

Charte graphique BLOG

Création d'un blog

Charte graphique = identité du client

- Couleurs

- HTML: attribut style

```
<... style="color: #123456" ...>
```

- CSS: color, background-color

```
... { ... color: #123456; ... }
```

- Logos

```

```

Création d'un blog

Charte graphique = identité du client



- Typographiques:
 - Polices personnalisées: CSS:

```
@font-face {  
    font-family: 'MaPolice';  
    src: url('MaPolice.ttf') format('truetype');  
}
```
 - Polices externes (Google fonts,...):
 - HTML: <link> entre les balises <head>

```
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
```
 - CSS: @import avant tout autre règle css et cela dans un fichier css ou entre les balises <head><style> du fichier html

```
@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');
```
- Référence dans le CSS:

```
body {  
    font-family: 'MaPolice', 'Roboto', sans-serif;  
}
```

Création d'un blog

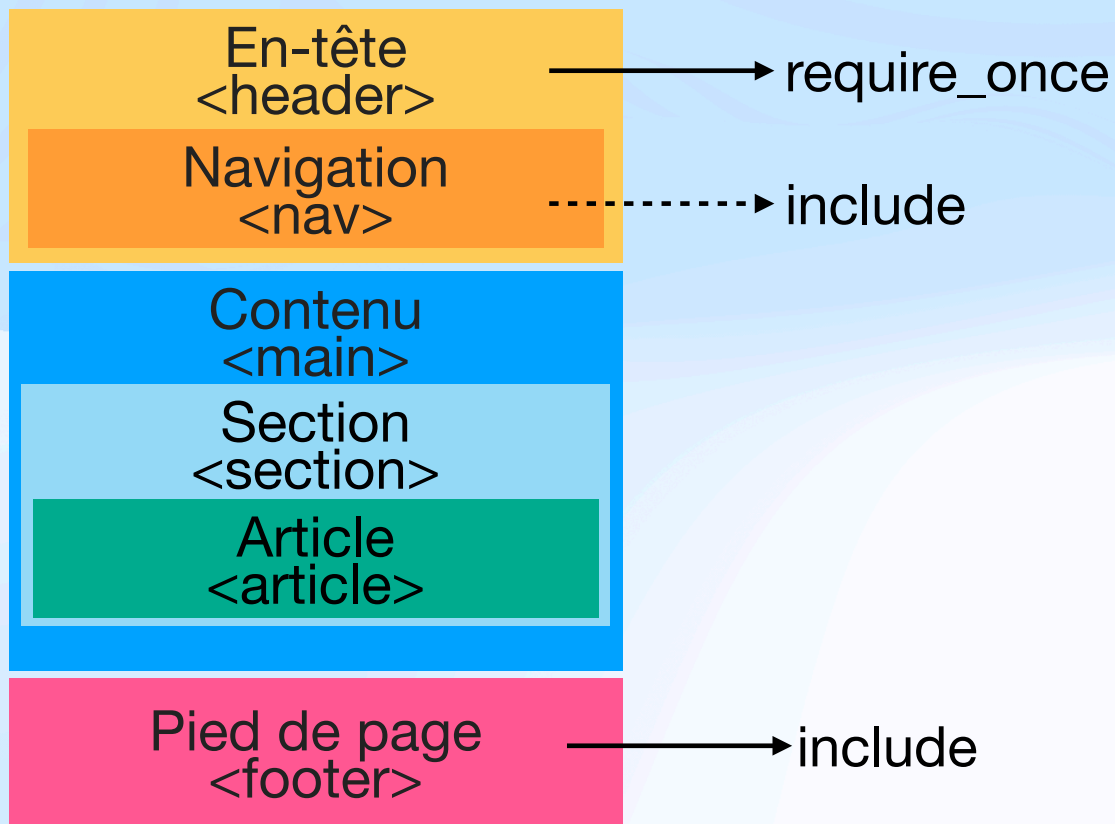
Conventions encore...

- Même logique sur tout le site permet aussi de savoir qu'on est toujours sur le même site.
- Logo qui redirige vers la home page pour ne plus être perdu.
- Chaque page doit avoir un nom bien placé qui doit ressortir et correspondre à ce que j'ai cliqué.
- Mise en évidence de l'endroit où on se trouve sur le site.
- Affichage du chemin depuis la home page (ex: en séparant les éléments par > et en mettant le dernier en gras et le tout en haut de la page)
- A tout moment on doit savoir à quel site appartient la page, où je suis dans la structure, dans quelle grande section, les options de navigation et où puis-je chercher.
 - ▶ On peut tomber sur une page sans passer par la page d'accueil (Lien externe,...)

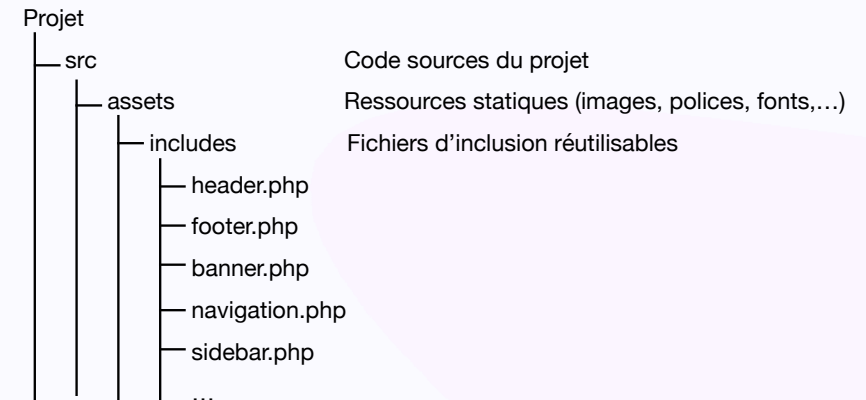
Organisation fichiers BLOG

Création d'un blog

PHP: Include / Require / Once



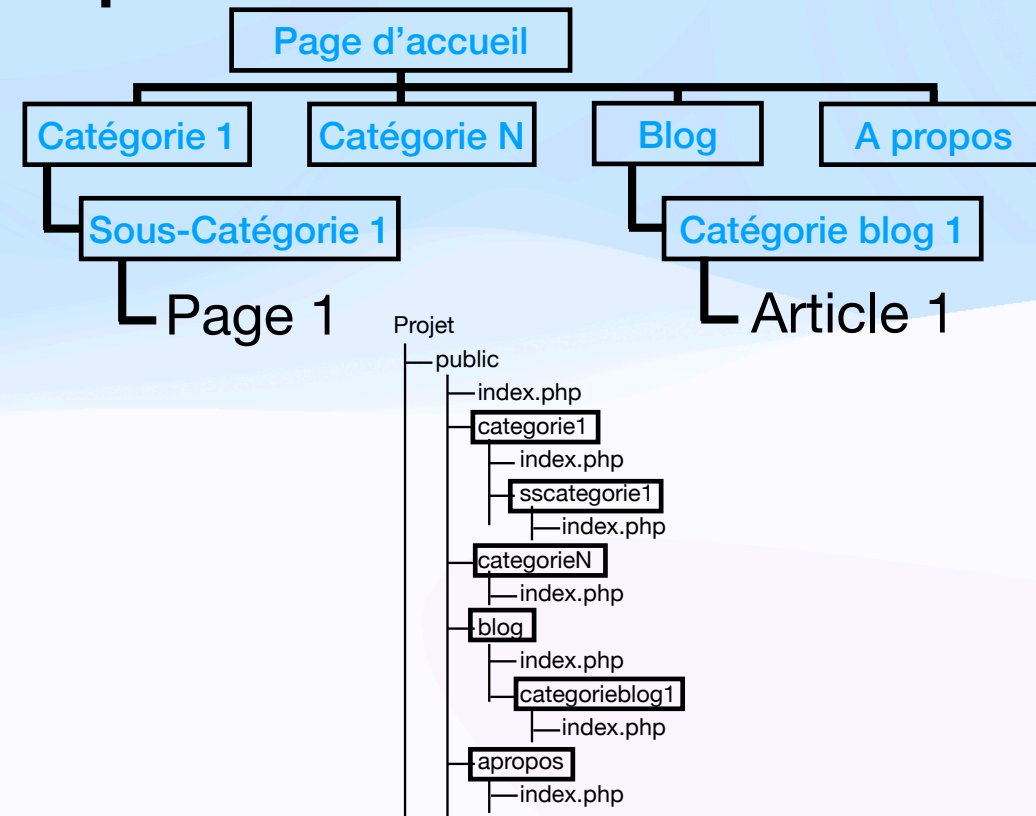
Fonction	Si fichier n'existe pas	Inclus le fichier plusieurs fois
include	Warning	Oui
require	Erreur fatale	Oui
include_once	Warning	Non
require_once	Erreur fatale	Non



Création d'un blog

Répertoires et types de fichiers: Exemple de convention

Projet	
—src	Code sources du projet
—assets	Ressources statiques (images, polices, fonts,...)
—css	CSS (Style Sheet) ou préprocesseur (SCSS, LESS,...)
—js	JS (JavaScript) incluant des modules et des scripts spécifiques
—images	Fichiers graphiques (PNG, JPG, SVG,...)
—Fonts	Police de caractère personnalisées ou téléchargées
—components	Composants réutilisables (HTML, CSS, JS,... boutons, en-têtes,...) (React: .jsx, .tsx,...)
—public	Fichiers directement accessibles par le navigateur sans traitement supplémentaire
—index.html	Point d'entrée principal du site (Home page)
—favicon.ico	Favicon du site (affichée dans l'onglet du navigateur)
—robots.txt.	Directives pour les robots de moteur de recherche pour guider l'indexation
—tests	Testing pour assurer le bon fonctionnement du code
—unit	Test unitaires (Test des fonctions isolées,...)
—integration.	Test d'intégration (Vérification du fonctionnement plus global)
—build	Version construite = Production = fichiers minimisé, optimisées prêts à être déployés
—...	
—node_modules.	Dépendances Node.js
—package.json	Configuration, dépendances et scripts de build/projet Node.js
—README.md.	Documentation pour expliquer le projet (dont la structure des répertoires)
— .gitignore	Liste des fichiers/répertoires à ignorer par Git





Création de tout type de sites web



Logiciel open source totalement gratuit

Wordpress

CMS (Content management System)

Le plus utilisé au niveau mondial

Représente environ 43% des sites internet

Questions

Charte graphique

- **Quels sont les éléments clés d'une charte graphique**
- **Comment cette charte graphique doit-elle être adaptée aux différents supports et formats?**
- **Avantage et inconvénients d'une plateforme prête à l'emploi?**
- **Installation de Wordpress**
- **Création d'un premier site**
- **Analyse de la structure / organisation des fichiers**

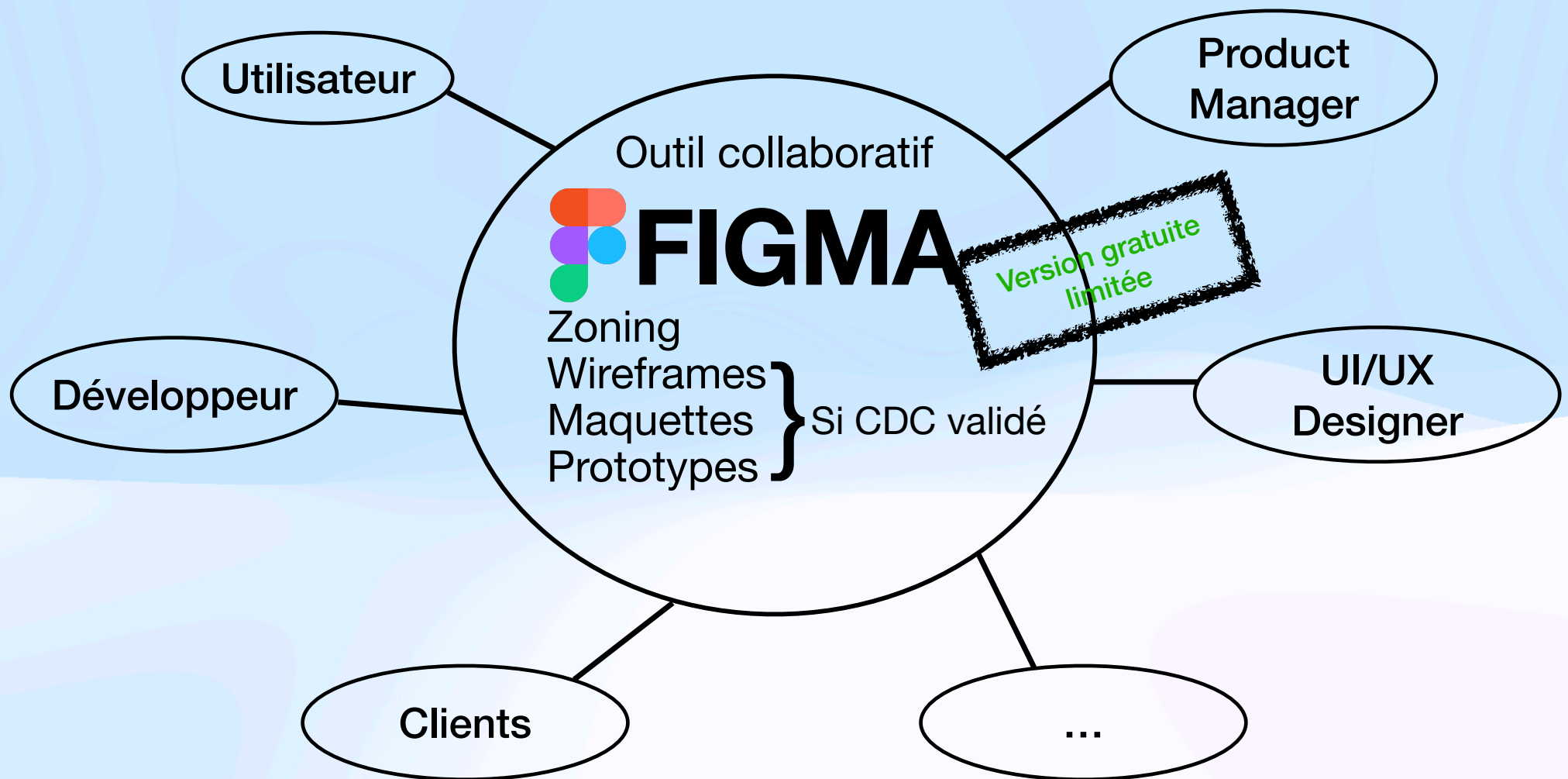
Objectif fin d'atelier 1

Cahier des charges du site internet de fin de module

- 1. Contexte:**
- 2. Objectifs:**
- 3. Cible utilisateur**
- 4. Fonctionnalités**
- 5. Contraintes techniques**
- 6. Architecture et arborescence de l'information :**
- 7. Zoning, Wireframes, maquette**
- 8. Charte graphique et identité visuelle**

FIGMA

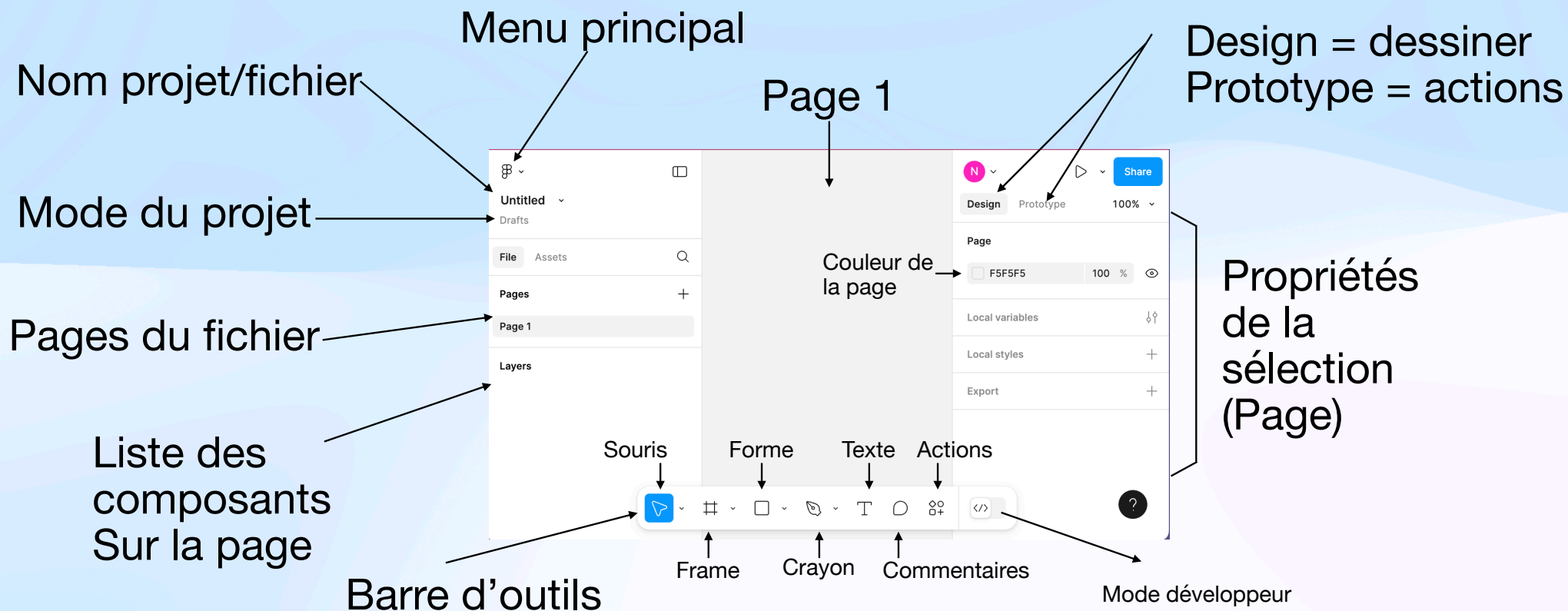
Introduction





FIGMA

L'environnement de travail





L'environnement de travail: Créer des « Design »

- Frame = conteneur = page d'un site,...

Point à modifier

Frame

Raccourci Clavier

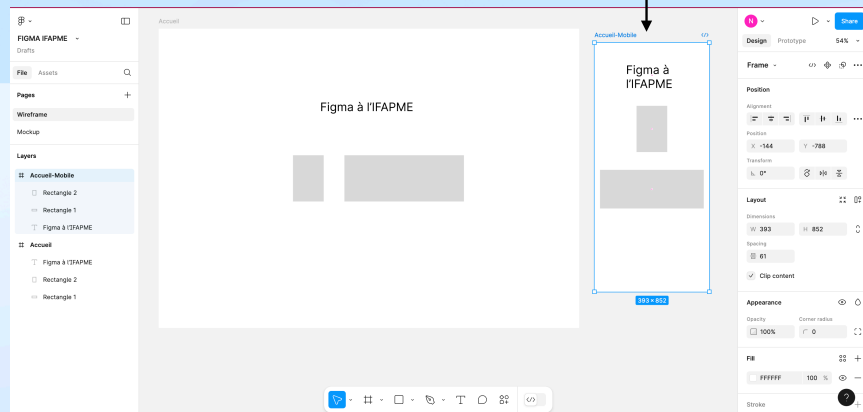
Propriétés de la Frame

Device	Dimensions
iPad Pro 11"	834 × 1194
iPad Pro 12.9"	1024 × 1366
MacBook Air	1280 × 832
MacBook Pro 14"	1512 × 982
MacBook Pro 16"	1728 × 1117
✓ Desktop	1440 × 1024
Wireframe	1440 × 1024
TV	1280 × 720
Slide 16:9	1920 × 1080

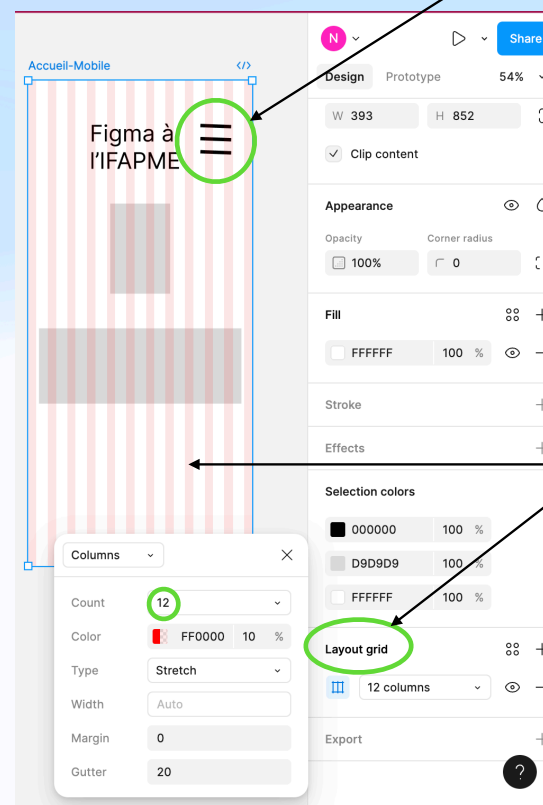


L'environnement de travail: Différentes versions

Frame iPhone 16



Menu hamburger
fait de 3 lignes



Partage de l'espace
en 12 colonnes
 $12 = 2 \times 6 = 3 \times 4$

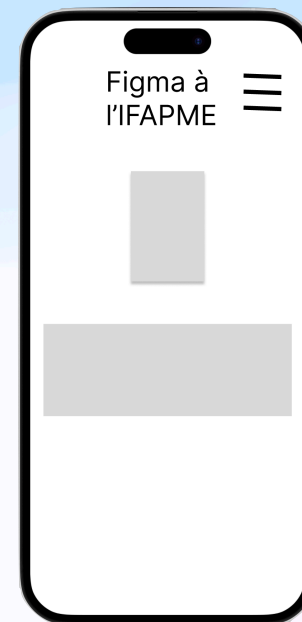
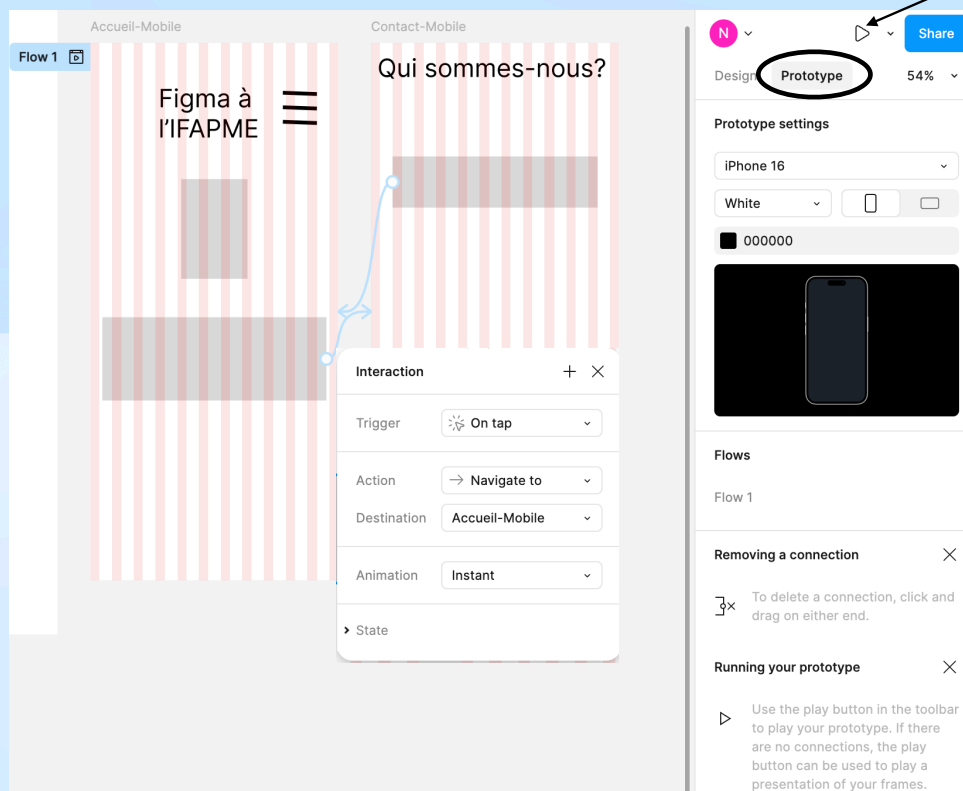
Permet de mieux
répartir les composants



FIGMA

L'environnement de travail: Prototypes

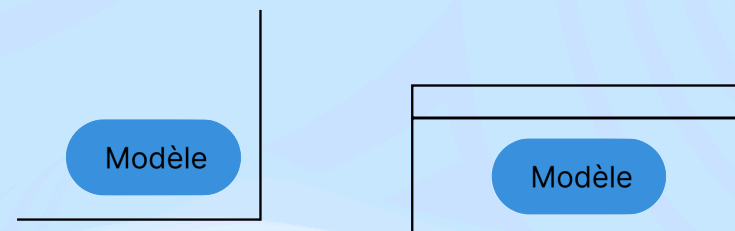
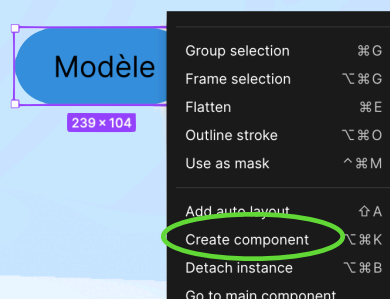
Démarrer le prototype





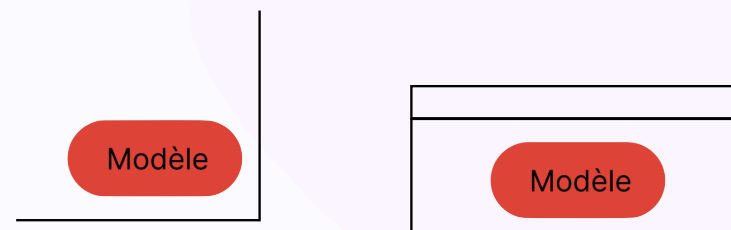
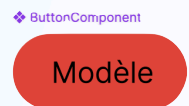
FIGMA

L'environnement de travail: Composants



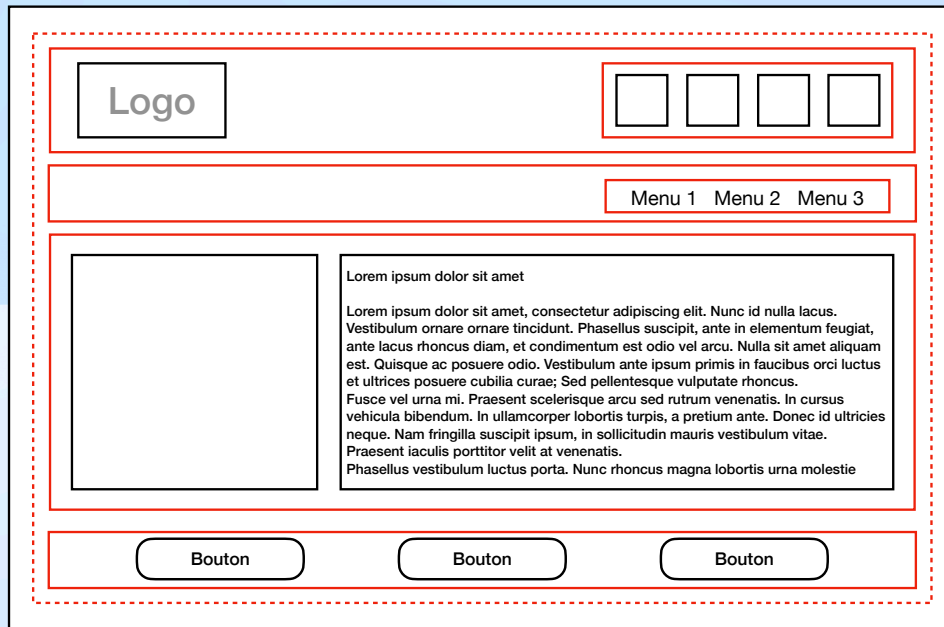
1. Créer un dessin
2. Sélectionner les éléments
3. Clic droit -> Create component

1. Créer des instances du composant (copier/coller)
2. Modifier le composant (Couleur, dimensions, font,...)
-> Les instances du composant sont modifiées



Création d'un blog

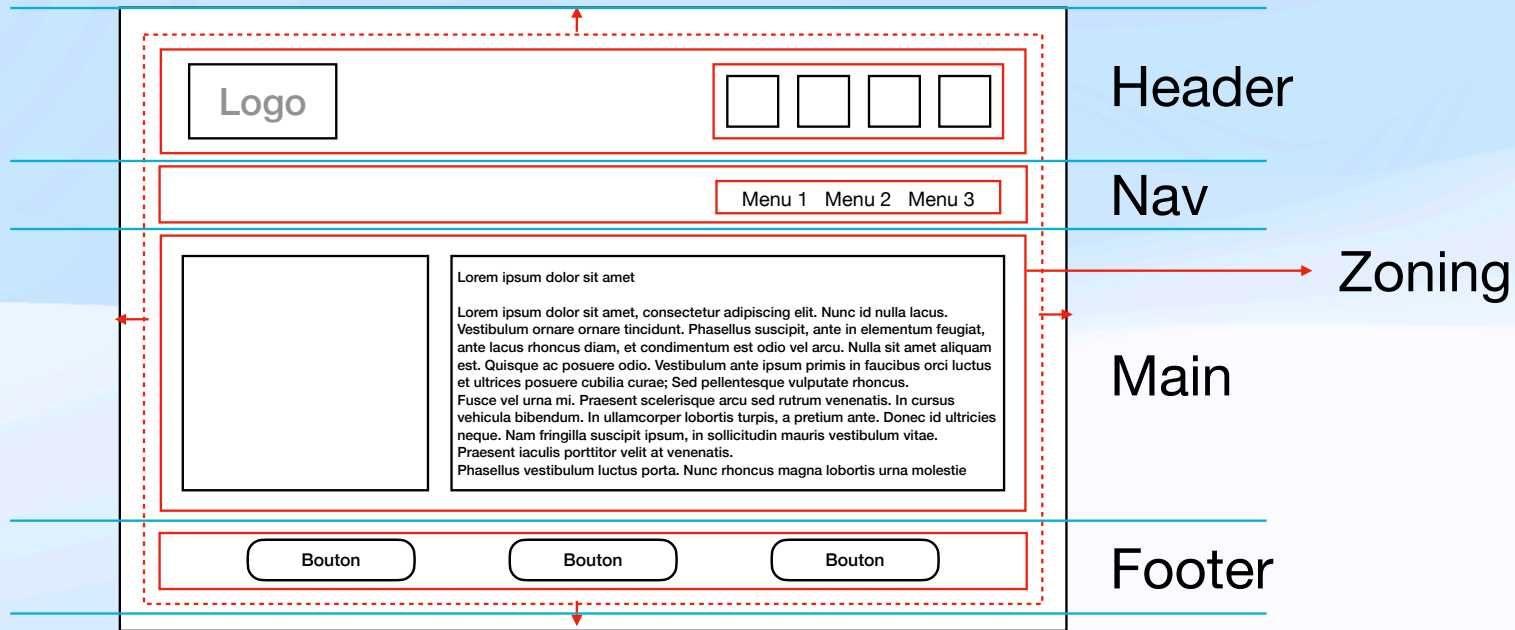
Zoning / Wireframe / Maquette dans Figma



De la maquette au code

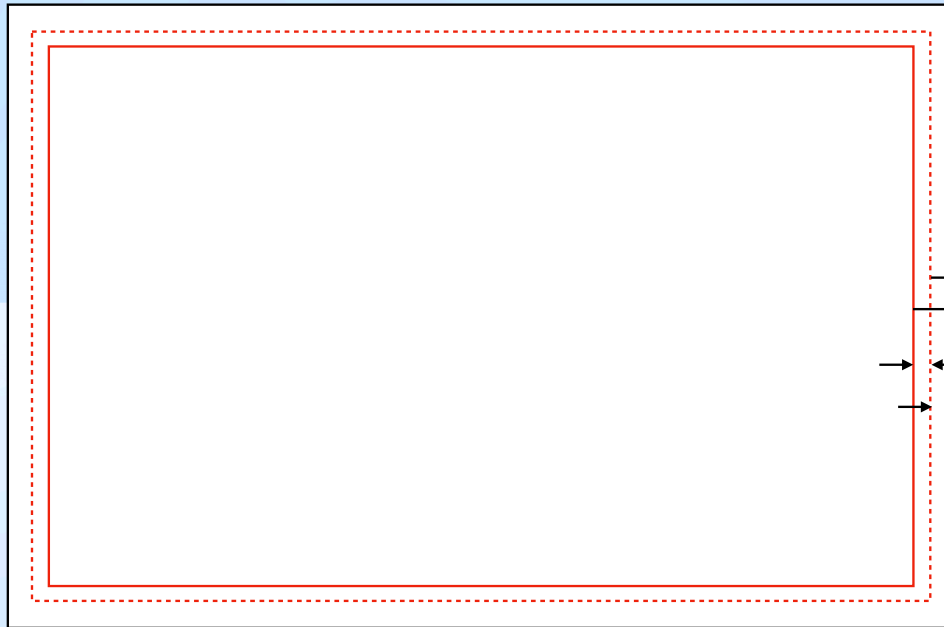
Création d'un blog

Zoning / Wireframe / Maquette et HTML



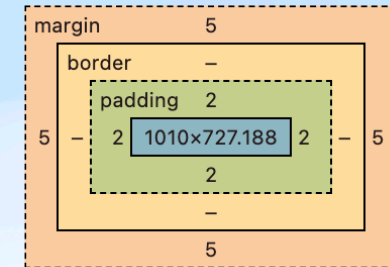
Création d'un blog

Zoning / Wireframe / Maquette et HTML (Desktop)



CSS:

```
* {  
  margin: 5px;  
  padding: 2px;  
  box-sizing: border-box;  
}
```



Remarque:

- `box-sizing: border-box;`
Largeur totale = $\text{width}(\text{content} + \text{padding} + \text{border}) + \text{Margin}(G + D)$
- `box-sizing: content-box;`
Largeur totale = $\text{width}(\text{content}) + \text{padding}(G/D) + \text{border}(G/D) + \text{Margin}(G/D)$

CSS: (<body>)

```
body {  
  font-family: Arial, sans-serif; ———> Liste ordonnée par priorité de police  
  padding: 0.1rem; ———> 1/10 de la valeur constante égale à la taille de la police  
}
```

Création d'un blog

Zoning / Wireframe / Maquette et HTML (Desktop)



HTML:

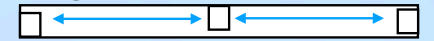
```
<header>
  <img class="img-logo" src="" alt="Logo">
  <div class="quick-launch border">
    <img class="img-quick-lauch" src="" alt="Lancement rapide 1">
    <img class="img-quick-lauch" src="" alt="Lancement rapide 2">
    <img class="img-quick-lauch" src="" alt="Lancement rapide 3">
    <img class="img-quick-lauch" src="" alt="Lancement rapide 4">
  </div>
</header>
```

CSS:

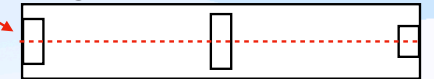
```
header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border: 2px solid black;
  margin-bottom: 1rem;
  padding: 1rem;
}
```

Active le mode flex-box sur le conteneur

Alignement horizontal



Alignement Vertical



```
.img-logo {
  width: 150px;
  height: auto;
  border: 2px solid black;
}
```

Permet de conserver le ratio de l'image

```
.quick-launch {
  display: flex;
  gap: 0.5rem;
}
```

Aligne les enfants sur une ligne horizontale par défaut

Espace de 0.5 rem entre les enfants

```
.img-quick-lauch {
  width: 50px;
  height: 50px;
  border: 2px solid black;
}
```

```
.border {
  border: 2px solid black;
  padding: 0.5rem;
}
```

Création d'un blog

Zoning / Wireframe / Maquette et HTML



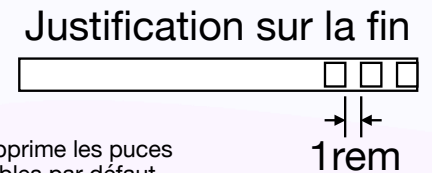
HTML:

```
<nav>
  <ul>
    <li><a href="#">Menu 1</a></li>
    <li><a href="#">Menu 2</a></li>
    <li><a href="#">Menu 3</a></li>
  </ul>
</nav>
```

CSS:

```
nav {
  border: 2px solid black;
  padding: 0.5rem;
  margin-bottom: 1rem;
  background-color: #f8f8f8;
}
```

```
nav ul {
  display: flex;
  justify-content: flex-end;
  gap: 1rem;
  list-style: none;
}
```



Supprime les puces
visibles par défaut

Style pour le lien

```
nav ul li a {
  text-decoration: none;
  color: black;
  font-weight: bold;
  padding: 0.5rem 1rem;
}
```

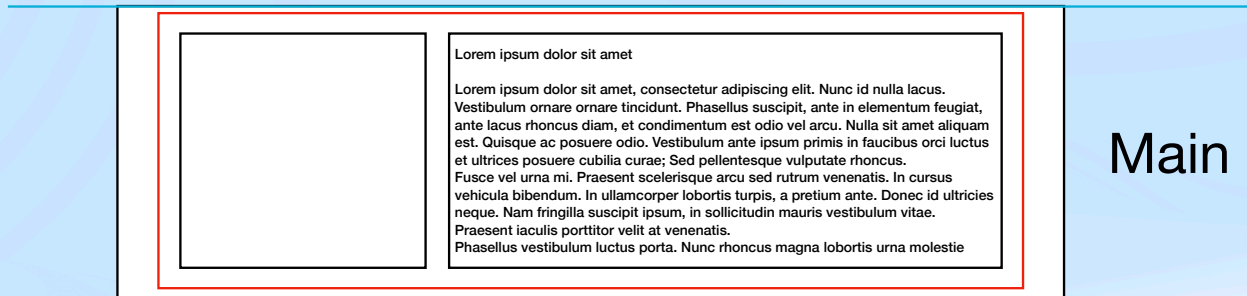
Supprime le soulignement,...

Espacement vertical

Espacement horizontal

Création d'un blog

Zoning / Wireframe / Maquette et HTML



Main

CSS:

```
main {  
  border: 2px solid black;  
  padding: 1rem;  
}
```

```
article {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 1rem;  
  margin-bottom: 1rem;  
}
```

Permet aux enfants d'aller à la ligne si pas assez de place

```
.img-art {  
  width: 100%;  
  max-width: 300px;  
  height: auto;  
}
```

Image aussi large que son conteneur

```
.desc-art {  
  flex: 1;  
  padding: 1rem;  
  border: 2px solid black;  
}
```

Occupe l'espace complet restant

HTML:

```
<main>  
  <article>  
    <img class="img-art" src="" alt="Image article">  
    <div class="desc-art">  
      <h2>Lorem ipsum dolor sit amet</h2>  
      <p>Lorem ipsum ... urna molestie</p>  
    </div>  
  </article>  
</main>
```


Création d'un blog

Zoning / Wireframe / Maquette et HTML



HTML:

```
<footer>
  <div class="boutons">
    <button class="bouton-arrondi">Bouton</button>
    <button class="bouton-arrondi">Bouton</button>
    <button class="bouton-arrondi">Bouton</button>
  </div>
</footer>
```

CSS:

```
footer {
  border: 2px solid black;
  padding: 0.1rem;
  margin-top: 1rem;
}
```

```
.boutons {
  display: flex;
  justify-content: space-evenly;
  gap: 1rem;
  flex-wrap: wrap;
}
```

Répartition même distance



```
.bouton-arrondi {
  padding: 10px 20px;
  font-size: 16px;
  color: black;
  background-color: white;
  border: 2px solid black;
  border-radius: 10px;
  min-width: 150px;
  text-align: center;
}
```

Création d'un blog

Zoning / Wireframe / Maquette et HTML

HTML:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

← Pour appareils mobile / responsive
Utilise toute la largeur de l'écran
Zoom initial sur la page: 100%

CSS:

```
@media (max-width: 768px) { ← Si largeur ≤ 768 pixels (= si tablette ou mobile)
```

```
  header {  
    flex-direction: column; ← Passe de ligne horizontale à verticale  
    text-align: center; ← Contenu centré  
  }
```

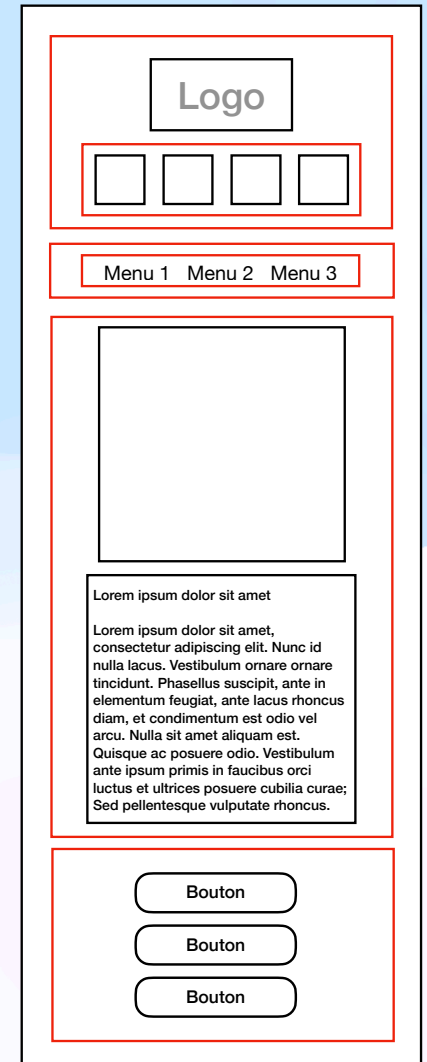
```
  nav ul {  
    justify-content: center; ← Centre horizontalement  
    flex-wrap: wrap; ← Autorise les enfants à aller à la ligne  
  }
```

```
  .quick-launch {  
    justify-content: center;  
  }
```

```
  article {  
    flex-direction: column;  
    align-items: center;  
  }
```

```
  footer .boutons {  
    flex-direction: column;  
    align-items: center;  
  }
```

```
}
```



Questions

De la maquette au code

- Application au site internet de fin d'atelier