

# 25

## Créer ou améliorer une activité

---

### 25.1. Mettre les mains dans le cambouis, est-ce nécessaire ?

Non, absolument pas ! Les fonctionnalités offertes par les activités H5P sont déjà très complètes et la grande majorité des utilisateurs n'éprouvera jamais le besoin de créer une nouvelle activité ni de modifier le code d'une activité existante. Mais vous pouvez imaginer que votre institution souhaite utiliser la fonte `Courier New` d'une couleur bleue ou que votre client ait besoin d'une fonctionnalité supplémentaire comme la vidéo à la place d'une image dans l'activité TRUE/FALSE. L'avantage du développement d'un éditeur d'activité H5P est que celui-ci ne demande que des connaissances basiques en programmation, car créer une activité H5P, c'est essentiellement assembler des composants existants, comme vous le feriez avec des briques LEGO<sup>®</sup>. Il suffit alors de décrire la structure désirée dans des fichiers spécialisés au sein du paquet H5P.

Vous trouverez ici un aperçu des innombrables possibilités existantes, qui mériteraient un ouvrage consacré à elles seules... Le lecteur souhaitant approfondir cette matière se référera à la [documentation officielle](#).

*Attention > Le site de H5P reprenant la documentation développeur officielle est incomplet et truffé d'erreurs. Celui-ci n'a pas été remis à jour depuis plusieurs années (au moment d'écrire ces lignes), et les commentaires sont parfois contradictoires. Les lignes suivantes sont issues de mes propres essais et découvertes et le code proposé fonctionne !*

### 25.2. Je dois connaître quoi ?

Les activités H5P sont programmées en JavaScript et sont basées sur le format d'échange JSON.

Le format JSON ([JavaScript Object Notation](#)) est important, comme nous allons le voir. C'est un format standard utilisé pour représenter des données structurées d'une manière identique au code JavaScript. Il se présente sous forme de chaînes de caractères et est facilement lisible et interprétable par l'utilisateur. Un objet JSON est stocké dans son propre fichier texte avec l'extension `*.json`.

Les données se présentent sous la forme :

```
"étiquette" : "valeur"
```

Les données en relation sont encadrées par des accolades et les groupements par des crochets. Le séparateur de champ est la virgule.

```
{
  "title":"True\False Question",
  "language":"und",
  "mainLibrary":"H5P.TrueFalse",
  "embedTypes":[
    "div"
  ],
  "license":"U",
  "defaultLanguage":"en",
  "preloadedDependencies:[
    {
      "machineName":"H5P.Image",
      "majorVersion":"1",
      "minorVersion":"1"
    },
    {
      "machineName":"H5P.TrueFalse",
      "majorVersion":"1",
      "minorVersion":"6"
    }
  ]
}
```

Si ce format reste très lisible et interprétable, une erreur de ponctuation ou d'accolade est vite arrivée... De plus, chaque accolade prenant une ligne en notation standard, le format est particulièrement verbeux.

À côté de JSON, les activités H5P utilisent également le format [CSS](#) afin de spécifier la couleur, les polices et autres éléments de design. Ce format se présente également sous forme de chaînes de caractères et est facilement lisible et interprétable par l'utilisateur, même si la terminologie utilisée demande une certaine expérience. Un objet CSS est stocké dans son propre fichier texte avec l'extension \*.css.

```
.h5p-true-false-answers {
  margin: 1em 0 1em;
}
.h5p-true-false-answer {
  position: relative;
  display: inline-block;
  box-sizing: border-box;
  cursor: pointer;
  border-radius: 4px;
  border: 2px solid #ddd;
  margin: 0 1em 1em 0;
```

```
padding: .5em 3.5em .5em .3em;
box-shadow: 0 1px 0 0 #ccc;
outline: none;
background: #fff;
}
```

La bonne nouvelle est que vous ne devrez pas programmer en JavaScript (ou si peu) pour facilement modifier le comportement ou le design d'une activité H5P ! En effet, tout ce que vous avez à faire est de décrire votre structure de données, et l'outil de génération de H5P se chargera de créer l'éditeur correspondant.

H5P est conçu autour de blocs de construction réutilisables appelés *bibliothèques* H5P. Par exemple, il existe des bibliothèques H5P pour ajouter de la vidéo, des images et du son à votre contenu et pour créer des éléments d'interface utilisateur. Tous les types de contenus H5P existants ont également été conçus pour être réutilisables. Donc si vous avez besoin de questions dans votre activité, il ne vous faudra que quelques minutes pour brancher l'un des dix types de questions existants : ils suivent tous la même structure.

### 25.3. Et j'ai besoin de quoi ?

Pour commencer, je vous conseille l'usage d'un éditeur de code reconnaissant la syntaxe JavaScript, tel que BBEdit, Sublime Text, Notepad++, Atom ou CodeRunner.

Pour tester le code que vous allez produire, je vous conseille d'utiliser un site web Moodle ou WordPress (avec le plugin H5P, qui fait office d'interpréteur), même si cette solution demande le plus de manipulations.

Une solution qui fonctionne bien pour tester le code consiste à utiliser l'application [Lumi](#), qui est un interpréteur en *stand-alone*. Celle-ci permet de tester localement des paquets H5P sans devoir passer par un téléchargement sur un site web. Gratuite et open-source, cette solution fonctionne plutôt bien. De plus, vous pouvez facilement contrôler le numéro de version des paquets utilisés, ce qui est un avantage certain pour le développement, comme vous le verrez dans la suite de ce chapitre.

L'application [Logiquiz](#) propose des fonctionnalités semblables à celles de Lumi, mais s'est révélée moins pratique à l'usage.

Vous pouvez aussi utiliser l'environnement de développement de H5P, nommé `h5pc1i`. Il permet d'augmenter nettement la productivité, mais il exige plus de connaissances techniques pour son installation et pour son utilisation. Aussi ce point, dépassant la portée de cet ouvrage, ne sera pas couvert. Vous pourrez vous reporter à la [documentation officielle](#).