

huisstijl: bedrijf

visie: welke visie hebben we voor Dali en buro CITE in zijn geheel?

Doelgroep: voor welke doelgroep is Dali bedoeld en willen we dit uitbreiden?

Marktgebied: welk markt gebied heeft Dali nu

- bedrijf: visie, doelgroep, advertentie mogelijkheden, marktgebied, partners, bginfo
- sjablonen: facturen/offertes, documenten, powerpoint, visite kaartjes, brochures
- logo: verschillende varianten, kleur of zwart wit, logo gebruik, bginfo
- pay-off: slogans/pay-off die gebruikt word, verplichting van slogan bij het logo?, slogan style/lettertype, bginfo
- kleuren: kleuren op verschillende objecten, kleur codes en complimenterende kleuren, bginfo
- typografie: welke lettertype (meerdere),
- fotografie: welke foto stijl, welke fotograaf, bginfo
- teksten: op welk platform wordt u gebruikt en op welke jij, wat voor spreek toon wordt er gehanteerd, beredenering spreektoon
- social media: profielfoto's, geposte afbeeldingen, headers

Rekening houden met:

versie beheer,

toegang tot bestand,

huisstijl controle

program

typografie

op de site

- welk lettertype worden gebruikt. Het beste is als er meerdere lettertypes zijn, dit is om rekening te houden met toestellen waar een bepaalde lettertype niet bestaat.
- Welke letter grote wordt er gehanteerd. Volgens de WCAG is het minimum 16px (moet nog nagekeken worden)
- Underline voor een link?
- Kleuren voor een link?
- Hoofdletter gebruik (bij titels en of een woord namen)

- Hoeveelheid tekst dat tegelijk in beeld is. Wegens toegankelijkheid voor mensen met autisme en of dyslexie is het aangeraden om grote hoeveelheden tekst te vermeiden. Dit kan voor stress en bewegende teksten zorgen.

In develop omgeving

- Aantal spaties die een tab vertegenwoordigen
- Netjes uitlijnen van je nested elementen
- Altijd eindigen met ; als dit kan (ook als het niet hoeft)
- Hoe worden class attribute genoteerd (mijn voorkeur is "dali-button-red")
- Hoe worden id attribute genoteerd (mijn voorkeur is "daliButtonRed")
- Waar wordt class attribute voor gebruikt (mijn voorkeur is voor css)
- Waar wordt id voor gebruikt (mijn voorkeur is voor javascript)
- Gebruiken we een lege regel tussen verschillende stukken code? (mijn voorkeur is wel doen)
- Worden kringel haken {} op de zelfde regel geopend of op een aparte regel.

Voorbeeld van op zelfde regel:

```
Function RedDaliButton () {  
  
}
```

- Worden const namen in hoofdletters getypt? (mijn voorkeur is wel doen)
- Worden var en let namen in camelcase getypt?
- Duidelijke namen voor variabelen, voorbeeld: daliButtonRed;

- Gebruik van semantic tags is verplicht

- Uitlijnen van html elementen als er teveel characters op 1 regel zitten?
- Worden docblocks per functie toe gevoegd of per stuk code in de functie

- Variabelen worden op de hoogste niveau aangeropen tenzij dit niet mogelijk is.

Voorbeeld:

```
function popup () {  
  const state = useState();  
  let hideRedDaliButton = ...  
}
```

- Op welke manier maken we bestand namen

- SCSS zo veel mogelijk nesten

- Geen elementen stylen tenzij dit echt voor iedere pagina moet gelden of als deze goed is genest
- @media voor verschillende scherm grote wordt onder in aangeroepen
- Schaalbaarheid van de website wordt via het mobile first principe gemaakt
- Moeten react geleverde functies/elementen uitgelegd worden in een docblock of moet er verwacht worden dat iedereen die kent?

Logo

Achtergrond van het DALI logo

(dit onderdeel zit erin zodat in de toekomst er altijd bekend is hoe het logo is ontstaan en waar het voor staat. Dit kan handig zijn als het ooit aangepast moet worden.)

Varianten

Er zijn verschillende varianten van het DALI logo zoals:

- Voor een banner/header
- Standaard klein formaat
- Klein breed formaat (voor een footer)
- Standaard uitgebreid formaat
- Standaarduitgebreid met slogan formaat
- Icoon formaat
- Zwart wit voor ieder formaat

Logo gebruik

1. Logo mag niet schuin geplaatst worden.
2. Logo mag niet half bedekt of maar half zichtbaar zijn.
3. Logo niet aangepast worden maar uit de bovenstaande formaten komen. (tenzij anders besproken)
4. Het logo moet goed zichtbaar zijn op de achtergrond

Kleuren

De kleuren die binnen DALI gehanteerd worden staan hieronder uitgelicht. Deze kleuren zijn gekozen met precisie, zodat deze voldoen aan de WCAG 2.1 richtlijnen.

- Het aanpassen van deze kleuren mag niet zomaar gedaan worden omdat dit gevolgen kan hebben op de toegankelijkheid van heel DALI.

- Bij noodzaak om de kleuren aan te passen dient eerst een onderzoek gedaan te worden op de nieuwe kleuren en of ze geschikt zijn volgens de WCAG richtlijnen.
- Aanpassingen worden gedaan met goedkeuring van meerdere ICT leden die het onderzoek moeten hebben goedgekeurd.
- De nieuwe kleur dient te voldoen aan de volgende criteria:
 - o De kleur dient een 4.5:1 of hoger te score op de kleur contrast op een van de zwart of wit kleuren (dit geldt niet voor grijs tinten omdat dit wat moeilijker valt, maar probeer wel boven de 3.0:1 te zitten)
 - o Bij het aanpassen van de opacity naar .3 moet een felle kleur een 4.5:1 score hebben met een witte tekst. (dit is vooral voor de dashboard/home page zodat de tekst op een blok zichtbaar blijft tijdens focus)

De volgende kleuren zijn goedgekeurd en kunnen gebruikt worden binnen DALI:

Tegen kleur	Kleur code
#eee	#222
#eee	#4c4c4c
#eee	#6b6b6b
#222	#898989
#222	#eee
#eee	#bd3c5b
#eee	#07786e
#eee	#017c21
#eee	#0069ce
#eee	#a700f4
#eee	#8e6333

Stijl

Welke stijl wordt er gebruikt om DALI te bouwen. Gaat dali voor een moderne look of meer een soort glassmorphism misschien wel een mannelijke look met hout bruin en metallic zwart. Het bepalen welke stijl je website is zorgt ervoor dat bij het maken van aanpassingen je niet opnieuw hoeft te verzinnen hoe je wilt dat het eruit ziet.

Zeker met meerdere developers kan het helpen om een web design mood board te maken. Dit is een board met een aantal kleuren en voorbeelden van elementen die ervoor zorgen dat het idee overgebracht wordt van de ene developer naar de ander.

[zoek uit welke stijl dali uitstraalt]

- teksten: op welk platform wordt u gebruikt en op welke jij, wat voor spreek toon wordt er gehanteerd, beredenering spreektoon

Rekening houden met:

versie beheer,

toegang tot bestand,

huisstijl controle

Vanaf hier naar beneden niet op letten. Dit gaat later weg!!!!

New

Bootstrap

Bootstrap word momenteel door een groot deel van de dali applicatie gebruikt. Wegens de toegankelijkheid moet het gebruik hiervan aanzienlijk omlaag door het beperken van de componenten die ingeladen worden. Bootstrap maakt voor veel van hun components gebruik van div's dit zorgt voor situaties waar er 6 div's genest zijn. Dit is niet goed voor de toegankelijkheid, betekend dat deze componenten niet meer gebruikt dienen te worden. Natuurlijk zijn dit niet alle bootstrap componenten, daarom heb ik hier een lijst met componenten die zonder zorgen gebruikt kunnen worden:

- Button (in onderzoek)
- Badge (in onderzoek)
- Deze lijst is nog niet compleet

Afkeur reden per component

- Dropdown
 - Er worden een aantal geneste div's gebruikt. Dit zou in deze situatie nog een section kunnen zijn, maar dit is niet aantepassen.
 - Het toevoegen van eventlisteners is lastig in verband met de eerste load waarbij de dropdown word gegenereerd. Kort gezegd de dropdown bestaat nog niet op het moment van eventlisteners toevoegen.
 - Note: deze problemen kunnen mogelijk omzeild worden met de juiste aanpak. Het kan ook zijn dat deze problemen niet altijd weergegeven worden in andere soorten dropdowns binnen bootstrap, maar deze inconsistentie is wat ervoor heeft gezorgd dat het is afgekeurd.

- Card
 - Er worden een hoop geneste div's gebruikt. Dit zijn onnodig veel div's die zelfs gebruikt worden voor de tekst

Samenvatting:

Beperk het gebruik van bootstrap tot de boven genoemde componenten. Dit heeft te maken met dat de andere componenten te veel div's genereren wat problemen opbrengt voor de toegankelijkheid.

Kleuren

De oude kleuren van dali waren niet conform de WCAG 2.1 richtlijnen. Dit heeft geleid tot het hernieuwen van de kleuren zodat deze wel voldoen. Met deze aanpassingen komen een set met regels voor het gebruiken van kleuren.

Regels

- Nieuwe en of kleine accent kleuren mogen niet zomaar bedacht worden. Deze dienen of al te bestaan en uit dit bestand te komen of onderzocht te worden doormiddel van kleur contrast checkers en intern besproken te worden.
- Elementen waar een kleur als achtergrond kleur word gebruikt dienen de juiste tekst kleur te gebruiken tenzei intern anders besproken is en de vervangende kleur moet minimaal een 4.5:1 en maximaal 17:1 contrast score hebben.
- Deze lijst is hoogst waarschijnlijk nog niet compleet

De nieuwe kleuren

Hier onder bevind zich een tabel met alle hex codes van de nieuwe dali kleuren en de kleur tekst die erop moet staan om het goed leesbaar te maken. Ook staan de oude dali kleuren erbij die vervangen worden.

Oude Dali kleuren	Nieuwe dali kleuren	Tegen kleur
#000000	#222	#eee
#5a5a5a	#4c4c4c	#eee
#aaaaaa	#6b6b6b	#eee
#cccccc	#898989	#222
#e6e6e6	N.V.T.	
#ffffff	#eee	#222
#c84664	#bd3c5b	#eee
#177f75	#07786e	#eee
#4caf50	#017c21	#eee
#58ce11	N.V.T.	
#0094ff	#0069ce	#eee
#b100fd	#a700f4	#eee

#f1b330	N.V.T.	
N.V.T.	#8e6333	#eee

Bij sommige kleuren is het voor moeilijk om een verschil te merken maar het verschil is er zeker. Ook is het op te merken dat er een aantal kleuren weg zijn en een erbij is gekomen. Dit komt omdat de kleuren geel (#f1b330), lime groen (#58ce11) en licht grijs (#e6e6e6) geen goede kleur contrast hebben met hun medekleuren. Wat hiermee bedoelt wordt is dat e kleur te veel op een andere kleur lijkt of (in het geval van geel) heel lelijk wordt. dit klinkt als een mening, maar dat is het niet omdat geel een hele donkere geel moet worden om genoeg kleur contrast te tonen is het mooier en logischer om geel te vervangen met bruin (#8e6333).

Bij de grijs tinten zou gezegt kunnen worden dat er ook te weinig contrast is en dat klopt. Om 3 grijs tinten te kunnen gebruiken is er een beetje relaxter gekeken naar de grijs tinten. Het beste is om de grijs tinten zo min mogelijk samen te gebruiken en mocht het toch nodig zijn gebruik dan of juist geen of juist alleen de kleuren #4c4c4c en #898989.

code

- imports always in the same order
- camelCase for things like variables, arrays, objects, functions, etc.
- camelCase for folder and non-component file names
- PascalCase for component file names
- use typescript barrels whenever possible
- const names should be all UPPERCASE
- always provide documentation above a non self explanatory function
- Separate function from the JSX if it takes more than 1 line
- seperate functionaliy as much as possible
- seperate logic an rendering
- avoid inline CSS
- try not to use class for javascript
- eindigen we alle js regels met ; of alleen wanneer nodig
- provide explanations about packages and new developments used like SCSS and - - Hooks
- import everything in 1 file instead of nesting imports
- keep everything indented
- witch indent length do we choose
- use "-" for class names (like-this-for-example)

opmerkingen react:

- een class word alleen gezien als class als het begint met een hoofdletter
- don't use `React.createElement` unless you'r initializing the app
- use PascalCase when naming components and filenames
- order inside a class extended `React.component`:
 - constructor
 - static methods
 - `getChildContext`
 - `componentWillMount`
 - `componentDidMount`
 - `componentWillReceiveProps`
 - `shouldComponentUpdate`
 - `componentDidUpdate`
 - `componentWillUnmount`
 - click- and event-handlers
 - get methods
 - render methods
 - render
- use `""` for JSX attributes but `"` for all other js
- camelCase for props
- multi line components need to have closing tags on their own line
- stateless components use function syntax
- Setting `propTypes` declarations is mandatory
 - Group them into required/none-required
 - Alphabetically sort each group
 - Separate them by a new line
- Prefix all none React methods within a component with an underscore.
- In the exception that you do not want to place a component wide variables on the state, you have to prefix it with an underscore.
- Name methods using `'_handle'` + triggering event, e.g. `_handleClick`
- Bind handler using the ES6 arrow syntax, so inside the callback it has always the right context
- Using “container” components for loading data from Stores

- toegankelijkheid

- provide alternative tekst for images (speak assistant)
- check if colors are compatible and can be next to eachother(color blind)
- make invisible(focus) skip buttons (keyboard only)
- provide written captions for videos (deaf people)
-

Back- End

Leesbaarheid

- Duidelijke Functie en/of variabele namen.
- Beschrijf kort maar krachtig wat een functie doet.
- Start variabele zonder hoofdletter.
- Niet te lange namen.
- Structuur in je code / gesorteerd.
- Niet meerdere keren dezelfde functies gebruiken of herschrijven.
- Goed uitlijnen van code en functies.