Applications Mobiles

420-5GM-BB - Cours 4

Pierre Prades & Mathieu Brodeur-Béliveau

COLLÈGE DE
BOIS-DE-BOULOGNE

Agenda de la séance

- Présences
- Rappel Android Studio
- Propriétés de Text
- Alignement
- Ajout d'image

Propriétés de Text

- Au dernier cours nous avons créer plusieurs Text avec Compose
- Nous avons changé son contenu, sa taille, son alignement, sa hauteur...

```
Text(
    text = "Je suis un texte!",
    fontSize = 100.sp,
    lineHeight = 100.sp,
    textAlign = TextAlign.Center
)
```

Propriétés de Text

- Changer son contenu
  - text = "Je suis un texte!"
- Changer sa taille
  - fontSize = 100.*sp*
- Changer sa hauteur (interligne en Word)
  - lineHeight = 100.*sp*
- Changer sa couleur
  - color = Color.Green

```
Text(
    text = "Je suis un texte!",
    fontSize = 100.sp,
    lineHeight = 100.sp,
    color = Color.Green
)
```

Propriétés de Text - Alignement

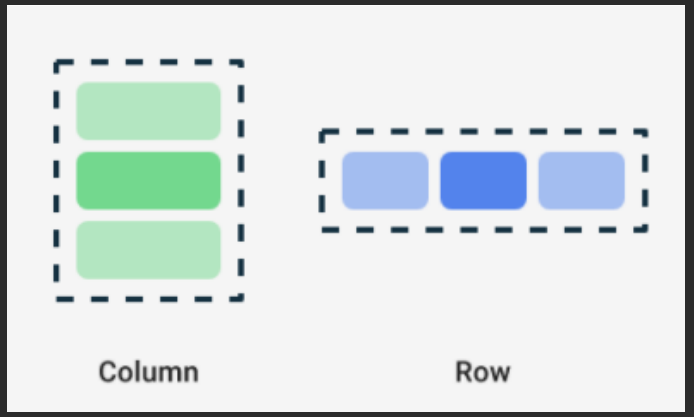
- Aligner au début ou à la fin
  - textAlign = TextAlign.Left / TextAlign.Start
  - textAlign = TextAlign.Right / TextAlign.End
- Justifié
  - textAlign = TextAlign.Justify
- Centrer
  - textAlign = TextAlign.Center

```
Text(
   text = "Je suis un texte!",
   textAlign = TextAlign.End
)
```

- Attention! Seulement par rapport à l'espace occupé du Text

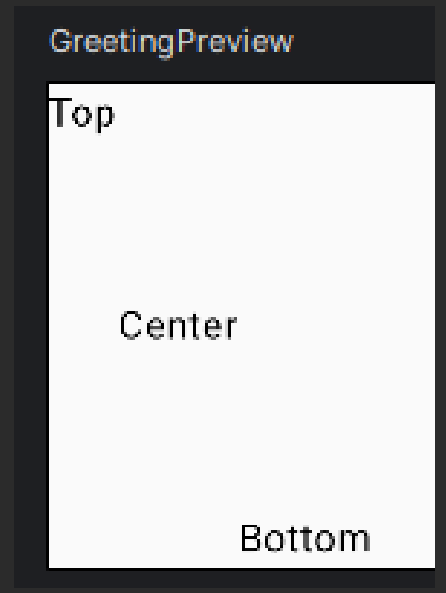
Propriétés de Text - Alignement

- Si on veut aligner un Text dans son parent
  - Dépend du parent en question!
- Row = peut forcer position verticale
- Column = peut forcer position horizontale
- Box = peut forcer où on veut



Alignement - Row

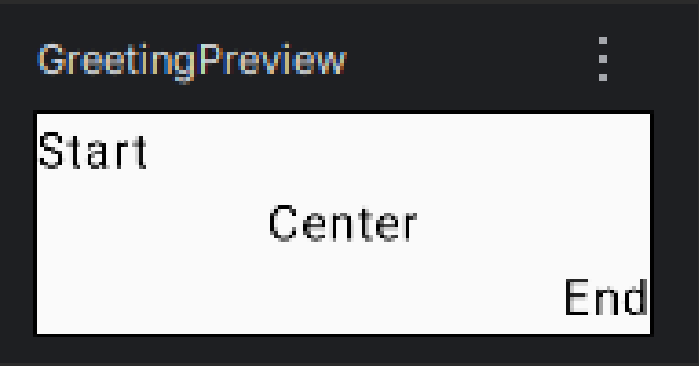
- En haut
  - modifier = Modifier.align(Alignment.Top)
- Au centre
  - modifier = Modifier.align(Alignment.CenterVertically)
- En bas
  - modifier = Modifier.align(Alignment.Bottom)




```
Row(modifier = Modifier.fillMaxHeight()) {
    Text(text = "Top", modifier = Modifier.align(Alignment.Top))
    Text(text = "Center", modifier = Modifier.align(Alignment.CenterVertically))
    Text(text = "Bottom", modifier = Modifier.align(Alignment.Bottom))
}
```

Alignement - Column

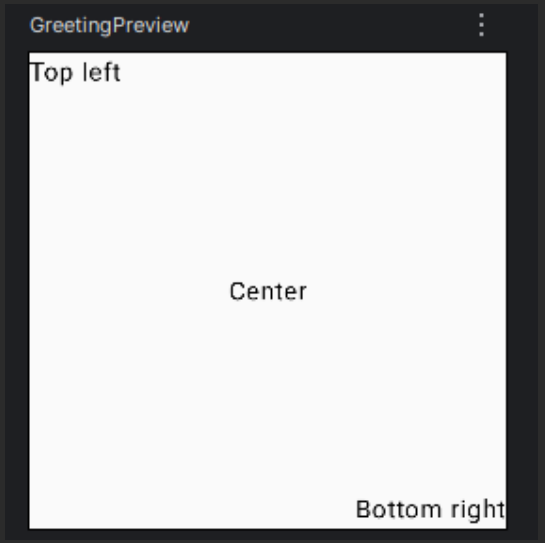
- À gauche
  - modifier = modifier = Modifier.*align*(Alignment.Start)
- Au centre
  - modifier = Modifier.*align*(Alignment.CenterHorizontally)
- À droite
  - modifier = Modifier.*align*(Alignment.End)




```
Column(modifier = Modifier.fillMaxSize()) {
    Text("Start", modifier = Modifier.align(Alignment.Start))
    Text("Center", modifier = Modifier.align(Alignment.CenterHorizontally))
    Text("End", modifier = Modifier.align(Alignment.End))
}
```

Alignement - Box

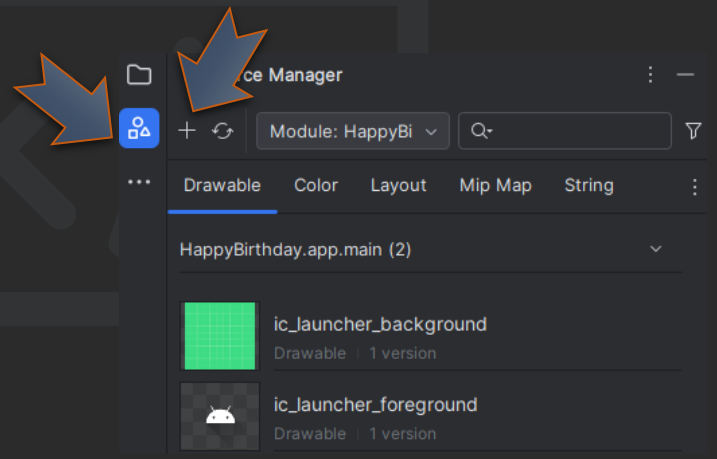
- En haut à gauche
  - modifier = Modifier.*align*(Alignment.TopStart)
- Au centre
  - modifier = Modifier.*align*(Alignment.Center)
- En bas à droite
  - modifier = Modifier.*align*(Alignment.BottomEnd)

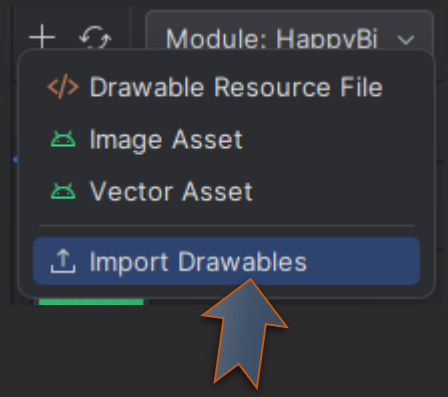


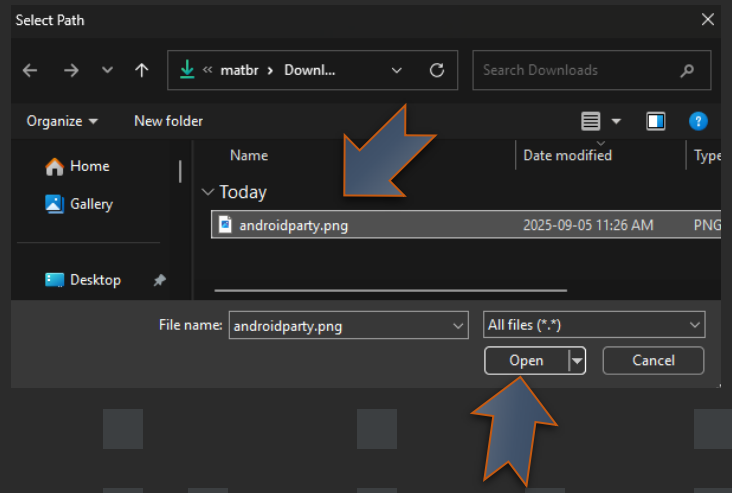

```
Box(modifier = Modifier.fillMaxSize()) {
    Text("Top left", modifier = Modifier.align(Alignment.TopStart))
    Text("Center", modifier = Modifier.align(Alignment.Center))
    Text("Bottom right", modifier = Modifier.align(Alignment.BottomEnd))
}
```

Ajout d'image

- Allez télécharger androidparty.png sur Léa!

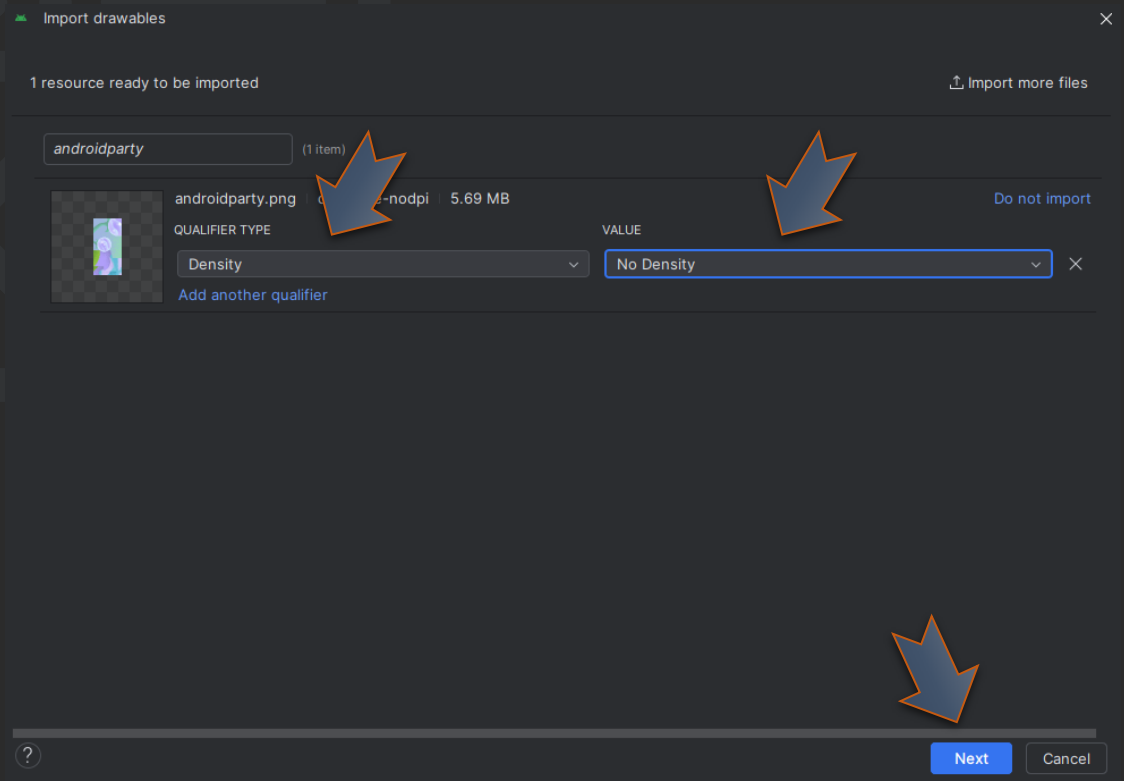


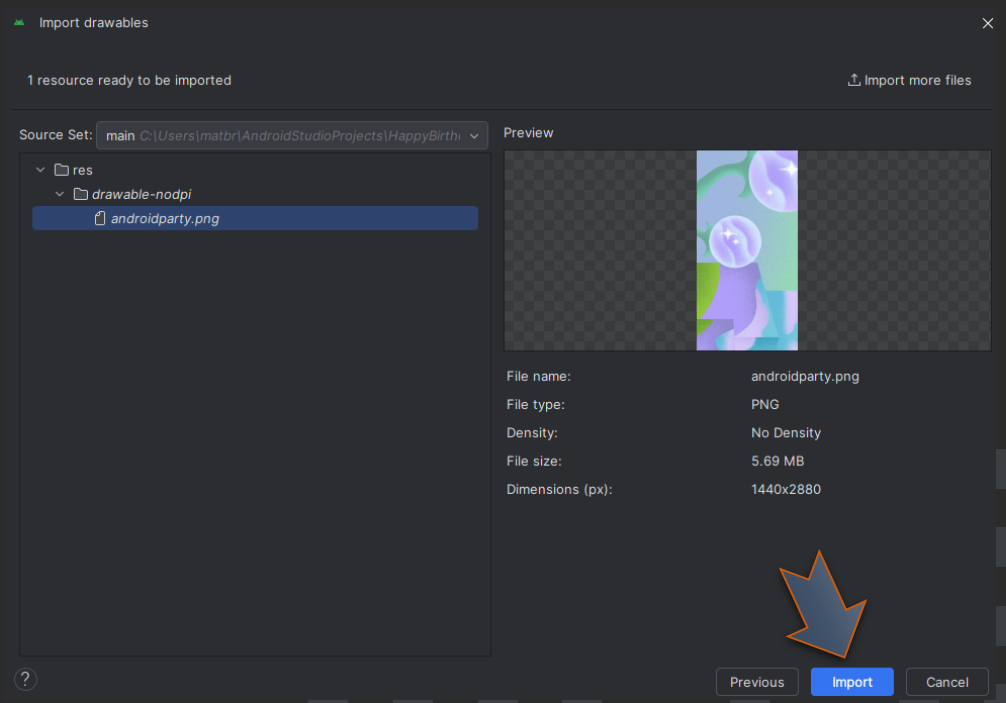




Ajout d'image

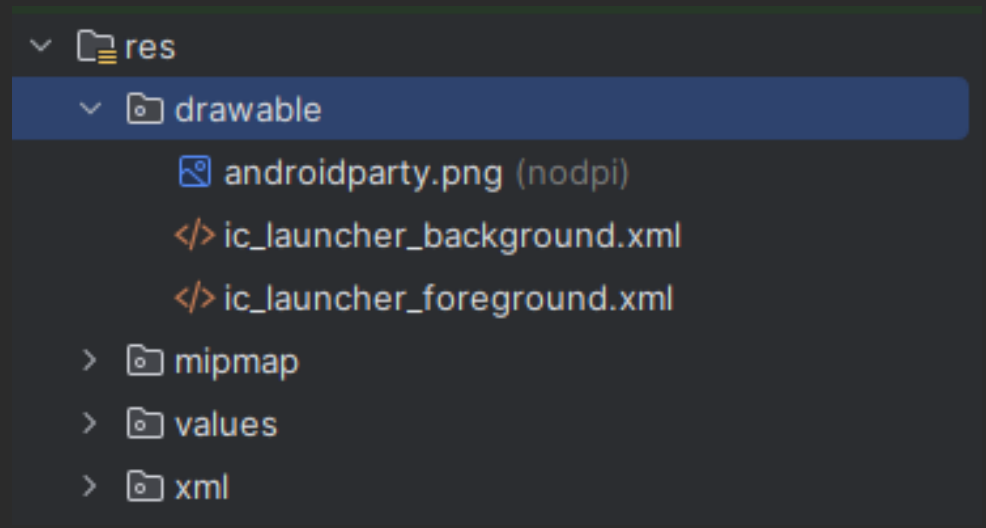
- Qualifier Type = Density Value = No Density





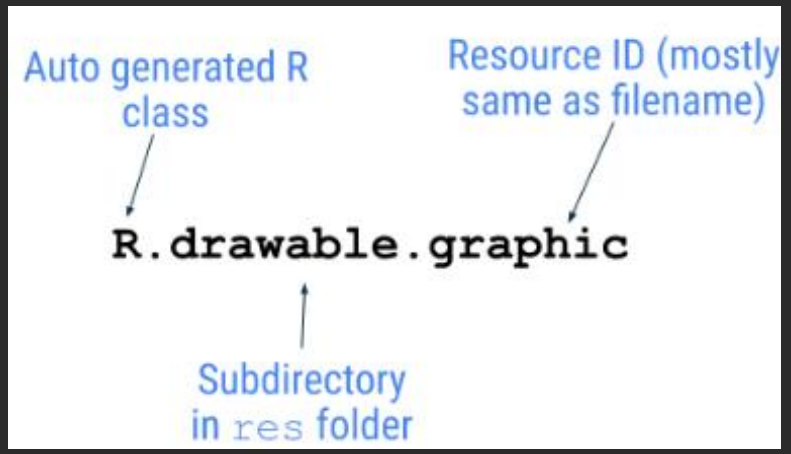
Ajout d'image

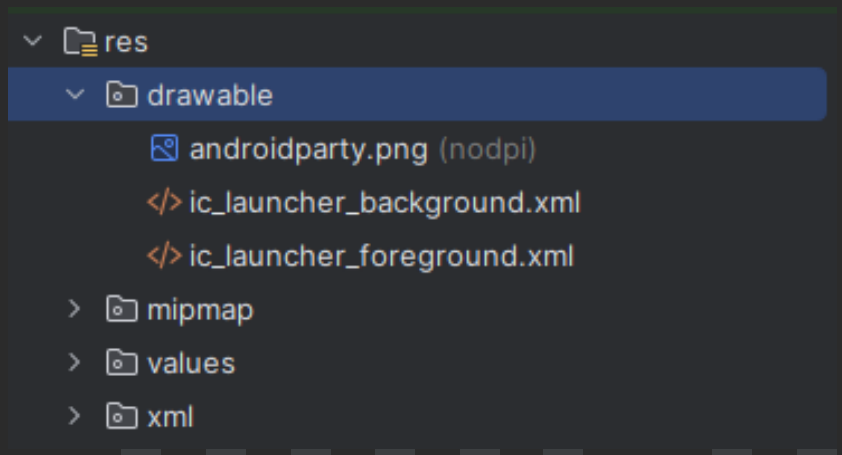
- Notre image est dans le dossier drawable
  - Du fichier de ressource (res)
- Comment le prendre dans notre code?
  - Ou n'importe quelle ressource?
- Classe R généré automatiquement!



Gestion des ressources

- Pour aller cherche une ressource
  - R.nom_dossier.*nom_fichier*
- Alors pour notre image :
  - R.drawable.*androidparty*
- Pour mipmap?
  - R.mipmap.*ic_launcher*
- Pour values?
  - R.string.*app_name*





Ajout d'image

- Pour afficher notre image, nous allons utilisé painterResource
- Il prend en paramètre la ressource (notre image)
- On peut mettre cela dans notre Compose Image
  - Avec la propriété painter
  - Il faut aussi spécifié contentDescription

```
val image = painterResource(R.mipmap.ic_launcher)
Image(
    painter = image,
    contentDescription = ""
)
```

Ajout d'image

- Si nous voulons l'afficher en arrière-plan
  - On met une box!
  - Les 2 vont être superposés

```
Box(modifier = modifier) {
    Image(
        painter = image,
        contentDescription = ""
    )
    Text(text = "Je suis un texte")
}
```

Ajout d'image

- Pour que notre Image couvre notre téléphone au complet
  - contentScale = ContentScale.Crop
- Fit = Scale x et y, maintien ratio
- FillBounds = Stretch to fill
- Crop = Fit + crop pour couvrir
- FillHeight = Scale y, maintien ratio
- FillWidth = Scale x, maintien ratio
- Inside = Fit, mais scale pas l'image

```
Image(
    painter = image,
    contentDescription = "",
    contentScale = ContentScale.Crop
)
```

Propriété générales utiles

- modifier.*fillMaxSize*()
  - Force l'élément a prendre l'écran complet
- modifier.*padding*(8.*dp*)
  - Ajoute un padding de 8 pixels autours de l'élément
  - modifier.*padding*(top = 8.*dp*)
  - modifier.*padding*(bottom = 8.*dp*)
  - modifier.*padding*(start = 8.*dp*)
  - modifier.*padding*(end = 8.*dp*)
- Dans une Image, on peut modifier l'alpha
  - alpha = 0.5F

Exercice

- Faire une application qui montre 4 images comme dans la capture ci-dessous:
  - Une courte description est placée en dessous de chaque chat

- Indice :
  - Allez voir la propriété modifier.*weight*
    - Pour occuper 50% de l'espace disponible
  - Modifiez la police pour 25.sp
  - Allez voir la propriété verticalArrangement
    - Pour centrer les chats et le texte
  - Pensez à factoriser!!

Exercice Supplémentaire

- Faire le projet Android Dev suivant: Projet : créer une application de carte de visite