

Présentation du langage Dart

Table des matières

I. Présentation du langage Dart	3
A. Historique	3
B. Utilisations de Dart	4
C. Caractéristiques de Dart	4
II. Exercice : Quiz	5
III. Outils de développement	7
A. DartPad	7
B. Installation du Dart SDK.....	7
C. Installation de Visual Studio Code.....	9
D. Premier programme Hello world	9
IV. Exercice : Quiz	10
V. Essentiel	11
VI. Auto-évaluation	11
A. Exercice	11
B. Test.....	12
Solutions des exercices	13

I. Présentation du langage Dart

Contexte

Aujourd'hui, il existe principalement deux systèmes d'exploitation pour les smartphones iOS et Android. Cette application doit être présente sur les *stores* des deux systèmes. Ainsi, elle sera utilisée par un maximum d'utilisateurs. Pendant longtemps il a été nécessaire de développer dans le langage spécifique à chaque plateforme : Java / Kotlin pour Android et objective-C / Swift pour iOS. Dans une entreprise, il fallait donc un développeur pour chaque plateforme.

Pour réduire le temps de développement, des « *frameworks* » sont apparus pour faire des développements *multiplate-formes*. Ainsi, le code est le même pour les deux plateformes. Plusieurs *frameworks* à base de JavaScript ont vu le jour.

En 2018, Google a lancé le *framework* « *Flutter* » pour réaliser des applications multiplate-formes. Il s'appuie sur le langage de programmation Dart. Il est nécessaire de maîtriser ce langage pour réaliser des applications fiables et robustes.

Dans cette unité pédagogique, nous allons introduire le langage Dart et les outils de développement. Ensuite nous réaliserons notre premier programme en langage Dart.

A. Historique

Dart est un langage *open source* développé par Google depuis 2011. Son objectif est de remplacer le JavaScript et d'être utilisé sur toutes les plateformes. Il permet de développer des applications iOS, Android et des sites internet en utilisant le même code.

Lors de la conférence de Goto à Artus, au Danemark, du 10 au 12 octobre 2011, le langage Dart a été présenté. Le projet a été initié par Lars Bak et Kasper Lund. Lars Bak est un programmeur danois. Il est célèbre pour son travail sur les machines virtuelles. Il a contribué à la création du navigateur Chrome. Kasper Lund est un ingénieur logiciel.

Le langage Dart 1.0 a vu le jour le 14 novembre 2013. La syntaxe de Dart a de nombreuses similitudes avec les langages tels que JavaScript, Java, C++. Les développeurs habitués à ces langages devaient se sentir à l'aise. Dart était destiné à remplacer JavaScript dans les navigateurs. Il possédait sa propre machine virtuelle (VM) pour exécuter des applications natives dans le navigateur Chrome.

En juillet 2014, la première édition de Dart a été approuvée par l'organisme ECMA lors de sa 107^e assemblée générale.

Au départ, l'accueil de Dart a été mitigé et critiqué par certains développeurs qui lui reprochaient de fragmenter le web. Cette trajectoire a été abandonnée en 2015 avec la version 1.9 de Dart pour se concentrer sur la compilation de Dart en JavaScript.

La grande question est de savoir si Google avait initialement prévu que Dart remplacerait JavaScript. Les ingénieurs de Google auraient été frustrés par le maintien du langage JavaScript pour les bases de Gmail et de Google Maps. Ils ont commencé à utiliser un langage alternatif. La société a même sorti Dartium, un navigateur web centré sur Dart, peut-être dans l'optique de remplacer Chrome. Quelles que soient les intentions de Google, les développeurs ont toujours préféré s'en tenir à JavaScript.

Malgré ses déboires, ce n'était pas la fin de Dart. En effet, le lancement de Flutter a relancé Dart. Il avait pour but de permettre la création d'interfaces natives dans iOS et Android. Google dit avoir conçu Flutter pour prendre en charge des graphiques « *sans pépins et sans secousses* » à la vitesse native de votre appareil.

En août 2018, la version 2.6 de Dart a introduit une nouvelle extension : *dart2native*. Elle prolonge la compilation native aux plate-formes de bureau. Avant, les développeurs ne pouvaient créer de nouveaux outils, utilisant des appareils Android et iOS.

Aujourd'hui, la version courante de Dart est la 2.13.4.

B. Utilisations de Dart

Le code peut s'exécuter de quatre façons.

1. Il peut être compilé en JavaScript pour faire fonctionner des navigateurs. Pour cela, Dart s'aide d'un compilateur source-source en JavaScript. Lors de l'exécution du code Dart dans un navigateur web, le code est précompilé à l'aide du compilateur dart2js. Une fois compilé, la compatibilité est assurée pour tous les navigateurs.
2. Dart peut fonctionner de manière autonome. Le kit de développement (SDK) Dart est livré avec une machine virtuelle autonome. Ainsi le code Dart peut s'exécuter dans un environnement d'interface de lignes de commande.
3. Le code Dart peut être compilé par AOT en code machine. Les applications développées avec le *framework* Flutter s'appuyant sur Dart sont déployées dans les *stores* sous forme de code Dart compilé par AOT.
4. Avec la version 2.6 de Dart, le compilateur dart2native génère un code exécutable autonome et natif. Avant, seuls les appareils mobiles iOS et Android pouvaient exécuter le code de manière autonome.

C. Caractéristiques de Dart

Dart a de nombreuses similitudes avec d'autres langages comme Java, C#, Swift et Kotlin. Il a pour objectif de concurrencer et de remplacer JavaScript. Il veut combler ses manques. Il souhaite être plus performant et plus rigoureux. Dart est supporté par tous les systèmes d'exploitation de Windows à Mac en passant par Linux. Dart est optimisé pour développer des applications rapides dans un large panel de plateformes.

Fondamental

Dart est composé de :

- variables et opérateurs,
- classes,
- fonctions,
- boucles,
- commentaires,
- structures de données,
- mécanisme `null safety`,
- typage statique.

Il est orienté objet et supporte toutes les fonctionnalités comme l'héritage, les interfaces et le typage par inférence. Il a aussi le mécanisme `null safety`. Ce mécanisme introduit dans les langages modernes comme Swift permet de prévenir des crashes lorsqu'une variable est à `null`.

Méthode

Pour ajouter une variable dans le code on utilise le mot clé `var`.

```
var monNom = "Jean";
var age = 20;
```

Le compilateur Dart avec le mécanisme d'inférence va déduire le type des variables.

La propriété `runtimeType` permet d'afficher le type de la variable. À l'aide d'un `print`, on pourra afficher le type des variables `monNom` et `age`.

```
print(monNom.runtimeType);  
print(age.runtimeType);
```

En résultat, cela affichera : `String` pour `monNom` et `int` pour `age`.

Dart utilise les différents types basiques :

<code>int (num)</code>	121
<code>double (num)</code>	3.14159
<code>bool</code>	true/false
<code>String</code>	'Bonjour'
<code>dynamic</code>	42/3.14/true/'Hello'

L'utilisation du mot clé `dynamic` à la place la place de `var` permet un typage dynamique.

```
dynamic nomnbresDePersonnes = 5;  
print(nomnbresDePersonnes.runtimeType);  
nomnbresDePersonnes= "5 personnes";  
print(nomnbresDePersonnes.runtimeType);
```

Le premier affichage `print` donnera en sortie : `int`. Au deuxième affichage, on verra `String`. Le type a dynamiquement changé.

Nous reviendrons sur les différents types, de manière plus détaillée, dans les prochains chapitres.

Exercice : Quiz

[solution n°1 p.15]

Question 1

Quand Dart 1.0 a-t-il vu le jour ?

- ☐ 2013
- ☐ 2011
- ☐ 2018

Question 2

Quel langage Dart a-t-il tenté de remplacer ?

- ☐ Kotlin
- ☐ Swift
- ☐ JavaScript

Question 3

Quels sont les deux ingénieurs logiciels qui ont initié le projet Dart ?

- ☐ Lars Bak et Kasper Lund
- ☐ Linux Torvald et Dennis Ritchie

Question 4

Quel mot-clé permet de définir une variable ?

- ☐ let
- ☐ var

Question 5

À quel type de programmation appartient Dart ?

- ☐ Procédural
- ☐ Orienté objet
- ☐ Réactif

Question 6

Quel compilateur permet de convertir Dart en JavaScript ?

- ☐ Dartnative
- ☐ Dart2js
- ☐ AOT

Question 7

Comment afficher le type d'une variable `maVariable` ?

- ☐ `print (typeof:maVariable)`
- ☐ `print (maVariable.runtimeType)`

Question 8

Quel mécanisme permet de prévenir les crashes ?

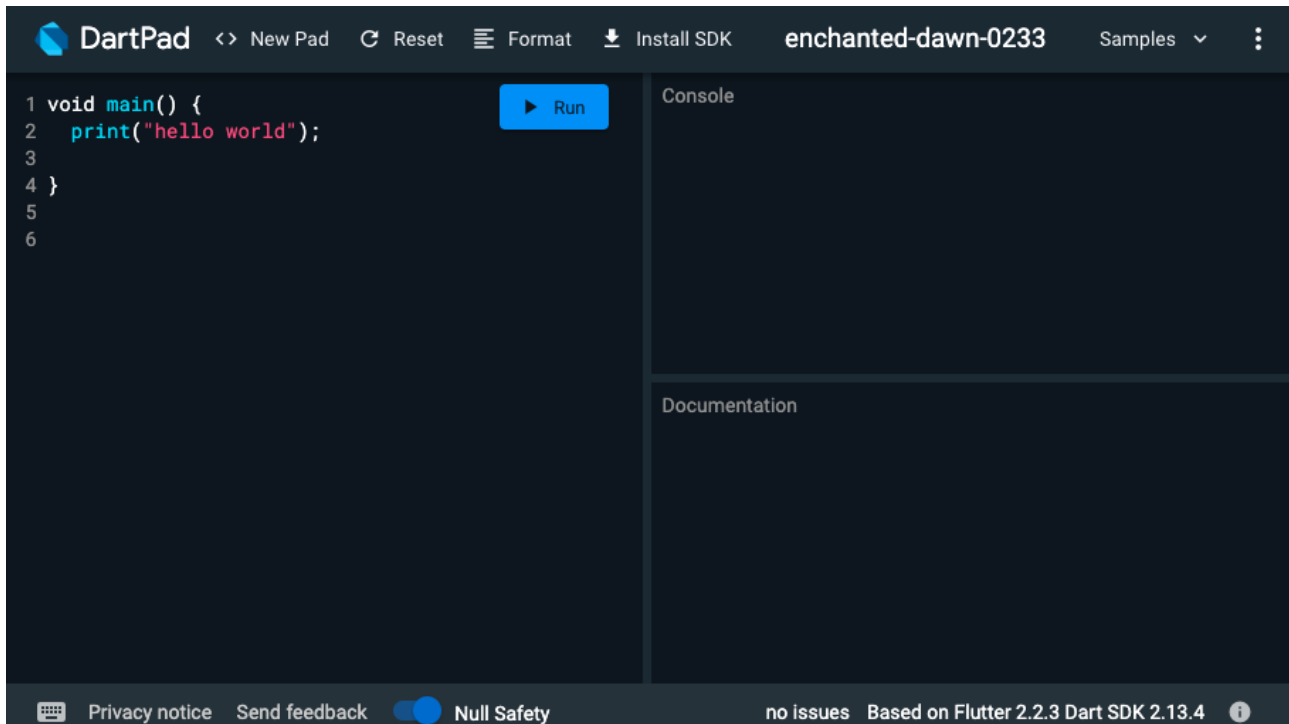
- ☐ Les interfaces
- ☐ `null safety`

III. Outils de développement

A. DartPad

DartPad est un outil en ligne dans un navigateur (Firefox, Chrome, Safari, etc.). Il est disponible à l'adresse : DartPad¹. Avec cet outil vous pouvez apprendre le Dart. Il prend en charge les bibliothèques principales de Dart. Il prend aussi en charge les bibliothèques Flutter et `dart:ui`.

DartPad ne prend pas en charge les bibliothèques tierces disponibles sur le référentiel de *packages* pub.dev.



Dans la fenêtre gauche, vous tapez votre code. Pour compiler et exécuter le code, il faut cliquer sur le bouton Run. Le résultat s'affiche dans la fenêtre de droite (console).

B. Installation du Dart SDK

Le kit de développement, Dart SDK, comprend une collection d'outils en lignes de commande qui permet de développer des applications Dart.

dart : ce Dart Virtual machine compile votre code.

dart2js : cet outil traduit du code Dart en JavaScript.

dart2native : cet outil compile du code Dart en un programme exécutable pour Windows, Linux ou MacOS.

Le kit de développement est disponible : Dart²

Les consignes d'installation sont disponibles pour Windows, MacOS et Linux.

Pour Windows, il faudra installer le gestionnaire de *package* Chocolatey disponible à l'adresse : Chocolatey³

1 https://dartpad.dev/?null_safety=true

2 <https://dart.dev/get-dart>

3 <https://docs.chocolatey.org/en-us/choco/setup#non-administrative-install>

Dans `cmd.exe` on exécutera :

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -
InputFormat None -ExecutionPolicy Bypass -Command "
[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" &&
SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
```

Une fois le gestionnaire de *package* installé, on pourra passer à l'installation du Dart SDK.

On procède toujours dans l'éditeur de `cmd.exe` ouvert en tant qu'administrateur.

Pour installer le Dart SDK :

```
C:\> choco install dart-sdk
```

Pour mettre à jour le Dart SDK :

```
C:\> choco upgrade dart-sdk
```

Pour MacOS, il faudra installer la gestion de *package* Homebrew. Exécuter la commande `brew` pour savoir s'il est installé.

S'il n'est pas installé, on exécutera dans le terminal la ligne de commande :

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Ensuite, pour installer Dart SDK, il faut exécuter les commandes suivantes :

```
$ brew tap dart-lang/dart
$ brew install dart
```

Pour mettre à jour le SDK, la commande est :

```
$ brew upgrade dart
```

Pour Linux, on utilisera le gestionnaire de *package* `apt-get`.

Pour l'installer et le configurer, on exécutera :

```
sudo apt-get update
$ sudo apt-get install apt-transport-https
$ sudo sh -c 'wget -qO- https://dl-ssl.google.com/linux/linux_signing_key.pub |
apt-key add -'
$ sudo sh -c 'wget -qO-
https://storage.googleapis.com/download.dartlang.org/linux/debian/dart_stable.list
> /etc/apt/sources.list.d/dart_stable.list'
```


Une fois le gestionnaire de *package* installé, on installe le Dart SDK :

```
$ sudo apt-get update
$ sudo apt-get install dart
```

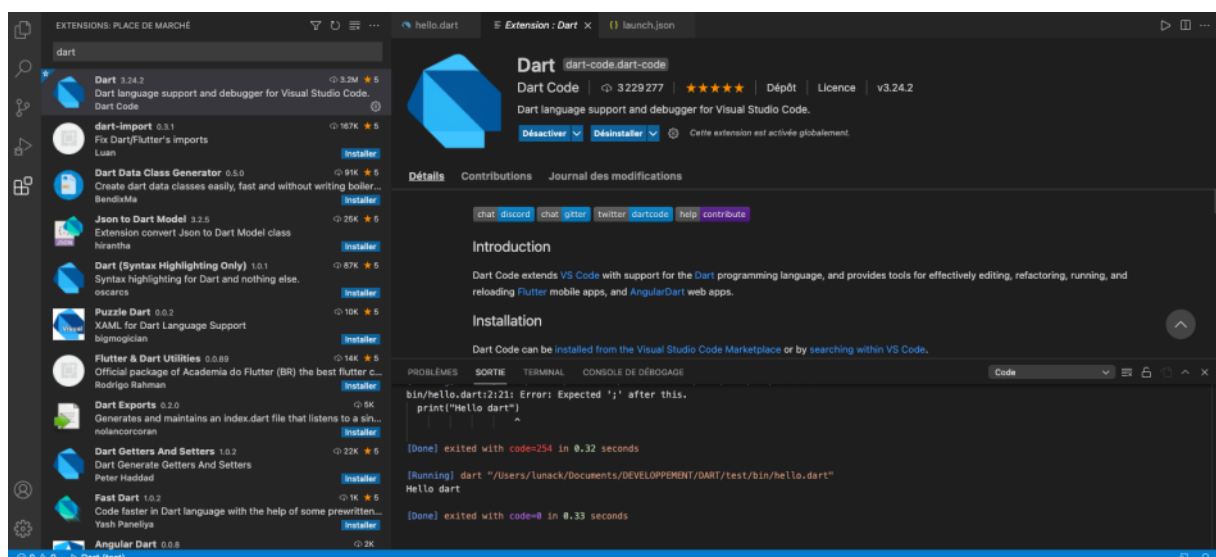
C. Installation de Visual Studio Code

Visual Studio Code est un IDE (environnement de développement intégré), *open source* et léger. Il est disponible sous Windows, Linux et MacOS. Il a été développé par Microsoft.

On le trouve à l'adresse : Visual Studio Code¹

Une fois téléchargé, on clique deux fois sur le *package* et on procède à son installation.

Il faut télécharger l'extension Dart : *.dart*.

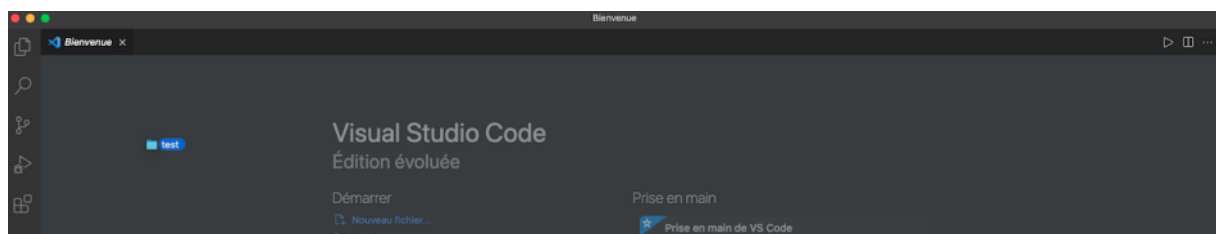


Voilà, maintenant, vous pouvez travailler.

D. Premier programme Hello world

Créez un dossier dans lequel vous allez éditer votre fichier principal.

Glissez-déposez le dossier dans la fenêtre principal de Visual Code.



Créez un nouveau fichier et enregistrez sous « *hello.dart* » (*.dart* est l'extension pour un fichier Dart).

Éditez le fichier *hello.dart* et écrivez la fonction principale *main*.

Pour commenter une ligne, on utilisera les symboles : *//*

¹ <https://code.visualstudio.com/>

Pour commenter un bloc, on utilisera `/*` pour commencer le bloc et `*/` pour terminer les blocs.



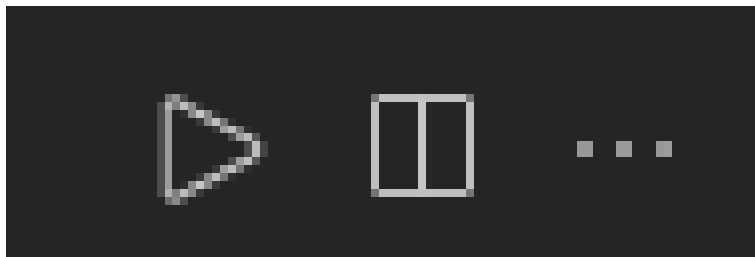
```

hello.dart x
hello.dart > ...
Run | Debug
1 void main() {
2   print("hello world");
3
4 }
5
6

```

Enregistrez le programme.

Exécutez-le en cliquant sur la troisième icône en partant de la gauche, en haut.



La sortie s'affiche dans la fenêtre du bas.



```

PROBLÈMES  SORTIE  TERMINAL  CONSOLE DE DÉBOGAGE
[Running] dart /Users/cunack/Documents/DEVELOPPEMENT/DART/test/hello.dart
hello world

[Done] exited with code=0 in 0.384 seconds

```

Voilà, vous avez fait votre premier programme. Vous pouvez continuer votre apprentissage de Dart.

Exercice : Quiz

[solution n°2 p.16]

Question 1

Comment s'appelle l'interface permettant de faire du Dart à l'intérieur du navigateur ?

- ☐ NoteDart
- ☐ DartPad
- ☐ DartEditor

Question 2

Comment s'appelle le gestionnaire de package pour installer le Dart SDK sous MacOS ?

- ☐ HomeBrew/Brew
- ☐ Apt-get
- ☐ Chocolatey

Question 3

Qui développe Visual Code ?

- ☐ Apple
- ☐ Microsoft
- ☐ Google

Question 4

Quel outil du SDK permet de compiler le code Dart ?

- ☐ dart2js
- ☐ dart

Question 5

Comment déclarer la fonction principale de notre programme ?

- ☐ `func main()`
- ☐ `void main()`
- ☐ `fu main()`

V. Essentiel

Au cours de cette unité pédagogique, nous avons vu l'historique du langage Dart depuis son premier lancement par Google en 2011.

Avant de s'imposer, le langage Dart a essuyé des critiques. Il était en concurrence avec JavaScript et visait à le remplacer.

Après quelques déboires, Dart a pris une autre dimension lorsque Google a lancé le *framework* Flutter. Ce dernier permet de créer des applications mobiles *cross-platform* en utilisant le même code.

Le langage Dart présente de nombreuses similitudes avec les langages modernes tels que Java, Swift, Kotlin. Il reprend les notions comme l'héritage, les interfaces et le typage par inférence. Il a aussi le mécanisme `null safety` qui permet de prévenir les crashes.

Ensuite, nous avons vu les différents outils nous permettant d'écrire des programmes en Dart : DartPad et Visual Studio Code. Pour ce dernier outil, l'installation du SDK Dart a été abordée. Nous avons ensuite effectué nos premières lignes de code à travers un traditionnel *Hello world*. Nous avons également initialisé quelques variables.

VI. Auto-évaluation

A. Exercice

Dans cet exercice, vous allez refaire un *Hello world* amélioré dans le DartPad.

Question 1

[solution n°3 p.17]

Faire quelques lignes, pour décrire notre programme.

Question 2

[solution n°4 p.17]

Déclarer la fonction principale et son bloc.

Question 3

[solution n°5 p.18]

Déclarer une variable pour stocker votre prénom.

Question 4

[solution n°6 p.18]

Afficher « *hello world* » et votre prénom sur deux lignes.

Question 5

[solution n°7 p.18]

Déclarer une variable `age` ne connaissant pas son type.

Question 6

[solution n°8 p.18]

Initialiser l'`age` en chaîne de caractères.

Question 7

[solution n°9 p.18]

Afficher le type de la variable.

B. Test

Exercice 1 : Quiz

[solution n°10 p.18]

Question 1

Quel *framework* utilise le langage Dart ?

- ☐ UIKit
- ☐ Swift UI
- ☐ Flutter
- ☐ Kotlin

Question 2

Qu'est-ce que Dartium ?

- ☐ Un langage de programmation
- ☐ Un *framework*
- ☐ Un navigateur web
- ☐ Un IDE

Question 3

Quelle société est à l'origine de Flutter ?

- ☐ Google
- ☐ Apple
- ☐ Yahoo
- ☐ Oracle

Question 4

Quel mécanisme permet de prévenir les crashes ?

- ☐ `null safety`
- ☐ Les optionnels
- ☐ Les exceptions

Question 5

Comment faut-il déclarer une variable pour qu'elle puisse changer de type ?


- ☐ `dynamic`
- ☐ `generic`
- ☐ `var`

Solutions des exercices

Exercice p. 5 Solution n°1**Question 1**

Quand Dart 1.0 a-t-il vu le jour ?


- ☒ 2013
- ☐ 2011
- ☐ 2018

 Dart 1.0 a été présenté en 2013. La syntaxe a de nombreuses similitudes avec JavaScript, Java, C++.

Question 2

Quel langage Dart a-t-il tenté de remplacer ?


- ☐ Kotlin
- ☐ Swift
- ☒ JavaScript

 Le langage Dart avait pour objectif de remplacer le langage JavaScript. Il a pour objectif d'être plus puissant et plus rigoureux.

Question 3

Quels sont les deux ingénieurs logiciels qui ont initié le projet Dart ?


- ☒ Lars Bak et Kasper Lund
- ☐ Linux Torvald et Dennis Ritchie

 Les deux ingénieurs qui ont initié le projet Dart sont Lars Bak et Kasper Lund. Ce sont des ingénieurs de Google.

Question 4

Quel mot-clé permet de définir une variable ?

- ☐ let
- ☒ var

 `var` permet de définir une variable. Lors de la compilation, Dart va déduire automatiquement le type de la variable (`String`, `int`, etc.).

Question 5

À quel type de programmation appartient Dart ?

- ☐ Procédural
- ☒ Orienté objet
- ☐ Réactif

Q Dart, comme de nombreux langages modernes, est orienté objet. Il reprend des mécanismes comme l'héritage, les interfaces ou le typage par inférence.

Question 6

Quel compilateur permet de convertir Dart en JavaScript ?

- ☐ Dartnative
- ☒ Dart2js
- ☐ AOT

Q Dart2js permet de convertir du Dart en JavaScript. Une fois compilé, le code pourra directement être exécuté par le navigateur.

Question 7

Comment afficher le type d'une variable `maVariable` ?

- ☐ `print (typeof:maVariable)`
- ☒ `print (maVariable.runtimeType)`

Q La fonction `print` permet de faire des affichages. `print (maVariable.runtimeType)` permet d'afficher le type de la variable. Cela permet de voir que Dart déduit implicitement le type de la variable.

Question 8

Quel mécanisme permet de prévenir les crashes ?

- ☐ Les interfaces
- ☒ `null safety`

Q Le mécanisme `null safety` permet de prévenir des crashes lorsqu'une variable est `null`. On retrouve ce système de prévention des crashes dans les langages modernes.

Exercice p. 10 Solution n°2

Question 1

Comment s'appelle l'interface permettant de faire du Dart à l'intérieur du navigateur ?


- ☐ NoteDart
- ☒ DartPad
- ☐ DartEditor

Q DartPad est l'éditeur de Dart en ligne. Il est compatible avec tous les navigateurs. Il intègre les librairies principales de Dart.

Question 2

Comment s'appelle le gestionnaire de package pour installer le Dart SDK sous MacOS ?


- ☒ HomeBrew/Brew
- ☐ Apt-get
- ☐ Chocolatey

 HomeBrew / Brew est le gestionnaire de package en lignes de commande. Il permet d'installer le SDK Dart.

Question 3

Qui développe Visual Code ?


- ☐ Apple
- ☒ Microsoft
- ☐ Google

 L'éditeur Visual Code a été développé par Microsoft. Il est gratuit. Il est présent sous Windows, MacOS et Linux.

Question 4

Quel outil du SDK permet de compiler le code Dart ?

- ☐ dart2js
- ☒ dart

 La commande `dart` permet de lancer le compilateur Dart. Ce dernier va compiler le code pour qu'il puisse être exécuté.

Question 5

Comment déclarer la fonction principale de notre programme ?

- ☐ `func main()`
- ☒ `void main()`
- ☐ `fu main()`

 `void main()` : La syntaxe a de fortes similitudes avec la syntaxe du langage C.

p. 11 Solution n°3

On commence un bloc de commentaire.

```
/* Ceci est un hello world amélioré  
on va déclarer plusieurs variables et les afficher  
*/
```

p. 12 Solution n°4

```
void main() {  
  
}
```

p. 12 Solution n°5

```
var monPrenom = "Eric"
```

p. 12 Solution n°6

```
print("hello world");
print(monPrenom);
```

p. 12 Solution n°7

```
dynamic age
```

p. 12 Solution n°8

```
age = "25 ans"
```

p. 12 Solution n°9

Pour afficher le type d'une variable il faut utiliser la méthode `runtimeType`.


```
print(age.runtimeType)
```

Exercice p. 12 Solution n°10

Question 1

Quel *framework* utilise le langage Dart ?


- ☐ UIKit
- ☐ Swift UI
- ☒ Flutter
- ☐ Kotlin

 Le *framework* Flutter utilise le langage Dart. Il a permis de relancer le langage Dart. Ce dernier avait été critiqué sur quelques points.

Question 2

Qu'est-ce que Dartium ?


- ☐ Un langage de programmation
- ☐ Un *framework*
- ☒ Un navigateur web
- ☐ Un IDE

 Dartium est un navigateur web avec l'objectif de remplacer Chrome. Il embarque une machine virtuelle Dart.

Question 3

Quelle société est à l'origine de Flutter ?


- ☒ Google
- ☐ Apple
- ☐ Yahoo
- ☐ Oracle

 Flutter a été lancé par Google en 2018. Il a relancé le langage Dart.

Question 4

Quel mécanisme permet de prévenir les crashes ?


- ☒ `null safety`
- ☐ Les optionnels
- ☐ Les exceptions

 `null safety` est une fonctionnalité de Dart pour prévenir les crashes quand une variable est `null`. On le retrouve dans les derniers langages modernes.

Question 5

Comment faut-il déclarer une variable pour qu'elle puisse changer de type ?

- ☒ `dynamic`
- ☐ `generic`
- ☐ `var`

 En définissant une variable `dynamic`, son type pourra changer. Une variable pourra être initialisée avec un entier. Cette variable pourra être utilisée pour une chaîne de caractères.