



University
of Basel

Software: Jupyter-Notebooks

Andreas Morel-Forster, Departement Mathematik und Informatik, Universität Basel

Jshell: Interaktives Programmieren

```
C:\Users\luetma00>jshell
| Welcome to JShell -- Version 10.0.1
| For an introduction type: /help intro

jshell> System.out.println("hello world");
hello world

jshell> 
```

- Jshell: Interaktives Programmieren in Java
 - Teil von Java seit Java 9
-

Jshell: Vorteile und Nachteile

Vorteile

- Kein Schreiben von Klassenrumpf und main Methode nötig
- Kompilieren und Ausführen stark vereinfacht
- Direktes Feedback erlaubt interaktives Arbeiten

Gut zum experimentieren.

Nachteile

- Nur textbasiert
- Editieren von etwas längeren Funktionen ist mühsam
- Keine Historie der Befehle

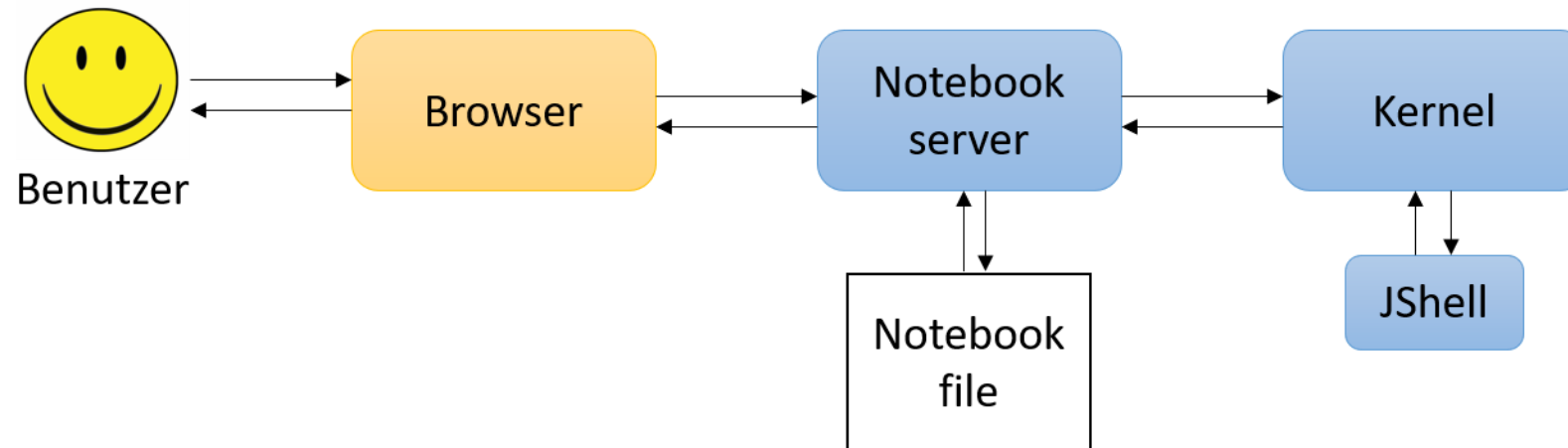
Nicht geeignet um Experimente später nachzuvollziehen.

Was sind Jupyter Notebooks

Webbasierte Programmierumgebung

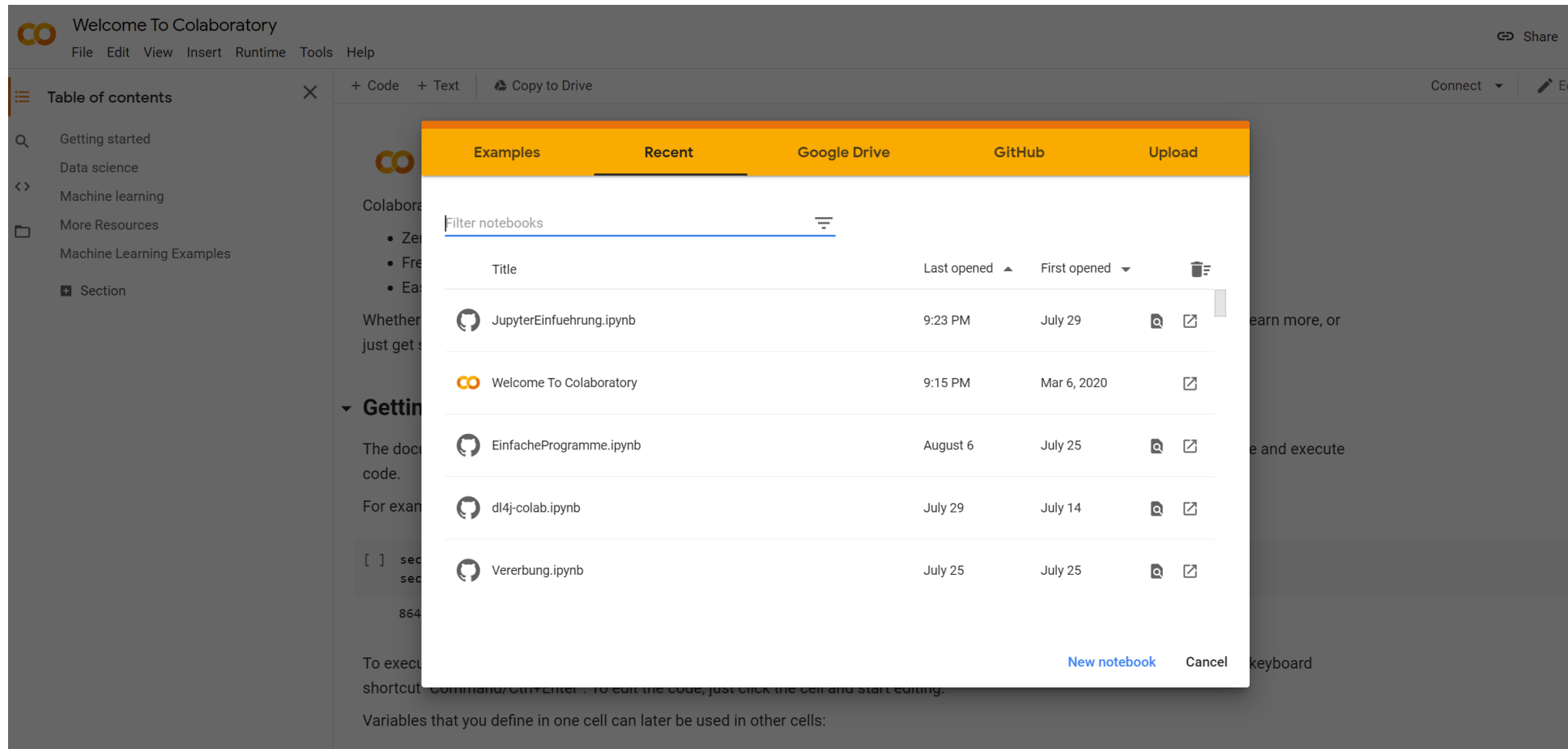
- Idee: Vereinen von Dokumentation und Programme
- Unterstützt viele verschiedene Programmiersprachen.

Java-Kernel: Basierend auf JShell



Demo

<https://colab.research.google.com>



Jupyter Notebooks: Unser Einsatz

- Interaktive Slides
- Mini-Übungen als interaktives Element
- Dokumentation von Experimenten / Programmen zum individuellen Lernen und zur Prüfungsvorbereitung

Zuweisungen und arithmetische Funktionen

Durch das Arbeiten mit Variablen können wir Programme allgemeiner schreiben und von konkreten Beispielen abstrahieren. Als Beispiel nehmen wir die Berechnung der Hypotenuse eines rechtwinkligen Dreiecks. Anstatt die Berechnung direkt mit konkreten Werten durchzuführen, können wir Variablen einführen und die Formel dann allgemein hinschreiben.

```
In [15]: double a = 3;
double b = 4;
```

Nun wissen wir vom Satz von Pythagoras, dass die Länge der Hypotenuse c mit der Formel $c^2 = a^2 + b^2$ berechnet werden kann.

```
In [20]: double aSquared = a * a;
double bSquared = b * b;
double cSquared = aSquared + bSquared;
```

Um c zu erhalten müssen wir noch die Wurzel ziehen. Dafür stellt uns Java die Funktion `Math.sqrt` zur Verfügung.

```
In [21]: double c = Math.sqrt(cSquared);
```

Die Werte der Variablen können wir dies mit dem Befehl `System.out.println(variable)` ausgeben lassen.

```
In [22]: System.out.println(c);
5.0
```

Mini Übung

- Definieren Sie eine Variable mit dem Namen `z` die nur ganze Zahlen enthalten kann und weisen Sie dieser den Wert 3 zu.
- Was passiert, wenn Sie statt 3 den Wert 3.0 zuordnen?
- Wie müssen Sie den Typ ändern, damit die Zuweisung `z = 3.0` funktioniert?

```
In [24]: // Schreiben Sie Ihre Lösungen hier hin. Neue Zeilen können Sie mit Alt + Enter hinzufügen.
```

Verzweigung

Im Programmieren muss man oft zwischen verschiedenen Fällen unterscheiden. Dafür können wir die `if else` Anweisung benutzen. Als erstes Beispiel schauen wir uns ein Programm an, das entscheidet ob die Variable `z` einen positiven oder negativen Wert beinhaltet.

```
In [97]: int z = 5;
```

Jupyter Notebooks: Nachteile

- Entwicklung grösserer Programme nicht möglich
- Keine Unterstützung von professionellen Entwicklungsumgebungen
- Teile von Notebooks können in beliebiger Reihenfolge ausgeführt werden → Verwirrungspotenzial

Super zum lernen, nicht geeignet zur Entwicklung grosser Programme.

Installation / Zugriff auf Jupyter-Notebook



- A. Webbasierte Lösung (colab.research.google.com)
 - + Keine Installation – Läuft auf allen Geräten
 - + Überall verfügbar
 - + Gratis
 - + Speichern im Google-Drive
 - Braucht einen Google Account

- B. Lokale Installation
 - + Speicher / Laden einfach
 - + Alles bleibt auf dem eigenen Rechner
 - Installation etwas aufwändiger

- Link zur Installationsanleitung auf Vorlesungsseite bit.ly/GymInfProg1

Aufgabe: Lernen Sie Jupyter Notebooks kennen.

- Verwenden Sie entweder Google-Colab oder führen Sie die lokale Installation durch
 - Schauen Sie sich das Notebook “*Jupyter Einführung*” / JupyterEinfuehrung.ipynb
 - ❖ Schauen Sie sich das Notebook “*Turtle Referenz*” / Turtle-Referenz.ipynb
-