

RAPPELS

```
<script>
```

QU'EST-CE QUE C'EST QUE ÇA ?

Une balise script ouvrante **HTML**

Peut apparaître dans un fichier HTML (.html) ou dans un fichier PHP (.php).

Pas dans un fichier JavaScript (.js)

HTML

```
<div>
  <h3>Un titre</h3>
  <p>Un paragraphe</p>
</div>
```

Balilage, des balises ouvrantes et fermantes, quelques auto-fermantes (
, , <meta/>).
Peut être représenté sous forme d'arbre (dans l'inspecteur).
Définit le sens du document. Interprété par le navigateur web.

CSS

```
body.nav-on nav {  
  transform: translateX(0);  
}
```

Sélecteurs correspondants à des *nœuds* du DOM (Document Object Model), c'est à dire à des éléments de l'arbre créé à partir des balises HTML.

Propriétés et valeurs.

Spécifient l'apparence du document. Interprétés par le navigateur web.

JAVASCRIPT

```
var button = document.querySelector('.toggleButton');  
button.addEventListener('click', function (e) {  
    document.querySelector('body').classList.toggle('nav-on');  
});
```

Pour ajouter de l'interactivité, déclencher des changements dans la page. Le JavaScript s'exécute côté client, dans le navigateur web.

PHP

```
<?php
if (empty($_POST['message'])) {
    echo '<p class="error">Merci de fournir un message</p>';
}
```

S'ouvre avec un *tag* `<?php` et se referme (éventuellement) avec un tag de fermeture `?>`

Exécuté côté serveur. Permet de générer le document HTML dynamiquement (à la demande), et de traiter les données envoyées par le document (formulaires).

Tout ce qui est hors des tags est directement envoyé au client sans interprétation.

JAVASCRIPT \neq PHP

Mais ressemblances de syntaxe...

JavaScript côté client. PHP côté serveur.

Pour accéder aux propriétés et méthodes :

- Le point (.) en JavaScript (`node.classList`)
- La flèche (->) en PHP (`$connexion->close()`)

WORDPRESS

EXÉCUTION

Trois grandes étapes :

1. Chargement
2. Requête
3. Template

1. wp-config, constantes, etc.
2. fonction de l'URL, requête DB pour récupérer les posts
3. appel du thème avec les données

EXTENSIBILITÉ

La personnalisation repose sur :

- Thèmes (templates, hooks)
- Plugins (hooks)

On ne touche pas au cœur sinon on casse les MAJ et extensions tierces.
Thème pour custom appearance. Plugins pour autres fonctionnalités.
Hooks : mécanisme principal dans thèmes et plugins.

TEMPLATES

Personnalisation du balisage des pages.

Voir [WordPress Template Hierarchy](#)



HOO*K ACTION*

```
add_action('wp_enqueue_scripts', function () {  
    // do something  
});
```

HOOK *FILTER*

```
add_filter('the_content', function ($content) {  
    return $content . '<p>© INSEEC 2017</p>';  
});
```

On trouve ces *points d'accroche* dans la **référence des actions** et dans la **référence des filtres**

Ou en explorant le code de WordPress :

```
do_action();  
apply_filters();
```

TYPES DE CONTENUS

- Post
- Page
- Revision
- Attachment

CUSTOM POST TYPES

On peut définir nos types de contenus personnalisés

```
register_post_type('mon_type_perso', []);
```

Voir la doc de [register_post_type](#)