

Cadriciel Web côté client

Cours 5 - Prog. et Styles (CSS) React

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Rappel des principes de bases Javascript et nuances Typescript
- 2 Introduction à TailwindCSS
- 3 Les itérations du HTML avec des structures et de la logique JS/TS
 - Étapes pour refactoriser/programmer de manière dynamique (itérations dans le HTML)

Rappel des principes de bases Javascript et nuances Typescript

Site intéressant pour revoir Javascript (W3School) :
<https://www.w3schools.com/js/default.asp>.

Au niveau de Typescript, le typage est maintenant **obligatoire**.

Par contre, il est permis de mettre un “wildcard” nommé **any** si on ne sait pas le type ou si l’on veut rapidement compiler le code.

Attention : Vercel **nécessite obligatoirement** au minimum l’utilisation des wildcards, et au mieux les types spécifiques.

TypeScript correctement typé (string)

```
1 function MyButton({ title }: { title: string }) {  
2   return (  
3     <button>{title}</button>  
4   );  
5 }  
6  
7 export default function MyApp() {  
8   return (  
9     <div>  
10      <h1>Welcome to my app</h1>  
11      <MyButton title="I'm a button" />  
12    </div>  
13  );  
14 }
```

TypeScript avec wildcard (any)

```
1 function MyButton({ title }: { title: any }) {  
2   return (  
3     <button>{title}</button>  
4   );  
5 }
```

Rappel des principes de bases Javascript et nuances Typescript

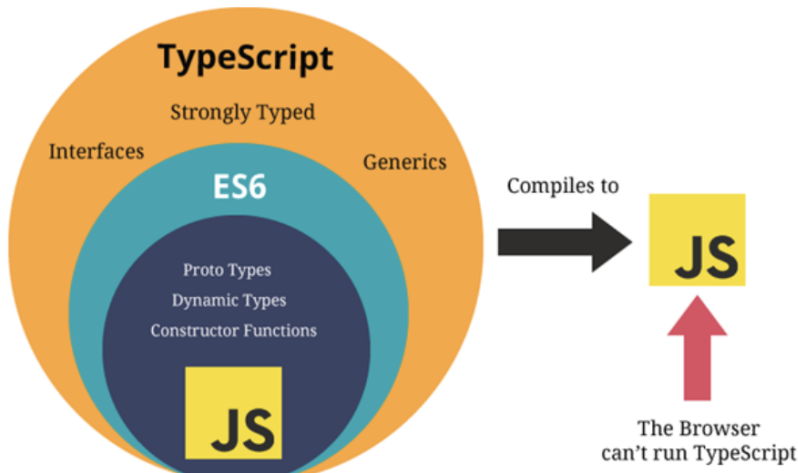


Figure: <https://www.acte.in/typescript-vs-javascript-tutorial>

Introduction à TailwindCSS

Tailwindcss est rendu très populaire. Celui-ci est une librairie permettant de rapidement ajuster le CSS avec des classes.

TailwindUI est un site offrant des artifacts gratuitement :
<https://tailwindui.com/components>.

La doc de Tailwindcss est importante pour comprendre les diverses classes : <https://tailwindcss.com/docs/installation>.

Introduction à TailwindCSS

Un exemple de bouton avec des classes TailwindCSS pour améliorer son apparence.

```
<button
  className="bg-transparent hover:bg-gray-500 text-gray-300 font-semibold hover:text-white py-2 px-4 border border-gray-500 hover:border-transparent rounded"
  onClick={() => setCount(count + 1)}
>
  Increment
</button>
```

Figure: bg-transparent hover:bg-gray-500 text-gray-300 font-semibold hover:text-white py-2 px-4 border border-gray-500 hover:border-transparent rounded

Introduction à TailwindCSS

Le CSS est directement appliqué sur l'élément, pas besoin de fichier CSS traditionnel, mais toujours possible de l'utiliser conjointement.

Si la classe est sur un élément parent, tous les enfants seront affectés. Même principe que le CSS traditionnel.

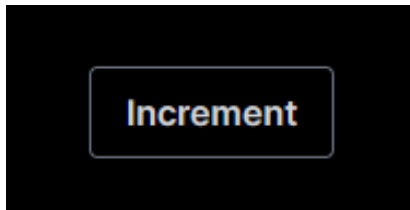
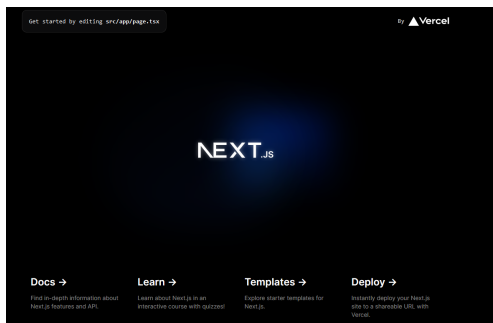


Figure: `bg-transparent hover:bg-gray-500 text-gray-300 font-semibold hover:text-white py-2 px-4 border border-gray-500 hover:border-transparent rounded`

Les itérations du HTML avec des structures et de la logique JS/TS

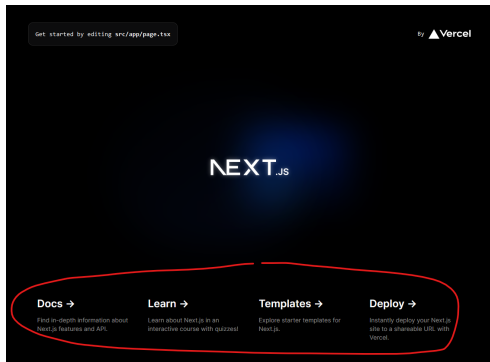
Un paradigme fondamentale de la programmation est de réduire les répétitions le plus possible.

Dans cette page, qu'elles parties se répète d'après vous ?



Les itérations du HTML avec des structures et de la logique JS/TS

Dans cette page, qu'elles parties se répète d'après vous ?



Les itérations du HTML avec des structures et de la logique JS/TS

Pourquoi ?

```
<a
  href="https://vercel.com/templates?framework=next.js&utm_source=
  className="group rounded-lg border border-transparent px-5 py-4
  target="_blank"
  rel="noopener noreferrer"
>
  <h2 className="mb-3 text-2xl font-semibold">
    Templates{" "}
    <span className="inline-block transition-transform group-hover
      -&gt;
    </span>
  </h2>
  <p className="m-0 max-w-[30ch] text-sm opacity-50">
    Explore starter templates for Next.js.
  </p>
</a>

<a
  href="https://vercel.com/new?utm_source=create-next-app&utm_med
  className="group rounded-lg border border-transparent px-5 py-4
  target="_blank"
  rel="noopener noreferrer"
>
  <h2 className="mb-3 text-2xl font-semibold">
    Deploy{" "}
    <span className="inline-block transition-transform group-hover
      -&gt;
    </span>
  </h2>
  <p className="m-0 max-w-[30ch] text-sm opacity-50">
    Instantly deploy your Next.js site to a shareable URL with Ver
  </p>
</a>
```

Étapes pour refactoriser/programmer de manière dynamique (itérations dans le HTML)

NB : Pour programmer directement, les étapes sont plus simples que lors de la refactorisation.

- Détecter les éléments HTML répétitifs et exhaustifs
- Faire une structure de données avec les informations différentes des éléments HTML répétitifs
- Isoler le HTML répétitif et supprimer les autres occurrences
- Faire une boucle et inclure le HTML isolé à l'intérieur
- Avec le sous objet du tableau, inclure les champs dans l'élément HTML
- Exécuter et voir si tout s'affiche comme auparavant

Étapes pour refactoriser/programmer de manière dynamique (itérations dans le HTML)

```
const links = [
  {
    id: 1,
    href: "https://vercel.com?utm_source=create-next-app&utm_medium=appdir-template&utm_campaign=create-next-app",
    title: "Docs",
    desc: "Find in-depth information about Next.js features and API.",
  },
  {
    id: 2,
    href: "https://nextjs.org/learn?utm_source=create-next-app&utm_medium=appdir-template-tw&utm_campaign=create-next-app",
    title: "Learn",
    desc: "Learn about Next.js in an interactive course with quizzes!",
  },
  {
    id: 3,
    href: "https://vercel.com/templates?framework=next.js&utm_source=create-next-app&utm_medium=appdir-template&utm_campaign=create-next-app",
    title: "Templates",
    desc: "Explore starter templates for Next.js.",
  },
  {
    id: 4,
    href: "https://vercel.com/new?utm_source=create-next-app&utm_medium=appdir-template&utm_campaign=create-next-app",
    title: "Deploy",
    desc: "Instantly deploy your Next.js site to a shareable URL with Vercel.",
  },
];
```

Étapes pour refactoriser/programmer de manière dynamique (itérations dans le HTML)

```
<div className="mb-32 grid text-center lg:mb-0 lg:w-full lg:max-w-5xl lg:grid-cols-4 lg:text-left">
  {links.map((link) => (
    <a
      key={link.id}
      href={link.href}
      className="group rounded-lg border border-transparent px-5 py-4 transition-colors hover:border-gray-300 hover:
      target="_blank"
      rel="noopener noreferrer"
    >
      <h2 className="mb-3 text-2xl font-semibold">
        {link.title}{
          <span className="inline-block transition-transform group-hover:translate-x-1 motion-reduce:transform-none">
            -&gt;
          </span>
        }
      </h2>
      <p className="m-0 max-w-[30ch] text-sm opacity-50">{link.desc}</p>
    </a>
  )
  )}
</div>
```

Étapes pour refactoriser/programmer de manière dynamique (itérations dans le HTML)

```
const links = [
  {
    id: 1,
    href: "https://vercel.com?utm_source=create-next-app&utm_medium=default-template&utm_campaign=welcome-banner",
    title: "This value is in the data structure",
    desc: "Hello World !",
  },
]
```

Figure: Il est maintenant possible de changer les valeurs dans la structure de données pour changer rapidement l'information de l'interface Web.

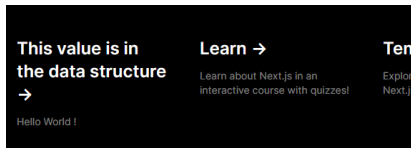


Figure: Résultat sur la page Web.

Bibliographie

- <https://react.dev/>
- <https://tailwindcss.com/>