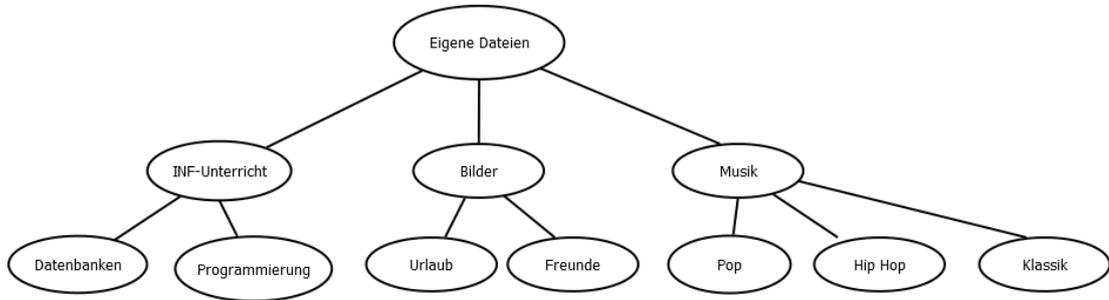


DYNAMISCHE DATENSTRUKTUREN: BÄUME



Eine der wichtigsten dynamischen Datenstrukturen in der Informatik ist der *Baum*. Mit Bäumen kann man Daten nämlich wunderbar hierarchisch ordnen.

Das heißt: Es gibt übergeordnete und untergeordnete Elemente. Ein konkretes Anwendungsbeispiel für eine Datenorganisation in Form eines Baums ist die Dateistruktur eines Computers:



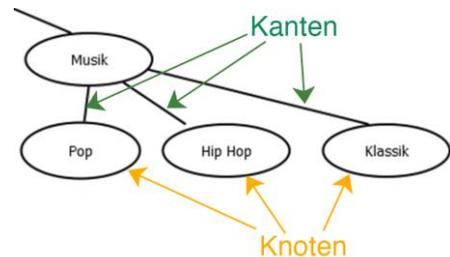
Im Folgenden zitieren wir Auszüge aus der offiziellen Handreichung des Landes Baden-Württemberg. Diese Materialien finden sich hier: tinyurl.com/schule-bw-informatik
An einigen Stellen haben wir diese Materialien angepasst. Im Wesentlichen geben wir die Ausführungen der Handreichung aber wortgetreu wieder.

Definition:

Ein Baum besteht aus **Knoten**, die durch **Kanten** verbunden sind. In dem abgebildeten Beispiel werden z.B. die Knoten **Musik** und **Pop** durch eine Kante verbunden.

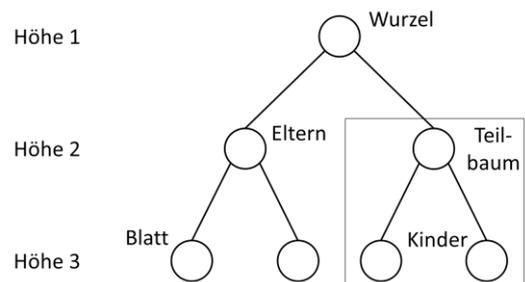


Es handelt sich nur dann um einen Baum, wenn es zwischen zwei beliebig wählbaren Knoten nur einen Weg gibt.



Bestandteile eines Baums

Wurzel	Knoten, der keine Eltern hat (= oberster Knoten im Baum)
Elternknoten	Vorgänger eines bestimmten Knotens
Kind	Nachfolger eines Knotens.
Blatt	Knoten, die keine Kinder haben (= unterste Knoten)
Teilbaum	Knoten und alle seine Kinder
Höhe	Anzahl der Knoten von der Wurzel bis zum Knoten. Auch Wurzel und Blatt werden mitgezählt.



Anwendungsbeispiele von Bäumen

- KO-System bei der Fußball Weltmeisterschaft
- Familien-Stammbaum
- Organigramm in Unternehmen
- Datei-Struktur im Rechner

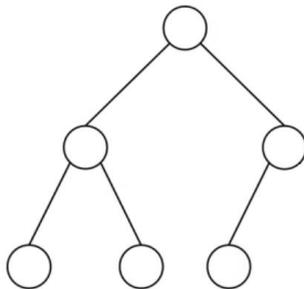
Binärbaum

Ein Binärbaum ist ein Spezialfall der Datenstruktur Baum.

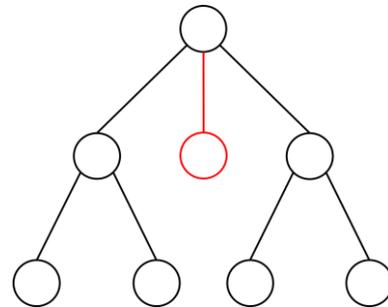


Ein Baum ist ein Binärbaum, wenn alle Knoten maximal zwei Kindknoten haben.

Binärbaum



Kein Binärbaum



GEORDNETER Binärbaum

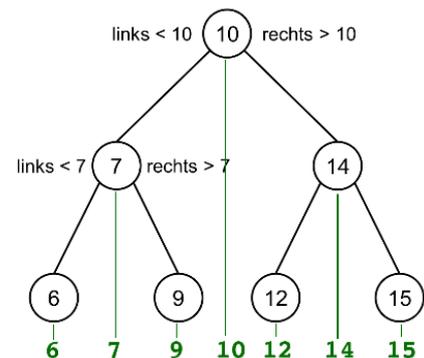
Der linke Teilbaum enthält nur kleinere Knoten als die Wurzel des Teilbaumes.

Der rechte Teilbaum enthält nur größere Knoten als die Wurzel des Teilbaumes.

Außerdem **müssen** alle Eltern-Knoten ein „linkes“ Kind haben und **können** noch ein „rechtes“ Kind haben.



Tipp: Ziehen Sie die Knoten in einer Reihe nach unten und prüfen Sie, ob die Zahlen aufsteigend sortiert sind.

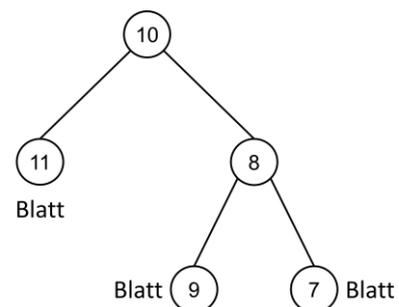


VOLLER Binärbaum

Jeder Knoten ist ein Blatt oder besitzt zwei Kinder.

{11, 9, 7} sind Blätter.

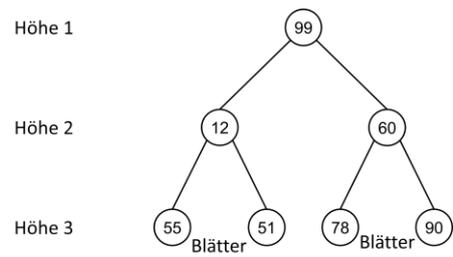
{10, 8} besitzen zwei Kinder



VOLLSTÄNDIGER Binärbaum

Der Baum ist voll und alle Blätter befinden sich auf der gleichen Höhe.

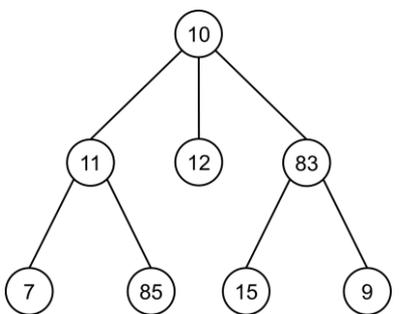
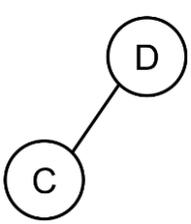
Alle Blätter {55, 51, 78, 90} sind auf Höhe 3.



Aufgaben:



1.	<p>Begründen Sie, ob es sich beim nachfolgenden Schaubild um einen Baum handelt.</p> <pre> graph TD A((A)) --- B((B)) A --- C((C)) B --- D((D)) C --- D </pre>
2.	<p>Markieren Sie einen Teilbaum im nachfolgenden Baum.</p> <pre> graph TD 10((10)) --- 11((11)) 10 --- 8((8)) 8 --- 9((9)) 8 --- 7((7)) </pre>
3.	<p>Beurteilen Sie, ob der nachfolgende Baum vollständig, voll und/oder geordnet ist.</p> <pre> graph TD 10((10)) --- 5((5)) 10 --- 20((20)) 5 --- 7((7)) 20 --- 19((19)) 20 --- 21((21)) </pre> <p>GEORDNET:</p> <p>VOLL:</p> <p>VOLLSTÄNDIG:</p>
4.	<p>Überführen Sie die Knoten {25, 50, 60, 75, 80} in einen vollen, geordneten Binärbaum.</p>

5.	<p>Überführen Sie den folgenden Baum in einen Binärbaum.</p>  <pre> graph TD 10((10)) --- 11((11)) 10 --- 12((12)) 10 --- 83((83)) 11 --- 7((7)) 11 --- 85((85)) 83 --- 15((15)) 83 --- 9((9)) </pre>
6.	<p>Fügen Sie nacheinander die Knoten {B, A} in den gegebenen Binärbaum ein. Achten Sie darauf, dass der Binärbaum weiterhin geordnet ist.</p>  <pre> graph TD D((D)) --- C((C)) </pre>

(In den Materialien der oben aufgeführten Quellen finden sich noch weitere Aufgaben und Lösungen!)