

App Inventor

trucs

Afficher une image
depuis le Web ou
Google Drive



SUJETS ABORDÉS

- Afficher une image depuis le Web et depuis Google drive
- Télécharger une image (ou fichier)
- Télécharger une liste d'images
- Gérer une liste d'objets depuis le Web
 - Image et autres attributs
 - Voir Galerie
 - ai2.appinventor.mit.edu/?galleryId=5461802291953664
 - Les applications sont disponibles dans la galerie (pierre.huguet50)



AFFICHER UNE IMAGE DEPUIS LE WEB

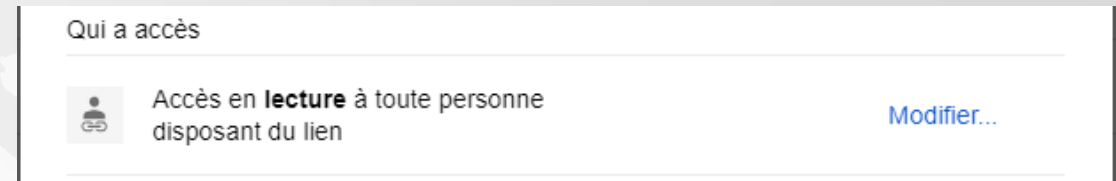
- Pour afficher une image accessible sur le Web par son adresse URL :
 - Copiez l'URL dans la barre d'adresse d'un navigateur et vérifiez qu'elle est affichée
 - Clic droit et `Copier l'adresse de l'image`
 - Vérifiez
 - que l'URL commence par http et se termine par un suffixe .jpeg ou .png
 - que l'image est de taille limitée (ex : 480 px), `Moyennes` `Couleur` `Réutilisation autorisée sans but commercial`
 - que vous avez le droit d'utiliser cette image
 - Collez l'URL dans la propriété ou la variable à afficher, par exemple :
https://cdn.pixabay.com/photo/2013/07/13/13/41/dice-161377_960_720.png
dans un composant image

The screenshot shows an Android Studio interface. On the left, there is a code editor with a snippet: `quand Screen1 .Initialise` and `faire mettre Image1 . Image à "https://cdn.pixabay.com/photo/2013/07/13/13/41/d..."`. A small window titled 'Screen1' shows a preview of the image being loaded. On the right, a larger preview window shows the final result: a screen titled 'Screen1' displaying four colorful dice (red, purple, green, yellow) on a white background. A large white arrow points from the code editor to the final preview window.



AFFICHER UNE IMAGE DEPUIS LE DRIVE

- Pour afficher une image depuis Google drive : drive.google.com
 - Clic droit "obtenir le lien de partage" que vous pouvez coller/tester dans un navigateur https://drive.google.com/open?id=1Kc55YphPVRkqhM4CUsEwVlpuBqeN_G0Q
 - Normalement c'est bon, mais vérifiez l'accessibilité avec clic droit et "partager« puis en bas du menu "avancé" vérifiez que l'accès est autorisé à toute personne disposant du lien
- **Attention** : Ce lien ne renvoie PAS une image, mais un fichier html
- Pour une image, utilisez l'adresse : <https://drive.google.com/uc?export=viewd&id=> suivie de l'identifiant de l'image, par exemple [1hXhXutU5hrmxKXRYXmz_KOAzRcBd0aGG](https://drive.google.com/uc?export=viewd&id=1hXhXutU5hrmxKXRYXmz_KOAzRcBd0aGG)



```
quand Screen1 .Initialise
faire mettre Image1 . Image à " https://drive.google.com/uc?export=viewd&id=166l..."
```





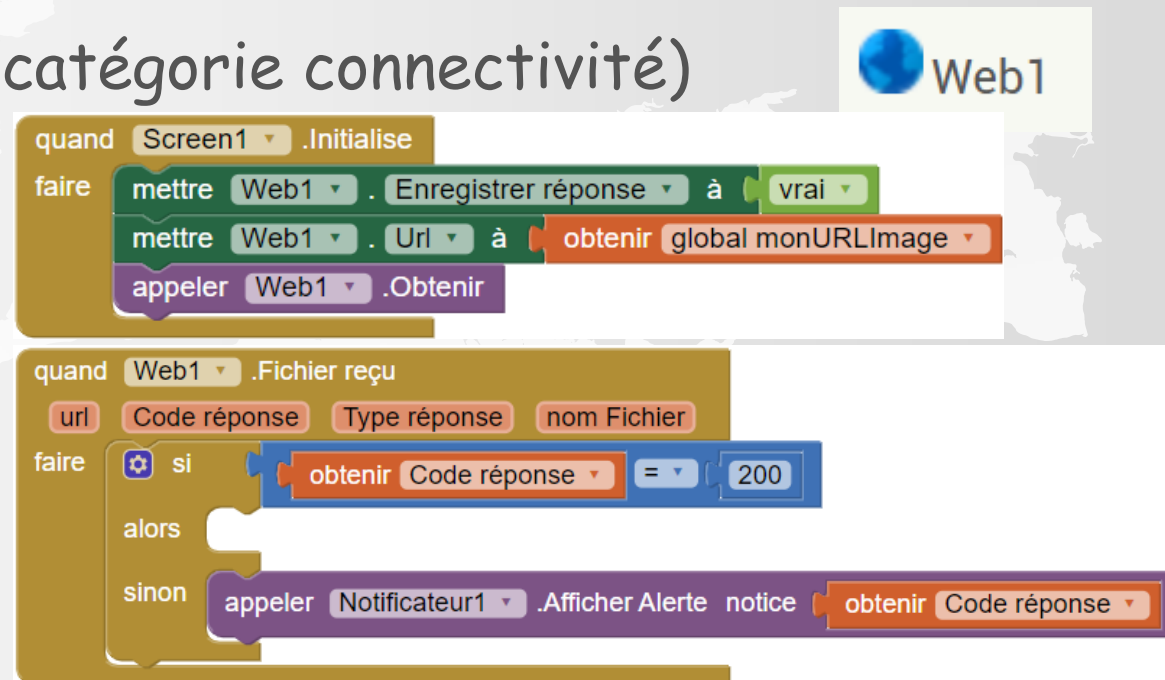
<http://onvaessayer.org>

Pierre Huguet, pierre.huguet50@gmail.com



TÉLÉCHARGER UNE IMAGE (OU FICHIER)

- On peut télécharger les images et les enregistrer localement
Ca évite de les charger depuis le Web à chaque affichage,
mais ça peut encombrer la mémoire (penser à supprimer les fichiers)
- Ajoutez un composant Web (catégorie connectivité)
- Pour télécharger le fichier
(image, son ou autre)
- Le résultat est renvoyé
avec un code (200)
et le nom du fichier
enregistré



The screenshot shows the 'Web1' component in the Android Studio interface. The code blocks are as follows:

```
quand Screen1 .Initialise
faire
  mettre Web1 . Enregistrer réponse à vrai
  mettre Web1 . Url à obtenir global monURLImage
  appeler Web1 .Obtenir

quand Web1 .Fichier reçu
url Code réponse Type réponse nom Fichier
faire
  si
    obtenir Code réponse = 200
  alors
  sinon
    appeler Notificateur1 .Afficher Alerte notice obtenir Code réponse
```



TÉLÉCHARGER UNE IMAGE (exemple)

Dans cet exemple :

- On demande (avec enregistrement de la réponse) le fichier à l'URL `monURLImage`
- Si le code renvoyé est valide (200) le `nom du fichier` enregistré est stocké dans la variable `monFichierImage`
- Qui peut-être affiché
- On peut détruire le fichier à la sortie de l'appli. (avec un composant fichier)

```
initialise global monURLImage à " https://drive.google.com/uc?export=viewd&id=1hXh..."
initialise global monFichierImage à ""

quand Screen1 .Initialise
faire
mettre Web1 . Enregistrer réponse à vrai
mettre Web1 . Url à obtenir global monURLImage
appeler Web1 .Obtenir

quand Web1 .Fichier reçu
url Code réponse Type réponse nom Fichier
faire
si
obtenir Code réponse = 200
alors
mettre global monFichierImage à obtenir nom Fichier
mettre Image1 . Image à obtenir global monFichierImage
sinon
appeler Notificateur1 .Afficher Alerte notice obtenir Code réponse

quand Screen1 .Retour pressé
faire
appeler Fichier1 .Effacer nom Fichier obtenir global monFichierImage
```





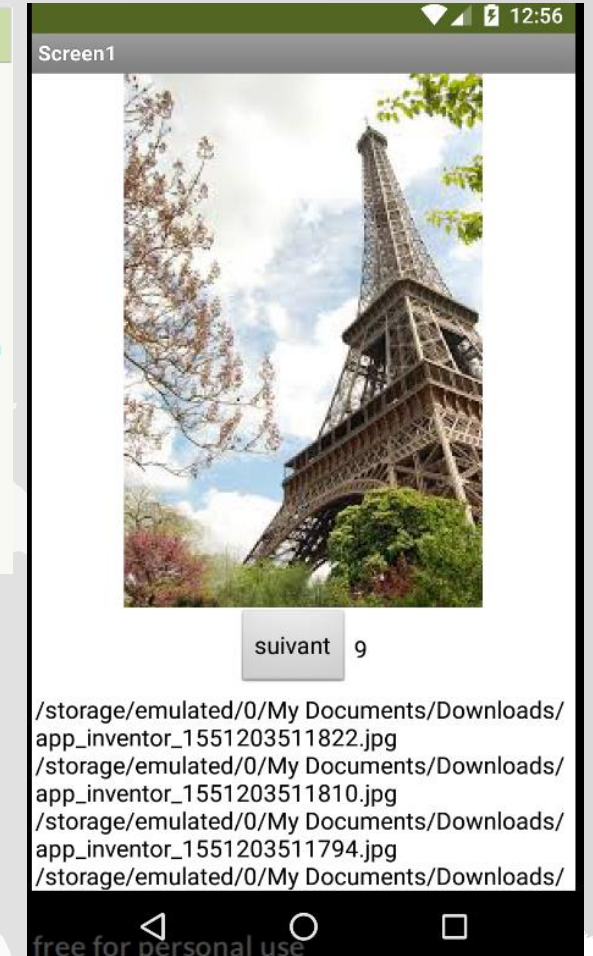
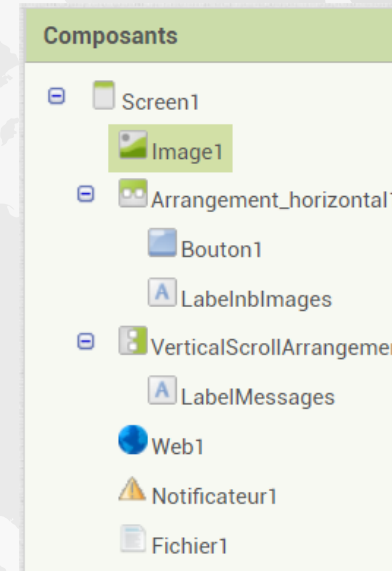
<http://onvaessayer.org>

Pierre Huguet, pierre.huguet50@gmail.com



TÉLÉCHARGER UNE LISTE D'IMAGES

- Pour ce test on a :
 - Un composant image (60% en hauteur)
 - Un arrangement horizontal avec
 - un bouton pour afficher l'image suivante
 - un label pour afficher le nombre d'images chargées, puis l'index de l'image affichée
 - Un arrangement vertical scrollable avec un label pour afficher les messages (taille auto)
- Un composant Web (pour les requêtes)
- Un composant fichier (pour effacer les images à la fin)
- Un notificateur (c'est toujours utile)



TÉLÉCHARGER UNE LISTE D'IMAGES

- On définit
 - une liste d'URLs
 - une liste pour les noms de fichiers de ces images (au départ une liste vide)
- Au démarrage
 - On complète la liste des noms de fichier pour qu'elle fasse la même longueur
 - et pour chaque URL, on fait une requête Web avec enregistrement du résultat

```
initialise global listImageURLs à créer une liste
" https://t3.ftcdn.net/jpg/00/36/43/64/240_F_36436..."
" https://t3.ftcdn.net/jpg/00/07/49/26/240_F_74926..."
" https://drive.google.com/uc?export=view&id=1hXhX..."
" https://drive.google.com/uc?export=view&id=1Kc55..."
" https://drive.google.com/uc?export=view&id=1XI7u..."
" https://drive.google.com/uc?export=view&id=1n3UR..."
" https://drive.google.com/uc?export=view&id=1LDUR..."
" https://drive.google.com/uc?export=view&id=17XLZ..."

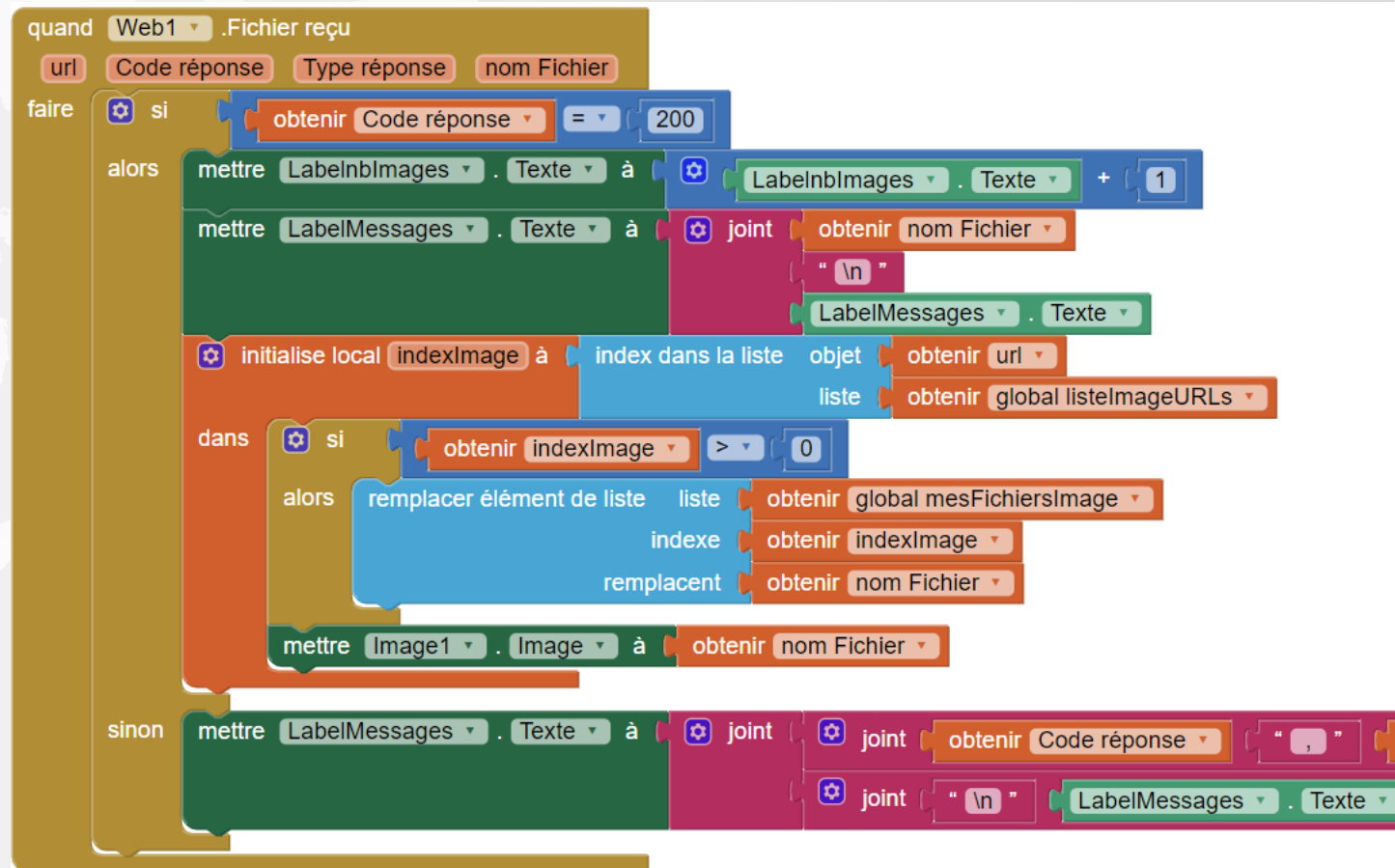
initialise global mesFichiersImage à créer une liste vide

quand Screen1 .Initialise
faire
  pour chaque imageURL dans la liste obtenir global listImageURLs
  faire
    ajouter éléments à la liste liste obtenir global mesFichiersImage
    item " "
  mettre Web1 . Enregistrer réponse à vrai
  pour chaque imageURL dans la liste obtenir global listImageURLs
  faire
    mettre Web1 . Url à obtenir imageURL
    appeler Web1 . Obtenir
```



TÉLÉCHARGER UNE LISTE D'IMAGES

- A la réception d'un fichier
 - Si le code réponse = 200
 - mettre à jour label nb Images
 - ajouter le nom du fichier au début du label de messages
 - rechercher l'index de l'URL de ce fichier
 - mettre à jour nom de fichier dans la liste des fichiers pour ce même index
 - afficher l'image (pour le fun)
 - Si le code réponse \neq 200
 - ajouter le code et l'URL aux messages



TÉLÉCHARGER UNE LISTE D'IMAGES

- Supprimer les fichiers à la sortie de l'application
- et ajouter un bouton pour vérifier que les images ont bien été chargées

```
quand Screen1 .Retour pressé
faire
  pour chaque fichierImage dans la liste
  faire
    appeler Fichier1 .Effacer nom Fichier
```

```
initialise global indexAff à 0

quand Bouton1 .Clic
faire
  mettre global indexAff à obtenir global indexAff + 1
  si
    obtenir global indexAff > taille de la liste liste
  alors
    mettre global indexAff à 1
  mettre Image1 . Image à choisir liste élément liste
  mettre LabelnImages . Texte à obtenir global indexAff
```





<http://onvaessayer.org>

Pierre Huguet, pierre.huguet50@gmail.com



GÉRER UNE LISTE DEPUIS LE DRIVE

Pour gérer et mettre à jour une liste d'objets sur le drive et l'utiliser depuis votre application

- commencez par définir les attributs des objets à gérer
 - Par exemple pour des produits : l'URL de la photo, le nom, le prix, la taille

| | | |
|--------------------|--------------|---|
| baskets | 56 taille 42 | https://t3.ftcdn.net/jpg/00/07/49/26/240_F_7492635_M6ofTd1OtrcGgnawtNNUMUTdM11GODob.jpg |
| chaussures de foot | 75 taille 39 | https://t3.ftcdn.net/jpg/00/36/43/64/240_F_36436460_WQbusrj5PUj1GhFEShsTY1JepN26MVUj.jpg |
| Oxwork | 78 taille 43 | https://drive.google.com/uc?export=view&id=10nxUDEtZcab90r8O0vQnnRfKDDOqOOXA |
| Nike | 87 taille 44 | https://drive.google.com/uc?export=view&id=1f5tP-32iSzjOcUQ0rQR49cUWS7ZmNN4x |
| adidas | 74 taille 39 | https://drive.google.com/uc?export=view&id=1a1D1XH-yfnNrxVpU5nUGQnYYgmsTP8xX |

- Vous pouvez préparer le travail avec Excel



GÉRER UNE LISTE DEPUIS LE DRIVE

Pour gérer et mettre à jour une liste d'objets sur le drive et l'utiliser depuis votre application

- commencez par définir les attributs des objets à gérer
- ensuite créez un fichier avec un objet par ligne et en séparant dans chaque ligne les champs par une virgule il ne doit y avoir **AUCUNE autre virgule** même dans les nombres

baskets,56, taille 42,https://t3.ftcdn.net/jpg/00/07/49/26/240_F_7492635_M6ofTd10trcGgnawtNNUMUTdM11GODob.jpg
chaussures de foot,75, taille 39,https://t3.ftcdn.net/jpg/00/36/43/64/240_F_36436460_WQbusrj5PUj1GhFEShsTY1JepN26MVUj.jpg
Oxwork,78, taille 43,https://drive.google.com/uc?export=view&id=10nxUDEtZcab90r8O0vQnnRfKDDOqOOXA
Nike,87,taille44,https://drive.google.com/uc?export=view&id=1f5tP-32iSzjOcUQ0rQR49cUWS7ZmNN4x
adidas,74,taille 39,https://drive.google.com/uc?export=view&id=1a1D1XH-yfnNrrjVpU5nUGQnYYgmsTP8xX



GÉRER UNE LISTE DEPUIS LE DRIVE

- Commencez avec deux objets simples

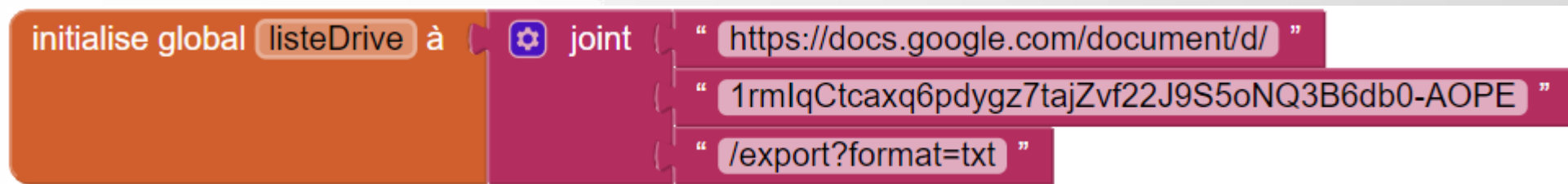
Nike,87,taille44, <https://drive.google.com/uc?export=view&id=1f5tP-32iSzjOcUQ0rQR49cUWS7ZmNN4x>
adidas,74,taille 39, <https://drive.google.com/uc?export=view&id=1a1D1XH-yfnNrjVpU5nUGQnYYgmsTP8xX>

- Si vous connaissez Excel ou un autre tableur vous pouvez
 - l'utiliser pour préparer
 - enregistrer au format csv
 - remplacer les ; par des ,



GÉRER UNE LISTE DEPUIS LE DRIVE

- Enregistrez ce fichier dans le drive et récupérez le lien partageable
- Dans App Inventor construisez l'adresse de récupération de cette liste, avec au milieu, l'id récupéré dans le lien partageable



- https://docs.google.com/document/d/FILE_ID/export?format=txt
où **FILE_ID** correspond à l'identifiant du fichier texte qui contient la liste
cf. doc à l'adresse <https://www.labnol.org/internet/direct-links-for-google-drive/28356/>
- (Dans App Inventor, il faut un composant Web et un labelMessages)



GÉRER UNE LISTE DEPUIS LE DRIVE


- Ensuite, au démarrage
 - Assignez l'URL au composant Web
 - ne PAS enregistrer la réponse dans un fichier
 - Et appeler Web.obtenir (get)

```
initialise global listeDrive à joint [ " https://docs.google.com/document/d/ "
                                     " 1rmlqCtcaxq6pdygz7tajZvf22J9S5oNQ3B6db0-AOPE "
                                     " /export?format=txt " ]

quand Screen1 .Initialise
faire
  mettre Web1 . Enregistrer réponse à faux
  mettre Web1 . Url à obtenir global listeDrive
  appeler Web1 .Obtenir
```



GÉRER UNE LISTE DEPUIS LE DRIVE

- On enregistre pas la réponse, elle revient avec **Web.texte reçu**
- Le texte reçu est une table de valeurs que l'on convertit en liste de listes avec 

que l'on range dans la liste des objets

```
quand Web1 . Texte reçu
  url Code réponse Type réponse Contenu de réponse
  faire
    si
      obtenir Code réponse = 200
      alors
        mettre global listeobjets à liste de table csv texte obtenir Contenu de réponse
      sinon
```



GÉRER UNE LISTE DEPUIS LE DRIVE

- Pour vérifier, on affiche le résultat dans la zone de message

On a une liste à 2 niveaux

Avec 2 objets dont chacun a 4 attributs

- la marque,
- le prix,
- la taille
- l'URL image

```
Screen1
((Nike 87 taille44 https://drive.google.com/open?
id=1f5tP-32iSzjOcUQ0rQR49cUWS7ZmNN4x) (a-
didas 74 taille 39 https://drive.google.com/open?
id=1a1D1XH-yfnNrxVpU5nUGQnYYgmsTP8xX))
liste de longueur : 2

4attributs :
Nike
87
taille44
https://drive.google.com/open?
id=1f5tP-32iSzjOcUQ0rQR49cUWS7ZmNN4x

4attributs :
adidas
74
taille 39
https://drive.google.com/open?id=1a1D1XH-
yfnNrxVpU5nUGQnYYgmsTP8xX
```

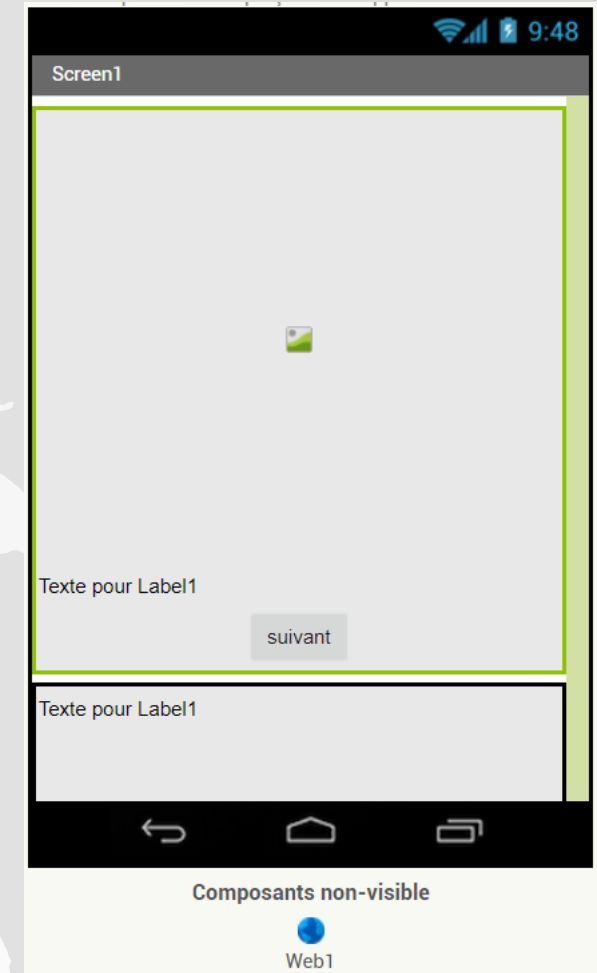
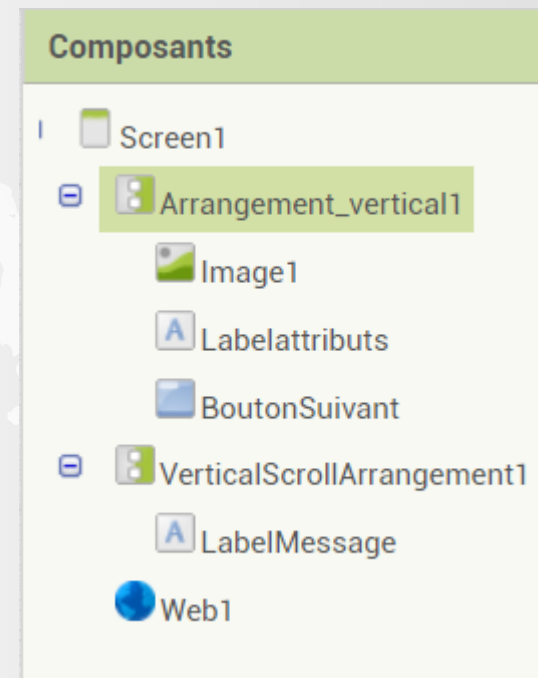
```
quand Web1 . Texte reçu
  uri Code réponse Type réponse Contenu de réponse
faire
  si obtenir Code réponse = 200
  alors
    mettre global listeobjets à liste de table csv texte obtenir Contenu de réponse
    mettre LabelMessages . Texte à obtenir global listeobjets
    appeler DecoderEtAfficherListe
  sinon
    mettre LabelMessages . Texte à joint joint obtenir Code réponse " " obtenir url
```

```
à DecoderEtAfficherListe
faire
  mettre LabelMessages . Texte à joint LabelMessages . Texte
  " \nliste de longueur : "
  taille de la liste liste obtenir global listeobjets
  " \n "
  pour chaque image dans la liste obtenir global listeobjets
  faire
    mettre LabelMessages . Texte à joint joint LabelMessages . Texte " \n "
    taille de la liste liste obtenir image
    " attributs : "
    pour chaque attribut dans la liste obtenir image
    faire
      mettre LabelMessages . Texte à joint joint LabelMessages . Texte " \n "
      obtenir attribut
  mettre LabelMessages . Texte à joint joint LabelMessages . Texte " \n "
```



GÉRER UNE LISTE DEPUIS LE DRIVE

- On peut maintenant afficher les produits
 - Dans le design, on ajoute
 - un arrangement vertical avec
 - une image
 - un label pour afficher les attributs
 - un bouton pour afficher l'objet suivant
- (on peut masquer le label messages ou le mettre dans un arrangement vertical de type scroll)



GÉRER UNE LISTE DEPUIS LE DRIVE

- Ensuite, à chaque fois que l'on appuie sur le bouton, on affiche le produit suivant
- Le script principal gère l'index et appelle une procédure avec en paramètre le produit à cet index
- La procédure d'affichage récupère les 4 attributs et les affiche

```
initialise global indexAff à 0
quand BoutonSuivant .Clic
faire
  mettre global indexAff à obtenir global indexAff + 1
  si obtenir global indexAff > taille de la liste liste obtenir global listeobjets
  alors mettre global indexAff à 1
  appeler afficherImageEtAttributs
    objet choisir liste élément liste obtenir global listeobjets
    index obtenir global indexAff
à afficherImageEtAttributs objet
faire
  initialise local marque à choisir liste élément liste obtenir objet index 1
  initialise local prix à choisir liste élément liste obtenir objet index 2
  initialise local taille à choisir liste élément liste obtenir objet index 3
  initialise local image à choisir liste élément liste obtenir objet index 4
  dans
    mettre Labelattributs . Texte à joint obtenir marque
    joint " " obtenir taille
    joint " prix : " obtenir prix
  mettre Image1 . Image à obtenir image
  mettre LabelMessage . Texte à obtenir image
```



GÉRER UNE LISTE DEPUIS LE DRIVE

