

Voorbeeld test CNN gallery

Dit is een voorbeeld eindtest voor dit vak en omvat alle leerstof t.e.m. JS deel 10.

- (strikt genomen deel 09, want deze test doet niks met JSON)

Indienen

Je hebt 3u tijd om deze opdracht af te werken en in te dienen.

Stop je oplossing in een Webstorm project en zip dit. Noem de zip file als volgt

test cnn familienaam voornaam.zip

(waarbij je natuurlijk je eigen namen invult)

~~Upload tenslotte deze zip file bij de opdracht op de leeromgeving.~~

Als je zip file unzippt wordt, moet alles gewoon werken zonder allerlei paden te moeten aanpassen of bestanden te verplaatsen.

Dit is een voorbeeld test, je kunt niet werkelijk iets indienen op de leeromgeving.

CNN foto galerij

De opdracht voor deze test is een webpagina te maken met de hieronder beschreven functionaliteit en uitzicht. Gebruik hierbij dezelfde technieken zoals in de cursus, extra libraries zijn niet toegelaten.

Op de site van cnn.com prijkte een tijdje geleden onderstaande foto galerij. Het auteursrecht voor de gegeven resources bij de opgave liggen dus bij CNN (of anderen), maar we zullen deze opdracht als fair use beschouwen ;)


[Aviation](#) | [Business Traveller](#) | [Destinations](#) | [Features](#) | [Food/Drink](#) | [Hotels](#) | [Partner Hotels](#)

Jordan Trail: A trek through history via ancient villages and wild wadis

By Justin Calderon, CNN
Updated 0130 GMT (0930 HKT) May 18, 2017



Photos: Jordan's jewels

Green start: The Jordan Trail begins from Um Qais in the north and traverses verdant forest and farmland.

1 of 11



(CNN) — Picture the Appalachian Trail in California, or the Camino de Santiago in Spain.

Then draw a route through more than 10,000 years of history, covering Neolithic ruins, Biblical sites, one of the [New 7 Wonders of the World](#), and russet landscapes that wouldn't look out of place on Mars.

More from Travel



And the best airline is...



The in-flight snack so popular it has its own emoji

(deze screenshot zit ook in de gegeven zip file zodat je het beter kunt zien)

Gegeven is een .zip bestand met daarin alle nodige resources : logo, thumbnails, de foto's, de teksten (in content.txt), story prentjes en een javascript file met gegevens over de foto's in de galerij.

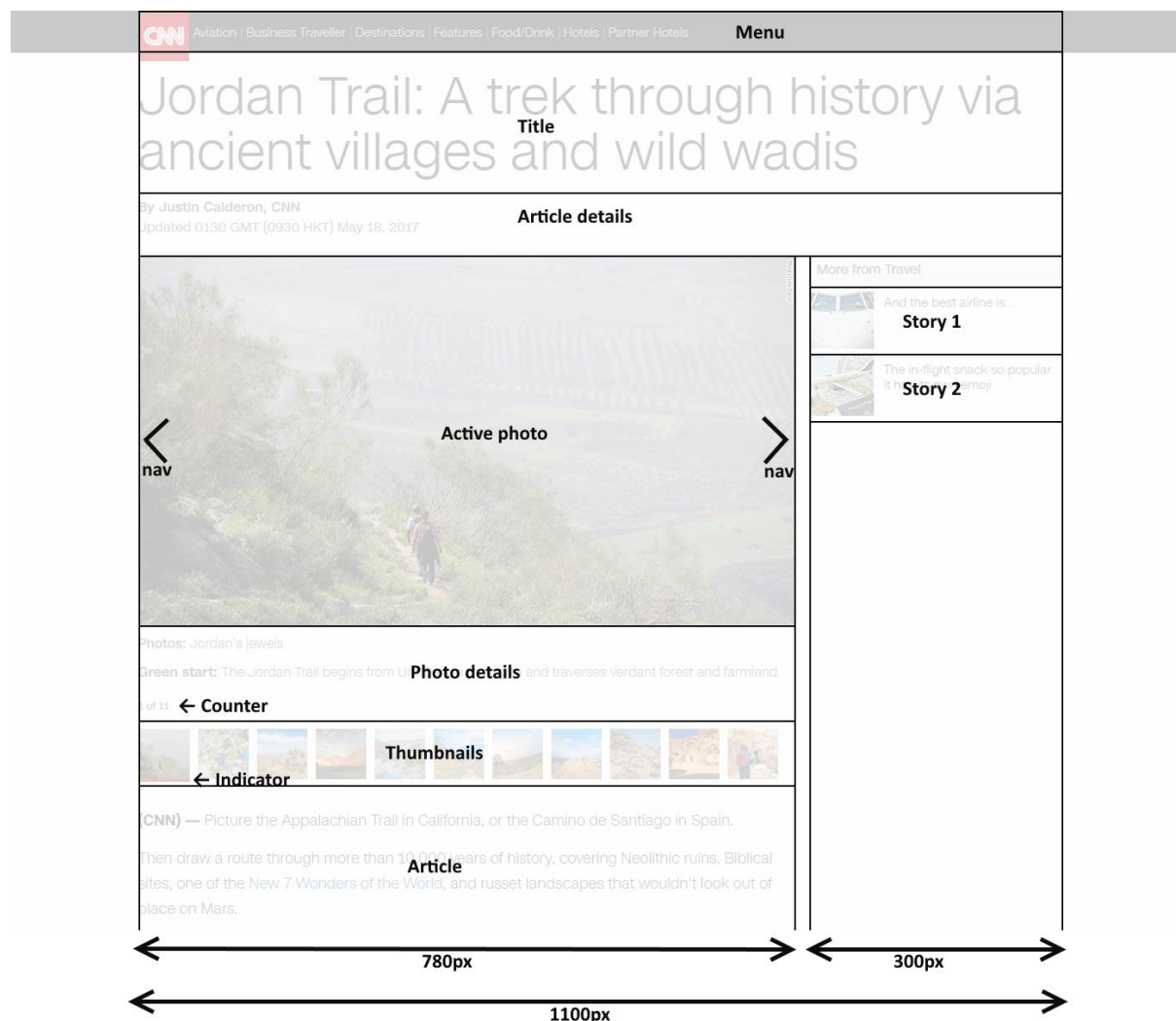
In de gegeven zip file zit ook een screenshot die je kunt raadplegen.

De HTML en CSS van deze webpagina zijn volledig gegeven!

De zip file van de opgave bevat een korte demonstratie video
die toont hoe de pagina reageert op gebruikersinteracties

Overzicht webpagina

De onderdelen van de webpagina zijn als volgt :



De linkerkolom toont de foto galerij en eronder een stukje (vaste) tekst. De inhoud van de galerij (foto's, thumbnail, beschrijving, etc.) wordt dynamisch opgebouwd d.m.v. Javascript op basis van het gegeven bestand '**gallerydata.js**'.

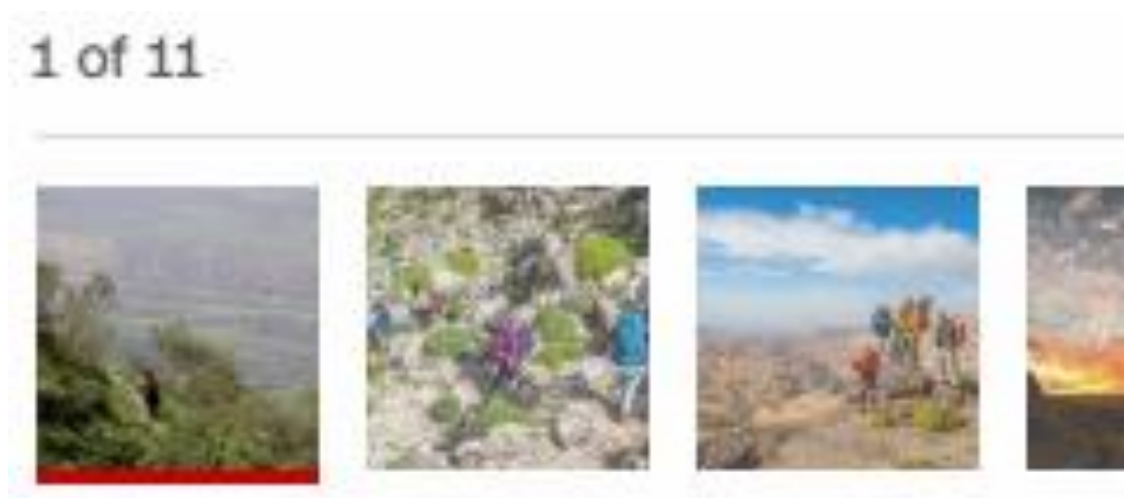
De galerij toont een aantal **thumbnails** (kleinere versie van de foto's) inclusief een teller (**counter**) en **één actieve foto**. De foto's en de thumbnails zijn aparte afbeeldingen uit de /photos en /thumbnails folder.

Merk op dat er links en rechts op de actieve foto, **navigatiepijltjes** staan.

In de CSS file is een styling regel voorzien voor de thumbnails, geef elke afbeelding de class 'thumbnail' om ze correct weer te geven.

Per foto is er ook een copyright vermelding en een korte beschrijving. De **copyright** vermelding en beschrijving (**description**) van de actieve foto staan eronder, op resp. de eerste en tweede regel van het onderdeel "**Photo details**".

Net daaronder staat een kleine teller (**counter**) tekst (bv. "1 of 11") die aangeeft waar we in de galerij aan het kijken zijn. Bijvoorbeeld, als de teller "5 of 11" toont, zitten we blijkbaar op het vijfde prentje van in totaal elf afbeeldingen te kijken.



Onder de thumbnail van de 'actieve foto' staat **een rode horizontale lijn**, als je inzoomt op de screenshot kun je ze beter zien. Indien de teller "3 of 11" toont, moet het rode streepje natuurlijk onder de derde thumbnail staan.

In de CSS file is een CSS-class 'activeThumbnail' gedefinieerd voor deze rode lijn :

```
.activeThumbnail {  
    border-bottom: 3px solid #c00;  
}
```

Onder de thumbnails staat in het 'Article' onderdeel nog een fragment van het eigenlijke artikel (vaste tekst in de HTML, heeft niks met de gevraagde Javascript code te maken).

In de gegeven HTML file staan enkele TODO's om aan te geven waar je iets zult moeten doen met je Javascript code. Daarom hebben enkele elementen al classes of id's gekregen, maar je mag gerust nog classes en/of id's toevoegen als je dit nodig hebt voor je code.

Normaliter zul je in de CSS file geen aanpassingen moeten doen.

Dynamische galerij

De volledige galerij moet dynamisch opgevuld worden en wel enkel (!) op basis van de gegevens in het bestand '**gallerydata.js**'.

Er is dus maar 1 html pagina! Het is NIET de bedoeling dat je een oplossing maakt met X verschillende pagina's die elk een andere foto tonen. Anders gezegd, als we de inhoud van gallerydata.js zouden veranderen (en herladen), dan moet de pagina een totaal andere galerij tonen.

Het bestand 'gallerydata.js' definieert **een variabele 'galleryData'** die naar een array met objecten wijst. Plaats je eigen Javascript code in een apart bestand in de /scripts folder, je mag je eigen code niet toevoegen in gallerydata.js!

- Je moet een <script> element toevoegen aan de HTML file voor 'gallerydata.js' en natuurlijk ook een <script> element voor je eigen Javascript file.
- Zorg ervoor dat de <script> elementen in de juiste volgorde staan (gallerydata.js komt VOOR je eigen Javascript file).

Voor elke foto in de galerij is er in **het galleryData array** een object met de volgende data:

- urlThumb
 - dit is de url van de kleine versie van de afbeelding
- urlFull
 - dit is de url van de grote versie van de afbeelding
- copyright
 - dit is een tekst met info en **bevat geen html tags**
- description
 - dit is een tekst met een beschrijving en **bevat soms html tags!**

Je mag de waarden van bovenstaande properties niet wijzigen, maar als je dit nodig acht, mag je wel nieuwe toevoegen.

Nogmaals : je eigen Javascript code stop je in een apart bestand, niet in gallerydata.js!!!

De urls verwijzen naar de afbeelding van de foto en de bijbehorende thumbnail. De afbeeldingen zijn nu nog mooi genummerd van 1 t.e.m. 11 maar je mag er niet van uitgaan dat dit altijd zo netjes zal zijn.

Het max. aantal foto's in de gallery is 11, maar zorg ervoor dat je oplossing ook correct werkt indien het er minder zijn. Dus ook het "of 11" in de counter moet correct blijven.

Kortom : als we de gegevens in gallerydata.js aanpassen of objecten toevoegen/verwijderen, dan moet je programma nog steeds correct werken en de juiste info tonen.

Galerij interacties

De gebruiker kan op een aantal manieren met de galerij interageren :

- Bij een **klik op de navigatiepijltjes** (i.e. < links en > rechts bovenop de actiefoto), wordt naar de vorige/volgende foto overgegaan
 - indien klik op "vorige" bij eerste foto : naar laatste foto springen
 - indien klik op "volgende" bij laatste foto : naar eerste springen
- **Klikken op een thumbnail** gaat meteen naar die positie in de galerij en toont de corresponderende foto als actieve foto.

Denk eraan, telkens de actieve foto wijzigt, moeten er een aantal zaken mee veranderen in de pagina :

- de afbeelding van de **actieve foto** (natuurlijk)
- de **counter** (teller)
- de **copyright** melding
- de **description** tekst
- de **rode lijn** onder de juiste thumbnail

De zip file van de opgave bevat een korte demonstratie video
die toont hoe de pagina reageert op gebruikersinteracties

Veel succes!