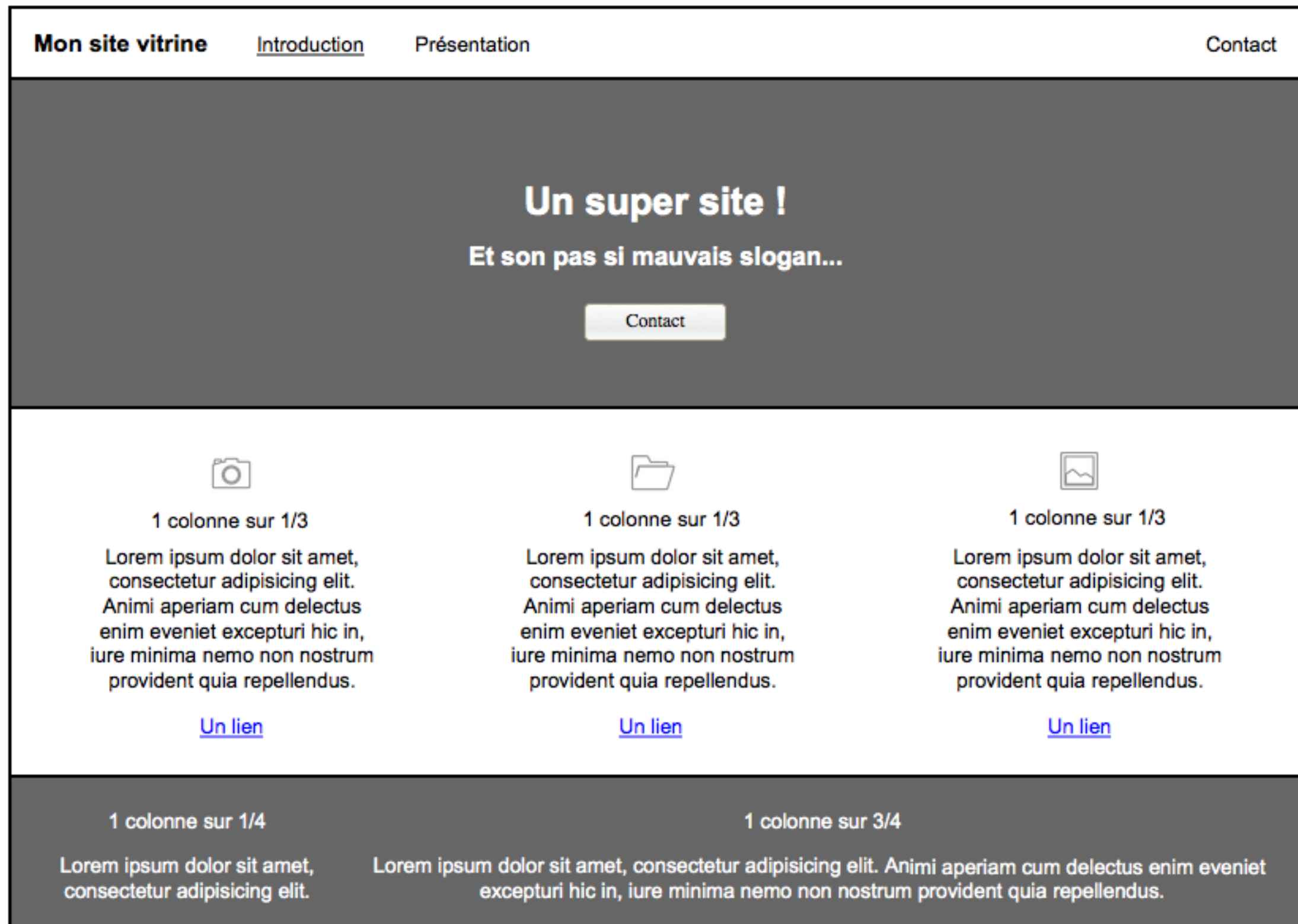


Formation Industrialisation Front-End

Pratique

Maquette

Maquette



Maquette

Mon site vitrine [Introduction](#) [Présentation](#) [Contact](#)



1 colonne sur 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Animi aperiam cum delectus enim eveniet excepturi hic in, iure minima nemo non nostrum provident quia repellendus.

[Un lien](#)

Nous contacter

Votre email

Votre message

Blah blah blah

Envoyer



1 colonne sur 1/3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Animi aperiam cum delectus enim eveniet excepturi hic in, iure minima nemo non nostrum provident quia repellendus.

[Un lien](#)

1 colonne sur 1/4

1 colonne sur 3/4

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Animi aperiam cum delectus enim eveniet excepturi hic in, iure minima nemo non nostrum provident quia repellendus.

4

Frameworks HTML/CSS/JS

Frameworks HTML/CSS/JS - Exercices

1. Avec un framework au choix, créer le site correspondant aux maquettes précédente. Le clic du bouton et du menu contact doit faire apparaître le formulaire de contact en popup (à faire si possible en JavaScript). Le site doit être responsive.
2. Ajouter suffisamment de contenu pour que le texte fasse plus d'une page.
Au scroll, faire changer l'élément souligné du menu pour lui faire correspondre la partie visible (à faire si possible en JavaScript)

Préprocesseurs CSS

Préprocesseurs CSS - Exercices

1. En utilisant le langage de Préprocesseur de votre Framework (Less/Sass), modifier la variable qui donne la couleur du lien et recompiler.
2. Utiliser cette variable pour modifier la couleur de fond des sections « grises »
3. Modifier le nombre de colonnes de votre framework et recompiler.

Modules JavaScript

Modules JavaScript - Exercices

1. Créer un module RequireJS (donc 1 fichier) pour :
 1. Le script qui fait apparaitre la popup contact
 2. Le script qui change le soulignement du menu
2. Ajouter un module UMD pour la validation de formulaire, utiliser une fonction constructeur (nommée Validator) si possible (les 2 champs sont obligatoires)
Tips : `function Validator(form, fields) { }`
`Validator.prototype.validate = function() {return true}`

Subsets JavaScript

Subsets JavaScript - Exercices

1. Réécrire la validation de formulaire dans le subset JavaScript de votre choix, exemple :
 1. Utiliser une classe au lieu d'une fonction constructeur
 2. Utiliser des types si possible
 3. Utiliser des Arrow-Function et du sucre syntaxique lorsque cela est possible

Gestion de dépendances

Gestion de dépendances - Exercices

1. Importez jQuery et votre framework via Bower ou npm (nous souhaiterions la version 1 de jQuery la plus récente, pas la 2)
2. Faire de votre fonction constructeur Validator un module npm public

Git

Git - Exercices

1. Créez un repository git
2. Créez un projet sur Github, modifier le user et email git en ligne de commande et synchroniser les 2 repositories
3. Ajoutez un fichier Readme.md et faire un commit sur la branche local, puis le publier sur Github
4. Remplir une demande d'amélioration dans le issue tracker de github. Nous souhaiterions changer la couleur principale du site (lien + fond). Créer une branche, faite la modification puis un merge vers la branche master. Publier sur Github



Grunt

Grunt - Exercices

1. Créer la tâche grunt qui va transformer tous vos fichiers LESS/SASS en CSS (dossier less ou sass vers dossier css)
2. Idem pour vos sources TypeScript/CoffeeScript...
3. En utilisant usemin, fusionner et minifier les sources

Tests Automatisés

Tests automatisés - Exercices

1. En utilisant Karma + le framework de tests de votre choix, écrire un tests pour votre module de validation
 1. 2 scénarios :
Si les 2 champs sont vides, la méthode valide doit retourner false
 2. Si les 2 champs sont remplis, la méthode valide doit retourner true

Scaffolding

Scaffolding - Exercices

1. Avec Yeoman, faire de votre projet un template pour de futurs projets.