

# Visiter le Paris de 1970

Jean Thomas

14 novembre 2025

# Contenu

- 1 La genèse
- 2 L'exploration
- 3 La réalisation
  - Récupération des photos
  - Affichage des photos
  - Mise en ligne

# "C'était Paris en 1970"



# Un concours de photographie amateur

- Organisé en 1970 par Ville de Paris et la FNAC
- But : photographier tout le territoire parisien
- Découpage en 1755 carrés de 250 mètres de côté
- Chaque candidat se voit attribuer un carré



# Mise en ligne

- Toutes ces photos ont été données à la Bibliothèque historique de la Ville de Paris
- Elles ont été classées et numérisées
- Elles sont mises à disposition sur leur site web depuis 2024...
- ... mais c'est atroce à utiliser
- Démo : <https://bibliotheques-specialisees.paris.fr/>
- Très frustrant de ne pas pouvoir partager la découverte de ce fonds !

# Contenu

- 1 La genèse
- 2 L'exploration
- 3 La réalisation
  - Récupération des photos
  - Affichage des photos
  - Mise en ligne

# Le déclencheur


Vacances scolaires de la Toussaint :

- 1 enfant chez ses grands-parents pendant 2 semaines
- + 1 enfant qui dort tôt le soir et fait des siestes
- = plus de temps (et d'énergie) pour bidouiller !

## Étape 0 : la licence

- <https://bibliotheques-specialisees.paris.fr/term-of-use>
- *Tous les éléments de ce site, y compris les documents téléchargeables, sont libres de droit. À l'exception de l'iconographie, la reproduction des pages de ce site est autorisée à la condition d'y mentionner la source.*

# Étape 1 : les données brutes

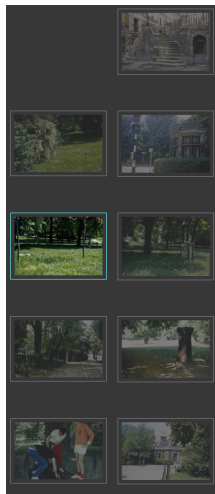
 Fonds C'était Paris en 1970. Photographies : carrés 341 à 522

 TÉLÉCHARGER

- Un fichier XML par partie du fonds
- Contient une hiérarchie par page de plan, puis par carré, et enfin par candidat
- Pour chaque candidat, on a une URL vers sa première photographie

## Étape 2 : les photos

- Chargement d'une page de la visionneuse pour un candidat
- Vignettes de toutes les photographies du candidat affichées
- En creusant un peu, on peut récupérer l'URL de chacune des vignettes
- Les vignettes ont une taille suffisante pour être affichées



## Étape 3 : le projet

Création d'un site web avec :

- Une page d'index avec toutes les pages du plan affichées
- Une page par page du plan
- Une page par carré :
  - ▶ Affichage de toutes les vignettes des photographies
  - ▶ Chaque vignette est un lien vers la visionneuse de la Bibliothèque de Paris

# Contenu
















1 La genèse

2 L'exploration

**3 La réalisation**

- Récupération des photos
- Affichage des photos
- Mise en ligne

# Le scrapping

- ▼  Fonds C'était Paris en 1970. Photographies : carrés 1 à 155
  - ▼  Carrés 1 à 9. 17 e arrondissement
    -  de Paris quadrillé pour le concours : feuille 1, carrés 1 à 9
    - ^  Carré 2
    - ▼  Carré 3
      -  Photographe non identifié. Diapositives
      -  Pauwelyn, Edgard (photographe). Epreuves
      -  Auclair (photographe). Epreuves
    - ^  Carré 4
    - ^  Carré 5
    - ^  Carré 6
    - ^  Carré 7
    - ^  Carré 8
    - ^  Carré 9
    - ^  Carrés 10 à 28. 17 e arrondissement

- Téléchargement manuel des 10 fichiers XML
- Parsing en Python des fichiers XML
- Itération sur :
  - ▶ Les pages du plan
  - ▶ Les carrés de chaque page
  - ▶ Les candidats de chaque carré
- Construction des URLs des vignettes de chaque candidat

# Stockage des données

```
jean@guido:/data/paris1970| tree pictures/carres/9
pictures/carres/9
├── b1637848
│   ├── BHVP_PH_2DP_012_0009_01_001.jpg
│   └── candidate.json
├── b1637851
│   ├── BHVP_PH_2DP_012_0009_03_001.jpg
│   └── candidate.json
├── b1637852
│   ├── BHVP_PH_2DP_012_0009_02_001.jpg
│   └── candidate.json
└── square.json
```

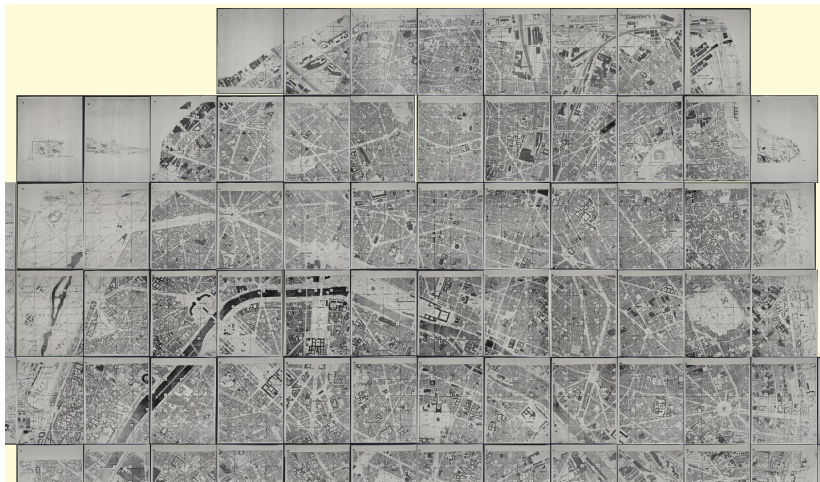
- Ma base de données : l'arborescence des fichiers
- Téléchargement de chaque page du plan
- Téléchargement de toutes les vignettes de chaque candidat
- Stockage des données de candidats et de carré dans un \*.json

# Création du HTML

- Uniquement du HTML et du CSS, pas de JS ou de framework compliqué
- Une page d'index avec le plan global
- Templates Jinja2 :
  - ▶ Une page par plan
  - ▶ Une page par carré

# Carte interactive

Construire un plan de Paris cliquable à partir des pages de plan :



# Carte interactive

Liens hypertexte et grille CSS :

```
<div class="fullmap">
  <div></div>
  <a href="...">
    
  </a>
  <a href="...">
    
  </a>
  ...
</div>
```

```
.fullmap {
  display: grid;
  grid-template-columns: auto auto [...];
}
```



# Menu

Balise HTML `<details>` et un peu de CSS :

```
<details>
  <summary>= Menu</summary>
  <p>
    <a href="...">...</a>
    ...
  </p>
</details>
```

```
header summary::marker {
  content: "";
}
header p {
  display: flex;
  flex-direction: column;
}
header a {
  font-size: 25px;
  padding: 5px 10px;
  text-decoration: none;
}
```

# OpenGraph

Une jolie prévisualisation quand le lien est partagé :



# OpenGraph

- Basé sur le protocole OpenGraph : <https://ogp.me/>
- De bêtes balises HTML meta dans le header :

```
<head>
```

```
  <meta name="description" content="Photographies numérisées du fonds C'était Par
```

```
  <meta property="og:title" content="C'était Paris en 1970" />
```

```
  <meta property="og:type" content="website" />
```

```
  <meta property="og:url" content="https://paris1970.jeantho.eu/" />
```

```
  <meta property="og:image" content="https://paris1970.jeantho.eu/paris1970.jpg"
```

```
</head>
```

# Compression

Compression avec Brotli des fichiers servis :

- `apt install brotli libnginx-mod-http-brotli-static`
- `brotli --best index.html`
- Configuration nginx :

```
server {  
    brotli_static on;  
    brotli_types text/plain text/css image/jpeg image/jpg;  
}
```

# Compression

Très efficace sur les pages HTML :

```
$ curl -s -w "%{size_download}\n" -o /dev/null "https://paris1970.jeantho.eu/index.html"
27176
$ curl -s -w "%{size_download}\n" -o /dev/null \
-H "Accept-Encoding: br" "https://paris1970.jeantho.eu/index.html"
1492
```

Mais pas trop sur des images déjà compressées :

```
$ curl -s -w "%{size_download}\n" -o /dev/null "https://paris1970.jeantho.eu/paris1970.jpg"
35637
$ curl -s -w "%{size_download}\n" -o /dev/null \
-H "Accept-Encoding: br" "https://paris1970.jeantho.eu/paris1970.jpg"
35227
```

Peut-être que d'autres algorithmes de compression auraient été plus adaptés ?

# Démo

<https://paris1970.jeantho.eu>