Galen Framework

Testons nos sites responsives !

Qui suis-je ?

Raphaël MORVAN

Lead developer Drupal chez Capgemini

Tombé dans Drupal depuis 2013

@RaphMo

Retour des Drupal Dev Days 2017

- Alejandro Gómez Morón (@agomezmoron)
- Óscar Castaño Calle

De quoi allons-nous parler ?

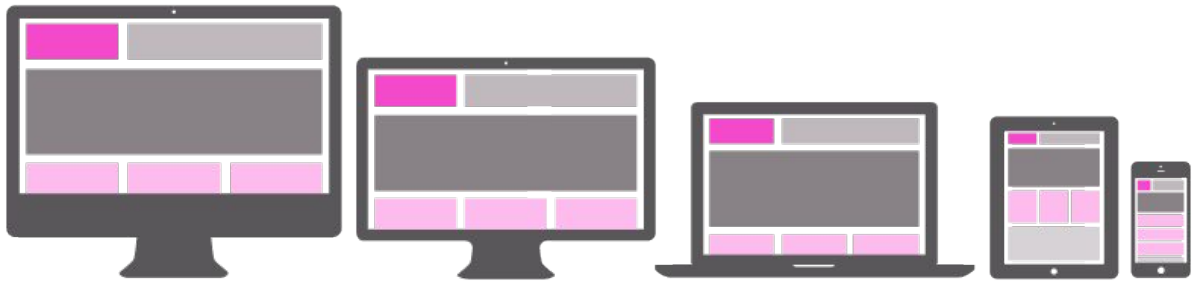
Au programme de cette session...

1. Notions préalables
2. Qu’est-ce que Galen Framework ?
3. Comment le mettre en place ?
4. Déclaration de nos tests
5. Démonstration

Révisons...

Notions de base : Responsive

- Objectif : rendre la navigation sur site internet la plus confortable possible selon son support (TV, ordinateur, tablette, téléphone mobile…)
- Avant : développement d'une version optimisée pour mobile
- Aujourd'hui : mise en place du responsive dès la phase de conception
- Utilisation de "grilles"
- Flexbox



Notions de base : CSS

- Permet de mettre en forme nos pages
- Se base sur le DOM de notre code HTML

Notions de base : Javascript

- Permet de dynamiser nos pages
- Se base sur les éléments du DOM

Pourquoi mettre en place des tests automatisés ?

Bah oui, pourquoi ?

Éviter les régressions

Maintenance du site dans le temps

Gagner du temps

... et donc de l'argent

Installation du framework

Framework basé sur Selenium

Prérequis : Java 1.8+ & npm

Installation : sudo npm install -g galenframework-cli

Possibilité de l'installer à la main (cf. documentation officielle)

Commande : galen

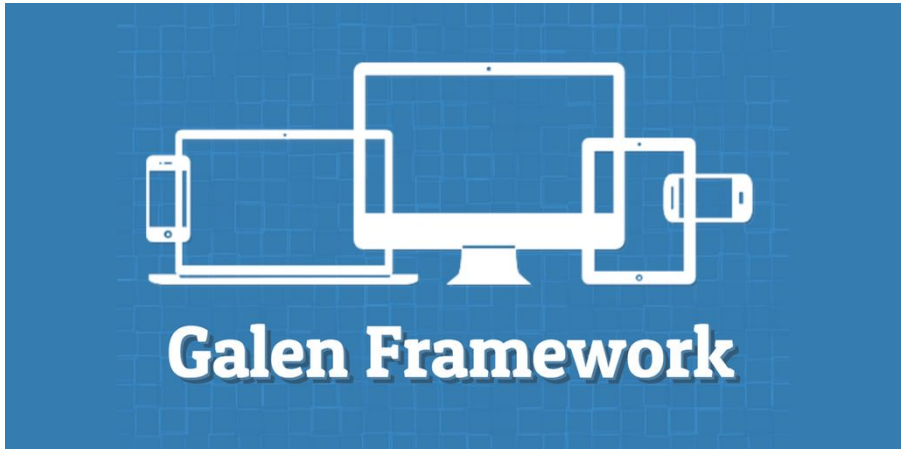
Comment lancer un test ?

Lancer un simple test :

```
galen check
```

Lancer une suite de tests :

```
galen test
```



Définition des objets

- Permettent de déclarer les différents éléments de notre DOM
- Sélection possible par :
  - id
  - css
  - xpath
- Possibilité de déclarer des objets multiples
- On peut même grouper nos objets !

Définition des objets : exemple

```
<body>
    <div id='search-bar'>
        <input type='text' name='search' value=''/>
        <a href='#' class='search-button'>Search</a>
    </div>
</body>
```

```
@objects
    search_panel            id      search-bar
    search_panel_input      xpath
//div[@id='search-bar']/input[@type='text']
    search_panel_button     css     #search-bar a
```

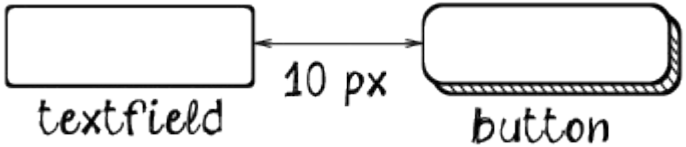
Et on en fait quoi de nos objets ?

Mais dis moi, que peut-on tester ...

- Une valeur attendue pour un attribut concernant notre objet (height, width...) :
  - width 30px
  - width > 30px
  - width < 30px
  - width >= 30px
  - width <= 30px
  - width ~ 30px

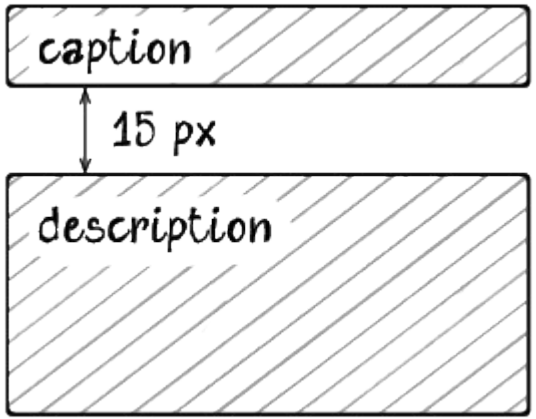
- La présence (ou non) de nos éléments

Mais aussi...



textfield:
    near button 10px left

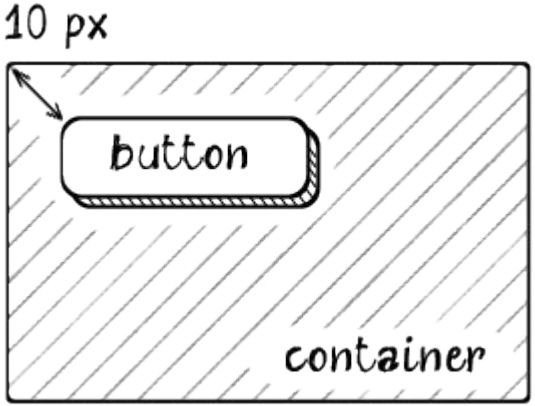
Et encore...



caption:
    above description 10 to 20 px

description:
    below caption 10 to 20 px

Puis tant qu'à y être...



button:
    inside container 10 px top left

Et toujours plus !

- Alignement des éléments
- Valeur des textes
- Vérification des propriétés CSS
- Alignement centré
- Navigateurs supportés : Firefox (par défaut), Chrome, IE, PhantomJS, Edge
- Rapports HTML (incluants les erreurs)
- Comparaison d’images
- Possibilité de lancer les tests sur Browserstack

Une démo ?

Sources disponibles sur Github

Lien du dépôt : https://github.com/RaphMo/galen-dclannion

Des questions ?