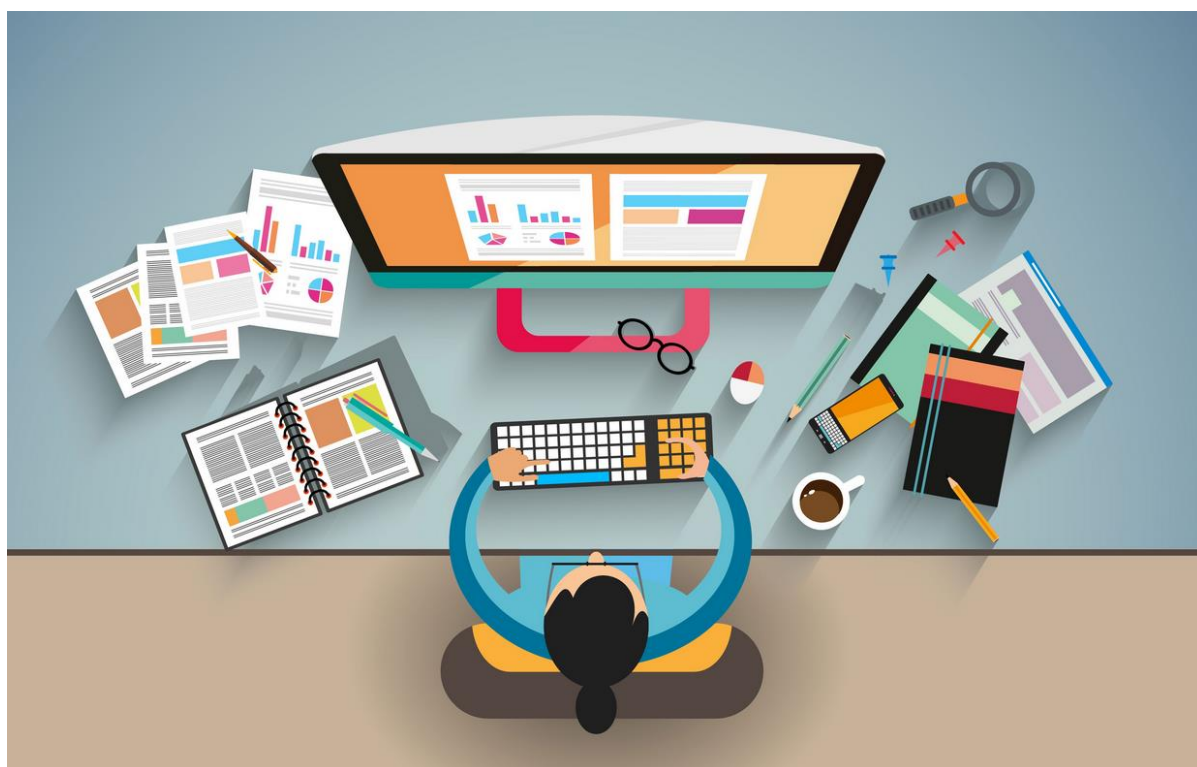


SUPPORT de COURS

- GRID SYSTEM | HTML CSS -



Formation organisée par le GRETA d'Aix

Contenus pédagogiques et supports : Emmanuel ADINARAYANIN

Le CSS GRID SYSTEM

Découverte

Exo02_01

grid lines vs grid cells

Ici il s'agit des lignes entre les colonnes ou bien de l'espace de colonne entre les lignes.
Une mise en page de 3 colonnes aura 4 gridLines et 3 gridCells

Analyser son code a l'aide de Firefox

pour l'instant seul firefox propose une aide a l'analyse des mise en pages basées sur grid system

Propriétés sur le grid-container parent

De même que pour le flex, la propriété display : grid; s'applique au parent et concerne uniquement les enfants directs.

Créer la grille

`display:grid;`

grid-template-columns

Cette propriété contiendra autant de valeurs que de colonnes (séparées par un espace) ou chaque valeur correspond à la largeur de la colonne: valeur valeur valeur ;
Ici toutes les unités de mesures sont acceptées même différentes par colonnes.

grid-template-row

Ici les %

toutes les unités sont acceptées mais dans le cas des %, la hauteur du grid container doit être renseignée

Unités disponibles pour grid-template-rows et grid-template-column

Unité fr

Avec les unités de différentes taille, il arrive souvent que le grid container laisse de l'espace disponible.

L'unité fr consiste à découper l'espace disponible en fractions dont le nombre dépendra du nombre de fr déclarés dans nos grid-templates-row et column

Unité auto

cette unité consiste a définir la dimension sur celle de son contenu

minmax(valeurMin , valeurMax)

Cette fonction permet de définir une fourchette de taille pour nos colonnes ou lignes très utiles pour l’affichage responsif.

Exo02_06

repeat(repetition , largeur)

```
repeat(5, 1fr)
/* Cette instruction correspond à 1fr 1fr 1fr 1fr 1fr */
```

```
repeat(3, 1fr 50px)
/* Cette instruction correspond à 1fr 50px 1fr 50px 1fr 50px */
```

gap

il s’agit de la gouttière interne, il n’y a pas d’espaces sur les bords du grid-container. Pour ce faire il suffirait d’ajouter un padding

row-gap

gouttière Vert

column-gap

gouttière Hrz

Placement automatique

Exo 03_1

Lorsqu’il y a plus de grid-items que de colonnes déclarées dans le grid-container, le navigateur place ces grid-items automatiquement en fonction des propriétés ci dessous. On parlera pour ces colonnes supplémentaires de **“colonnes implicites”**

grid-auto-flow

- column
 - Ajouter une colonne supplémentaire s’il y a trop de grid-items
- dense
 - Dans le cas de trous dans la grille, dense remplit les espaces non assignés automatiquement
- row (default)
 - Ajoute une rangée supplémentaire

grid-auto-column

défini la largeur des colonnes implicites

grid-auto-row

défini la hauteur des colonnes implicites

Propriétés sur les grid-items enfants

Placement manuel

Exo 03_02

grid-column-start

numéro de debut de gridLine

grid-column-end

numéro de debut de gridLine

grid-column

Instruction raccourcie

```
grid-row : 1 / 3
```

grid-row-start

numéro de debut de gridLine

grid-row-end

numéro de debut de gridLine

grid-row

Instruction raccourcie

```
grid-row : 1 / 3
```

Il est possible de compter les lignes à partir de la fin, il suffira d'utiliser des valeurs négatives ... -1 correspondra à la dernière grid line, -2 à l'avant dernière, etc...

Nommer les gridLines

Au moment du **grid-template-row** insérer les gridlines en utilisant [nomDeGridLine]:

```
.grid-container{
    grid-template-columns:[debut] 1fr [milieu] 400px [fin];
}
header{
    grid-column: debut/fin;
}
```

Dans le cas où on utilise le mot clé end et start dans les nom de gridLines, on peut raccourcir ainsi:

```
.grid-container{
    grid-template-columns: [page-start] 1fr [milieu] 250px
    [page-start];
}
header, footer{
    grid-column: page;
```

Utiliser les zones de grilles (grid Area)

exo_03_03

grid-template-area

Sur le parent

ici les mots clés sont indépendants des noms de balises, c'est par simple commodité qu'on retrouve header main et footer.

De même la propriété écrite sur plusieurs lignes est un moyen de faciliter la lecture de la grille

```
.grid-container{
    grid-template-columns: 1fr 250px;
    grid-template-areas:
        "header header"
        "main aside"
        "footer footer";
}
```

grid-area

sur l'enfant

```
header{
    grid-area: header;
}

footer{
    grid-area: footer;
}
```

Span

Exo 03-5

S'applique à l'enfant

Utilisé dans le grid-row ou grid-columns

permet de spécifier un nombre de cellule à couvrir

```
// s'étale du gridline 1 sur 2 cellules
grid-row : 1 / span 2
// s'étale du gridline 2 sur 3 cellules
grid-row : 2 / span 3
// s'étale sur 2 cellules jusqu'au gridline 3
grid-row : span 2 / 3
```

Changer l'ordre

order

S'applique aux enfants

Cette propriété est par défaut à 0 (comme flexbox)

Les éléments avec le même n° d'ordre suivent le flux naturel

exo_03-06

- Créer 8 zones de grille 50px / 1fr
- placer le 3eme élément sur toute la largeur (1/-1)
- changer la valeur du 1er élt

```
order:1;
```

Aligner les éléments dans les gridCells

S'applique au parent

Exo 04_01

Les alignements en justify-items ou align-items placent l'enfant dans sa gridCell. La taille de l'enfant est basée sur celle du contenu sauf si l'alignement est **stretch** dans ce cas l'enfant occupe tout l'espace du gridCell

justify-items

Axe horizontal

- start
- end
- center
- stretch (default)

align-items

Axe vertical

- start
- end
- center
- stretch (default)

place-items

Cette propriété raccourcie permet de gérer l'axe Hrz et Vert en 1 seule ligne

```
place-items: stretch end;
/* alignementY alignementX*/
```

Aligner la grille dans son conteneur

S'applique au parent

Exo04_02

Dans le cas où la grille est plus petite que le grid container

justify-content

Axe horizontal

- start
- end
- center
- space-around
- space-between
- stretch (default)

align-content

Axe horizontal

- start
- end
- center
- space-around
- space-between
- stretch (default)

Aligner un seul élément dans la grille

Exo04_02

S'applique aux enfants

justify-self

- start
- end
- center
- stretch

align-self

- start
- end
- center
- stretch

Mise en page élaborée, concepts avancés

Gérer pied de page

Exo 05_01

Placer des blocs de footer avec menu de navigation peut se faire aisément en combinant flex et grid system

Gérer l'entête

Mobile-first

Exo 05_02

1. Réaliser une grille de 3 colonnes (100px 1fr 60px)
2. forcer le nav sur la 2ème ligne
3. centrer les éléments dans leurs cells
4. aligner la figure et le nav en stretch / corriger attribut HTML img
5. Placer le ul en grid-container et faire 3 colonnes : 3em 1fr 3em
6. les li commencent en 2
7. masquer le nav par défaut

Gérer l'entête

media query 768px

Exo 05_03

Pla

1. placer le header sur 1 ligne
2. placer entete sur 2 colonnes de 150 et la 3eme en 1fr
3. faire remonter le nav a côté du reste (réinitialiser le départ du nav sur auto)
 - a. aligner à droite
 - b. transformer le nav en flex

Gérer le contenu

Exo 05_04 : index.html

Superposer la légende à l'image

1. Définir figure en grille
2. placer img et figcaption dans même gridCell
3. aligner figcaption sur end
4. gérer hauteur du figCaption en survol (50%)
5. pour la transition, penser à la valeur de début pour height

Gérer le responsive

Exo 05_05 : index.html

Mobil-first

1. Gérer la grille du ul
2. placer un gap d'1em
3. définir une hauteur des lignes implicites à 300px

Mediaquerie 768px

1. Placer les cartes sur 2 colonnes

Gérer les cartes spéciales .grid-big .grid-small .grid-end

.big-grid

Cette carte doit occuper 2 gridcells en hauteur et une seule sur la dernière colonne

Mediaquerie 992px

passer en 3 colonnes et donner 2 colonnes par défaut aux li

.big-grid

Cette carte doit occuper: 1 colonne sur la dernière colonne et 2 lignes

.small-grid

Ce bloc commence à la premiere colonne sur 1 gridCell

.grid-end

Ce bloc doit être sur la dernière colonne

```
grid-column-end:-1
```

Gérer le placement automatique des nouvelles gridCells avec dense

ul{

```
grid-auto-flow:dense;
```


}

Centrer un site dans la page avec grid

Exo final : text.html

centrer l'article sans utiliser de largeur ni de marges auto:

1. Créer une grille de 3 colonnes pour le conteneur de l'article
1fr 992px 1fr
2. Placer l'article en colonne 2
3. Gérer la media query pour largeur 1200px

Créer une mise en page responsive sans mediaqueries

Exo final: folder.html