

## Exercice 1 :

structure d'une grille Définir les grilles suivantes, sur lesquelles on laisse les items se positionner implicitement et sans contrainte : 1. grille de 4 colonnes, différentes largeurs fixées en utilisant différentes unités classiques (px, %, rem) – prévoir 5 rangées de différentes hauteurs également fixées en utilisant différentes unités classiques. 2. grille de 4 colonnes, avec une colonne de largeur fixée, les 3 autres se répartissant l'espace restant à égalité, puis avec le rapport 3-2-1. 3. réduire le nombre de rangées explicitement prévues dans la structure à 3. Que se passe-t-il ? Quelle est la taille des rangées ajoutées aux 3 premières ? Faire en sorte que les rangées supplémentaires aient une taille fixée. 4. en utilisant repeat() changer la définition de la grille pour qu'elle soit formée de 5 colonnes de tailles identiques occupant tout l'espace disponible, et 3 rangées d'une hauteur de 8rem. Les rangées supplémentaires avoir une taille fixée à 4rem.

## Exercice 2: taille et placement implicite des items

Prévoir une grille de 5 colonnes identiques et des rangées de 5rem. Mettre une couleur de background sur les items pour mieux les visualiser. Dans un premier temps, on dimensionne différents items et on les laisse se positionner implicitement. Dans chaque cas, donner une couleur à l'item et expérimenter les différents modes de positionnement implicite (row, column, dense). 1. item 4 : 3 colonnes en largeur 2. item 6 : 2 colonnes en largeurs 3. item 7 : 3 rangées en hauteur 4. item 10 : 2 colonnes et 2 rangées

## Exercice 3: placement explicite d'items

On conserve la grille et les items dimensionnés de l'exercice précédent, et on ajoute des items placés explicitement.

1. item 16 : colonne 1 et rangée 3, taille 1 dans les 2 sens
2. item 13 : colonne 4, largeur 2, rangée 1 hauteur 3
3. item 8 : colonne 4, largeur 1, rangée 4 hauteur 3
4. item 9 : colonne 3, largeur 2, rangée 4 hauteur 1 – que se passe-t-il ?
5. item 20 : colonne 6, rangée 7

## Exercice 4 : minmax(), auto-fit

Faire une copie de votre fichier html et repartir avec un grille fraiche pour réaliser les cas suivants :

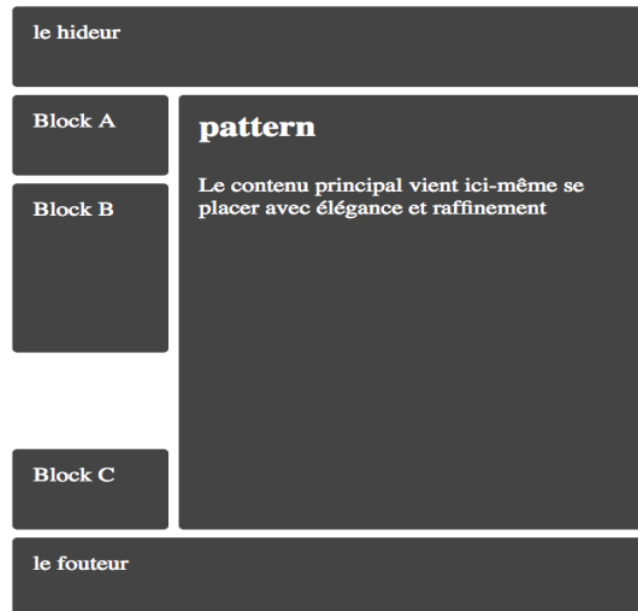
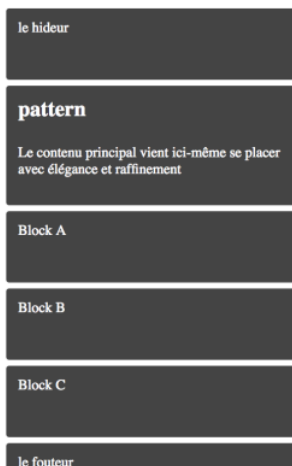
1. grille de 3 colonnes avec les caractéristiques suivantes :
  - a. colonne 1 : largeur fluide de 15 % avec un mini de 100px
  - b. colonne 2 : largeur variable mini 15rem maxi 25rem
  - c. colonne 3 : largeur fluide 30 % avec un mini de 15rem
2. 5 colonnes identiques de 100px de large au moins et 200px de large au plus,
3. 5 colonnes identiques de chacune 20 % de large et une largeur mini de 100px
4. 1 colonne de 25 % de large avec 1 mini de 150px, suivie de 4 colonnes identiques occupant l'espace restant,
5. 5 colonnes identiques occupant l'ensemble de l'espace, ayant une largeur mini de 10rem,

6. autant de colonnes que la place le permet, toutes identiques, ayant une largeur mini de 10rem.  
Le nombre de colonnes doit donc varier suivant la largeur du viewport.
7. différence entre auto-fit et auto-fill : construire un document avec 2 conteneurs de grilles, chacun contenant le même nombre d'items (max 5). Chaque grille est gérée comme celle du point 6 : colonnes identiques de largeur mini 10rem. Une utilise auto-fill et l'autre auto-fit : quelle est la différence entre les 2 ?

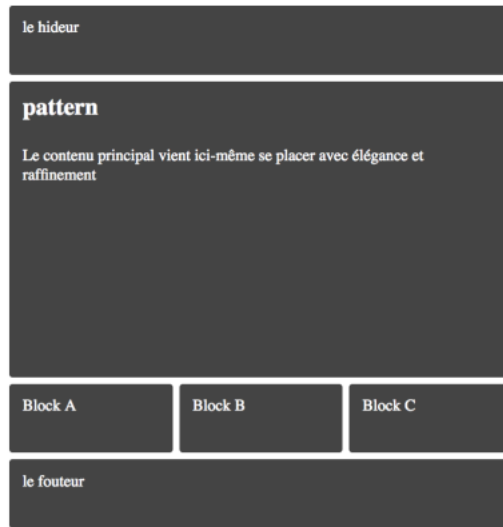
### Exercice 5 : patterns

Reproduire les patterns de layout ci-dessous.

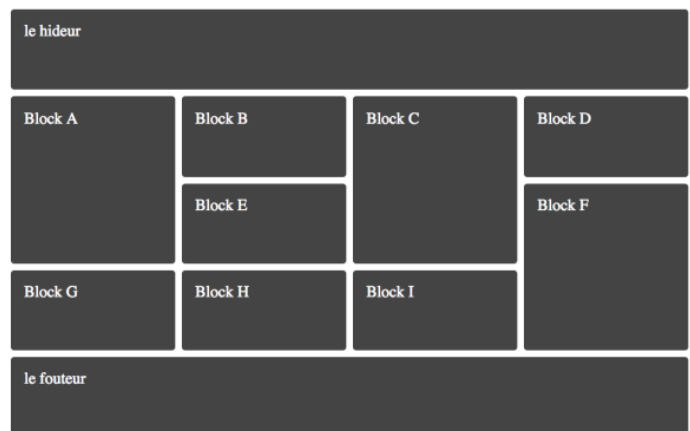
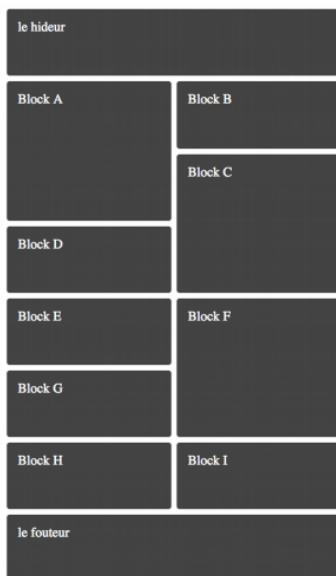
1. grille 1 colonne → 2 colonnes, sur petit écran la grille peut déborder vers le bas en fonction du contenu, mais chaque élément fait au moins 120px de hauteur ; sur écran large la grille occupe exactement la place disponible.



2. même caractéristiques mais disposition différente en écran large :



3. Autant de colonnes que possibles avec des items multi-rangées. Les colonnes et les rangées font au moins 120px/200px de hauteur / largeur.



4. contenu interne organisé de manière identique au cas 3, on ajoute en plus une colonne sur le côté dont la hauteur est toujours identique à celle du contenu interne.

