

Dart Demo

Notizbuch: Mobile Usergroup Zentralschweiz

Erstellt: 18.11.2018 18:26

Geändert: 18.11.2018 19:00

Autor: loana.albisser@bluewin.ch

Ort: 47°1'25 N 8°17'31 E

<https://codelabs.developers.google.com/codelabs/from-java-to-dart/#5>

Bicycle-Klasse definieren

```
class Bicycle {  
    int speed;  
    int gear;  
}  
  
void main() {  
}
```

- Alle Methoden sind per Default public
- speed und gear sind Properties

Konstruktor definieren

```
Bicycle(this.speed, this.gear)
```

- Kurzform eines Konstruktors
- Parameter werden automatisch initialisiert
- Muss kein Body haben

Klasse instantieren

```
void main() {  
    var myBike = Bicycle(2,1);  
    print(myBike);  
}
```

- new-Keyword ist optional
- var kann verwendet werden

Output verbessern

```
@override  
String toString() => 'Bicycle: $speed kmh';
```

- Mit Override-Annotation kann Methode überschrieben werden
- mit '\$'-kann Variable in String verwendet werden
- Einzeilige Funktionen können mit dem Arrow kurz geschrieben werden

Read-only Variable verwenden

- speed vom Konstruktor entfernen
- _ hinzufügen um dies Private machen
- _speed mit 0 initialisieren
- speed-getter hinzufügen

```
class Bicycle {
    int _speed = 0;
    int gear;

    Bicycle(this.gear);

    int get speed => _speed;
}

void main() {
    var myBike = Bicycle(2);
    print myBike;
}
```

- Variable werden mit null initialisiert
- _ markiert eine private Variable

Methode hinzufügen

```
speedUp(increment) {
    _speed += increment;
}
```

- Rückgabetype, Parametertyp kann weggelassen werden

Default Wert verwenden

```
speedUp({increment = 1}) {
    _speed += increment;
}
```

Liste von Bicycles erstellen

```
var bikes = [];
bikes.add(myBike);
var yourBike = Bicycle(3);
bikes.add(yourBike);
print(bikes);
print(bikes.length);
print(bikes[0]);
```

- Lists, Sets und Maps können verwendet werden

