

Auto évaluation - TypeScript

Partie 1 – Questions de cours (8 points)

- Quelle est la principale différence entre JavaScript et TypeScript ?

Réponse : TypeScript est un sur-ensemble de JavaScript qui ajoute le typage statique et la compilation vers JavaScript. JavaScript est interprété, tandis que TypeScript est compilé avant exécution

- Cite deux avantages de TypeScript et un inconvénient.

Réponse : inconvénient : ajoute une complexité d'apprentissage et avantages

- Que signifie “typage statique” ?

Réponse : Le typage statique signifie que les types des variables et fonctions sont connus et vérifiés avant l'exécution, au moment de la compilation. Cela permet de détecter les erreurs de type à l'avance

- Quelle commande permet d'initialiser un projet Node.js avant d'installer TypeScript ?

Réponse : `npm init -y`

- Quelle est la commande pour compiler un fichier TypeScript (app.ts) vers JavaScript ?

Réponse : `npx tsc app.ts` ou `npx tsc src/app.ts --outDir dist`

- À quoi sert le fichier `tsconfig.json` ?



Réponse : Il configure les options du compilateur TypeScript (comme le dossier de sortie, la version ES ciblée, la vérification stricte, etc.)

- Quelle est la différence entre `any` et `unknown` ?

Réponse : `any` désactive totalement la vérification de type (la variable peut contenir n'importe quoi), tandis que `unknown` nécessite une vérification du type avant son utilisation, ce qui le rend plus sûr

- Que se passe-t-il si vous essayez d'ajouter un nombre et une chaîne de caractères en TypeScript ?

Réponse : TypeScript détecte une erreur de typage à la compilation, car l'opération n'est pas valide entre `number` et `string`

	Technologie du Web	
	Javascript	

Partie 2 – Types et syntaxe (6 points)

Complétez ou corrigez les extraits suivants :

- Déclare une variable note de type nombre et initialise-la à 17.

Réponse : `let note: number = 17;`

- Écris un tableau de chaînes nommé prenomns contenant 'Alice', 'Bob', 'Charlie'.

Réponse : `let prenomns: string[] = ['Alice', 'Bob', 'Charlie'];`

- Corrige l'erreur de typage suivante : `let age: number = '25';`

Réponse : `let age: number = 25;` ou `let age: string = '25';`

- Déclare un objet personne contenant un nom (string), un âge (number), et actif (boolean).

Réponse : `let personne: { nom: string; age: number; actif: boolean } = { nom: 'Alice', age: 30, actif: true};`

- Crée un type personnalisé Produit avec les propriétés nom: string et prix: number. Puis déclare une variable article de ce type.

Réponse : `type Produit = { nom: string; prix: number;};`
`let article: Produit = { nom: 'Ordinateur', prix: 1200};`

- Déclare un enum Couleur contenant Rouge, Vert et Bleu. Puis assigne la valeur Rouge à une variable maCouleur.

Réponse : `enum Couleur { Rouge, Vert, Bleu}`
`let maCouleur: Couleur = Couleur.Rouge;`

Partie 3 – Exercice pratique (6 points)

Corrige le code suivant pour qu'il soit valide en TypeScript :

Programme initial

```
const compteur =
document.querySelector("#compteur");
let i = 0;

const increment = (e) => {
i++;
compteur.querySelector("span").innerText = i;
};

compteur.addEventListener("click", increment);
```

Programme corrigé pour TypeScript

```
// On récupère l'élément HTML avec l'ID "compteur".
// En TypeScript, querySelector peut retourner "Element | null".
// Donc on précise le type et on vérifie qu'il n'est pas null ensuite.
const compteur = document.querySelector("#compteur") as HTMLElement | null;

// On initialise la variable i à 0
let i = 0;

// On déclare la fonction "increment" avec un paramètre d'événement typé
correctement.
// Ici, l'événement est un MouseEvent (car on va l'utiliser sur un clic).
const increment = (e: MouseEvent): void => {
// On incrémente le compteur
i++;
// Avant d'accéder à "querySelector", on s'assure que "compteur" n'est pas null.
// De plus, on précise le type du "span" pour éviter une erreur TypeScript.
if (compteur) {
const span = compteur.querySelector("span") as HTMLElement | null;
if (span) {
// innerText est une propriété de HTMLElement
span.innerText = i.toString(); // on convertit i (number) en string
}
}
};

// Même chose ici : on vérifie que "compteur" n'est pas null avant d'ajouter un écouteur
if (compteur) {
compteur.addEventListener("click", increment);
}
```