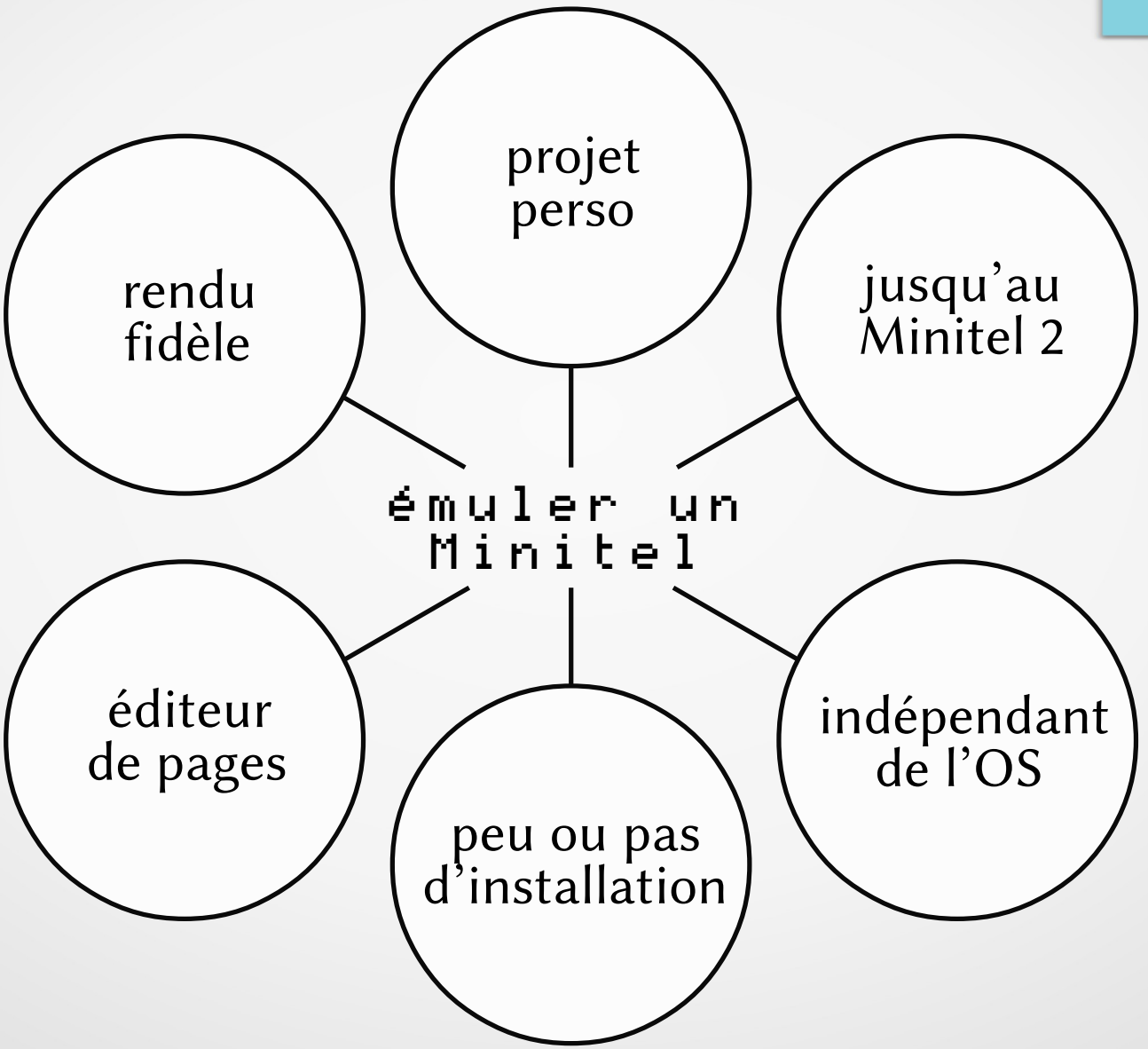
F

Zigazou

Emuler un Minitel

Frédéric
BISSON

Mon projet


projet perso
rendu fidèle
jusqu'au Minitel 2
émuler un Minitel
éditeur de pages
peu ou pas d'installation
indépendant de l'OS

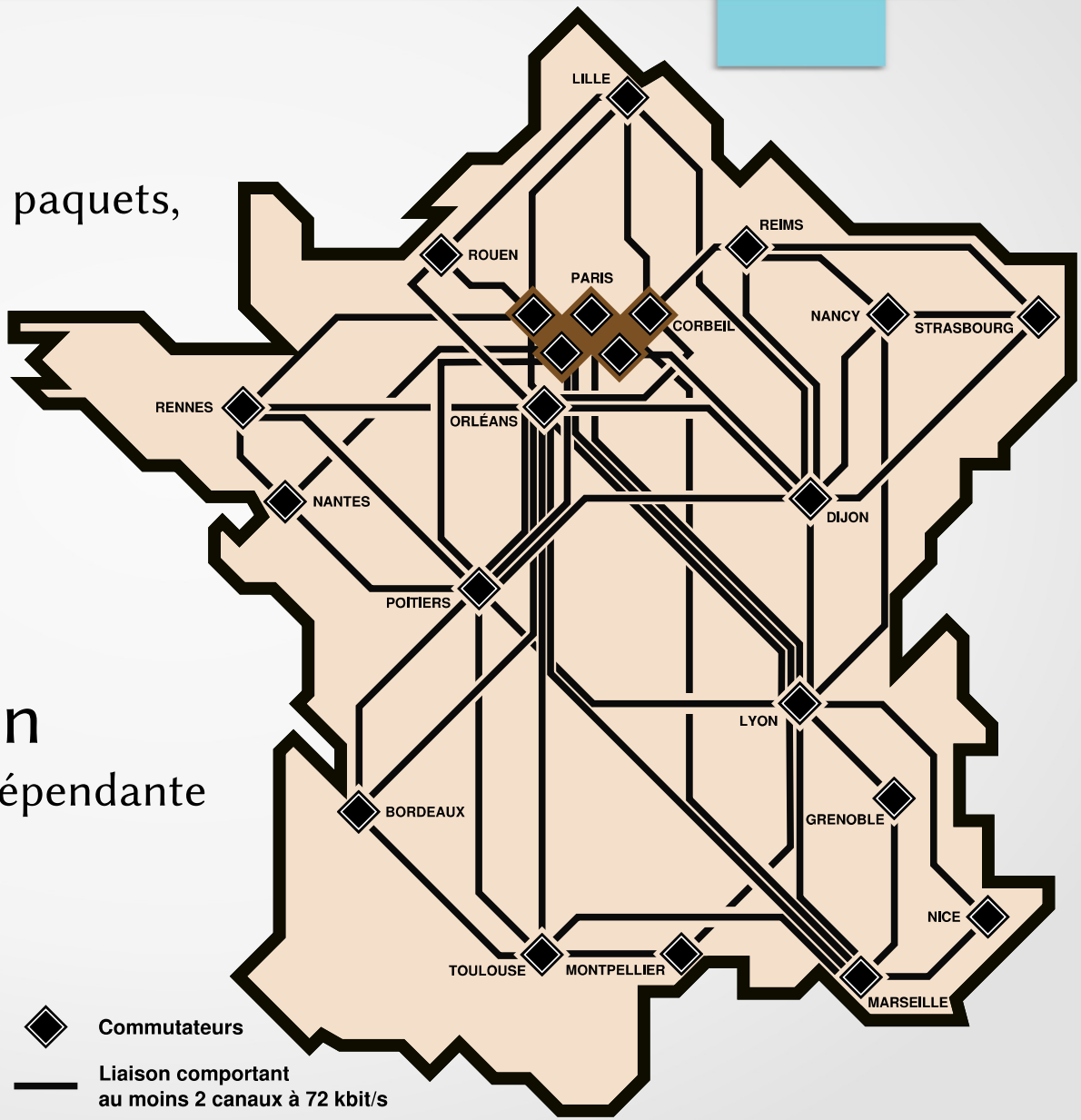
Petite histoire

- Retard téléphonique 1960-1970
  Plusieurs mois pour obtenir une ligne, sketch du 22 à Asnières
- Rattrapage téléphonique
  PTT 1er investisseur en 1972
  Réseau entièrement automatisé en 1978
- Après-rattrapage
  Transpac, Minitel, monétique

Fernand Raynaud, Le 22 à Asnières – 1966

Transpac / X.25

- Réseau public
  point à point, commutation de paquets, circuits virtuels
- Multi-débit
  de 50 à 64 000 bit/s
- Hétérogène
  accès direct, modem, telex
- Nouvelle tarification
  indépendante de la distance, dépendante du débit et du temps
- Ouvert en 1978



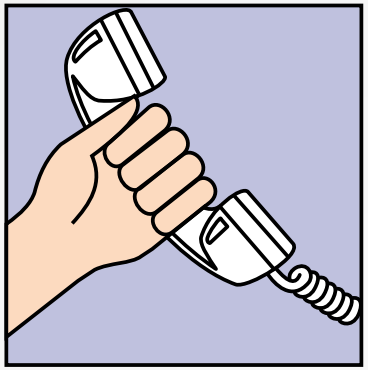
En 1981...


F
Estimations
CII HONEYWELL BULL
PRESIDENTIELLE
1981
A2
E1

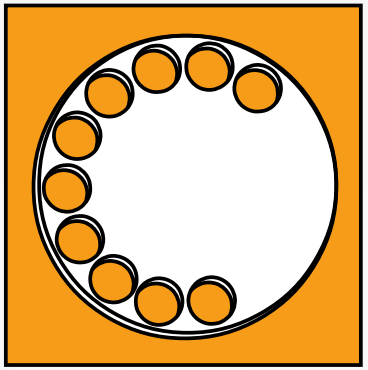
. . . élections

Minitel?

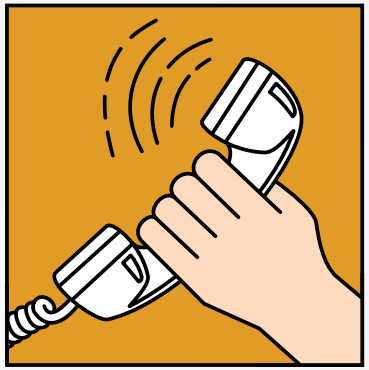
Mettez en marche votre MINITEL.



1. Décrochez votre combiné.

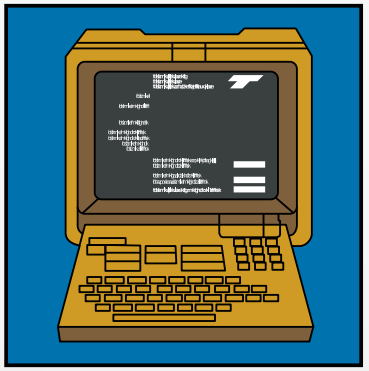


2. Composez le n° d'appel.

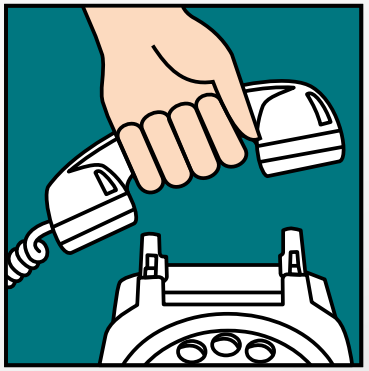


3. Attendez le signal continu.

4. Appuyez sur la touche.

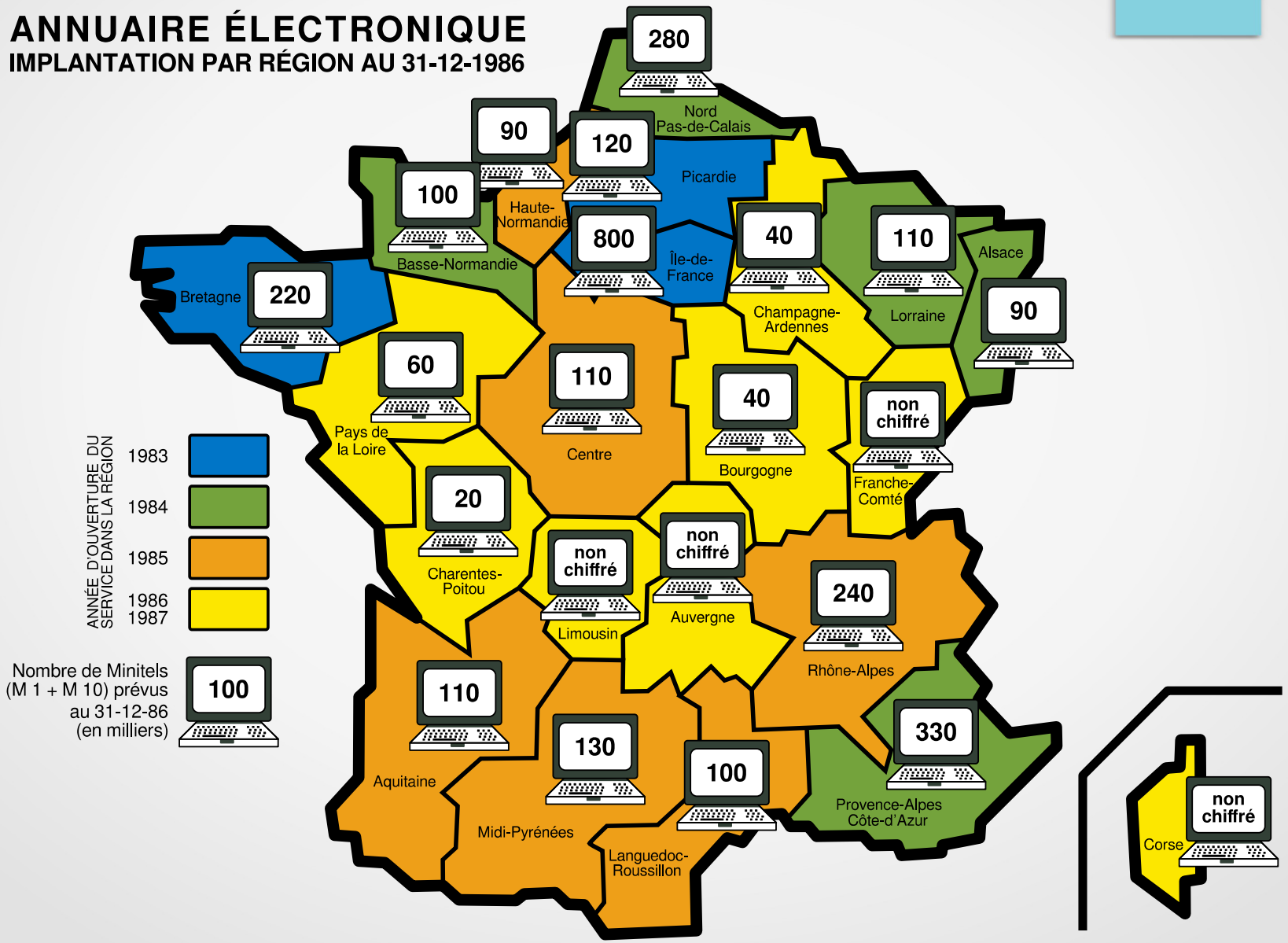


5. Attendez la page.



6. Raccrochez votre combiné.

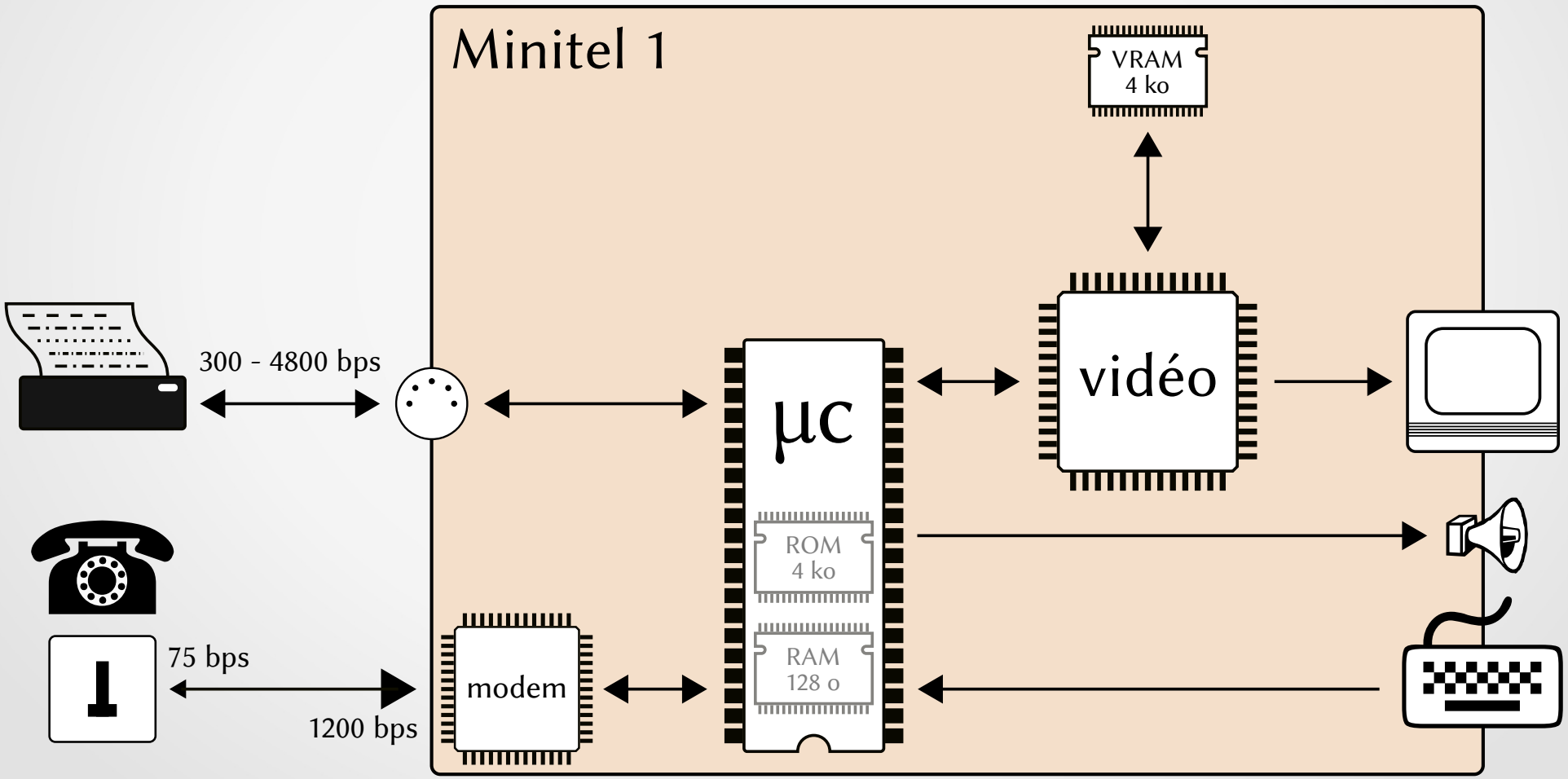
Après-rattrapge



Un Minitel

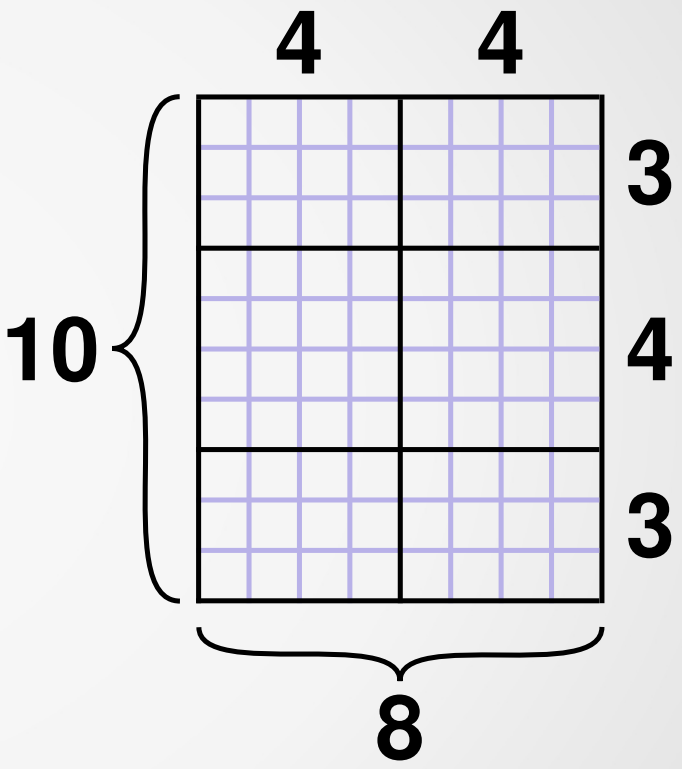
- Terminal passif
  distribué gratuitement, noir & blanc, coûtait 1000 F à fabriquer
- Norme Videotex
  utilisée aussi pour le télétexte
- 40×25 caractères
  alphabétiques ou semi-graphiques, attributs
- 8 niveaux de gris
  réellement 8 couleurs
- 1200 bit/s
  120 caractères par seconde !

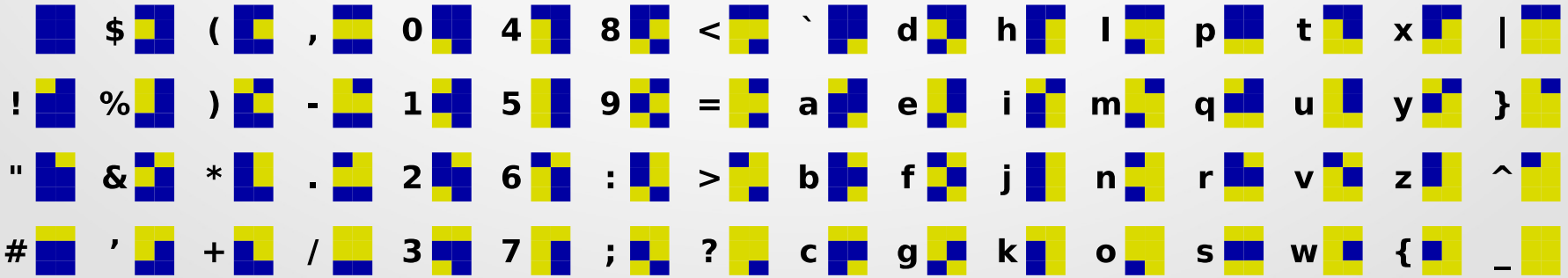
Schéma de principe



Semi-graphique ?

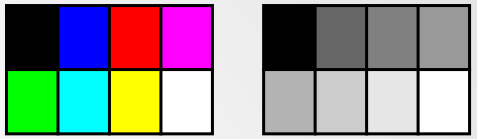
- Jeu de caractères spéciaux
  64 combinaisons de 2×3
- 2 couleurs maximum
  par bloc de 2×3
- Pixels de tailles différentes
  2/3 des pixels ne sont pas carrés !
- Résolution : 80×72



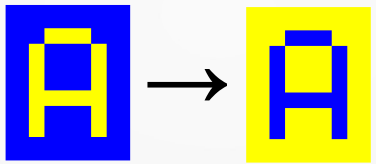


Attributs

- Couleurs d'avant-plan, d'arrière-plan
  noir, bleu, rouge, magenta, vert, cyan, jaune, blanc



- Clignotement
  de l'avant-plan
- Inversion vidéo
  avant-plan ↔arrière-plan



- Soulignement
- 4 tailles
  1×/2× largeur/hauteur

hello hello
hello hello

- Attention aux contraintes !

Videotex

- Flux continu
  aucune contrainte de temporalité ⟶ automate !
- Basé sur la norme Ascii
  7 bits, différence code de contrôle vs donnée
  quelques différences (^, {, }, ~, `)
- Commandes = code de contrôle
  les paramètres des commandes sont obligatoirement des codes de données, espace = caractère spécial
- Caractères supplémentaires
  utilisés pour les caractères accentués/spéciaux

Videotex, exemple

effacement de l'écran
ligne 12, colonne 5
double taille
inversion vidéo
avant-plan jaune

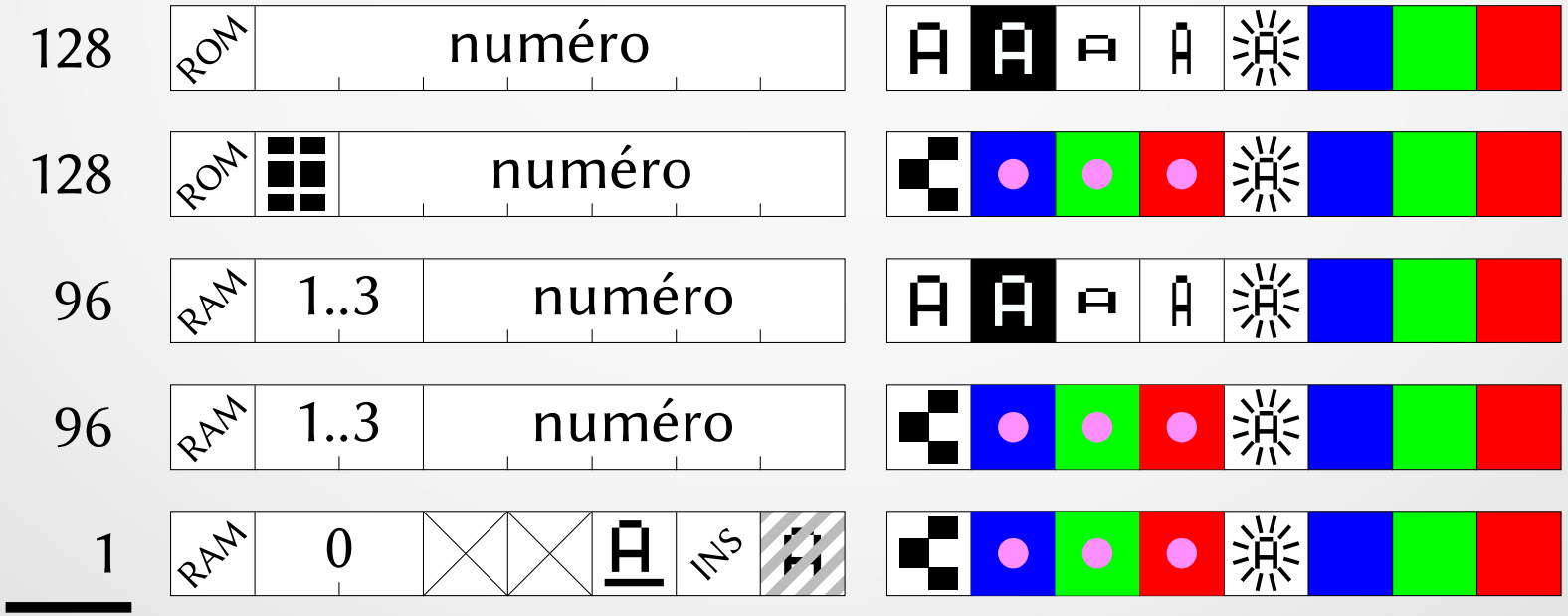
0C	1F 4C 45	1B 4F	1B 5D	1B 43

43	6F	64	65	75	72	73	20	65	6E	20	53	65	69	6E	65

Codeurs en Seine

Mémoire de page

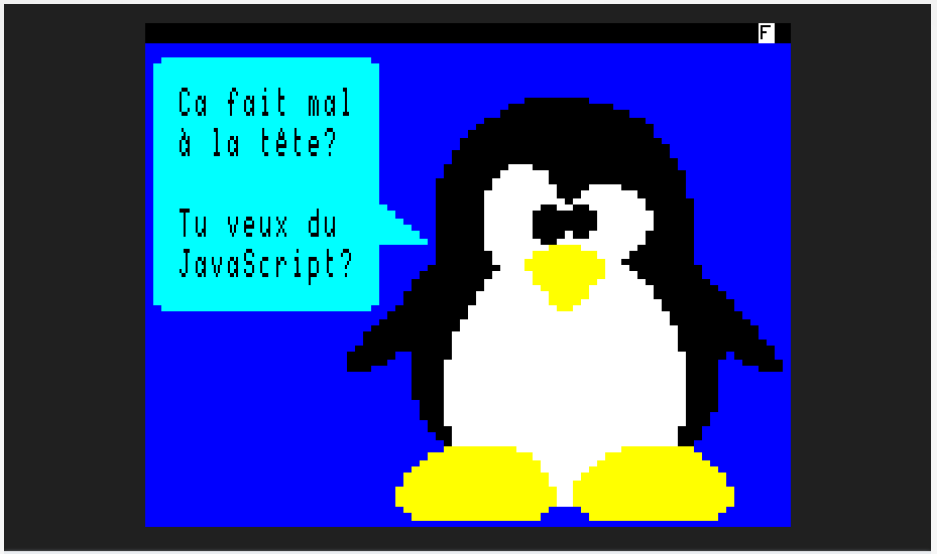
- Videotex est lié au processeur graphique
  famille EF934x, origine des contraintes
- Chaque élément est encodé sur 2 octets
  40×25×2 = 2000 octets



449 caractères affichables

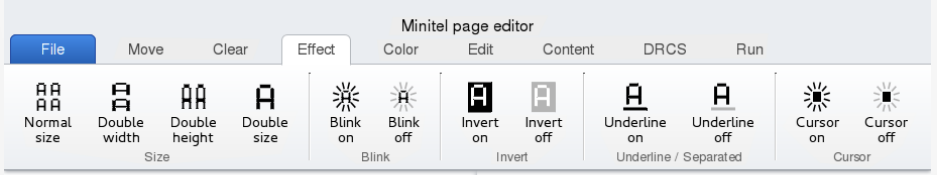
F

Editeur de pages

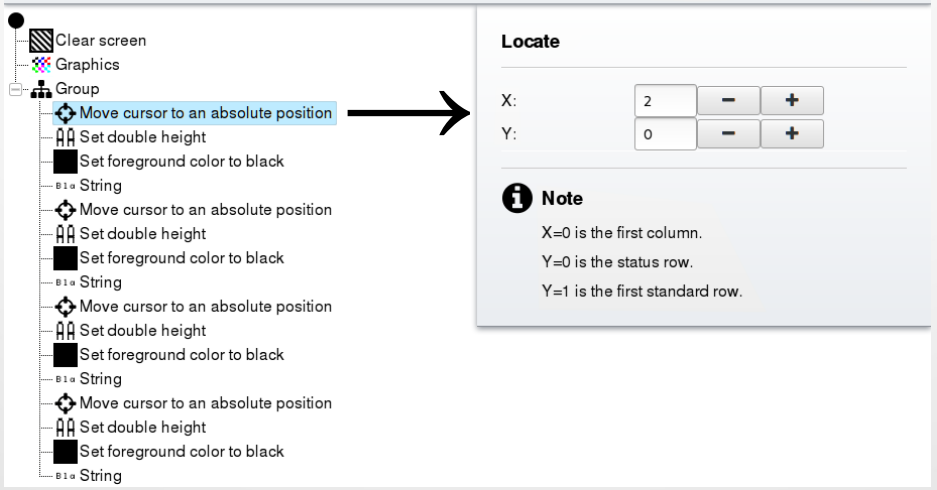


émulateur
rendu simultané

ruban

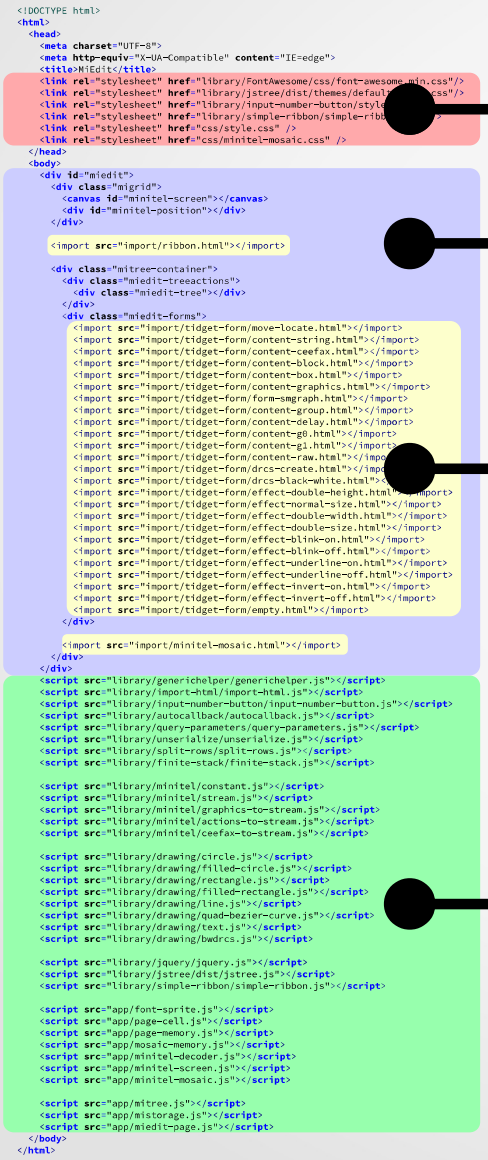


arborescence
des éléments

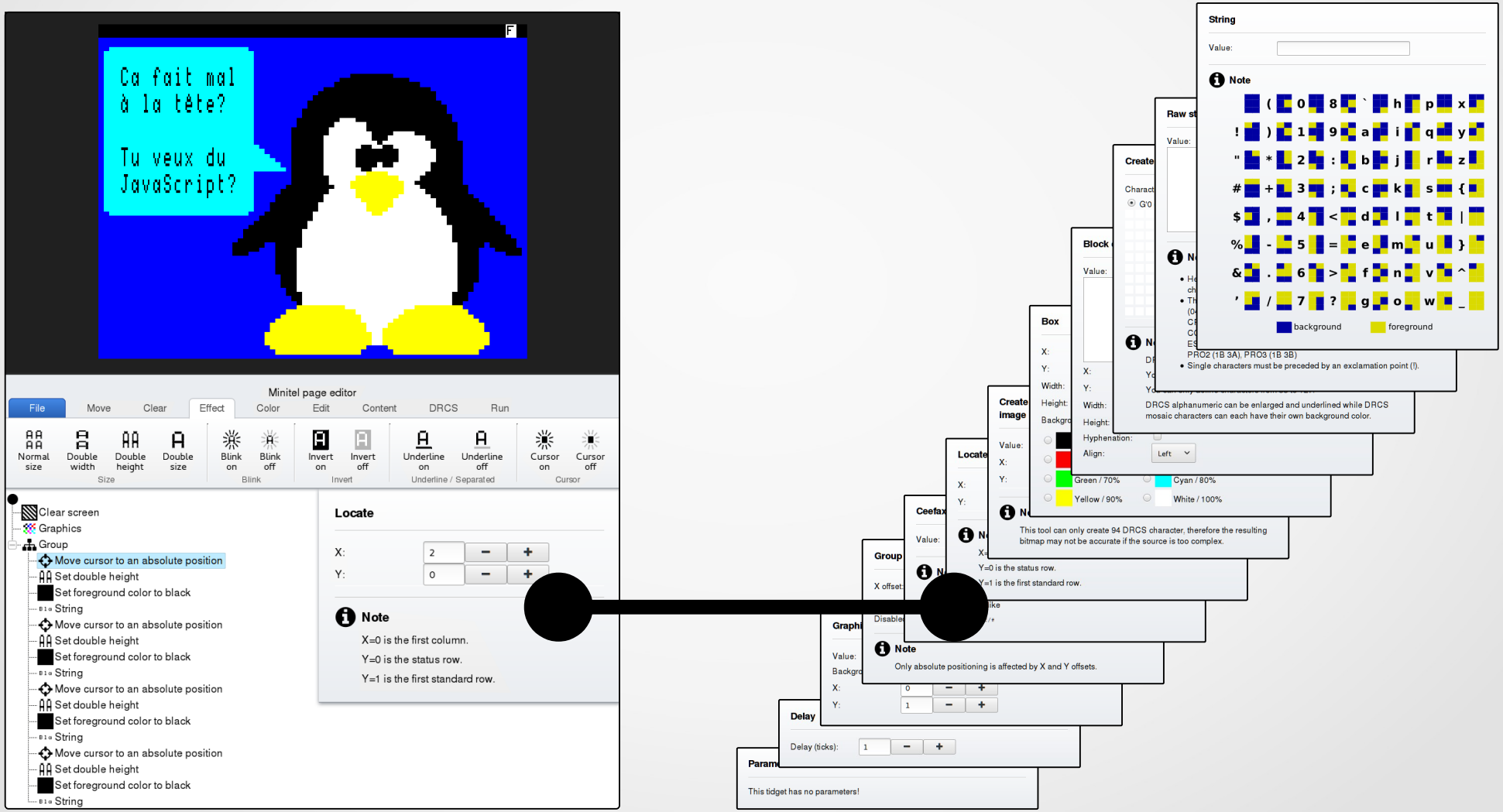


formulaire
contextuel

Séparation


présentation – CSS
contenu – texte + structure
imports
logique – JavaScript

Import HTML


Ça fait mal à la tête?
Tu veux du JavaScript?
Minitel page editor
File
Move
Clear
Effect
Color
Edit
Content
DRCS
Run
Normal size
Double width
Double height
Double size
Size
Blink on
Blink off
Blink
Invert on
Invert off
Invert
Underline on
Underline off
Underline / Separated
Cursor on
Cursor off
Cursor
Clear screen
Graphics
Group
Move cursor to an absolute position
Set double height
Set foreground color to black
String
Move cursor to an absolute position
Set double height
Set foreground color to black
String
Move cursor to an absolute position
Set double height
Set foreground color to black
String
Move cursor to an absolute position
Set double height
Set foreground color to black
String
Locate
X:
2
Y:
0
Note
X=0 is the first column.
Y=0 is the status row.
Y=1 is the first standard row.
String
Value:
Note
background
foreground
Box
Width:
Height:
Hyphenation:
Align:
Left
Green / 70%
Cyan / 80%
Yellow / 90%
White / 100%
This tool can only create 94 DRCS character, therefore the resulting bitmap may not be accurate if the source is too complex.
Y=0 is the status row.
Y=1 is the first standard row.
Note
Only absolute positioning is affected by X and Y offsets.
Delay
Delay (ticks):
1
This tidget has no parameters!

Balise custom

- Balise personnalisée
- Import côté client
  fonctionne en local sous Firefox mais pas Chrome
- Chargement asynchrone / parallèle

```
<div class="miedit-forms">
  <import src="move-locate.html"></import>
  <import src="content-string.html"></import>
  <import src="content-ceefax.html"></import>

  <!-- ... -->

  <import src="effect-invert-on.html"></import>
  <import src="effect-invert-off.html"></import>
  <import src="empty.html"></import>
</div>
```

Promesse

```
const importHTML = {
    import: function(element) {
        return new Promise((resolve, reject) => {
            const src = element.getAttribute("src")
            const xhr = new XMLHttpRequest()
            xhr.onerror = function(e) {
                reject("Error " + e.target.status + " while importing " + src)
            }

            xhr.onloadend = function onLoadEnd(event) {
                /* ... */
                resolve()
            }

            xhr.open("GET", src, true)
            xhr.send()
        })
    },

    install: function() {
        const importTags = Array.prototype.slice.call(
            document.getElementsByTagName("import")
        )
        return Promise.all(importTags.map(importHTML.import))
    },
}

importHTML.install().then(/* ... */).then(/* ... */)
```

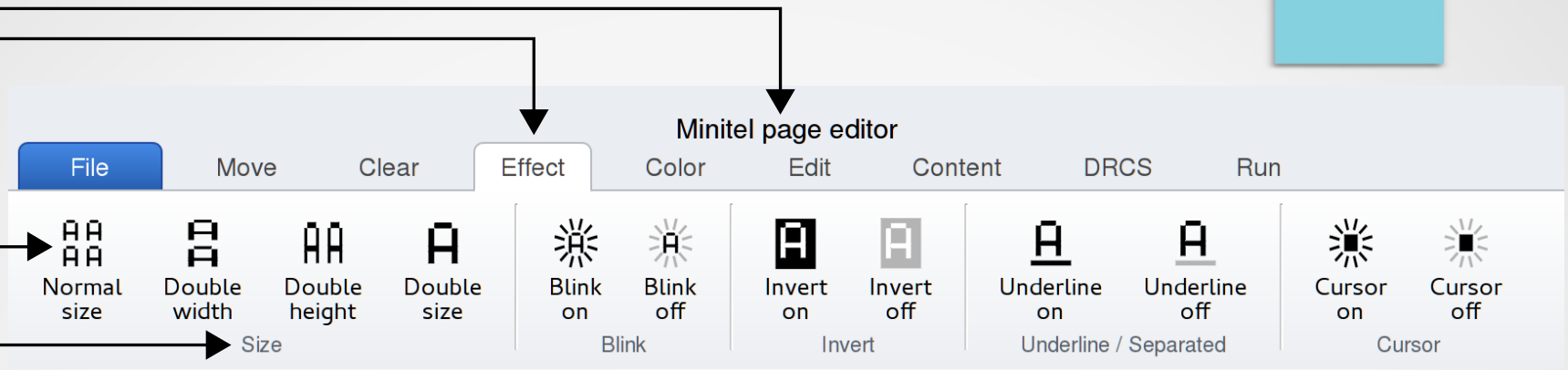
La promesse n'est pas tenue

La promesse est tenue

Appel asynchrone

Toutes les promesses doivent être tenues

Ruban




```
<div class="simple-ribbon" id="ribbon">
  <h2>Minitel page editor</h2>
  <!-- ... -->
  <h3>Effect</h3>
  <div id="effect-tab">
    <div>
      <h4>Size</h4>
      <button class="large">
        <img src="miicons.svg#effect-normal-size" /><br/>Normal<br/>size
      </button>
      <!-- ... -->
    </div>
    <hr/>
    <div>
      <h4>Blink</h4>
      <!-- ... -->
    </div>
    <!-- ... -->
  </div>
</div>
```

Fragments SVG

- Définition ⟶ SVG

```
<view id="file-import"
      viewBox="96 -320 32 32" />
```

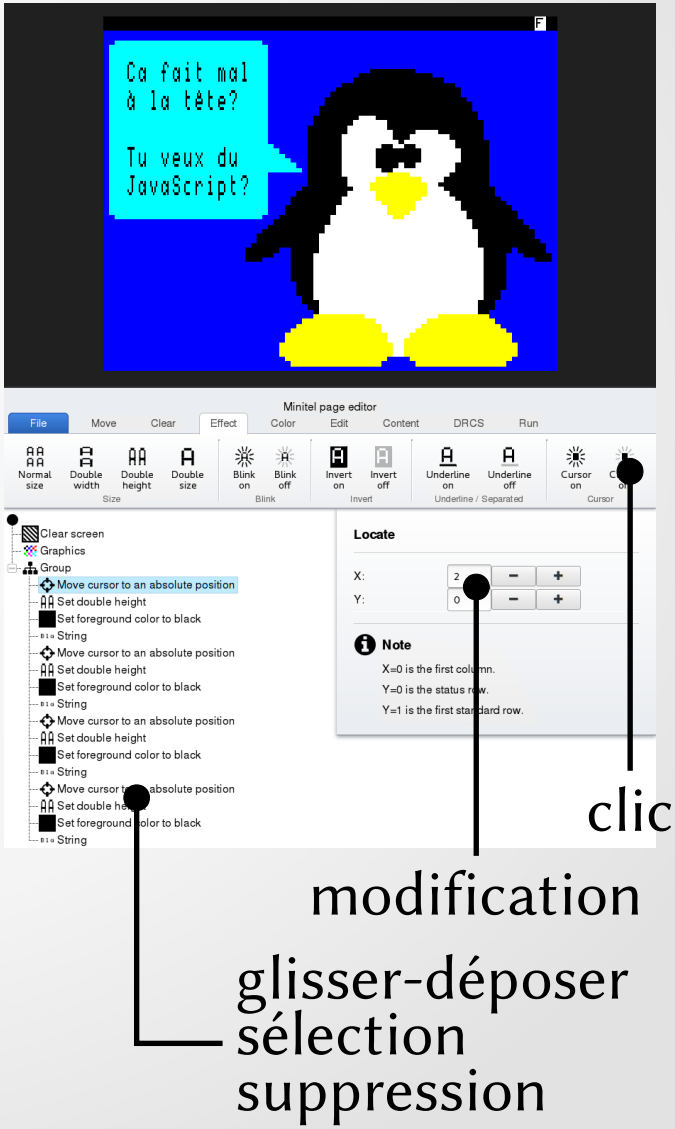
⚠ Coordonnée Y inversée !

- Utilisation ⟶ HTML, CSS

```
<img src="icon.svg#file-import" />
```

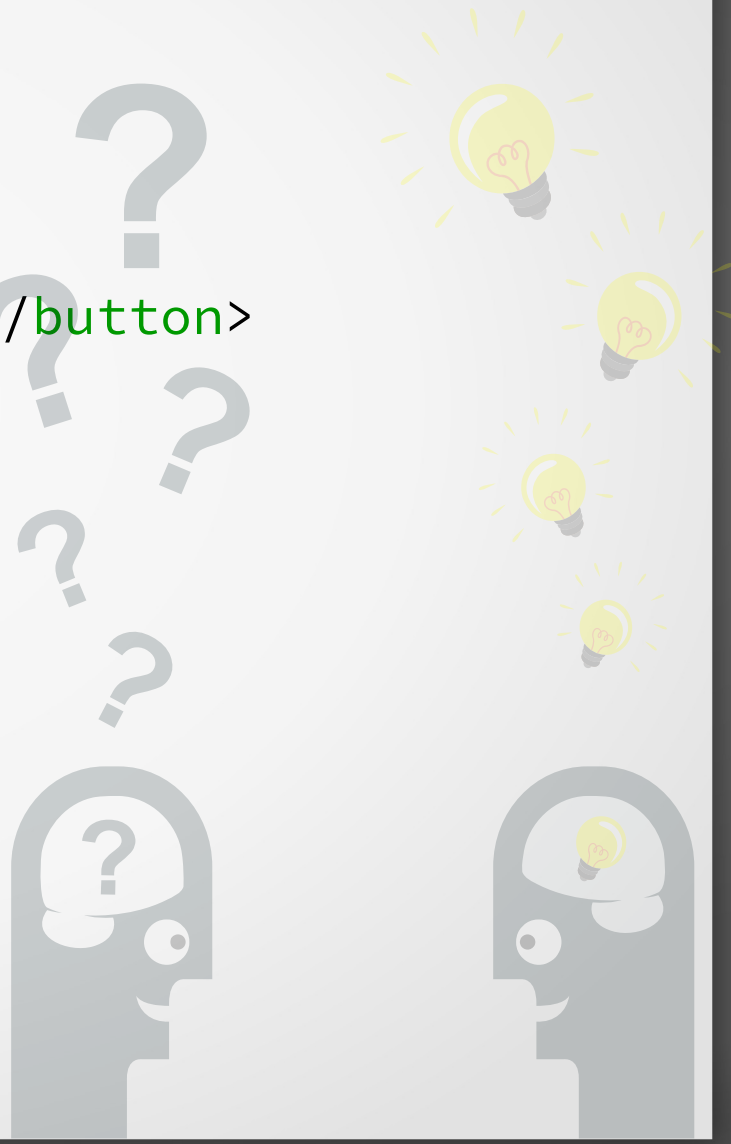
Callbacks

- Liaison GUI ⟶ application
- Événements jsTree
  déjà générés par jsTree
- Événements ruban
  entraîne généralement la création d'un élément
- Événements formulaire
  force le rafraîchissement de la fenêtre d'émulation

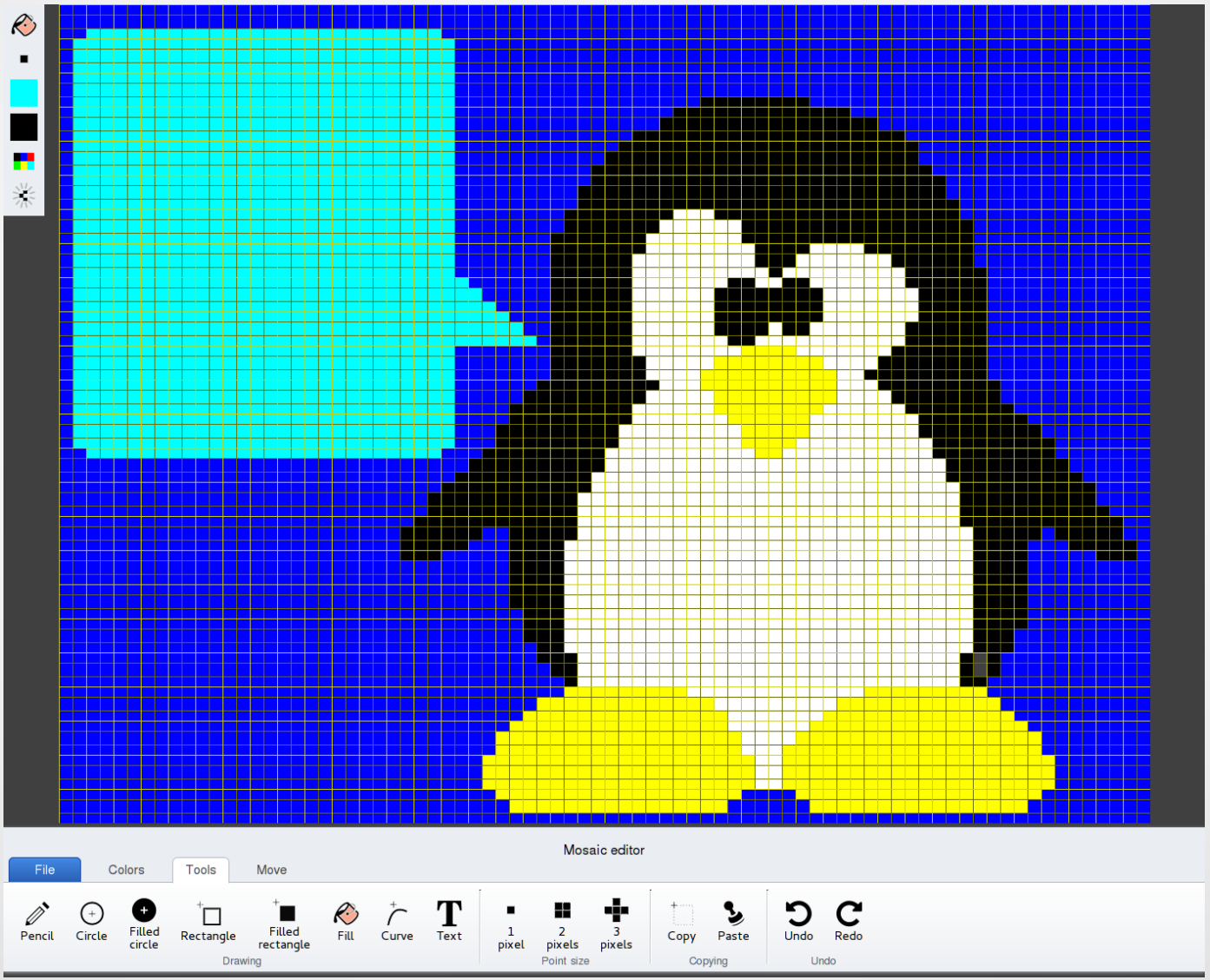


Attribut de donnée

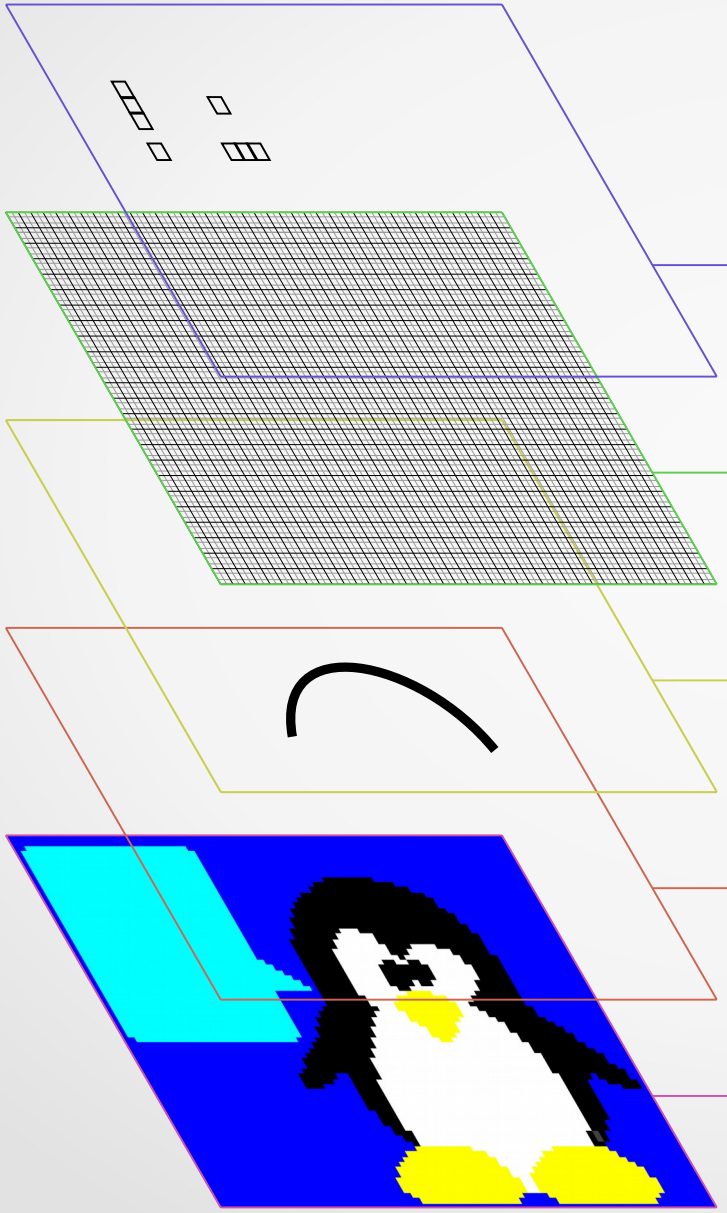
- Script autocallback.js
  apparie les événements d'un élément à une callback
- Définition
  ```
  <button data-call="onExport">Export</button>
  ```
- Utilisation
  ```
  class MiEditPage {
      constructor() {
          ribbon.autocallback(this)
      }

      onExport() {
          ...
      }
  }
  ```



Editeur graphique



Calques



Calque des erreurs

Calque de grille

Calque de recopie

Calque de prévisualisation

Calque de dessin

Empilage

- Peu importe le type de balise
  canvas, div…
- L'empilage simplifie la vie
  - 1 calque = 1 type d'objet
  - Empilage du navigateur **très** rapide
  - Vidage de calque **très** rapide
- Insensibiliser un calque aux événements
  En CSS : `pointer-events: none`

Générateur

- Suppression des `for`

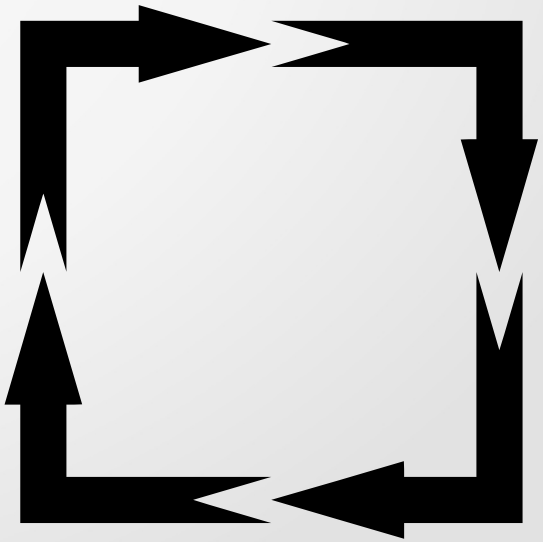
```
for(let i = 0; i < this.grid.cols; i++) {
    /* ... */
}
```

- Remplacement par des `range`

```
range(this.grid.cols).forEach(i ⇒ {
    /* ... */
})
```

- Différence

  gestion des sorties de boucle,
  code exécuté = un appel de fonction

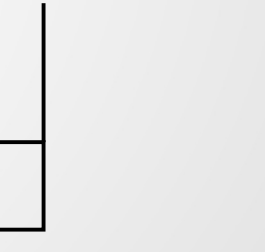


Générateur

générateur

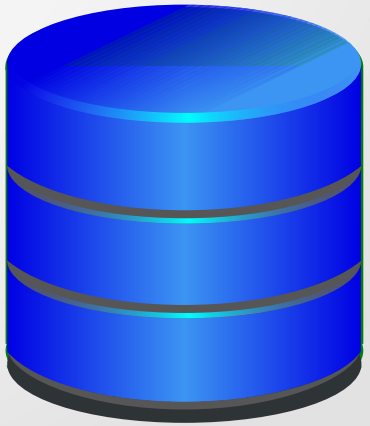
```
function* range(start, end, step) {
    if(start === undefined || start === end) return

    if(end === undefined) {
        end = start
        start = 0
    }

    if(step === undefined) step = 1

    step = Math.abs(step)

    if(end < start) {
        for(let i = end; i > start; i -= step) yield i
    } else {
        for(let i = start; i < end; i += step) yield i
    }
}

range().__proto__.forEach = function(func) {
    for(let i of this) func(i)
}
```

interruption



Web Storage

- Enregistrement dans le navigateur
  - à la session : `sessionStorage`
  - permanent : `localStorage`
- Clé-valeur
- Éditable avec les outils de développement
- API simple



Web Storage

Liste

```
const keys = Object.keys(localStorage)
```

Écriture

```
localStorage.setItem(key, JSON.stringify(value))
```

Suppression

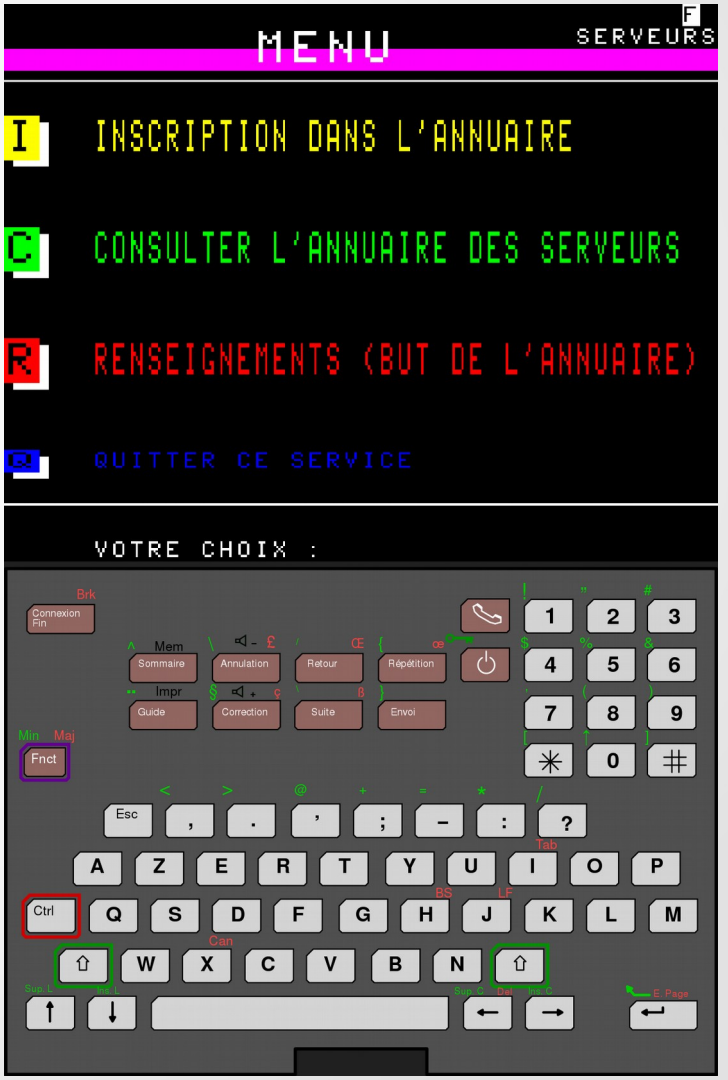
```
localStorage.removeItem(key)
```

Lecture

```
const value = JSON.parse(localStorage.getItem(key))
```

Web Socket

- Liaison full-duplex
  l'envoi des touches au serveur et l'envoi des pages du serveur au client sont asynchrones
- Client simple
  mais quelques soucis quand un proxy intervient…
- Serveur moins simple
  Apache requiert `mod_wstunnel` pour faire cohabiter le serveur web et les websockets



Client

création d'une WebSocket

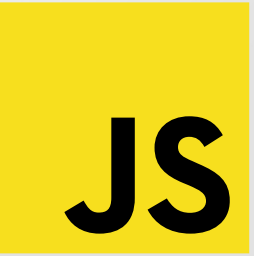
connexion sécurisée

connexion établie

```
const socket = new WebSocket("wss://minitel.ouep.eu/ws/")

socket.onopen = event => {
    socket.send(/* Value or JSON */)

    socket.onmessage = function(event) {
        /* Work with the event.data array */
    }
}
```

envoi d'un message

message reçu



Serveur

```
import asyncio
import websockets

@asyncio.coroutine
def minitel(websocket, path):
    yield from websocket.send(""" Home page """)
    while True:
        byte = yield from websocket.recv()
        """ ... """
        yield from websocket.send(byte)

def start_server():
    start_server = websockets.serve(minitel, "localhost", 30001)

    asyncio.get_event_loop().run_until_complete(start_server)
    asyncio.get_event_loop().run_forever()

start_server()
```

envoi de données

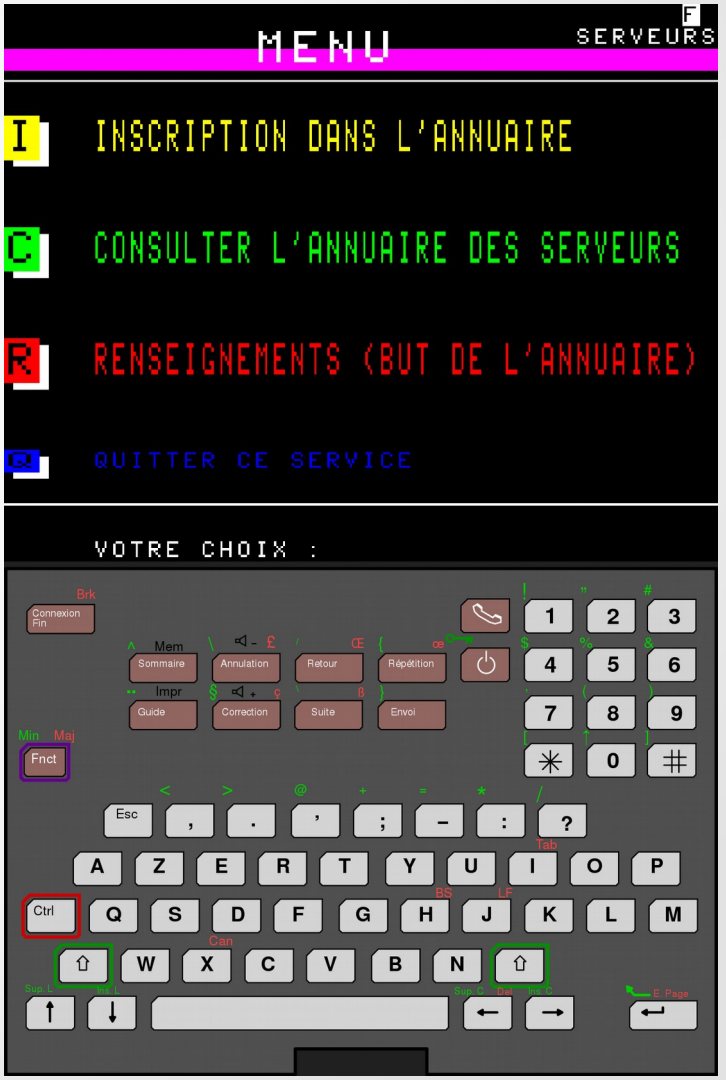
réception de données

ouverture du port

lancement du serveur

Son d'une touche

- Séparation en 2 parties
  appui et relachement
- Nécessité d'une file
  Permet de jouer plusieurs sons simultanément
- Duplication d'un son
  `node.cloneNode(true)`

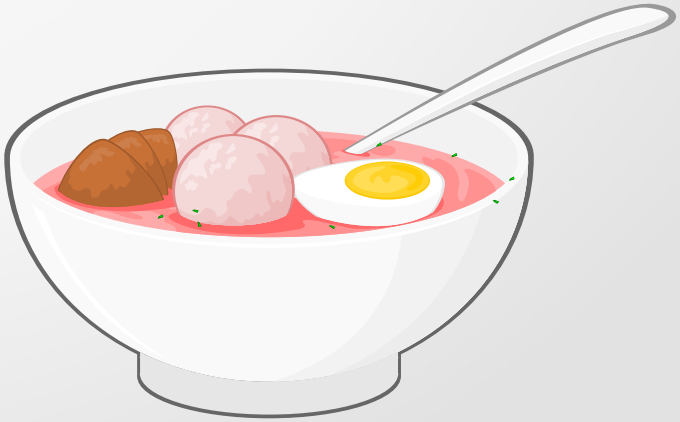


EcmaScript 6

- Support large
  publiée en juin 2015
- Plein de bons trucs !
  fonctions fléchées, modules, classes, portée lexicale au niveau des blocs, itérateurs et générateurs, promesses, patrons de décomposition, optimisation des appels terminaux, tableaux associatifs, ensembles, tableaux binaires, meilleur support Unicode, structures de données prédéfinies extensibles...
- Mais...
  implémentation des modules encore expérimentale,
  forEach non supporté par NodeList

Recette

- Utilisation du mode strict
  activé automatiquement dans les `class`
- Abandon de `var`
  au profit de `let` et `const`, une seule déclaration par `let`/`const`
- Exécution asynchrone
  ne jamais bloquer l'exécution d'un script, utilisation des promesses
- Fonctions fléchées
  une solution aux problèmes de `this`
- Abandon des points-virgules
- No jQuery
  sauf en cas de nécessité… jsTree



F

Merci de votre attention!

https://zigazou.github.io/miedit