

Fichiers statiques

Les vues et les modèles d'authentification fonctionnent, mais ils sont très simples pour le moment. Un peu de [CSS](#) peut être ajouté pour ajouter du style à la mise en page HTML que vous avez construite. Le style ne changera pas, il s'agit donc d'un fichier *statique* plutôt que d'un modèle.

Flask ajoute automatiquement une vue `static` qui prend un chemin relatif au répertoire `flaskr/static` et le sert. Le modèle `base.html` a déjà un lien vers le fichier `style.css` :

```
{{ url_for('static', filename='style.css') }}
```

Outre les CSS, d'autres types de fichiers statiques peuvent être des fichiers contenant des fonctions JavaScript, ou une image de logo. Ils sont tous placés dans le répertoire `flaskr/static` et référencés avec `url_for('static', filename='...')`.

Ce tutoriel n'est pas axé sur l'écriture de CSS, vous pouvez donc simplement copier ce qui suit dans le fichier `flaskr/static/style.css` :

`flaskr/static/style.css`

```
html { font-family: sans-serif; background: #eee; padding: 1rem; }
body { max-width: 960px; margin: 0 auto; background: white; }
h1 { font-family: serif; color: #377ba8; margin: 1rem 0; }
a { color: #377ba8; }
hr { border: none; border-top: 1px solid lightgray; }
nav { background: lightgray; display: flex; align-items: center; padding: 0 1rem; }
nav h1 { flex: auto; margin: 0; }
nav h1 a { text-decoration: none; padding: 0.25rem 0.5rem; }
nav ul { display: flex; list-style: none; margin: 0; padding: 0; }
nav ul li a, nav ul li span, header .action { display: block; padding: 0 1rem; }
nav ul li a { padding: 0 1rem; }
nav ul li a { border-bottom: 1px solid lightgray; display: flex; align-items: center; }
nav ul li a h1 { flex: auto; margin: 1rem 0 0.25rem 0; }
nav ul li a h1 { margin: 1rem 0; padding: 1rem; background: #cae6f6; border: 1px solid #ccc; }
nav ul li a h1 { display: flex; align-items: flex-end; font-size: 0.85em; }
nav ul li a h1 > div:first-of-type { flex: auto; }
nav ul li a h1 { font-size: 1.5em; margin-bottom: 0; }
nav ul li a .about { color: slategray; font-style: italic; }
nav ul li a .body { white-space: pre-line; }
nav ul li a:last-child { margin-bottom: 0; }
nav ul li a form { margin: 1em 0; display: flex; flex-direction: column; }
nav ul li a label { font-weight: bold; margin-bottom: 0.5em; }
nav ul li a input, nav ul li a textarea { margin-bottom: 1em; }
nav ul li a textarea { min-height: 12em; resize: vertical; }
input.danger { color: #cc2f2e; }
input[type=submit] { align-self: start; min-width: 10em; }
```

 v: latest ▾

Vous pouvez trouver une version moins compacte de `style.css` dans le [code d'exemple](#).

Allez sur <http://127.0.0.1:5000/auth/login> et la page devrait ressembler à la capture d'écran ci-dessous.



The screenshot shows a web browser window displaying the Flaskr application. The header is a light gray bar with the word 'Flaskr' in a large, bold, blue font on the left, and two links, 'Register' and 'Log In', in a smaller blue font on the right. Below the header, the main content area has a white background. It starts with the title 'Log In' in a large, bold, blue font. Underneath the title is a horizontal line. Then, there are two input fields: the first is labeled 'Username' in a bold black font, and the second is labeled 'Password' in a bold black font. Both labels are positioned to the left of their respective input boxes. At the bottom of the form is a button labeled 'Log In' in a gray font.

Vous pouvez en savoir plus sur les CSS en consultant [Mozilla's documentation](#). Si vous modifiez un fichier statique, rafraîchissez la page du navigateur. Si la modification ne s'affiche pas, essayez de vider le cache de votre navigateur.

Continuer vers [Blueprint du blog](#).