Le langage de programmation DART

1. Présentation

- Dart est un langage de programmation open-source créé par Google, basé sur des classes et éventuellement typé, destiné à la création d'applications Web à la fois sur le client et le serveur.
- Les objectifs de conception de Dart sont les suivants:
  - Créez un langage structuré mais flexible pour la programmation Web.
  - Rendez Dart familier et naturel aux programmeurs et donc facile à apprendre.

2. Qu’est-ce que le langage Dart ?

- **Dart** est un langage de programmation, qui a été développé principalement par Google. Il est standardisé par Ecma (Ecma est une organisation internationale de normalisation des systèmes d’information et de communication ainsi que des appareils électroniques grand public).
- La programmation dans **Dart** doit être une alternative attrayante à **JavaScript** dans des navigateurs Web modernes.
- D’après les développeurs de Dart, les faiblesses de JavaScript ne peuvent plus être corrigées par son développement.
- Le langage Dart est développé depuis 2010 et lancé la même année.
- Comme les navigateurs ne pouvaient et ne peuvent pas le manier en mode natif et que JavaScript peut être exécuté dans tous les navigateurs modernes, il existe le Compiler Dart2js, autrement dit « Dart vers JavaScript ».
- Le langage Dart ressemble aux langages de programmation orientés objet établis, dont Swift, C# ou Java font partie, qui sont soumis à des paradigmes de programmation déterminés.
- Les règles de combinaison de signes établis, c’est-à-dire la syntaxe, ressemblent au langage de programmation C. Cette « parenté » facilite grandement la prise en main de sorte que l’on peut y accéder sans avoir à surmonter trop d’obstacles.

3. Avantages et particularité de Dart

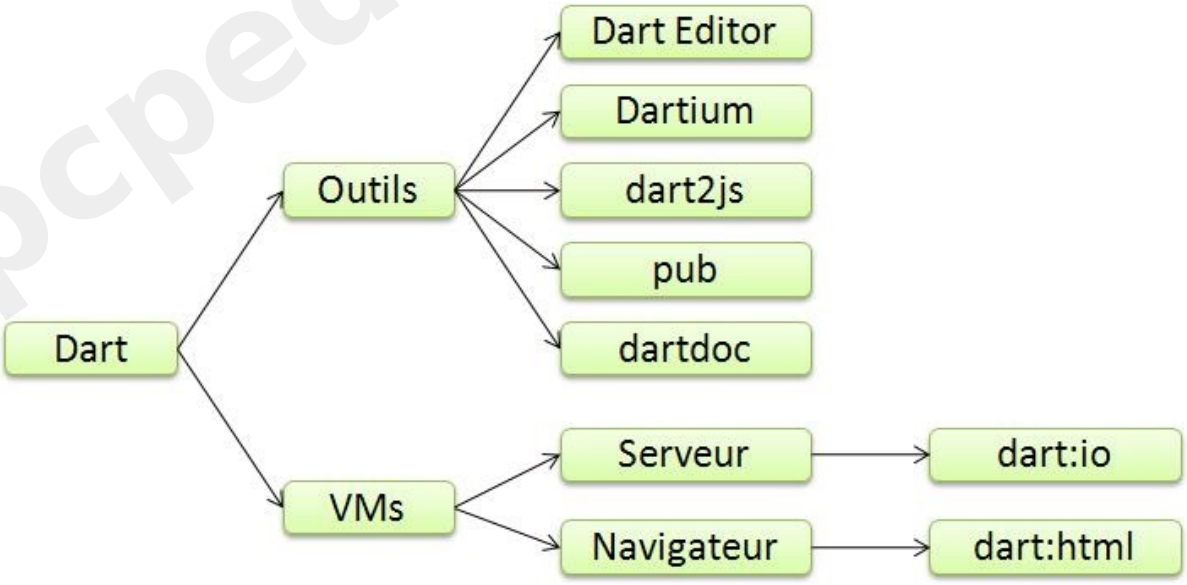
- Contrairement à de nombreux langages, Dart a été conçu dans le but de rendre le processus de développement aussi confortable et rapide que possible pour les développeurs. Il est donc livré avec un ensemble assez complet d'outils intégrés tels que son propre gestionnaire de paquets, divers compilateurs / transpilateurs, un analyseur et

apcpedagogie.com

un formateur.
- La machine virtuelle Dart et la version Just-in-Time rendent les modifications de code immédiatement exécutables.
- Une fois en production, le code peut être compilé en langue native, donc aucun environnement spécial n'est requis pour s'exécuter. En cas de développement web, Dart est transpilé en JavaScript.
- En ce qui concerne la syntaxe, Dart est très similaire à des langages comme JavaScript, Java et C++, donc apprendre Dart en connaissant l'un de ces langages est une opération assez simple.
- De plus, Dart prend en charge l'asynchronie et il est extrêmement facile de travailler avec des générateurs et des itérables.

4. L'écosystème

- Dart n’est pas seulement un langage, c’est une plateforme de développement qui comprend :
  - Un SDK avec une VM serveur
  - Un IDE (DartEditor), mais aussi des plugins pour Eclipse et IntelliJ
  - Une VM cliente (Dartium)
  - Un outil de gestion des dépendances (Pub Package Manager)
  - Un compilateur Dart vers JavaScript (dart2js)
  - Un générateur de documentation à partir du code (dartdoc)



5. **Éditeur Dart**

- Bien qu'il soit possible d'utiliser un simple éditeur de texte pour écrire du code Dart, utiliser un IDE est assez pratique. Dart Editor est basé sur Eclipse Rich Client Platform (RCP), un framework permettant de créer facilement des IDE. Par conséquent, son aspect est très similaire à Eclipse. En revanche, ne vous attendez pas aux mêmes fonctionnalités qui se limitent à une auto-complétion (sommaire), une analyse de code statique montrant erreurs et warnings, et une navigation facilitée entre les fichiers, classes, méthodes, etc.
- Il est bien sûr possible d'exécuter le code avec ou sans debug, ou bien de le convertir en Javascript.
- A l'heure où j'écris ces lignes, Dart Editor est loin d'être parfait, mais il s'améliore de version en version.

6. **Les Machines Virtuelles**

- Les Machines Virtuelles Dart sont le cœur du langage. Il en possède deux pour des usages différents. La première (VM Serveur) permet d'exécuter une application côté serveur en ligne de commande dans une console. La seconde (VM Client) est embarquée dans un navigateur tel que Dartium, comme nous le verrons dans le point suivant, au même titre que la Machine Virtuelle JavaScript. Ceci permet d'avoir la partie cliente et la partie serveur d'une application codées dans le même langage.
- L'unique différence entre ces deux versions est leurs librairies. La VM Serveur contient la library dart:io, qui permet d'effectuer les opérations d'entrée/sortie habituelles (manipulation de fichiers, sockets, etc.). La VM Cliente contient la library dart:html, permettant de manipuler le DOM (tags, events, etc.).
- Cette distinction est nécessaire pour des raisons de sécurité évidentes concernant la library dart:io. En revanche, bien que compréhensible, l'absence de la library dart:html de la VM Serveur rend compliqué, – mais pas impossible -, l'exécution en ligne de commande de tests unitaires validant du code utilisant cette library.
- Pour exécuter des tests unitaires de classes utilisant la library dart:html, il faut utiliser Dartium, ce qui est déconcertant, voire un problème pour l'industrialisation.

7. **Dartium**

- **Dartium** est une version modifiée de Chomium (la version Open Source de Chrome), avec une Machine Virtuelle Dart embarquée.
- Il reconnaît la balise **<script type="application/dart" …>** et exécute le code Dart dans le navigateur sans avoir à le convertir en JavaScript.
- Associé à Dart Editor, il permet de débugger simplement. Il vous suffit d'ajouter un breakpoint dans l'éditeur, rafraîchir votre page dans le navigateur, et le débugger de

l'éditeur s'arrêtera au breakpoint lorsqu'il l'aura atteint, vous permettant d'inspecter les variables et d'avancer step par step sur les instructions qui vous intéressent.

8. **dart2js**

   - dart2js permet de compiler du Dart en JavaScript, depuis Dart Editor ou en ligne de commande.
   - dart2js compile toutes les sources de votre application, ainsi que les librairies utilisées, en un seul fichier JavaScript.
   - Par défaut, le code n'est pas minifié. Pour ce faire, il suffit d'ajouter l'option –minify.
   - A noter qu'avant compilation, après construction de l'Abstract Syntax Tree du code, toutes les branches non utilisées (code mort ou fonctionnalités d'une librairie non utilisées dans notre application) sont « élaguées » ce qui permet de réduire considérablement le code généré. Ce mécanisme est connu sous le nom Tree Shaking.
     Source: https://blog.soat.fr/2013/01/dart-getting-started/

9. **DartDoc**

   - **DartDoc** est un outil permettant de générer de la documentation à partir du code source Dart. Pour générer la documentation de votre code Dart avec DartDoc, vous pouvez suivre ces étapes :
   - **Installer DartDoc**: Avant de pouvoir générer de la documentation, assurez-vous que DartDoc est installé. Vous pouvez l'installer à l'aide de pub, le gestionnaire de paquets de Dart :

     ```
     pub global activate dartdoc
     ```

   - Si vous utilisez Flutter, vous pouvez également utiliser flutter pub :

     ```
     flutter pub global activate dartdoc
     ```

   - **Documentez votre code** : Avant de générer de la documentation, assurez-vous que votre code source est correctement documenté. Vous pouvez ajouter des commentaires de documentation au-dessus des classes, des fonctions, des variables, des paramètres, etc., en utilisant des balises spécifiques à Dart, telles que /// pour les commentaires de documentation. Voici un exemple :

     ```
     /// Cette fonction additionne deux entiers.
     ///
     ```

```
/// [a] est le premier entier.
/// [b] est le deuxième entier.
/// Retourne la somme de [a] et [b].
int addition(int a, int b) {
  return a + b;
}
```

- Vous pouvez ajouter des commentaires de documentation aux classes, aux fonctions et aux membres de votre code.
- **Générez la documentation**: Une fois votre code correctement documenté, vous pouvez générer la documentation en exécutant DartDoc sur votre projet. Utilisez la commande suivante dans le répertoire racine de votre projet :
- Ou, si vous utilisez Flutter :

```
flutter pub global run dartdoc
```

- **Visualisez la documentation**: Après avoir généré la documentation, un répertoire doc sera créé à la racine de votre projet. Ouvrez le fichier HTML index.html situé dans le répertoire doc dans un navigateur web pour visualiser la documentation générée.
- DartDoc générera une documentation complète pour votre projet, y compris des informations sur les classes, les fonctions, les variables, les paramètres, les types de retour, etc., en utilisant les commentaires de documentation que vous avez ajoutés à votre code.
- Il est important de documenter votre code de manière approfondie, car une documentation de qualité facilite la compréhension de votre code par vous-même et par d'autres développeurs qui pourraient collaborer sur le projet.

Les forces et les faiblesses de Dart

- Dart présente une série d’avantages ainsi que quelques inconvénients.

1. **Avantages**

   - Le langage Dart est open source et donc accessible par tout le monde gratuitement. Google, grande entreprise leader, est à l’origine de Dart, ce qui offre une perspective à long terme pour le développement du langage.
   - Sa syntaxe le rend facile à apprendre pour les programmeurs. Les développeurs ont simplifié de nombreux termes qui sont inutilement complexes chez d’autres

apcpedagogie.com

langages et les ont regroupés intelligemment.

- Lorsque l'on a déjà manipulé au moins une fois C#, on parvient très rapidement à utiliser Dart. Le langage de programmation a été développé pour le Web.
- Grâce à une traduction directe (et très rapide) en JavaScript, Dart fonctionne sur tous les navigateurs modernes mobiles et de bureau. Un simple éditeur de texte suffit au travail de développement.
- Toutefois, des connaissances approfondies du langage de programmation sont nécessaires. Le mieux est de travailler avec des éditeurs spéciaux comme Android Studio (Google) ou Visual Studio Code (Microsoft).

2. **Inconvénients**

- Le langage Dart est relativement récent, ce qui a pour conséquence que la communauté d'entre-aide est relativement restreinte et les supports d'apprentissage ne sont aussi nombreux que pour JavaScript. Mais il faut admettre que la situation s'améliorera rapidement.
- La première installation d'un éditeur et du contexte technique correspondant sur un ordinateur est néanmoins bien documentée, mais relativement complexe.
- Selon ses détracteurs, Dart est l'un des nouveaux langages qui ont été lancés sur le marché au lieu de perfectionner les langages existants.

Cours Flutter

Source : https://apcpedagogie.com/le-langage-de-programmation-dart · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com