

REA COLLEGE

auteur: Tom Bartels

versie: 4-1-2018

In deze les komen de volgende onderwerpen aan bod:

- Page Speed
- Time to First Byte
- Web Page Performance Test
- PageSpeed Insights
- Afbeeldingen comprimeren
- Minimaliseren van code
- Above the fold & scroll
- Modulaire CSS
- Asynchrone JavaScript

- Een belangrijk onderwerp binnen SEO is de **page speed** van een pagina.
- Sinds de mobiele revolutie van het internet is de laadtijd van een pagina belangrijker dan ooit.
- Wanneer je op je mobieltje "even snel" iets wil opzoeken (zoals actuele reisinformatie of een feitje om je gelijk te halen in een discussie) wil je niet lang hoeven wachten totdat de pagina ingeladen is.

Page Load Speed
A Critical Factor For SEO




- Testen hebben aangetoond dat gebruikers afhaken als het te lang duurt voordat de pagina is ingeladen.
- Er is dus een verband tussen de laadtijd en de **bounce rate** van een pagina.




As page load time goes from:

1s to 3s the probability of bounce **increases 32%**



1s to 5s the probability of bounce **increases 90%**



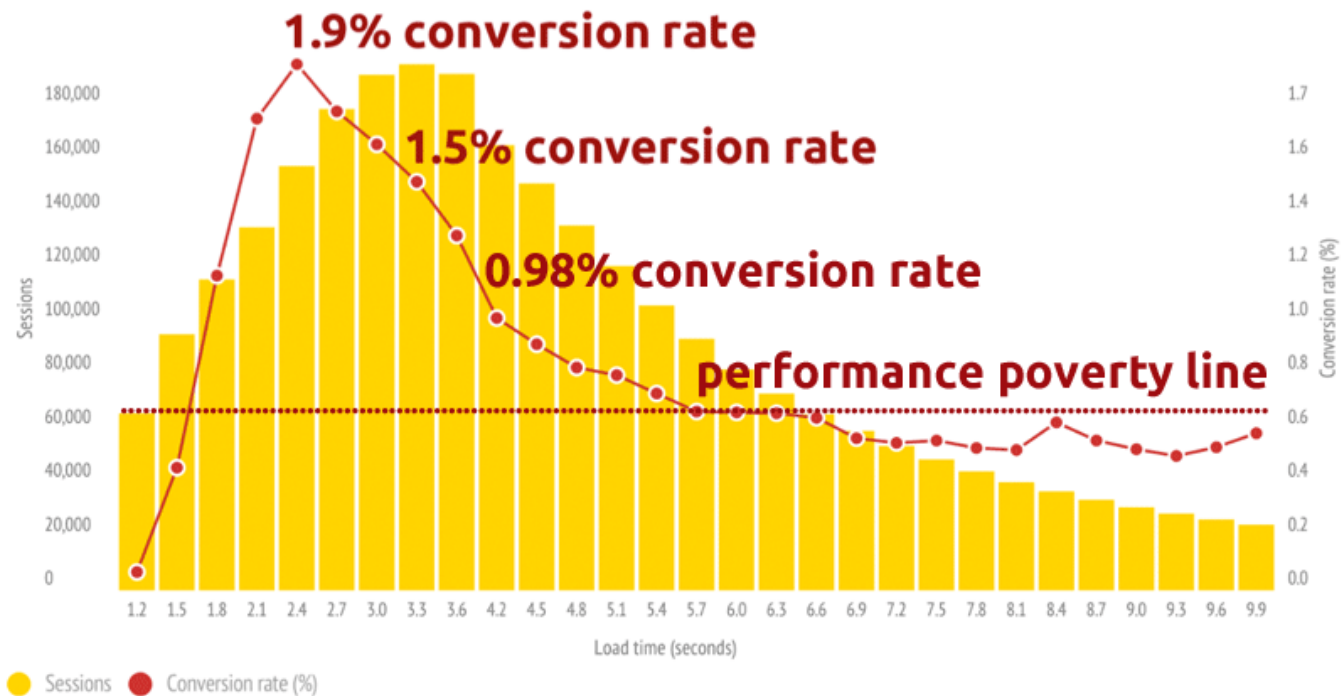
1s to 6s the probability of bounce **increases 106%**



1s to 10s the probability of bounce **increases 123%**



- Andersom is er ook een verband tussen de laadtijd en conversie.
- Websites met pagina's die meer dan 2 seconden laadtijd hebben, hebben gemiddeld veel minder conversie (bijv. het plaatsen van een bestelling).



- Op 9 april 2010 heeft Google aangekondigd de laadtijd van de pagina ook mee te nemen als factor om de ranking van een bepaalde pagina binnen de SERP te bepalen.
 - <https://webmasters.googleblog.com/2010/04/using-site-speed-in-web-search-ranking.html>
- Belangrijk om te realiseren is dat een lagere laadtijd niet direct leidt tot een betere SERP-positie.
- Het omgekeerde is wel waar – je loopt risico gestraft te worden wanneer je een (te) trage website hebt.

- Sinds 4 november 2016 heeft Google een "mobile first" approach. Dat betekent dat ze de mobiele versie van een pagina prioriteit zullen geven boven de desktop versie bij het indexeren.
 - <https://webmasters.googleblog.com/2016/11/mobile-first-indexing.html>
- Hierdoor is het nog belangrijker geworden dat je ervoor zorgt dat je pagina snel ingeladen wordt, wil je een penalty van Google voorkomen.

Page Load Speed
A Critical Factor For SEO



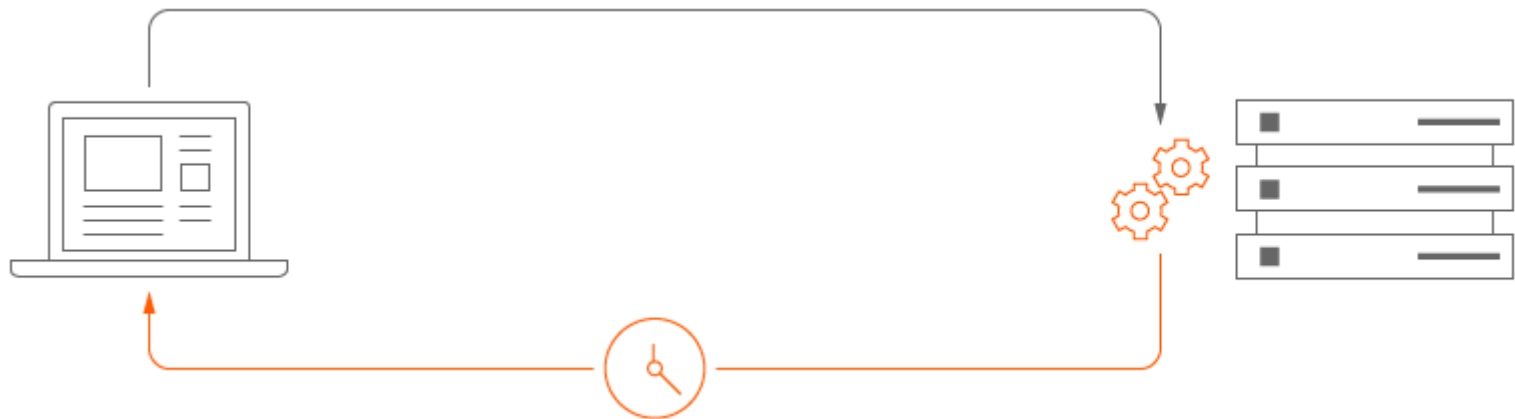
- De laadtijd van een pagina wordt doorgaans bepaald door het binnen halen van alle *resources*. Deze zijn immers nodig om de pagina te tonen.
- De voornaamste onderdelen die gedownload moeten worden zijn:
 - HTML code
 - CSS code
 - JavaScript code
 - Afbeeldingen
 - Lettertypes
 - Video bestanden
 - Flash bestanden

- Voordat de browser kan beginnen met het downloaden van de pagina met alle resources, moet er eerst een verbinding worden gelegd tussen de *client* en de *server*.
- Elke keer wanneer je een pagina bezoekt wordt er dus een verzoek gestuurd naar de server waar de website op draait.
- Als reactie op dat verzoek zal de server – mits de pagina gevonden werd – beginnen met het beantwoorden van het verzoek.
- De server zal dan in kleine pakketjes alle benodigde data versturen naar de server.

REACOLLEGE

Time to First Byte

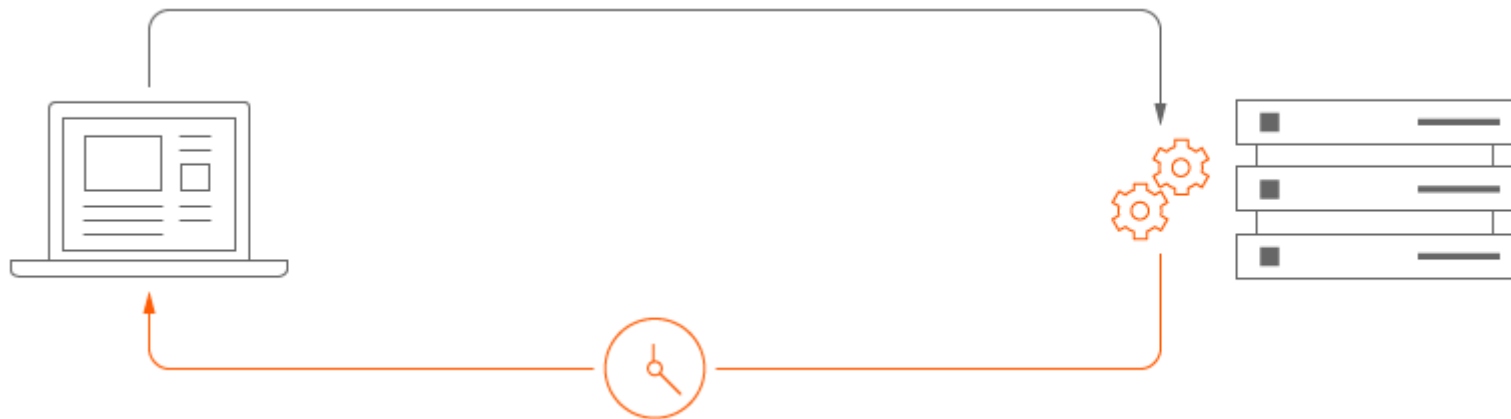
- De tijd die verstrijkt tussen het bezoeken van de URL en het moment dat het eerste pakketje data de client heeft bereikt noemen we de **Time to First Byte**.
- Deze **TTFB** gaat dus nog vooraf aan het daadwerkelijke downloaden van de resources.



REACOLLEGE

Time to First Byte

- Deze TTFB moet natuurlijk zo laag mogelijk zijn.
- Google adviseert in te zetten op maximaal ca. 200 milliseconden.
 - <https://developers.google.com/speed/docs/insights/Server>
- Je kunt hier op de website zelf echter weinig aan doen. Het zijn immers de server(s) en de serverinstellingen die de hoogte van de Time to First Byte bepalen.



- Een erg populair meetinstrument om te testen hoe hoog je page speed (inclusief de TTFB) is, is de website <https://www.webpagetest.org>.
- Met de **Web Page Performance Test** kun je een pagina naar keuze opgeven en deze laten testen vanaf een bepaalde bronlocatie.
 - De laadtijd van een pagina wordt immers ook (deels) bepaald door de plek waarvan je verbindt.
- Vervolgens krijg je een uitgebreid overzicht terug van alle testresultaten.

REACOLLEGE

Web Page Performance Test

Web Page Performance Test for

www.mikkoku.org

From: Amsterdam, NL - Go Daddy - Firefox - Cable
1/2/2018, 6:14:11 PM

Need help improving?

A

A

A

A

A

X

First Byte Time

Keep-alive Enabled

Compress Transfer

Compress Images

Cache static content

Effective use of CDN

- Summary
- Details
- Performance Review
- Content Breakdown
- Domains
- Screen Shot
- Image Analysis

Tester: WPT-EU-02-146.255.33.147
First View only
Test runs: 3
Re-run the test

[Raw page data](#) - [Raw object data](#)
[Export HTTP Archive \(.har\)](#)
[View Test Log](#)

Performance Results (Median Run)

	Load Time	First Byte	Start Render	Speed Index	Document Complete			Fully Loaded				
					Time	Requests	Bytes In	Time	Requests	Bytes In	Certificates	Cost
First View (Run 2)	2.562s	0.248s	1.027s	1267	2.562s	44	468 KB	2.928s	46	593 KB	11 KB	\$5---

[Plot Full Results](#)

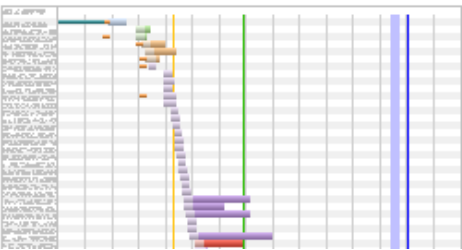
Test Results


Run 1:

Waterfall

Screen Shot

First View (2.612s)

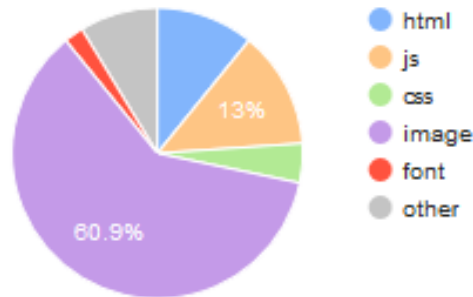




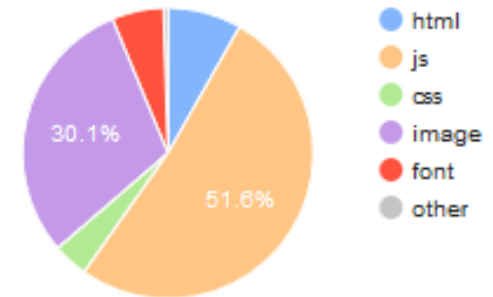
- Wanneer je naar beneden scrollt kun je de verhoudingen bekijken tussen de afzonderlijke soorten resources die ingeladen moeten worden.
- Hierbij wordt een onderscheid gemaakt tussen het aantal requests en het aantal bytes dat de bestanden in beslag nemen.

Content Breakdown

Requests

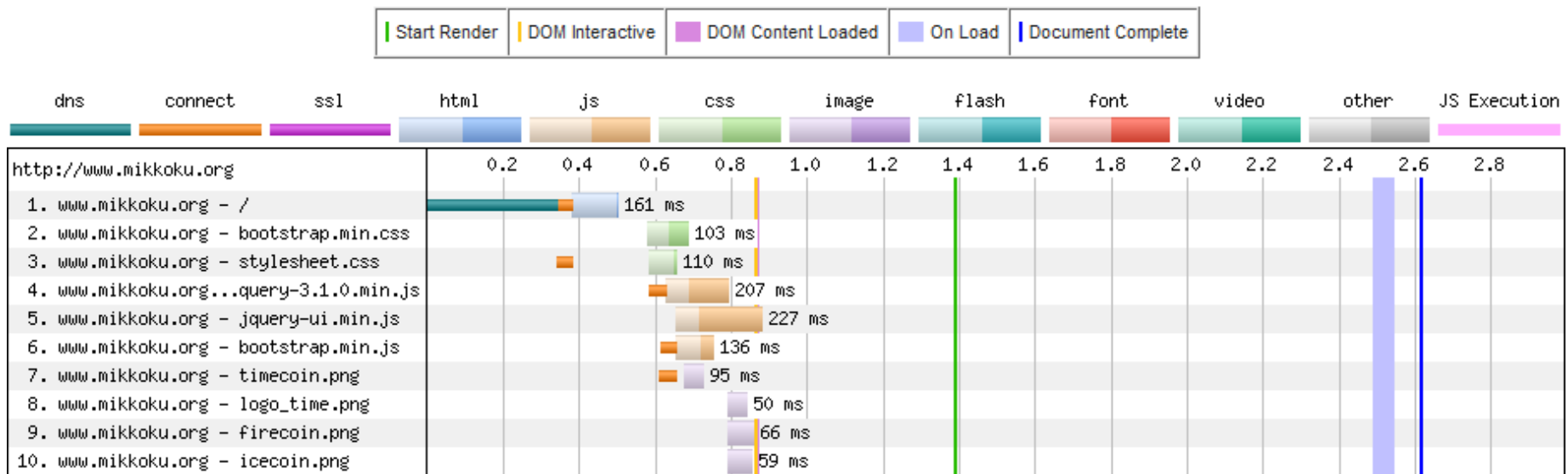


Bytes



- Wanneer je op de grafiek klikt krijg je de Waterfall View te zien.
- Hierin zie je precies hoelang het bij elke resource duurt om de verbinding te leggen (de lichte kleur) en het bestand daadwerkelijk te downloaden (de donkere kleur).

Waterfall View



- Ook de tool Seoptimer, waar we al eerder mee kennis hebben gemaakt, kan ons iets vertellen over de page speed en de TTFB.



Your Performance is very good!

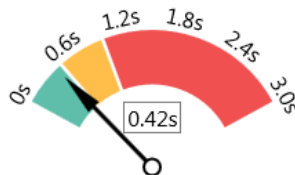
Page load speed and performance is important to ensure a positive user experience. Continue monitoring your performance to ensure you stay on top.

Page Speed Info

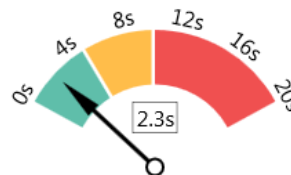
Your Page request's Server Response Time is reasonably low which is good for load speed and user experience.



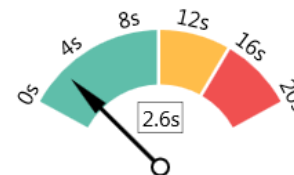
Server Response



All Page Content Loaded



All Page Scripts Complete



REACOLLEGE PageSpeed Insights

- Verreweg het belangrijkste hulpmiddel om je page speed te verbeteren is **PageSpeed Insights**, een dienst van Google.

PageSpeed Tools > Insights

HANDLEIDINGEN NASLAG VOORBEELDEN ONDERSTEUNING

PageSpeed Insights

http://mikkoku.org/

ANALYSEREN



Mobiel



Desktop

Good

92 / 100

Goed bezig. Op deze pagina zijn de meeste praktische tips voor prestaties al toegepast. De pagina zou goede functionaliteit aan gebruikers moeten leveren.



- Het programma geeft je niet alleen meldingen van wat er mis gaat, maar bovenal uitgebreide aanwijzingen over hoe je deze punten kan verbeteren.

Goed bezig. Op deze pagina zijn de meeste praktische tips voor prestaties al toegepast. De pagina zou goede functionaliteit aan gebruikers moeten leveren.

! Mogelijke optimalisaties

JavaScript en CSS in content boven de vouw die het weergeven blokkeren, verwijderen

▸ [Informatie over de oplossing voor dit probleem](#)

Gebruikmaken van browsercaching

▸ [Informatie over de oplossing voor dit probleem](#)

✓ Gevonden optimalisaties

▸ [Details weergeven](#)



**De resultaten worden 30 seconden in het cachegeheugen opgeslagen. Als u wijzigingen heeft aangebracht aan uw pagina, wacht u 30 seconden voordat u de test opnieuw uitvoert.*

- Een aantal van de verbeteringen die ook genoemd worden door PageSpeed Insights zullen we nu wat dieper op ingaan.
- De technieken die in de rest van de les behandeld zullen worden zijn allemaal client side en relatief snel voor elkaar te krijgen.
 - Vaak kun je met een aantal kleine *tweaks* al veel voor elkaar krijgen om de laadtijd van de pagina te optimaliseren.

- Afbeeldingen behoren tot één van de grootste factoren die de page speed bepalen.
- Vrijwel elke website maakt wel gebruik van afbeeldingen op voor zowel de layout als voor illustraties.
- Afbeeldingen kosten per definitie meer tijd om in te laden dan tekst.
- Het is daarom best practice dat je het aantal afbeeldingen toch tot een minimum probeert te beperken.
- Wanneer je iets voor elkaar kunt krijgen met pure CSS waardoor je geen afbeelding meer nodig hebt, heeft dat dus de voorkeur.
 - Denk bijvoorbeeld aan CSS3 gradients of borders.

- Niet alleen het downloaden van een afbeelding kost tijd, maar ook het leggen van de verbinding met het bestand.
- Het is qua page speed dus beter om één groot bestand te hebben met daarin verschillende losse afbeeldingen verzameld. Deze techniek noemen we **sprites**.
- Sprites lenen zich niet goed voor "echte" illustraties, maar wel voor layout-elementen zoals buttons en icoontjes.



- Ook moet je rekening houden met de bestandsgrootte van je bestand.
 - Gebruik dus nooit ongecomprimeerde bestandstypes als BMP en TIFF – deze zijn veel te groot!
- Wanneer je een afbeelding wil verkleinen op je pagina kun je dat verkleinen beter al van tevoren doen met een grafisch programma. Des te kleiner de afbeelding (qua afmetingen), des te kleiner de bestandsgrootte.
 - Als je met HTML of CSS een afbeelding verkleind zal de client immers toch de volledige grootte van de afbeelding moeten inladen.
- Bij het maken van JPG bestanden in Photoshop heb je tevens de mogelijkheid om zelf de mate van compressie in te stellen zodat je volledige controle hebt over de bestandsgrootte.

- Bij PNG en JPG bestanden kun je tevens nog een ander trucje toepassen om deze beter geschikt te maken voor het web.
- Deze bestandstypes kunnen namelijk nog verder worden gecomprimeerd zonder dat er kwaliteitsverlies plaatsvindt.
- Zie ook <https://developers.google.com/speed/docs/insights/OptimizeImages>.



Compressed by 90%- 5.93 KB



Uncompressed- 57.5 KB

- Wanneer je je afbeeldingen al van tevoren wil comprimeren voor het web kun je gebruik maken van tools als:
 - <http://compresspng.com/nl/> of <http://compressjpeg.com/nl/>.
 - <https://compressor.io/compress>.



Nederlands ▼

PNG VERKLEINEN

- 1 | Klik op UPLOAD en kies maximaal 20 PNG-afbeeldingen die je wilt comprimeren. Wacht tot de upload en de compressie processen zijn voltooid.
- 2 | Klik op DOWNLOAD ALLES om alle gecomprimeerde bestanden in één keer gegroepeerd in een ZIP-archief te downloaden. Je kunt ook elke afbeelding afzonderlijk downloaden.

- Als je website al live staat kun je je resources ook eenvoudig comprimeren door middel van de tool PageSpeed Insights.

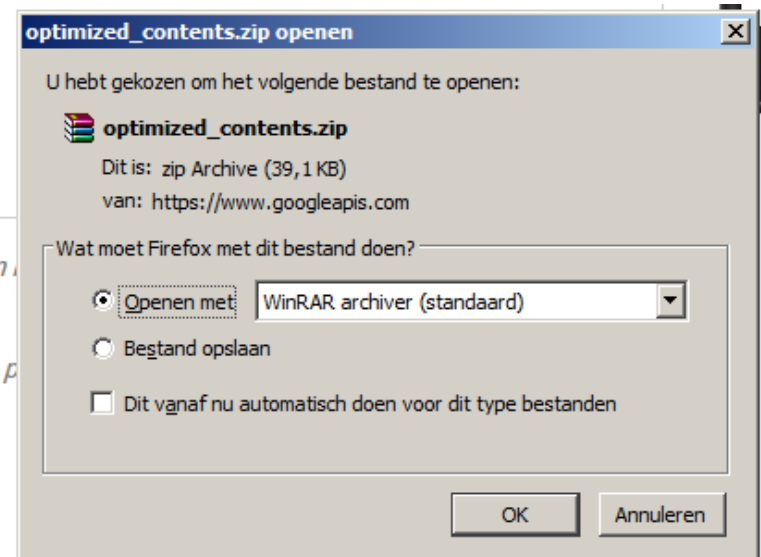
✓ Gevonden optimalisaties

► [Details weergeven](#)

Download geoptimaliseerde [afbeeldings](#)-, [JavaScript](#)- en [CSS-bronnen](#) voor deze pagina.

**De resultaten worden 30 seconden in het cachegeheugen opgeslagen. Als u wijzigingen maakt, moet u de test opnieuw uitvoeren voordat u de test opnieuw uitvoert.*

**Deze test controleert of algemene praktische tips voor prestaties zijn toegepast op een pagina. Het kan de functionaliteit voor gebruikers, maar garandeert dit niet. [Meer informatie](#).*



- Ook onze resources met code kunnen verkleind worden.
- Door alle lege ruimtes (spaties, tabs, witregels, etc.) weg te halen komt alle tekst tegen elkaar aan te staan waardoor er minder karakters in je bestand overblijven.

```

1  /* ----- General
2
3  @font-face {
4      src: url(
5          '../fonts/Orator-Std-Medium_34151.ttf'
6      );
7      font-family: Orator;
8  }
9
10 body {
11     background-color: #B97441;
12     color: rgba(247,251,255,1.0);
13     background-image: url(
14         '../img/layout/mikkoku_bg.png');
15     background-repeat: repeat-x;
16 }
17
18 h1, h2, h3, h4, h5, h6 {
19     font-family: Orator, Calibri, Arial,
20     sans-serif;
21 }

```

```

1  @font-face{src:url(
2      '../fonts/Orator-Std-Medium_34151.ttf');
3      font-family:Orator}body{background-color:
4      #B97441;color:rgba(247,251,255,1.0);
5      background-image:url(
6          '../img/layout/mikkoku_bg.png');
7      background-repeat:repeat-x}h1,h2,h3,h4,h5
8      ,h6{font-family:
9      Orator,Calibri,Arial,sans-serif}textarea{
10     resize:none}hr{color:
11     rgba(247,251,255,1.0);background-color:
12     rgba(247,251,255,1.0);height:4px}#
13     container{max-width:960px;min-height:
14     773px;background-repeat:no-repeat}.
15     mikkoku-img{border:1px solid
16     rgba(205,205,215,1.0);margin-bottom:40px
17     }.clearfix{clear:both}.back-to-top{width:
18     70px;height:70px;padding-top:6px;
19     padding-left:6px;font-size:50px;
20     box-sizing:box-size;position:fixed;

```

- Minder karakters betekent automatisch een kleinere bestandsgrootte.
- We noemen dat het **minimaliseren** van bestanden.
- Zie ook <https://developers.google.com/speed/docs/insights/MinifyResources>.

```

1  /* ----- General
2
3  @font-face {
4      src: url(
5          '../fonts/Orator-Std-Medium_34151.ttf'
6      );
7      font-family: Orator;
8  }
9
10 body {
11     background-color: #B97441;
12     color: rgba(247,251,255,1.0);
13     background-image: url(
14         '../img/layout/mikkoku_bg.png');
15     background-repeat: repeat-x;
16 }
17
18 h1, h2, h3, h4, h5, h6 {
19     font-family: Orator, Calibri, Arial,
20     sans-serif;
21 }

```

```

1  @font-face{src:url(
2      '../fonts/Orator-Std-Medium_34151.ttf');
3      font-family:Orator}body{background-color:
4      #B97441;color:rgba(247,251,255,1.0);
5      background-image:url(
6          '../img/layout/mikkoku_bg.png');
7      background-repeat:repeat-x}h1,h2,h3,h4,h5
8      ,h6{font-family:
9      Orator,Calibri,Arial,sans-serif}textarea{
10     resize:none}hr{color:
11     rgba(247,251,255,1.0);background-color:
12     rgba(247,251,255,1.0);height:4px}#
13     container{max-width:960px;min-height:
14     773px;background-repeat:no-repeat}.
15     mikkoku-img{border:1px solid
16     rgba(205,205,215,1.0);margin-bottom:40px
17     }.clearfix{clear:both}.back-to-top{width:
18     70px;height:70px;padding-top:6px;
19     padding-left:6px;font-size:50px;
20     box-sizing:box-size;position:fixed;

```

- Het minimaliseren van bestanden heeft vooral zin bij CSS- en JS-bestanden.
 - Bij HTML heeft het minder zin – HTML code wordt sowieso doorgaans automatisch gegenegeerd door PHP.

```

1  /* ----- General
2
3  @font-face {
4      src: url(
5          '../fonts/Orator-Std-Medium_34151.ttf'
6      );
7      font-family: Orator;
8  }
9
10 body {
11     background-color: #B97441;
12     color: rgba(247,251,255,1.0);
13     background-image: url(
14         '../img/layout/mikkoku_bg.png');
15     background-repeat: repeat-x;
16 }
17
18 h1, h2, h3, h4, h5, h6 {
19     font-family: Orator, Calibri, Arial,
20     sans-serif;
21 }

```

```

1  @font-face{src:url(
2      '../fonts/Orator-Std-Medium_34151.ttf');
3      font-family:Orator}body{background-color:
4      #B97441;color:rgba(247,251,255,1.0);
5      background-image:url(
6          '../img/layout/mikkoku_bg.png');
7      background-repeat:repeat-x}h1,h2,h3,h4,h5
8      ,h6{font-family:
9      Orator,Calibri,Arial,sans-serif}textarea{
10     resize:none}hr{color:
11     rgba(247,251,255,1.0);background-color:
12     rgba(247,251,255,1.0);height:4px}#
13     container{max-width:960px;min-height:
14     773px;background-repeat:no-repeat}.
15     mikkoku-img{border:1px solid
16     rgba(205,205,215,1.0);margin-bottom:40px
17     }.clearfix{clear:both}.back-to-top{width:
18     70px;height:70px;padding-top:6px;
19     padding-left:6px;font-size:50px;
20     box-sizing:box-size;position:fixed;

```

- Uiteraard wordt het aanbevolen om je bestanden pas te minimaliseren nadat je klaar bent met ontwikkelen en op het punt staat je website te uploaden.
- Je kunt ook eventueel twee verschillende versies erop na houden – een ongecomprimeerde versie voor ontwikkeling en een gecomprimeerde versie voor productie.
- Gecomprimeerde CSS en JavaScript bestanden hebben vaak een achtervoegsel als `.min.css` of `.min.js` om aan te geven dat het een geminimaliseerd bestand betreft.
- Bij veel plug-ins die je kunt downloaden kun je tevens kiezen of je de gecomprimeerde of de niet-gecomprimeerde versie wil hebben.

- Het kost te veel werk om je bestanden steeds handmatig te minimaliseren. Gelukkig zijn hier verschillende hulpmiddelen voor.
- Voor Notepad++ kun je bijvoorbeeld de plug-in CSSmin gebruiken, te downloaden op <https://github.com/ethanpil/npp-cssmin>.
- Je kunt je code ook online minimaliseren met websites als:
 - <https://www.minifier.org/>
 - <https://cssminifier.com/>
 - <https://csscompressor.com/>
 - <https://javascript-minifier.com/>
 - <https://jscompress.com/>

- Voor de volgende twee optimalisaties zullen we eerst even kort in moeten gaan op het concept **above the fold**.
 - In het Nederlands wordt dat ook wel "boven de vouw" genoemd.
- Als je een pagina laadt, is in eerste instantie alleen het bovenste deel zichtbaar.
- Op websites met veel content – en in het bijzonder op *Single Page Applications* – zul je naar beneden moeten scrollen om de volledige content te kunnen lezen.
- Het zichtbare deel van een website dat je ziet zonder te scrollen wordt ook wel **above the scroll** genoemd.

REACOLLEGE Above the fold & scroll



- Normaal gesproken wordt code altijd van boven naar beneden uitgevoerd – regel voor regel.
- Dat betekent dat eerst de volledige `<head>` met alle CSS en JavaScript code en JS code die daarin voorkomt eerst ingeladen (en uitgevoerd) moet worden voordat er een begin gemaakt kan worden aan de `<body>`.
- Een nieuwe ontwikkeling binnen de page speed optimalisatie is om ervoor te zorgen dat alles *above the fold* zo snel mogelijk zichtbaar is, terwijl het laden en verwerken van alles *below the fold* eventueel iets uitgesteld kan worden.
- <https://developers.google.com/speed/docs/insights/PrioritizeVisibleContent>

- Het doel van deze gedachte is om een gebruiker zo snel mogelijk in ieder geval "iets" te laten zien van de website en de algemene wachttijd tot een minimum te beperken.
- Om dit technisch te kunnen realiseren zullen we meer controle moeten uitoefenen over de volgorde waarin alles wordt ingeladen en uitgevoerd.

- In de meeste gevallen zal er in de head van je website altijd een koppeling staan naar één of meerdere stylesheets met `<link>` elementen.
 - Een alternatief is een blok met interne CSS in een `<style>` element.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section class="top-content">...</section>
    <section class="bottom-content">...</section>
    <footer>...</footer>
  </body>
</html>
```

De vouwlijn

- Normaliter bevat onze stylesheet *alle* CSS voor de gehele pagina – dus ook voor de tweede section en de footer die beneden de vouwlijn staan.
- In feite zijn de stijlregels voor deze onderdelen niet per se nodig om alles boven de vouwlijn correct weer te geven.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section class="top-content">...</section>
    <section class="bottom-content">...</section>
    <footer>...</footer>
  </body>
</html>
```

De vouwlijn

- Het is momenteel ook toegestaan om ook blokken CSS (of een volledige stylesheet) buiten de `<head>` in te laden als dat eraan kan bijdragen dat het deel van de pagina *above the fold* sneller ingeladen kan worden.
- <https://developers.google.com/speed/docs/insights/OptimizeCSSDelivery>

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section class="top-content">...</section>
    <section class="bottom-content">...</section>
    <footer>...</footer>
  </body>
</html>
```

De vouwlijn

- Je kunt dan dus de essentiële CSS zo vroeg mogelijk inladen en de minder belangrijke CSS uitstellen tot nadat de HTML code van de body is ingeladen.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      <!-- (belangrijkste) stijlregels voor above the fold... -->
    </style>
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section class="top-content">...</section>
    <section class="bottom-content">...</section>
    <footer>...</footer>
  </body>
</html>
<link rel="stylesheet" href="css/style.css"> <!-- de rest van de CSS -->
```

De vouwlijn

- Op deze manier kan alles boven de vouw dus zo snel mogelijk gepresenteerd worden zonder dat we hoeven te wachten op minder essentiële stijlregels.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      <!-- (belangrijkste) stijlregels voor above the fold... -->
    </style>
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section class="top-content">...</section>
    <section class="bottom-content">...</section>
    <footer>...</footer>
  </body>
</html>
<link rel="stylesheet" href="css/style.css"> <!-- de rest van de CSS -->
```

De vouwlijn

- Ook binnen JavaScript kunnen we maatregelen treffen om ervoor te zorgen dat alles boven de vouw zo snel mogelijk presenteerbaar is.
- Net als HTML en CSS worden regels JavaScript code één voor één uitgevoerd, van boven naar beneden.
 - Een opdracht die je uitvoert op regel 15 binnen een bestand moet wachten totdat het uitvoeren van een opdracht op regel 14 helemaal afgerond is
- Dat is niet anders wanneer je meerdere JavaScript bestanden gebruikt. De browser zal eerst alle opdrachten in het eerste bestand uitvoeren voordat hij überhaupt aan het tweede bestand begint.

- Door de keyword **async** toe te voegen aan een `<script>` element met een verwijzing naar een extern JavaScript-bestand vertel je in feite aan de browser dat hij niet op dit bestand hoeft te wachten.
- Het bestand wordt dan tegelijkertijd op de achtergrond ingeladen terwijl de browser gewoon doorgaat met het inladen van de rest van de content.
 - Lees meer op https://www.w3schools.com/tags/att_script_async.asp.
- Asynchrone JavaScript speelt ook een belangrijke rol binnen **AJAX**.

```
<script src="js/javascript-plugin.js" async></script>
```

- Veel blokken JavaScript code hoeven pas uitgevoerd te worden zodra een gebruiker een bepaalde handeling uitvoert. Dat is dus niet essentieel om de alles van de website *boven de vouw* zo snel mogelijk te laten zien.
 - Dergelijke JavaScript code kan dus prima asynchroon ingeladen worden.
- <https://developers.google.com/speed/docs/insights/BlockingJS>

```
<script src="js/javascript-plugin.js" async></script>
```

- Toch dien je goed op te letten bij asynchrone JavaScript. Het is geen antwoord voor alles.
- Wanneer een bepaalde regel JavaScript code gebruik maakt van JavaScript code uit een ander bestand, mag dat andere bestand niet asynchroon worden ingeladen.
 - Je loopt dan immers het risico dat de benodigde code er dan nog niet is.

```
<script src="js/jquery-3.2.1.min.js" async></script>
```

```
<!-- .click() is een functie van jQuery dus dit werkt niet! -->  
<script>  
    $("button").click(function() {...});
```

- Datzelfde geldt natuurlijk ook voor bibliotheken die op elkaar voortborduren zoals de vele jQuery libraries.

```
<script src="js/jquery-3.2.1.min.js" async></script>
```

```
<!-- zijn gebaseerd op jQuery dus gaan ook niet werken! -->
```

```
<script src="js/jquery-ui.min.js"></script>
```

```
<script src="js/jquery.maphilight.min.js"></script>
```

- Datzelfde geldt natuurlijk ook voor bibliotheken die op elkaar voortborduren zoals de vele jQuery libraries.
- De jQuery uitbreidingen zelf asynchroon laten inladen werkt in dit voorbeeld natuurlijk wél en wordt dan ook ten zeerste aanbevolen.

```
<script src="js/jquery-3.2.1.min.js"></script>
```

```
<!-- dit mag wel! -->
```

```
<script src="js/jquery-ui.min.js" async></script>
```

```
<script src="js/jquery.maphilight.min.js" async></script>
```