

Comprendre le main.dart

Table des matières

I. La fonction main(), ça sert à quoi ?	3
II. Exercice : Quiz	3
III. Spécificités de la fonction main()	4
IV. Exercice : Quiz	5
V. Exécution de la fonction main() et ses arguments	6
VI. Exercice : Quiz	6
VII. Essentiel	7
VIII. Auto-évaluation	8
A. Exercice	8
B. Test	8
Solutions des exercices	9

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

Contexte

Vous souhaitez apprendre à coder, mais ne savez pas par où commencer ? Vous êtes noyé par le nombre d'informations disponibles, si bien que vous préférez laisser tomber et remettre l'apprentissage à plus tard ?

C'est à peu près la réaction que votre programme Dart aura si vous ne lui donnez pas un point de départ clair pour l'exécution de ses tâches. Ne sachant pas par où commencer, il ne se lancera tout simplement pas.

Heureusement, l'indispensable fonction `main()` est là pour vous sortir de l'impasse ! Dans ce cours, nous verrons dans un premier temps l'origine de la fonction `main`, avant d'en étudier les spécificités puis les arguments.

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 !).

```
1 program HelloWorld(output);
2 begin
3   writeln('Le Dart c'est super');
4   readln;
5 end.
```

Plus communément, certains langages comme le Python vont tout simplement exécuter les instructions dans leur ordre d'apparition. Un programme en Python aurait tout simplement commencé par lire la première ligne du document qu'on lui aurait donné.

```
1 print(« Le Dart c'est super »)
```

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 !

```
1 void main() {
2   print(« Le Dart c'est super »);
3 }
```

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.

Exercice : Quiz

[solution n°1 p.11]

Question 1

En Dart, par où va commencer l'exécution du code d'un programme ?

- ☐ Par la première ligne
- ☐ Là où il y a l'instruction `begin`
- ☐ À la fonction `main()`

Question 2

Si je veux écrire un court programme affichant « *Hello World* » en Dart, quelle est la syntaxe ?

- ☐ Void main() {
 print(« Le Dart c'est super »);
}
- ☐ print(« Le Dart c'est super »)
- ☐ program HelloWorld(output);
begin
 writeln('Le Dart c'est super');
 readln;
end.

Question 3

Le Dart s'inspire du :

- ☐ Python
- ☐ Pascal
- ☐ C
- ☐ Ada

Question 4

En Python, par où commence l'exécution du code ?

- ☐ Par la fonction main()
- ☐ Par la première ligne
- ☐ Par l'instruction begin

Question 5

La syntaxe ci-dessous est correcte.

```
1 void main() {
2   maFonction();
3 }
4 void maFonction() {
5   print("Hello world");
6 }
```

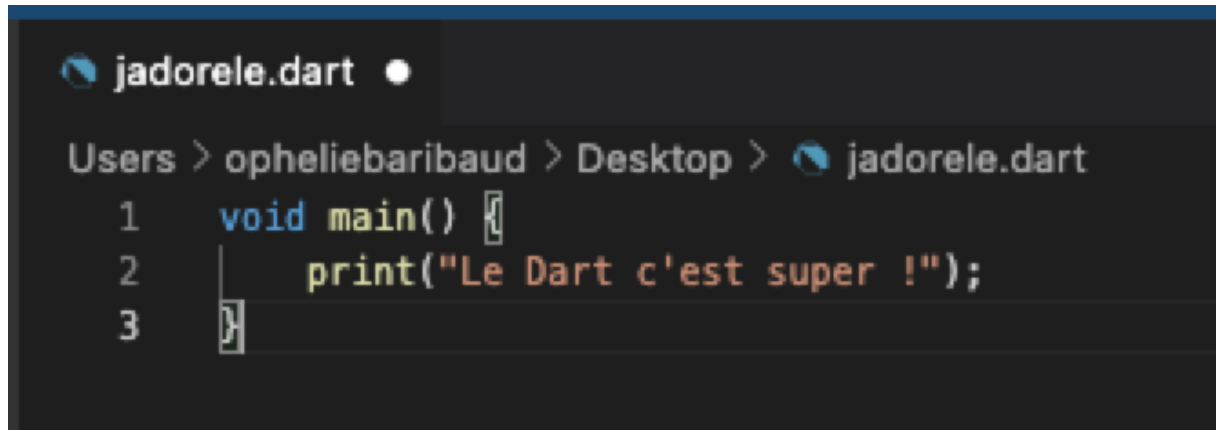
- ☐ Vrai
- ☐ Faux

III. Spécificités de la fonction main()

La fonction `main()` présente plusieurs spécificités comparée à des fonctions traditionnelles :

1. Elle est la première fonction à être exécutée dans un programme en Dart.
2. 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.

3. La fonction `main()` est souvent présente dans un fichier portant le nom de l'application (exemple : *nomdemonapp.dart*) ou tout simplement dans un fichier *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.
4. 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.
5. La fonction `main()` ne prend (dans la grande majorité des cas) aucun argument.



```
jadorele.dart •
Users > opheliebaribaud > Desktop > jadorele.dart
1 void main() {
2     print("Le Dart c'est super !");
3 }
```

Exercice : Quiz

[solution n°2 p.12]

Question 1

Je peux nommer ma fonction de départ comme je le souhaite.

- ☐ Vrai
- ☐ Faux

Question 2

Je peux nommer le fichier ayant ma fonction de départ comme je le souhaite.

- ☐ Vrai
- ☐ Faux

Question 3

Une fonction retourne toujours quelque chose.

- ☐ Vrai
- ☐ Faux

Question 4

Il est possible de mettre des arguments dans une fonction `main()`.

- ☐ Vrai
- ☐ Faux

Question 5

De quel type est la fonction `main` ?

- ☐ `int`
- ☐ `void`
- ☐ `string`
- ☐ `double`

V. 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 monfichiercontenantlemain.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<String>`). 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 :

```
1 void main(List<String> args) {
2   print(args);
3 }
```

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` ».

Vous pouvez alors renseigner vos arguments comme suit :

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

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 ?

Complément

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.

Exercice : Quiz

[solution n°3 p.13]

Question 1

La commande `dart run` est une exécution :

- ☐ Autonome
- ☐ Native
- ☐ Anticipée
- ☐ En JavaScript

Question 2

Quelle(s) syntaxe(s) est/sont juste(s) pour l'exécution d'un programme en Dart ?

- ☐ `run dart monfichier.dart`
- ☐ `run dart ./monfichier.dart`
- ☐ `dart run monfichier.dart`
- ☐ `dart run ./Desktop/monfichier.dart`

Question 3

Les arguments d'une fonction `main()` peuvent être de plusieurs types.

- ☐ Vrai
- ☐ Faux

Question 4

À quel moment dois-je indiquer les arguments de la fonction `main` ?

- ☐ Au moment du lancement du programme
- ☐ Grâce à son appel depuis une autre fonction
- ☐ Juste après le lancement du programme

Question 5

Le nombre d'arguments est limité.

- ☐ Vrai
- ☐ Faux

VII. Essentiel

Le main c'est quoi ?

- La fonction `main()` est le point de départ d'un programme en Dart.
- Certains langages ont d'autres moyens de savoir par où ils doivent commencer l'exécution d'un programme. En Python, c'est la première ligne qui s'exécute, en Pascal c'est indiqué par `begin` et `end`, et en C c'est également avec une fonction `main()`.

Les spécificités de la fonction `main()`

- Il faut obligatoirement une fonction `main` (et une seule).
- Le nom du fichier dans lequel se situe la fonction `main` n'est pas important, mais est souvent le nom de l'application ou `main.dart`.
- La fonction ne retourne rien (elle est de type `void`).

L'exécution de la fonction `main()` et ses arguments

- Exécuter un programme en Dart nécessite l'utilisation de la commande `run` (`dart run monfichier.dart`).
- Il est possible de passer des arguments à la fonction `main` au moment où le programme est lancé.
- Les arguments sont récupérés sous forme d'une `List` de `String`.
- Au moment de l'appel du programme, il suffit de placer les arguments à la fin de la commande (`dart run monfichier.dart argument1 argument2`).

VIII. Auto-évaluation

A. Exercice

Est-ce que vous aussi, vous vous demandez à peu près toutes les 10 minutes l'heure qu'il est ? Vous regardez inlassablement votre montre ou votre smartphone, etc. et si nous trouvions une manière plus originale de nous indiquer l'heure ?

Nous allons faire de vous une personne 2.0, grâce à un programme qui vous indiquera l'heure qu'il est à chaque fois que vous exécuterez ce programme. Une bonne manière de d'ores et déjà mettre la technologie à votre service.

Question 1

[solution n°4 p.14]

Vous devez créer un programme qui, quand vous mettrez l'heure en argument au moment de son lancement, affichera « *Il est 14 h 45 !* » ou « *Il est 17 h 22 !* », etc.

Indice :

Dans le terminal, la commande permettant de récupérer l'heure est `$ (date +%R)`.

Question 2

[solution n°5 p.14]

Vous devez créer un programme avec une fonction `main()` appelant une autre fonction qui affiche l'heure et la date (sans la mettre en argument cette fois-ci).

Indice :

En Dart, il est possible de récupérer l'heure avec `DateTime.now()` ;

B. Test

Exercice 1 : Quiz

[solution n°6 p.14]

Question 1

En Dart, par où va commencer l'exécution du code d'un programme ?

- ☐ Par la première ligne
- ☐ Là où il y a l'instruction `begin`
- ☐ À la fonction `main()`

Question 2

De quel type est la fonction `main` ?

- ☐ `int`
- ☐ `void`
- ☐ `string`
- ☐ `double`

Question 3

Je peux nommer ma fonction de départ comme je le souhaite.

- ☐ Vrai
- ☐ Faux

Question 4

Quelle(s) syntaxe(s) est/sont juste(s) pour l'exécution d'un programme en Dart ?

- ☐ `dart run ./Desktop/monfichier.dart`
- ☐ `run dart monfichier.dart`
- ☐ `run dart ./monfichier.dart`
- ☐ `dart run monfichier.dart`

Question 5


Je n'ai le droit de mettre qu'un argument.

- ☐ Vrai
- ☐ Faux

Solutions des exercices

Exercice p. 3 Solution n°1**Question 1**

En Dart, par où va commencer l'exécution du code d'un programme ?

- ☐ Par la première ligne
- ☐ Là où il y a l'instruction begin
- ☒ À la fonction main()
-  La fonction main() est le point d'entrée de l'exécution de tout programme en Dart. La première ligne, c'est en Python, et l'instruction begin c'est en Pascal !


Question 2

Si je veux écrire un court programme affichant « *Hello World* » en Dart, quelle est la syntaxe ?

- ☒


```
Void main() {  
  print(« Le Dart c'est super »);  
}
```
- ☐

```
print(« Le Dart c'est super »)
```
- ☐

```
program HelloWorld(output);  
begin  
  writeln('Le Dart c'est super');  
readln;  
end.
```
-  Un programme en Dart doit forcément avoir une fonction main(), qui est le point de départ d'exécution du programme. Une simple ligne avec print() ne suffit pas, sauf en Python par exemple !

Question 3


Le Dart s'inspire du :

- ☐ Python
- ☐ Pascal
- ☒ C
- ☐ Ada
-  Le Dart s'inspire notamment du langage C (et par extension du C++, C#, etc.). En C, il faut aussi une fonction main().

Question 4

En Python, par où commence l'exécution du code ?

- ☐ Par la fonction main()
- ☒ Par la première ligne
- ☐ Par l'instruction begin

-  En Python, l'exécution du code commence par la première ligne du fichier. La fonction main() est utilisée en C et Dart, et l'instruction begin en Pascal.


Question 5

La syntaxe ci-dessous est correcte.

```
1 void main() {
2   maFonction();
3 }
4 void maFonction() {
5   print("Hello world");
6 }
```

☒ Vrai

☐ Faux

-  Il est tout à fait possible d'appeler d'autres fonctions depuis la fonction main(). C'est d'ailleurs toujours comme ça que cela fonctionne !


Exercice p. 5 Solution n°2

Question 1

Je peux nommer ma fonction de départ comme je le souhaite.

☐ Vrai

☒ Faux


-  La fonction de départ doit toujours s'appeler main.

Question 2

Je peux nommer le fichier ayant ma fonction de départ comme je le souhaite.

☒ Vrai

☐ Faux


-  Au moment de l'exécution du programme, vous indiquerez dans quel fichier se situe la fonction main(). Le nom que vous lui donnez est donc libre.

Question 3

Une fonction retourne toujours quelque chose.


☐ Vrai

☒ Faux

-  C'est vrai la plupart du temps, mais pas toujours ! Il y a des exceptions, comme avec la fonction main qui ne retourne rien.


Question 4

Il est possible de mettre des arguments dans une fonction main().

- ☒ Vrai
- ☐ Faux
-  Souvent, il n'est pas nécessaire de mettre des arguments. Néanmoins, cela est tout à fait possible et peut, dans certains cas de figure, s'avérer utile !

Question 5


De quel type est la fonction main ?

- ☐ int
- ☒ void
- ☐ string
- ☐ double
-  Puisque la fonction main ne retourne rien, son type est void.

Exercice p. 6 Solution n°3

Question 1


La commande `dart run` est une exécution :

- ☒ Autonome
- ☐ Native
- ☐ Anticipée
- ☐ En JavaScript
-  Une exécution autonome peut être lancée dans le terminal. La commande `dart run` est donc une exécution autonome.

Question 2

Quelle(s) syntaxe(s) est/sont juste(s) pour l'exécution d'un programme en Dart ?

- ☐ `run dart monfichier.dart`
- ☐ `run dart ./monfichier.dart`
- ☒ `dart run monfichier.dart`
- ☒ `dart run ./Desktop/monfichier.dart`

-  La bonne syntaxe est `dart run`. L'inverse ne sera pas compris par votre terminal ! Concernant la suite, il est possible d'indiquer uniquement le nom du fichier si celui-ci est dans le dossier courant, ou d'indiquer le chemin pour le trouver.

Question 3

Les arguments d'une fonction `main()` peuvent être de plusieurs types.

- ☐ Vrai
- ☒ Faux

- Q Les arguments d'une fonction `main()` sont toujours de type `List<String>`. Au moment où vous lancez votre programme, vous n'avez en effet la possibilité d'indiquer que du texte.

Question 4

À quel moment dois-je indiquer les arguments de la fonction `main` ?

- ☒ Au moment du lancement du programme
- ☐ Grâce à son appel depuis une autre fonction
- ☐ Juste après le lancement du programme

- Q Les arguments sont toujours indiqués au moment du lancement du programme, et donc à la suite de la commande d'exécution du programme quand il s'agit d'une exécution « *autonome* ».

Question 5

Le nombre d'arguments est limité.

- ☐ Vrai
- ☒ Faux

- Q Le nombre d'arguments n'est pas limité - vous en mettez autant que vous voulez !

p. 8 Solution n°4

```
1 void main(List<String> args) {
2   print(« Il est » + args[0] « ! »);
3 }
```

Exécution du code avec la commande :

```
dart run quelleheureilest.dart $(date +%R)
```

p. 8 Solution n°5

```
1 void main() {
2   afficherlheure();
3 }
4
5 void afficherlheure() {
6   final String heure = DateTime.now().toString();
7   print(heure);
8 }
```


Exercice p. 8 Solution n°6

Question 1

En Dart, par où va commencer l'exécution du code d'un programme ?

- ☐ Par la première ligne
- ☐ Là où il y a l'instruction `begin`


☒ À la fonction `main()`

 La fonction `main()` est le point d'entrée de l'exécution de tout programme en Dart. La première ligne, c'est en Python, et l'instruction `begin` c'est en Pascal !

Question 2

De quel type est la fonction `main` ?


- ☐ `int`
- ☒ `void`
- ☐ `string`
- ☐ `double`

 Puisque la fonction `main` ne retourne rien, son type est `void`.

Question 3

Je peux nommer ma fonction de départ comme je le souhaite.


- ☐ Vrai
- ☒ Faux

 La fonction de départ doit toujours s'appeler `main`.

Question 4

Quelle(s) syntaxe(s) est/sont juste(s) pour l'exécution d'un programme en Dart ?

- ☒ `dart run ./Desktop/monfichier.dart`
- ☐ `run dart monfichier.dart`
- ☐ `run dart ./monfichier.dart`
- ☒ `dart run monfichier.dart`

 La bonne syntaxe est `dart run`. L'inverse ne sera pas compris par votre terminal ! Concernant la suite, il est possible d'indiquer uniquement le nom du fichier si celui-ci est dans le dossier courant, ou d'indiquer le chemin pour le trouver.

Question 5

Je n'ai le droit de mettre qu'un argument.

- ☐ Vrai
- ☒ Faux

Q Vous mettez autant d'arguments que vous le souhaitez !