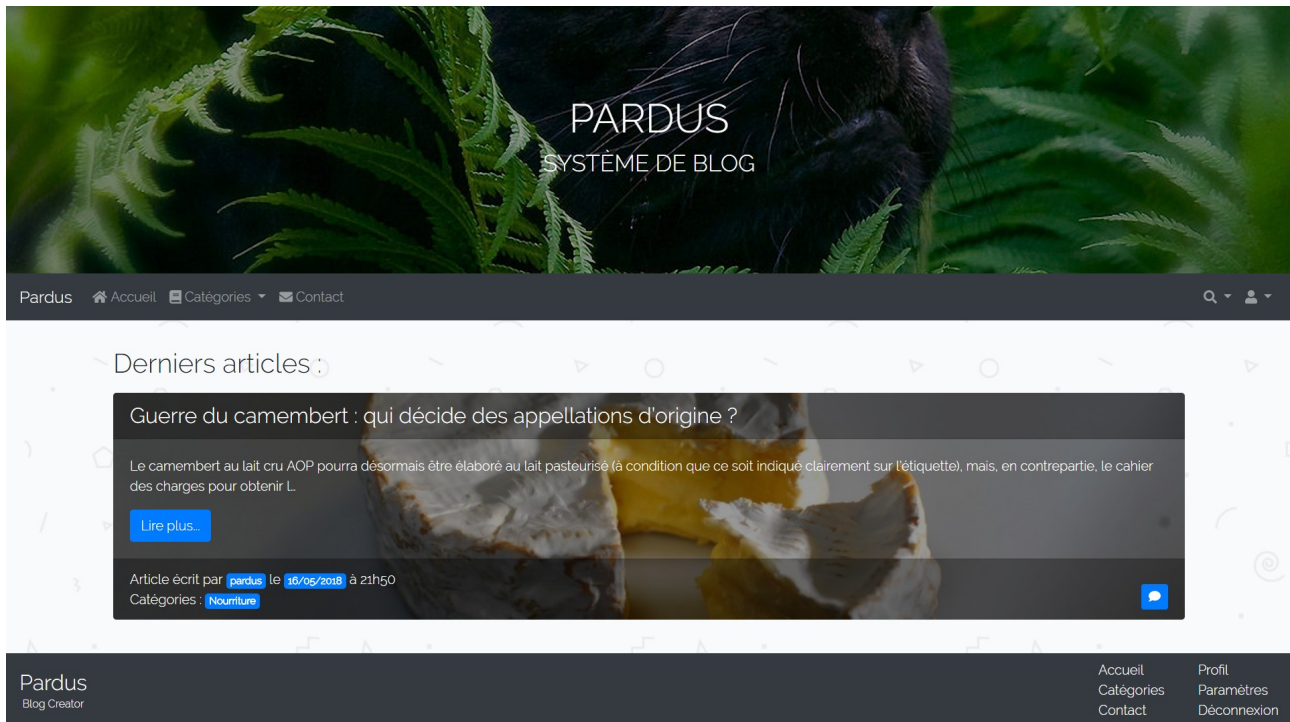


# Pardus

Système permettant la création d'un Blog



## INTRODUCTION :

Internet est désormais un moyen de communication à la portée de tous. De ce fait, de nombreux médias ou même particuliers ont décidé d'y transférer leurs activités pour en augmenter la visibilité. Néanmoins, gérer un site web nécessite dans la majorité des cas des connaissances en programmation et opter pour une compagnie spécialisée peut nécessiter un lourd budget. Pour remédier à ce problème, de nombreux développeurs ont développé des systèmes prêts à l'emploi, que l'on appelle « CMS » (« Content Management System », soit « Système de gestion de contenu » en français). Un CMS est un programme informatique utilisant une base de données et permet de gérer entièrement le contenu depuis un panneau d'administration ainsi que l'apparence d'un site web. Ces derniers peuvent s'adapter à une demande particulière, comme la gestion d'un site d'actualité ou même celle d'une boutique en ligne, mais aussi d'être modulables, permettant de flexibiliser le système vers de nouvelles demandes.

## PROJET RÉALISÉ PAR :

BONNET Guillaume et MUNOS Romain

## PRÉSENTATION DU PROJET :

Nous avons décidé de nous pencher sur un système de gestion avec une très forte demande : celle des blogs. En effet, certains systèmes de gestion très poussés peuvent tout de même effrayer certaines personnes souhaitant partager leurs centres d'intérêts, du fait que le système en question déborde de fonctionnalités inconnues pour de nouveaux utilisateurs. Nous avons alors décidé de faire un système de gestion simple, permettant la création d'un blog classique, comportant les fonctionnalités de base.

## CAHIER DES CHARGES :

Pour mener à bien ce projet, nous avons décidé de choisir comme langage de programmation Python, car c'est un langage multiplate-forme, à la fois simple d'utilisation et riche en fonctionnalité, qui peuvent être étendus grâce à

un vaste catalogue de bibliothèques, et disposant d'une grande communauté active. Pour la création de notre système, nous nous sommes servi de la bibliothèque Django, permettant la réalisation du système même, ainsi que Pillow, une bibliothèque pour la gestion des images.

Django a la particularité de permettre la séparation de chaque projet en plusieurs applications, visant chacune une ou plusieurs fonctionnalités, permettant de rendre indépendante chaque application et de pouvoir les ré-utiliser dans d'autres projets.

Pour la mise en page du site, nous avons utilisé le langage de balisage HTML, ainsi qu'en complément le langage informatique CSS. Pour le style, nous nous sommes basé sur un framework nommé Bootstrap. Ce dernier permet de styliser facilement et dispose de nombreuses fonctionnalités. Nous avons tout de même écrit une feuille de style personnalisée, reprenant certains éléments stylisés par Bootstrap pour les adapter à nos envies/objectifs.

Font Awesome, lui, est là pour générer des icons vectoriels. Cela nous a permis de rendre plus explicite certaines actions à l'aide de ces pictogrammes.

## PAGES :

- TOUTES LES PAGES :
  - En-tête :
    - Titre du blog
    - Slogan du blog (facultatif)
    - Titre de la page (hors Accueil)
  - Barre de navigation :
    - Liens
    - Barre de recherche
  - Pied de page :
    - Titre du blog
    - Slogan du blog
    - Liens
- VISITEUR :
  - Accueil :
    - Prévisualisation d'articles
  - Catégorie :
    - Liste des catégories
  - Article :
    - Contenu de l'article
    - Détails sur l'article : Auteur, date de publication, catégories
    - Espace commentaire
  - Contact :
    - Formulaire de contact
- UTILISATEUR :
  - Espace membre :
    - Espace d'inscription et de connexion
  - Profil :
    - Onglet «Profil» : Informations sur l'utilisateur
    - Onglet « Articles » : Articles publiés par l'utilisateur
    - Onglet «Paramètres» : Modification des informations de l'utilisateur
    - + pages accessibles à un Visiteur

- ADMINISTRATEUR :
  - Tableau de bord :
    - Gestion des articles : Création, modification ou suppression d'articles
    - Gestion des commentaires : Visualisation ou suppression de commentaires
    - Gestion des utilisateurs : Création, modification ou suppression d'utilisateurs
    - Paramètres du site
    - + pages accessibles à un Utilisateur

#### LANGAGES (AINSI QUE BIBLIOTHÈQUES ET FRAMEWORKS) UTILISÉS :

- HTML 5
- CSS 3
  - Bootstrap (version 4.1.1) (compatibilité version : 4.x.x)
  - Pardus (feuille de style propre au projet)
- JavaScript
  - JQuery (version : 3.3.1) (compatibilité version : 1.9.1 – 3.x.x) (dépendance de Bootstrap et Pardus)
  - Popper (version : 1.14.3) (compatibilité version : 1.13.x) (dépendance de Bootstrap)
  - Bootstrap (version 4.1.1) (compatibilité version : 4.x.x)
  - Font Awesome (version : 5.0.13) (compatibilité version : 4.x.x – 5.x.x)
  - Pardus (script propre au projet)
- Python
  - Django (version : 2.0.5) (compatibilité version : 2.x.x)
  - Pillow (version : 5.1.1) (compatibilité version : 5.x.x)

#### RÉPARTITION DES TÂCHES :

NOMS	TÂCHES	HEURES
BONNET Guillaume	Modèle HTML & CSS de la page d'accueil	4H
	Modèle HTML & CSS de la page des catégories	3H
	Modèle HTML & CSS de la page du filtre d'articles par catégorie	1H
	Modèle HTML & CSS de la page du filtre d'articles par date	1H
	Modèle HTML & CSS de la page du filtre d'articles par recherche	1H
	Modèle HTML & CSS de la page de profil utilisateur	3H
	JavaScript propre au projet	2H
	Gestion des modèles Django	5H
	Gestion de l'administration Django (avec MUNOS Romain)	2H
	Distribution des URLs Django (avec MUNOS Romain)	1H
	Gestion des vues Django (avec MUNOS Romain)	10H
MUNOS Romain	Modèle HTML & CSS de la page d'article	3H

	Modèle HTML & CSS de la page de contact	3H
	Modèle HTML & CSS de la page de connexion	2H
	Modèle HTML & CSS de la page d'inscription	2H
	Modèle HTML & CSS de la page des paramètres utilisateurs	2H
	Gestion des formulaires Django	6H
	Gestion de l'administration Django (avec BONNET Guillaume)	2H
	Distribution des URLs Django (avec BONNET Guillaume)	1H
	Gestion des vues Django (avec BONNET Guillaume)	10H

## PRÉSENTATION DU PROGRAMME :

Pour créer ce programme, nous avons exécuté les commandes de base de création d'un projet Django, à savoir, pour créer le projet, nous avons exécuté cette commande :

```
django-admin startproject Pardus
```

Par la suite, nous avons créé une application « core » pour notre projet, qui contient la gestion du système entier (hors administration), à l'aide de la commande suivante :

```
python manage.py startapp core
```

Voici l'architecture obtenue à la suite de l'exécution de ces deux commandes :

```
core/
Pardus/
templates/
manage.py
```

De plus, certains fichiers sont générés dès la création du projet et de l'application. Pour le projet « Pardus » :

```
Pardus/
    settings.py - Paramètres du projet [1]
    urls.py - URLs du projet [2]
    wsgi.py
```

Pour l'application « core » :

```
core/
    admin.py - Gestion de l'application dans django-admin [3]
    apps.py
    forms.py - Formulaires de l'application [4]
    models.py - Modèles de l'application [5]
    tests.py
    urls.py - URLs de l'application [6]
    views.py - Vues de l'application [7]
```

[1] : Ce fichier permet de définir les paramètres du projet. C'est ici qu'on définit, par exemple, les applications liées aux projets, la langue du projet, le chemin vers les fichiers statiques et fichiers envoyés par les utilisateurs.

[2] : Ce fichier permet la distribution des URLs du projet. On définit notamment l'URL vers l'application « core », ou encore l'URL vers l'administration prête à l'emploi et délivrée par défaut par Django (django-admin)

[3] : Ce fichier permet de gérer l'intégration de notre application « core » dans l'application « django-admin ». On gère donc comment s'affiche la prévisualisation de chaque modèle dans l'administration ainsi la mise en forme de l'édition.

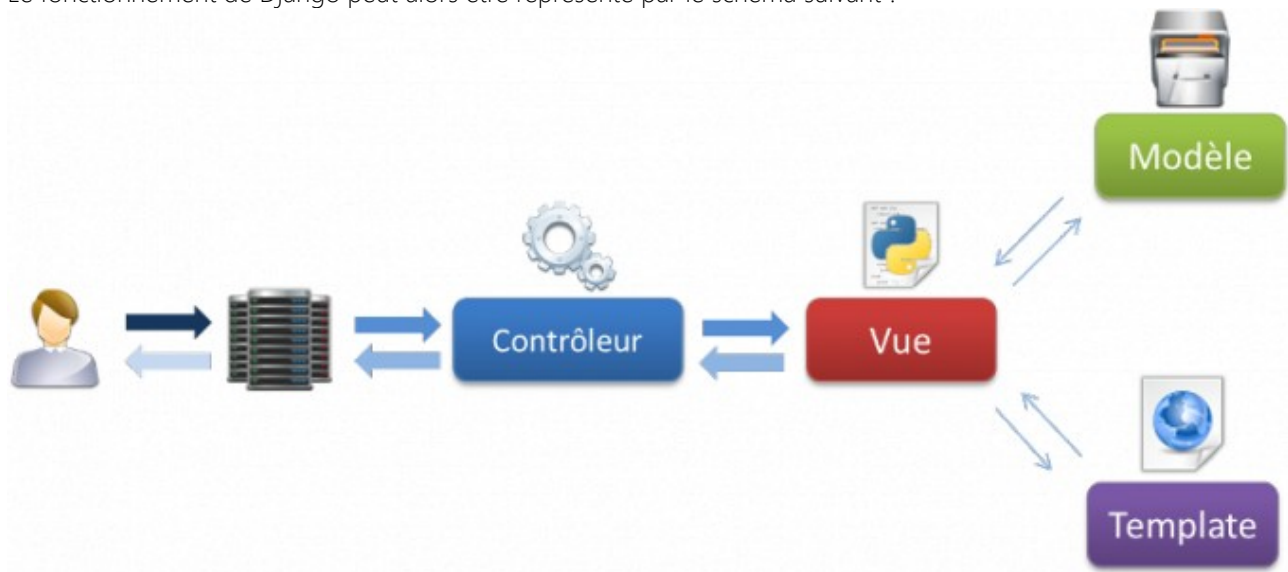
[4] : Ce fichier permet de gérer les formulaires de l'application. C'est ici que nous avons défini le formulaire pour la page de contact, la page des paramètres, la page de connexion et de déconnexion, ainsi que l'insertion des commentaires.

[5] : Ce fichier permet de définir les modèles. C'est eux qui définiront les propriétés de chaque donnée. Nous avons créé un modèle pour les Articles, les Catégories ainsi que les Commentaires.

[6] : Ce fichier s'occupe de la distribution des URLs. On lie une vue à une URL et l'on nomme l'URL pour pouvoir la réutiliser par la suite. À savoir : « path() » est la version simplifiée de « re\_path() ». « re\_path() » permet d'avoir des URLs plus complexe, gérant notamment une URL qui comporte un paramètre facultatif (le paramètre « id » pour l'URL « profile »)

[7] : Ce fichier est le centre même de l'application. C'est ici que l'on définit les vues, qui se chargeront de faire le lien entre les modèles HTML, les modèles Django ainsi que les formulaires. Les vues permettent donc de générer et de renvoyer une page HTML en fonction de la demande du visiteur.

Le fonctionnement de Django peut alors être représenté par le schéma suivant :



Source : OpenClassrooms (<https://openclassrooms.com/>)

## PROBLÈMES RENCONTRÉS LORS DE LA CRÉATION DU PROGRAMME :

- HTML/CSS :
  - 01/03/18 :
    - Problème de mise en page avec l'en-tête de la page d'accueil (corps de l'en-tête non-centré verticalement)
    - Réglé : Utilisation d'une autre propriété CSS
  - 22/03/18 :
    - Problème au niveau du pied de page : ce dernier ne se fixait pas en bas de la page si la page ne prenait pas toute la hauteur de l'écran
    - Réglé : Utilisation de JavaScript pour fixer le pied de page si la page ne prenait pas toute la hauteur de l'écran
- JavaScript :
  - 22/03/18 :
    - Problème au niveau du pied de page : Restait en « position: fixed » si la page était redimensionnée

→ Régulé : Adaptation des propriétés appliqués au pied de page en fonction de la taille de la page lorsque la fenêtre du navigateur est redimensionnée

- 14/03/18 :
  - Problème au niveau de JavaScript sur la page « Profil » : le script pour fixer le pied de page en bas ne fonctionne pas
  - Non réglé
- Python (Django) :
  - 04/04/18 :
    - Problème : Accès à la page « Paramètres » même si l'utilisateur n'est pas connecté
    - Régulé : Vérifie que l'utilisateur est bien connecté pour accéder à la page, sinon le renvoi vers la page de connexion (LoginRequiredMixin)

## EXÉCUTION DU PROGRAMME :

Pour exécuter ce programme, il est nécessaire de vérifier que les bibliothèques Python suivantes sont bien installées et mises à jour :

- Django, version  $\geq 2.0.0$
- Pillow, version  $\geq 5.0.0$

NOTE : `%Python%` représentera le chemin vers l'exécutable « python.exe » comportant les bibliothèques ci-dessus.

À exécuter dans l'ordre suivant :

`%Python% manage.py makemigrations`

*Inscrit dans un fichier tous les modèles existant ainsi que leurs propriétés*

`%Python% manage.py migrate`

*Crée une base de donnée SQLite 3 avec une table pour chaque modèle ainsi que leurs propriétés*

`%Python% manage.py createsuperuser`

*Crée un super-utilisateur qui aura tous les pouvoirs sur le site (accès à l'administration, création, modification, suppression, ...)*

`%Python% manage.py runserver 8000`

*Lance le serveur sur le port 8000. L'accès au site sera donc le suivant : <http://127.0.0.1:8000>*

Le résultat dans la console devrait ressembler à ci-dessous :

```
Migrations for 'core':
core\migrations\0001_initial.py
- Create model Article
- Create model Category
- Create model Comment
- Create model ContactMessage
- Add field categories to article
```

```
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, core, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying core.0001_initial... OK
  Applying sessions.0001_initial... OK
```

```
Username (leave blank to use 'guil.bonnet'): pardus
Email address: email@email.com
Password:
Password (again):
Superuser created successfully.
```

```
Performing system checks...

System check identified no issues (0 silenced).
May 17, 2018 - 09:21:51
Django version 2.0.5, using settings 'Pardus.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

## CE QUE NOUS A APPORTÉ CE PROJET :

Lors de son développement, notre projet nous a apporté de nombreuses connaissances personnelles. En effet, celui-ci nous a demandé de nombreuses réflexions afin d'aboutir à nos objectifs et arriver au programme actuel. De plus, cela nous a demandé de l'investissement et des recherches afin de répondre au cahier des charges que nous nous sommes imposés pour la réalisation de ce projet. En effet, nous avons eu besoin de perfectionner nos connaissances en ce qui concerne la création de site web, car un blog est avant tout un site web, et il nous fallait un site « modèle » servant d'exemple pour la création de ce dernier.

Cependant, ce qui nous a principalement posé problème concerne la création du système en lui-même. Cela nous a pris beaucoup de temps, notamment car nous n'avions jamais programmé en Python auparavant et que nous possédions ainsi peu de connaissances.

Ce projet nous a donc été avant tout un apport en nouvelles connaissances, et celles-ci nous sera certainement utile dans l'avenir que ce soit pour nos futurs études ou bien pour notre futur profession.

De plus, notre projet est fonctionnel, certes, mais est loin d'être perfectionné. De nombreuses modifications et améliorations peuvent toujours y être apportés. Souhaitant tout deux poursuivre nos études dans le domaine de l'informatique suite à l'obtention de notre baccalauréat, ce projet pourrait donc être poursuivit jusqu'à, pourquoi pas, la diffusion en ligne de celui-ci et donc de permettre, comme expliqué précédemment, d'aider et de simplifier la création de futurs blogs.

On peut également ajouter que ce projet nous a tout deux permis d'expérimenter à petite échelle ce qu'est le travail dans le monde de l'informatique. Bien que ce projet ai été bâti sur notre passion pour l'informatique, nous avons pu apprécier une forme professionnelle dans le travail en groupe que nous avons effectué lors de ce projet.