**[Evaluatie 5wd2-superhero-website](https://github.com/PiusX-5BIN/5wd2-superhero-website-Mesbih)**

# Naam

### Markup

| A | B | C | D | E |
| --- | --- | --- | --- | --- |
| Elementen worden correct genest. Elk element bestaat uit de juiste openings- en sluitingstag. Elementen zijn voorzien van de vereiste attributen en elk attribuut is ingevuld met een correcte waarde. Bronnen zijn lokaal opgeslagen. | Elementen worden correct genest. Elk element bestaat uit de juiste openings- en sluitingstag. Elementen zijn voorzien van de vereiste attributen. Attributen bevatten een foutieve waarde. Bronnen zijn lokaal opgeslagen. | Elementen worden correct genest. Elk element bestaat uit de juiste openings- en sluitingstag. Elementen zijn niet altijd voorzien van de vereiste attributen. attributen bevatten een foutieve waarde. Bronnen zijn lokaal opgeslagen. | Elementen worden niet correct genest. Elk element bestaat uit de juiste openings- en sluitingstag. Elementen zijn niet voorzien van de vereiste attributen. attributen bevatten een foutieve waarde. Externe bronnen worden gebruikt, waar interne bronnen een betere oplossing zijn. | Elementen worden niet correct genest. Elementen missen de juiste openings- of sluitingstag. Attributen werden niet correct gebruikt. Er zijn te weinig elementen aanwezig om de inhoud correct over te brengen. |
| De semantisch juiste elementen werden gebruikt. De id's en classes van elementen zijn ingevuld waar nodig en verbeteren de semantiek. Elementen zijn onderverdeeld in grotere en kleinere elementen, en er is gekozen voor een optimale oplossing. | De semantisch juiste elementen werden gebruikt. De id's en classes van elementen zijn ingevuld waar nodig en verbeteren de semantiek. Elementen zijn onderverdeeld in grotere en kleinere elementen. Er zijn echter betere oplossingen die je had kunnen bedenken en gebruiken. | De semantisch juiste elementen werden niet altijd gebruikt. De id's en classes van elementen zijn ingevuld waar nodig en verbeteren de semantiek. Elementen zijn onderverdeeld in grotere en kleinere elementen. | De semantisch juiste elementen werden gebruikt. De id's en classes van elementen zijn niet ingevuld waar nodig en verbeteren de semantiek onvoldoende. Elementen zijn onderverdeeld in grotere en kleinere elementen. | De semantisch juiste elementen werden niet of verkeerd gebruikt. De id's en classes zijn onvoldoende ingevuld. Elementen zijn onvoldoende onderverdeeld in grotere en kleinere elementen. |
|  |  |  |  |  |

### Opmaak

| A | B | C | D | E |
| --- | --- | --- | --- | --- |
| Selectors zijn correct en voldoende specifiek gedefinieerd. Er werd gekozen voor de correcte stijlregels met de juiste waardes en eenheden. Selectors werden samengevoegd waar nodig. | Selectors zijn correct en voldoende specifiek gedefinieerd. Er werd gekozen voor de correcte stijlregels met de juiste waardes en eenheden. Selectors worden te weinig samengevoegd. | Selectors zijn correct en voldoende specifiek gedefinieerd. Selectors worden te weinig samengevoegd. Er werd gekozen voor de correcte stijlregels, maar de waardes en eenheden werden niet correct gebruikt. | Selectors zijn correct gedefinieerd, maar zijn vaak niet specifiek genoeg. Selectors worden te weinig samengevoegd. Er werd gekozen voor de correcte stijlregels, maar de waardes en eenheden werden niet correct gebruikt. | Selectors zijn verkeerd of niet specifiek genoeg gedefinieerd. Selectors worden te weinig samengevoegd. Stijlregels, waardes en eenheden werden niet correct gebruikt. |

### Statische Websites

| A | B | C | D | E |
| --- | --- | --- | --- | --- |
| Layout, ruimte en interactie zijn correct geïmplementeerd. De website is responsive en Mobile First ontwikkeld. Stijlregels zijn duidelijk en overzichtelijk verzameld in externe stylesheets. Je gebruikt relatieve eenheden (em, rem) en custom properties (variabelen) waar nodig. | Layout, ruimte en interactie zijn correct geïmplementeerd. De website is responsive en Mobile First ontwikkeld. Stijlregels zijn duidelijk en overzichtelijk verzameld in externe stylesheets. Je gebruikt custom properties (variabelen) waar nodig. | De interactie op jouw website is onvoldoende geïmplementeerd. Layout en ruimte zijn correct. De website is responsive en Mobile First ontwikkeld. Stijlregels zijn duidelijk en overzichtelijk verzameld in externe stylesheets. | Ruimte en interactie zijn onvoldoende geïmplementeerd. De website is onvoldoende responsive. Je layout werkt en je hebt de website voor en deel ontwikkeld volgens het Mobile First principe. | Layout, ruimte en interactie zijn onvoldoende geïmplementeerd. De website is niet responsive. Er werd niet gewerkt volgens het Mobile First principe. |

### Je index:

Verwijder dubbele id-attributen: zoals id="SuperheroProfile" en id="Profile". Zorg ervoor dat elk id-attribuut uniek is in de HTML-pagina.

Gebruik betekenisvolle afbeeldingsnamen: De afbeeldingsbronnen hebben momenteel generieke namen zoals "Spiderman1.jpg", "Mj.jpg", enz. Het is beter om meer betekenisvolle namen te geven die de inhoud van de afbeeldingen beschrijven, bijvoorbeeld "spiderman-profile.jpg" of "mj-testimonial.jpg". Dit verbetert de leesbaarheid van de code en is nuttig voor toegankelijkheid.

Organiseer CSS-klassen: Het is een goede praktijk om CSS-klassen te organiseren en te groeperen op basis van hun functie en herbruikbaarheid. Dit verbetert de onderhoudbaarheid van de code en vergemakkelijkt toekomstige aanpassingen. Je kunt bijvoorbeeld gerelateerde klassen groeperen in aparte secties en deze secties scheiden met commentaar om het overzicht te behouden.

Voeg alt-attributen toe aan afbeeldingen: Het is belangrijk om voor elke afbeelding een alt-attribuut toe te voegen dat een beschrijving geeft van de afbeelding. Dit helpt bij toegankelijkheid voor mensen met visuele beperkingen en is ook nuttig als de afbeelding niet kan worden weergegeven.

Gebruik het type-selector spaarzaam: Het gebruik van het type-selector (bijv. h1, p, img) kan handig zijn, maar vermijd overmatig gebruik ervan in combinatie met class-selectors. Het is meestal beter om te vertrouwen op class-selectors om de stijl van elementen aan te passen.

Optimaliseer afbeeldingen: Zorg ervoor dat de afbeeldingen geoptimaliseerd zijn voor webgebruik. Je kunt afbeeldingen comprimeren zonder merkbare kwaliteitsverlies om de laadtijd van de pagina te verbeteren. Je site is nu niet responsive voor kleinere schermen.

Gebruik externe bestanden voor JavaScript: Als je JavaScript-code hebt, is het beter om deze in externe bestanden te plaatsen en deze bestanden via een script-tag te koppelen aan de HTML-pagina. Dit houdt de HTML-code schoon en maakt het gemakkelijker om de JavaScript-code te beheren en te hergebruiken.

De CSS-compositie: Over het algemeen ziet de CSS-compositie er redelijk in orde uit, maar er zijn een paar punten waarop je de code kunt verbeteren:

Gebruik consistente indentatie: Zorg ervoor dat de CSS-regels consequent zijn ingesprongen om de leesbaarheid te verbeteren en de code netter te maken.

Wees consistent in het gebruik van een puntkomma: Hoewel de meeste regels correct zijn beëindigd met een puntkomma, ontbreken er enkele puntkomma's in de code. Zorg ervoor dat je elke regel afsluit met een puntkomma om fouten te voorkomen.

Wees voorzichtig met het gebruik van ID-selectors: ID's moeten uniek zijn in een document, dus het is beter om class-selectors te gebruiken voor herbruikbare stijlen. In de gegeven code wordt een ID-selector #pSuperheroProfile gebruikt, maar het lijkt erop dat het eigenlijk #SuperheroProfile moet zijn om overeen te komen met de HTML-sectie-id.

Vermijd het gebruik van harde pixels voor afmetingen: Hoewel het instellen van exacte pixels voor afmetingen in sommige gevallen nodig kan zijn, kan het de responsiviteit van de website belemmeren. Overweeg om relatieve eenheden zoals percentages of **em** te gebruiken, vooral voor elementen die zich aanpassen aan verschillende schermformaten.

Maak gebruik van class-namen met betekenis: Zorg ervoor dat de class-namen betekenisvol zijn en de functie van de elementen beschrijven. Dit maakt de code gemakkelijker te begrijpen en te onderhouden.

Minimaliseer het gebruik van absolute positionering: Absolute positionering kan soms handig zijn, maar het kan ook problemen veroorzaken bij responsieve lay-outs. Probeer waar mogelijk gebruik te maken van flexbox of grid-layouts in plaats van absolute positionering.

Consolideer vergelijkbare eigenschappen: In sommige gevallen worden eigenschappen herhaald voor verschillende selectors. Overweeg om vergelijkbare eigenschappen te consolideren en op één plek te plaatsen om de code DRY (Don't Repeat Yourself) te houden.

De utilities-classes lijken redelijk in orde. Hier zijn een paar opmerkingen:

tussentitels: De class.tussentitels ziet er goed uit met betrekking tot de font-weight, font-family, font-size en kleur. Het lijkt er echter op dat de afbeelding die als achtergrond voor de body wordt gebruikt een maroon-achtergrondkleur heeft. Als de maroon-tekst niet goed leesbaar is op die achtergrond, wil je misschien de tekstkleur aanpassen voor betere leesbaarheid.

body: Het toevoegen van een achtergrondafbeelding aan de body met background-image is prima. Zorg er echter voor dat je een fallback-achtergrondkleur toevoegt voor het geval de afbeelding niet kan worden geladen. Dit helpt ervoor te zorgen dat de inhoud leesbaar blijft, zelfs als de afbeelding ontbreekt.

minititel: De class .minititel ziet er goed uit met betrekking tot de kleur, lettertype en lettergrootte. Zorg ervoor dat de achtergrondkleur of achtergrondafbeelding van de container waarin de minititels zich bevinden voldoende contrast biedt om de tekst leesbaar te maken.

pictures: De class .pictures lijkt in orde met de toegevoegde rand rond de afbeeldingen.

Het is ook aan te raden om de site op verschillende apparaten en schermformaten te testen om ervoor te zorgen dat de stijlen goed worden weergegeven.