

# Installation et Utilisation d'Ornement

Ornement est une solution logicielle permettant d'**enrichir son texte à l'aide de balises**. L'avantage d'Ornement est **sa simplicité d'installation et d'utilisation**. Ce document à pour but de décrire comment s'utilise Ornement à travers toutes les balises disponibles. Le standard javascript utilisé est l'**ES6**

<b>Installation</b>	<b>1</b>
<b>Utilisation</b>	<b>1</b>
Démon	1
Le moteur	2
Intégrer Ornement dans votre projet	2
<b>Utilisation des différentes balises</b>	<b>3</b>
La balise T	3
La balise L	4
La balise M	5
La balise C	5
<b>Ornement et les failles XSS.</b>	<b>5</b>

## Installation

Pour pouvoir installer Ornement vous pouvez procéder de deux manières:

- Vous pouvez installer Ornement à l'aide de la commande **npm install ornement**.
- Vous pouvez suivre les différentes étapes pour installer Ornement dans votre projet:
  - Télécharger Ornement à partir de son repository dans le dossier que vous souhaitez, cela peut-être dans votre projet.
  - C'est bon, il suffit d'une seule étape pour installer Ornement!

## Utilisation

### Démon

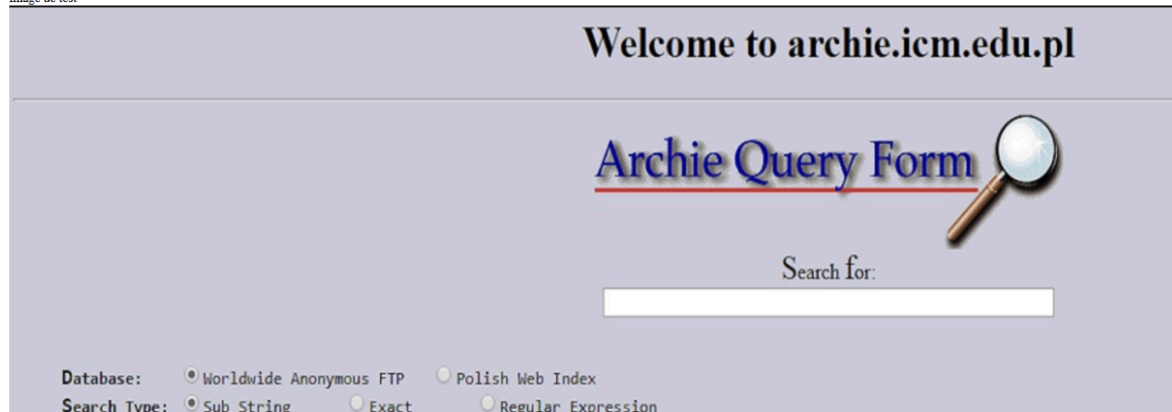
Vous pouvez rapidement voir comment Ornement fonctionne à l'aide du projet disponible sur GitHub, à l'aide de Node-CLI ou à l'aide du projet React disponible dans Demo > ornement\_react\_demo:

- Si vous avez installé Ornement directement depuis git via git clone:
  - À l'aide d'un terminal, installez http-server (npm install http-server) , puis lancez le serveur (http-server)(Vous pouvez aussi utiliser Live Server de VS Code si vous êtes plus à l'aise avec).
  - ouvrez votre navigateur et tapez <http://localhost:8080>.

- Rendez vous jusqu'au dossier où vous avez installé Ornement.
- Allez ensuite dans Démo et ouvrez testForm.html.
- Dans la zone de texte, entrez les différents jeux de test plus bas. voici un exemple:

```
[{type:"b"}]
#Le [!{font-style:"u"}]HTML[/!] (Hypertext Markup Language):
technologie permettant de créer des pages web.;
#test:
#image de test
[!{src:"https://darktouffe.com/files/media/image_moteur_recherche_archie.jpg"}]image_moteur_recherche_archie[/!];
#test 2;
# test code [C]
titre_tag = item.select_one('.s-item_title').span
prix_tag = item.select_one('.s-item_price')
image_tag = item.select_one('.s-item_image-wrapper').img
lien_tag = item.select_one('.s-item_link')
[/C];
#test 3;
[/!]
```

• Le [HTML](#) (Hypertext Markup Language): technologie permettant de créer des pages web.  
 • test  
 • image de test



- C'est comme cela que vous utilisez Ornement avec du Vanilla JS.
- Si vous utilisez directement Node-CLI:
  - Après avoir installé Ornement, rendez-vous dans package.json et modifiez le fichier de la manière suivante:

```
1 {
2   "type": "module",
3   "dependencies": {
4     "ornement": "^0.1.0"
5   }
6 }
7
```

- Créer un dossier de test puis un fichier javascript (ici [test.js](#))
- Entrez le code suivant, vous pouvez remplacer le contenu de input par un des différents jeux de test plus bas:

```
// test.js
import { Ornement } from 'ornement';

const input = [L(type:"b")]

#Le [T{font-style:"u"}]HTML[/T] (Hypertext Markup Language): technologie permettant de créer des pages web.;
#test;
#image de test [M{src:"https://darktouffe.com/files/media/image_moteur_recherche_archie.jpg"}]image_moteur_recherche_archie[/M];
#test 2;
# test code [C]
titre_tag = item.select_one('.s-item__title').span
prix_tag = item.select_one('.s-item__price')
image_tag = item.select_one('.s-item__image-wrapper').img
lien_tag = item.select_one('.s-item__link')
[/C];
#test 3;
[/L];

const result = Ornement(input);
console.log(result);
```

- Exécutez la commande **node test.js**. Vous devriez obtenir un résultat comme celui-ci:

```
<ul><li>Le <span style="text-decoration: underline;">HTML</span> (Hypertext Mark
up Language): technologie permettant de créer des pages web.</li><li>test</li><l
i>image de test <img src='https://darktouffe.com/files/media/image_moteur_recher
che_archie.jpg' /></li><li>test 2</li><li>test code <pre>
titre_tag = item.select_one('.s-item__title').span
prix_tag = item.select_one('.s-item__price')
image_tag = item.select_one('.s-item__image-wrapper').img
lien_tag = item.select_one('.s-item__link')
</pre></li><li>test 3</li></ul>
```

- Si vous voulez utiliser le projet React il faut:
  - Vous rendre dans Demo > ornement\_react\_demo à l'aide d'un terminal, puis exécuter la commande `npm start`.
  - Rendez-vous ensuite à l'aide de votre navigateur sur <http://localhost:3000>. Vous devriez tomber sur une page comme celle-ci dessous. Entrez les jeux de test dans la zone de texte.



- Voici le résultat obtenu



Voici les différents jeux de tests:

- [T{style:"t2", font-style:"u"}]Introduction[/T]
- [M{src:"https://es.wikipedia.org/static/images/icons/wikipedia.png"}]Wikipedia Logo[/M]
- [T{link:"[https://fr.wikipedia.org/wiki/Ada\\_Lovelace](https://fr.wikipedia.org/wiki/Ada_Lovelace)"}][https://fr.wikipedia.org/wiki/Ada\\_Lovelace](https://fr.wikipedia.org/wiki/Ada_Lovelace)[/T]
- Différents éléments dans une liste:
  - [L{type:"b"}]

#Le [T{font-style:"u"}]HTML[/T] (Hypertext Markup Language): technologie permettant de créer des pages web.;

#test;

#test code [C]

titre\_tag = item.select\_one('.s-item\_\_title').span

prix\_tag = item.select\_one('.s-item\_\_price')

mage\_tag = item.select\_one('.s-item\_\_image-wrapper').img

ien\_tag = item.select\_one('.s-item\_\_link')

[/C];

#test 3 <test>;

[/L]

- [L{type:"b"}]

#Le [T{font-style:"u"}]HTML[/T] (Hypertext Markup Language): technologie permettant de créer des pages web.;

```

#test;

#image de test
[M{src:"https://darktouffe.com/files/media/image\_moteur\_recherche\_archie.jpg"}]i
mage_moteur_recherche_archie[/M];

#test 2;

# test code [C]

titre_tag = item.select_one('.s-item__title').span

prix_tag = item.select_one('.s-item__price')

mage_tag = item.select_one('.s-item__image-wrapper').img

lien_tag = item.select_one('.s-item__link')

[/C];

#test 3;

[/L]

```

## Le moteur

Le moteur est composé du fichier **Ornement.js** et du dossier **Parsers**. Ce dossier est lui-même composé de **divers parsers** dont le but est de **générer du code html** à partir des balises d'Ornement qui sont utilisées.

## Intégrer Ornement dans votre projet

Pour intégrer Ornement dans votre projet, comme vu sur la démo il existe deux moyens de le faire.

### Vanilla JS

il faut installer le dossier dans votre projet comme mentionné dans Installation (plus haut). L'utilisation d'une balise se fait dans un champs input ou textarea dans un premier temps **(1)**. Il vous faut ensuite importer Ornement et récupérer le contenu de ce champ **(2)**. Enfin vous passerez au moteur Ornement le contenu récupéré, qui générera et vous retournera les balises html **(3)**.

Nous allons voir en détail l'utilisation d'Ornement en Javascript (**ES6 et versions ultérieurs**), je m'appuie ici sur les fichiers présent dans le dossier Demo qui sont à votre disposition:

1. Créer un champs input ou textarea, et attribuez-lui un ID, ici Ornement.

```
<textarea id="Ornement"></textarea>
```

Créer une balise qui servira à stocker les balises Obtenues à l'aide d'Ornement, ici je crée une balise div, avec pour ID result.

```
<div id="result">
</div>
```

2. Importer Ornement et récupérer les données:
  - a. Importer le moteur Ornement de la manière suivante (adaptez le chemin de fichier à votre situation).

```
import { Ornement } from "../Ornement.js";
```

- b. Récupérer les données de la manière suivante.

```
const textareaValue = document.getElementById("Ornement").value;
```

3. Générer les balises de la manière suivante:

- a. créer une variable dont le but est d'être lié à la balise div ayant pour ID result.

```
const resultDiv = document.getElementById("result");
```

- b. Passez dans Ornement, le contenu récupéré (textareaValue), puis intégrer le résultat retourné par Ornement dans la div result.

```
// Call the Ornement function and display the result
resultDiv.innerHTML = Ornement(textareaValue);
```

Le résultat doit maintenant s'afficher dans votre balise!

N'hésitez pas à vous appuyer sur les fichiers présents dans le dossier Demo.

## Projet Node

Après avoir installer Ornement avec npm install ornement, il vous faut simplement importer Ornement comme dans la démo:

```
1 // test.js
2 import { Ornement } from 'ornement';
```

Vous pouvez ensuite utiliser Ornement de la même manière que dans la Démo, en sauvegardant le résultat dans une variable par exemple.

```

const input = `[L{type:"b"}]

#Le [T{font-style:"u"}]HTML[/T] (Hypertext Markup Language): technologie permettant de créer des pages web.;

#test;

#image de test [M(src:"https://darktouffe.com/files/media/image_moteur_recherche_archie.jpg")]image_moteur_recherche_archie[/M];

#test 2;

# test code [C]

titre_tag = item.select_one('.s-item_title').span
prix_tag = item.select_one('.s-item_price')
image_tag = item.select_one('.s-item_image-wrapper').img
lien_tag = item.select_one('.s-item_link')

[/C];

#test 3;

[/L];

const result = Ornement(input);

```

## React

Pour pouvoir utiliser Ornement avec React, commencez par créer un nouveau projet avec React à l'aide de la commande **npx create-react-app ornement\_react\_demo**. Installez Ornement avec npm install ornement. Ensuite, vous pouvez par exemple créer un composant TestOrnement comme ci-dessous:

```

You, 4 minutes ago | 1 author (You)
1  import { Ornement } from '..ornement';
2  import { useState } from 'react';
3
4
5  function TestOrnement() {
6      var [content, setContent] = useState("");
7      var [result, setResult] = useState("");
8
9      function HandleContentChange(e) {
10         setContent(e.target.value);
11     }
12
13     function ParseContent(){
14         setResult(Ornement(content));
15     }
16
17     return(
18         <>
19             <textarea onChange={HandleContentChange}></textarea>
20             <br/>
21             <button onClick={ParseContent}>Parse to HTML</button>
22             <br/>
23             <div dangerouslySetInnerHTML={{ __html: result }}></div>
24         </>
25     )
26 }
27
28 export default TestOrnement;

```

Le code ci-dessous effectue les étapes suivantes:

1. On commence par importer Ornement.

2. A l'intérieur du composant on crée une zone de texte. Cette zone de texte sera liée à l'événement **handleContentChange** qui sauvegarde tout changement de la zone de texte dans l'état **content**.
3. On crée ensuite un bouton lié à un événement **Onclick**, avec pour callback **ParseContent**, qui traduira le contenu de content et changera l'état **result**. On crée une div où le résultat sera affiché.

Vous pouvez ensuite intégrer ce composant où vous le souhaitez.

## Utilisation des différentes balises

voici le cas général de l'utilisation d'une balise, identifié ici par une base S (elle n'existe donc pas!) :

- **[S{attribut1:"value1", etc.}]Texte[/S]** pour ajouter du texte enrichi, une liste, du code ou une image.
  - **S** correspond à la balise
  - **[S...]** correspond à la balise ouvrante utilisée.
  - **{attribut1:"value1", etc.}** sont les attributs de la balise.
  - **Texte** correspond au texte de la balise.
  - **[/S]** correspond à la balise fermante utilisée.
- Un caractère d'échappement sera prévu pour pouvoir utiliser les symboles librement.

/!\ entre attribut1: et "value1" il ne faut pas d'espace!

La suite du document décrit les diverses balises utilisables, pour chacune d'elles vous pourrez les tester à l'aide du fichier testForm.html.

## La balise T

Cette balise permet d'enrichir un texte de n'importe quelle longueur. Elle s'utilise de la manière suivante :

**[T{attribut1:"value1", etc.}]Texte[/T]**

Les différents attributs permettent d'utiliser la balise de différente manière :

- **font-style**: Permet de styliser le texte. Les valeurs possibles sont:
  - **i**: italique, **[T{font-style:"i"}]Texte[/T]**.
  - **b**: gras, **[T{font-style:"b"}]Texte[/T]**.
  - **u**: souligné, **[T{font-style:"u"}]Texte[/T]**
  - **s**: barré, **[T{font-style:"s"}]Texte[/T]**
  - Vous pouvez utiliser tout ou partie de ces éléments ensembles, exemple : **[T{font-style:"bui"}]Texte[/T]**.
- **link**: Permet d'insérer un lien au texte, **[T{link:"https://fr.wikipedia.org/wiki/Ada\_Lovelace"}]https://fr.wikipedia.org/wiki/Ada\_Lovelace[/T]**.



- **color**: Permet de donner une couleur au format hexadécimal uniquement, [T{color:"#00FF00"}]Texte[/T] (donnera un texte de couleur verte).
- **highlight-color**: Permet de donner une couleur de surlignage au format hexadécimal uniquement, [T{highlight-color:"#00FF00"}]Texte[/T] (donnera un texte souligné en couleur verte).
- **style**: Permet de donner un style au texte. Voici les valeurs possibles:
  - tx: Pour que votre texte soit compris comme un titre, x peut aller de 1 à 6, [T{style:"t3"}]Sources:[/T], [T{style:"t2"}]Sources:[/T], [T{style:"t5"}]Sources:[/T].
  - sup: Pour que votre texte soit en indice, test[T{style:"sup"}]2[/T].
  - sub: Pour que votre texte soit en exposant, test[T{style:"sub"}]2[/T].

Vous pouvez cumuler les divers attributs, par exemple si vous voulez un texte de couleur rouge, surligné en jaune, en gras et souligné, vous utiliserez [T{color:"#FF0022",highlight-color:"#FFFF00",font-style:"ub"}]Plein d'attributs cumulés[/T]. Vous pouvez mettre des espaces, après la virgule, entre les attributs.

## La balise L

Cette balise permet de créer des listes à puces ou des listes numérotées. Elle s'utilise de la manière suivante:

**[L{type:"type"}]**

**#élément1;**

**#élément2;**

**...**

**#élément3;**

**[/L]**

Voici les critères et règles à respecter:

- "type" doit toujours être spécifié, il peut prendre le type vide "", liste à puce "b", ou numérotée "n". Le type vide correspond par défaut au type bullet point:
  - Type liste à puce:
 

```
[L{type:"b"}]
#élément 1;
#élément 2;
#élément 3;
[/L]
```

ou

```
[L{type:""}]
#élément 1;
#élément 2;
#élément 3;
[/L]
```
  - Type numérotée:

```
[L{type:"n"}]
#élément 1;
#élément 2;
#élément 3;
[/L]
```

- Entre la balise ouvrante et le premier élément de la liste, entre chaque élément de la liste, entre le dernier élément de la liste et la balise fermante, il faut un saut de ligne.
- chaque élément de la liste commence par un diez “#” se termine par un point-virgule “;”, entre le diez et l’élément ou entre l’élément et le point-virgule, il peut y avoir plein d’espaces.

## La balise M

La balise M permet d’inclure des médias en ligne, ces médias peuvent être des images, des vidéos ou des audios. Elle s'utilise de la manière suivante :

**[M{src:"lien\_media"}]Texte[/M]**

Voici les caractéristiques de cette balise:

- L’attribut src sert à renseigner l’adresse du média, celui-ci doit obligatoirement être sur internet.
- Entre les balises M, le texte n’est pas obligatoire il vous permet cependant de vous rappeler de l’image que vous avez mis dans le mode édition.
- Les types de médias autorisés sont les suivants:
  - 'jpg', 'jpeg', 'png', 'gif', 'bmp', 'webp' pour les images.
  - 'mp3', 'wav', 'ogg', 'aac' pour les audios.
  - 'mp4', 'webm', 'avi', 'mkv' pour les vidéos.

## La balise C

La balise C permet d’inclure du code stylisé. Elle ne possède pas d’attribut. Elle s'utilise de la manière suivante :

```
[C]
Code
[/C]
```

Il faut sauter une ligne après la balise ouvrante, et sauter une ligne à la fin du code écrit.

# Ornement et les failles XSS.

Ornement dispose d'une protection basique contre les failles XSS, en effet, elle transforme les chevrons < en &lt; et > en &gt;. Cependant il faut mettre en place les protections basiques contre les failles XSS. Il vous est donc conseillé d'utiliser un logiciel comme DOMPurify ou un équivalent.

Pour toutes les autres failles de sécurité comme les injections SQL, il faudra de même mettre en place les défenses nécessaires. **Ornement n'est pas responsable de tout problème de sécurité lié à son utilisation.**