



# **RESPONSIVE DESIGN**

# RESPONSIVE DESIGN

- Préambule & principe du responsive design
- le média pour le print
- les deux informations à préciser dans une page web
- Le mode flexBOX
- les différents types de responsive design
- Les points de cassures
- Le cas des images & vidéos
- La gestion du menu en responsive design
- Les unités de mesures relatives
- Les grid layout CSS

The background of the slide is a repeating pattern of tropical foliage. It includes large monstera leaves with prominent veins, smaller heart-shaped leaves, and clusters of small, bright green berries or flowers. The pattern is rendered in a light gray line-art style with some green accents.

RWD

# INITIATION AUX MEDIA QUERIES

OPTIMISER SON SITE QU'IMPORTE SON SUPPORT

# RESPONSIVE DESIGN

## A LA DÉCOUVERTE DES MEDIA QUERIES

### MEDIA PRINT

→ 1 page HTML mais 2 feuilles de style  
1 pour l'écran  
1 pour l'impression

# LE MEDIA PRINT

MEDIA PRINT

À LA DÉCOUVERTE DES MEDIA QUERIES

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Menu HTML</title>
    <meta charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="styles/stylemenu.css" rel="stylesheet">
    <link href="styles/impression.css" media="print" rel="stylesheet">
  </head>
```

# LE MEDIA PRINT

MEDIA PRINT

A LA DÉCOUVERTE DES MEDIA QUERIES

```
@page {  
    size:21.0cm 29.7cm;  
    margin-top:2.1cm;  
    margin-bottom:1.9cm;  
    margin-left:2cm;  
    margin-right:2cm;  
}  
  
.saut2page{  
    /* saut de ligne apres le bloc*/  
    page-break-after : always;  
    /* saut de ligne avant le bloc ==> page-break-before : always;*/  
}
```

# RESPONSIVE DESIGN

## TRAVAUX PRATIQUES

**FAIRE UNE PAGE WEB AVEC UN  
BACKGROUND ET DES COULEURS  
PUIS UNE CSS POUR L'IMPRESSION**



The background of the slide is a repeating pattern of botanical illustrations. It includes various types of leaves, such as large monstera leaves with prominent veins, smaller fern fronds, and clusters of small, round berries. The illustrations are rendered in a minimalist style using black outlines and light green accents on a white background.

RWD

# LES DIFFÉRENTES APPROCHES DU RWD

OPTIMISER SON SITE QU'IMPORTE SON SUPPORT



# RESPONSIVE DESIGN

4

approches pour du responsive Design

→ 4 approches différents

# RESPONSIVE DESIGN

1

Deux sites

→ DEUX SITES DIFFÉRENTS ! avec un sous domaine



# RESPONSIVE DESIGN

2

Le liquid

→ le LIQUIDE DESIGN



# RESPONSIVE DESIGN

3

l'adaptatif

→ l'adaptatif

# RESPONSIVE DESIGN

## 4

le Rwd

→ adaptif + liquid = le responsive design :

The background of the slide is a repeating pattern of tropical plants. It includes large monstera leaves, smaller heart-shaped leaves, and clusters of small green berries. The pattern is rendered in a light grey line-art style with some green accents. A black rectangular box with a thin white border is centered on the slide, containing the main text. A thick green horizontal bar is positioned above the black box, and a thick green vertical bar is on the right side of the black box.

**AVANT DE COMMENCER**

# **AVANT DE COMMENCER**

OPTIMISER SON SITE QU'IMPORTE SON SUPPORT





SOMMAIRE

# RESPONSIVE DESIGN

# RESPONSIVE DESIGN

## 2

éléments à insérer en html

→ DEUX éléments doivent être précisés en HTML

# RESPONSIVE DESIGN

L'attribut lang="fr"

→ mettre l'attribut lang="fr" dans la balise html

```
1 <!doctype html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title>Methode 1 - TP F
```

# RESPONSIVE DESIGN

L'attribut meta="viewport"

→ le meta "viewport" dans la partie <head> de ma page

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

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

# RESPONSIVE DESIGN

Chapitre n°4

## DEUX FAÇONS DE PRÉCISER LES MEDIA QUERIES...

The background of the slide is a repeating pattern of botanical illustrations. It includes various types of leaves, such as large monstera leaves with prominent veins, smaller fern fronds, and clusters of small, round berries. The drawings are in a light grey line-art style, with some elements highlighted in a vibrant lime green color. A central white rectangular area with a black border contains the main text, and a horizontal lime green bar is positioned above the text box.

**DEUX MANIÈRES POUR LES MEDIAS QUERIES**

# **2 MANIÈRES DE DÉCLARER LES MEDIAS QUERIES**

OPTIMISER SON SITE QU'IMPORTE SON SUPPORT



# RESPONSIVE DESIGN

## 2

façons de préciser les media queries

DEUX manières pour préciser les media queries :

# RESPONSIVE DESIGN

## 1

l'attribut media dans la balise LINK

DEUX manières pour préciser les media queries :

# RESPONSIVE DESIGN

1ère méthode : 1 css par media

## MEDIA QUERIES

→ 1 page HTML mais 3 feuilles de style  
1 CSS orange  
1 CSS bleue  
1 CSS vert

# RESPONSIVE DESIGN

## 2

l'attribut media dans la balise LINK

DEUX manières pour préciser les media queries :

# RESPONSIVE DESIGN

2ème méthode : 1 seule CSS

## MEDIA QUERIES

→ 1 page HTML mais 3 feuilles de style  
1 CSS orange  
1 CSS bleue  
1 CSS vert

# RESPONSIVE DESIGN

## Méthode 3

### Les points de rupture





LES TROIS POINTS DE CASSURE

# LES 3 POINTS DE CASSURE

OPTIMISER SON SITE QU'IMPORTE SON SUPPORT

# RESPONSIVE DESIGN

# 3

les trois principaux points de rupture

Pour Smartphone  
Pour Tablette  
Pour écran

# RESPONSIVE DESIGN

## CODE

```
1
2  /* ***** POUR SMARTPHONE ***** */
3  @media screen and (max-width:767.98px) {
4
5  }
6  /* ***** POUR TABLETTE ***** */
7  @media (min-width: 768px) and (max-width:991.98px) {
8
9  }
10
11 /* ***** POUR ECRAN ***** */
12 @media screen and (min-width: 992px) {
13
14 }
```

The background of the slide is a repeating pattern of tropical foliage. It includes large monstera leaves, smaller heart-shaped leaves, and clusters of small green berries. The pattern is rendered in a light gray line-art style with some green accents. A solid lime green horizontal bar is positioned above the main text box.

**LE CAS DES IMAGES & VIDÉOS**

# **LE CAS DES IMAGES & VIDÉOS**

POUR LES RENDRE RESPONSIVE DESIGN

# LE CAS DES IMAGES & DES VIDÉOS

```
img,  
video{  
    max-width: 100%;  
    height: auto;  
}
```



The background of the slide is a repeating botanical pattern. It features various types of leaves, including large monstera leaves with prominent veins, smaller fern fronds, and clusters of small, round berries. The pattern is rendered in a light gray line-art style with some elements in a vibrant lime green color. A large, white rectangular area with a black border is centered on the slide, containing the main text. This area is framed by a thick lime green border on its top, bottom, and right sides, and a thinner one on the left side.

**LES MENUS EN RESPONSIVE DESIGN**

# **LA GESTION DES MENUS**

POUR LES RENDRE RESPONSIVE DESIGN



# RESPONSIVE DESIGN

<https://bitsofco.de/the-target-trick/>

<https://jenseign.com/apprendre-html-css/pratique-exemple/menu-burger/>

The background of the slide is a repeating botanical pattern. It features various types of leaves and flowers in a light gray line-art style, interspersed with small clusters of bright green berries. The pattern is dense and covers the entire slide area.

**LES UNITÉS RELATIVES**

# **LES UNITÉS RELATIVES**

POUR LES RENDRE RESPONSIVE DESIGN

# LES UNITÉS RELATIVES

## Les pixels

C'est l'unité la plus utilisé

Un pixel c'est un point.

## Les em

par défaut la valeur de 1 équivaut à 100% de la taille par défaut

Les em sont relatifs à leur parent

## Les rem

Par défaut, la taille de la police du Body est de 16px. 1rem (sans espace) équivaut donc à la valeur de la racine (1er bloc parent);

Pour se mettre à gérer ses tailles en Rem, rien de compliqué. Il suffit de définir la racine du document à 62.5% sur la balise html puis de gérer ses tailles en Rem en sachant que 10px correspond à 1.0rem.

## Les vh et vw

C'est l'équivalent des pourcentage mais par rapport au viewport. vh:100% permet par exemple de parfaitement centrer verticalement un élément.

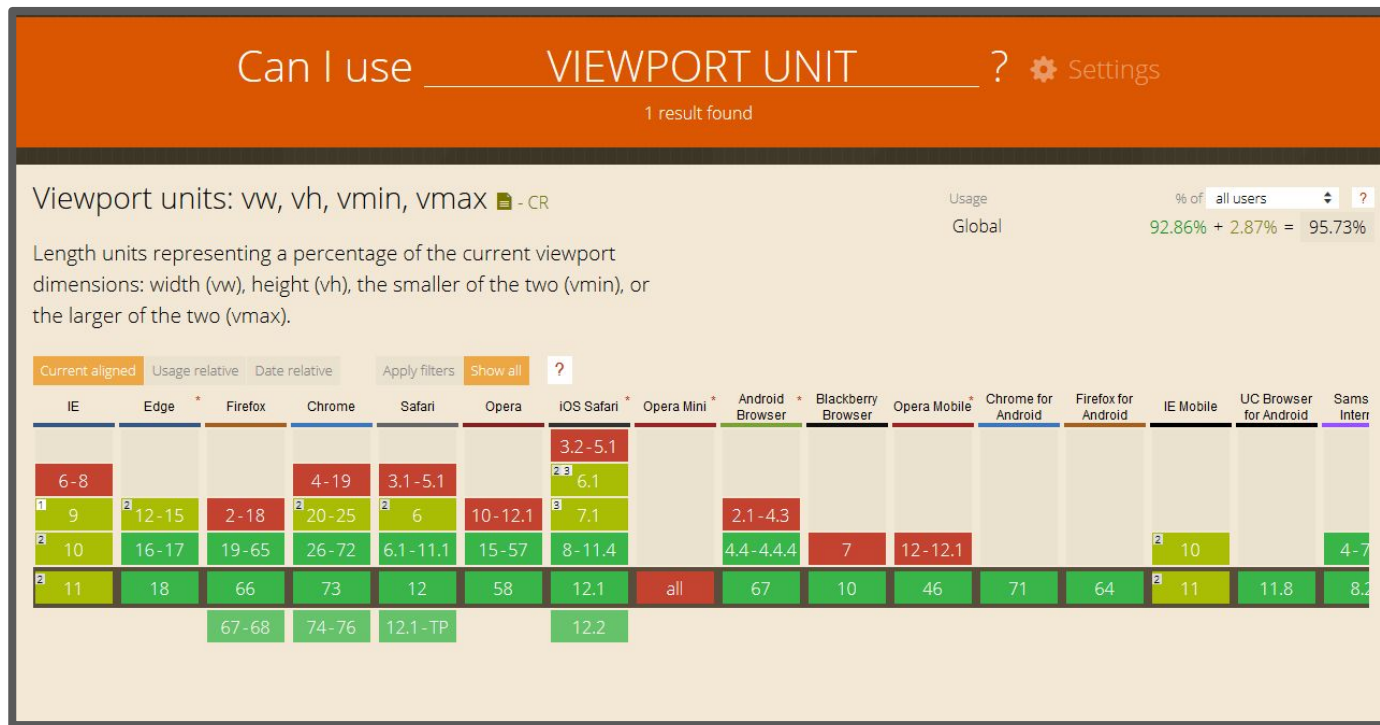
# LE REM

## LE REM

```
html {  
    font-size: 62.5%; /* Ré-équilibrage du Rem face au Pixel pour des calculs simples */  
}  
/* Exemple d'application */  
p {  
    font-size: 12px; /* Pour les navigateurs qui ne supportent pas le Rem */  
    font-size: 1.2rem; /* Pour les navigateurs « Responsive » */  
}
```

# VH & VW

VH et VW



# VH & VW

VH et VW

**1 vh = 1% du viewport en hauteur**

**1 vw = 1% du viewport en largeur**

## **Exemple:**

Si l'on souhaite centrer verticalement  
un texte dans un bloc en mode flex  
on utilisera :  
height: 100vh

```
.flex{  
  display: flex;  
  width: 100%;  
  height: 100vh;  
  justify-content: center;  
  align-items: center;  
  text-align : center;  
}
```





POSITIONNEMENT DES BLOCS

# LE MODE FLEX BOX

POUR LES RENDRE RESPONSIVE DESIGN

# LE MODE FLEX BOX

01

**display: flex**

02

**flex-wrap**

03

**order: 1**

04

**justify-content**

05

**align-items**

06

**content-items**

mais aussi

`flex-grow : (par défaut 0)`  
`flex-shrink : (par défaut 0)`  
`flex-basis : (par défaut auto)`



# LE MODE FLEX BOX

01

```
.container{  
    display: flex;  
}
```

**display: flex;**

Il sera possible de changer l'ordre des blocs avec la propriété css : order

Deux propriétés importantes pour centrer horizontalement ou verticalement...en mode flex

# LE MODE FLEX BOX

02

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
}
```

**flex-wrap : wrap**  
*(par défaut : nowrap)*

Deux propriétés importantes pour  
centrer horizontalement ou  
verticalement...en mode flex

# LE MODE FLEX BOX

# 03

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
}  
.col1{  
  width: 200px;  
  order: 2  
}  
.col2{  
  width: 200px;  
  order: 1  
}
```

**align-items**

sera possible de changer l'ordre des blocs avec la propriété css : order

Deux propriétés importantes pour centrer horizontalement ou verticalement...en mode flex

```
justify-content: center;  
align-items: center;
```

# LE MODE FLEX BOX

04

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: center;  
}
```

**justify-content:**  
**center**

Il sera possible de changer l'ordre  
des blocs avec la propriété css :  
order

Deux propriétés importantes pour  
centrer horizontalement ou  
verticalement...en mode flex

**justify-content: space-evenly | center | flex-start | flex-end**

# LE MODE FLEX BOX

05

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
  justify-content : center;  
  align-items: center;  
}
```

**align-items : center;**

Il sera possible de changer l'ordre des blocs avec la propriété css : **order**

Deux propriétés importantes pour centrer horizontalement ou verticalement...en mode flex

**align-content: space-evenly | center | flex-start | flex-end**

# LE MODE FLEX BOX

## 06

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
  justify-content : center;  
  align-items: center;  
  align-content : center;  
}
```

**align-content :**  
**center**

Il sera possible de changer  
l'ordre des blocs avec la  
propriété css : order

Deux propriétés  
importantes pour centrer  
horizontalement ou  
verticalement...en mode  
flex

# LE MODE FLEX BOX

## 07

```
.container{
  display: flex;
  flex-wrap: wrap;
  justify-content : center;
  align-items: center;
}
.col1{
  flex-grow : (par défaut 0)
  flex-shrink : (par défaut 0)
  flex-basis : (par défaut auto)
}
.col2{
  flex-grow : (par défaut 0)
  flex-shrink : (par défaut 0)
  flex-basis : (par défaut auto)
}
```

Le mode flex se place dans le bloc `.container`

Il sera possible de changer l'ordre des blocs avec la propriété css : `order`

Deux propriétés importantes pour centrer horizontalement ou verticalement...en mode flex



POSITIONNEMENT DES BLOCS

# LES GRID LAYOUT

POUR LES RENDRE RESPONSIVE DESIGN



# LES GRID LAYOUT CSS

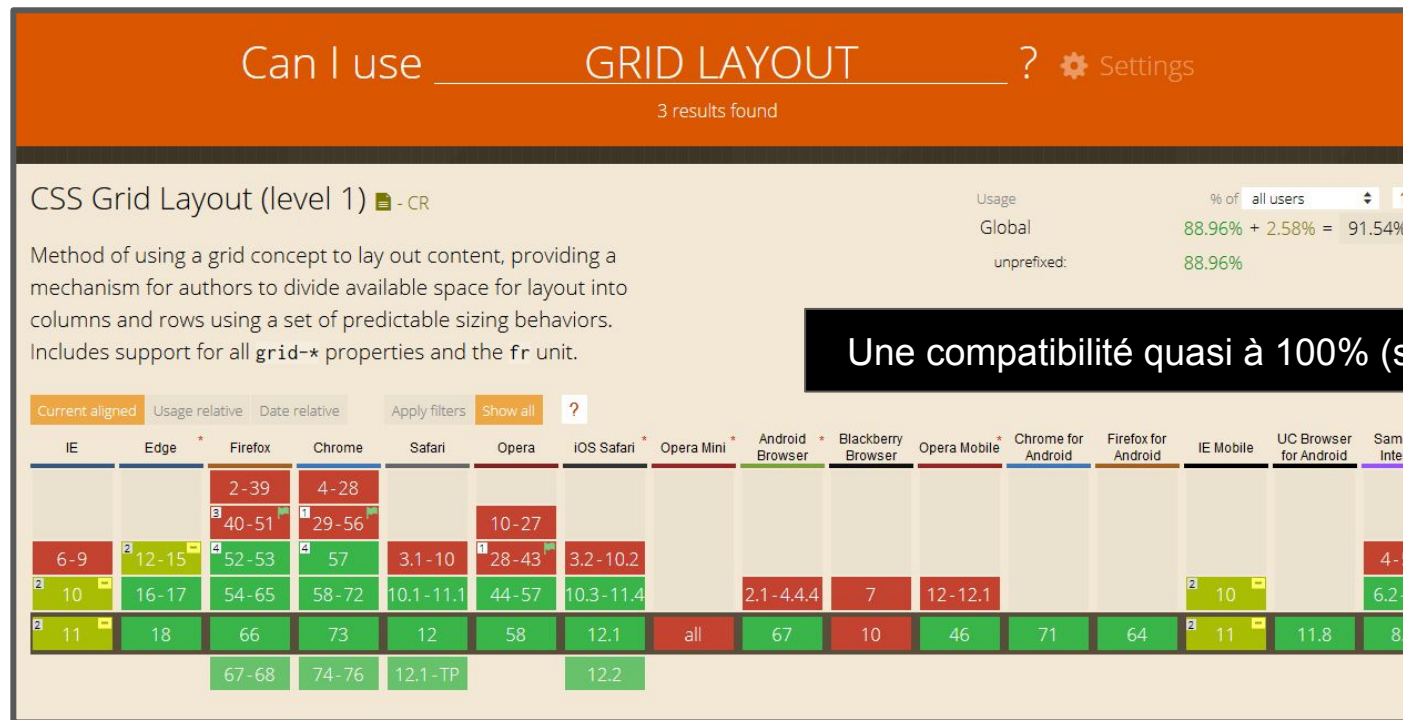
```
<body>
  <div class="grid">
    <div>- 1 -</div>
    <div>- 2 -</div>
    <div>- 3 -</div>
    <div>- 4 -</div>
    <div>- 5 -</div>
    <div>- 6 -</div>
    <div>- 7 -</div>
    <div>- 8 -</div>
    <div>- 9 -</div>
  </div>
</body>
```

```
.grid>* {
  padding: 5px;
  font-size: 24px;
  font-family: monospace;
  color: #1d1d1d;
  text-align: center;
  /*border: 1px solid black;*/
}

.grid>*:nth-child(even) {
  background-color: #bada55;
  /*border: 5px solid #caea00;*/
}

.grid>*:nth-child(odd) {
  background-color: #fafafa;
  /* border: 5px solid #eaeaea;*/
}
```

# LES GRID LAYOUT CSS



Une compatibilité quasi à 100% (sauf O mini)

# LES GRID LAYOUT CSS

## SOMMAIRE

```
ly>  
<div class="grid">  
  <div>- 1 -</div>  
  <div>- 2 -</div>  
  <div>- 3 -</div>  
  <div>- 4 -</div>  
  <div>- 5 -</div>  
  <div>- 6 -</div>  
  <div>- 7 -</div>  
  <div>- 8 -</div>  
  <div>- 9 -</div>  
</div>  
dy>
```

```
.grid{  
  display: grid;  
  grid-template-column : 33.33% 33.33% 33.33% ;  
  grid-template-rows: 33.33% 33.33% 33.33% ;  
  min-height: 100vh;  
  grid-gap : 20px; /* pour une marge entre chaque div */  
}
```



```
display: grid
```

```
display-template-columns : 20% 20% 20% 20%;
```

```
display-template-rows: 20% 20% 20% 20% 20%;
```

```
grid-column-gap: 3;
```

Cette méthode va permettre de préciser les éléments d'une grille directement via l'élément parent.

C'est l'élément parent qui va expliquer la construction de la grille

**L'avantage**, c'est que le bloc container va pouvoir définir les colonnes et l'apparence des lignes, et ça, c'est un gros point positif.

Ce n'est pas un nouveau système qui remplace le mode flex, mais c'est un système qui va pouvoir répondre à de nouvelles situations.

<https://caniuse.com/#search=GRID>



## Grille en pur CSS : une nouvelle unité

```
display: grid
```

```
display-template-columns : 200px 1fr;
```

```
grid-column-gap: 20px;
```

```
grid-column-gap: 20px;
```

```
grid-auto-flow: column;
```

```
.grid {
  display: grid;
  grid-template-columns: 20% 60% 20%;
  grid-column-gap: 20px;
}
```

```
body {
  margin: 0;
  padding: 0;
}

.grid > div {
  background-color: #ccc;
  border: solid 1px #999;
}

.grid {
  display: grid;
  grid-template-columns: 200px 1fr;
  grid-column-gap: 20px;
}
```

Plutôt que de mettre des pourcentages, on peut utiliser une nouvelle unité 1fr, il est possible de combiner une unité fractionnée avec des pixels, Exemple ci joint.

```
grid-template-columns: 1fr 1fr 1fr 1fr;
grid-template-rows: 1fr 1fr 1fr 1fr;
```

Pour l'espacement que l'on veut mettre entre chaque colonne



```
grid-template-columns: 1fr 1fr 1fr 1fr;  
grid-template-rows: 1fr 1fr 1fr 1fr;
```

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(2, 50%);  
  grid-template-rows: repeat(12, 1fr);  
  grid-column-gap: 20px;  
}
```

```
.child {  
  grid-column-start: 2;  
  grid-row-start: 2;  
}
```

```
.child {  
  grid-column-start: 1;  
  grid-column-end: span 2;  
  grid-row-start: 2;  
}
```

Plutôt que de mettre des pourcentages, on peut utiliser une nouvelle unité 1fr,... plutôt que de répéter cette unité par autant de fois que j'aurai de lignes ou de colonnes, je peux utiliser une nouvelle commande, la commande "repeat"



`display: grid``display-template-columns : 20%``display-template-rows: 20%`

Cette méthode est bien pratique et bien plus simple que le mode flottant, car on ne va agir que sur le bloc conteneur.

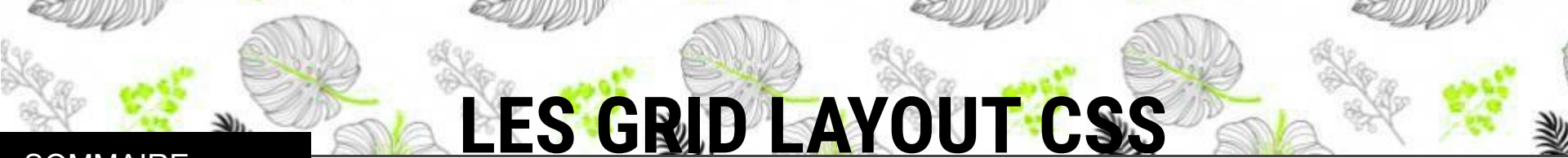
```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 3;
9 }
```

`grid-column-start: 3;`

Le gros avantage de ce nouveau format c'est qu'il nous **permet de piloter le comportement des blocs enfant directement depuis l'élément principal**, permettant ainsi d'éviter la soupe d'élément HTML

`grid-column-gap: 3;`

Pour l'espacement que l'on veut mettre entre chaque colonne



# LES GRID LAYOUT CSS





CONCLUSION

RÉSUMÉ

# RESPONSIVE DESIGN

## RÉSUMONS !

<https://html5-editor.net>