



PROJET JAVASCRIPT :

Objectif : Réaliser un site web à l'aide HTML 5, CSS 3 et Javas Script

Ce site sera constitué de 5 pages HTML:

- Une page de connexion : index.html
- Une page d'accueil qui reprend le site (sans bootstrap) : accueil.html
<https://startbootstrap.com/previews/freelancer>
- Un page contenant une toDoList : taches .html
- Une page contenant une liste de produits et un panier : produits.html
- Une page contenant un jeux de casse brique : briques.html

1 - La Page de connexion (connexion.html)

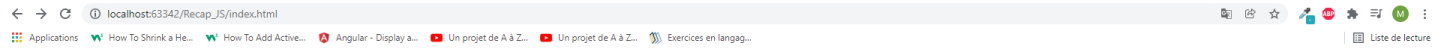
- Cette page doit contenir un formulaire avec 3 champs
- Un champ email
- Un champ Mot de passe
- Un champ nombre entre 0 et 999 aléatoire
- Si un des 3 champs est vide, on déclenche une erreur
- Si email, mot de passe et nombre aléatoire ne sont pas strictement égale aux valeurs Javascript, on déclenche une erreur
- Lorsque les 3 champs matches, on cache le formulaire et on déclenche un loader l'aide la fonction javascript setTimeout(votre fonction, délais en milliseconde)

ALGO :

- déclarer des constantes email et mot de passe
- créer une variable et générer un nombre aléatoire avec Math.round et Math.random
- Récupérer id du label nombre aléatoire et afficher un nombre aléatoire
- Créer une fonction connexion() javascript
- Récupérer les valeurs des 3 champs (faite du debug : console.log(champ))
- Vérifie que les champs ne sont pas vide avec une structure conditionnelle (if else)
- Afficher des erreurs en cas de champ vide
- Si les champs ne sont pas vide : vérifié l'égalité entre les champs email, mot de

passer et nombre aléatoire

- Si les 3 champs matches : effectués une redirection vers la page accueil.html
- Cacher le formulaire et afficher le loader
- sinon afficher une erreur de connexion
- LES +
- Créer une fonction qui vide les 3 champs en cas d'erreurs



CONNEXION

Email
mic@gmail.com

Mot de passe
Mot2P@sse

CODE : 544
483

CONNEXION

2 – La page d'accueil (accueil.html)

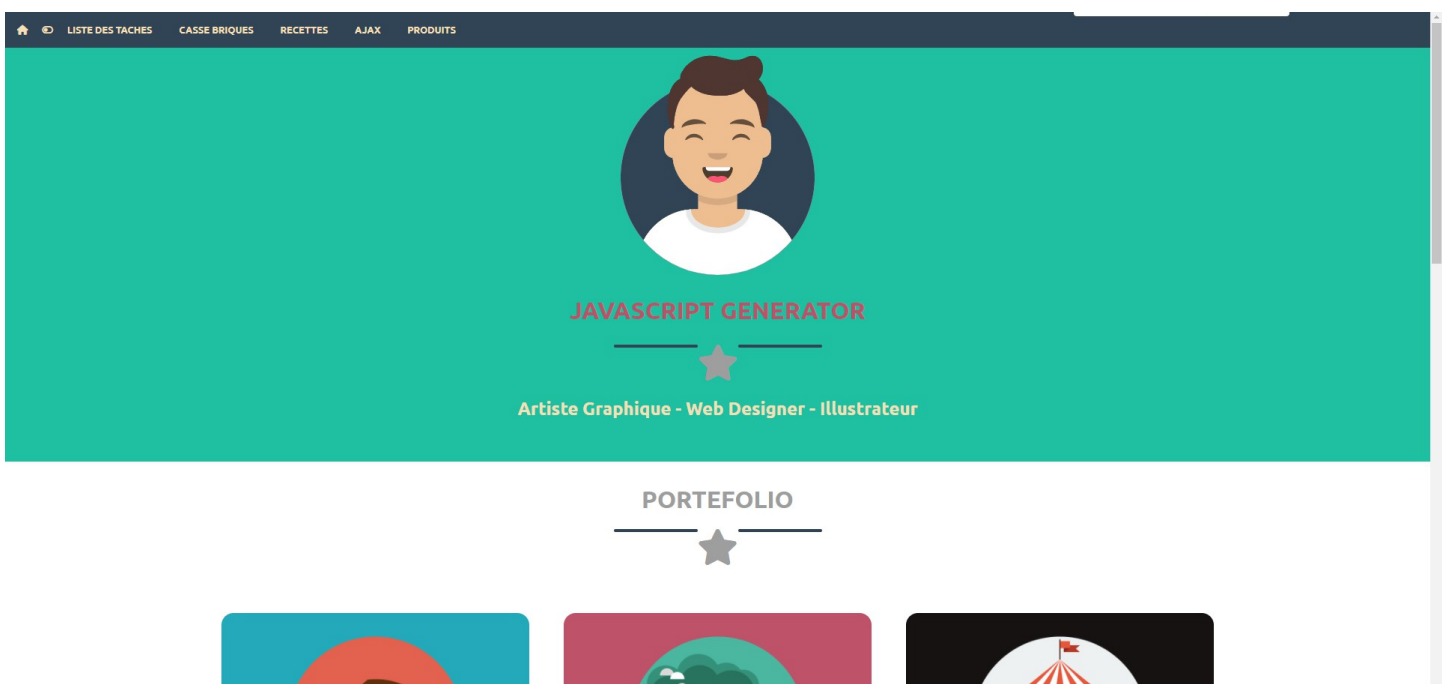
- Cette page reprend le modèle du site :
- <https://startbootstrap.com/previews/freelancer>
- Créer 5 fichiers javascript (bloc1, 2, 3, 4 et footer.js)
- Dans chaque bloc, récupérer l'id de la div HTML et générer du contenu à l'aide de javascript et la concaténation ES6 (back quotes AltGr + 7)
- Votre site doit être responsive sans bootstrap (Flex Box)
- accueil .html appelle les 5 fichiers javascript pour générer son contenu
- Utilisé la méthode getElementById() pour récupérer des éléments HTML
- Utilisé la méthode createElement() pour créer des balises HTML
- Utilisé la méthode appendChild() pour ajouter des nœuds enfants aux nœuds parents
- Utilisé les Flex box pour le responsive
- Notamment display : flex
- justify-content : center
- align-items : center
- Et flex-wrap : wrap

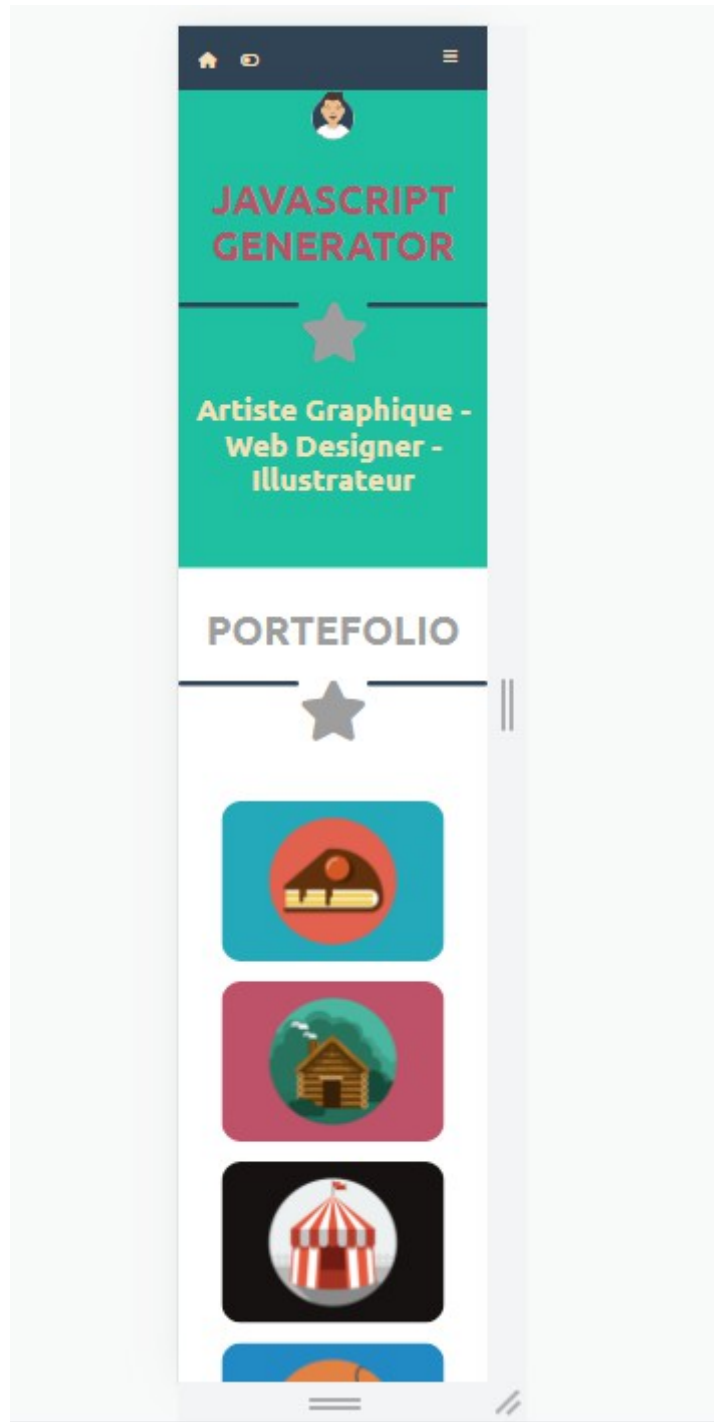

```
Recap_JS > js > footer.js

Project
└─ Recap_JS C:\web_dev\Recap_JS
   └─ css
   └─ img
   └─ js
      └─ bloc1.js
      └─ bloc2.js
      └─ bloc3.js
      └─ bloc4.js
      └─ connexion.js
      └─ footer.js
      └─ leftMenu.js
      └─ listeTaches.js
      └─ menu.js
      └─ produits.js
      └─ accueil.html
      └─ index.html
      └─ produits.html
      └─ taches.html

External Libraries
Scratches and Consoles

15
16 <header id="menu" class="menuClasse"></header>
17 <aside id="leftAside" class="aside"></aside>
18
19 <div class="bloc1" id="bloc1">
20
21 </div>
22
23 <div class="bloc2" id="bloc2">
24
25 </div>
26
27 <div class="bloc3" id="bloc3">
28
29 </div>
30
31 <div class="bloc4" id="bloc4">
32
33 </div>
34
35 <footer id="footer"></footer>
36
37 <script src="js/menu.js" type="text/javascript"></script>
38 <script src="js/leftMenu.js" type="text/javascript"></script>
39 <script src="js/bloc1.js"></script>
40 <script src="js/bloc2.js"></script>
41 <script src="js/bloc3.js"></script>
42 <script src="js/bloc4.js"></script>
43 <script src="js/footer.js"></script>
44 </body>
45 </html>
```





3 - Le menu horizontale

- SOURCES : https://www.w3schools.com/howto/howto_js_topnav_responsive.asp
- Créer un menu Horizontale responsive et afficher les liens vers vos pages (HTML ou JS)
- Quand écran est < à 768px cacher les liens et faite apparaître un bouton pour ouvrir votre menu sous forme de bloc (font-awesome)
- Créer une fonction javascript capable de cacher et afficher votre bloc de menu

ALGO :

- Déclencher une fonction a l'aide de l'attribut onclick= 'maFonction()' HTML sur le bouton visible quand l'écran est < 768px
- Récupérer votre menu a l'aide de javascript et attribut HTML id (getElementById())
- Dans une structure conditionnelle : si votre menu possède une certaine classe, ajouter une seconde classe définie dans votre fichier responsive.css, sinon laisser la classe par défaut (toggle)

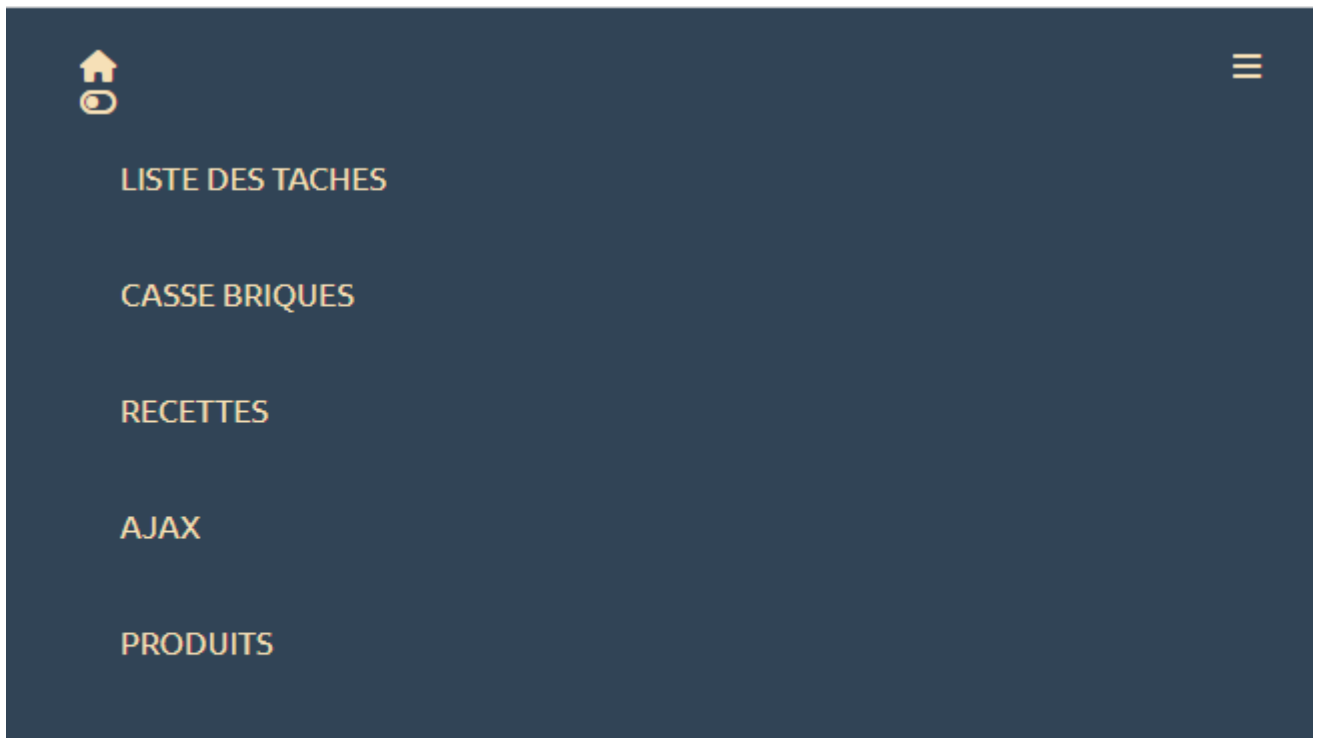
```
function showHideMenu(){  
    let menuParent = document.getElementById( elementId: "menu");  
    if(menuParent.className === "menuClasse"){  
        menuParent.className += " responsive";  
    }else{  
        menuParent.className = "menuClasse"  
    }  
}
```



LISTE DES TACHES



LISTE DES TACHES



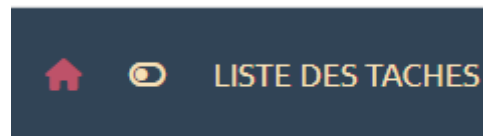
LISTE DES TACHES

4 - Le menu sur la gauche

- SOURCES : https://www.w3schools.com/howto/howto_js_sidenav.asp
- Créer un bouton dans menu Horizontale qui déclenche un menu sur la gauche qui prend 100% de la page (icône home font-awesome)
- Sur ce bouton appelé une fonction javascript a l'aide de l'attribut `onclick='maFonction()'` HTML
- Créer un fichier `menuGauche.js` qui déclenche votre menu au clic sur l'icône du menu horizontale
- Créer 2 fonctions javascript capable de cacher et afficher votre bloc de menu gauche
- Le menu est caché par défaut a l'aide de CSS (sa largeur est égale a 0 et devient 100% au clic sur le bouton et la fonction `ouvrirMenuGauche()`)

ALGO :

- Dans `menu.js` : créer une fonction pour ouvrir le menu de gauche
- Dans `menuGauche.js` :
- Récupérer votre bloc HTML a l'aide de son attribut `id` et javascript
- Ajouter votre navigation à l'aide de `innerHTML()` et la concaténation ES6
- Créer une icône qui appel une fonction pour fermer le menu (code : `×`)

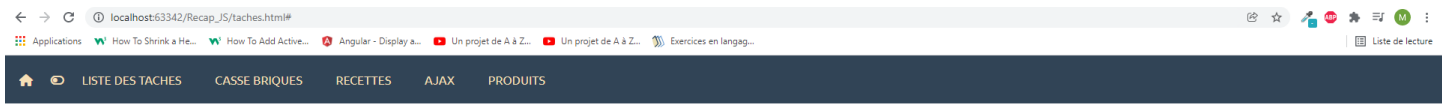


5- Le mode sombre

- SOURCES : https://www.w3schools.com/howto/howto_js_toggle_dark_mode.asp
- Ajouter une icône au menu horizontale (font awesome toggle-off et on)
- Au clic sur cette icône, on ajoute une classe à la balise <body> qui colore le fond en sombre et la couleur du texte devient blanche

ALGO

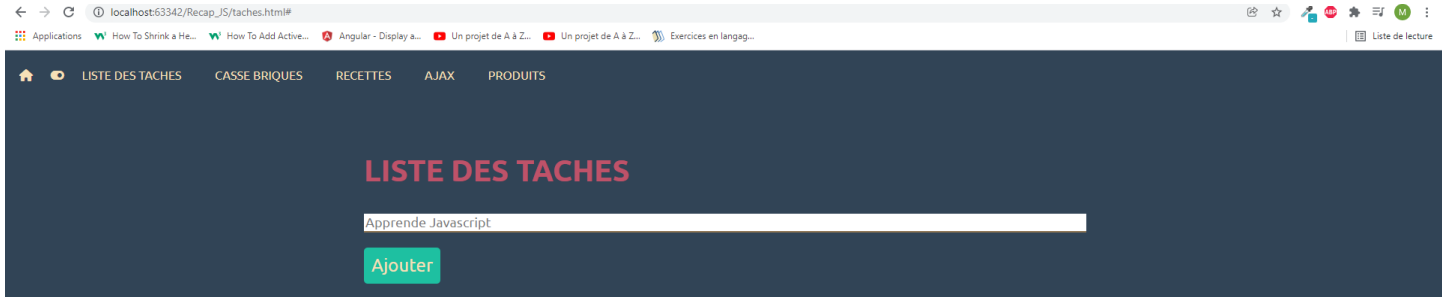
- Créer une classe CSS .dark-mode avec un background-color sombre + color : white
- La balise HTML de votre icône appelle une fonction à l'aide de l'attribut onclick='maFonction()' (ICONE)
- Dans votre fichier menu.js :
- Créer une fonction qui récupère la balise body
- ajouter la classe au body à l'aide de la méthode toggle()
- Dans une structure conditionnelle (if else)
- Si votre menu possède une classe (ex : fa fa-toggle-off) changer la classe à l'aide de toggle pour modifier l'icône



LISTE DES TACHES

Apprendre Javascript

Ajouter



6- La liste des taches

- SOURCES : https://www.w3schools.com/howto/howto_js_todolist.asp
- Dans un champ input de type 'text' l'utilisateur entre une tache à effectuer
- Lors de la soumission via un bouton et l'appel d' une fonction javascript, on génère un élément HTML qui affiche la tache entrée par l'utilisateur
- Un bouton supprimer est également généré dans chaque tache affichée pour supprimer cette dernière
- Au clic sur la tache ajoutée on change la couleur de fond, on barre la tache et on ajoute une coche a l'aide d'un pseudo élément CSS::before



ALGO :

- Cote HTML : Un titre + un input type 'text' + id + placeholder
- Un bouton qui appel une fonction nouvelleTaches() a l'aide de l'attribut onclick='nouvelleTaches()'
- Une liste vide + un attribut id='ce que tu veux'
- COTE JAVASCRIPT
- Créer une fonction nouvelletaches()

- Générer un nœud élément HTML avec la méthode createElement('li')
- Ajouter une classe pour le CSS (element.className = 'votre attribut classe CSS')
- Récupérer la valeur de input HTML a l'aide de son ID et la méthode .value()
- document.getElementById('id').value
- Tester avec console.log(variableInput)
- Si le champ est vide déclenché une erreur
- Sinon ajouter la valeur de l'input a
- Ajouter a en tant qu'enfant (.appendChild())
- Vider le champ input type 'text' après avoir ajouté la tache
- Générer un bouton avec du texte supprimer (ou X) grâce a DOM et la méthode .createElement('button')
- Ajouter du texte au bouton à l'aide de la méthode .innerHTML('Du texte')
- Ajouter une classe au bouton avec la méthode .className('maClasse CSS')
- Ajouter les boutons a chaque (.appendChild())
- Récupérer tous les éléments qui possède la classe du bouton (.getElementByClassName('maClasse'))
- Dans une boucle for(init ; condition ; incrémentation){INSTRUCTION}, récupérer tous les nœuds élément qui on l'attribut classe concerné
- A chaque bouton ajouter un clic (addEventListener('click', function(){INSTRUCTION}))
- bouton[i].onclick = function(){INSTRUCTION}
- Récupérer le parent du bouton donc (bouton[i].parentElement)
- Au clic cacher le à l'aide de CSS
- parent.style,.display = 'none'

Un effet au clic sur chaque générer

- Créer une fonction barrerTache()
- Récupérer votre liste a l'aide de l'attribut ID HTML
- A chaque clic sur une tache, déclenché un événement
- Si contient un alors on change de classe CSS à l'aide de toggle()
- Appeler votre fonction en bas de page
- Créer et Ajouter les propriétés CSS a la classe ajouter-retirée via javascript
- Ici c'est la classe .checked
- La coche est réalisée à l'aide du pseudo-element .classe::before
- Et les propriétés CSS border

7 - La Page du casse briques (briques.html)


- SOURCES : https://developer.mozilla.org/fr/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript

8 - Les produits, un panier et les calculs (TVA 20% + TTC) (produits.html)

- Afficher des produits à partir d'un tableau d'objet avec une boucle forEach()
- Au clic sur chaque produit, ce dernier disparaît du tableau produit et est ajouté au tableau panier et réciproquement
- Dans le panier lors de l'ajout de quantité, on affiche le nom du produit + la quantité + le prix Hors taxe + le montant de TVA à 20% + le montant Toutes Taxes Comprise


NOS PRODUITS

Enceintes KrK




Description :
Des enceintes de pro pour du son superbe
Prix HT : 125.25 Euros

Iphone Apple



Description :
Telephone Apple dernier cris, reconditionner
Prix HT : 789.25 Euros

SNES MINI



Description :
Version minifier de la Super Nintendo
Prix HT : 325.25 Euros

PANIER

NOS PRODUITS

Enceintes KrK



Description :
Des enceintes de pro pour du son superbe
Prix HT : 125.25 Euros

SNES MINI



Description :
Version minifier de la Super Nintendo
Prix HT : 325.25 Euros

PANIER




Iphone Apple

Quantite(s)

Prix HT :
789.25 Euros

NOS PRODUITS

Enceintes KrK



Description :
Des enceintes de pro pour du son superbe
Prix HT : 125.25 Euros

SNES MINI



Description :
Version minifier de la Super Nintendo
Prix HT : 325.25 Euros

PANIER

Iphone Apple X Quantite(s) : 2 = 1578.5 Euros

Prix HT: 1578.5 Euros + TAUX 20% : 316 Euros
= 1894.5 Euros

TOTAL TTC :
1894.5EUROS



Iphone Apple

Quantite(s)

Prix HT :
789.25 Euros

ALGO :

- **ETAPES 1**

- Cote HTML :

- Une div conteneur principale
- A l'intérieur : une div 70% de large pour afficher le produits (titre + liste)
- Une autre div de 30% large pour afficher le panier (titre + div total HT + div total TTC + liste pour le panier

- Cote Javascript :

- LES VARIABLES GLOBALES (utilisable partout)
- Créer un tableau d'objet (tableau associatif clé + valeur) avec les propriétés suivante :
- id + nomProduits + imageProduits + descriptionProduits + prixProduits + produitQuantite + produitsTTC

[illegible]

- Créer une variable qui stocke un tableau vide panier = []
- Créer une fonction afficherProduit()
- A partir du tableau produits : créer une boucle forEach() avec un alias produit qui appelle une fonction anonyme (rappel en ES6 () => {})
- Générer un nœud élément HTML à l'aide de la méthode createElement('li')
- Ajouter une classe et un id dynamique via ES6 (rappel : pour appelé une variable : back quotes (AltGr + 7) + `\${alias.propriété.id})
- Ajouter des éléments HTML à chaque avec la concaténation ES6 (nomProduits + description + image + prix HT)
- Ajouter chaque au parent
- **ETAPES 2**
- Sur chaque déclenché un clic avec la méthode addEventListener('click', () =>

{INSTRUCTION})

- Ajouter l'objet cliqué au tableau vide panier via la méthode push()
- Caché l'objet ajouter avec du CSS (display : none)
- Créer une fonction en dehors du scope de la fonction afficherProduit() qui s'appelle afficherPanier()

```
//Fonction afficher les produits du tableau d'objet produits
function afficherProduit(){
    //La méthode forEach() est une boucle qui permet d'exécuter une fonction donnée sur chaque élément d'un tableau.

    tableauProduits.forEach((produit :{...}) => {
        //generation d'un noeud element <li>
        let produitElement = document.createElement( tagName: "li");
        //Ajout d'une classe
        produitElement.className = "produit-card";
        //Ajout d'un id dynamique
        produitElement.id = `produit${produit.id}`;
        //Ajout de contenu HTML avec la concatenation ES6
        //Accès aux données d'objet avec alias (produits).element de l'objet
        //ex : produit.nomProduit
        produitElement.innerHTML =
        `
        <h3 class="titre-produit">${produit.nomProduits}</h3>
        
        <p>Description : </p>
        <p>${produit.descriptionProduits}</p>
        <p>Prix HT : ${produit.prixProduit} Euros</p>
        `
        //Ajout des <li> a <ul>
        carteProduits.appendChild(produitElement)
        //Au clic sur chaque <li> on ajoute l'objet cliquer au tableau vide panier
        produitElement.addEventListener( type: "click", listener: () => {
            panier.push(produit);
            //On cache l'element cliqué du tableau produits
            produitElement.style.display = "none"
            //On appel la fonction afficher le panier
            afficherPanier()
        });
    });
}
```

• ETAPES 3

- Dans la fonction afficherPanier()
- Créer une variable vide de type String = ''
- Créer une boucle de lecture forEach() sur le tableau panier rempli dans l'étape 2
- Ajouter a votre variable vide des éléments HTML via la concaténation ES6 et l'opérateur +=
- nom produit + image + un select - option pour les quantité + prix HT + **une croix pour supprimer l'élément du panier**
- Ajouter un ID dynamique au <select><option>
- Ajouter tous ces élément a le liste panier HTML via son attribut ID et innerHTML = votre variable remplie
- EXEMPLE

```

function afficherPanier() {

    //On creer une variable vide
    let ingredientChoisi = "";
    //On redup le tableau ingredients choisis rempli par les click sur les ingredients
    panier.forEach(produit => {
        //On ajoute les elements de la boucle a notre variable
        ingredientChoisi +=

        <li class="produit-card-panier" id="produitAjouter${produit.id}">
            <div id="btn-supprimer${produit.id}" class="supprimer-produit">✕</div>
            
            <h3 class="titre-produit">${produit.nomProduits}</h3>

            <select class="select-quantite" id="selectQuantite${produit.id}">
                <option selected>Quantite(s)</option>
                <option>1</option>
                <option>2</option>
                <option>3</option>
                <option>4</option>
            </select>

            <p class="prix-para">Prix HT :<br /> ${produit.prixProduit} Euros</p>
        </li>

    });

    document.getElementById( elementId: "panier").innerHTML = ingredientChoisi;
}

```

• ETAPES 4

• Supprimer un produit du panier

- Créer une boucle de parcours sur le tableau panier via forEach() et un alias
- Récupérer l'id dynamique de chaque produit du tableau Produit
- Idem pour le tableau panier
- Idem pour chaque croix (bouton-supprimer) du tableau panier
- Ajouter un événement clic sur le bouton supprimer (la croix) via la méthode addEventListener('click', function(){INSTRCUTION})
- Re-afficher le produit dans le tableau produit via CSS display : block
- Créer une variable et stocker l'index du tableau panier a l'aide de la méthode indexOf(alias)
- Faire une copie du tableau panier via la méthode splice(index, nombre d'élément a supprimer) et supprimer le produit Panier concerné a l'aide de la méthode remove()
- Vider les champs le montant HT + TVA + TTC (.innerHTML = '')

```

//supprimer un produit du panier
panier.forEach(produitSupprimer => {

    let produit = document.getElementById( elementId: `produit${produitSupprimer.id}`);
    let produitAjouter = document.getElementById( elementId: `produitAjouter${produitSupprimer.id}`);
    let btnSupprimer = document.getElementById( elementId: `btn-supprimer${produitSupprimer.id}`);

    btnSupprimer.addEventListener( type: "click", listener: () => {
        produit.style.display = "block";
        //produitAjouter.style.display = "none"
        let produitIndex = panier.indexOf(produitSupprimer);
        //On retire l'ingredient du tableau grace a son index
        panier.splice(produitIndex, deleteCount: 1);
        //On supprime l'ingredient de l'affichage
        produitAjouter.remove();
        document.getElementById( elementId: `totalHT${produitSupprimer.id}`).innerHTML = "";
        document.getElementById( elementId: `totalTTC${produitSupprimer.id}`).innerHTML = "";
        document.getElementById( elementId: "totalTTC").innerHTML = ""

    });
});

```

- Hors de la boucle de lecture
- Récupérer les <div> totalHTConteneur et totalTTConteneur via l'attribut id des balises HTML
- Récupérer la valeur du <select><option> via son ID dynamique
- Créer une variable pour stocké le prixHT dans une div et ajouter un id dynamique
- Ajouter cette div au parent HTML à l'aide .appendChild(totalHTConteneur)
- Créer une variable pour stocké le prixTTC dans une div et ajouter un id dynamique
- Ajouter cette div au parent HTML à l'aide .appendChild(totalTTConteneur)


```

//Les conteneurs
let prixHTConteneur = document.getElementById( elementId: "totalHTConteneur");
let prixTTConteneur = document.getElementById( elementId: "totalTTConteneur")

//Recup quantiter depuis le select
let selectQuantite = document.getElementById( elementId: `selectQuantite${produitSupprimer.id}`);

let prixHT = document.createElement( tagName: "div");
prixHT.id = `totalHT${produitSupprimer.id}`

prixHTConteneur.appendChild(prixHT)

let prixTTC = document.createElement( tagName: "div");
prixTTC.id = `totalTTC${produitSupprimer.id}`
prixTTConteneur.appendChild(prixTTC)

```

- Récupérer la valeur du <select><option> via la méthode `addEventListener('change', function(){INSTRUCTION})`
- Créer une variable `index` et stocker la valeur du select via la méthode `selectedIndex`
- Créer une variable qui est égale à `(index multiplier prixHT)`
- Créer une variable et récupérer l'id dynamique du conteneur Prix TTC
- Dans une variable stocker le calcul de la TVA et faire un arrondi via l'objet `Math.round(prixHT * 0,2)`
- Créer une variable `prixTTC` qui est égale à `prixHT + montant TVA`
- Créer une variable `tableauTTC` = à un tableau vide `[]`
- Créer une boucle de parcours de ce tableau via la méthode `forEach` + un alias qui retourne une fonction
- Créer une variable `somme = 0`
- Ajouter au `tableauTTC` vide le montant TTC du tableau produits via la méthode `push()`
- Dans une boucle `for` parcourez le `tableauTTC`
- Ajouter la somme à chaque élément du `tableauTTC`
- Récupérer la div à l'aide de son ID `totalTTC`
- Ajouter les calcul `HT + TVA = TTC` via la méthode `innerHTML`


```

selectQuantite.addEventListener( type: "change", listener: () =>{
    let index = selectQuantite.selectedIndex;
    console.log(index)
    //Afficher prix HT
    let prixHT = document.getElementById( elementId: `totalHT${produitSupprimer.id}`);
    let prixDynamiqueHt = index * produitSupprimer.prixProduit;

    //TTC
    let prixTTCConteneur = document.getElementById( elementId: `totalTTC${produitSupprimer.id}`);
    let prixTva = Math.round( x: prixDynamiqueHt * 0.2);

    let prixTTC = prixDynamiqueHt + prixTva;

    produitSupprimer.produitTTC = prixTTC
    let tableauTTC = [];

    tableauProduits.forEach(alias => {
        let sum = 0
        tableauTTC.push(alias.produitTTC)
        console.log(tableauTTC)
        for (let i = 0; i < tableauTTC.length; i++){
            sum += tableauTTC[i];
            console.log(sum)
            let afficherTTC = document.getElementById( elementId: "totalTTC");
            afficherTTC.innerHTML = "<p class='nom'>TOTAL TTC : </p>" + "<b class='nom'>" + sum + "</b>" + "EUROS"
        }
    })

    prixHT.innerHTML = `<br /><span class="nom">${produitSupprimer.nomProduits} X Quantite(s) : ${index} = ${prixDynamiqueHt} Euros</span><br />`;
    prixTTCConteneur.innerHTML = `<br /><span class="quantite">Prix HT: ${prixDynamiqueHt} Euros + TAUX 20% : ${prixTva} Euros = ${produitSupprimer.produitTTC} Euros</span><br />`;

    afficherPanier();

```

En dehors de la boucle forEach()

- Afficher le prixHT et TTC a l'aide de innerHTML
- Rafraîchisseur la page avec l'appel de la fonction récursive afficherPanier