Support de soutenance - Projet 5 : Premiers pas sur le langage JavaScript[Introduction]  
Bonjour à tous,Je suis ravi de vous présenter aujourd'hui le projet 5 de ma formation d'intégrateur web. Ce projet porte sur la dynamisation d'un site internet statique pour une imprimerie familiale nommée Print it, en remplaçant la bannière par un carousel interactif. J'ai travaillé sur ce projet en utilisant les fondamentaux de JavaScript. Permettez-moi de vous expliquer les étapes clés de mon travail.[Explication des étapes clés]Étape 1 : Mise en place de l'environnement de travail  
Pour commencer, j'ai configuré mon environnement de travail en installant un éditeur de code, comme VS Code, afin de pouvoir écrire et tester mon code JavaScript. J'ai également ajouté les fichiers HTML et CSS nécessaires à mon projet, ainsi que les images à afficher dans le carousel.Étape 2 : Ajout des éléments de navigation du carousel  
J'ai ajouté du code HTML et CSS pour créer les flèches de navigation du carousel. Ce code permet de positionner les flèches à gauche et à droite du carousel. J'ai également ajouté des classes CSS pour styliser ces flèches.Étape 3 : Ajout des Event Listeners sur les flèches de navigation  
En utilisant JavaScript, j'ai mis en place des Event Listeners qui se déclenchent au clic sur les flèches de navigation. Ces Event Listeners sont capables de différencier le clic sur la flèche de gauche de celui sur la flèche de droite.Étape 4 : Affichage des bullets points  
J'ai modifié le code HTML et CSS pour afficher des bullets points en bas du carousel. Ces bullets points permettent de signaler la position de l'image actuellement affichée dans le carousel. J'ai également utilisé du JavaScript pour mettre en évidence le bullet point correspondant à l'image affichée.Étape 5 : Modification du slide affiché  
Au clic sur les flèches de navigation, j'ai programmé le changement d'image dans le carousel. À chaque clic, l'image change, ainsi que le bullet point correspondant. J'ai également ajouté la modification du texte correspondant à l'image affichée dans le carousel.Étape 6 : Ajout du défilement infini  
Pour améliorer l'expérience utilisateur, j'ai ajouté le défilement infini dans le carousel. J'ai pris en compte les cas particuliers du clic sur la flèche de gauche pour la première diapositive et celui sur la flèche de droite pour la dernière diapositive. Ainsi, lorsque l'utilisateur atteint la première ou la dernière image, le carousel affiche respectivement la dernière ou la première image pour donner l'effet de défilement infini.[Difficultés rencontrées]  
Pendant la réalisation de ce projet, j'ai rencontré certaines difficultés. Par exemple, j'ai dû comprendre comment manipuler les classes CSS pour changer l'apparence des flèches et des bullets points. J'ai également dû trouver une solution pour adapter le code afin qu'il fonctionne correctement dans différents navigateurs web. Ces défis m'ont permis d'approfondir ma compréhension du langage JavaScript et des principes de base de la programmation.[Conclusion]  
Ce projet a été une excellente opportunité pour moi d'explorer et de pratiquer les fondamentaux de JavaScript dans un contexte réel. J'ai appris à construire un carousel interactif à partir de zéro, en utilisant les fonctionnalités de base du langage JavaScript. Je suis satisfait du résultat obtenu et je suis prêt à répondre à vos questions.Merci pour votre attention.

Question 1 : Pourquoi avez-vous choisi de dynamiser la bannière du site avec un carousel ?  
Réponse : J'ai choisi d'utiliser un carousel car cela permet d'afficher de manière attrayante plusieurs images différentes sur une même partie du site. Cela permet également de mieux mettre en avant les produits et services de l'imprimerie familiale.Question 2 : Pourquoi avez-vous utilisé JavaScript pour réaliser le carousel plutôt qu'une librairie existante ?  
Réponse : Dans le cadre de ce projet, nous devions être capables de programmer en utilisant les fondamentaux de JavaScript, c'est pourquoi j'ai réalisé le carousel manuellement, sans utiliser de librairie. Cela me permet également de mieux comprendre le fonctionnement interne du carousel et d'avoir unôle total sur ses fonctionnalités.Question 3 : Comment avez-vous difféié le clic sur la flèche de gauche et celui sur la flèche de droite ?  
Réponse : J'ai ajouté des Event Listeners aux deux flèches de navigation et à chaque clic, je vérifie la classe HTML de la flèche qui a été cliquée. En de cette classe, j'ai programmé les actions spécifiques pour passer à la slide précédente ou à la slide suivante.Question 4 : Comment avez-vous signalé la position de l'image actuellement affichée ?  
Réponse : J'ai ajouté des bullets points en bas du carousel grâce à du code HTML et CSS. J'ai utilisé des classes spécifiques pour indiquer le slide actuel et les autres slides. Au fur et à mesure que le carousel défile, je change dynamiquement les classes pour que le bon bullet point soit signalé.Question 5 : Comment avez-vous géré le défilement infini des images dans le carousel ?  
Réponse : J'ai pris en compte les cas particuliers du clic sur la flèche de gauche pour la première diapositive et celui de la flèche de droite pour la dernière diapositive. Lorsque l'utilisateur clique sur une flèche, je vérifie si l'image suivante ou précédente est la première ou la dernière. Dans ce cas, je fais apparaître respectivement la dernière ou la première image, créant ainsi l'effet de défilement infini.

1. Tout'abord, vous avez créé un tableau slides qui contient une liste d'objets. Chaque objet représente une diapositive et contient deux propriétés : image qui représente le chemin d'accès à l'image de la diapositive, et tagLine qui représente le texte de légende de la diapositive. Les balises HTML dans la propriété tagLine sont encodées en HTML pour éviter toute injection de code (ex. "<" pour "<")
2. Ensuite, vous avez récupéré les flèches de navigation du carousel à partir du HTML en utilisant les sélecteurs de classe .arrow\_left et .arrow\_right. Vous avez également récupéré les éléments HTML de l'image et du texte de légende avec les sélecteurs .banner-img et #banner p.
3. Vous avez initialisé la variable currentSlideIndex à 0, qui représente l'index de la diapositive actuelle dans le tableau slides.
4. Vous avez également récupéré l'élément HTML qui représente le conteneur des bullet points du carousel avec le sélecteur .dots.
5. Ensuite, vous avez utilisé une boucle forEach pour parcourir chaque objet dans le tableau slides. À l'intérieur de la boucle, vous avez créé un élément HTML de type span pour représenter chaque bullet point. Vous avez également ajouté la classe "dot" à chaque bullet point et ajouté le bullet point au conteneur des bullet points en utilisant dotsContainer.appendChild(slide.bulletPoint). Cette étape permet de générer dynamiquement les bullet points pour chaque diapositive.
6. Vous avez ajouté la classe "dot\_selected" au premier bullet point de la première diapositive en utilisant slides[0].bulletPoint.classList.add("dot\_selected"). Cette étape permet de mettre en évidence visuellement le premier bullet point pour signaler la diapositive actuelle.
7. Ensuite, vous avez attaché des écouteurs d'événements aux flèches de navigation. Lorsque la flèche de gauche est cliquée, vous effectuez les opérations suivantes :  
      - Vous supprimez la classe "dot\_selected" du bullet point actuel de la diapositive en cours.  
      - Vous décrémentez currentSlideIndex pour passer à la diapositive précédente. Si currentSlideIndex devient -1 (car il s'agit de la première diapositive), vous le mettez à l'index de la dernière diapositive (slides.length - 1), ce qui permet le défilement infini du carousel.  
      - Vous mettez à jour l'image et la légende en utilisant imageSlide.src et tagLine.innerHTML en fonction de la diapositive actuellement affichée.  
      - Vous ajoutez la classe "dot\_selected" au bullet point de la nouvelle diapositive en cours pour le mettre en évidence visuellement.
8. Vous avez fait la même chose pour la flèche de droite, mais cette fois-ci vous effectuez les opérations suivantes :  
      - Vous supprimez la classe "dot\_selected" du bullet point actuel de la diapositive en cours.  
      - Vous incrémentez currentSlideIndex pour passer à la diapositive suivante. Si currentSlideIndex devient égal à slides.length (car il s'agit de la dernière diapositive), vous le mettez à 0 pour revenir à la première diapositive, permettant ainsi le défilement infini du carousel.  
      - Vous mettez à jour l'image et la légende en utilisant imageSlide.src et tagLine.innerHTML en fonction de la diapositive actuellement affichée.  
      - Vous ajoutez la classe "dot\_selected" au bullet point de la nouvelle diapositive en cours pour le mettre en évidence visuellement.J'espère que cette explication détaillée vous aide à comprendre comment votre code fonctionne. N'hésitez pas à me poser d'autres questions si nécessaire !