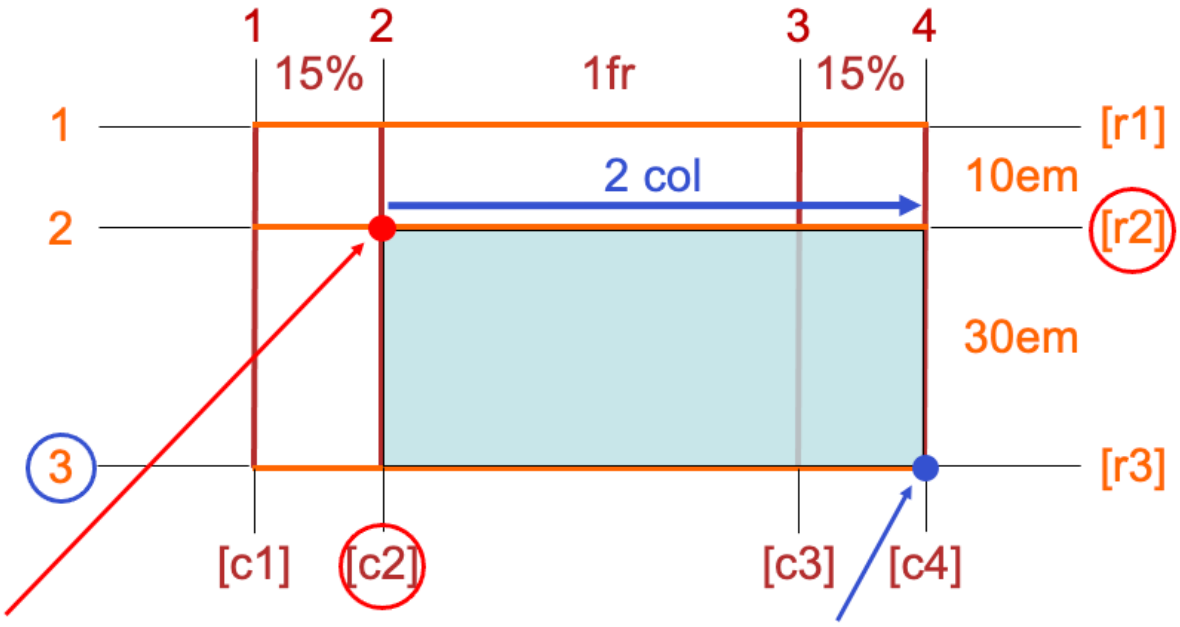



# Grid layout

Jean-Christophe **DUBOIS**
jean-christophe.dubois@univ-rennes.fr

# Mode d'affichage

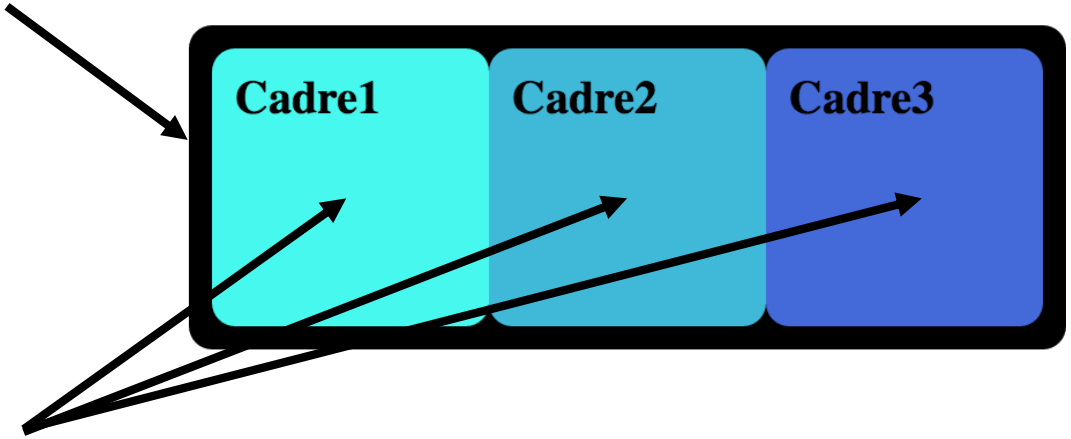
- `display :` changement de mode de positionnement/affichage

```
display: inline | block | inline-block
         | flex | inline-flex
         | grid                  /* grille bloc */
         | inline-grid           /* grille en ligne */
         | none;
```

- Déclaration d'un conteneur de type grille :
  - en mode bloc : `grid`
  - ou en ligne : `inline-grid`

# Grid-container vs Grid-items

- 2 types d'éléments :
  - un conteneur/container




  - des éléments/items
    - Cadre1
    - Cadre2
    - Cadre3

# Définition de la grille (conteneur)

- `grid-template-columns:` déclaration des colonnes

```
grid-template-columns: 25px 200px 25px
                     | 10% auto 10%
```

*3 colonnes dont 1 de taille adaptée au contenu*

```
                     | 50px 1fr 1fr
```

*1 colonne de 50px et 2 colonnes de dimension égale (fr: fraction de la place restante)*

```
                     | repeat(4, 25%)
```

*4 colonnes de dimension identique*

```
                     | repeat(2, 15% 10px)
```

*2 colonnes de 15% et 10px*

# Définition de la grille (conteneur)

- `grid-template-rows:` déclaration des lignes

```
grid-template-rows: 50px 100px 50px;
                  | minmax(50px,auto) 200px;
```

*1 colonne comprise entre 50px et une dimension adaptable et 1 colonne de 200px*

Possibilité de nommer les séparations

```
grid-template-rows: [r1] 25px [r2] 200px [r3]
```

# Définition de la grille (conteneur)

- `grid:` déclaration des colonnes et des lignes

```
grid: 50px 2fr 1fr / 80px 200px 50px;
```

Déclaration équivalente à :

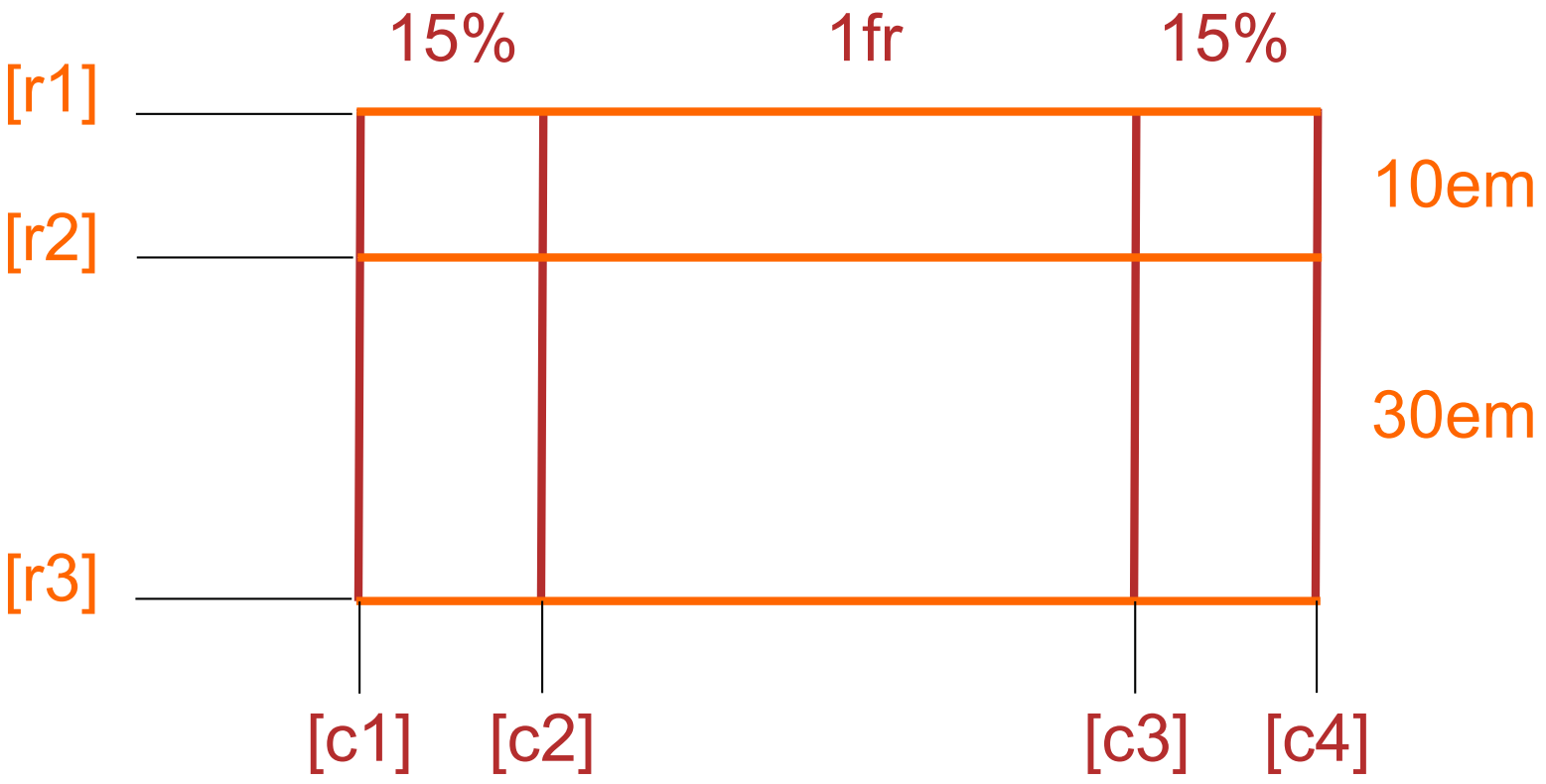
```
grid-template-columns: 50px 2fr 1fr;
```

et

```
grid-template-rows: 80px 200px 50px;
```

# Définition de la grille (conteneur)

- Grille définie par l'intermédiaire des dimensions des colonnes et des lignes




grid-template-columns: [c1] 15% [c2] 1fr [c3] 15% [c4];

# Gouttières de la grille (conteneur)

- `grid-column-gap:` espace entre les colonnes

```
grid-column-gap: 10px | 1em | 1%;
```

- `grid-row-gap:` espace entre les lignes

```
grid-row-gap: 10px | 1em | 1%;
```

Sans grip-row-gap, la valeur indiquée avec grid-column-gap est utilisée pour espacer les lignes

✓ Syntaxe raccourcie :

- `grid-gap:` espace entre les colonnes et les lignes

```
grid-gap: 10px 10px | 1em 2em;
```

# Définition par défaut (conteneur)

- `grid-auto-columns:` dimension des colonnes par défaut, si nécessaire

```
grid-auto-columns: 20% | 200px;
```

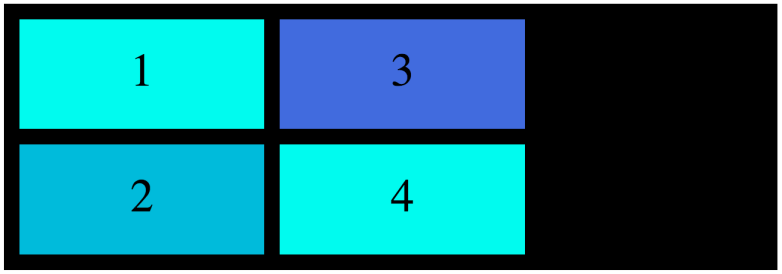
- `grid-auto-rows:` dimension des lignes par défaut, si nécessaire

```
grid-auto-rows: 20% | 200px;
```

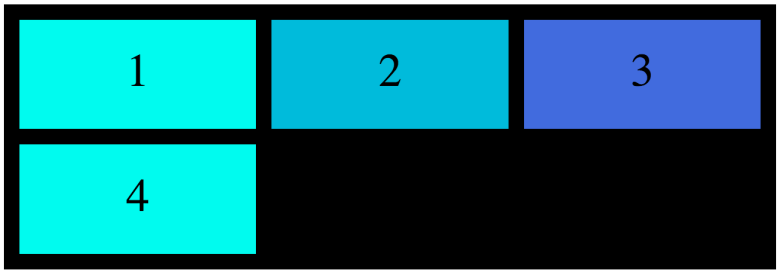
# Définition par défaut (conteneur)

- `grid-auto-flow:` remplissage auto. de la grille à l'aide des éléments. Seuls les emplacements vides sont utilisés.

```
grid-auto-flow: column;  // remplis. col. après col.
```

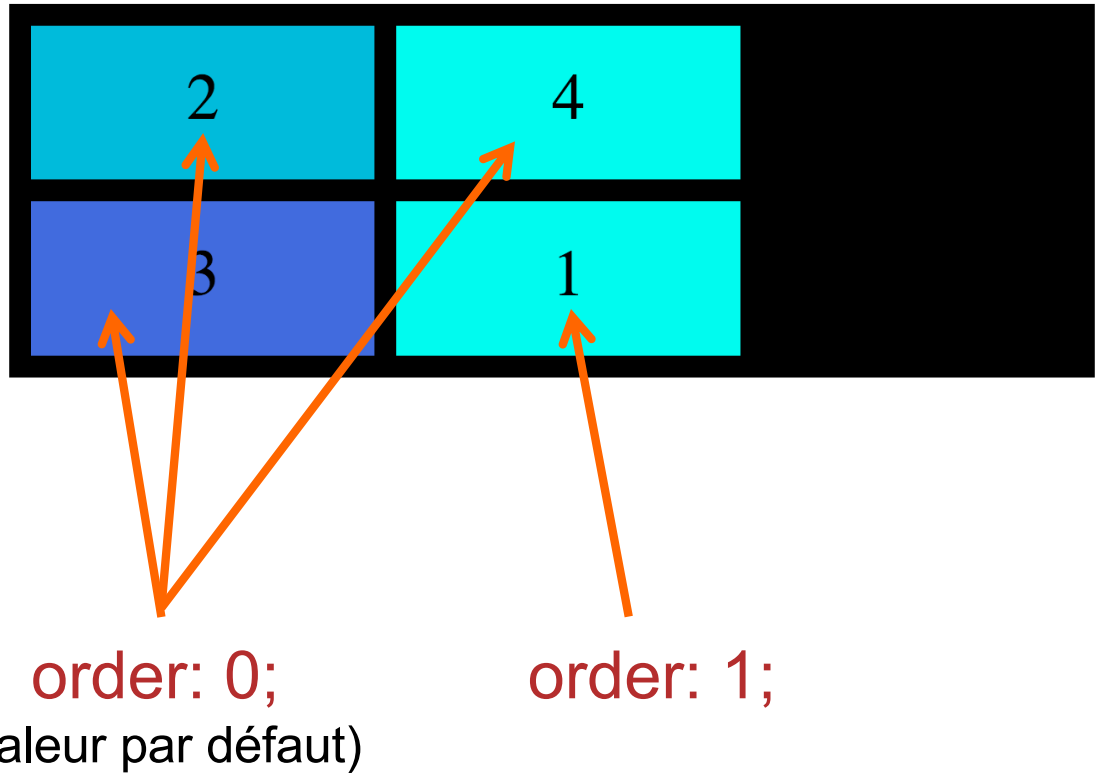



```
row                      // remplis. ligne après ligne
```



# Réordonnancement

- `order:` permet le réordonnancement des zones



# Début et fin des éléments

- `grid-column-start:`
  `grid-row-start:` début d'un élément

```
grid-row-start: 1;          // n° du séparateur
grid-column-start: c2;      // nom du séparateur
```

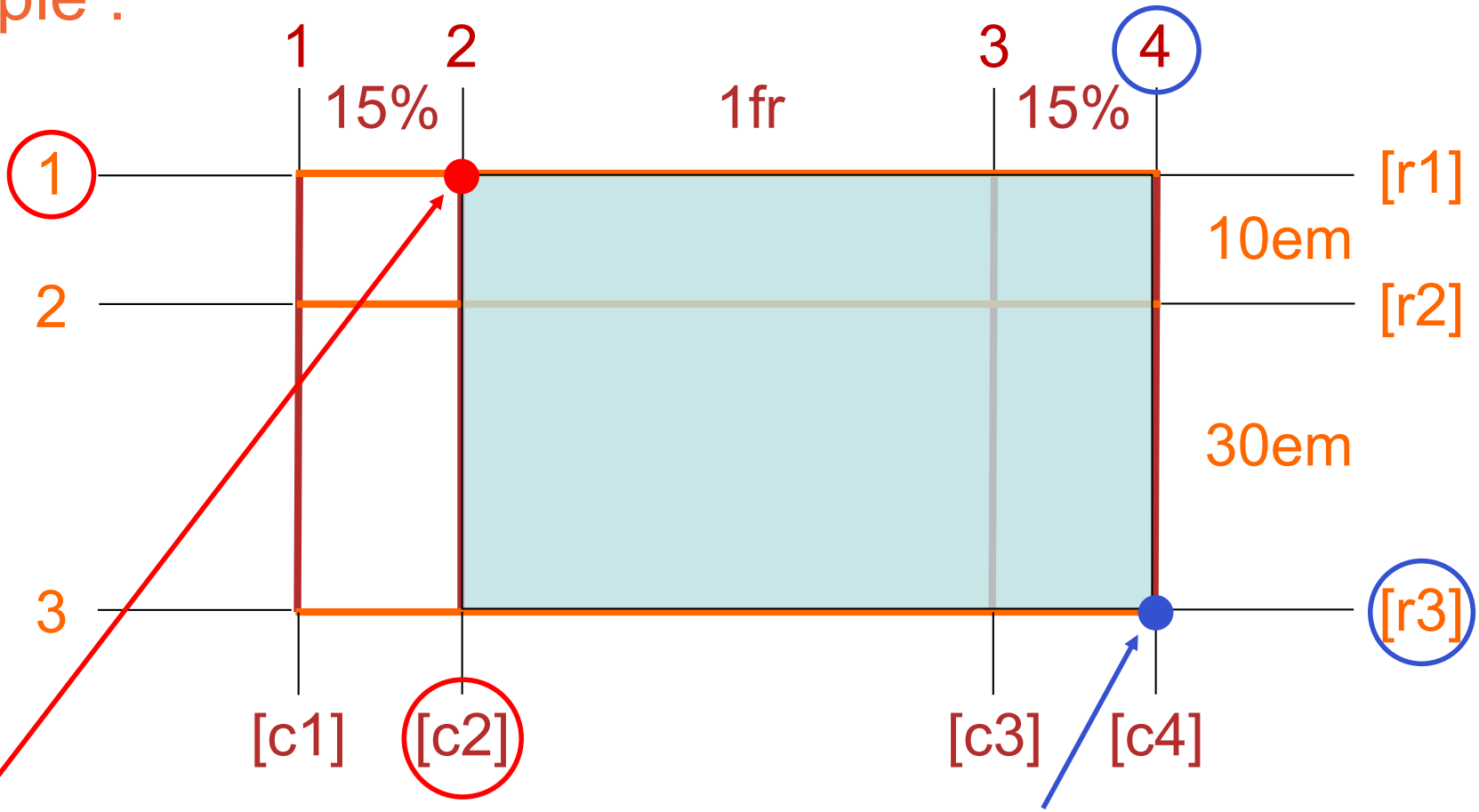
- `grid-column-end:`
  `grid-row-end:` fin d'un élément

```
grid-column-end: 4;         // n° du séparateur
grid-row-end: r3;           // nom du séparateur
```

Par défaut, un élément s'étend sur une seule ligne ou colonne

# Début et fin des éléments

- Exemple :




```
grid-row-start: 1;
grid-column-start: c2;
```

```
grid-row-end: r3;
grid-columns-end: 4;
```

# Dimension des éléments

- `grid-column-start:`
  `grid-column-end:` largeur de l'élnt

```
grid-column-end: span 2;        // élnt réparti sur 2 col.
                                // depuis sa position auto.
```

- `grid-row-start:`
  `grid-row-end:` hauteur de l'élnt

```
grid-row-end: span 3;           //  élnt réparti sur 3 lignes
                                // depuis sa position auto.
```

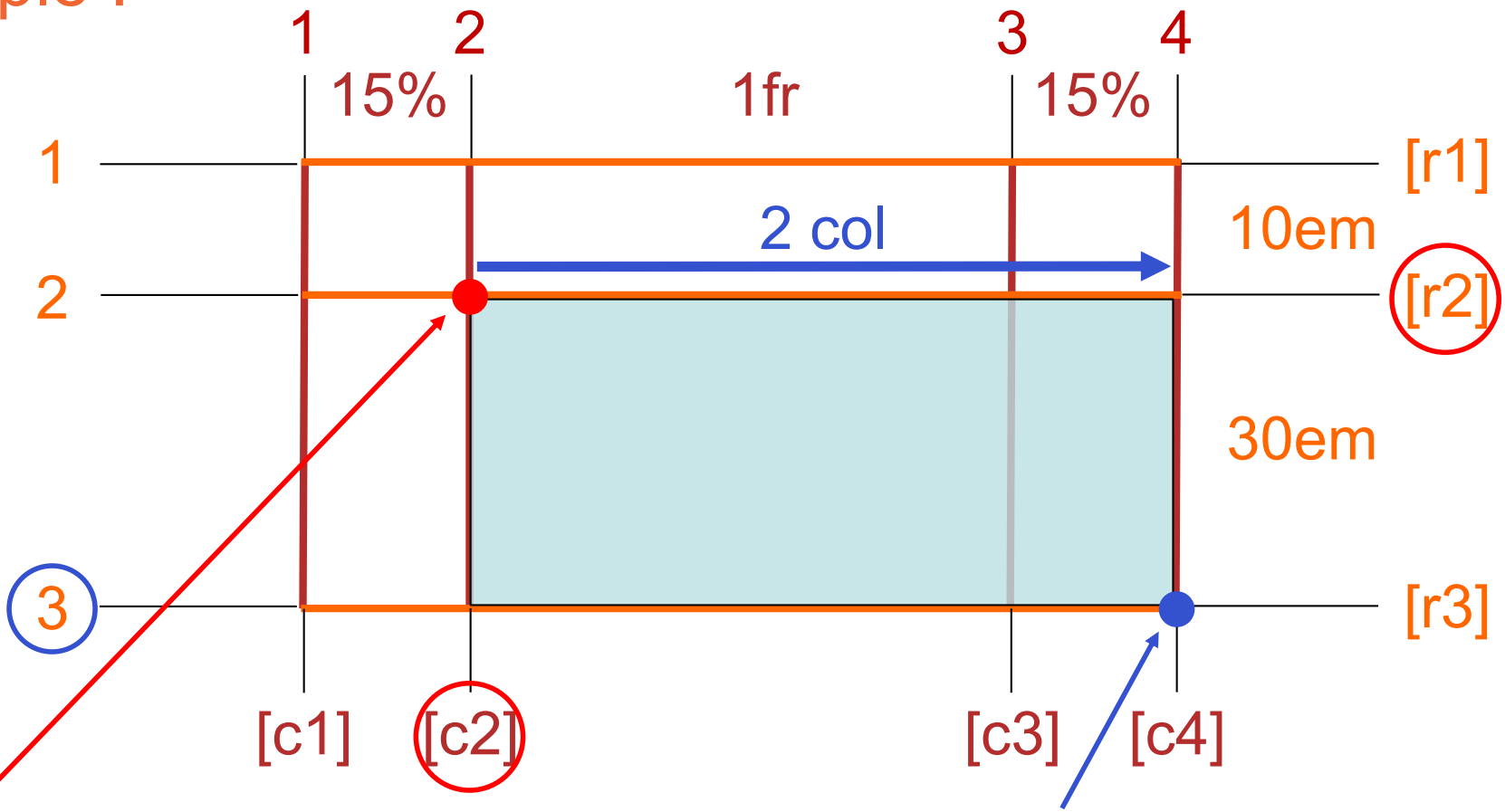
Syntaxe raccourcie :

- `grid-column:`
  `grid-row:` début et fin/dimension d'un élément

```
grid-row: 2 / 4;                // équiv. 2 / span 2
grid-column: 2 / span 3;        // équiv. 2 / 5
```

# Dimension des éléments

- Exemple :




```
grid-row-start: r2;
grid-column-start: c2;
```

```
grid-row-end: 3;
grid-columns-end: span 2;
```

# Définition de zones

- `grid-area:` définition d'une zone pour un élément en indiquant successivement la ligne et la colonne de début puis la ligne et la colonne de fin.

```
grid-area: 1 / 2 / 4 / 6; // 2 notations équiv.
                  Déb.Lig / Déb.Col / Fin Lig / Fin Col
```

`grid-area` est un raccourci des commandes `grid-row` et `grid-column`.

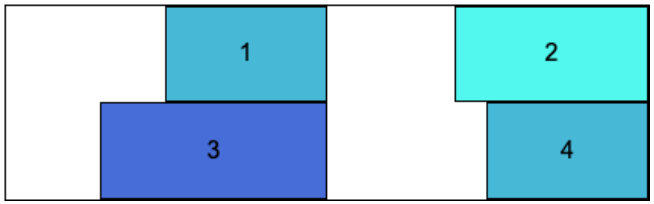
`z-index` repositionne les zones qui se chevauchent

https://cssgridgarden.com/#fr

# Alignement des éléments dans la grille

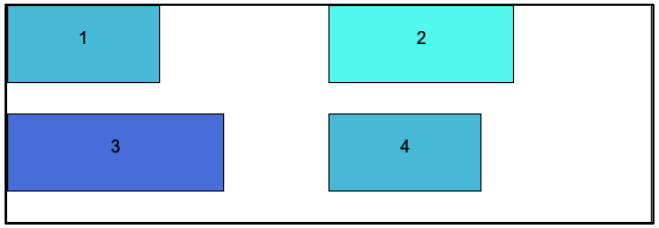
- `justify-items:` alignement horizontal

```
justify-items: start   | end
               | center | stretch;
```




- `align-items:` alignement vertical

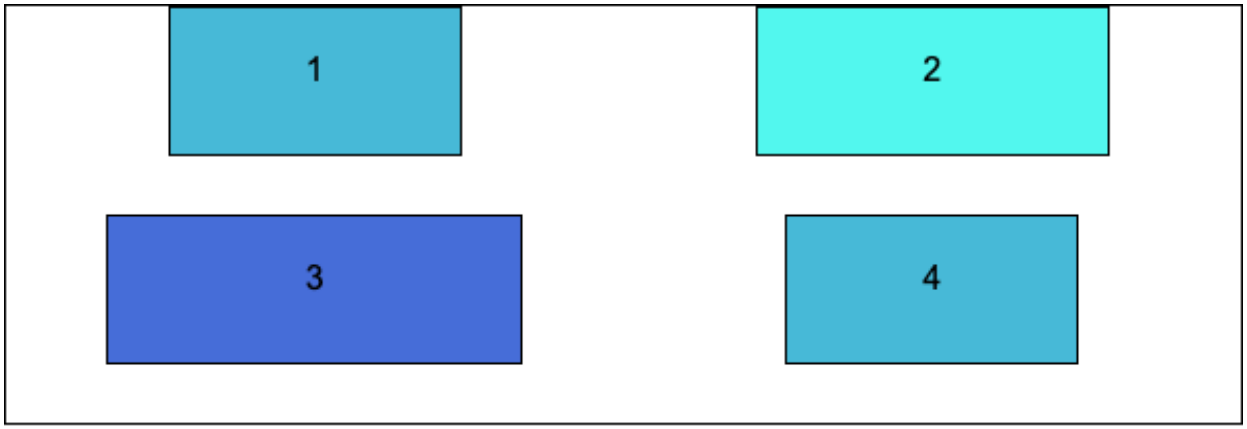
```
align-items: start    | end
             | center | stretch;
```



# Alignement des élnts dans la grille

- `place-items:` alignement vertical et horizontal

```
place-items: align-items justify-items
             | start center;
             | stretch ; // stretch dans
                        // les 2 dimensions
```



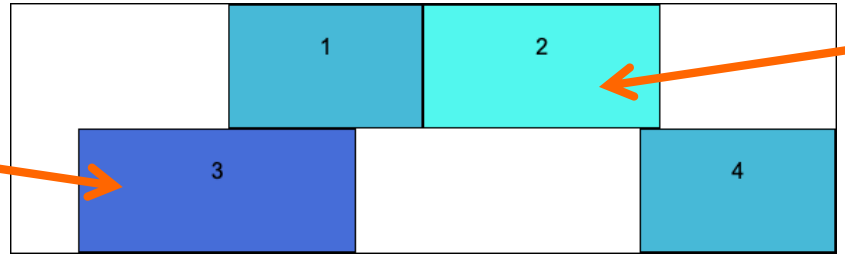
# Alignement d'un élément

- `justify-self:` alignement horizontal

```
justify-self: start |end |center |stretch;
```

justify-self: start;

justify-self: center;




Avec justify-items: end;

- `align-self:` alignement vertical

```
align-self: start |end |center |stretch;
```

# Alignement d'un élément

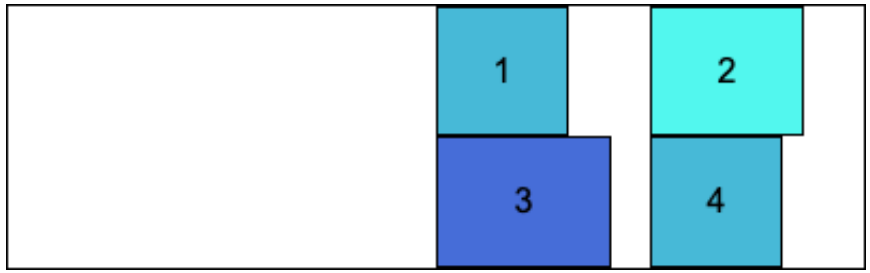
- `place-self:` alignement vertical et horizontal

```
place-self: align-self justify-self
          | start  center;
          | end ;        // end dans les
                         // 2 dimensions
```

# Alignement de la grille

- `justify-content:` alignement horizontal de toute la grille dans son conteneur

```
justify-content: start
                | end
                | center
                | stretch
                | space-around
                | space-between
                | space-evenly;  *
```




*// * même espace au début,*
*// à la fin et entre chaque colonne*

# Alignement de la grille

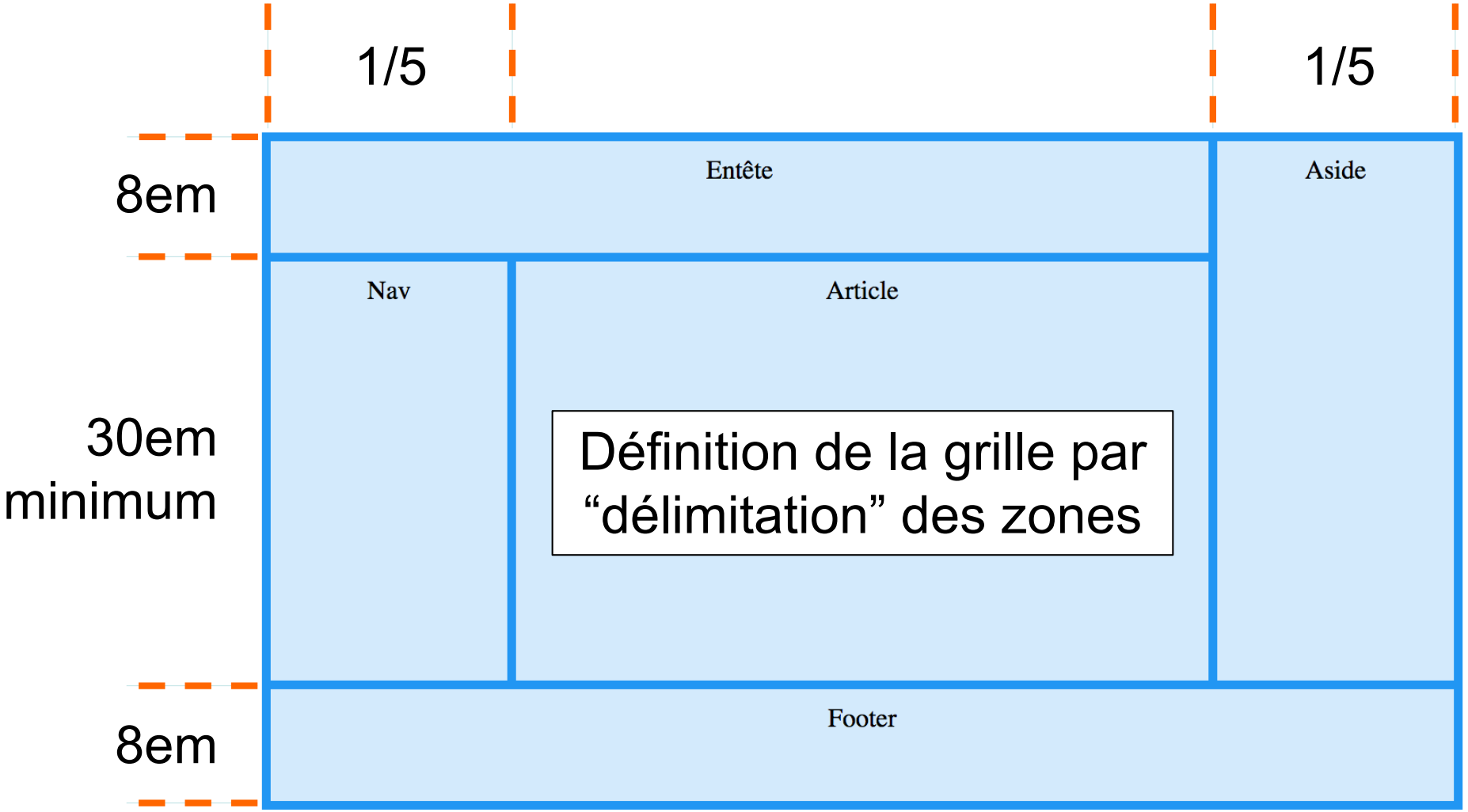
- `align-content:` alignement vertical de la grille dans son conteneur

```
align-content: start
             | end
             | center
             | stretch
             | space-around
             | space-between
             | space-evenly;
```

- `place-content:` alignement vertical et horizontal

```
place-content:align-content/justify-content
            | stretch / center
            | space-around ;
```

# Exemple de grille



# Définition de zones

- `grid-template-areas:` définition dans le conteneur d'une grille à l'aide de noms de zone

```
grid-template-areas: "nom1 nom1 nom1"
                     "nom2  .     ."
                     "nom2  .     .";
```

`nom1` occupe 3 zones, `nom2` occupe 2 zones

`.` indique que la zone n'a pas d'élément attribué

- `grid-area:` attribution d'une zone pour chaque élément

```
grid-area: nom1;
```

# Exemple de grille