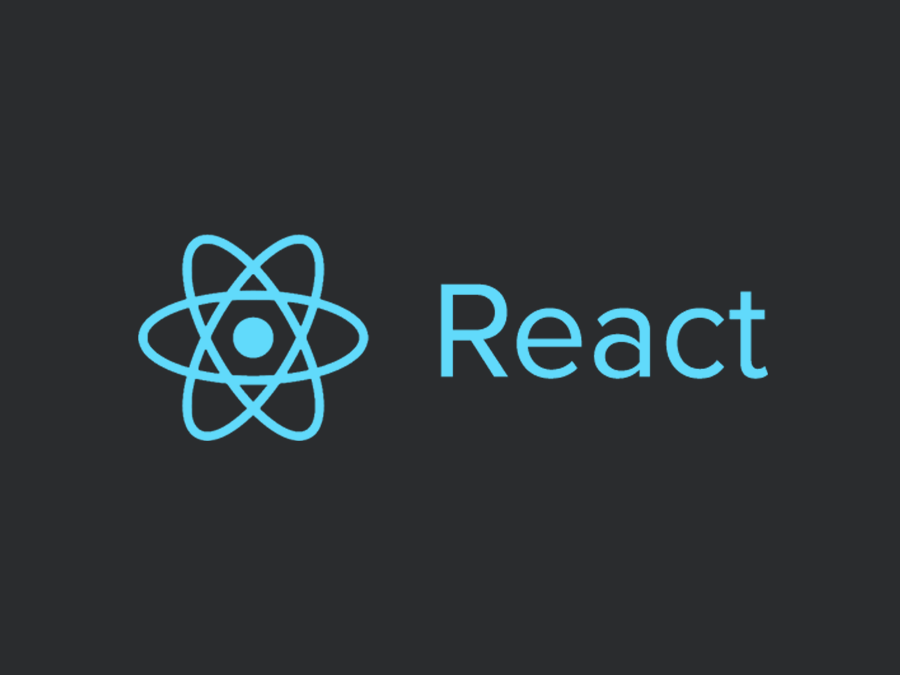
**Atelier ReactJS**



*Tous les exercices sont indépendants et sont à réaliser avec le générateur d’application officiel* ***React Create App***



### 

### *Pour ReactJS (et ses 11 parties), acceptes le changement de tout ce qu’on t’a appris, et que l’on t’as mentis pendant des années en te disant de bien ranger tes fichiers CSS/JS/HTML dans des nouveaux dossiers….*

**xExercice 0**

x Créer une application ReactJS avec un composant Display initialisé dans son état(state) votre nom, votre prénom, votre age, votre ville, votre adresse, vos coordonnées GPS(Objet avec longitude et latitude) et une liste de vos chanteurs favoris (tableau), une url d’image de votre acteur favoris, une url de site préféré

x Le nom et prénom seront concaténés et afficher dans un h3, La liste sera affichée dans des listes à puces. L’image sera affichée et l’url sera cliquable.

X Créer un composant Age composé de 2 boutons permettant de rajeunir ou de vieillir l’âge.

*Bonus: A partir de GMAP, Créer un composant Map permettant de créer une une carte avec le marker qui positionne le centre de la carte et le point sur les coordonées GPS*

Si tu le veux, tu peux aussi installer et importer un composant externe:

<https://www.npmjs.com/package/react-gmaps>

ou intégrer la Map avec la Doc Google

<https://developers.google.com/maps/documentation/javascript/?hl=fr>

**xExercice 0.1**

xCréer une application React qui permet d’avoir 2 composants:

x1 composant Display qui affichera le texte saisi

x1 composant Saisi qui disposera d’un textarea et d’un label afin de saisir du texte

*Bonus 1: Intégrer le framework Bootstrap à l’application*

*xBonus 2: Si le contenu saisi fait moins de 3 lettres, ne pas afficher le message saisie mais le message d’erreur “Votre message est trop court”*

*xBonus 3: Créer un événement utilisateur qui permet (dans le composant Saisi) que lorsque l’on quitte la zone de saisie, le texte entré soit en minuscule et que l’on supprime les blancs*

*xBonus 4: Créer un composant CountWord qui permet d’afficher le nombre de mot*

**xExercice 0.2:**

xReproduire l’application suivante (sans la CSS) en React:

xTu aura besoin de:

* L’objet Date
* La fonction setInterval()

<http://react-clock-basic.drminnaar.me/>

*Bonus: Importe la librairie Moment et utilise Moment plutôt que Date puis formate la date tel que: “Le 16 Mars 2017 à 20:30 51s”*

*Bonus 2: Toutes les 130 secondes, générer une notification affichant “Sexy ReactJS <3” avec le composant react-toastify ci-dessous (installer et l’importer puis l’utiliser)*

[*https://github.com/fkhadra/react-toastify*](https://github.com/fkhadra/react-toastify)

**Exercice 0.3:**

xReproduire l’application suivante avec Bootstrapp en React:

<http://react-clicker.drminnaar.me/>

*Bonus: Si le compteur atteint la valeur de 20 , afficher un message alert-success “Le compteur va exploser”*

*Bonus 2: Imaginons que notre compteur soit la températur en celius,*

*Si le compteur est negatif, afficher un icon de froid, si le compteur est à plus de 30 afficher un icon de chaleur etc…*

[*http://erikflowers.github.io/weather-icons/*](http://erikflowers.github.io/weather-icons/)

*Si tu es à l’aise avec les composants React, tu peux aussi utiliser* [*https://www.npmjs.com/package/react-weathericons*](https://www.npmjs.com/package/react-weathericons)

*Bonus 3: Créer un bouton “Auto” qui augmente le compteur (température) toutes les 3 secondes.*

**Exercice 0.4:**

X A partir d’une liste de 15 films et leurs notes au choix, reproduire l’application suivante  en React:

<http://react-movie-cards.drminnaar.me/>

*Bonus: Créer un input range slider afin de limiter l’affichage de films*

*Bonus 2: Créer un filtre pour limiter par note l’affichage des films*

**Exercice 0.5**

xReproduire le mini-jeu MasterMinds en React:

[**http://react-masterminds.drminnaar.me/**](http://react-masterminds.drminnaar.me/)

**Exercice 0.6:**

xReproduire la mini Application Items en ReactJS

<https://codepen.io/makenosound/pen/rstvx>

x Bonus: Créer un bouton afin de réinitialiser le jeu et permettre de rejouer.

**Exercice 0.7:**

A partir de l’API de Pokemen(disponible ici: <https://pokeapi.co/docsv2/>) puis du Module Axios, lister les pokemon et faire un moteur de recherche sur le nom des pokemon

<https://alik0211.ru/pokedex/>

**Exercice 1.0:**

Créer un formulaire de Signup (Enregistrement utilisateur) avec un bouton submit et ayant pour champs:

* Sexe(boutons radios) Au moin 1 sexe coché
* Username (minimum 6 caractères alphas numériques)
* Age (de 18 à 99 ans)
* Téléphone
* Password (minimum 5 caractères alphas et 1 caractère spécial)
* Hobbies (case à cocher) Au moins 2 Hobbies coché
* Sports (case à cocher) Au moins 1 Sport coché

1. Valider tous les champs avec React
2. Valider le formulaire à l’aide du Composant React-Validation

<https://www.npmjs.com/package/react-validation>

Bonus: Si tous les champs sont bons, cacher le fomulaire et afficher une notification avec React Toastify <https://fkhadra.github.io/react-toastify/>

Bonus: Ajouter le champs Repeat Password et contrôler à la saisi que le Password et  Repeat Password soit identiques

**Exercice 1.1:**

Créer et Afficher une liste de 10 commentaires avec pour champs (id,contenu,username de l’auteur, note,date de création)

Permettre de pouvoir créer un commentaires dans la liste

Permettre de pouvoir supprimer un commentaire dans la liste

Permettre de pouvoir ajouter en favoris un ou plusieurs commentaires dans la liste

**Exercice 1.2:**

Dans une liste en Grille Bootstrap, permettre d’afficher des photos avec leurs légendes(textes)

Permettre de pouvoir créer/supprimer un photo dans la liste

Bonus: Ajouter du style au survol des images et leurs légendes:

<https://www.hongkiat.com/blog/css-libraries-image-hover-effect/>

Bonus 2: Pouvoir ajouter ou supprimer des photos du Panier (tableau de photos) puis afficher le Panier

**Exercice 1.3:**

Créer le jeux TicTacpoe en ReactJS en suivant ce tutoriel officiel de la documentation:

<https://reactjs.org/tutorial/tutorial.html>

**Exercice 2.0:**

Créer une composant Signup qui permet de s’inscrire avec son nom, son email et son mot de passe (avec le repeat password validé)

Puis une fois que son inscription est valide, rediriger l’utilisateur vers un composant Welcome

**Exercice 2.1:**

Créer une composant Home dans App ayant un menu vers les sous-composants:

* Posts (Affiche les posts depuis une API <https://jsonplaceholder.typicode.com/posts>)
* Comments (Affiche les commentaires depuis une API <https://jsonplaceholder.typicode.com/comments>)
* Album (Affiche les albums depuis une API <https://jsonplaceholder.typicode.com/albums>)
* Photo (Affiche les photos depuis https://jsonplaceholder.typicode.com/photos)

Créer un composant Loader composé d’un loader.gif qui s’affichera le temps de chargement de la liste des différentes API:

<https://loading.io/spinners/typing/lg.-text-entering-comment-loader.gif>

**Exercice 2.2:**

Créer une Liste de personnages de BD (les champs sont au choix).

Afficher cette liste derrière une route /liste.

Afficher dans une nouvelle route un formulaire permettant de créer un nouveau personnage.

Une fois que le formulaire est valide, rediriger vers la liste de personnages.

**Exercice 2.3:**

Créer une Route privée (/private) qui soit accessible si le compteur de MagicButton soit supérieur à 20.

Plus tard, nous pourrions imaginer que cette route soit accessible quand nous serions authentifier.

Vous pouvez vous inspirer de la documentation: https://reacttraining.com/react-router/web/example/auth-workflow

**Exercice 2.99999:**

Nous allons créer un formulaire Signup en 4 Etapes.

**Etape 1:**

Demande de Nom, Email, Mot de Passe, Repeat password.

**Etape 2:**

Demande de l’adresse, ville, cp, longitude, latitude

**Etape 3:**

Une longue biographie d’au moin 10 mots avec un compteur de mot

**Etape 4:**

Photo 1,Photo 2,Photo 3, Photo 4 (URLs)

**Etape Final:**

Rediriger vers /moncompte qui affichera en *4 sous-onglets*

* Mon Compte: Récap de l’étape 1
* Ma Localisation: Récap de l’étape 2 (avec GMAP en option pour localiser)
* Mes Photos: 4 photos en gallerie grid
* Ma Biographie: La Biographie

*Bonus: A tous moment lors des étapes d’enregistrement, l’utilisateur peut revenir en arrière à l’étape précédente avec un bouton*

*Bonus 2: A tout moment, l’utilisateur peut “reseter” son compte avec un bouton et revenir à la première étape*

*Bonus Ultime: Utiliser la célèbre suite de composants React Validation pour valider les formulaires*

[*https://www.npmjs.com/package/react-validation*](https://www.npmjs.com/package/react-validation)

**Banque de liens**

**Documentation React**

[**https://reactjs.org/docs/hello-world.html**](https://reactjs.org/docs/hello-world.html)

**React Cheat Sheet**

[**https://devhints.io/react**](https://devhints.io/react)

**React Fundamentals**

[**https://tylermcginnis.com/courses/react-fundamentals/**](https://tylermcginnis.com/courses/react-fundamentals/)

**React OpenClassRoom**

[**https://openclassrooms.com/courses/build-web-apps-with-reactjs**](https://openclassrooms.com/courses/build-web-apps-with-reactjs)

**React pour les Débutants**

[**https://www.udemy.com/reactjs-pour-les-debutants/learn/v4/t/lecture/6212226?start=409**](https://www.udemy.com/reactjs-pour-les-debutants/learn/v4/t/lecture/6212226?start=409)

**Programmation Fonctionnelle expliquée à ma grand-mère**

[**https://www.youtube.com/watch?v=-Cdt-1tWJ3M**](https://www.youtube.com/watch?v=-Cdt-1tWJ3M)

**React How To**

[**https://github.com/petehunt/react-howto**](https://github.com/petehunt/react-howto)

**Doc React Create App**

[**https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md**](https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md)

**Le fonctionnement de React**

[**http://blog.ippon.fr/2016/04/19/comment-bien-debuter-en-reactjs/**](http://blog.ippon.fr/2016/04/19/comment-bien-debuter-en-reactjs/)

**React fondamentaux**

[**https://www.youtube.com/watch?v=bTTjQzrYuFk**](https://www.youtube.com/watch?v=bTTjQzrYuFk)

**React tutoriaux**

[**https://www.youtube.com/watch?v=CpYRr\_t4AFc**](https://www.youtube.com/watch?v=CpYRr_t4AFc)

**React Demos**

[**https://github.com/ruanyf/react-demos**](https://github.com/ruanyf/react-demos)

**React Overview TutorialPoint**

[**https://www.tutorialspoint.com/reactjs/index.htm**](https://www.tutorialspoint.com/reactjs/index.htm)

**React Todo Liste**

[**https://www.youtube.com/watch?v=Qpso16P60UA**](https://www.youtube.com/watch?v=Qpso16P60UA)

**Introduction à React**

<http://putaindecode.io/fr/articles/js/react/>

**React in 5 Minutes**

<https://medium.freecodecamp.org/learn-react-js-in-5-minutes-526472d292f4>

**React tips**

<https://www.upwork.com/hiring/development/reactjs-tips-best-practices/>

**Best Practicies**

<https://engineering.musefind.com/our-best-practices-for-writing-react-components-dec3eb5c3fc8?gi=da276abe5929>

**Best Practicies Part 2**

<https://github.com/markerikson/react-redux-links/blob/master/react-architecture.md>

**React Pattern and Tips**

<https://vasanthk.gitbooks.io/react-bits/>