Snark factory

PETITS EXERCICES DE DEV

1. Un peu de front

Côté Front : On a une div "A" qui peut prendre dynamiquement n'importe quelle taille (en fonction d'autres critères). On doit créer une fonction "resized" qui sera appelée à chaque changement de taille de cette div. Dans cette div, on veut afficher soit une image "portrait" soit une image "paysage" en fonction du ratio que prend le div. Cette image devra être affichée à l'intérieure de la div "A" avec un espace de 15 pixels entre les bords du div et l'image.

```
<div class="A">
    <img id="imgA" src="portrait.png|paysage.png"/>
</div>
```

1) indiquez comment vous feriez pour déclencher la fonction "resized" à chaque changement de taillde la div "A"

2) écrivez le style CSS + l'intérieur de la fonction "resized" pour avoir ce résultat.

Vous pouvez utiliser jQuery ou une autre lib que vous connaissez (à préciser) pour vous faciliter la vie si vous le souhaitez.

2. Analyse de données + Typescript

Si vous ne connaissez pas Typescript, vous pouvez résoudre le problème avec du simple Javascript. Si vous ne connaissez pas Javascript, vous pouvez résoudre le problème dans le langage que vous souhaitez.

Soit les 2 types suivants :

```
type Command = {
    _id: string
    date: number
    totalPrice: number
    products: { [key: string]: number }
}

type Product = {
    ref: string
    name: string
    price: number
}
```

Dans l'objet products à l'intérieur du type Command, les clés correspondent au champs ref du type Product et le nombre correspond au nombre de produits de cette référence dans la commande.

Par exemple :

```
const commands = [
    {
        _id: "1",
        date: 1234566789900,
        totalPrice: 475,
        products: {
            refA: 3,
            refB: 5
        }
    },
    {
        _id: "1",
        date: 1234566789900,
        totalPrice: 475,
        products: {
            refB: 4
        }
    }
];

const products = [
    {
        ref: "refA",
        name: "A",
        price: 75
    },
    {
        ref: "refB",
        name: "B",
        price: 50
    }
];
```

On a donc dans une commande qui contient 3 produits A et 5 produits B et une autre commande qui ne contient que 4 produits B. Nous avons 2 fonctions qui nous permettent de récupérer dans la base de données les différents éléments :

```
function getCommands(): Promise<Array<Command>>;
// retourne de manière asynchrone une liste de toutes les commandes.

function getProducts(refList: Array<string>|null): Promise<Array<Product>>;
// retourne de manière asynchrone une liste de produits correspondants à la liste de référence passée.
// si refList est null, cette focntion retourne tous les produits.
```

1) Faire une fonction prenant toutes les commandes qui affiche la liste des produits avec à côté la quantité de ces produits dans l'ensemble des commandes.

```
function analyzeCommands() {

}
```

Le résultat avec notre exemple devrait être

```
const result = {
    refA: 3,
    refB: 9
};
```

3. Extraction de données

Si vous ne connaissez pas Javascript, vous pouvez résoudre le problème dans le langage que vous souhaitez.

On a les données suivantes qui représentent l'organisation d'une entreprise avec ses services et sous-services.

Pour chacun d'eux, on a la date de la dernière vérification de sécurit

```
const organisation = {
    name: "entreprise",
    securityCheck: "03/10/2021",
    services: [
        {
            name: "commercial",
            securityCheck: "03/10/2021",
            services: [
                {
                    name: "vente",
                    securityCheck: "14/11/2021"
                },
                {
                    name: "marketing",
                    securityCheck: "30/06/2020",
                },
                {
                    name: "communication",
                    securityCheck: "08/03/2021",
                    services: [
                        {
                            name: "design",
                            securityCheck: "27/07/2021"
                        }
                    ]
                }
            ]
        },
        {
            name: "technique",
            securityCheck: "24/08/2021",
            services: [
                {
                    name: "electronique",
                    securityCheck: "14/11/2021"
                },
                {
                    name: "robotique",
                    securityCheck: "30/06/2020",
                    services: [
                        {
                            name: "mécanique",
                            securityCheck: "12/08/2021"
                        },
                        {
                            name: "automatisme",
                            securityCheck: "01/09/2021"
                        },
                        {
                            name: "IA",
                            securityCheck: "07/09/2021"
                        }
                    ]
                }
            ]
        },
        {
            name: "logistique",
```

```
            securityCheck: "03/10/2021",
            services: [
                {
                    name: "transport",
                    securityCheck: "14/11/2021"
                },
                {
                    name: "etiquetage",
                    securityCheck: "30/06/2020"
                },
                {
                    name: "nettoyage",
                    securityCheck: "05/09/2021"
                },
            ]
        },
    ]
}
```

1) Écrivez une fonction qui puisse lister les services et sous-services qui n'ont pas eu de vérification de sécurité depuis plus de N mois.

```
function securityUncheckedSince(data, nbMonth) {
    // ...
}
```

Qui retournera un résultat du type :

```
const result = securityUncheckedSince(organisation, 3);
// result =
result = [
    "marketing",
    "communication",
    "design",
    "technique",
    "robotique",
    "mécanique",
    "automatisme",
    "etiquetage",
    "nettoyage"
];
```

L'ordre de retour dans le tableau n'est pas important.

On ne doit pas voir apparaitre plusieurs fois le même nom de service.