Comprendre le main.dart

1. La fonction main(), ça sert à quoi ?

- Imaginez-vous avec une pile de documents, comprenant chacun une série d'informations cryptées liées les unes aux autres avec, quelque part, une seule et unique information vous permettant de déchiffrer tous les autres documents. Autant chercher une aiguille dans une botte de foin !
- Nous pourrions alors nous dire que la personne qui a rédigé et empilé cette série de documents aurait pu prendre le soin d'indiquer où était l'information clé : avec un Post-it, une couleur de feuille différente, en la mettant en haut de la pile, etc.
- En programmation, il y a toujours un point de départ dans l'exécution d'un programme. Tout comme dans une pile de documents, il y a plusieurs façons d'indiquer où est l'information de départ.
- En Pascal par exemple, le fichier comprenant les toutes premières instructions commence par **begin** et termine par **end**. D'autres langages comme l'Ada « marquent » la fonction de départ (un peu comme un coup de Stabilo !).
- En C, C++ et C# cependant, le point de départ sera toujours la fameuse fonction main(). Et un langage qui s'inspire grandement du C... le Dart !

> - *La fonction **main()** sera donc le point de départ de l'exécution de votre programme en Dart. Depuis cette fonction, vous pouvez ensuite appeler d'autres fonctions, qui à leur tour appelleront d'autres fonctions... et ainsi de suite.*
> *La fonction **main()** est précédée par le type de retour **void**, qui indique que la fonction ne renvoie rien. La parenthèse **()** indique qu'il s'agit d'une fonction et le code DART est indiqué dans les crochets "**{....}**".*

2. Spécificités de la fonction main()

- La fonction `main()` présente plusieurs spécificités comparée à des fonctions traditionnelles :
  - Elle est la première fonction à être exécutée dans un programme en Dart.
  - Il faut obligatoirement une (et une seule !) fonction avec le nom main. Cela n'est pas une convention, c'est tout simplement essentiel au fonctionnement de votre programme.
  - La fonction `main()` est souvent présente dans un fichier portant le nom de l'application (exemple : nomdemonapp.dart) ou tout simplement dans un fichier

apcpedagogie.com

main.dart (comme dans une application Flutter par défaut). Il s'agit là bien de conventions - libre à vous d'appeler ce fichier comme vous le souhaitez.
- La fonction `main()` ne retourne rien et est donc de type void. Pour rappel, le type void permet de retourner un résultat de n'importe quel type, ou pas de résultat du tout.
- La fonction `main()` ne prend (dans la grande majorité des cas) aucun argument.

3. **Exécution de la fonction main() et ses arguments**

   - Maintenant que vous savez l'essentiel concernant cette irremplaçable fonction main, encore faut-il réussir à l'exécuter !
   - Pour ça, rien de plus simple. Il y a quatre façons d'exécuter du code en Dart (compilé en JavaScript, autonome, compilation anticipée ou native), mais celle qui nous intéresse aujourd'hui est l'exécution dite « autonome ».
   - L'exécution autonome permet d'exécuter du code Dart dans une interface de ligne de commande. Le terminal, en d'autres termes !
   - Pour exécuter un programme en Dart, il nous suffit alors d'écrire en ligne de commande : **`dart run monfichier-contenant-le-main.dart`**
   - Et si nous rajoutions des arguments au moment de lancer notre programme ? Mais nous venons de dire que la fonction main() ne prenait pas d'argument !
   - C'est vrai dans la (très) grande majorité des cas, mais en réalité, c'est tout à fait possible et cela peut parfois s'avérer utile.
   - Nouvelle spécificité de la fonction main : les arguments d'une fonction main() peuvent être récupérés en List de String (List). Ils ne peuvent pas être d'un autre type. Même si vous ne renseignez qu'un argument, celui-ci sera placé dans une liste.
   - La fonction `main()` peut alors s'écrire comme suit :

```
void main(List args) {
  print(args);
})
```

4. La fonction `main()` étant la première fonction appelée, c'est la seule fonction pour laquelle vous pouvez renseigner les arguments au moment où vous lancez le programme avec « dart run ».
5. Vous pouvez alors renseigner vos arguments comme suit :

```
dart run monfichiercontenantlemain.dart argument1 argument2 argument3
```

apcpedagogie.com

6. Peut-être vous demandez-vous alors à quoi cela peut bien servir. Parfois, on peut avoir besoin de lancer un programme qui saura prendre en compte un paramètre extérieur au moment de son exécution. Et si, par exemple, un programme pouvait s'exécuter différemment, selon l'heure de la journée à laquelle il est lancé ?

> *Après avoir appris l'utilité de la fonction main, encore faut-il savoir l'appeler. La fonction main ne s'appelle pas comme toutes les autres fonctions, car celle-ci s'exécute dès le lancement de votre programme. Pour l'appeler, il faut donc savoir comment exécuter votre programme ! Pour ça, rien de plus simple. Il suffit de taper « dart run monfichier.dart » et la fonction main contenue dans votre fichier s'exécutera.*

Cours Flutter

Source : https://apcpedagogie.com/comprendre-le-main-dart · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com