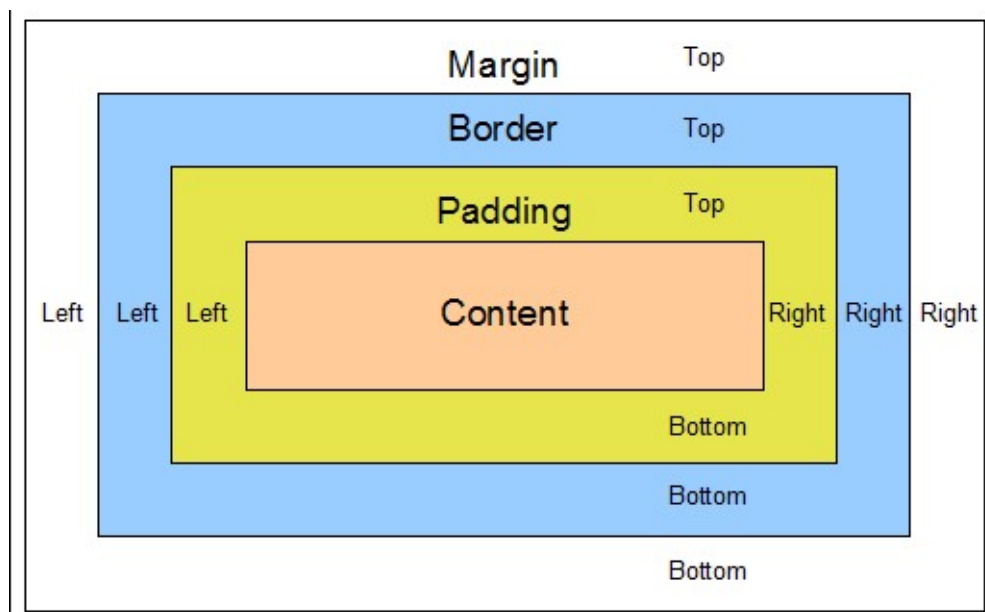


Exemple concerné : Technique d'impression en cellule, basée sur le principe des cellules de tableau. Le document est formé par un agencement de cellules dont les encadrements respectifs forment un tableau. Avec cette technique vous pourrez mettre en page toute sorte de tableau des plus simples aux plus complexe.

### 1-Zone des cellules.

Les classes incluses dans le programme reprennent du point de vue des zones la meme logique que les css sur le web . C'est-à-dire :



- L'encadrement de la cellule est la ligne qui sépare la zone margin de la zone la zone border.
- Cet encadrement n'est pas monolithique : il est défini par 4 tiret ( deux horizontaux et deux verticaux) dont l'épaisseur peut aller de 0 à x. A zéro le tiret composant l'encadrement est invisible.

On peut donc en déduire qu'en qu'agglomérant les cellules les unes aux autres on peut arriver a faire n'importe quel tableau.

Par Exemple pour faire une colonne on colle les cellules les unes aux autres verticalement en laissant les propriétés BorderTop et BorderBottom à 0.

- Evidemment pour la premiere cellule du haut BorderTop devra etre à 1.
- Pour la dernière cellule du Bas on met BorderBottom à 1

Si on ne le faisait pas on aurait obtenu une grille de cellule.

### 2-Types de cellules.

Vous trouverez dans Class\_Cell.cs les classes utilisées pour la technique d'impression en cellule. Bien entendu une fois que vous avez compris cette technique, rien ne vous empêche de les améliorer , voir de créer vos propre classes. C'est une technique assez puissante qui peut vous permettre de créer les états les plus avancés qui soient sans vraiment de limite.

Pour la matérialiser j'ai créé 3 classes simples ( qui héritent toutes d'une classe abstraite CommonCell):

- TextCell : Cellule dédiée à l'insertion de texte basique. Donc aligné à gauche par défaut. Si vous insérez un nombre ( au format string ), et que vous faite une colonne par agglomération de cellule ça ne conviendra donc pas puisque les nombres doivent être alignés à droite ( en principe comme dans une addition à la main ...).
- NumericCell : Cellule dédié à l'insertion de nombre. Donc toujours aligné à droite .
- MoneyCell : Insertion de nombre mais le sigle euro (€) est rajouté à la fin du nombre. Aligné à droite comme NumericCell.

Nous pourrions imaginer des classes plus complexe dont les cellules seraient rattachées à des zone d'impressions prenant toute la largeur de la page mais limités en hauteur (comme sous l'éditeur d'état de Microsoft Access) . Tout est possible avec cette technique il n'y a de limite que votre imagination...

### **3-Description des propriétés.**

Rappel : on met un F a la fin de ces nombres en c sharp lorsque ce sont des float et un M lorsque ce sont des decimaux ( lorsque l'on veut définir la valeur d'une MoneyCell par exemple).

```
.XCoord // Coordonné horizontale supérieur gauche du cadre margin
.YCoord // Coordonné verticale supérieur gauche du cadre margin
.Width // largeur de la cellule
.Height // hauteur de la cellule

.MargingTop // epaisseur de la zone haute de marging de la cellule
.MargingBottom // epaisseur de la zone basse de marging de la cellule
.MargingLeft // epaisseur de la zone gauche de marging de la cellule
.MargingRight // epaisseur de la zone droite de marging de la cellule

.Border_LineTop // (0 invisible/1 visible) etat du tiret haut de la cellule
.Border_LineBottom // (0/1) etat du tiret bas de la cellule
.Border_LineLeft // (0/1) etat du tiret gauche de la cellule
.Border_LineRight // (0/1) etat du tiret droit de la cellule

.PaddingTop // epaisseur de la zone haute de padding de la cellule
.PaddingBottom // epaisseur de la zone basse de padding de la cellule
.PaddingLeft // epaisseur de la zone gauche de padding de la cellule
.PaddingRight // epaisseur de la zone droite de padding de la cellule

// couleur et epaisseur du tiret de la bordure
.Border_LinePen = new Pen(Color.Black, 0.2F);

// type et taille de la police de caractere utilisée dans la cellule.
.CtArea_FontCell = new Font("Courier New", 8);
// couleur de la police de caractere utilisée dans la cellule.
CtArea_Brush = Brushes.Black;

// Pour les textCell uniquement :alignement à droite, centré , gauche du texte.
// Les NumericCell et MoneyCell étant des nombres ils sont toujours aligné a droite.
// Donc CtArea_HorizAlign n'existe pas pour ces deux dernières catégories de cellules.
.CtArea_HorizAlign = 0 // Aligné à gauche
.CtArea_HorizAlign = 1 // Aligné au centre
.CtArea_HorizAlign = 2 // Aligné a droite

// Pour les textCell uniquement :alignement vertical en haut, centré , bas du texte.
// Les NumericCell et MoneyCell étant des nombres ils sont toujours aligné a droite.
// Donc CtArea_HorizAlign n'existe pas pour ces deux dernières catégories de cellules.
.CtArea_VertiAlign = 0 // Aligné en haut
.CtArea_VertiAlign = 1 // Aligné au centre
.CtArea_VertiAlign = 2 // Aligné en bas

//Pour les NumericCell l'alignement a droite se fait par rapport à la virgule
//Il faut donc spécifier le nombre de chiffre maximal après la virgule
//Cela permet d'avoir des chiffres correctement alignés verticalement.
.DecimalNum

// Affecte une valeur à la cellule
.CtArea_Data= 1.123
.CtArea_Data= « Bonjour »

//Design la feuille sur laquelle on va dessiner
.pg = (PrintPageEventArgs)e;

//Ordre de dessiner la cellule
.Draw_Cell();
```

- Par défaut les zones de margin et padding sont à zéro
- Une bonne idée est de dessiner votre tableau sur une feuille de papier puis de mesurer avec une règle.
- Pour vous en sortir vous saisissez : `e.Graphics.PageUnit = GraphicsUnit.Millimeter;`  
Ce qui permettra d'avoir les mêmes unités de mesure (millimètre) que sur votre règle.