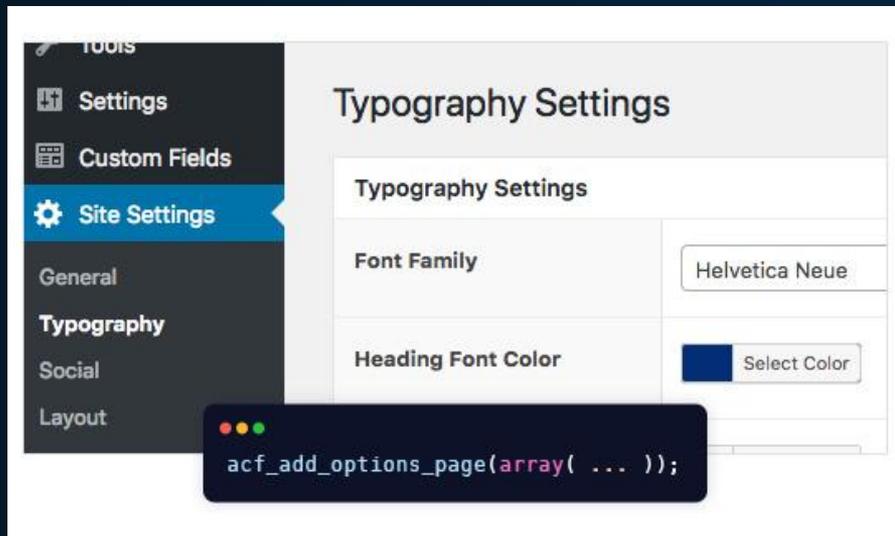




**ACF PRO**



# La page d'options

---

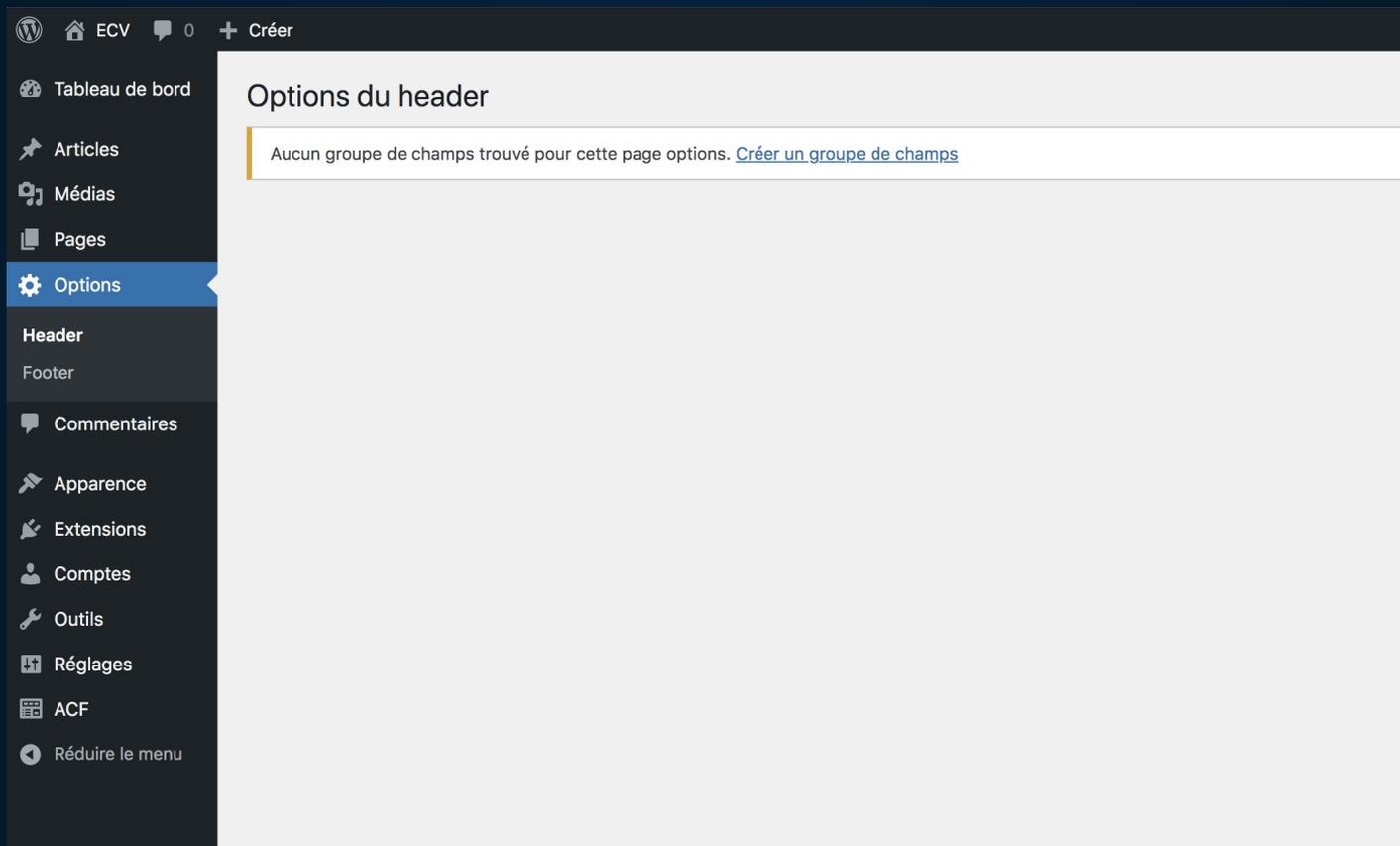
La page d'options fournit un ensemble de fonctions pour ajouter des pages d'administration supplémentaires au back office.

Toutes les données enregistrées sur une page d'options sont globales et peuvent être affichées sur n'importe quelle page du site (ex : le header et le footer).

## Ajouter le menu au back office dans le fichier functions.php

```
if (function_exists('acf_add_options_page')) {
    acf_add_options_page(array(
        'page_title'      => 'Theme General Settings', // titre sur la page d'options
        'menu_title'     => 'Theme Settings', // nom dans le menu
        'menu_slug'      => 'theme-general-settings', // identifiant unique ; affiché aussi dans l'URL
        'post_id'        => 'header_footer', // utilisé pour récupérer les données stockées
        'capability'     => 'edit_posts', // rôle permettant d'éditer un post,
        'redirect'       => true, // false = permet l'accès du menu parent
        'update_button'  => __('Update', 'acf'), // texte du bouton d'enregistrement
        'updated_message' => __('Options Updated', 'acf'), // message de confirmation d'enregistrement
        'icon_url'       => 'dashicons-admin-generic', // nom du dashicons wordpress
        'position'       => 10, // position dans le menu
    ));
    // Ajout d'un sous-menu "Header"
    acf_add_options_sub_page(array(
        'page_title'      => 'Theme Header Settings',
        'menu_title'     => 'Header',
        'post_id'        => 'custom_header',
        'parent_slug'    => 'theme-general-settings', // slug du parent
    ));
    // Ajout d'un sous-menu "Footer"
    acf_add_options_sub_page(array(
        'page_title'      => 'Theme Footer Settings',
        'menu_title'     => 'Footer',
        'post_id'        => 'custom_footer',
        'parent_slug'    => 'theme-general-settings',
    ));
}
```

# Les pages d'options sont maintenant visibles dans le menu du back office.



# Maintenant on peut ajouter notre groupe de champs sur notre page d'options.

The screenshot shows the WordPress dashboard with the 'Advanced Custom Fields PRO' settings for a group named 'Options du header'. The 'Emplacement' section is active, showing the rule 'Page d'options est égal à Option du header'. The 'Règles' section is also visible, with the text 'Créez une série de règles pour déterminer les écrans sur lesquels ce groupe de champs sera utilisé'. The 'Réglages' section is also visible, with the 'Actif' checkbox checked.

**Tableau de bord**

- Articles
- Médias
- Pages
- Options
- Commentaires
- Apparence
- Extensions
- Comptes
- Outils
- Réglages
- ACF**
- Groupes de champs
- Ajouter
- Outils
- Mises à jour
- Réduire le menu

**Advanced Custom Fields PRO** | Groupes de champs | Outils | Mises à jour

## Ajouter un nouveau groupe de champs

Options du header

Ordre	Intitulé	Nom	Type
Aucun champ. Cliquez sur le bouton + Ajouter un champ pour créer votre premier champ.			

+ Ajouter un champ

### Emplacement

**Règles**  
Créez une série de règles pour déterminer les écrans sur lesquels ce groupe de champs sera utilisé

Montrer ce groupe quand

Page d'options est égal à Option du header et

ou

Ajouter une règle

### Réglages

**Actif**

**Style** Dans un bloc

**Position** Normal (après le contenu)

# J'ai ajouté un champ de type groupe avec 2 images à l'intérieur pour avoir 2 versions de logo (desktop + mobile).

1 Logos header\_logo Groupe

**Titre du champ**  
Ce nom apparaîtra sur la page d'édition

Logos

**Nom du champ**  
Un seul mot, sans espace.  
Les « \_ » et « - » sont autorisés

header\_logo

**Type de champ**

Groupe

**Instructions**  
Instructions pour les auteurs.  
Affichées lors de la saisie du contenu

**Requis ?**

Non

**Sous-champs**

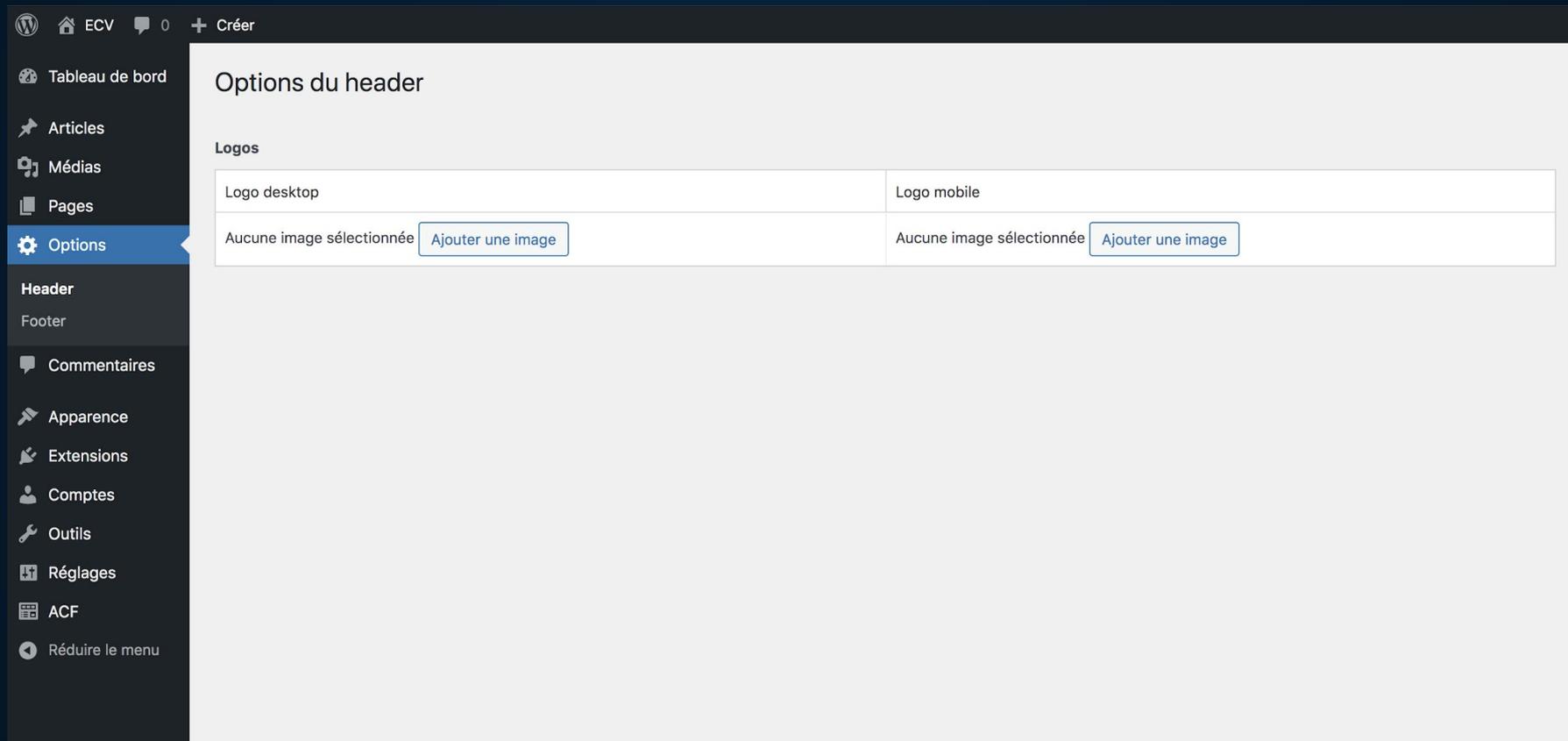
Ordre	Intitulé	Nom	Type
1	Logo desktop	header_logo_desktop	Image
2	Logo mobile	header_logo_mobile	Image

+ Ajouter un champ

**Disposition**  
Définit le style utilisé pour générer les champs sélectionnés

Bloc  Tableau  Rangée

# On peut retourner sur notre menu Options > Header pour voir notre groupe avec nos 2 champs image.



WordPress dashboard navigation: Home, ECV, 0 notifications, + Créer

Left sidebar menu:

- Tableau de bord
- Articles
- Médias
- Pages
- Options**
- Header
- Footer
- Commentaires
- Apparence
- Extensions
- Comptes
- Outils
- Réglages
- ACF
- Réduire le menu

## Options du header

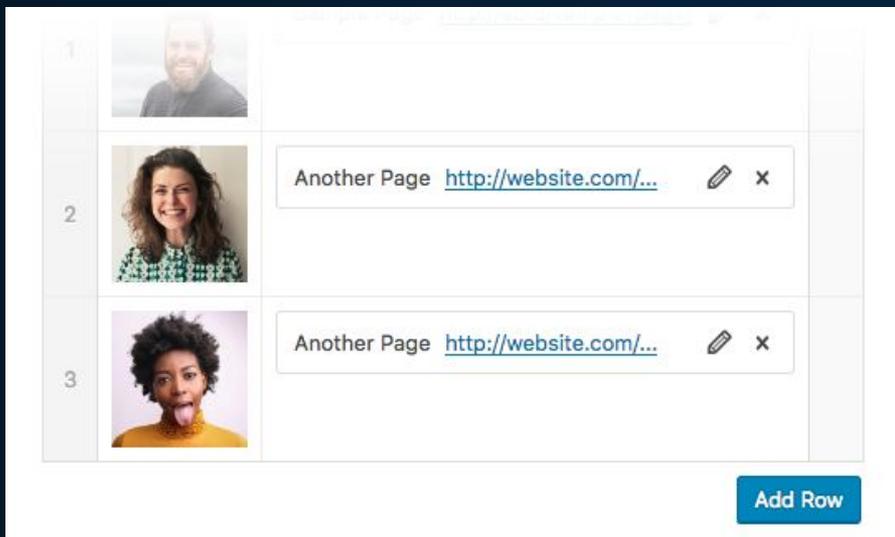
Logos

Logo desktop	Logo mobile
Aucune image sélectionnée <a href="#">Ajouter une image</a>	Aucune image sélectionnée <a href="#">Ajouter une image</a>

## Afficher les données du champ d'options

```
<?php
// le champ "header_logo" étant un type groupe, on récupère les données dans un tableau
// custom_header = la valeur définie de notre "post_id" dans la fonction de création du menu
$logos = get_field('header_logo', 'custom_header');
?>


```



# Le Repeater

Le champ répéteur vous permet de créer un ensemble de sous-champs qui peuvent être répétés.

Tous les types de champ peuvent être ajoutés en tant que sous-champ ce qui vous permet de créer et de gérer des données très personnalisées.

# J'ai créé un champ répéteur avec 3 sous-champs à l'intérieur. Il est aussi possible d'ajouter un répéteur dans un répéteur.

4

**Acteurs** movie\_actor Répéteur  
Modifier Dupliquer Déplacer Supprimer

**Titre du champ**  
Ce nom apparaîtra sur la page d'édition  
Acteurs

**Nom du champ**  
Un seul mot, sans espace.  
Les « \_ » et « - » sont autorisés  
movie\_actor

**Type de champ**  
Répéteur

**Instructions**  
Instructions pour les auteurs.  
Affichées lors de la saisie du contenu

**Requis ?**  
 Non

**Sous-champs**

Ordre	Intitulé	Nom	Type
1	Photo de l'acteur	movie_actor_img	Image
2	Nom de l'acteur	movie_actor_name	Texte
3	Nom du personnage	movie_actor_cname	Texte

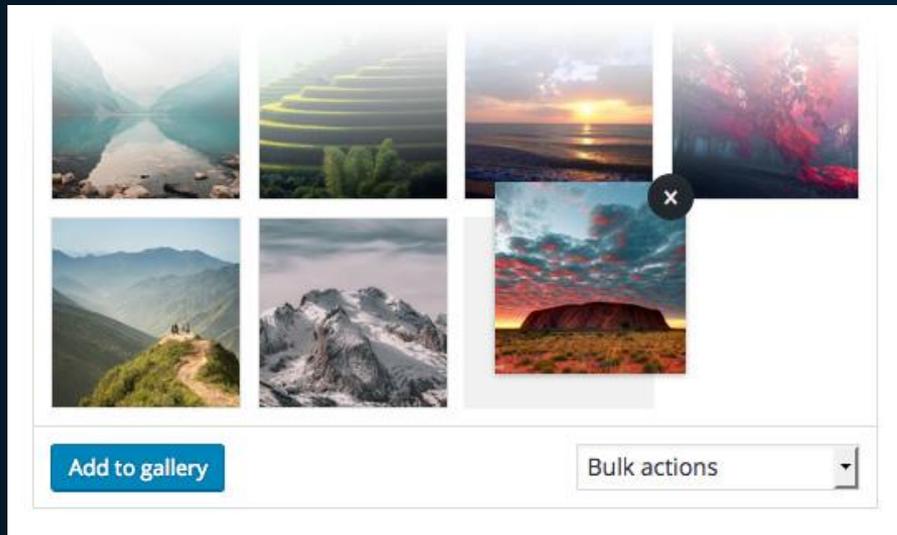
+ Ajouter un champ

## Afficher les données du repeater dans votre modèle

```
// charger le champ dans une variable
$rows = get_field('repeater_field_name');
// check si des lignes existent dans le tableau
if ($rows) {
    // boucle sur les lignes
    foreach ($rows as $row) {
        // afficher la valeur du champ
        echo $row['field_name'];
    }
}
```

## Dans mon cas avec les acteurs

```
<?php $acteurs = get_field('movie_actor'); ?>
<?php if ($acteurs) : ?>
    <div class="box_acteurs">
        <?php foreach ($acteurs as $acteur) : ?>
            <div class="box_acteur">
                ">
                <p><b><?= $acteur['movie_actor_name'] ?></b></p>
                <p><?= $acteur['movie_actor_cname'] ?></p>
            </div>
        <?php endforeach; ?>
    </div>
<?php endif; ?>
```



# La galerie

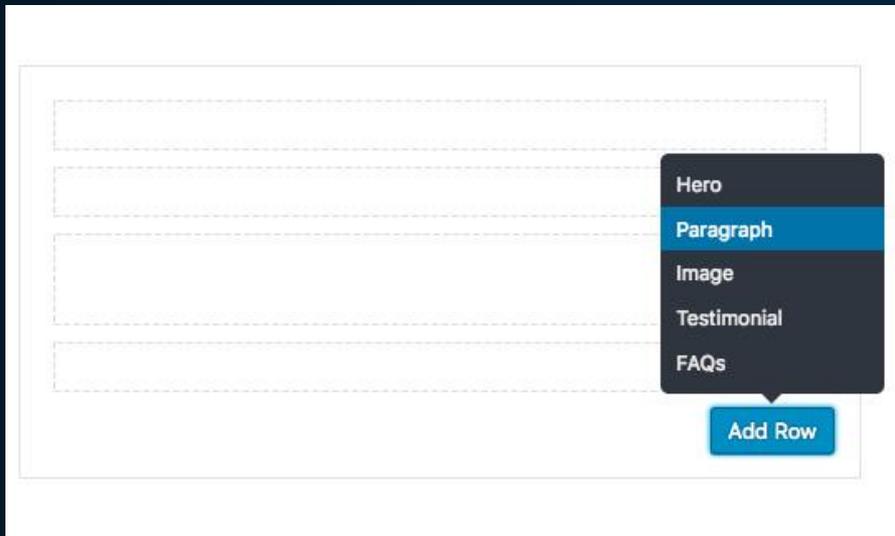
---

Le champ galerie fournit une interface simple et intuitive pour gérer une collection d'images et offre la possibilité d'ajouter plusieurs informations (description, texte alternatif, légende, etc).

Plusieurs images peuvent être ajoutées, modifiées et triées facilement.

## Afficher les données du champ galerie

```
$images = get_field('gallery');  
if ($images) : ?>  
    <?php foreach ($images as $image) : ?>  
        <a href="<?= $image['url']; ?>">  
            " />  
        </a>  
        <p><?= $image['caption']; ?></p>  
    <?php endforeach; ?>  
<?php endif; ?>
```



# Le flexible content

---

Le champ de contenu flexible est un gestionnaire de mise en page de contenu complet.

Il permet de définir des groupes de sous-champs (mises en page) et ajoutez, modifiez et réorganisez-les pour créer un contenu hautement personnalisé!

# Le champ de contenu flexible est légèrement différent des autres champs ACF car les blocs sont créé au premier niveau du champ flexible

**Disposition :**  
C'est ici qu'on va gérer les blocs de notre constructeur

1 Constructeur construct Contenu flexible  
Modifier Dupliquer Déplacer Supprimer

**Titre du champ**  
Ce nom apparaîtra sur la page d'édition

Constructeur

**Nom du champ**  
Un seul mot, sans espace.  
Les « \_ » et « - » sont autorisés

construct

**Type de champ**

Contenu flexible

**Instructions**  
Instructions pour les auteurs.  
Affichées lors de la saisie du contenu

**Requis ?**

Non

**Disposition**  
Réorganiser  
Supprimer  
Dupliquer  
Ajouter

Intitulé

Nom

Disposition Bloc

Min

Max

Ordre	Intitulé	Nom	Type
Aucun champ. Cliquez sur le bouton + Ajouter un champ pour créer votre premier champ.			

+ Ajouter un champ

# J'ai ajouté un bloc titre avec un champ texte et un bloc double colonne avec 2 champs d'éditeur de contenu

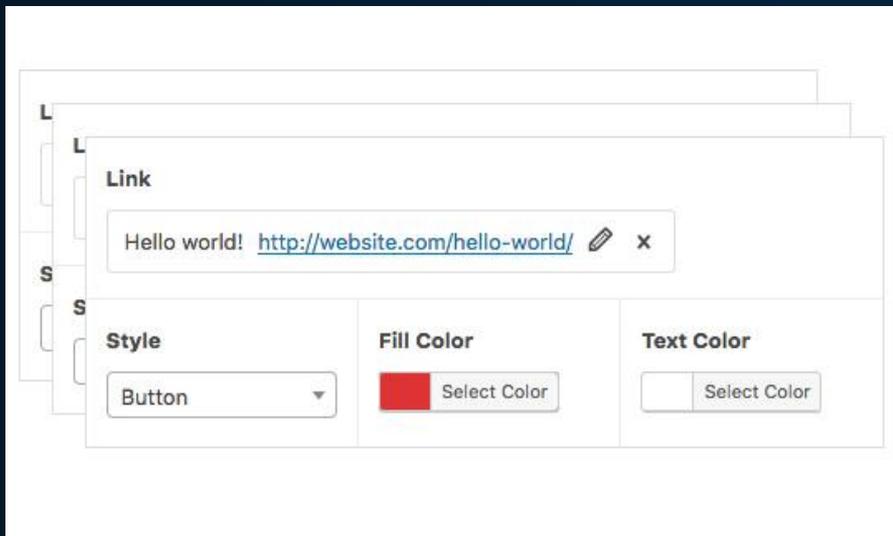
Disposition	Intitulé	Titre		
	Nom	construct_block_01		
	Disposition	Bloc	Min	Max
	<b>Ordre</b>	<b>Intitulé</b>	<b>Nom</b>	<b>Type</b>
	1	Titre	construct_block_01_title	Texte
	<a href="#">+ Ajouter un champ</a>			
Disposition Réorganiser Supprimer Dupliquer Ajouter	Intitulé	Double colonnes		
	Nom	construct_block_02		
	Disposition	Tableau	Min	Max
	<b>Ordre</b>	<b>Intitulé</b>	<b>Nom</b>	<b>Type</b>
	1	Colonne de gauche	construct_block_02_left	Éditeur de contenu
	2	Colonne de droite	construct_block_02_right	Éditeur de contenu
	<a href="#">+ Ajouter un champ</a>			
Intitulé du bouton	Ajouter un bloc			

# J'ai ajouté mes 2 blocs sur ma page, on a aussi la possibilité de déplacer les blocs pour réorganiser notre page

The screenshot displays the WordPress page builder interface. On the left is a dark sidebar with navigation options: Tableau de bord, Articles, Médias, Pages (highlighted), Toutes les pages, Ajouter, Options, Commentaires, Apparence, Extensions, Comptes, Outils, Réglages, ACF, and Réduire le menu. The main workspace is titled 'Constructeur' and shows a page layout with two columns. The first column is labeled '1 Titre' and contains a 'Titre' block. The second column is labeled '2 Double colonnes' and is split into 'Colonne de gauche' and 'Colonne de droite', each with a 'Ajouter un média' button and a rich text editor toolbar. A tooltip in the bottom right corner identifies the 'Ajouter un bloc' button and lists 'Titre' and 'Double colonnes' as available blocks.

## Afficher les données du contenu flexible

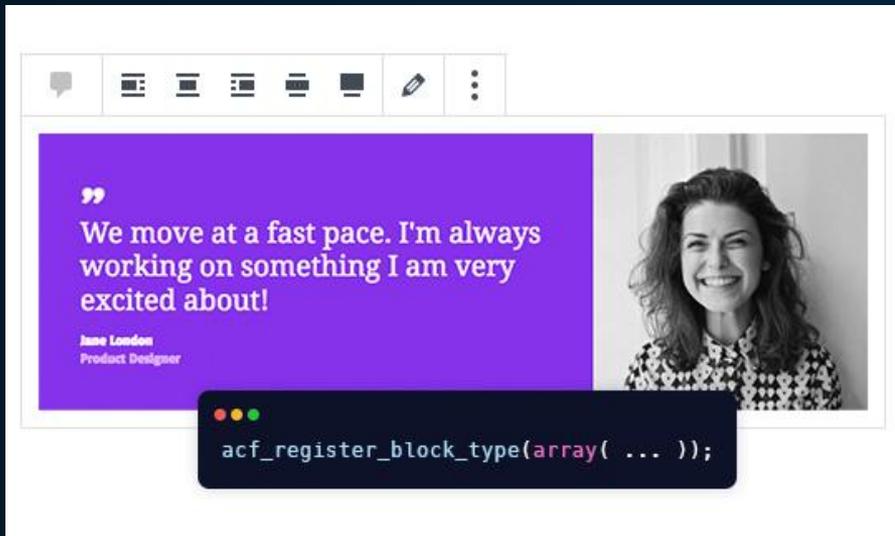
```
<?php
// on check si il y a des blocs dans le constructeur de notre page
if (have_rows('construct')) :
    while( have_rows('construct') ): the_row(); // on boucle sur nos blocs
        // un switch ou un if else pour tester le bloc à afficher
        switch (get_row_layout()) :
            // notre bloc titre
            case 'construct_block_01': ?>
                <h2><?php the_sub_field('construct_block_01_title'); ?></h2>
            <?php break;
            // notre bloc double colonnes
            case 'construct_block_02': ?>
                <div class='box_col'>
                    <div class="col_left"><?php the_sub_field('construct_block_02_left'); ?></div>
                    <div class="col_right"><?php the_sub_field('construct_block_02_right'); ?></div>
                </div>
            <?php break;
        endswitch;
    endwhile;
endif;
```



# Le clone

---

Le champ clone vous permet de réutiliser les champs et groupes de champs existants.



# Le bloc Gutenberg

Les blocs ACF sont hautement personnalisables et puissamment dynamiques. Ils s'intègrent aux champs personnalisés et permet de créer des nouveau blocs pour l'éditeur Gutenberg

## 1. Enregistrer un nouveau bloc dans le functions.php

Similaire à l'enregistrement d'un type de post, la fonction `acf_register_block_type()` vous permet d'enregistrer un type de bloc personnalisé pour l'éditeur Gutenberg à partir de votre fichier `functions.php`. Cette fonction accepte un tableau de paramètres que vous pouvez utiliser pour personnaliser votre bloc.

Vous trouverez tous les paramètres disponibles sur la page de la fonction depuis la documentation :

[https://www.advancedcustomfields.com/resources/acf\\_register\\_block\\_type/](https://www.advancedcustomfields.com/resources/acf_register_block_type/)

# 1. Enregistrer un nouveau bloc dans le functions.php

```
function ecv_init_block_types() {
    if (function_exists('ecv_init_block_types')) {
        acf_register_block_type(array(
            'name'           => 'ecv_custom', // nom unique
            'title'         => __('Bloc personnalisé 1'), // nom visible du bloc
            'description'   => __('Description du bloc'), // affiché à la sélection du bloc
            'category'     => 'common', // la catégorie du bloc [ common | formatting | layout | widgets | embed ]
            'icon'          => 'book-alt', // dashicons Wordpress
            'keywords'     => array('custom', 'bloc', 'personnalisé'), // mots clés utilisés pour la recherche
            'post_types'   => array('post', 'page'), // ou afficher le bloc
            'mode'         => 'auto', // type d'affichage [ auto | preview | edit ]
            'render_template' => 'template-parts/blocks/custom/custom.php', // l'emplacement du template du bloc
            'render_callback' => 'ecv_block_render_callback', // remplace le "render_template"
            'enqueue_style' => get_template_directory_uri().'/template-parts/blocks/custom/style.css', // charge le fichier css
            'enqueue_script' => get_template_directory_uri().'/template-parts/blocks/custom/script.js', // charge le fichier js
            'enqueue_assets' => function() { // pour charger plusieurs assets
                wp_enqueue_style('block-testimonial', get_template_directory_uri().'/template-parts/blocks/custom/style.css');
                wp_enqueue_script('block-testimonial', get_template_directory_uri().'/template-parts/blocks/custom/script.js');
            },
            'supports'     => array('align' => false, /* ... */), // active / désactive certaines features
            'example'     => array( // permet de personnaliser l'aperçu à la sélection du bloc
                'attributes' => array(
                    'mode'           => 'preview',
                    'data'          => array(
                        'cb_bg_color' => '#eeeeee', // données affichées dans l'aperçu
                        'cb_text'    => 'Exemple de texte'
                    )
                )
            )
        ));
    }
}

add_action('acf/init', 'ecv_init_block_types');
```

## ● 2. Créer le groupe de champ ACF

La prochaine étape consiste à créer un groupe de champs ACF pour votre bloc. Vous pouvez utiliser tous les champs ACF dans votre bloc sans limitation

La documentation déconseille d'utiliser trop de champs complexes et de rester sur un nombre raisonnable de champs.

À partir des règles d'emplacement, utilisez la règle "Bloc" avec votre type de bloc précédemment créé.

## 2. Créer le groupe de champ ACF

Ordre	Intitulé	Nom	Type
1	Couleur du fond	cb_bg_color	Couleur
2	Image	cb_img	Image
3	Texte	cb_text	Texte

[+ Ajouter un champ](#)

---

### Emplacement ▲

<b>Règles</b> Créez une série de règles pour déterminer les écrans sur lesquels ce groupe de champs sera utilisé	Montrer ce groupe quand		
	<input type="text" value="Bloc"/>	<input type="text" value="est égal à"/>	<input type="text" value="Bloc personnalisé 1"/>
	ou		
	<input type="button" value="Ajouter une règle"/>		

### 3. Afficher le bloc dans le template

Ensuite il faudra créer le template du bloc à l'emplacement défini dans le "render\_template" de la fonction de création du bloc.

Vous pouvez utiliser toutes les fonctions ACF dans votre template ( `get_field()`, `the_field()`, `have_rows()`, ... )

### 3. Afficher le bloc dans le template défini dans le “render\_template”

```
<?php
// id utilisé pour le css personnalisable de notre bloc
$id = 'custom_'.$block['id']; // $block['id'] = id unique du bloc généré à la création de celui-ci
if (!empty($block['anchor'])) $id = $block['anchor']; // (optionnel) remplace l'id par l'ancre

// classe utilisée pour le css / js général de notre bloc
$class = 'custom_bloc_01'; // classe par défaut
if (!empty($block['className'])) $class .= ' '.$block['className']; // classe optionnel à définir dans le bloc
if (!empty($block['align'])) $class .= ' align_'.$block['align']; // (optionnel) choix de l'alignement dans les options du bloc

// charger les valeurs ACF du bloc avec une valeur par défaut
$bg_color = get_field('cb_bg_color') ? : '#eee';
$image = get_field('cb_img') ? : 123; // 123 = id de l'image par défaut à trouver dans les médias de WordPress
$text = get_field('cb_text') ? : 'Mon texte par défaut';
?>

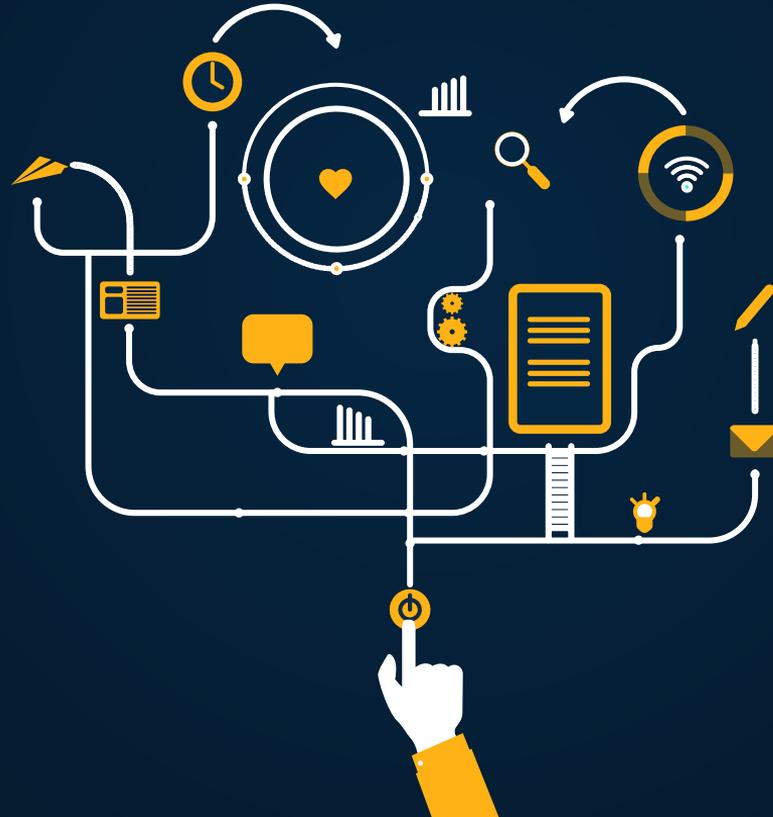
<div id="<?=$id; ?>" class="<?=$class; ?>">
  <div class="image"><?=$wp_get_attachment_image($image, 'full'); ?></div>
  <p class="text"><?=$text; ?></p>
</div>

<style type="text/css">
  <?=$ '#'.$id; ?> {
    background: <?=$background_color; ?>;
  }
</style>
```

# LA DOC ACF

<https://www.advancedcustomfields.com/resources/>

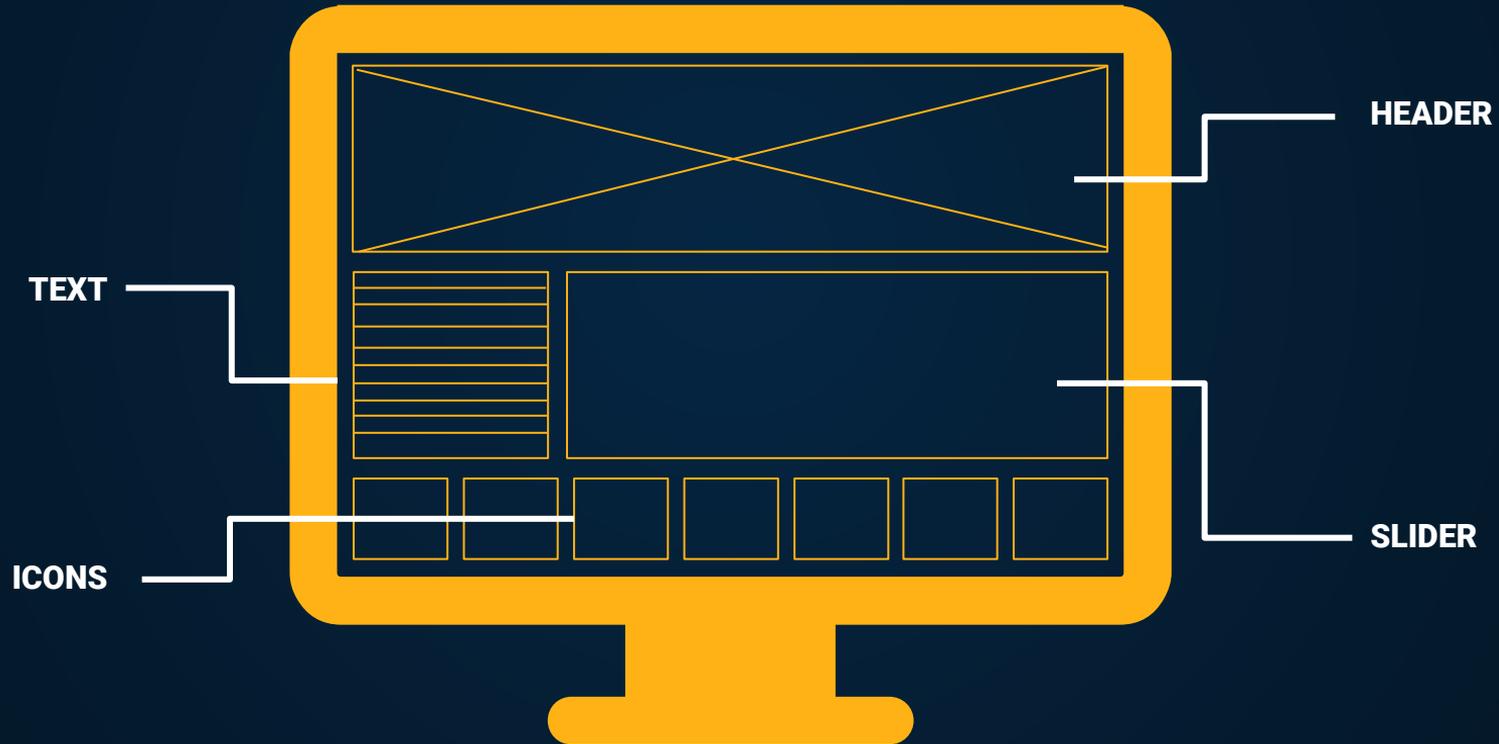
---



# EXERCICE 1

Ajouter un menu d'options au back office

---



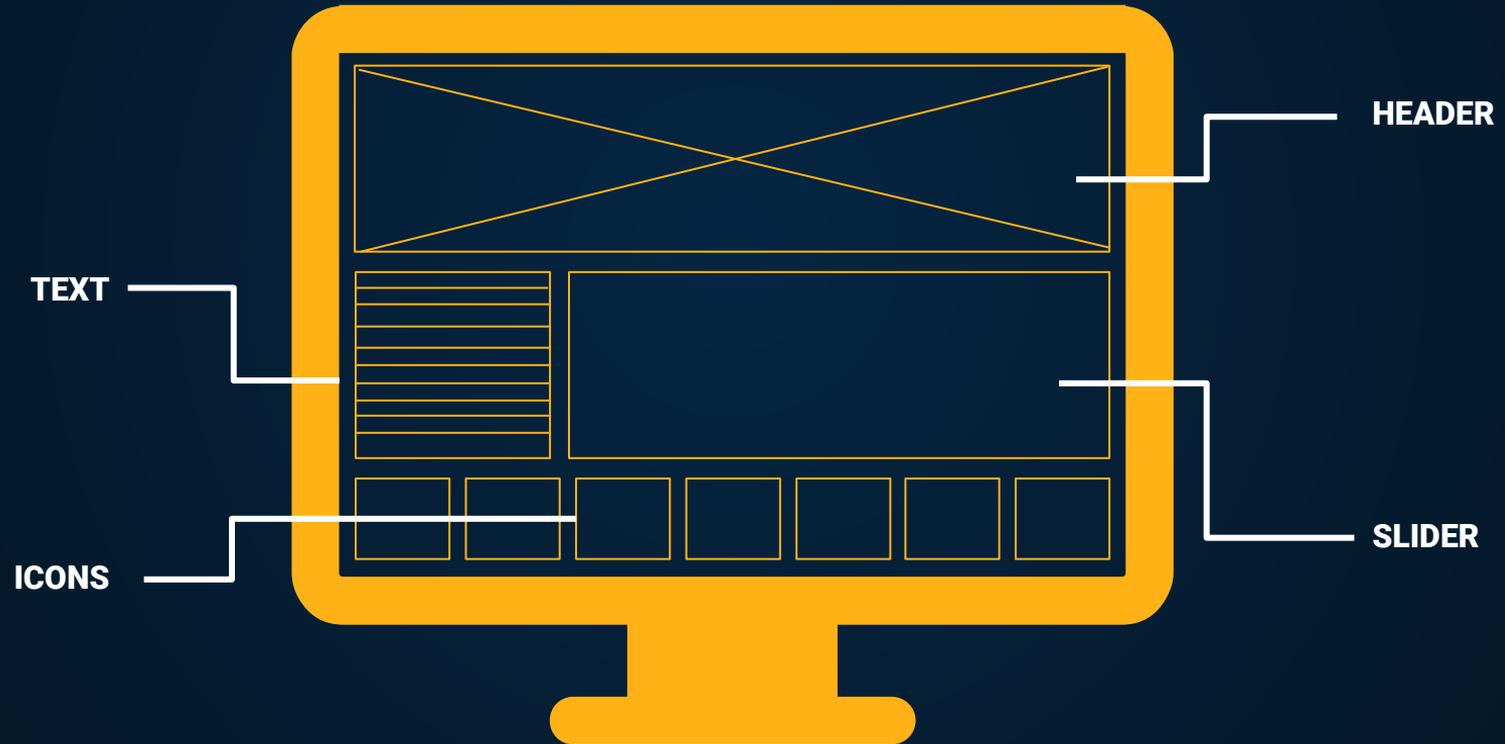
# Ajouter une page dans le back office et afficher les données dans le footer

1. Installer un nouveau Wordpress avec sa base de données => <https://fr.wordpress.org/download/>
2. Installer un nouveau thème underscores => <https://underscores.me>
3. Installer ACF Pro
4. Ajouter une page d'options dans le back office pour configurer une liste de réseaux sociaux (2 minimum) avec :
5. Ajouter un groupe de champs avec :
  - ➔ Un champ répéteur
    - ➔ Un champ image pour le logo du réseau
    - ➔ Un champ texte ou url pour le lien du réseau
6. Afficher la liste des réseaux dans le template footer.php

# EXERCICE 2

## Création d'un constructeur de page

---



# Création d'un constructeur de page sur les articles et les pages

1. Créer un modèle de page unique utilisable avec les pages et les articles :

```
/*  
* Template Name: Nom_du_modèle  
* Template Post Type: post, page  
*/
```

2. Ajouter un groupe de champ contenu flexible pour créer un constructeur de page à l'aide de plusieurs blocs :

→ Un bloc titre avec un champ texte

→ Un bloc de texte centré avec un champ de zone de texte

→ Un bloc double colonnes deux éditeurs de contenu (plus de liberté sur le choix d'un texte ou d'une image) avec une disposition en tableau

→ Une galerie d'image avec le champ galerie

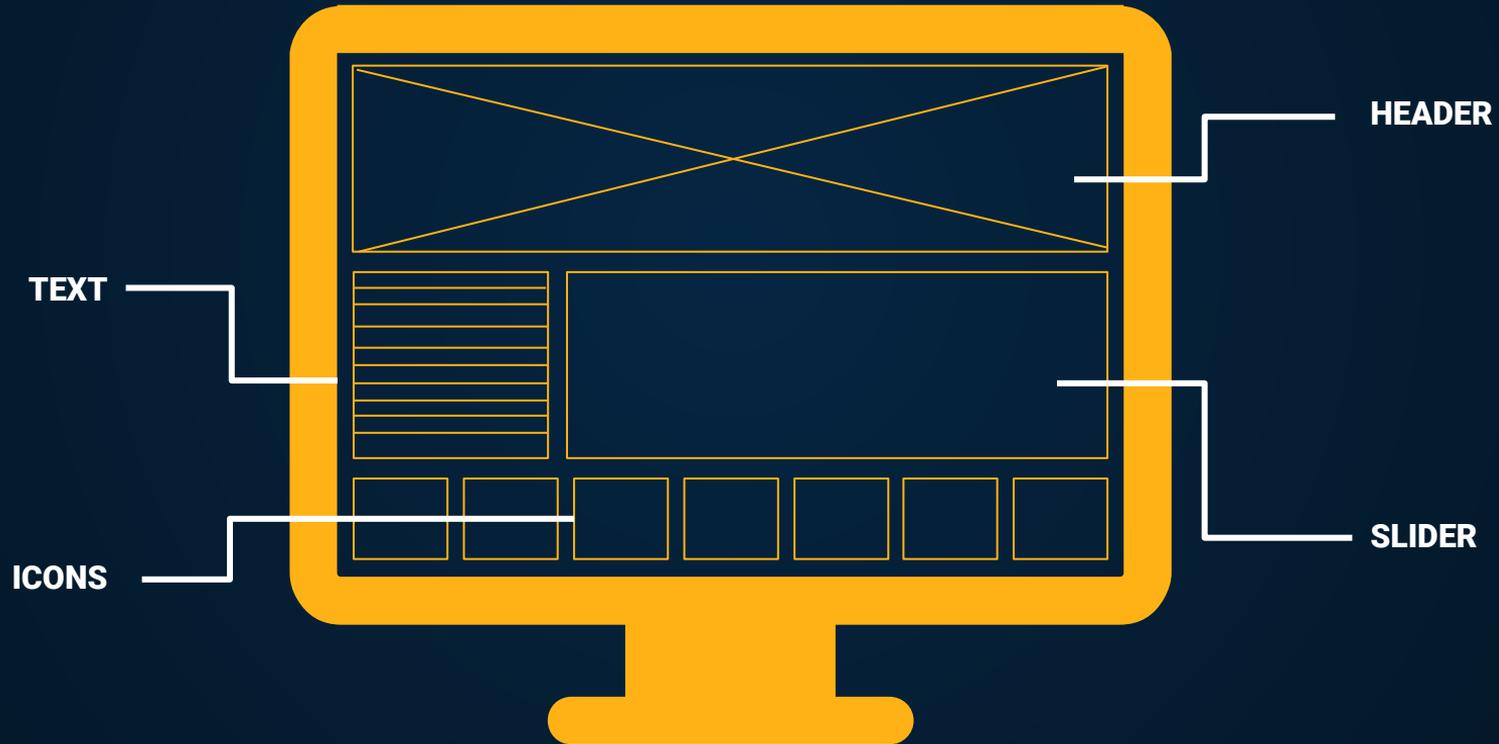
3. Enregistrer les données dans une page ou un article

4. Afficher les blocs du champ flexible dans le modèle créé (cf: 1)

# EXERCICE 3

## Création d'un bloc pour l'éditeur Gutenberg

---



## Création d'un bloc pour l'éditeur Gutenberg



*C'est bientôt le week-end !*

# Création d'un bloc pour l'éditeur Gutenberg

- Une div avec la possibilité de modifier sa couleur de fond avec un **champ ACF couleur** qui contient :
  - ➔ Une image
    - ↳ Un **champ ACF image** avec l'ID de l'image
    - ↳ Afficher l'image avec la fonction "wp\_get\_attachment\_image(ID\_image, taille)" dans le template
    - ↳ Avec une image par défaut
    - ↳ Avoir la possibilité d'aligner l'image à gauche ou à droite avec un **champ ACF bouton radio**
  - ➔ Un texte
    - ↳ Un **champ ACF texte**
    - ↳ Avec un texte par défaut
    - ↳ Avoir la possibilité d'aligner le texte à gauche, centré ou à droite avec un **champ ACF bouton radio**
    - ↳ Avoir la possibilité de modifier la couleur du texte avec un **champ ACF couleur**
  - ➔ Une bordure
    - ↳ Pouvoir afficher ou non la bordure avec un **champ ACF vrai / faux**
    - ↳ Avoir la possibilité de modifier la couleur de la bordure avec un **champ ACF couleur**
    - ↳ Afficher la sélection de la couleur uniquement si la bordure est activée (logique conditionnelle ACF)
- Afficher les 3 sélecteurs de couleur dans un **champ ACF accordéon**
- Personnaliser l'icône du bloc en modifiant le dashicon, sa couleur et / ou la couleur du fond
- Gérer l'aperçu du bloc
- Avoir un fichier css à part pour le style non personnalisable du bloc
- **Utiliser la documentation !**

# Création d'un bloc pour l'éditeur Gutenberg

Ordre	Intitulé	Nom	Type
1	Image	cb_01_img	Image
2	Alignement de l'image	cb_01_img_align	Bouton radio
3	Texte	cb_01_text	Texte
4	Alignement du texte	cb_01_text_align	Bouton radio
5	Bordure	cb_01_border	Vrai / Faux
6	Couleurs		Accordéon
7	Couleur du fond	cb_01_bg_color	Couleur
8	Couleur du texte	cb_01_text_color	Couleur
9	Couleur de la bordure	cb_01_border_color	Couleur

# Création d'un bloc pour l'éditeur Gutenberg

The screenshot displays the WordPress Gutenberg editor interface. On the left is a dark sidebar menu with various options: 'Tableau de bord', 'Articles', 'Médias', 'Pages' (highlighted), 'Toutes les pages', 'Ajouter', 'Options', 'Commentaires', 'Apparence', 'Extensions', 'Comptes', 'Outils', 'Réglages', 'ACF', and 'Réduire le menu'. The main editor area has a top toolbar with icons for adding blocks, editing, undo, redo, help, and settings. Below the toolbar is a search bar labeled 'Rechercher'. The main content area is divided into two tabs: 'Blocs' (selected) and 'Compositions'. Under the 'Blocs' tab, there is a grid of block options under the heading 'TEXTE'. The first block, 'Bloc personnalisé 1', is highlighted with a blue border. Other visible blocks include 'Paragraphe', 'Titre', 'Liste', 'Citation', 'Code', 'Classique', 'Préformaté', and 'Citation en exergue'. On the right side of the editor, a preview window shows a post with a background image of beer being poured. A dark blue box with the text 'Weekend :)' is overlaid on the image. Below the preview, a legend identifies the highlighted block as 'Bloc personnalisé 1' with the description 'Description du bloc personnalisé'. The text 'pour choisir un bloc' is visible on the right side of the preview area.

# Création d'un bloc pour l'éditeur Gutenberg

The screenshot displays the WordPress Gutenberg editor interface. On the left is a dark sidebar with navigation options: Tableau de bord, Articles, Médias, Pages (selected), Toutes les pages, Ajouter, Options, Commentaires, Apparence, Extensions, Comptes, Outils, Réglages, ACF, and Réduire le menu. The main editor area shows a 'Home' page with a large image of a bartender pouring beer. Below the image is a dark blue banner with the text 'C'est bientôt le week-end !'. On the right, the 'Blocs' panel is open, showing the configuration for 'Bloc personnalisé 1'. The configuration includes: 'Image' (Aucune image sélectionnée, with an 'Ajouter une image' button), 'Alignement de l'image' (Gauche selected, Droite), 'Texte' (empty input field), 'Alignement du texte' (Droite selected, Gauche, Centré), 'Bordure' (Afficher la bordure checked), 'Couleurs' (background, text, and border color selection buttons), and 'Avancé' (expanded).

Document → Bloc personnalisé 1