

### **Le maxitel ?**

Posté par [devnewton \(site Web personnel\)](#) le 26/11/20 à 08:31. En réponse au journal [Gemini et Solid, deux alternatives au Web \(qu'il faut qu'on m'explique\)](#). Évalué à 5.

Une approche complémentaire serait d'avoir:

- un routeur pour les sites (web codé proprement, gemini, gopher...);
- un routeur pour les applications.

Pour ce dernier, je propose de virer HTML et CSS :-)

On garderait:

- https comme protocole pour récupérer des ressources et appeler des API REST;
- javascript pour écrire les applications ;
- un minimum d'API js.

Lorsque l'on se connecte sur <https://monappli.com>, le routeur:

- télécharge <https://monappli.com/app.js> ;
- crée un onglet et un contexte dépendant de l'utilisateur et de son matériel ;
- exécute app.js dans ce contexte.

De quoi est composé le contexte? Cela dépend:

- si le routeur est en mode console, c'est un terminal ;
- si le routeur est graphique, c'est un canvas + des API pour accéder au clavier/souris/pad/tactile...;
- si l'utilisateur est aveugle, c'est une plage braille ;
- si le routeur est un téléphone à cadran ou un assistant personnel, c'est une API pour faire de la synthèse vocale.

Le script app.js a de quoi interroger le contexte pour savoir quel rendu et quelles interactions il peut faire.

Ainsi on obtient un "maxitel" (simple comme un minitel mais avec presque tout côté client) facile à implémenter (un interpréteur et quelques API simples, mais pas des normes gigantesques et imbitables comme HTML+CSS).

--

Le post ci-dessus est une grosse connerie, ne le lisez pas sérieusement.