

# Découverte de DartPad

# Table des matières

<b>I. Introduction au DartPad</b>	<b>3</b>
A. Définition du DartPad.....	3
B. Accéder au DartPad.....	3
C. Différentes fonctionnalités du DartPad.....	4
<b>II. Exercice : Quiz</b>	<b>4</b>
<b>III. Prise en main du DartPad</b>	<b>6</b>
A. Exécution d'un programme .....	6
B. Modification d'un programme.....	7
C. Écriture d'un programme .....	8
<b>IV. Exercice : Quiz</b>	<b>10</b>
<b>V. Fonctionnalités avancées</b>	<b>11</b>
A. Formatage du code .....	11
B. Réinitialisation du programme.....	13
C. Documentation .....	14
D. Versions utilisées.....	15
<b>VI. Exercice : Quiz</b>	<b>16</b>
<b>VII. Essentiel</b>	<b>17</b>
<b>VIII. Auto-évaluation</b>	<b>17</b>
A. Exercice .....	17
B. Test.....	18
<b>Solutions des exercices</b>	<b>18</b>

## I. Introduction au DartPad

**Environnement de travail :** DartPad

**Pré requis :** pas de pré-requis

**Durée du cours :** 1 h 30

### Contexte

Lors de l'apprentissage d'un langage de programmation, on est amené à réaliser des exercices de code dont la difficulté croît avec le temps. La pratique du code et l'entraînement répété permet de progresser rapidement et d'appliquer ce qui est vu dans les cours théoriques.

Selon le langage de programmation, un environnement de développement doit être mis en place : éditeur de code, kit de développement, bibliothèque logiciel, compilateur, etc. Celui-ci doit être régulièrement mis à jour et permet au développeur d'effectuer son travail.

Dans le but de simplifier cette démarche pour les débutants, le langage Dart a mis à disposition un outil très utile : le DartPad. Ce dernier permet d'écrire du code en Dart directement dans un éditeur en ligne, pour ensuite l'exécuter.

À travers ce cours, nous allons décrire les différentes fonctionnalités de DartPad, montrer des exemples et effectuer des exercices dans le but de se familiariser avec cet outil qui se révèle bien utile dans l'apprentissage de Dart.

## A. Définition du DartPad

### Définition

**Le DartPad** est un outil Open Source qui permet d'exécuter du code Dart dans n'importe quel navigateur Web moderne : on peut donc le considérer comme un éditeur de code en ligne.

Son principal avantage est qu'il permet à n'importe quel utilisateur de manipuler le langage Dart sans avoir à installer un environnement de code. Il se révèle donc très utile lors de l'apprentissage des bases du langage.

## Présentation d'un code en DART

## B. Accéder au DartPad

Pour accéder au DartPad, il suffit d'ouvrir un navigateur Web moderne (Chrome, Firefox, Safari, Edge) puis de se rendre à l'adresse suivante :

Dartpad<sup>1</sup>

Il est aussi possible de taper « *DartPad* » dans Google et de se rendre sur le premier lien disponible.

Une fois sur le site de DartPad, celui-ci est directement accessible et utilisable. Nous allons voir dans la prochaine sous-partie les différentes fonctionnalités.

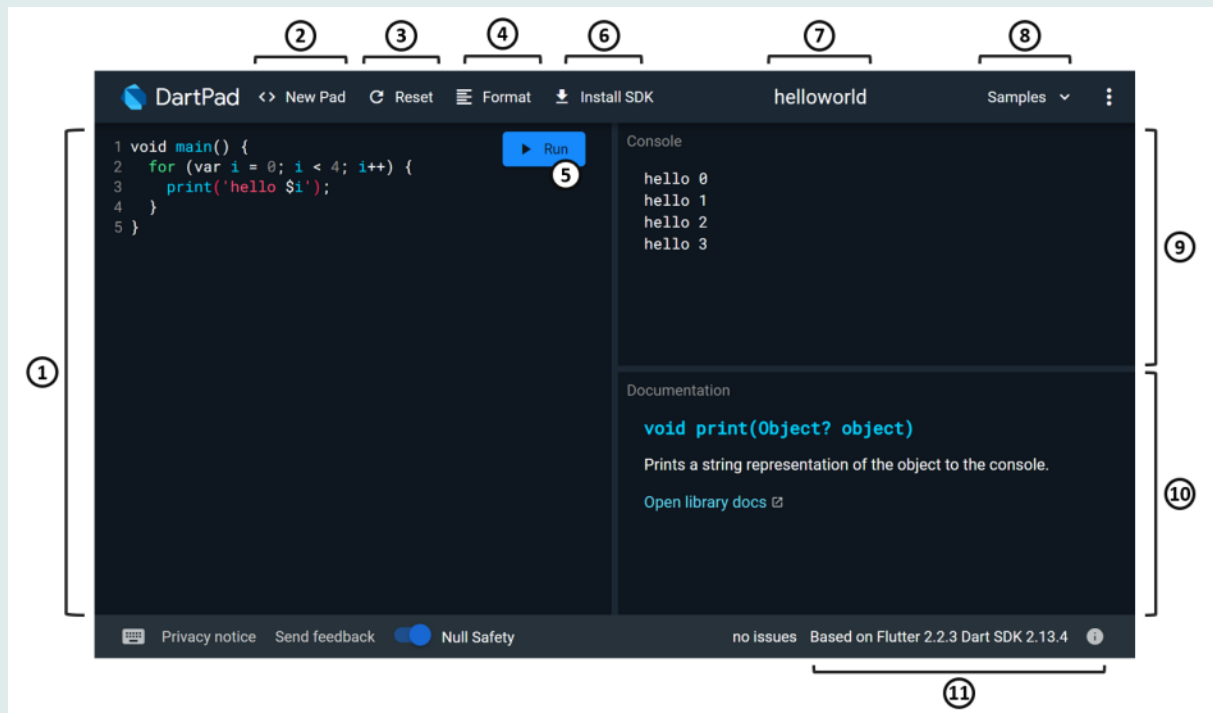
---

<sup>1</sup> <https://dartpad.dev/>

## C. Différentes fonctionnalités du DartPad

### Fondamental

Voici une capture du DartPad avec les annotations correspondantes :



1. Panneau d'édition du code.
2. New Pad : crée un nouveau Pad que vous pouvez renommer, pour partir de zéro.
3. Reset : efface votre travail et restaure l'éditeur dans son état initial.
4. Format : lance le formatage du code.
5. Run : exécute le code.
6. Install SDK : redirige vers une page pour installer le kit de développement Dart.
7. Nom du DartPad en cours.
8. Samples : met à disposition quelques programmes en Dart à titre d'exemple.
9. Console : affiche le résultat du code (output du programme).
10. Documentation : affiche des informations complémentaires par rapport à un élément de votre code (sur lequel est situé votre curseur).
11. Informations sur les versions du SDK et des différents modules utilisés.

La plupart des fonctionnalités évoquées ci-dessus seront utilisées dans les parties suivantes de ce cours : prise en main du DartPad et fonctionnalités avancées.

## Exercice : Quiz

[solution n°1 p.19]

### Question 1

Le DartPad permet d'exécuter du code :

- ☐ Dans n'importe quel langage
- ☐ En langage Dart

Question 2

Le DartPad peut être considéré comme un éditeur de code en ligne.

- ☐ Vrai
- ☐ Faux

Question 3

Le DartPad n'est pas gratuit à utiliser.

- ☐ Vrai
- ☐ Faux

Question 4

Le DartPad met à disposition des exemples de code.

- ☐ Vrai
- ☐ Faux

Question 5

Quelle fonctionnalité le DartPad ne possède-t-il pas ?

- ☐ Un éditeur de code
- ☐ Une console
- ☐ Un debugger avancé

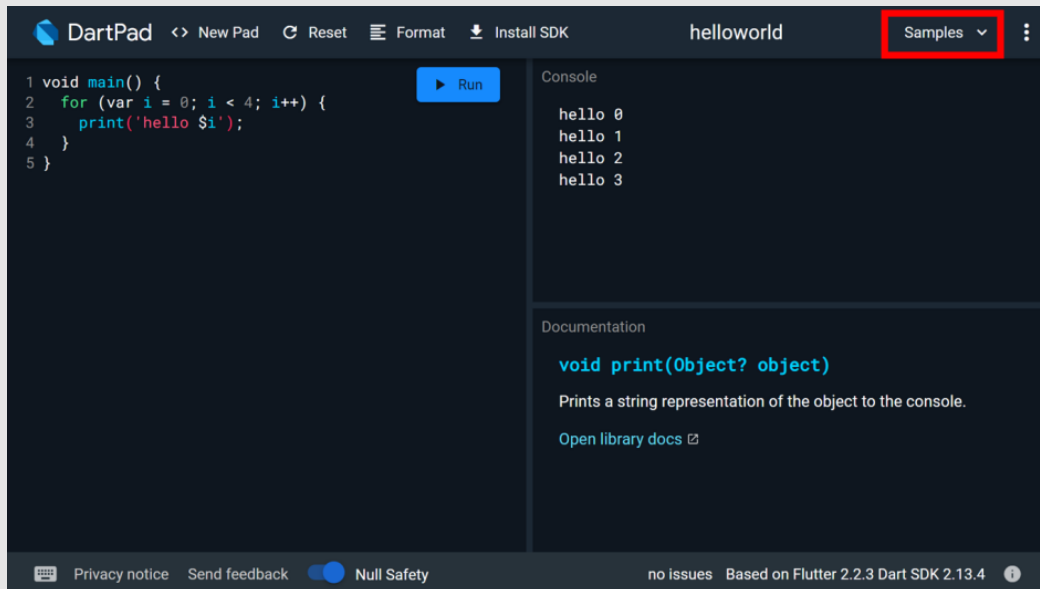
### III. Prise en main du DartPad

#### A. Exécution d'un programme

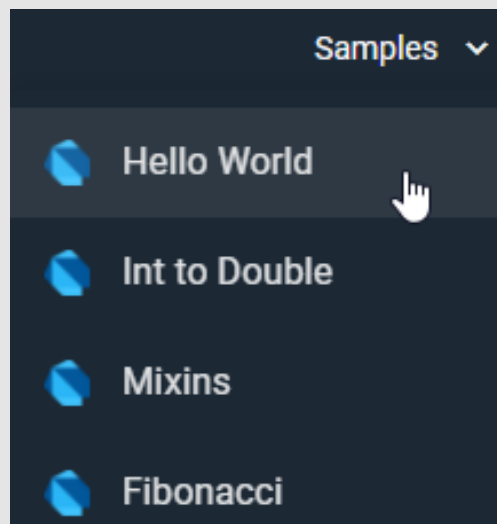
##### Méthode

Nous allons exécuter un premier programme avec le DartPad.

Pour cela, nous utilisons les « *Samples* » fournis par le DartPad en haut à gauche.



Nous sélectionnons le programme « **Hello World** » :



```
void main() {
  for (var i = 0; i < 4; i++) {
    print('hello $i');
  }
}
```

Vous devriez voir apparaître le programme ci-dessus à droite dans le panneau d'édition du code. Celui-ci est un code fonctionnel : il est donc possible de l'exécuter dès maintenant.

#### Remarque

Si les « *Samples* » proposés par DartPad ne sont pas les mêmes (DartPad évolue au fil du temps), vous pouvez exécuter n'importe quel autre « *Samples* » de la même façon. Pour la sous-partie suivante, vous n'aurez qu'à copier le code vu ci-dessus dans les captures.

#### Méthode

Avant d'exécuter le code, il faut toutefois prendre le temps de bien comprendre ce que fait le code. Ici, il s'agit simplement d'une boucle *for* qui va itérer 4 fois. À chaque tour de boucle, on affiche dans la console « *hello* » ainsi que l'index en cours.

Vous pouvez maintenant exécuter le code avec le bouton Run :



La console affiche alors la chose suivante, comme ce qui était prévu :

#### Console

```
hello 0  
hello 1  
hello 2  
hello 3
```

## B. Modification d'un programme

#### Méthode

Maintenant que nous sommes capables d'exécuter du code Dart au sein du DartPad, nous allons modifier le code et l'exécuter à nouveau par la suite. De cette façon, nous validerons le bon fonctionnement grâce au résultat affiché dans la console.

Le programme initial présent dans l'éditeur de code est celui-ci :

```
void main() {  
    for (var i = 0; i < 4; i++) {  
        print('hello $i');  
    }  
}
```

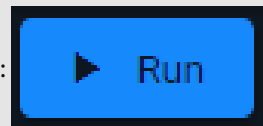
Nous souhaitons maintenant effectuer 10 tours de boucles (au lieu de 4) et afficher « *hello world* » (plus l'index) au lieu de « *hello* » (plus l'index).

Vous pouvez essayer d'effectuer les modifications de votre côté avant de regarder la solution ci-dessous.

Effectivement, il y a deux petites modifications à réaliser pour atteindre ce résultat :

```
void main() {
  for (var i = 0; i < 10; i++) {
    print('hello world $i');
  }
}
```

Vous pouvez maintenant exécuter le code avec le bouton Run :



La console affiche alors la chose suivante :

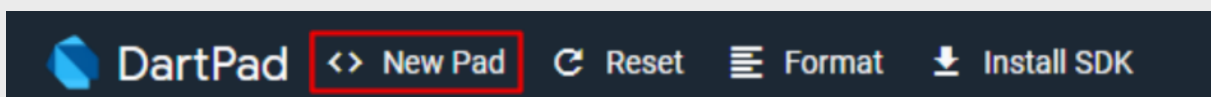
```
Console
hello world 0
hello world 1
hello world 2
hello world 3
hello world 4
hello world 5
hello world 6
hello world 7
hello world 8
hello world 9
```

## C. Écriture d'un programme

### Exemple

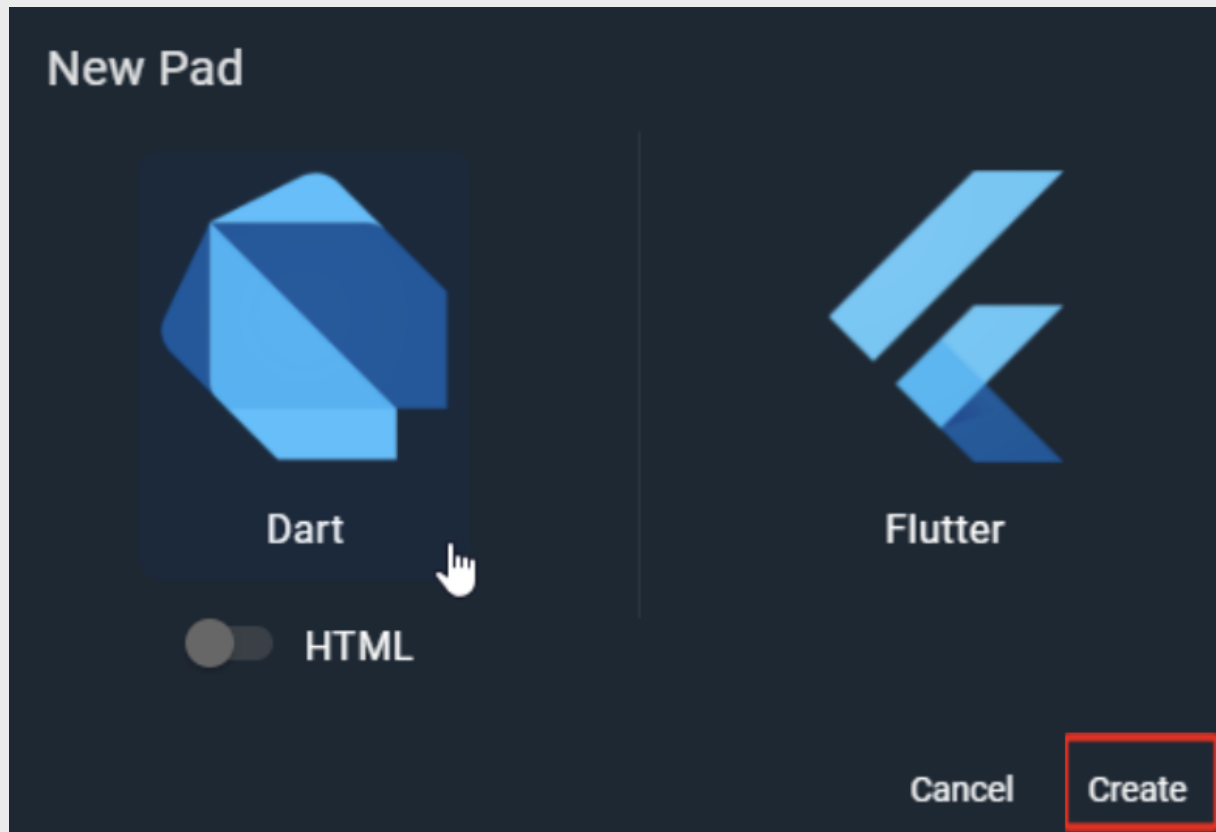
Afin de terminer cette deuxième partie du cours consacré à la prise en main du DartPad, nous allons réaliser un programme en partant de zéro.

Pour cela, cliquez sur « **New Pad** » afin de créer un nouveau DartPad.

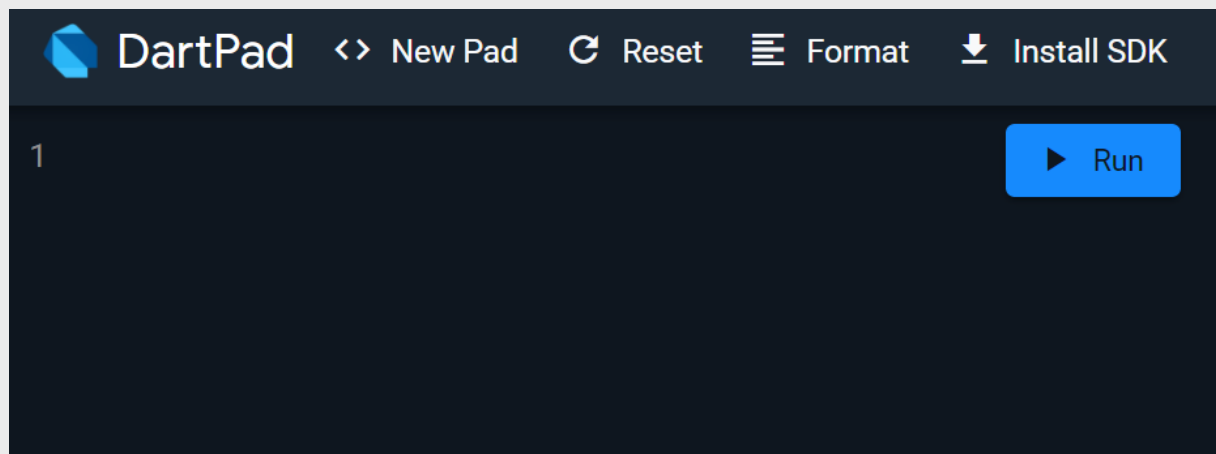


Ensuite, sélectionnez « **Dart** » et cliquez sur « **Create** ».





Effacez ensuite ce qui se situe dans l'éditeur de code pour repartir de zéro.



Pour commencer, il est indispensable de créer la fonction « Main ». Il s'agit du point d'entrée du programme.

```
1 void main() {  
2  
3 }
```

Ensuite, il est alors possible d'écrire n'importe quel code fonctionnel au sein de cette fonction.

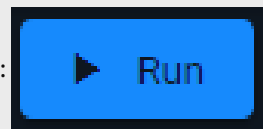
Pour l'exemple, nous allons créer deux variables contenant des nombres entiers, afficher ces entiers dans la console, puis afficher le résultat de leur somme.

Vous pouvez essayer d'effectuer cela de votre côté avant de regarder la solution ci-dessous.

Le code final ressemble à cela (plusieurs solutions sont possibles) :

```
1 void main() {  
2     int a = 3;  
3     int b = 8;  
4  
5     print("Premier entier : ${a}");  
6     print("Deuxième entier : ${b}");  
7     print("Sommes des deux entiers ${a+b}");  
8 }
```

Vous pouvez maintenant exécuter le code avec le bouton Run :



La console affiche alors la chose suivante :

```
Console  
  
Premier entier : 3  
Deuxième entier : 8  
Sommes des deux entiers 11
```

## Exercice : Quiz

[solution n°2 p.19]

### Question 1

Comment exécuter un programme avec le DartPad ?

- ☐ En appuyant sur le bouton « Run »
- ☐ Cela se fait automatiquement

### Question 2

Les exemples de code fournis par le menu déroulant Samples sont tous fonctionnels.

- ☐ Vrai
- ☐ Faux

### Question 3

Il est impossible de modifier un exemple de code fourni par le menu déroulant Samples.

- ☐ Vrai
- ☐ Faux

#### Question 4

Quel bouton permet de créer une nouvelle instance de DartPad ?

- ☐ Reset
- ☐ Format
- ☐ New Pad

#### Question 5

Une fois le code exécuté, où s'affiche le résultat en sortie ?

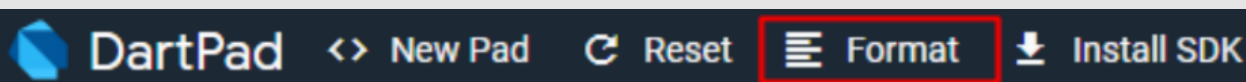
- ☐ Dans le panneau « *Console* »
- ☐ Dans le panneau « *Documentation* »

## V. Fonctionnalités avancées

### A. Formatage du code

#### Méthode

L'option de formatage du code permet d'autoformater le code présent dans l'éditeur de code. Formater du code signifie : appliquer une mise en forme qui respecte certaines normes. Ici, vous n'avez pas besoin de connaître les normes, tout s'applique en cliquant sur le bouton « **Format** » :



Reprenons le code du sample « *Hello World* » :

```
void main() {  
  for (var i = 0; i < 4; i++) {  
    print('hello $i');  
  }  
}
```

Appliquez une mise en forme aléatoire :

```
void main() {  
  
    for (var i = 0; i < 4;  
        i++) {  
        print('hello $i')  
  
        ;  
    }  
}
```

Enfin, cliquez sur le bouton « **Format** » en haut à gauche de la page.

```
void main() {  
    for (var i = 0; i < 4; i++) {  
        print('hello $i');  
    }  
}
```

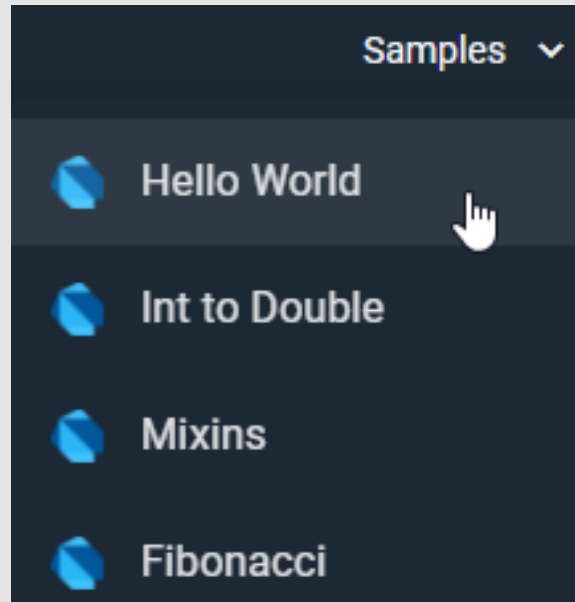
Le code a alors repris sa forme d'origine, c'est-à-dire celle qui respecte le bon formatage d'un code en Dart. Lorsque vous codez au sein du DartPad, vous pouvez prendre l'habitude de formater le code régulièrement afin d'avoir une mise en forme correcte.

## B. Réinitialisation du programme

### Méthode

Le DartPad offre la possibilité de réinitialiser le programme en cours. Cela s'effectue en cliquant sur le bouton « **Reset** » en haut à gauche de la page.

Par exemple, si vous avez sélectionné un sample depuis la liste des « *Samples* » disponibles :

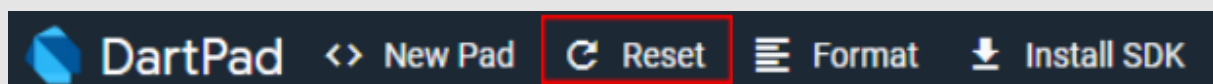


```
void main() {  
  for (var i = 0; i < 4; i++) {  
    print('hello $i');  
  }  
}
```

Et que vous modifiez ce programme (de n'importe laquelle des façons) :

```
void main() {  
  for (var i = 0; i < 10; i++) {  
    print('hello world $i');  
  }  
}
```

Vous pouvez revenir à la version de base en cliquant sur le bouton « **Reset** » :



```
void main() {  
  for (var i = 0; i < 4; i++) {  
    print('hello $i');  
  }  
}
```

Cette action aura pour effet d'effacer votre travail et de restaurer l'éditeur dans son état initial.

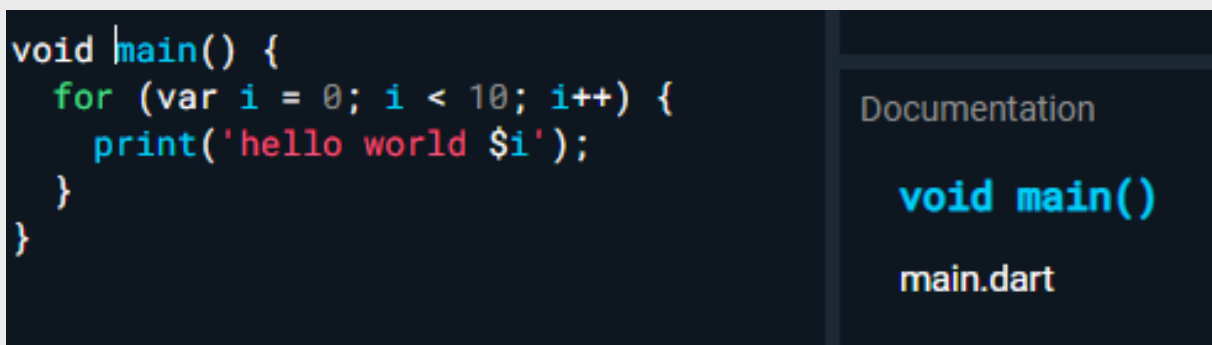
### C. Documentation

Le panneau « *Documentation* » affiche des informations complémentaires par rapport à un élément de votre code (sur lequel est situé votre curseur), lorsque cela est possible.

Lorsque votre curseur arrive sur un nouvel élément du code, DartPad va rechercher dans la documentation du langage Dart pour essayer de trouver une correspondance. Si l'élément est présent dans la documentation, le panneau « *Documentation* » en donne les références, parfois accompagné d'une définition et d'un lien vers les documentations officielles.

#### Exemple

Exemple lorsque le curseur est sur l'élément « *main* » :



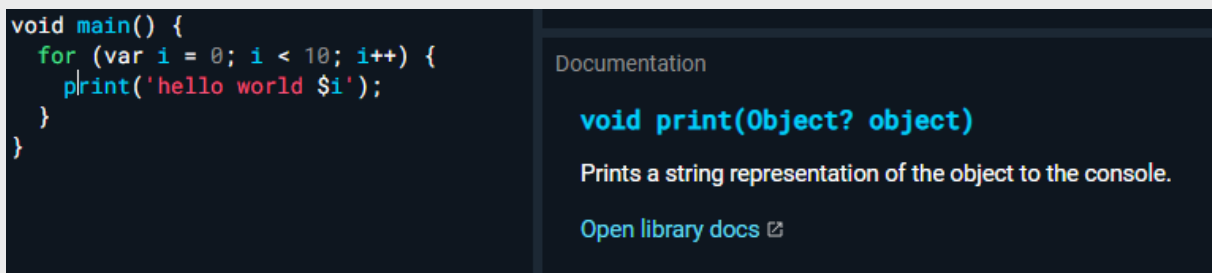
The screenshot shows a code editor with the following Dart code:

```
void main() {  
  for (var i = 0; i < 10; i++) {  
    print('hello world $i');  
  }  
}
```

The cursor is positioned on the word `main`. To the right, a 'Documentation' panel is visible, displaying:

```
void main()  
main.dart
```

Exemple lorsque le curseur est sur l'élément « *print* » :



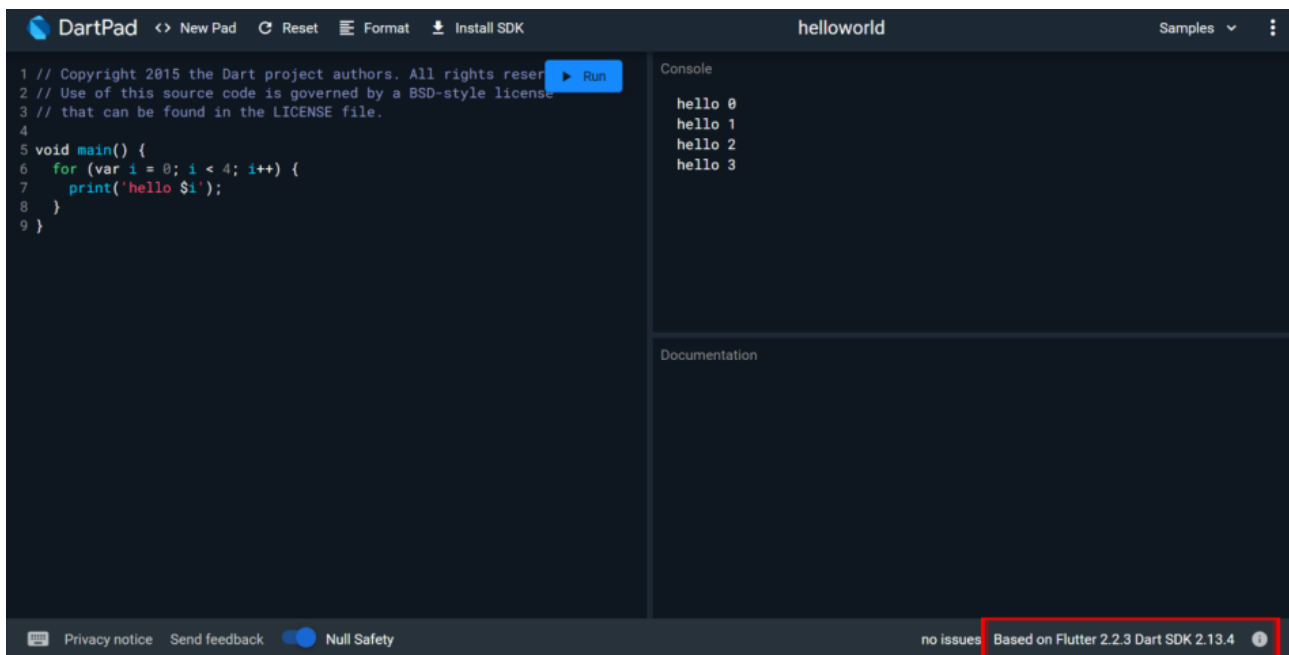
The screenshot shows the same code editor as before, but the cursor is now on the word `print`. The 'Documentation' panel on the right displays:

```
void print(Object? object)  
Prints a string representation of the object to the console.  
Open library docs
```

Cette fonctionnalité se révèle très pratique lorsque l'on souhaite approfondir ou comprendre une notion que l'on ne maîtrise pas complètement au sein du code.

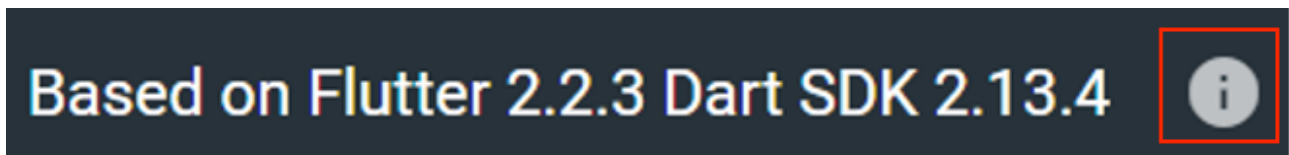
## D. Versions utilisées

En bas à droite du DartPad, on retrouve des informations relatives aux versions de Dart et Flutter utilisées par le DartPad.



En général, ces versions correspondent aux dernières versions disponibles de Dart et Flutter.

En cliquant sur l'icône d'information :



On retrouve les versions des différents packages utilisés au sein de DartPad.

Pub package versions	
async	2.6.1
boolean_selector	2.1.0
characters	1.1.0
charcode	1.2.0
clock	1.1.0
cloud_firestore	2.4.0
cloud_firestore_platform_interface	5.3.0
cloud_firestore_web	2.3.0
cloud_functions	3.0.0
cloud_functions_platform_interface	5.0.8
cloud_functions_web	4.0.10
collection	1.15.0

Il n'est pas nécessaire de s'intéresser à tout ce système de versions si l'objectif est d'apprendre les bases du langage Dart.

Ces informations sont utiles dans le cas où l'on installe par la suite un véritable environnement de développement, pour vérifier la correspondance avec ce que l'on a pu effectuer dans DartPad.

## Exercice : Quiz

[solution n°3 p.20]

### Question 1

À quoi sert le bouton « **Format** » du DartPad ?

- ☐ À formater mon navigateur Internet
- ☐ À autoformater le code

### Question 2



Le bouton « **Reset** » permet de réinitialiser l'intégralité du DartPad.

- ☐ Vrai
- ☐ Faux

#### Question 3

Le panneau « *Documentation* » permet d'avoir des informations lorsque vous placez votre curseur sur certains éléments du code.

- ☐ Vrai
- ☐ Faux

#### Question 4

Il est possible de connaître la version exacte de Dart utilisée par le DartPad.

- ☐ Vrai
- ☐ Faux

#### Question 5

Quelle autre information est disponible dans le DartPad ?

- ☐ Le nombre de fois où vous avez ouvert DartPad
- ☐ La version des différents packages utilisés par le DartPad
- ☐ Le nombre de caractères utilisés dans l'éditeur de code

## VII. Essentiel

DartPad se révèle être un outil de qualité pour apprendre les bases du langage Dart. En fournissant un éditeur de code en ligne facilement accessible, il devient possible de coder en Dart et d'exécuter son programme sans avoir à installer tout un environnement de développement.

Les fonctionnalités que l'on retrouve sont celles d'un véritable éditeur de code : console pour la sortie du programme, formatage, documentation, etc.

Nous avons vu dans ce cours comment exploiter au mieux DartPad. De la prise en main aux fonctionnalités avancées, on constate rapidement que l'outil est très facile à manipuler. Des exemples de code sont notamment mis à disposition.

Une fois les bases apprises, il faudra passer à un environnement de développement complet : les fonctionnalités y sont quand même supérieures et il s'agit du vrai cadre de travail pour un développeur. DartPad se veut comme une étape transitoire dans le parcours d'un débutant et a pour but de faciliter les premiers pas avec le langage Dart.

## VIII. Auto-évaluation

### A. Exercice

Dans cet exercice, nous allons utiliser et mettre en application les différentes fonctionnalités du DartPad vues dans ce cours. Les actions demandées doivent être effectuées via le DartPad<sup>1</sup> :

#### Question 1

[solution n°4 p.22]

Ouvrez DartPad dans un navigateur Internet.

---

<sup>1</sup> <https://dartpad.dev/>

### Question 2

[solution n°5 p.22]

Créez une nouvelle instance de DartPad.

### Question 3

[solution n°6 p.22]

Utilisez le panneau d'édition du code pour créer un programme qui affiche uniquement « *Hello world* » dans la console. Exécutez le programme.

### Question 4

[solution n°7 p.23]

Réinitialisez l'éditeur de code. Exécutez le programme.

### Question 5

[solution n°8 p.23]

Ajoutez des espaces inutiles au sein du code puis auto-formatez le code.

### Question 6

[solution n°9 p.24]

Repérez la version du Dart SDK puis affichez-la dans la console avec la fonction `print`.

## B. Test

### Exercice 1 : Quiz

[solution n°10 p.24]

#### Question 1

Quel est l'intérêt d'utiliser un outil comme DartPad ?

- ☐ Il apporte des fonctionnalités plus avancées que les éditeurs de code classiques
- ☐ Il permet de s'entraîner sur le langage Dart sans avoir à installer tout l'environnement de développement

#### Question 2

DartPad rassemble l'ensemble des fonctionnalités que l'on peut retrouver dans un environnement de développement classique.

- ☐ Vrai
- ☐ Faux

#### Question 3

Quelles fonctionnalités avancées proposent DartPad ?

- ☐ Reset, autoformatage, debug
- ☐ Reset, autoformatage, documentation
- ☐ Reset, debug, documentation

#### Question 4

DartPad est accessible n'importe quand et de n'importe où via un navigateur Web moderne.

- ☐ Vrai
- ☐ Faux

#### Question 5


Les développeurs professionnels utilisent DartPad pour coder en Dart.

- ☐ Vrai
- ☐ Faux

## Solutions des exercices


**Exercice p. 4 Solution n°1****Question 1**

Le DartPad permet d'exécuter du code :

- ☐ Dans n'importe quel langage
- ☒ En langage Dart
-  Le DartPad est destiné à exécuter du code uniquement pour le langage Dart, d'où son nom.


**Question 2**

Le DartPad peut être considéré comme un éditeur de code en ligne.

- ☒ Vrai
- ☐ Faux
-  Le DartPad s'exécute dans la plupart des navigateurs Web modernes.


**Question 3**

Le DartPad n'est pas gratuit à utiliser.

- ☐ Vrai
- ☒ Faux
-  Le DartPad est complètement gratuit et Open Source.


**Question 4**

Le DartPad met à disposition des exemples de code.

- ☒ Vrai
- ☐ Faux
-  Le DartPad propose des exemples de code via le menu déroulant Samples.

**Question 5**


Quelle fonctionnalité le DartPad ne possède-t-il pas ?

- ☐ Un éditeur de code
- ☐ Une console
- ☒ Un debugger avancé
-  Le DartPad ne possède pas de fonctionnalité pour debugger le code.

**Exercice p. 10 Solution n°2**


### Question 1

Comment exécuter un programme avec le DartPad ?

- ☒ En appuyant sur le bouton « *Run* »
- ☐ Cela se fait automatiquement
-  Une fois le code prêt, on l'exécute en appuyant sur le bouton « *Run* » en haut à gauche de l'éditeur de code.


### Question 2

Les exemples de code fournis par le menu déroulant Samples sont tous fonctionnels.

- ☒ Vrai
- ☐ Faux
-  Une fois sélectionnés, les exemples de code sont directement exécutables.


### Question 3

Il est impossible de modifier un exemple de code fourni par le menu déroulant Samples.

- ☐ Vrai
- ☒ Faux
-  Une fois sélectionné, un exemple de code est entièrement modifiable.


### Question 4

Quel bouton permet de créer une nouvelle instance de DartPad ?

- ☐ Reset
- ☐ Format
- ☒ New Pad
-  Le bouton « *New Pad* » en haut à gauche du DartPad permet de recréer une instance (en sélectionnant la cible du programme : Dart ou Flutter).

### Question 5


Une fois le code exécuté, où s'affiche le résultat en sortie ?

- ☒ Dans le panneau « *Console* »
- ☐ Dans le panneau « *Documentation* »
-  Le panneau « *Console* » affiche le résultat (output) du programme exécuté.

### Exercice p. 16 Solution n°3


**Question 1**

À quoi sert le bouton « **Format** » du DartPad ?

- ☐ À formater mon navigateur Internet
- ☒ À autoformater le code
-  En appuyant sur le bouton « *Format* », cela applique automatiquement la bonne mise en forme au code.


**Question 2**

Le bouton « **Reset** » permet de réinitialiser l'intégralité du DartPad.

- ☐ Vrai
- ☒ Faux
-  Le bouton « *Reset* » réinitialise uniquement le programme en cours dans l'éditeur de code.


**Question 3**

Le panneau « *Documentation* » permet d'avoir des informations lorsque vous placez votre curseur sur certains éléments du code.

- ☒ Vrai
- ☐ Faux
-  Le panneau « *Documentation* » se réfère à la documentation Dart pour apporter des informations complémentaires lorsque cela est possible.


**Question 4**

Il est possible de connaître la version exacte de Dart utilisée par le DartPad.

- ☒ Vrai
- ☐ Faux
-  La zone d'information tout en bas à droite du DartPad apporte cette information.

**Question 5**

Quelle autre information est disponible dans le DartPad ?

- ☐ Le nombre de fois où vous avez ouvert DartPad
- ☒ La version des différents packages utilisés par le DartPad
- ☐ Le nombre de caractères utilisés dans l'éditeur de code
-  En cliquant sur le bouton d'information en bas à droite, il est possible de consulter la version des différents packages utilisés par le DartPad.

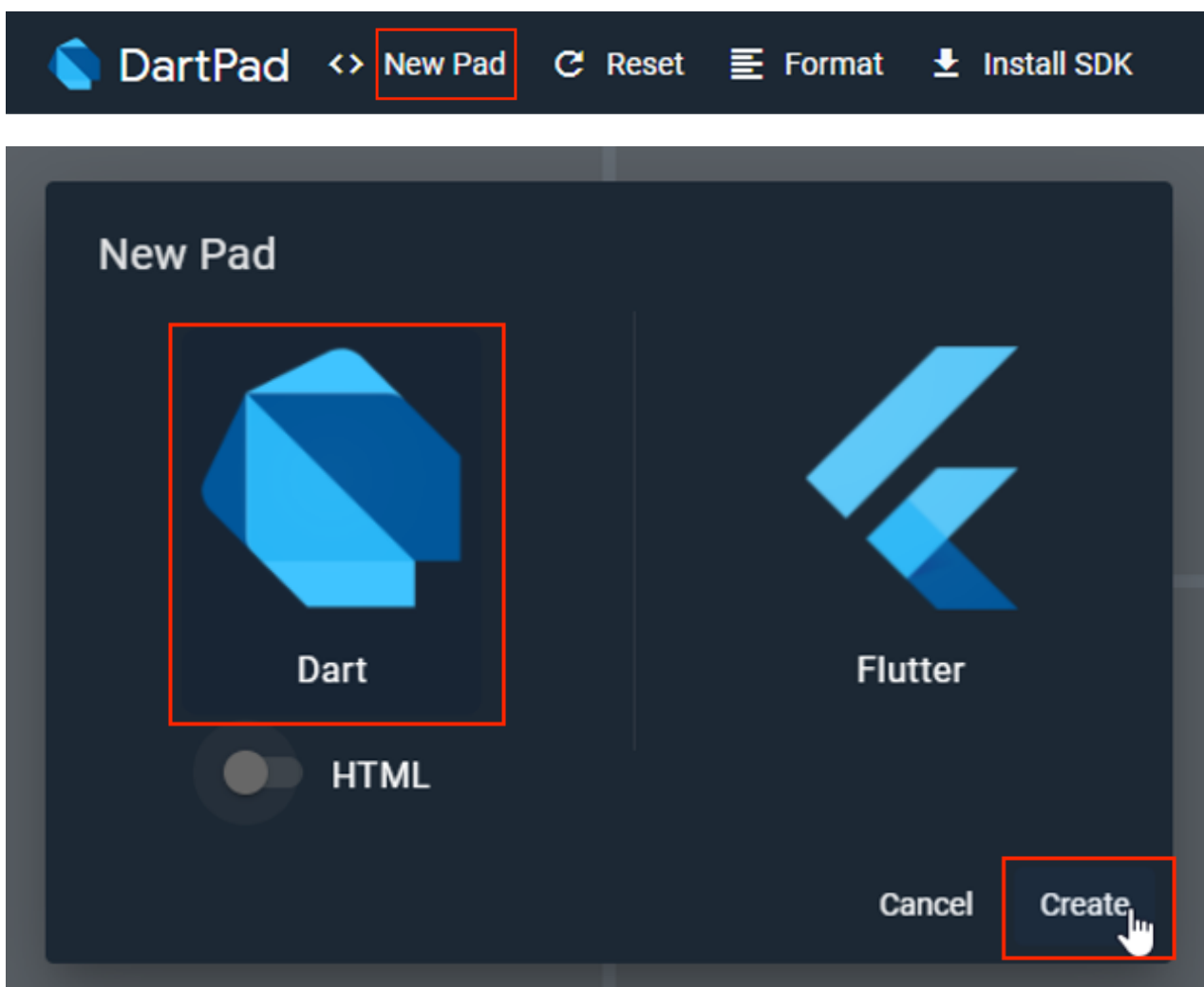
p. 17 **Solution n°4**

Entrez l'URL <https://dartpad.dev/> dans un navigateur moderne.



p. 18 **Solution n°5**

Cliquez sur « **New Pad** », sélectionnez Dart puis cliquez sur « **Create** ».



p. 18 **Solution n°6**

Voici un exemple du programme attendu et le résultat affiché dans la console :

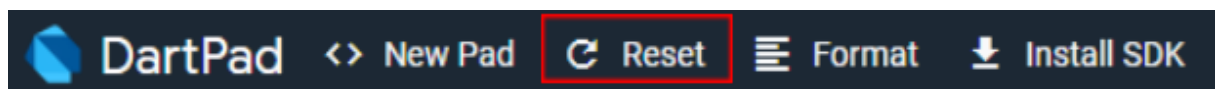
```
void main() {  
    print("Hello world");  
}
```

Console

Hello world

p. 18 Solution n°7

Cliquez sur « **Reset** », puis sur le bouton « **Run** » pour exécuter le code :

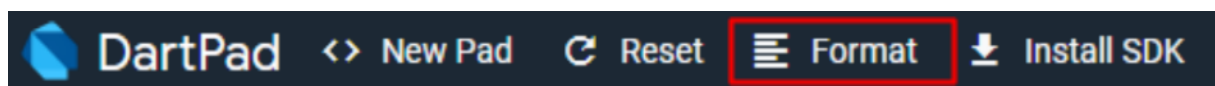


p. 18 Solution n°8

Ajoutez des espaces inutiles :

```
void main() {  
    for (var i = 0; i < 4; i++) {  
  
        print('hello $i');  
  
    }  
}
```

Puis cliquez sur « **Format** » :



Le code s'autoformate et revient à la normale :

```
void main() {  
  for (var i = 0; i < 4; i++) {  
    print('hello $i');  
  }  
}
```

p. 18 Solution n°9

Repérez la version de Dart SDK en bas à droite du DartPad :

no issues Based on Flutter 2.2.3 Dart SDK 2.13.4

Puis utilisez la fonction « *print* » pour afficher le résultat dans la console :

```
void main() {  
  print("Dart SDK 2.13.4");  
}
```

Console

Dart SDK 2.13.4

Exercice p. 18 Solution n°10

Question 1

Quel est l'intérêt d'utiliser un outil comme DartPad ?

- ☐ Il apporte des fonctionnalités plus avancées que les éditeurs de code classiques
- ☒ Il permet de s'entraîner sur le langage Dart sans avoir à installer tout l'environnement de développement
- ☐ De façon provisoire, DartPad permet d'exécuter du code Dart sans devoir installer un environnement de développement complet.




**Question 2**

---

DartPad rassemble l'ensemble des fonctionnalités que l'on peut retrouver dans un environnement de développement classique.

☐ Vrai

☒ Faux

 Bien que DartPad apporte de nombreuses fonctionnalités, il est loin d'être aussi complet qu'un vrai environnement de développement.

**Question 3**


---

Quelles fonctionnalités avancées proposent DartPad ?

☐ Reset, autoformatage, debug

☒ Reset, autoformatage, documentation

☐ Reset, debug, documentation

 Le DartPad ne propose pas de fonctionnalité de debug.

**Question 4**

---

DartPad est accessible n'importe quand et de n'importe où via un navigateur Web moderne.

☒ Vrai

☐ Faux

 DartPad se veut gratuit et accessible à tout moment.


**Question 5**

---

Les développeurs professionnels utilisent DartPad pour coder en Dart.

☐ Vrai

☒ Faux

 DartPad est principalement dédié à l'apprentissage ou pour tester de petits programmes.