CSS challenge

FDND 24-25

# #5 POSITION

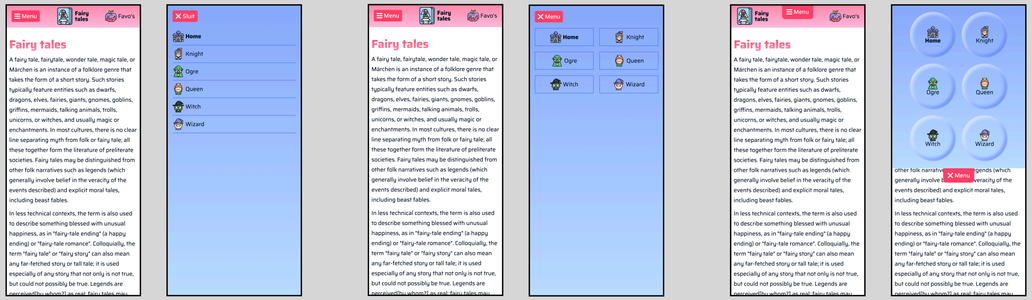
## oefening 2
## hamburger menu

**JS 3-stap**

# Responsive burgers

Veel websites hebben (zeker op small screen) een hamburger menu. Als het scherm groter wordt, wordt dan vaak (een deel van) het menu direct getoond.

Je kunt oefenen met 3 varianten:




**2-buttons**
uitleg in 10 stappen

Voorbeeld
codepen.io/shooft/live/dyxvZzK




**fixed-button**
uitleg in 11 stappen

Voorbeeld
codepen.io/shooft/live/eYqvyLP




**moving-button**
uitleg in 11 stappen

Voorbeeld
codepen.io/shooft/live/vYoxpwK

Voor de **blauwe** en **rode** piste

**JS 3-stap - Oefening 2**

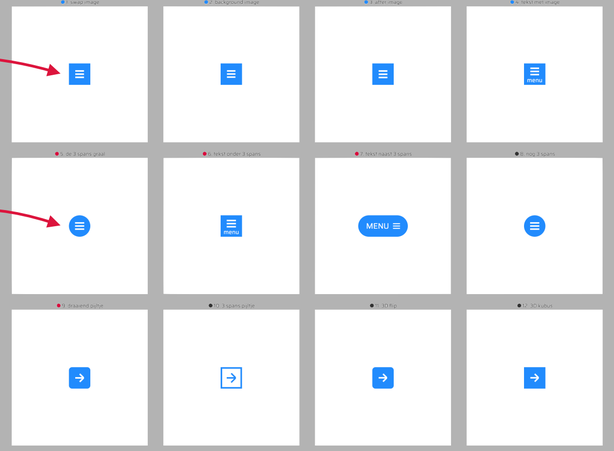
# Hamburger buttons

Hamburger menu's zijn niet ideaal. Menu-items in een hamburger menu worden bijv. minder vaak bezocht dan menu-items die direct op het scherm staan (uit zicht maakt onbemind). Maar à la - de site die je na maakt kan een hamburger menu bevatten.

**De hamburger button**

Er bestaan vele hamburger button varianten. Op codepen.io/shooft/pen/JjQLVeB vind je er twaalf.

**Variant 1 en 5** komen het vaakst voor.




populair

Nb. Buttons met **het woord menu** in de button werken het best.

Nb. Gebruikers komen niet voor coole buttons. Ze komen voor content. **Besteed** dan ook niet te veel **tijd** aan coole buttons, maar wel **aan** de **content**.

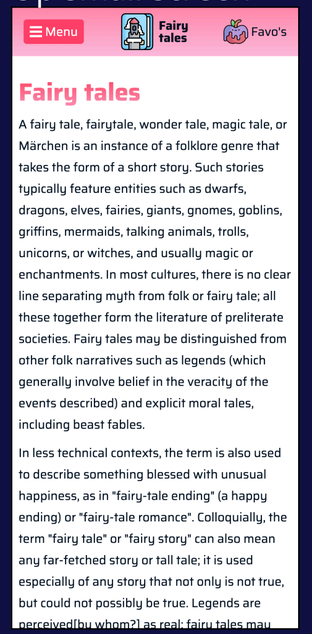
**JS 3-stap - Oefening 2**

# Responsive 2-buttons burger

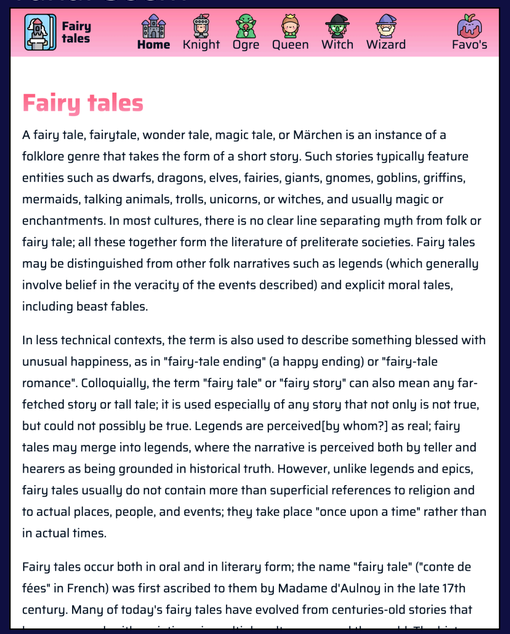
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

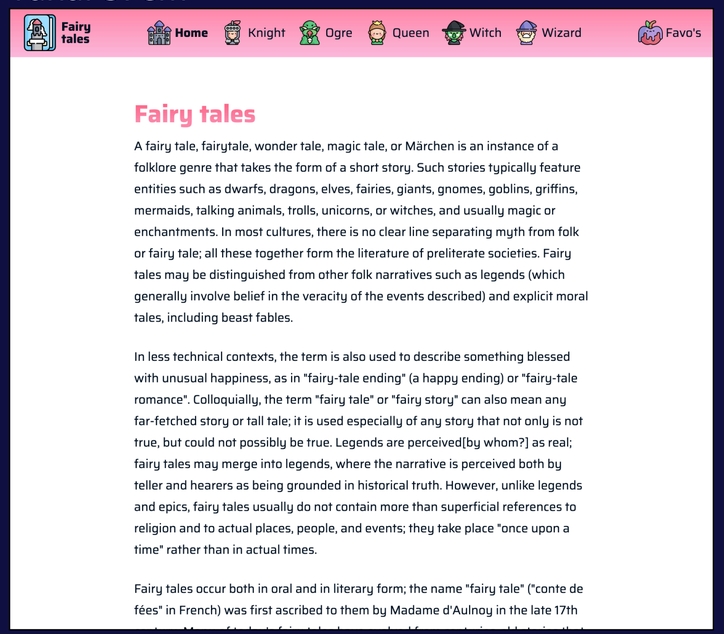
Op small screen



Vanaf 38em




Vanaf 57em



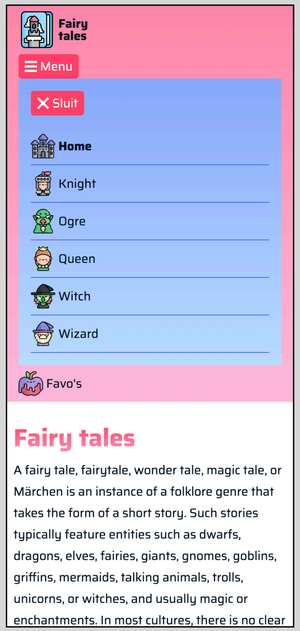

Voor de **blauwe**
en **rode** piste

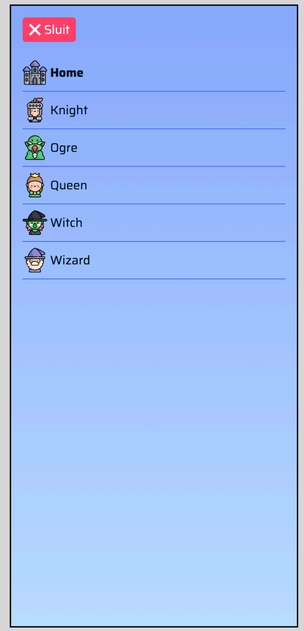
**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK




De basisvormgeving krijg je cadeau.

**nav**

1. Positioneer de nav fixed.
2. En dek het hele scherm met de nav af (hint: left, right, top, bottom).

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

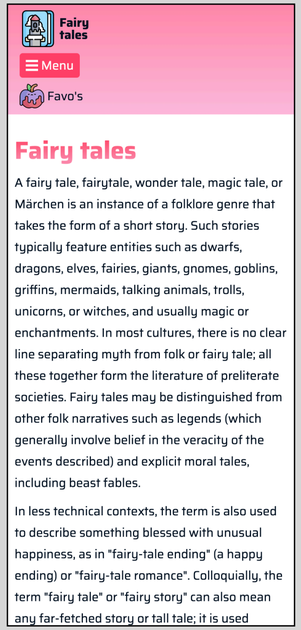
De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

**nav**

1. Schuif de nav uit beeld met
   translate: xxx% yyy%;

Nb. De nav moet zijn eigen breedte (100%) naar links bewegen.

Nb. Naar links is negatief.



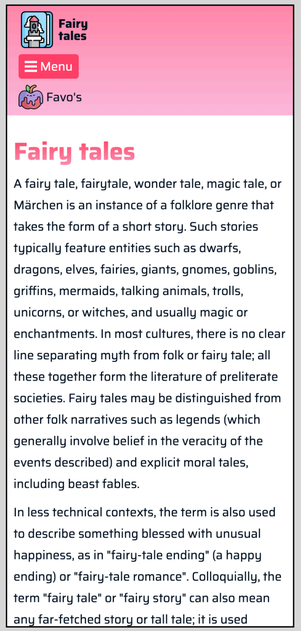

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

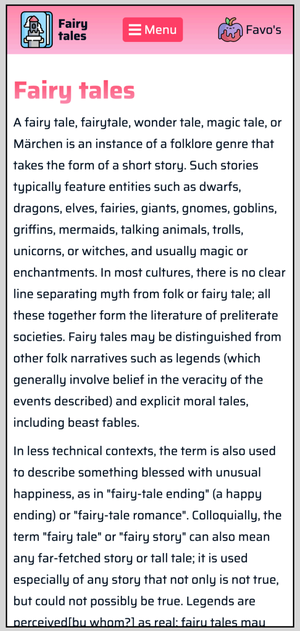
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK



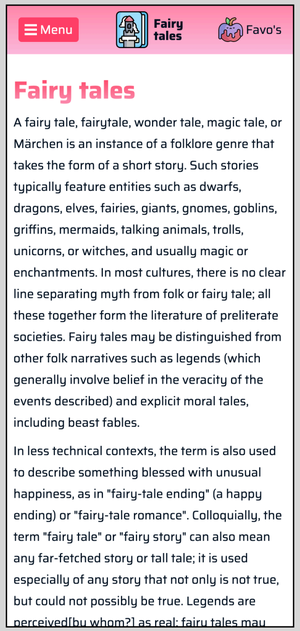

**header**

1. Flex.
2. Extra ruimte netjes verdelen.
3. Verticaal in het midden.



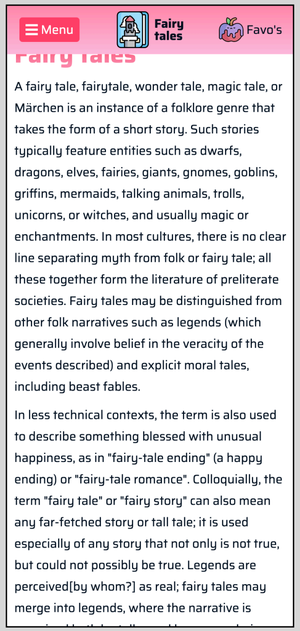

**header > button**

3. Als eerste in de rij (tip: order).




**header**

4. Bovenaan laten plakken.



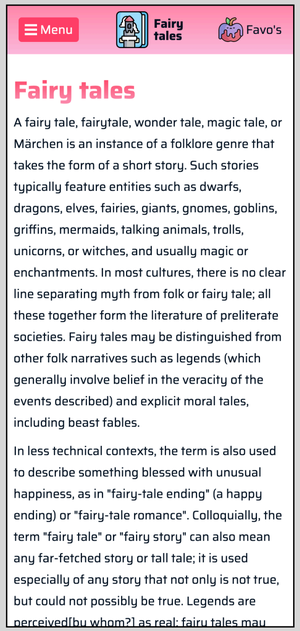

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

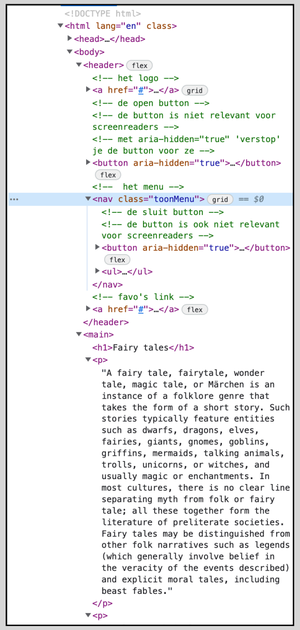
De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK




**JS 3-stap 1/2**

1. Zoek de “menu” button (querySelector) en sla die op in een variabele.
2. Laat de button naar clicks luisteren (addEventListener) en roep de functie “openMenu” aan als er geklikt wordt.
3. In de functie “toonMenu”:
   - Zoek de nav op en sla die op in variabele.
   - Voeg de class “toonMenu” toe aan de nav (classList & add).

Nb. Op het scherm zie je niets veranderen. In de inspector kun je wel controleren of de class wordt toegevoegd.




Verder met de CSS

**nav.toonMenu**

1. Maak de verschuiving uit stap 2 weer ongedaan met translate: xxx% yyy%; (het menu staat dan weer in beeld).

**nav**

2. Voeg een transitie van 0.3s toe aan de nav om die soepel heen en weer te schuiven.

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

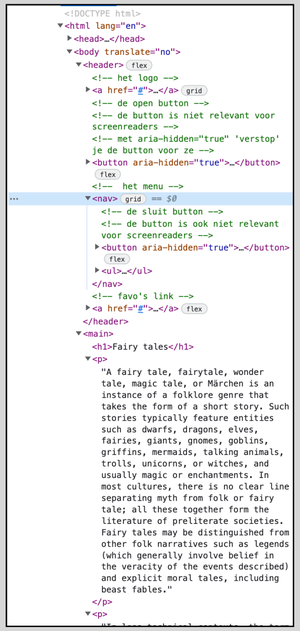
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

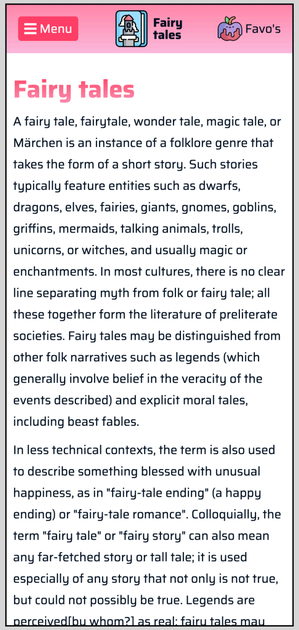
**JS 3-stap 2/2**

1. Zoek de “sluit” button (querySelector) en sla die op in een variabele.
2. Laat de button naar clicks luisteren (addEventListener) en roep de functie “sluitMenu” aan als er geklikt wordt.
3. In de functie “sluitMenu”:
   - Zoek de nav op en sla die op in variabele.
   - Verwijder de class “toonMenu” van de nav (classList & remove).

Bonus: Verwijder de class “toonMenu” ook als je op de Escape toets drukt.




De CSS van nav.toonMenu is niet meer van toepassing. De nav schuift daardoor weer uit beeld.




Voor de **blauwe** en **rode** piste

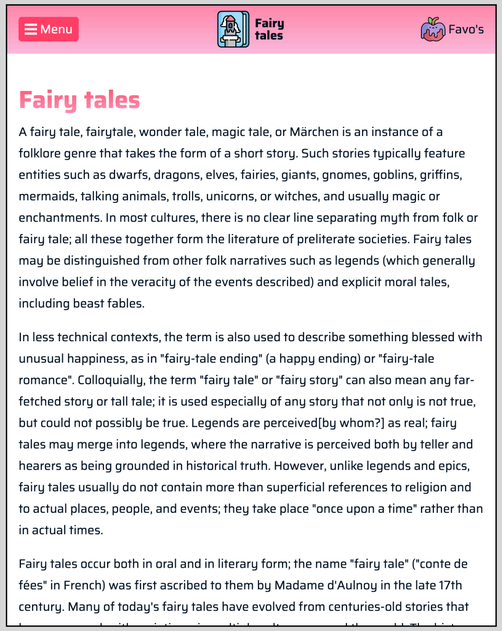
# vanaf hier
# responsiveness

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

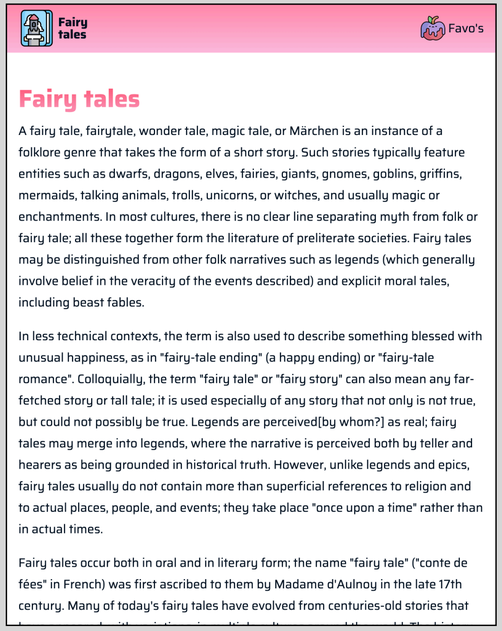



Nu verder met de responsiveness. Het menu past vanaf 38em in beeld. De buttons zijn dan niet meer nodig. Die als eerste verstoppen.

In de 38em media query:

**header > button**
1. Verstoppen.

**nav button**
2. Verstoppen.



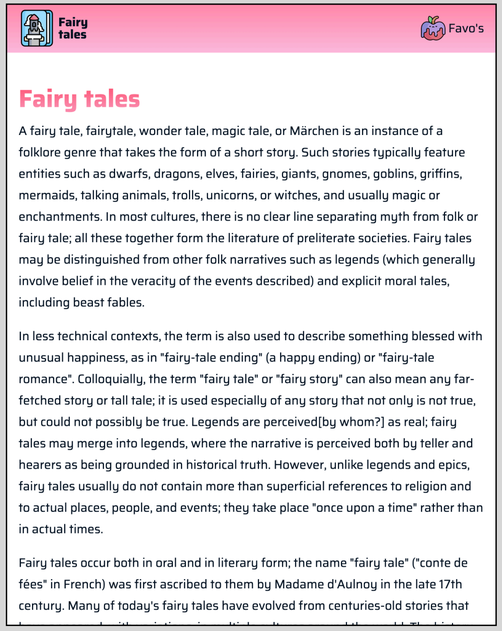

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

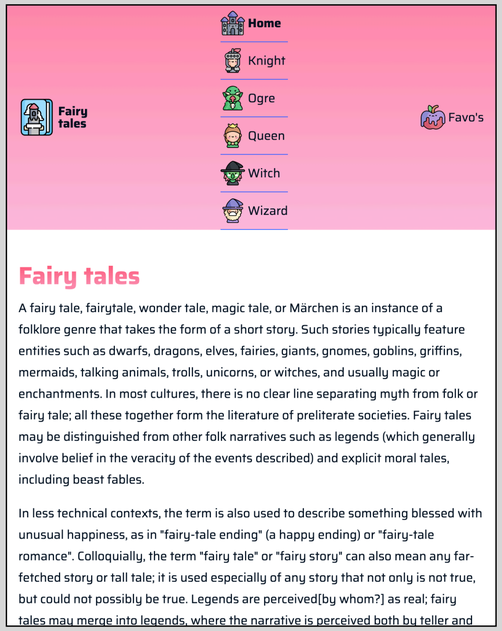



De nav weer op zijn ‘normale’ plek zetten. Daarvoor moet je in de media query een aantal dingen ‘ongedaan’ maken.

**nav**

1. Position (tip: static)
2. Translate (tip: unset)
3. Padding
4. Background

Nb. Het klopt dat het er niet uit ziet. Dat komt zo.



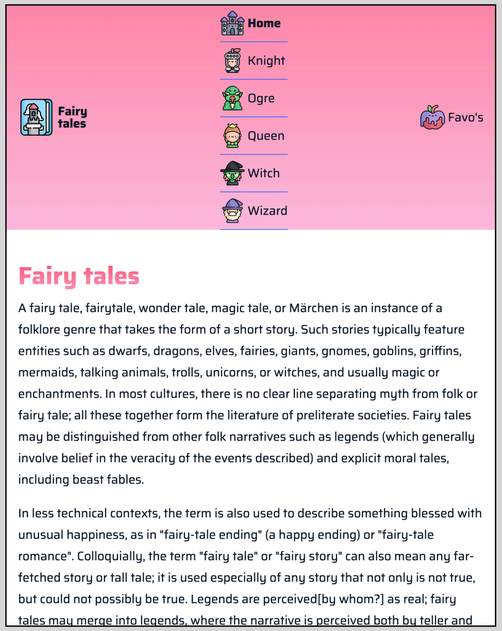

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

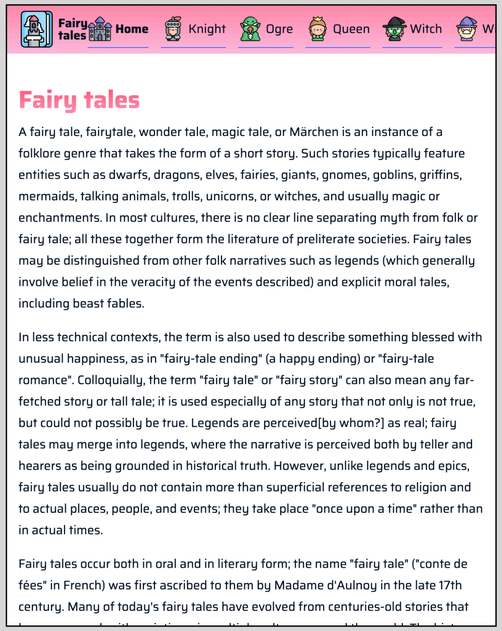
De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK




**ul**

1. Zet de li’s (in de media query) naast elkaar met 1em ruimte ertussen (tip: flex-en maar).

Nb. Het klopt dat het er nog niet uit ziet.
Dat komt in de volgende stap.



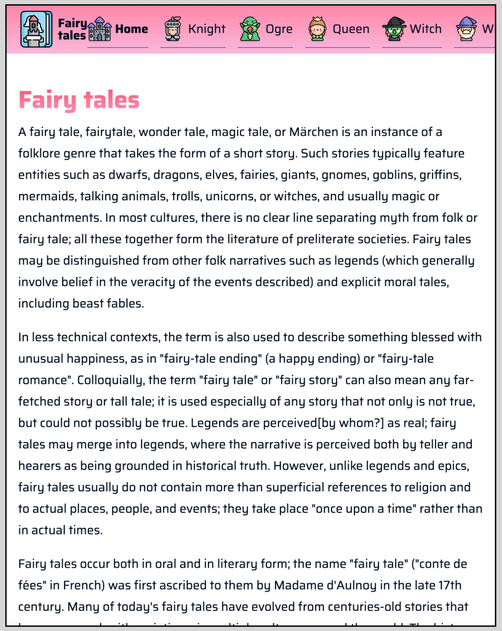

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK

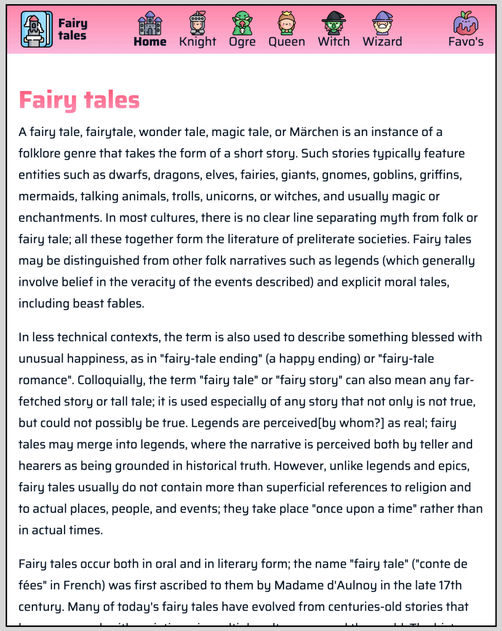



**nav a**

1. Zet de kinderen van de a’s in de nav boven elkaar (tip: flex-en plus flex-direction).
2. Verwijder de border.

**header > a:last-of-type**

3. Zet de kinderen van de favo’s a ook boven elkaar.



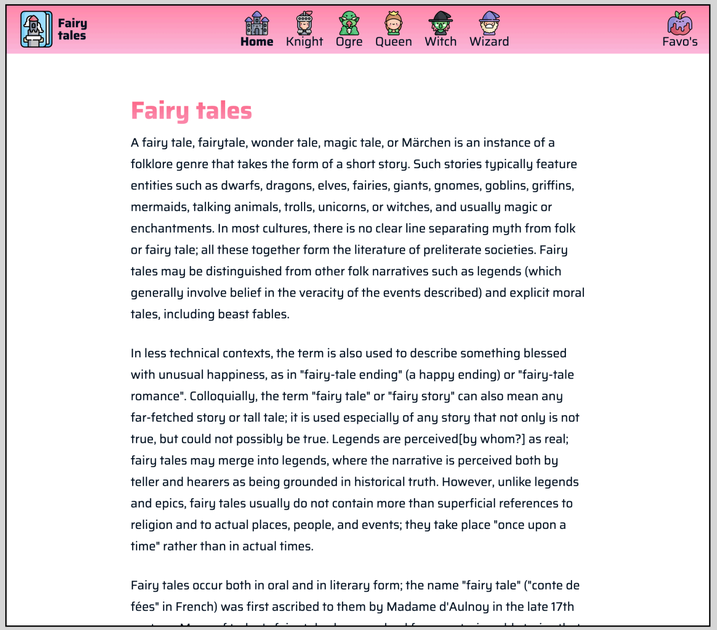

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive 2-buttons burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerVXx
Uitwerking:
codepen.io/shooft/pen/dyxvZzK



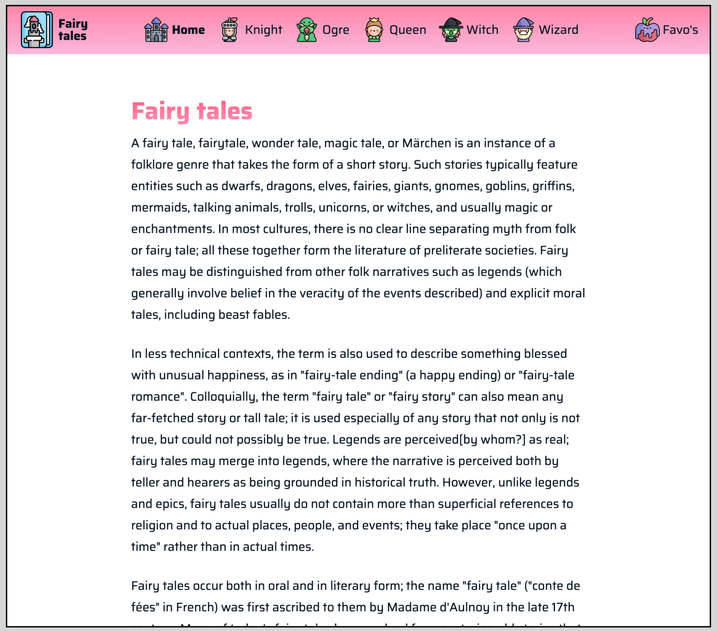

Als het scherm nog groter wordt passen de plaatjes en teksten weer naast elkaar.

In de 57em media query:

**nav a, header > a:last-of-type**

1. Zet de kinderen van de a’s weer naast elkaar (tip: flex-direction).

Et voilà!




Voor de **blauwe**
en **rode** piste

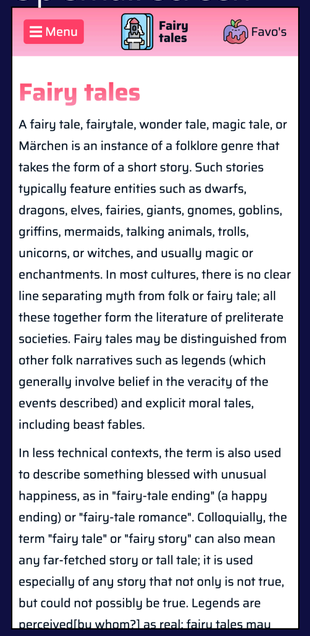
**JS 3-stap - Oefening 2**

# Responsive fixed-button burger

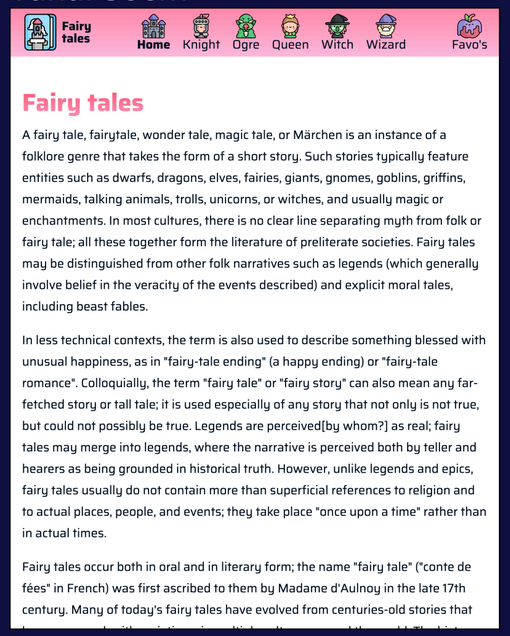
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

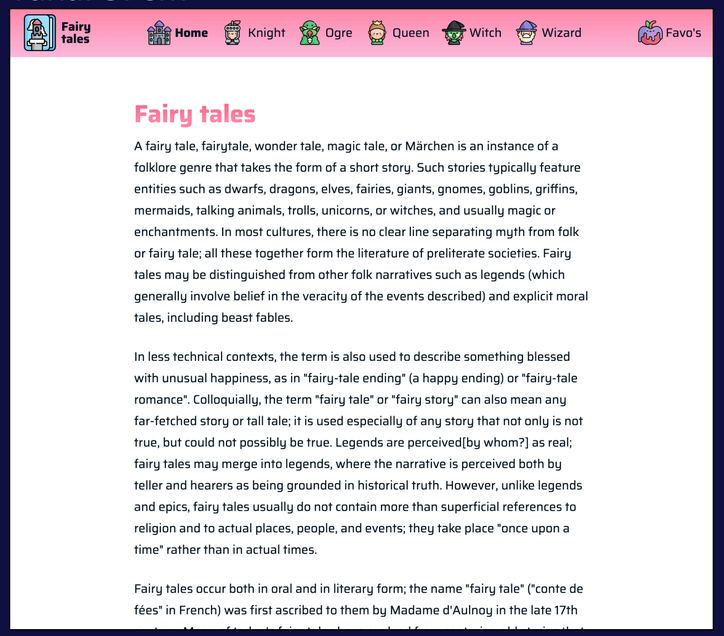
Op small screen



Vanaf 38em




Vanaf 57em



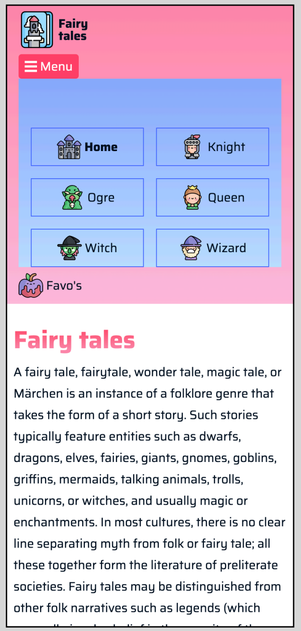

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP




De basisvormgeving krijg je cadeau.

**ul**

1. Positioneer de ul fixed.
2. En dek het hele scherm met de ul af (hint: left, right, top, bottom).

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

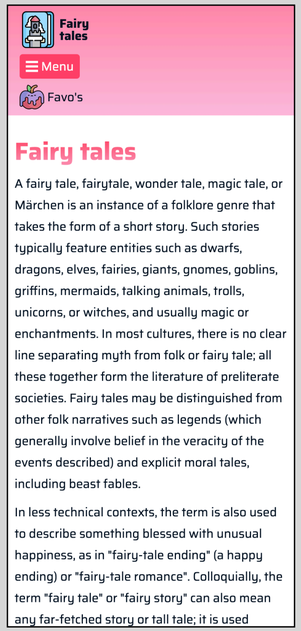
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

**nav**

1. Schuif de ul uit beeld met
   translate: xxx% yyy%;

Nb. De ul moet zijn eigen breedte (100%) naar rechts bewegen.



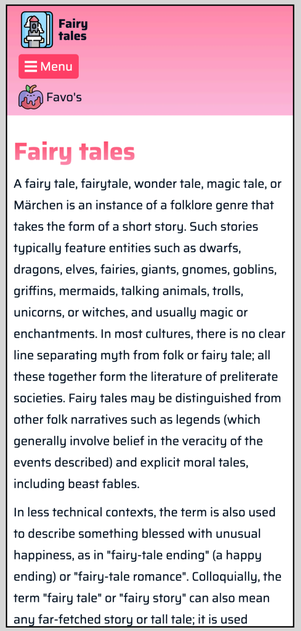

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

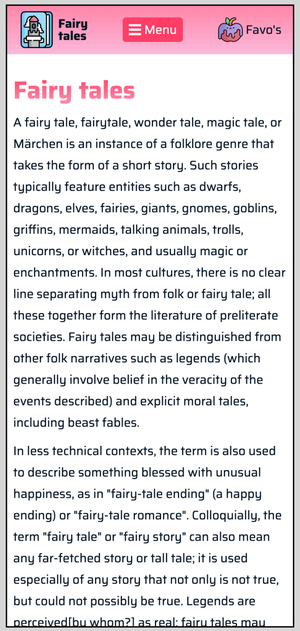
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP



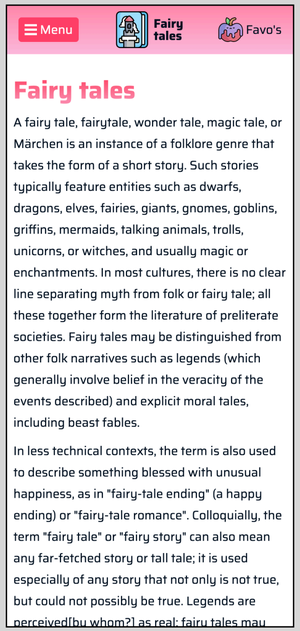

**header**

1. Flex.
2. Extra ruimte netjes verdelen.
3. Verticaal in het midden.



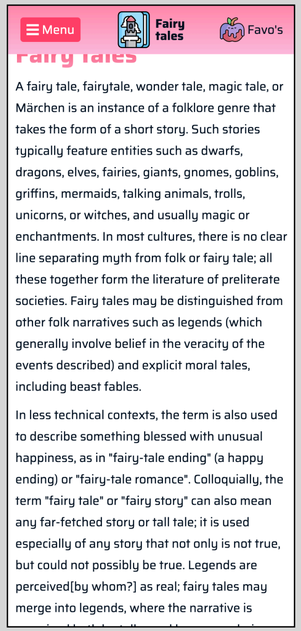

**nav**

4. Als eerste in de rij (tip: order).




**header**

5. Bovenaan laten plakken.



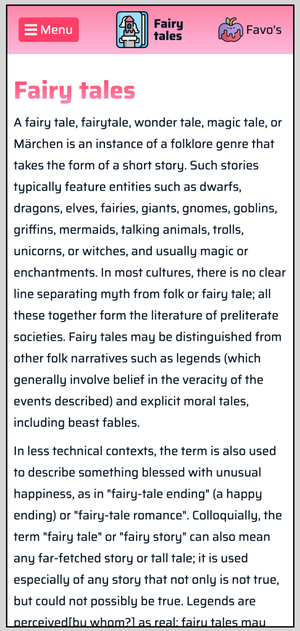

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

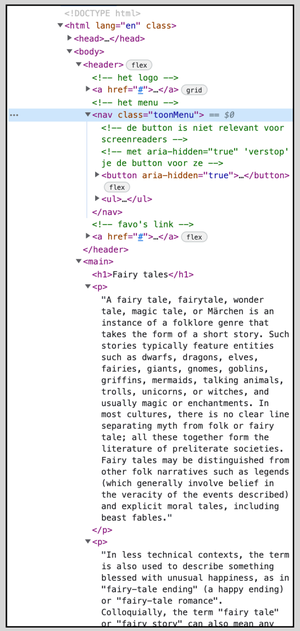



## JS 3-stap 1/2

1. Zoek de “menu” button (querySelector) en sla die op in een variabele.
2. Laat de button naar clicks luisteren (addEventListener) en roep de functie “openMenu” aan als er geklikt wordt.
3. In de functie “toonMenu”:
   - Zoek de nav op en sla die op in variabele.
   - Voeg de class “toonMenu” toe aan de nav (classList & toggle).

Nb. Op het scherm zie je niets veranderen. In de inspector kun je wel controleren of de class wordt toegevoegd.

Bonus: Verwijder de class “toonMenu” ook als je op de Escape toets drukt.




Verder met de CSS

**nav.toonMenu ul**

1. Maak de verschuiving uit stap 2 weer ongedaan met translate: xxx% yyy%; (het menu staat dan weer in beeld).

**ul**

2. Voeg een transitie van 0.3s toe aan de ul om die soepel heen en weer te schuiven.

Nb. De class wordt toegevoegd aan de nav. Zo kunnen de ul en de button beide veranderen (allebei kinderen van de nav).

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

De button is onder de ul gepositioneerd.
Dat is onhandig ;-)

**nav button**

1. Positioneer de button boven de ul.
   (tip: z-index & position:relative;)

Het is ook een goed idee om de hamburger in een kruis te veranderen.

**nav.toonMenu button img**

1. Wijzig de content van de img naar het kruis.

Nb. De url voor het kruis staat in je code.

Voor de **blauwe**
en **rode** piste

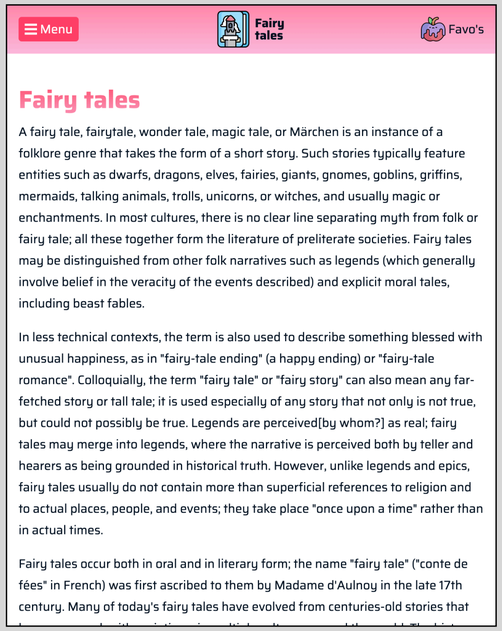
# vanaf hier
# responsiveness

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

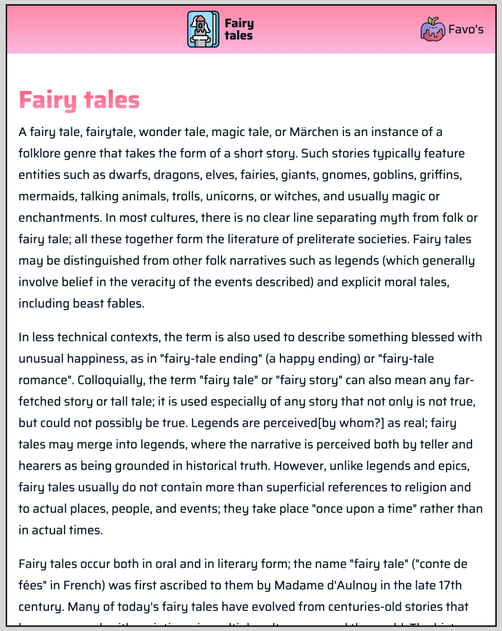
De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP




Nu verder met de responsiveness. Het menu past vanaf 38em in beeld. De button is dan niet meer nodig. Die als eerste verstoppen.

In de 38em media query:

**nav button**
1. Verstoppen.



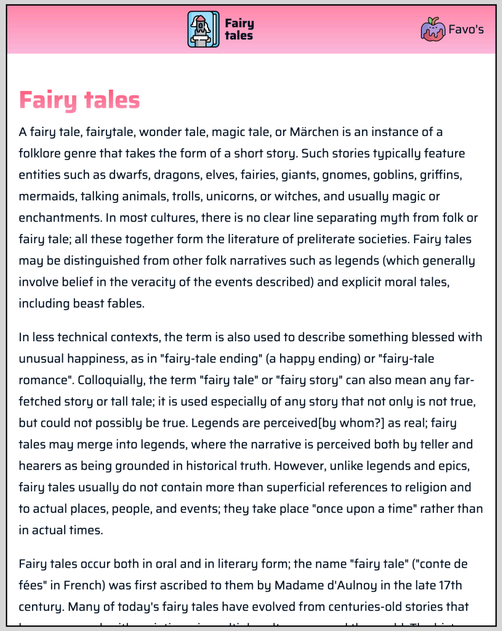

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

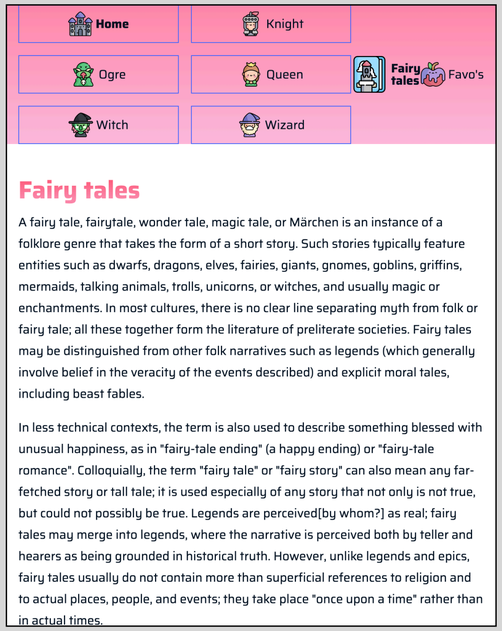



De ul weer op zijn ‘normale’ plek zetten. Daarvoor moet je in de media query een aantal dingen ‘ongedaan’ maken.

**nav ul**

1. Position (tip: static)
2. Translate (tip: unset)
3. Padding
4. Background

Nb. Het klopt dat het er niet uit ziet. Dat komt zo.



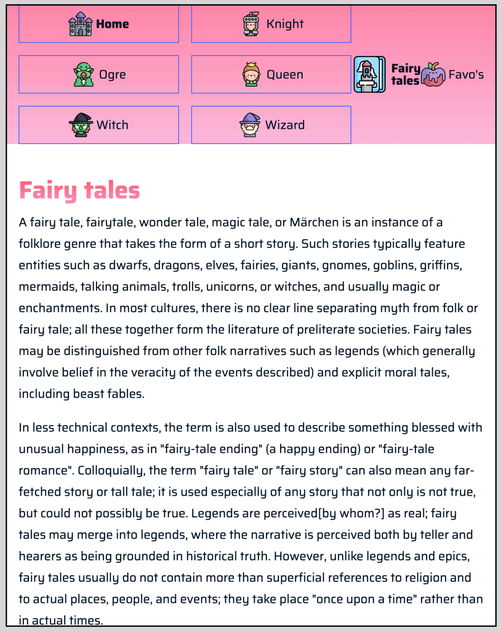

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

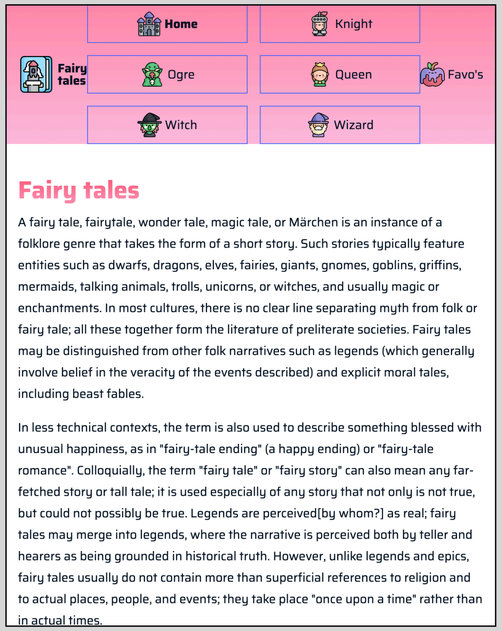



De nav weer op zijn normale plek zetten, zodat die in het midden staat.

**nav**

1. Midden in de rij (tip: order).

Nb. Het klopt dat het er nog niet uit ziet. Dat komt zo.



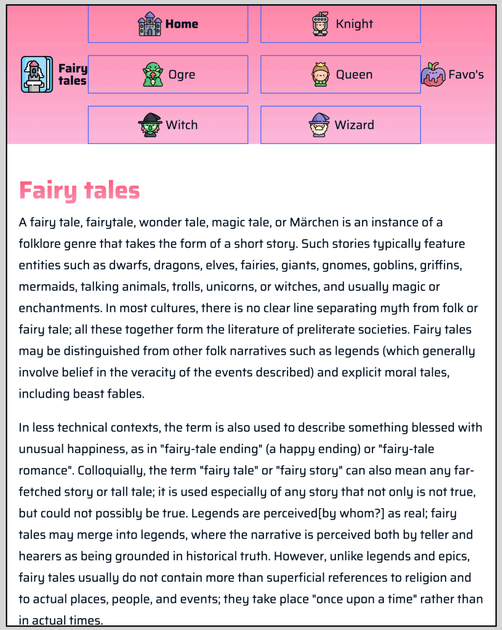

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

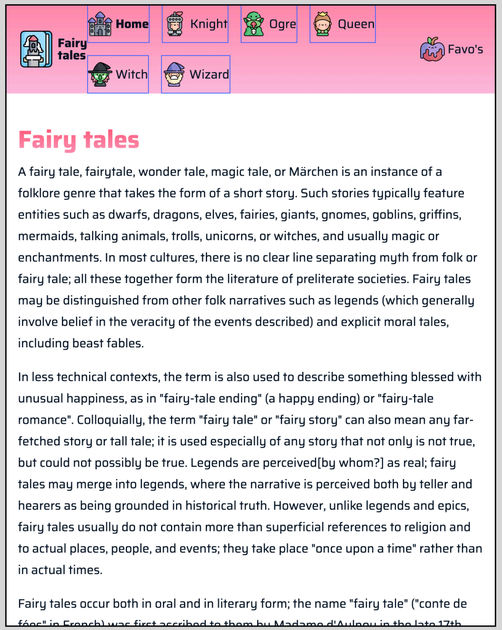



De li’s hebben een minimale breedte en vullen samen de hele breedte als er ruimte over is.

**li**

1. Zorg in de media query dat de li’s geen minimale breedte meer hebben (tip: flex-basis).
2. Zorg in de media query dat de li’s niet meer groeien als er ruimte over is (tip: flex-grow).

Nb. Het klopt dat het er nog steeds niet uit ziet. Dat komt in de volgende stap.



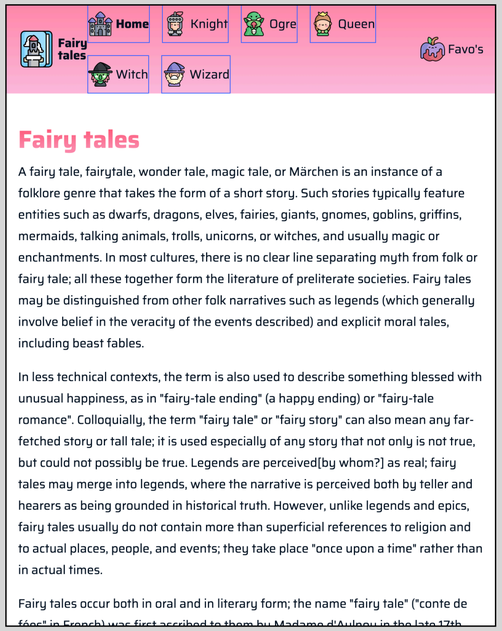

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP

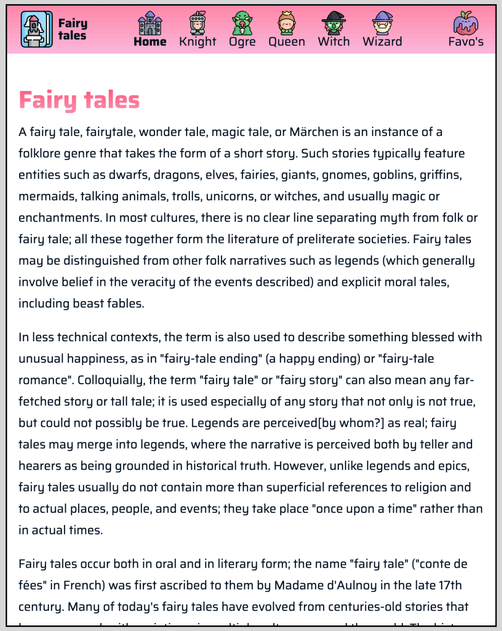



**nav a**

1. Zet de kinderen van de a’s in de nav boven elkaar (tip: flex-en plus flex-direction).
2. Verwijder de border.

**header > a:last-of-type**

3. Zet de kinderen van de favo’s a ook boven elkaar.



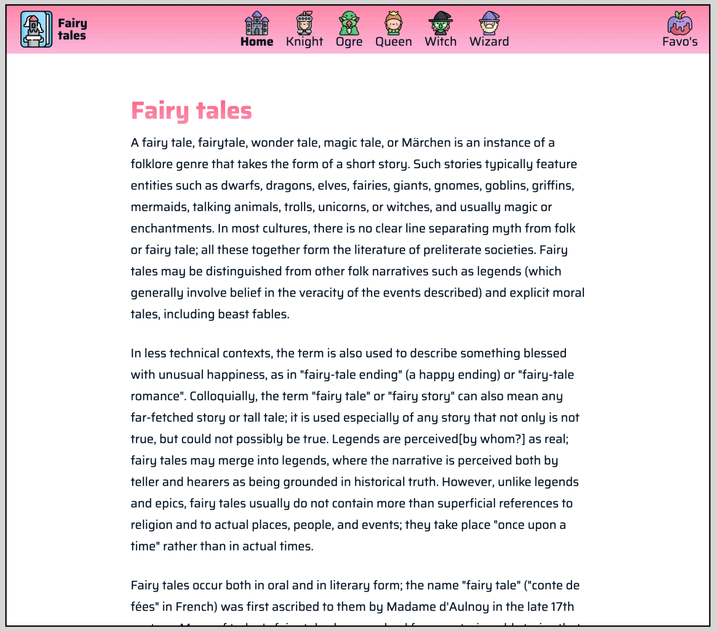

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive fixed-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/poMepOV
Uitwerking:
codepen.io/shooft/pen/eYqvyLP



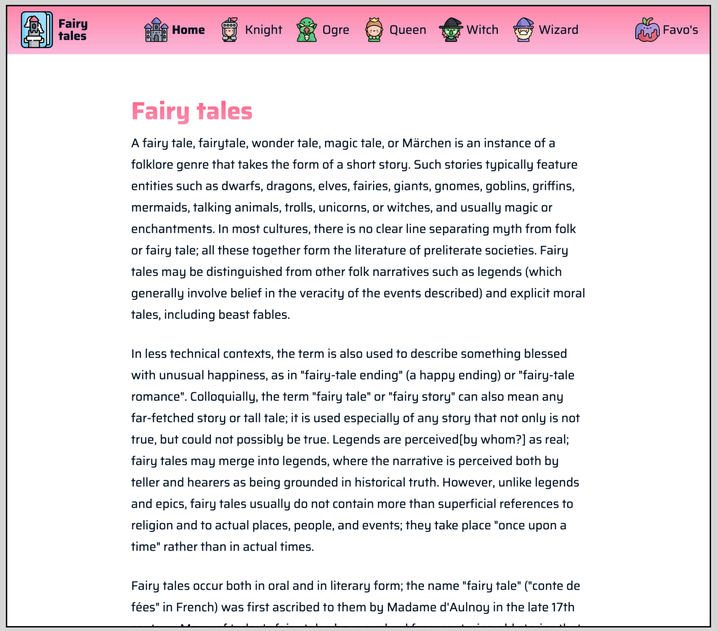

Als het scherm nog groter wordt passen de plaatjes en teksten weer naast elkaar.

In de 57em media query:

**nav a, header > a:last-of-type**

1. Zet de kinderen van de a’s weer naast elkaar (tip: flex-direction).

Et voilà!




Voor de **blauwe**
en **rode** piste

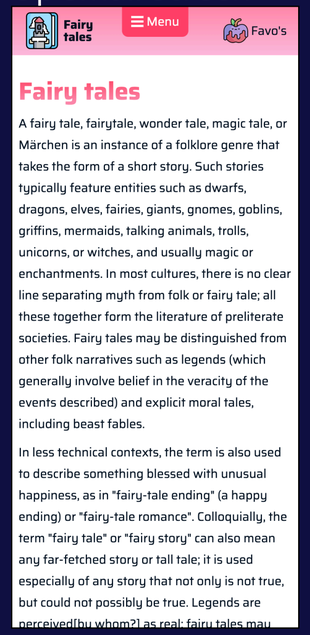
**JS 3-stap - Oefening 2**

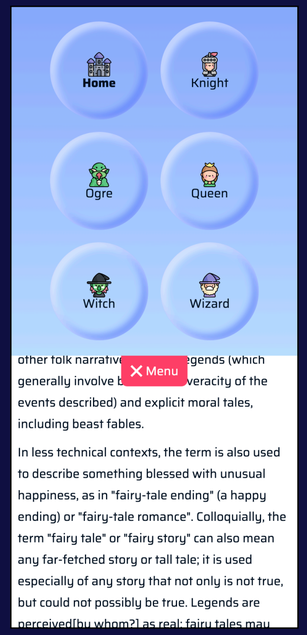
# Responsive moving-button burger

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

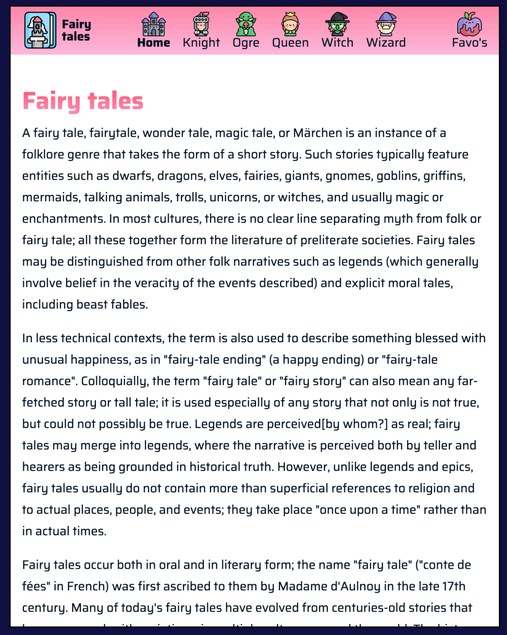
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

Op small screen

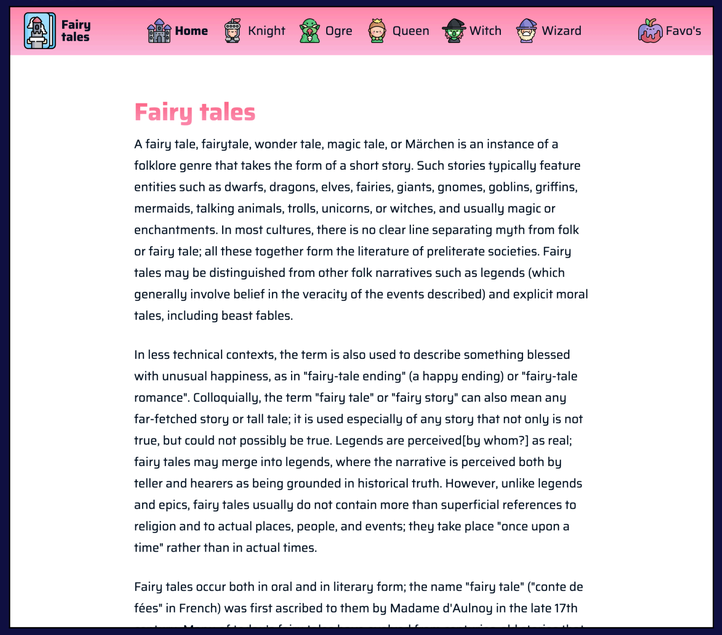



Vanaf 38em




Vanaf 57em




Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

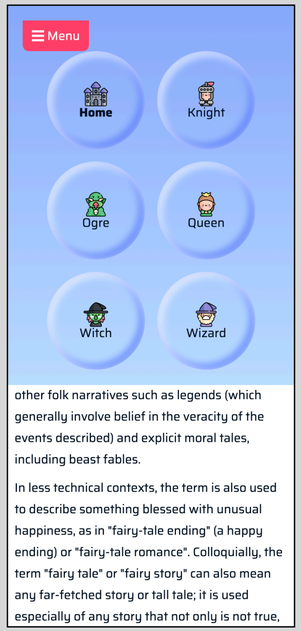
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

De basisvormgeving krijg je cadeau.

**ul**

1. Positioneer de ul fixed.
2. Over de hele breedte bovenaan het scherm (hint: left, right, top).



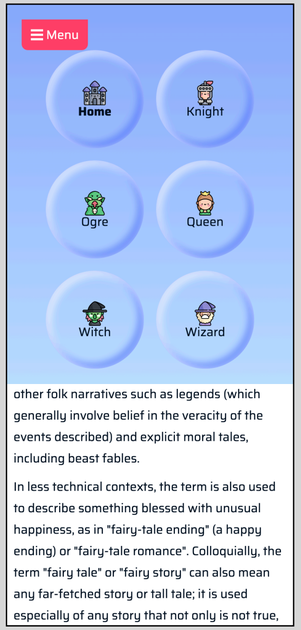

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

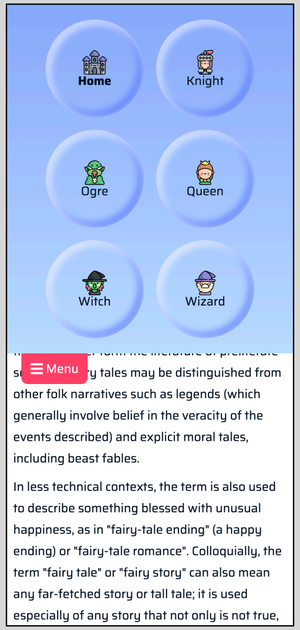
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK




**nav button**

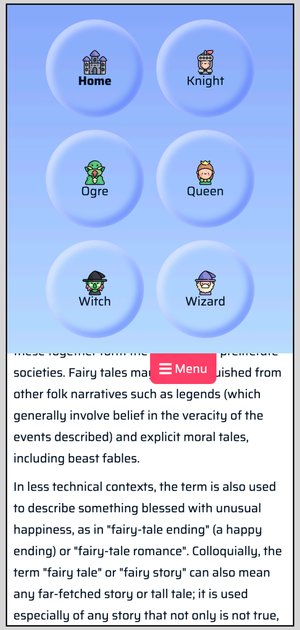
1. Positioneer de button op een vaste plek in de nav.
2. Zet de button onder de nav (tip: gebruik daarvoor niet bottom maar top - de afstand is de hele hoogte van de nav:100%).




**nav button**

3. Zet de button met left 50% van de linkerkant.

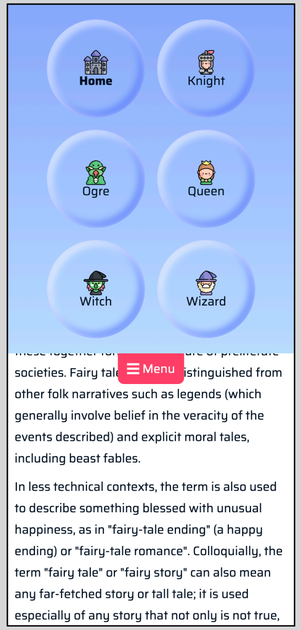
Nb. De button staat nu iets te ver. Dat fix je in de volgende stap.




**nav button**

4. Verplaats de button 50% van zijn eigen breedte naar links met translate: x% y%;

Nb. Naar links is negatief.



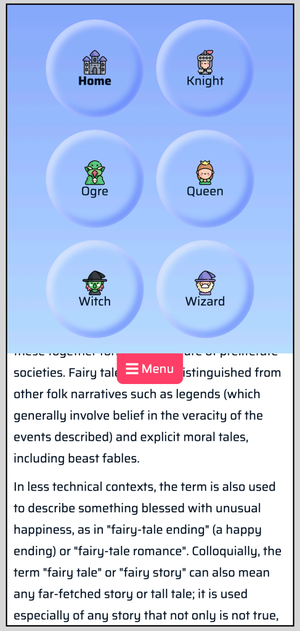

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

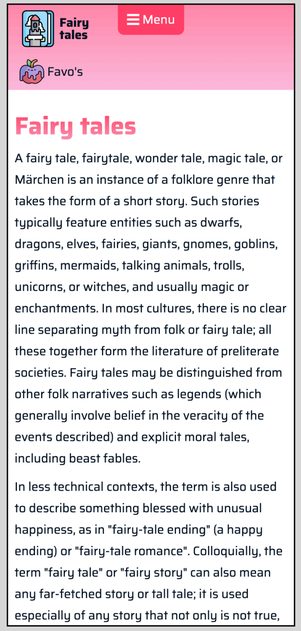



**nav**

1. Schuif de nav uit beeld met
   translate: x% y%;

Nb. De nav moet zijn eigen hoogte (100%) naar boven bewegen.

Nb. Naar boven in negatief.



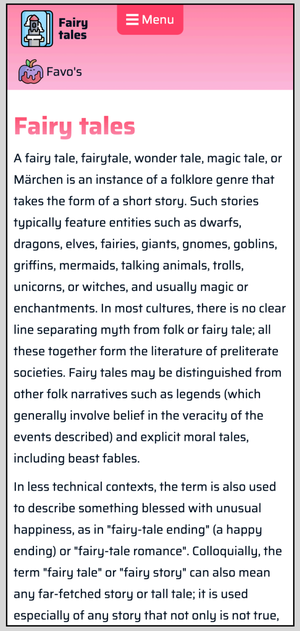

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

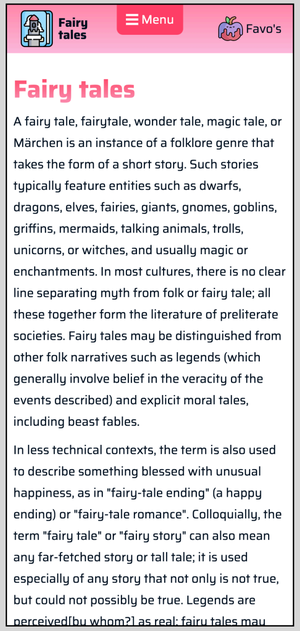
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK



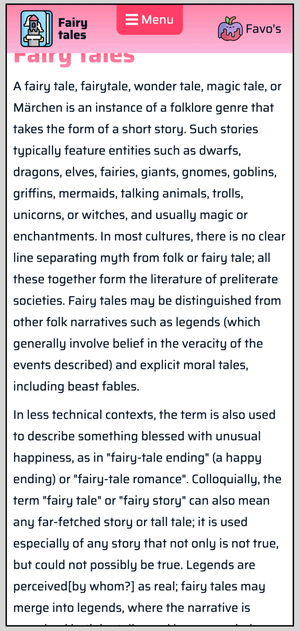

**header**

1. Flex.
2. Extra ruimte netjes verdelen.
3. Verticaal in het midden.




**header**

4. Bovenaan laten plakken.



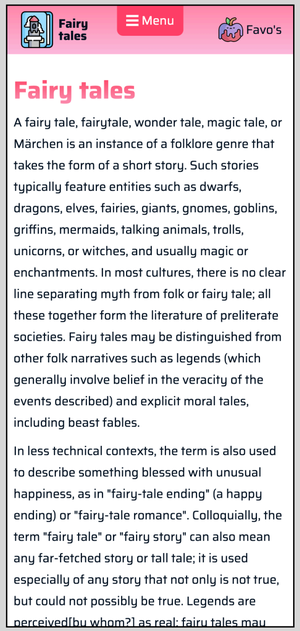

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

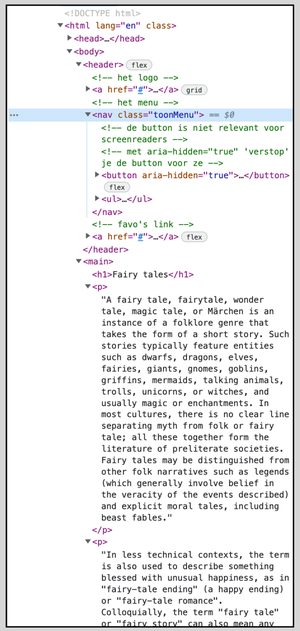



## JS 3-stap 1/2

1. Zoek de “menu” button (querySelector) en sla die op in een variabele.
2. Laat de button naar clicks luisteren (addEventListener) en roep de functie “openMenu” aan als er geklikt wordt.
3. In de functie “toonMenu”:
   - Zoek de nav op en sla die op in variabele.
   - Voeg de class “toonMenu” toe aan de nav (classList & toggle).

Nb. Op het scherm zie je niets veranderen. In de inspector kun je wel controleren of de class wordt toegevoegd.

Bonus: Verwijder de class “toonMenu” ook als je op de Escape toets drukt.



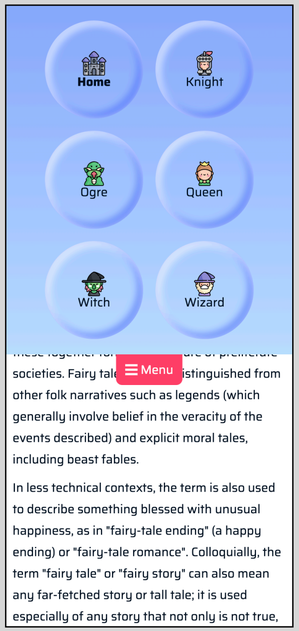

Verder met de CSS

**nav.toonMenu**

1. Maak de verschuiving uit stap 3 weer ongedaan met translate: x% y%; (het menu staat dan weer in beeld).

**nav**

2. Voeg een transitie van 0.3s toe aan de nav om die soepel heen en weer te schuiven.



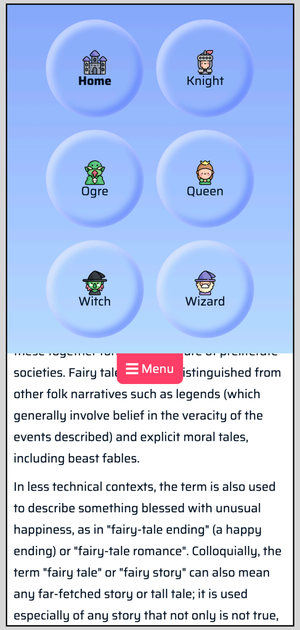

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

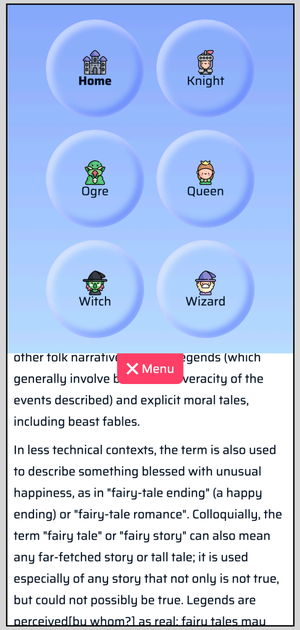



Het is een goed idee om de hamburger in een kruis te veranderen.

**nav.toonMenu button img**

1. Wijzig de content van de img naar het kruis.

Nb. De url voor het kruis staat in je code.




Voor de **blauwe** en **rode** piste

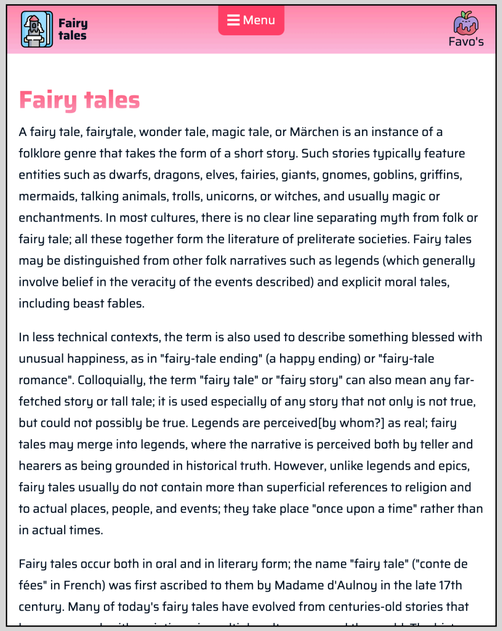
# vanaf hier
# responsiveness

**JS 3-stap**

# Responsive moving-button burger

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

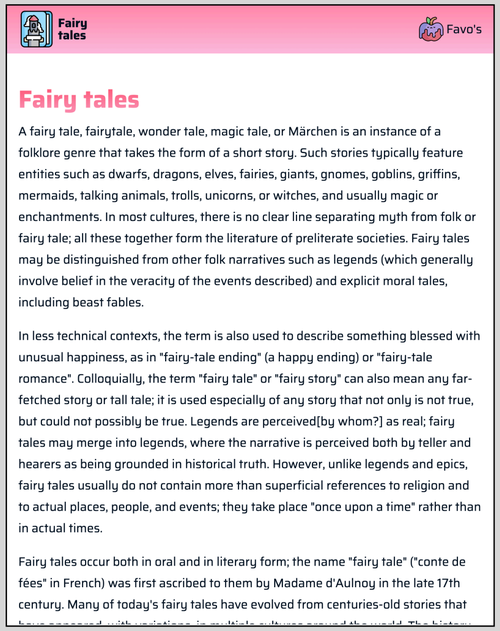
Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen




Nu verder met de responsiveness. Het menu past vanaf 38em in beeld. De button is dan niet meer nodig. Die als eerste verstoppen.

In de 38em media query:

**nav button**
1. Verstoppen.



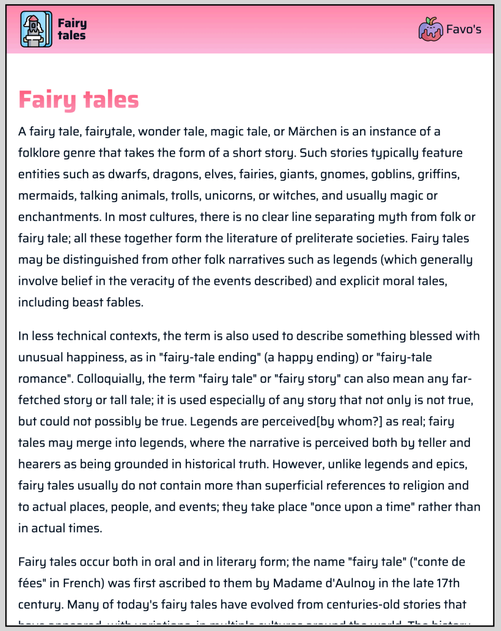

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

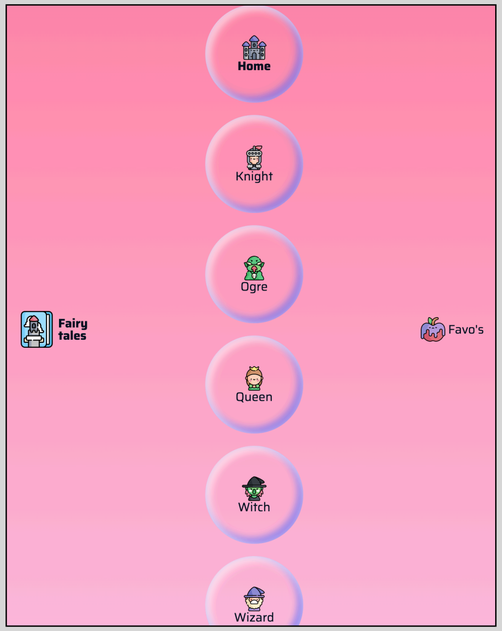



De nav weer op zijn ‘normale’ plek zetten. Daarvoor moet je in de media query een aantal dingen ‘ongedaan’ maken.

**nav**

1. Position (tip: static)
2. Translate (tip: unset)
3. Padding
4. Background

Nb. Het klopt dat het er niet uit ziet. Dat komt zo.



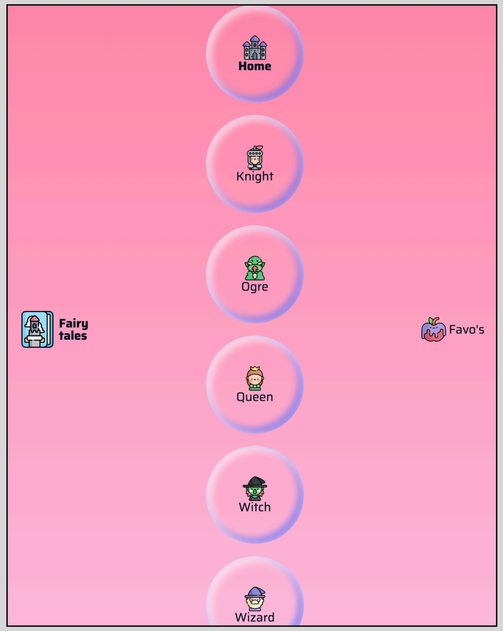

Voor de **blauwe** en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK



De ul heeft display grid en nog wat andere properties om de li’s in zo veel kolommen van 8em als er passen te layout-en.

Om ze in een rij naars elkaar te zetten moet de ul display flex krijgen (de overige grid properties die de ul had zijn dan gelijk ook niet meer relevant).

In de 38em media query:

**ul**
1. flex.

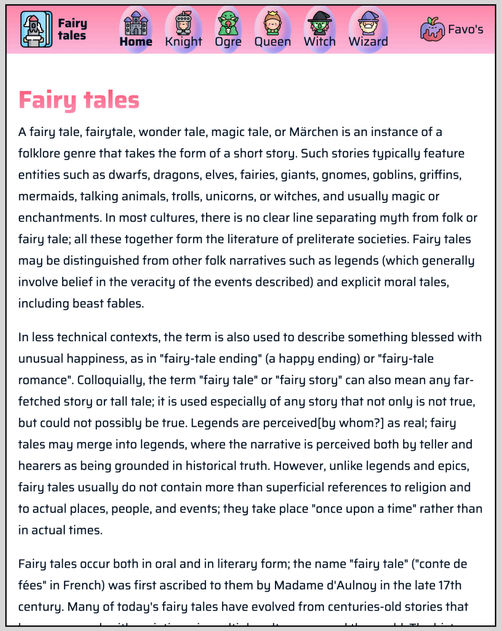
Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK

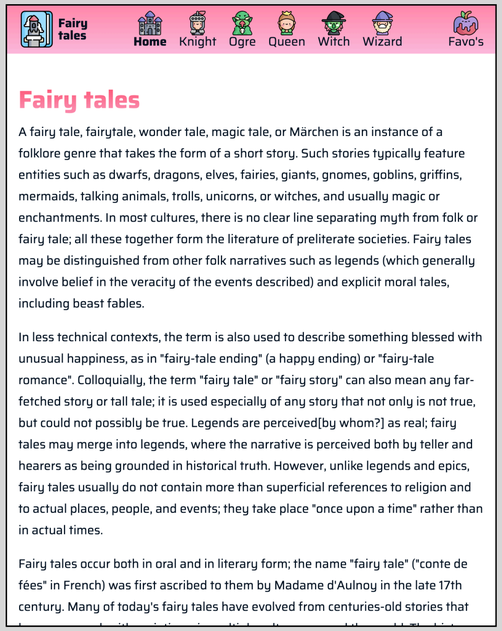



**nav a**

1. Maak de aspect-ratio ongedaan (tip:unset).
2. Verwijder de border-radius (tip:unset).
3. Verwijder de box-shadow (tip:unset).

**header > a:last-of-type**

4. Zet de kinderen van de favo’s a boven elkaar (tip: flex-en plus flex-direction).



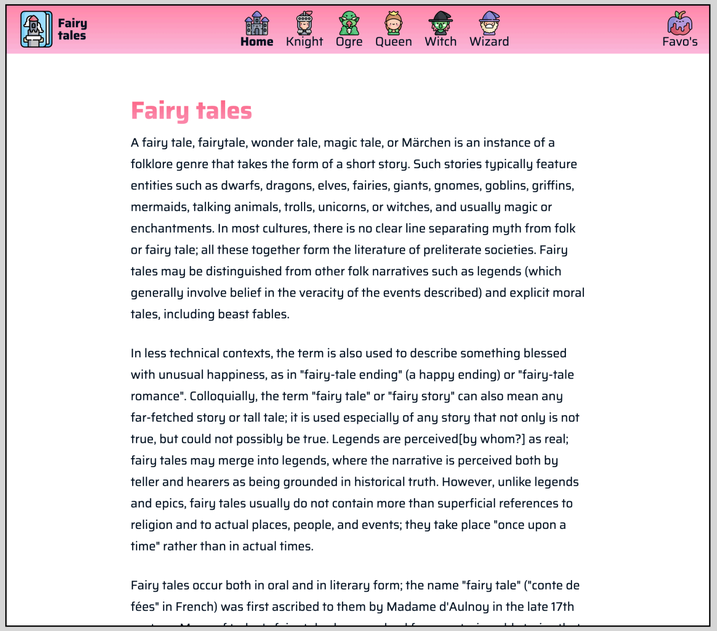

Voor de **blauwe**
en **rode** piste

**JS 3-stap**

# Responsive moving-button burger

Eerst een ‘hamburger’ - daarna het menu op een breder scherm direct tonen

De code voor jou:
codepen.io/shooft/pen/qBerpgW
Uitwerking:
codepen.io/shooft/pen/vYoxpwK



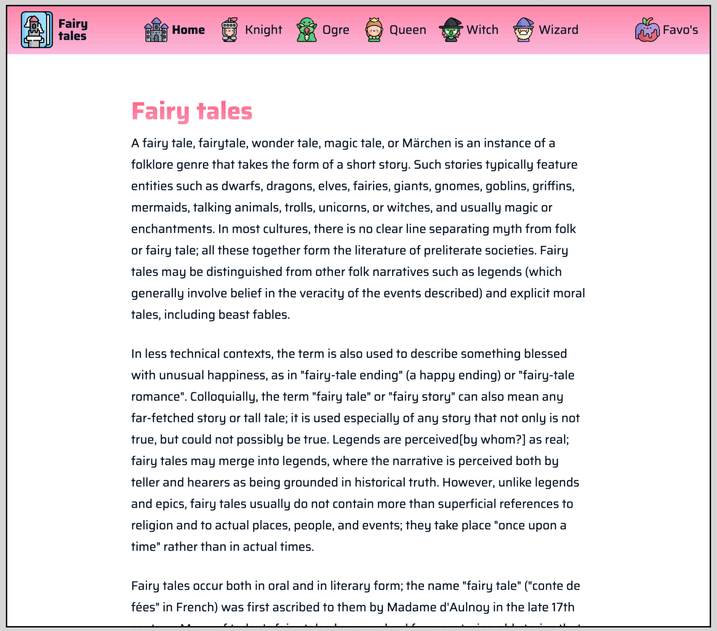

Als het scherm nog groter wordt passen de plaatjes en teksten weer naast elkaar.

In de 57em media query:

**nav a, header > a:last-of-type**

1. Zet de kinderen van de a’s weer naast elkaar (tip: flex-direction).

Et voilà!




Voor de **blauwe**
en **rode** piste

# #2 POSITION

## positioning 2

CSS challenge

DAY 24-25