Daarna kunnen we weer op rood drukken en dan gaat het rode licht aan.

We kunnen dit ook in één stap uitvoeren. Dus het licht staat op groen, we drukken op rood. Het licht gaat dan op oranje en na één seconde gaat het licht daarna vanzelf op rood.

Wat we moeten doen is de code veranderen die nu wordt uitgevoerd als we rood drukken terwijl het licht op groen staat.

Als het goed is, wordt daar nu het licht op op oranje gezet. Dat blijft zo.

Maar vlak daarna zal het licht op rood moeten worden gezet. Dat moet na 1 seconde. We kunnen daarvoor het volgende commando gebruiken:

```
setTimeout(function() { switchLight('red'); }, 1000);
```

Wat hier staat is dat de functie switchLight('red') na 1000 milliseconden moet worden uitgevoerd. 1000 milliseconden is 1 seconde, dus na 1 seconde wordt het licht op rood gezet.

Testen

Als je op een lamp drukt dan gaat deze lamp aan. Behalve als het licht op groen staat en je druk op rood. Dan zal eerst voor 1 seconde het oranje licht aan gaan en daarna zal het rode licht aan gaan.

Als dit werk dan kan je door gaan.

--

Stap 6, oranje flash

Het oranje licht gaat nu automatisch aan en uit. De functie van kunnen klikken op oranje is nu overbodig. Deze functie gaan we nu wijzigen. Aan het eind van deze stap gaat het oranje licht knipperen en gaan de andere lichten uit.

Als we op het oranje licht klikken dan zal die aan gaan, maar we hebben het algoritme al zo verandert dat het oranje licht vanzelf aan gaat als we het licht op rood willen zetten.

Wat we gaan doen is dat we het stoplicht 'uit' gaan zetten als we op het oranje licht klikken.

Welke stappen gaan we doen?

1. We gaan een style maken die het oranje licht laat knipperen.
2. We maken een nieuwe functie en die noemen we switchOff(). Deze functie voegt de nieuwe stijl toe aan het oranje licht en zet het groene en rode licht uit.

Flash stijl

In de style sectie plaatsen we twee nieuwe stijlen:

```
.lamp-flash {
  animation: flash 1s infinite;
}

@keyframes flash {
    0%, 100% { background-color: darkgray; } /* Light off (dark gray color) */
    50% { background-color: orange; } /* Light on (orange color) */
}
```

Als we nu de stijl lamp-flash toevoegen aan het oranje licht dan zal hij gaan knipperen.

Functie

In de nieuwe functie switchOff() zetten we het oranje licht op knipperen.

```
function switchOff() {
  document.getElementById("orange").classList.add('lamp-flash');
```

```
}
```

Functie aanroepen

Als je op het oranje licht klikt dan moet je in plaats van `switchLight('red')` nu `switchOff()` aanroepen.

Pas dit aan.

Testen

Als je nu op het oranje licht drukt dan gaat het oranje licht knipperen.

Nice, maar er zijn nog wel een paar stappen die niet goed werken.

Dat gaan we nu oplossen.

Oranje dan andere lichten uit.

Pas de *switchOff*() aan en zet de andere lichten uit. Kijk in de bestaande code hoe je de lichten uitschakelt en plaats deze code in de *switchOff* functie. Vergeet niet de *glow* ook uit te zetten!

En dan moet je de flash ook nog weer uit zetten.

Dat doe je met deze code:

```
document.getElementById("orange").classList.remove('lamp-flash');
```

Op het moment dat het stop licht uit staat (dus oranje knippert) dan kan je deze weer aan zetten door op het rode of groene licht te klikken.

Bedenk nu zelf waar je de code moet zetten om het stoplichte weer aan te zetten.

Tekst aanpassen

Als het oranje licht knippert moet de tekst onderaan het stoplicht veranderen naar '-'

Testen

Test of alles werkt aan de hand van volgende tabel.

Status	Klikt op	Resultaat	Tekst	Werkt?

Alles uit (begin)	rood	rood aan, rest uit.	-	
rood aan	groen	groen aan, rest uit	STOP	
groen aan	rood	oranje aan 1 sec, dan rood aan, rest uit.	GO	
rood of groen aan	oranje	alles uit en oranje gaat knipperen	-	
oranje knippert	rood	rood aan, rest uit	STOP	
oranje knippert	groen	groen aan, rest uit	GO	

-

Challenge, voetgangerslicht

Challenge

Een challenge is een uitdaging waarin je met hetgeen wat je hebt geleerd wordt gevraagd om wat te maken.

Je krijgt geen code en de stapjes worden niet allemaal beschreven. Er wordt gevraagd om wat te maken en dan mag je verder zelf bedenken hoe je dat aanpakt.

Voetgangerslicht

Stoplichten staan nooit alleen.

Maak een tweede licht voor voetgangers.

Koppel de twee verkeerslichten aan elkaar zodat als het voetgangerslicht op rood staat het verkeerslicht op groen staat en andersom. Beide lichten staan gelijktijdig op oranje.

Dus de lichten werken als volgt samen:

Bestaande verkeerslicht	Nieuwe voetgangerslicht
Rood	Groen
Groen	Rood
Oranje	Oranje

--