

# Semaine 03 – Stabiliser responsive (fondations)

**Lundi matin: savoirs uniquement**

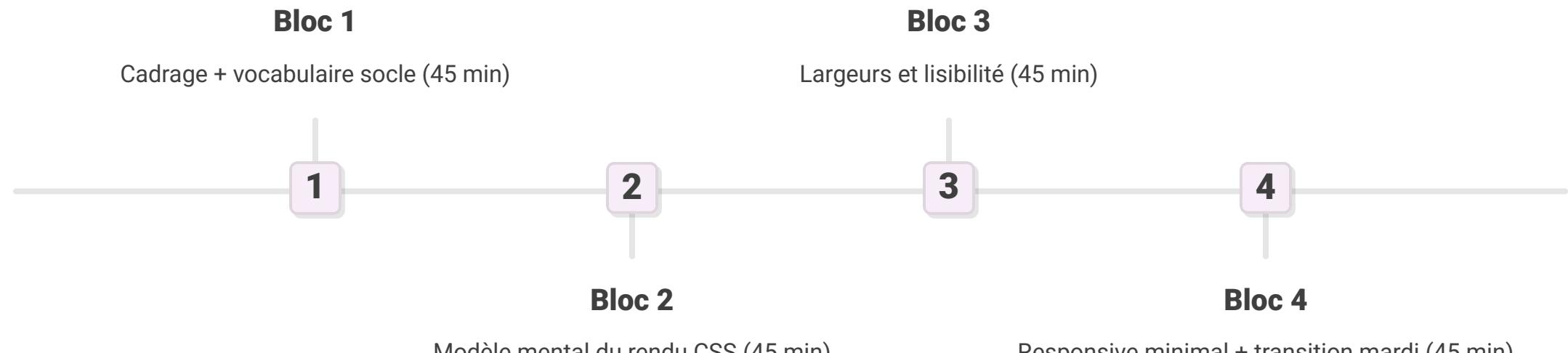
LUNDI MATIN

SAVOIRS

- Focus: display, largeurs robustes, media query minimale
- Durée indicative : 2 min

# Timing (durée indicative)

Durée totale: 3h (180 min)



# Objectifs de la séance

01

**Expliquer l'effet de display sur le flux**

02

**Distinguer width et max-width**

03

**Utiliser % et rem dans des cas simples**

04

**Lire et écrire @media (max-width: 768px)**

05

**Identifier un risque de régression CSS**

Durée indicative : 4 min

# Cadre et limites

## Contraintes

- Aucun livrable apprenant à remettre ce lundi
- Évaluation Moodle uniquement (shortanswer, matching, cloze)

Durée indicative : 4 min

## Périmètre

- Pas de flex dans cette séance
- Focus : display, largeurs, media query

## Règle essentielle

Règle: petit changement → vérification mobile + desktop

# Vocabulaire socle (partie 1)

 CONCEPTS FONDAMENTAUX

`display: <value>;`

Définit le comportement de rendu dans le flux

**display: default;**

Ordre d'affichage standard des éléments  
HTML sur la page

**display: block;**

Rupture de ligne. Empile verticalement.  
Largeur 100%

**display: inline;**

Suit le flux texte. Ignore width/height

**display: inline-block;**

En ligne mais accepte dimensions

**display: none;**

Masque l'élément et retire sa place du flux

Durée indicative : 8 min

# Vocabulaire socle (partie 2)

---

## width vs max-width

**width** : largeur fixe/relative

**max-width** : limite haute flexible

## % et rem

**%** : relatif au parent

**rem** : relatif à racine

## media query

@media : règles conditionnelles  
selon taille écran

## breakpoint

Seuil où la mise en page change



## régression

Bug introduit involontairement par  
modification CSS

### Objectif :

"Je change sans casser ailleurs"

Durée indicative : 10 min

# Situation de départ

BLOC 2 – MODÈLE MENTAL



## Sur desktop

Une page est lisible sur desktop



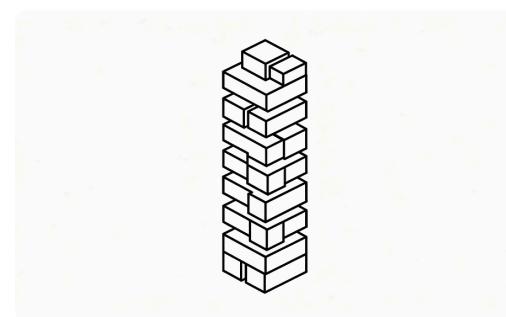
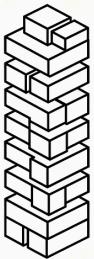
## Sur mobile

un bloc sort de l'écran, un élément disparaît

**Question :** quel rôle joue display dans ces effets?

Durée indicative : 5 min

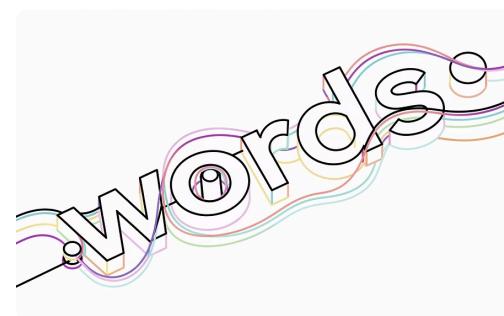
# Notion – Comment le navigateur affiche



## display: block

Rupture de ligne. S'empile verticalement

Ex : <div>, <p>, <h1>



## display: inline

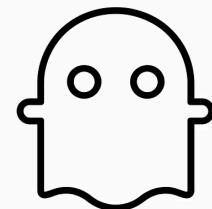
Suit le flux texte. Pas de dimensions

Ex : <span>, <a>, <strong>



## display: inline-block

En ligne mais accepte dimensions  
Usage : boutons, badges, grilles simples



## display: none

Complètement retiré du rendu

Usage : masquer temporairement

Durée indicative : 8 min

# Méthode – Lire le flux sans se perdre

## DÉMARCHE SYSTÉMATIQUE



### Repérer l'élément touché

Inspecter dans DevTools



### Vérifier sa valeur `display`

Consulter valeur actuelle



### Modifier une seule propriété

Une propriété à la fois



### Comparer mobile et desktop

Tester mobile ET desktop

Durée indicative : 8 min

# Exemple guidé – Avant / après sur display

Avant :

```
.title {  
  display: inline;  
}
```

Cas de test :

```
.helper {  
  display: none;  
}
```

Après :

```
.title {  
  display: block;  
}
```

Variante :

```
.badge {  
  display: inline-block;  
  width: 5rem;  
}
```

Durée indicative : 10 min

# Micro-activité – Diagnostic express

## EXERCICE PRATIQUE



### Symptôme A

élément disparu

Quelle correction ?



### Symptôme B

structure cassée

Quelle correction ?

## Formuler votre diagnostic

01

**Symptôme**

03

**Test**

Durée indicative : 10 min

02

**Hypothèse**

04

**Résultat**

# Synthèse – Bloc 2

## POINTS CLÉS

**display structure le rendu**

Contrôle ruptures de ligne et dimensions

**none n'est pas "invisible", c'est "retiré"**

Retiré du flux, pas simplement caché

**Toujours tester l'effet sur deux tailles d'écran**

Mobile + desktop systématiquement

Durée indicative : 4 min

# Situation de départ

BLOC 3 – LARGEURS

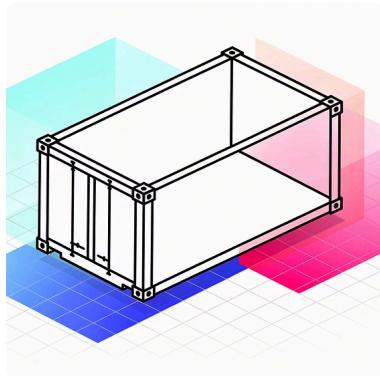
**Une carte en width: 900px déborde sur mobile**

**Une image trop large casse la mise en page**

Question : quelles largeurs rendent le layout robuste?

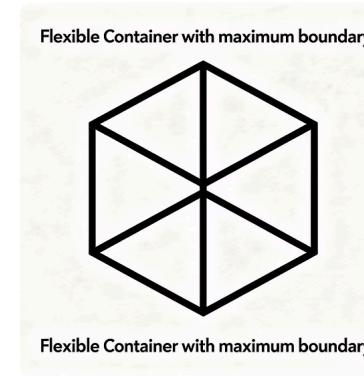
Durée indicative : 5 min

# Notion – `width` vs `max-width`



**width**

largeur visée



**max-width**

limite haute de sécurité

## Combinaison utile

```
.container {  
    width: 100%;  
    max-width: 60rem;  
}
```

Durée indicative : 10 min

# Notion – `%` et `rem`

% : relatif au parent

rem : relatif à la racine, bon pour lisibilité

Objectif : éviter rigidité et débordements

Durée indicative : 8 min

# Exemple guidé – Correction de débordement

💡 RÉSOLUTION

## Avant : Rigide

```
.card {  
    width: 900px;  
}
```

```
.hero-image {  
    width: 1200px;  
}
```

## Après : Robuste

```
.card {  
    width: 100%;  
    max-width: 40rem;  
}
```

```
.hero-image {  
    width: 100%;  
    max-width: 100%;  
}
```

Durée indicative : 10 min

# Activité guidée – Largeurs robustes



EXERCICE

Corriger 3 cas: carte, image, bloc texte.

## Cas 1 : Carte produit

```
.product-card {  
    width: 800px;  
}
```

Règle choisie : width: 100%; max-width: 50rem;

Raison : Assure que la carte ne dépasse pas 50rem sur les grands écrans et s'adapte aux plus petits.

## Cas 2 : Bannière

```
.banner-img {  
    width: 1400px;  
}
```

Règle choisie : width: 100%; max-width: 100%;

Raison : Permet à l'image de toujours remplir l'espace disponible sans créer de débordement horizontal.

## Cas 3 : Texte

```
.text-content {  
    width: 1000px;  
}
```

Règle choisie : width: 100%; max-width: 45rem;

Raison : Maintient la lisibilité du texte sur desktop et s'ajuste pour les écrans mobiles.

**Critère :** Pas de scroll horizontal en mobile

Durée indicative : 8 min

# Synthèse – Bloc 3

💡 PRINCIPES

## Préférer max-width

Préférer max-width à largeur fixe quand possible.

Durée indicative : 4 min

## Vérifier cas limites

Vérifier cas limites (texte long, image large).

## Valeurs simples

Garder des valeurs simples et justifiables.

# Notion – Responsive minimal

BLOC 4 – MEDIA QUERY

## Une seule media query aujourd'hui

- Cible imposée: @media (max-width: 768px)

## Le breakpoint 768px

```
@media (max-width: 768px) {  
    /* Écrans ≤768px */  
}
```

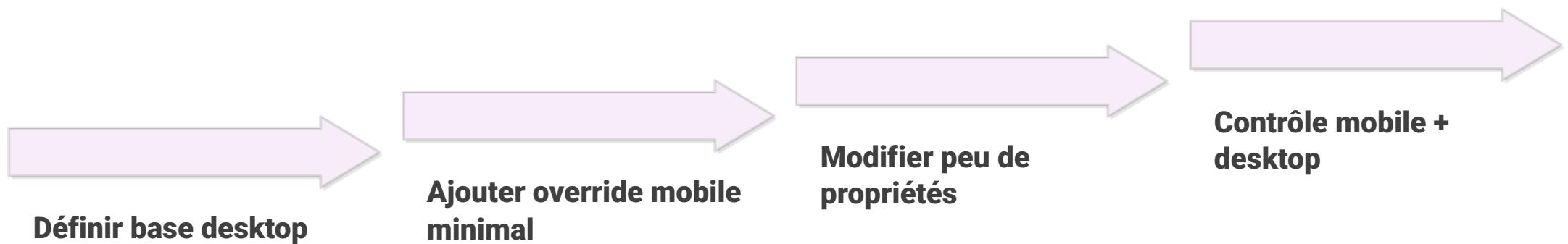
Sépare mobile/tablette de desktop

Les règles internes surchargent la base

Durée indicative : 8 min

# Méthode responsive – Adapter sans régression

 DÉMARCHE



Durée indicative : 8 min

# Exemple – Zone existante

APPLICATION

## Base desktop

```
.info-panel {  
    width: 42rem;  
    padding: 2rem;  
}
```

## Override mobile

```
@media (max-width: 768px) {  
    .info-panel {  
        width: 100%;  
        padding: 1rem;  
    }  
}
```

---

### Vérification :

- lisible mobile
- stable desktop

Durée indicative : 10 min

# LA MECANIQUE DU LAYOUT CSS : DU FLUX À L'ADAPTATION

CONTENU  
HTML BRUT



## LE MOTEUR DE RENDU (CSS)

### 1. LE COMPORTEMENT (La propriété 'display')

**block**

Prend toute la largeur, empile verticalement.

**inline**

Suit le texte, ignore hauteur/marges verticales.

**inline-block**

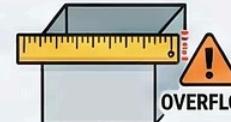
Hybride: s'aligne comme du texte, se dimensionne comme un bloc.

**none**

Invisible. N'existe plus dans le flux.

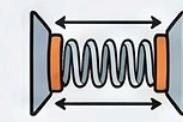
### 2. LE DIMENSIONNEMENT (Tailles et Unités)

#### RIGIDE VS SOUPLE



**width**

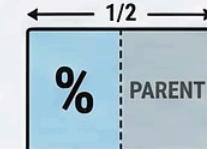
Taille fixe.  
Risque de débordement.



**max-width**

Taille maximum.  
S'adapte pour éviter l'overflow.

#### UNITÉS CLÉS



**%  
(Pourcentage)**

Relatif au conteneur parent.



Relatif à la taille de police racine (accessibilité).

### 3. L'ADAPTATION (Responsiveness)

Le CSS change les règles selon la taille de l'écran.

**block**

**inline-block**

**inline-block**

#### media query

Le CSS change les règles selon la taille de l'écran.



**breakpoint**

**768px**

Tablette

**1024px**

Desktop



Desktop



Mobile

#### RÉGRESSION

Attention!

Corriger une vue peut casser l'autre. Tester partout.