

# Kapitel 1: Die ersten Schritte<sup>1</sup>

## Dart Editor installieren

1. Laden Sie den Dart Editor herunter: <https://www.dartlang.org/tools/editor/>
2. Entpacken Sie das Zip mit dem Dart Editor auf Ihre Festplatte.
3. Falls Sie noch keine Java Runtime installiert haben, laden Sie diese herunter und installieren Sie sie. Dart Editor benötigt Java in der Version 6 oder höher.
4. Starten Sie den Dart Editor mit einem Doppelklick auf die ausführbare Datei im Installationsordner.

Bei Problemen mit der Installation, suchen Sie auf <https://www.dartlang.org/tools/editor/troubleshoot.html> nach einer Lösung.

## Was enthält die Installation?



DartEditor

Im Dart Editor wird programmiert. Damit können Dart-Projekte erstellt, editiert und verwaltet werden.



Chromium

Eine Spezielle Version des Chrome Web Browsers, genannt **Dartium**, welcher die Dart VM (virtual machine) enthält. Dart-Programme können direkt in diesem Browser laufen. Der Dart Editor führt Projekte automatisch darin aus.




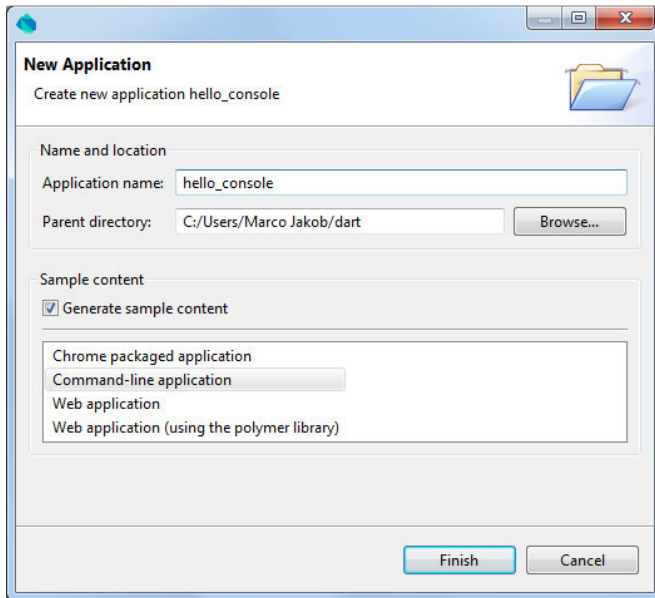
dart-sdk

Das Verzeichnis **dart-sdk** enthält das Dart software development kit. Hier befinden sich die Dart Standard-Bibliotheken.

<sup>1</sup> Arbeitsblätter dieses Kapitels basieren z.T. auf dem Buch „Einführung in Java mit Greenfoot“ von Michael Kölling (2010). Ideen und Konzepte von Kara wurden entwickelt von Jürg Nievergelt, Werner Hartmann, Raimond Reichert et al., <http://www.swisseduc.ch/informatik/karatojava/>, abgerufen Februar 2011.

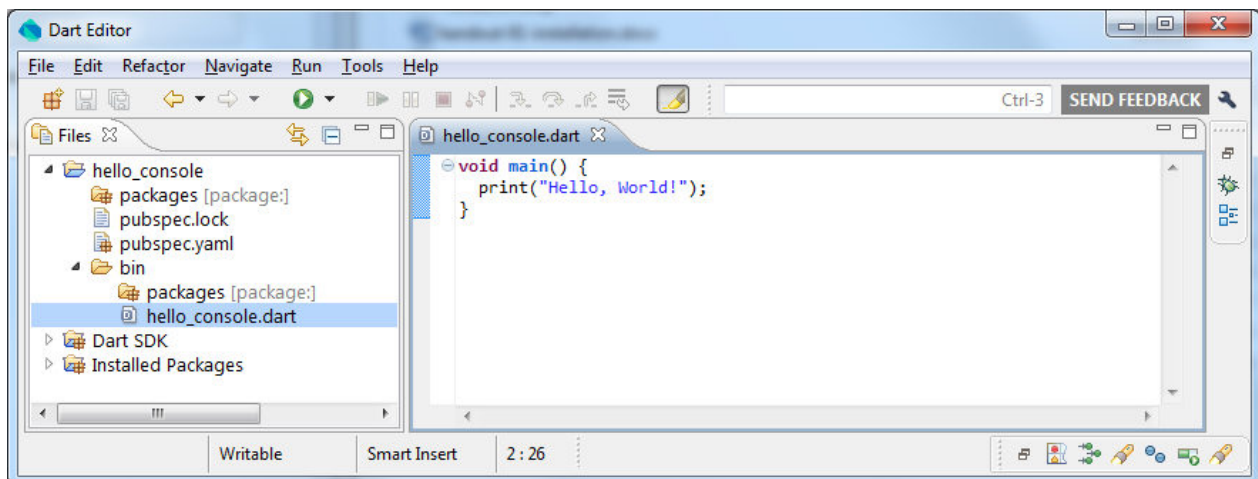
## Erstes Programm mit Konsole

1. Klicken Sie auf den *New Application* Knopf  (oder auch über das Menu *File* → *New Application...*).
2. Geben Sie einen Namen für die Applikation an, z.B. **hello\_console**.
3. Markieren Sie die **Generate sample content** Checkbox und **Command-line application**.



Die Standardstruktur für Konsolen-Applikationen wird generiert. Eine erste Dart-Datei wurde erstellt mit folgendem Inhalt:

```
void main() {  
  print("Hello, World!");  
}
```



4. Klicken Sie mit der rechten Maustaste auf die Dart-Datei und wählen Sie **Run**. Nun sollte in der Konsole unten Rechts „Hello, World!“ erscheinen.

## Das Kara Szenario

Sie brauchen nun den Ordner **dart-kara-chapter-1** entpackt auf Ihrem Computer. Darin befindet sich ein Dart Projekt. Sie finden den Ordner entweder auf dem USB-Stick oder auf der Internetseite <http://edu.makery.ch> unter *Learn Dart*.

1. Öffnen Sie den Ordner **dart-kara-chapter-1** im Dart Editor mit dem Menu *File* → *Open Existing Folder...*
2. Das Projekt erzeugt eine grafische Oberfläche in einem Web-Browser, deshalb ist es ein **Web-Projekt**. Damit ein Projekt als Web-Projekt funktioniert, braucht es einen **web-Ordner**. Es ist ausserdem abhängig vom **browser package**. Diese Abhängigkeit wird in der Datei **pubspec.yaml** definiert, worauf der Dart Editor automatisch die Dateien herunterlädt in den Ordner **packages**.

**Hinweis:** Falls das browser package oder die darin enthaltenen Dateien nicht vorhanden sind, dann kann dies durch ein Rechtsklick auf *pubspec.yaml* → *Pub Install* behoben werden.


## Kara kennen lernen

Das Kara-Projekt enthält im Ordner **web** eine Datei **my\_kara.dart**. Wir werden vor allem mit dieser Datei arbeiten. Falls Sie möchten, können Sie natürlich auch die anderen Dateien studieren.

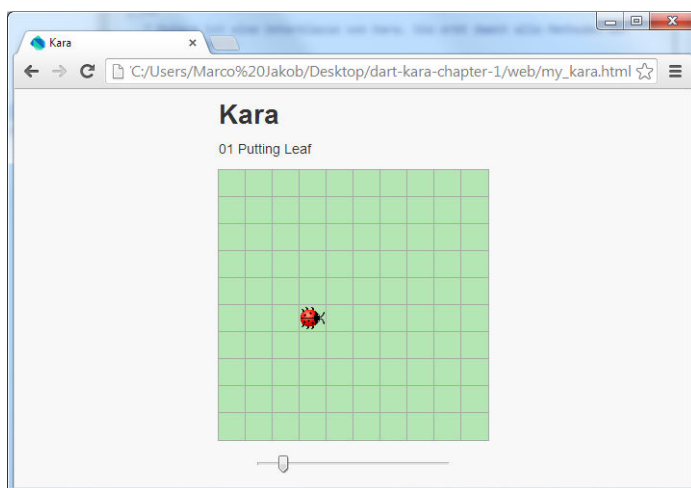
Öffnen Sie nun die Datei **my\_kara.dart**.

In dieser Datei hat es zwei Blöcke, genannt Funktionen (oder auch Methoden), mit Namen **act()** und **main()** und ein paar **blaue Kommentare** zur Beschreibung. Die main-Funktion braucht es, damit der Computer weiss, wo das Programm beginnt. Uns wird vorläufig aber vor allem die **act()**-Methode beschäftigen.

Starten Sie das Programm: Rechtsklick auf die Datei **my\_kara.dart** → *Run in Dartium*

**Tipp:** Von jetzt an können Sie das Programm mit der Taste  in der Taskleiste starten oder mit *Ctrl+r*.

Es erscheint nun das erste Szenario im Dartium Browser ähnlich folgender Abbildung:



- Die *Welt*: Dies ist der Bereich, in dem das Programm ausgeführt wird und in dem wir verfolgen können, was passiert. Im Kara-Szenario ist die Welt eine grüne Wiese mit Gitterlinien.
- Der *Regler*: Mit dem Schieberegler kann die Geschwindigkeit von Kara eingestellt werden.
- *Neustart*: Um das Programm neu zu starten kann man, wie in der Webentwicklung üblich, einfach die Internetseite neu laden (Taste F5).

## Klassen und Objekte

**Achtung: Die folgenden Maus-Interaktionen sind im aktuellen Prototyp noch nicht implementiert!**

Für unsere Projekte verwenden wir die Programmiersprache **Dart**. Dart ist eine sogenannte *objektorientierte* Sprache. Für objektorientierte Programmierung sind die Konzepte von Klassen und Objekten von grundlegender Bedeutung.

Betrachten wir einmal die **Klasse MyKara**. Die **Klasse MyKara** steht für das allgemeine Konzept eines Käfers – sie ist sozusagen wie ein Bauplan, aus dem wir einzelne Käfer erstellen können. Die erzeugten Käfer werden als **Objekte** (oder auch Instanzen) bezeichnet.

Mit der Maus erstellen wir neue Objekte wie folgt: Rechte Maustaste<sup>2</sup> auf die Wiese und Menüpunkt *new MyKara()* auswählen. Die Objekte können auch mit der Maus verschoben werden.



### AUFGABE (NOCH NICHT MÖGLICH!)

Erzeugen Sie ein Objekt von Kara. Erzeugen Sie mehrere Objekte von den Kleeblättern (Leaf).

## Mit Objekten interagieren

Zum Interagieren mit Objekten in der Welt klicken wir sie mit der rechten Maustaste an, um das **Objektmenu** aufzurufen. Das Objektmenu von **Kara** zeigt uns, was dieser spezielle Kara alles machen kann. Diese Operationen werden in Java **Methoden** genannt.

### AUFGABE (NOCH NICHT MÖGLICH!)

a) Was bewirkt die Methode **move ()** ?

**Lösung: Kara geht ein Feld nach vorne.**

b) Platzieren Sie zwei Karas in Ihrer Welt und sorgen Sie dafür, dass sie sich anschauen. Welche Methode brauchen Sie dazu?

**Lösung: turnLeft() oder turnRight()**

c) Testen Sie die weiteren Methoden. In welche zwei Arten können die Methoden unterteilt werden?

**Lösung: Methoden mit void führen nur eine Aktion aus.**

**Methoden mit bool öffnen ein Fenster mit dem Methodenergebnis.**

<sup>2</sup> Wenn die Maus, z.B. bei Mac OS-Rechnern, nur eine Taste hat, gleichzeitig mit der Maustaste die Ctrl-Taste drücken!

## Ausführen von Programmen

Bis jetzt haben wir immer nur mit Mausklicks mit den Objekten interagiert. Es gibt aber noch eine weitere Möglichkeit, nämlich indem wir Programme schreiben.

### AUFGABE 1

a) Öffnen Sie die Datei **my\_kara.dart** im Dart Editor. Was bewirkt die Methode `act()`?

**Lösung: Schritt vorwärts, Drehung nach rechts, Schritt vorwärts**

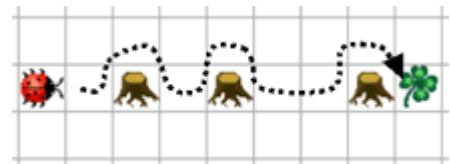
b) Ändern Sie den Inhalt der `act()`-Methode so, dass Kara zuerst einen Schritt macht, dann ein Kleeblatt legt und wieder einen Schritt macht. (Am Anfang der Klasse finden Sie als Kommentar alle Aktionen, die Kara durchführen kann.)

*Beachten Sie:* Nach jedem Befehl muss ein Strichpunkt stehen!

c) Speichern Sie Ihre Änderung (Ctrl+s). Um das Programm zu überprüfen können Sie einfach im Dartium Browser F5 drücken.

### AUFGABE 2

In der `main()`-Funktion sehen Sie, dass das Szenario 01 geladen wird. Ändern Sie es ab in `scenario02()`. Nun sollte bei einem Start die Welt automatisch wie rechts abgebildet erscheinen:



*Hinweis:* Die Welt wird jeweils aus der Datei **scenarios.dart** geladen.

*Sie können in dieser Datei herausfinden, wie ein Szenario zusammengebaut wird.*

Schreiben Sie jetzt ein Programm, welches Kara auf dem angegebenen Weg zum Kleeblatt führt (siehe Grafik). Er muss dabei um die Bäume herumlaufen. Beim Kleeblatt angekommen, soll er es aufheben.

## Kara erhält neue Methoden

### AUFGABE 3

Wenn Sie die Aufgabe 2 korrekt gelöst haben, so wird Ihr Programm wahrscheinlich **drei gleiche Teile** enthalten, nämlich für das Herumgehen um jeden Baum. Dies können wir zur besseren Übersicht noch etwas erweitern, indem wir eine neue Methode einführen. Unterhalb der `act()`-Methode erstellen wir eine neue Methode:

```
void goAroundTree() {  
  
}
```

Jede Methode hat eine öffnende und schliessende, geschweifte Klammer `{ }`. Schreiben Sie zwischen die geschweiften Klammern der Methode die Befehle, die es braucht, um um den Baum zu kommen.

Benutzen Sie nun innerhalb der `act()`-Methode die Methode `goAroundTree()` für jeden der drei Bäume.