

**Saarbrücken, 28.09.2013**

(Einige Folien werden z.B. im Adobe Reader animiert dargestellt.)





# Some Statistics

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

# Some Statistics

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

1990-2000

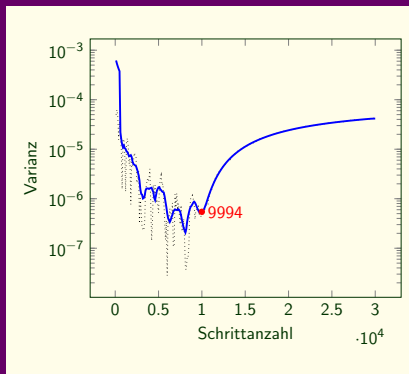
1990-2000

1990-2000

1990-2000

1990-2000

# Some Statistics



# Phase 1

## Einfachheit

- ▶ Anfangszustand: Komplette einfarbige Ebene
- ▶ Bildung einfacher, meist symmetrischer Muster
- ▶ Gedanke: „Einfache Regel → Einfache Muster“
- ▶ Dauer: 1 - ca. 300

## Phase 2

### Chaos

- ▶ Erzeugung eines großen unregelmäßigen Flickenteppiches
- ▶ Gedanke: „Ameise vollführt zufällige Bewegung.“
- ▶ Dauer: 300 - ca. 10000



# Phase 3

## Hervortretende Ordnung

- ▶ Erzeugung einer Straße in südöstlicher Richtung.
- ▶ Durchlauf eines Zyklus' von 104 Schritten.
- ▶ Gedanke: „Ameise bewegt sich gemäß einer bestimmten Regel“
- ▶ Dauer: ab ca. 10000

## Technischer Hintergrund

- ▶ Vollständig realisiert mit **Lua<sup>A</sup>T<sub>E</sub>X**
- ▶ Klasse beamer - Erstellung der Folien
- ▶ Paket animate - Erstellung von Animationen
- ▶ Paket pgfplots - Erstellung des Funktionsplots
- ▶ Paket **tikz** - Erstellung der Kacheln und Balken
- ▶ **Lua** - Generieren des Ameisenmodells

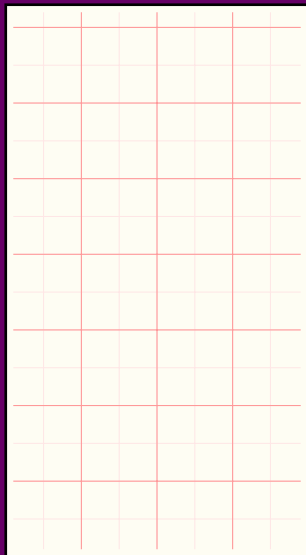
## Arbeiten mit TikZ

- ▶ TikZ: „TikZ ist kein Zeichenprogramm“
- ▶ Entwickelt von Till Tantau (Universität Lübeck)
- ▶ Idee: Erstellung von Grafiken mit Hilfe von Anweisungen
- ▶ z.B. <http://www.texample.net/tikz/examples/>

# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

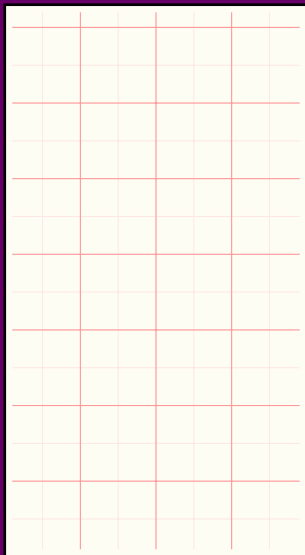


# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

```
\draw[color=blue] (0.5,-1)--(53:2.5);
```

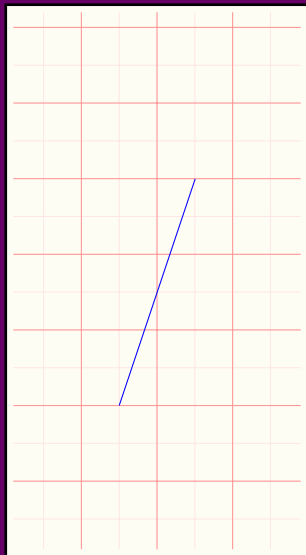


# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

```
\draw[color=blue] (0.5,-1)--(53:2.5);
```



# Arbeiten mit TikZ

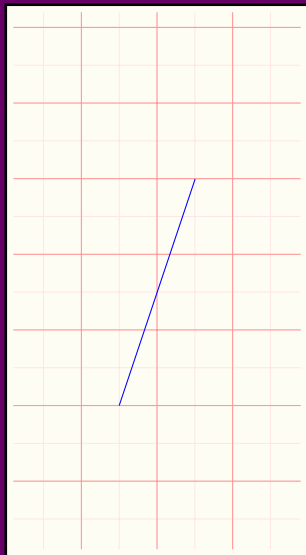
## Grundlegender Aufbau

Anweisung:

```
\draw[color=blue] (0.5,-1)--(53:2.5);
```

► **Befehle**

`\draw`, `\fill`, `\clip`, `\node`, ...



# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

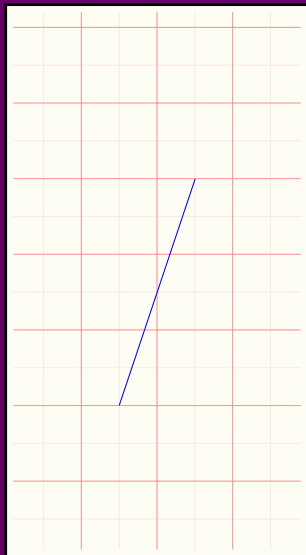
```
\draw[color=blue] (0.5,-1)--(53:2.5);
```

- ▶ **Befehle**

`\draw`, `\fill`, `\clip`, `\node`, ...

- ▶ **Attribute**

`color`, line width, opacity, ...





# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

```
\draw[color=blue] (0.5,-1)--(53:2.5);
```

- ▶ **Befehle**

\draw, \fill, \clip, \node, ...

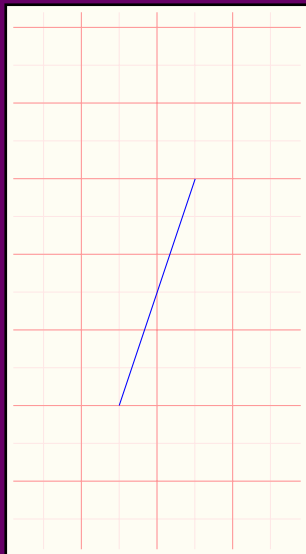
- ▶ **Attribute**

color, line width, opacity, ...

- ▶ **Koordinaten**

absolut:  $(x,y)$ , auch polar  $(\alpha:r)$

relativ:  $++(\Delta x,\Delta y)$



# Arbeiten mit TikZ

## Grundlegender Aufbau

Anweisung:

```
\draw[color=blue] (0.5,-1)--(53:2.5);
```

- ▶ **Befehle**

`\draw`, `\fill`, `\clip`, `\node`, ...

- ▶ **Attribute**

color, line width, opacity, ...

- ▶ **Koordinaten**

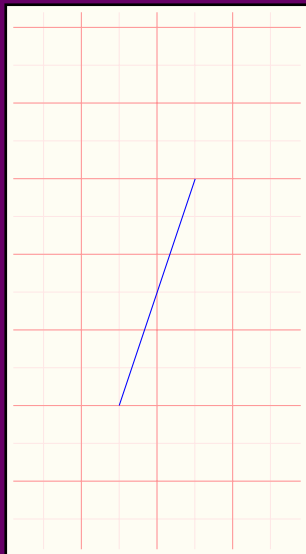
absolut:  $(x,y)$ , auch polar  $(\alpha:r)$

relativ:  $++(\Delta x,\Delta y)$

- ▶ **Sonderzeichen**

`--` : Verbinder

`;` : Abschluss



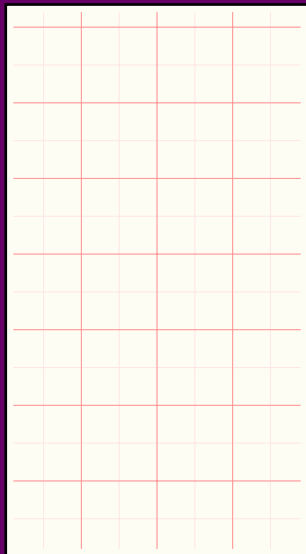
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw (-0.5,-2.5) -- (2,0) -- (0.5,2)  
-- cycle;
```

- ▶ Dreieck in absoluten Koordinaten



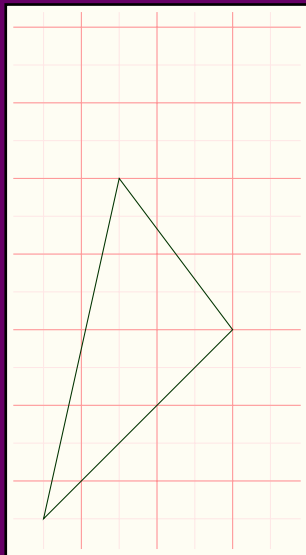
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw (-0.5,-2.5) -- (2,0) -- (0.5,2)  
-- cycle;
```

- ▶ Dreieck in absoluten Koordinaten



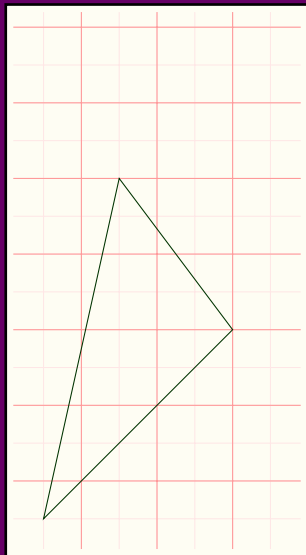
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw (-0.5,-2.5) -- ++(2.5,2.5) --  
++(-1.5,2) -- cycle;
```

- ▶ Dreieck in absoluten Koordinaten
- ▶ Dreieck in relativen Koordinaten



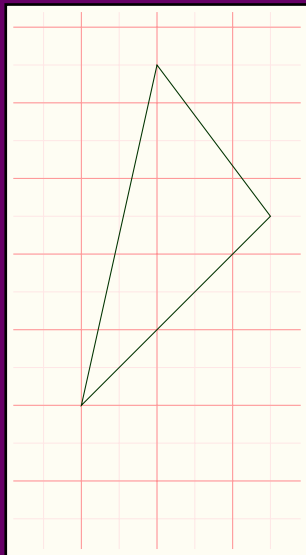
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw (0,-1) -- ++(2.5,2.5) --  
++(-1.5,2) -- cycle;
```

- ▶ Dreieck in absoluten Koordinaten
- ▶ Dreieck in relativen Koordinaten
- ▶ Verschobenes Dreieck



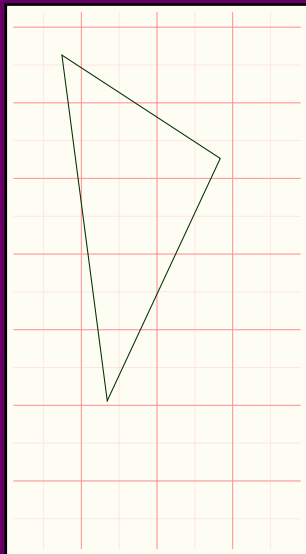
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw[rotate=20] (0,-1) --  
++(2.5,2.5) -- ++(-1.5,2) -- cycle;
```

- ▶ Dreieck in absoluten Koordinaten
- ▶ Dreieck in relativen Koordinaten
- ▶ Verschobenes Dreieck
- ▶ (Um Ursprung) Gedrehtes Dreieck



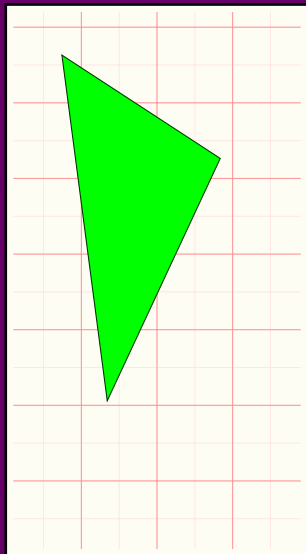
# Arbeiten mit TikZ

## Dreiecke malen

Anweisung:

```
\draw[rotate=20,fill=green] (0,-1) --  
++(2.5,2.5) -- ++(-1.5,2) -- cycle;
```

- ▶ Dreieck in absoluten Koordinaten
- ▶ Dreieck in relativen Koordinaten
- ▶ Verschobenes Dreieck
- ▶ (Um Ursprung) Gedrehtes Dreieck
- ▶ Mit grüner Farbe gefüllt

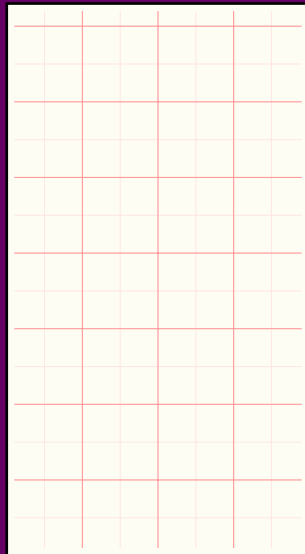




# Arbeiten mit TikZ

## Punkte malen

Anweisung:

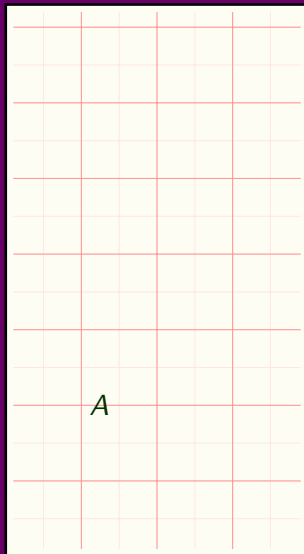


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);
```

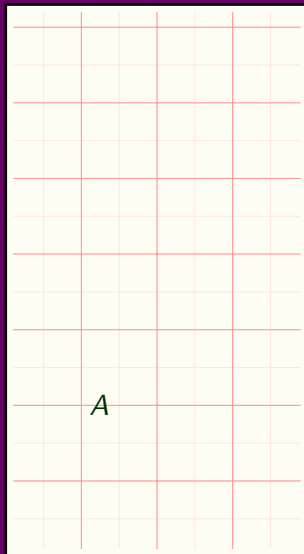


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);
```

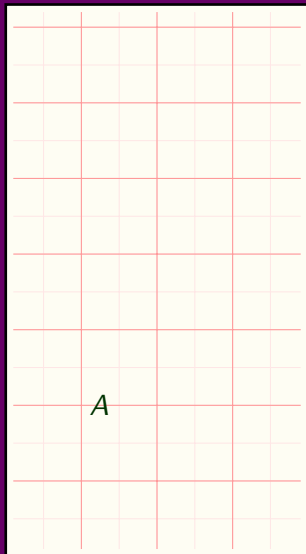


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);
```

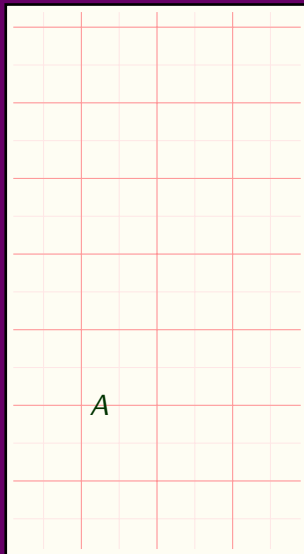


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);
```

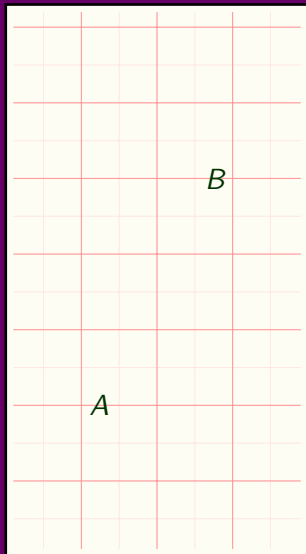


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);  
\coordinate[label=right:$B$] (B) at  
(1.5,2);
```



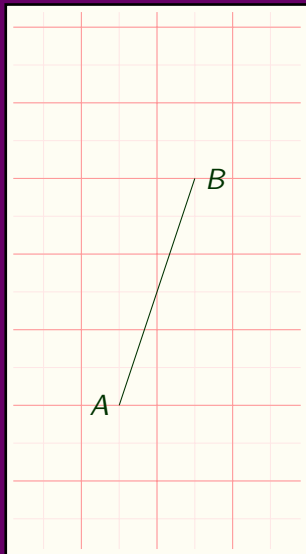
# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);  
\coordinate[label=right:$B$] (B) at  
(1.5,2);
```

```
\draw (A) -- (B);
```



# Arbeiten mit TikZ

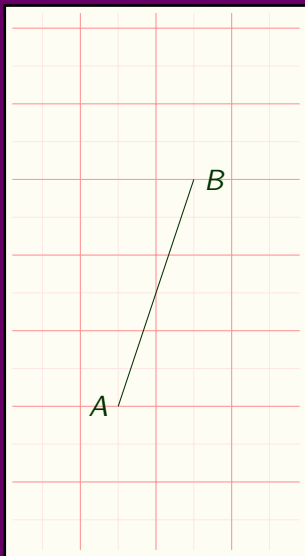
## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at
(0.5,-1);
\coordinate[label=right:$B$] (B) at
(1.5,2);

\def\Punkt{(0,0) circle (0.7mm)}

\draw (A) -- (B);
```









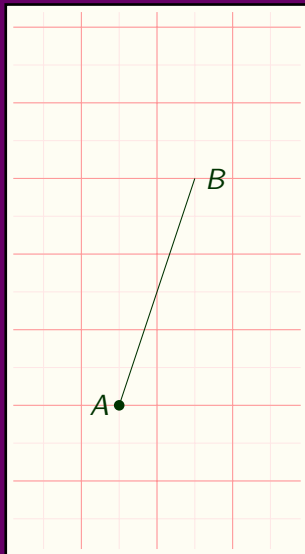


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);  
\coordinate[label=right:$B$] (B) at  
(1.5,2);  
  
\def\Punkt{(0,0) circle (0.7mm)}  
  
\fill[shift={A}] \Punkt;  
  
\draw (A) -- (B);
```

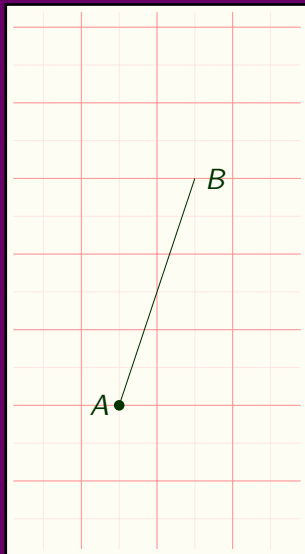


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);  
\coordinate[label=right:$B$] (B) at  
(1.5,2);  
  
\def\Punkt{(0,0) circle (0.7mm)}  
\def\KreuzPunkt{(0,0)++(-.125,-.125)--  
++(.25,.25)++(0,-0.25)-- ++(-.25,.25)}  
  
\fill[shift={A}] \Punkt;  
  
\draw (A) -- (B);
```

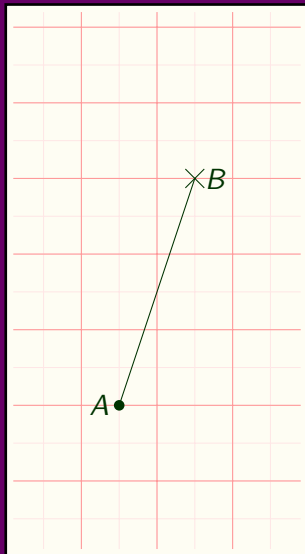


# Arbeiten mit TikZ

## Punkte malen

Anweisung:

```
\coordinate[label=left:$A$] (A) at  
(0.5,-1);  
\coordinate[label=right:$B$] (B) at  
(1.5,2);  
  
\def\Punkt{(0,0) circle (0.7mm)}  
\def\KreuzPunkt{(0,0)++(-.125,-.125)--  
++(.25,.25)++(0,-0.25)-- ++(-.25,.25)}  
  
\fill[shift={A}] \Punkt;  
\draw[shift={B}] \KreuzPunkt;  
  
\draw (A) -- (B);
```



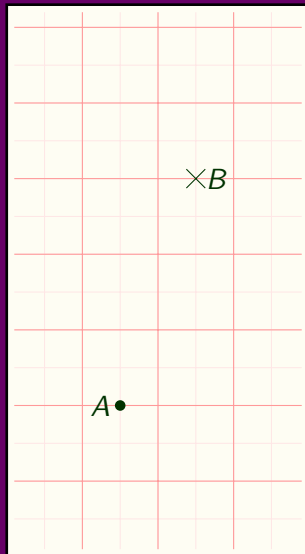
# Arbeiten mit TikZ

## Geraden malen

In die Präamble: `\usetikzlibrary{calc}`

`\begin{scope}`

`\end{scope}`



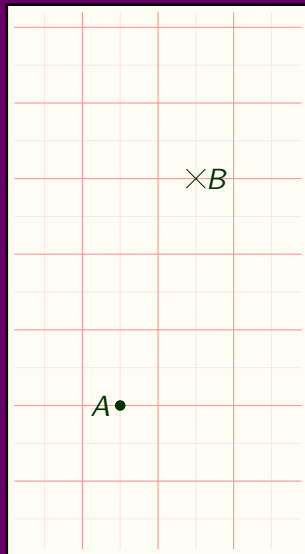
# Arbeiten mit TikZ

## Geraden malen

In die Präamble: `\usetikzlibrary{calc}`

`\begin{scope}`

`\end{scope}`



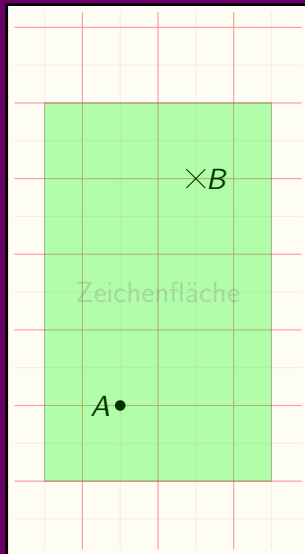


# Arbeiten mit TikZ

## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};
```

```
\end{scope}
```



# Arbeiten mit TikZ

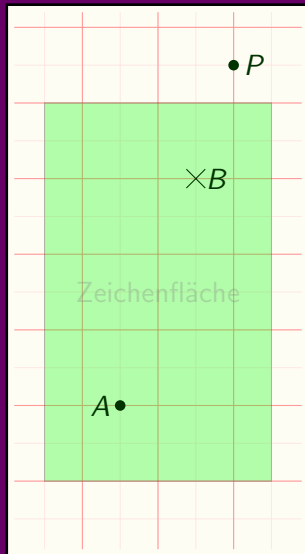
## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};

\coordinate[label=right:$P$] (P) at
($ (A)!+1.5!(B) $);

\fill[shift={ (P) }] \Punkt;

\end{scope}
```



# Arbeiten mit TikZ

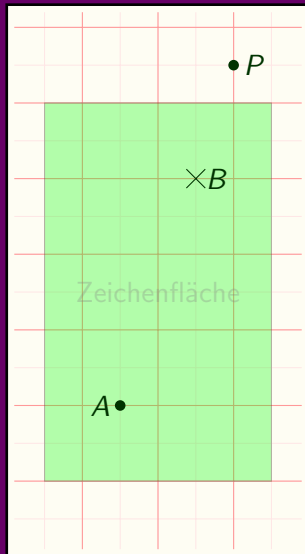
## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};

\coordinate[label=right:$P$] (P) at
($(A)!+1.5!(B)$);

\fill[shift={ (P) }] \Punkt;

\end{scope}
```



# Arbeiten mit TikZ

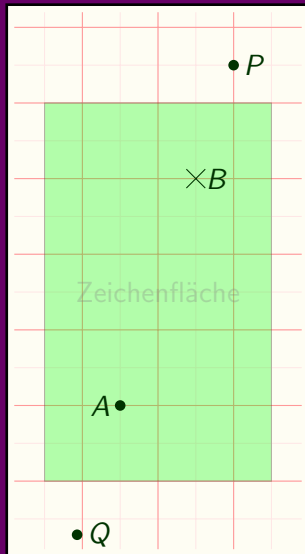
## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};

\coordinate[label=right:$P$] (P) at
($(A)!+1.5!(B)$);
\coordinate[label=right:$Q$] (Q) at
($(A)!-1.8cm!(B)$);

\fill[shift={ (P) }] \Punkt;
\fill[shift={ (Q) }] \Punkt;

\end{scope}
```



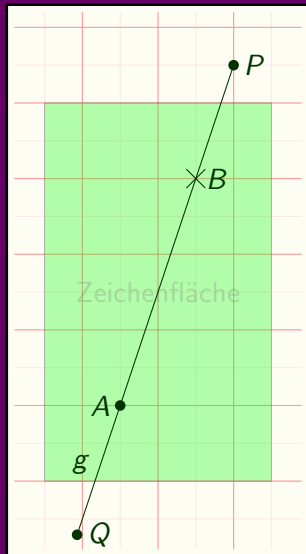
# Arbeiten mit TikZ

## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};

\coordinate[label=right:$P$] (P) at
($(A)!+1.5!(B)$);
\coordinate[label=right:$Q$] (Q) at
($(A)!-1.8cm!(B)$);

\fill[shift={ (P) }] \Punkt;
\fill[shift={ (Q) }] \Punkt;
\draw (P)--(Q) node[left,pos=.85] {$g$};
\end{scope}
```



# Arbeiten mit TikZ

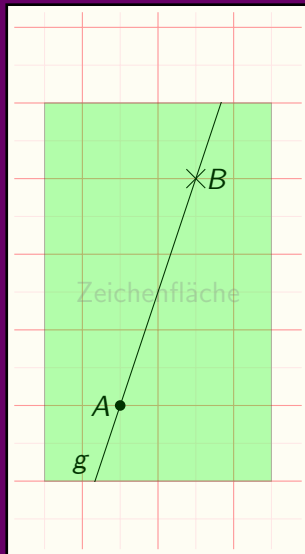
## Geraden malen

```
In die Präamble: \usetikzlibrary{calc}
\begin{scope}
\draw[fill=green,opacity=0.3] (-0.5,-2)
rectangle (2.5,3) node[pos=0.5]
{\color{gray}Zeichenfläche};
\clip (-0.5,-2) rectangle (2.5,3);

\coordinate[label=right:$P$] (P) at
($(A)!+1.5!(B)$);
\coordinate[label=right:$Q$] (Q) at
($(A)!-1.8cm!(B)$);

\fill[shift={ (P) }] \Punkt;
\fill[shift={ (Q) }] \Punkt;
\draw (P)--(Q) node[left,pos=.85] {$g$};

\end{scope}
```



# Arbeiten mit TikZ

## Vorteile

- ▶ Zeichnen geometrischer Objekte (Punkt, Strecke, Gerade, etc.)
- ▶ Sieht schick aus :-)

# Arbeiten mit TikZ

## Vorteile

- ▶ Zeichnen geometrischer Objekte (Punkt, Strecke, Gerade, etc.)
- ▶ Sieht schick aus :-)

## Nachteile

- ▶ Aufbau und Syntax wird schnell komplex
- ▶ Nicht gezeigt: Bestimmung von Schnittpunkten möglich, jedoch aufgrund der Berechnung mit Hilfe des  $\text{T}_\text{E}\text{X}$ -Kernels eher langsam



## Arbeiten mit Lua

- ▶ Lua: Portugisisch für Mond
- ▶ Entwickelt von Roberto Ierusalimschy u.a. (1993)
- ▶ Einfache und schnelle Skriptsprache
- ▶ Paradigmen: imperativ, funktional, objektorientiert
- ▶ Literatur: *Programmieren in Lua*. Open Source Press, 3. Auflage. (2013)

# Arbeiten mit Lua

## Lua-Konsole

### Variablen in Lua

# Arbeiten mit Lua

## Lua-Konsole

### Variablen in Lua

- ▶ **Datentyp:**  
Zahlen, Zeichenketten, Wahrheitswerte

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
```

## Variablen in Lua

- ▶ **Datentyp:**  
Zahlen, Zeichenketten, Wahrheitswerte

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
```

```
> b = "Elbe"
```

## Variablen in Lua

- ▶ **Datentyp:**

Zahlen, **Zeichenketten**, Wahrheitswerte

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743  
> b = "Elbe"  
> c = true
```

## Variablen in Lua

- ▶ **Datentyp:**  
Zahlen, Zeichenketten, **Wahrheitswerte**

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743  
> b = "Elbe"  
> c = true  
> print(a,b,c)
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ **Ausgabe:**  
print()

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ **Ausgabe:**  
print()



# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ **Besonderer Datentyp:**  
nil

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ **Besonderer Datentyp:**  
nil

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ **Besonderer Datentyp:**  
nil

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ Besonderer Datentyp:  
nil
- ▶ **Zuweisung:**  
mehrfach möglich

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
> a,b = b,a
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ Besonderer Datentyp:  
nil
- ▶ **Zuweisung:**  
mehrfach möglich

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
> a,b = b,a
> print(a,b)
Elbe 3.5743
```

## Variablen in Lua

- ▶ **Datentyp:**  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ **Ausgabe:**  
print()
- ▶ **Besonderer Datentyp:**  
nil
- ▶ **Zuweisung:**  
mehrfach möglich

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
> a,b = b,a
> print(a,b)
Elbe 3.5743
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ Besonderer Datentyp:  
nil
- ▶ Zuweisung:  
mehrfach möglich
- ▶ **Operationen und Relationen:**  
+, -, \*, /, >, <, etc.

# Arbeiten mit Lua

## Lua-Konsole

```
> a = 3.5743
> b = "Elbe"
> c = true
> print(a,b,c)
3.5743 Elbe true
> print(d)
nil
> a,b = b,a
> print(a,b)
Elbe 3.5743
> print(a .. b)
Elbe3.5743
```

## Variablen in Lua

- ▶ Datentyp:  
Zahlen, Zeichenketten, Wahrheitswerte
- ▶ Ausgabe:  
print()
- ▶ Besonderer Datentyp:  
nil
- ▶ Zuweisung:  
mehrfach möglich
- ▶ **Operationen und Relationen:**  
+, -, \*, /, >, <, etc.



# Arbeiten mit Lua

## Lua-Konsole

```
> if <Bedingung> then  
    <>true-Anweisungen>  
else  
    <>false-Anweisungen>  
end
```

## Kontrollstrukturen und Funktionen

- ▶ **Selektion**

# Arbeiten mit Lua

## Lua-Konsole

```
> if d then  
    print(d)  
else  
    print("d nicht definiert!")  
end
```

## Kontrollstrukturen und Funktionen

- ▶ **Selektion**

# Arbeiten mit Lua

## Lua-Konsole

```
> if d then
  print(d)
else
  print("d nicht definiert!")
end
d nicht definiert!
```

## Kontrollstrukturen und Funktionen

- ▶ **Selektion**

# Arbeiten mit Lua

## Lua-Konsole

```
> if d then
    print(d)
else
    print("d nicht definiert!")
end
d nicht definiert!
> l = d or 17
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion

# Arbeiten mit Lua

## Lua-Konsole

```
> if d then
  print(d)
else
  print("d nicht definiert!")
end
d nicht definiert!
> l = d or 17
> print(l)
17
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion

# Arbeiten mit Lua

## Lua-Konsole

```
> repeat  
  <Anweisungen>  
until <Abbruchbedingung>
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ **Wiederholungen**

# Arbeiten mit Lua

## Lua-Konsole

> **repeat**

    <Anweisungen>

**until** <Abbruchbedingung>

> **while** <Bedingung> **do**

    <Anweisungen>

**end**

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ **Wiederholungen**

# Arbeiten mit Lua

## Lua-Konsole

```
> function <Name>(<Parameter>)  
  <Anweisungen>  
  return <Rückgabeparameter>  
end
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ Wiederholungen
- ▶ **Funktionen** als Variablen



# Arbeiten mit Lua

## Lua-Konsole

```
> function f(x)
  z=2*x+5
  return z
end
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ Wiederholungen
- ▶ **Funktionen** als Variablen

# Arbeiten mit Lua

## Lua-Konsole

```
> function f(x)
  z=2*x+5
  return z
end
> f(4)
13
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ Wiederholungen
- ▶ **Funktionen** als Variablen

# Arbeiten mit Lua

## Lua-Konsole

```
> function f(x)
```

```
  z=2*x+5
```

```
  return z
```

```
end
```

```
> f(4)
```

```
13
```

```
> f = function (x)
```

```
  return 2*x+5
```

```
end
```

## Kontrollstrukturen und Funktionen

- ▶ Selektion
- ▶ Wiederholungen
- ▶ **Funktionen** als Variablen

# Arbeiten mit Lua

## Lua-Konsole

### Arbeit mit Tabellen

# Arbeiten mit Lua

## Lua-Konsole

### Arbeit mit Tabellen

- ▶ **Tabelle als Feld**

# Arbeiten mit Lua

## Lua-Konsole

```
> a = {3,7.42,22,11.3}
```

## Arbeit mit Tabellen

- ▶ **Tabelle als Feld**

# Arbeiten mit Lua

## Lua-Konsole

```
> a = {3,7.42,22,11.3}
```

```
> print(a[2])
```

```
7.42
```

## Arbeit mit Tabellen

- ▶ **Tabelle als Feld**

# Arbeiten mit Lua

## Lua-Konsole

```
> a = {3,7.42,22,11.3}
```

```
> print(a[2])
```

```
7.42
```

```
> a[2]=5
```

## Arbeit mit Tabellen

- ▶ **Tabelle als Feld**



# Arbeiten mit Lua

## Lua-Konsole

```
> a = {3,7.42,22,11.3}  
> print(a[2])  
7.42  
> a[2]=5  
> a[#a+1]=4
```

## Arbeit mit Tabellen

- ▶ **Tabelle als Feld**

# Arbeiten mit Lua

## Lua-Konsole

### Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ **Tabelle als Datensatz**

# Arbeiten mit Lua

## Lua-Konsole

```
> Termin={Tag="Montag"}
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ **Tabelle als Datensatz**

# Arbeiten mit Lua

## Lua-Konsole

```
> Termin={Tag="Montag"}
```

```
> print(Termin["Tag"])
```

```
Montag
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ **Tabelle als Datensatz**

# Arbeiten mit Lua

## Lua-Konsole

```
> Termin={Tag="Montag"}
```

```
> print(Termin["Tag"])
```

Montag

```
> Termin["Ort"]="Konferenzraum"
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ **Tabelle als Datensatz**

# Arbeiten mit Lua

## Lua-Konsole

```
> Termin={Tag="Montag"}
```

```
> print(Termin["Tag"])
```

Montag

```
> Termin["Ort"]="Konferenzraum"
```

```
> Termin.Zeit="13:30"
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ **Tabelle als Datensatz**

# Arbeiten mit Lua

## Lua-Konsole

### Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ Tabelle als Datensatz
- ▶ **Tabelle als Objekt**

# Arbeiten mit Lua

## Lua-Konsole

```
>function Termin:Ausgabe()  
  print("Tag: " .. self.Tag)  
  print("Zeit: " .. self.Zeit .. " Uhr")  
  print("Ort: " .. self.Ort)  
end
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ Tabelle als Datensatz
- ▶ **Tabelle als Objekt**



# Arbeiten mit Lua

## Lua-Konsole

```
>function Termin:Ausgabe()  
  print("Tag: " .. self.Tag)  
  print("Zeit: " .. self.Zeit .. " Uhr")  
  print("Ort: " .. self.Ort)  
end
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ Tabelle als Datensatz
- ▶ **Tabelle als Objekt**

# Arbeiten mit Lua

## Lua-Konsole

```
>function Termin:Ausgabe()  
  print("Tag: " .. self.Tag)  
  print("Zeit: " .. self.Zeit .. " Uhr")  
  print("Ort: " .. self.Ort)  
end  
>  
> Termin:Ausgabe()  
Tag: Montag  
Zeit: 13:30 Uhr  
Ort: Konferenzraum
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ Tabelle als Datensatz
- ▶ **Tabelle als Objekt**

# Arbeiten mit Lua

## Lua-Konsole

```
>function Termin:Ausgabe()  
  print("Tag: " .. self.Tag)  
  print("Zeit: " .. self.Zeit .. " Uhr")  
  print("Ort: " .. self.Ort)  
end  
> Termin.Ort="Raum 26h"  
> Termin:Ausgabe()  
Tag: Montag  
Zeit: 13:30 Uhr  
Ort: Raum 26h
```

## Arbeit mit Tabellen

- ▶ Tabelle als Feld
- ▶ Tabelle als Datensatz
- ▶ **Tabelle als Objekt**

# Arbeiten mit Lua

## Vorteile

- ▶ Objektorientierte Programmierung möglich
- ▶ „1000x schneller bei Berechnungen als TikZ“  
(C. Meigen, Dante2012)

## TikZ und Lua

- ▶ Lua $\LaTeX$ : „Wenn der Löwe den Mond anbetet“ (Patrick Gundlach, Dante 2010)
- ▶ Einbettung von Lua in  $\TeX$
- ▶ Motivation: Generalüberholung des 8Bit $\TeX$ -Systems
- ▶ Features: Unicode, direkter Eingriff ins Satzsystem, direkter Lua-Zugriff
- ▶ Herbert Voß: *Einführung in Lua $\TeX$  und Lua $\LaTeX$* . lehmanns media (2013)

# TikZ und Lua

Idee: Geometrische Objekte

**L**A<sub>T</sub>E<sub>X</sub>-Dokument

```
\begin{luacode*}
```

```
\end{luacode*}
```

# TikZ und Lua

Idee: Geometrische Objekte

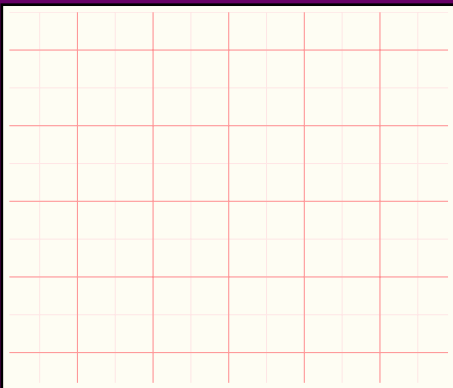
**L<sup>A</sup>T<sub>E</sub>X**-Dokument

```
\begin{luacode*}  
require("lua/geometrie.lua")
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}  
require("lua/geometrie.lua")  
geobib:startBild()
```

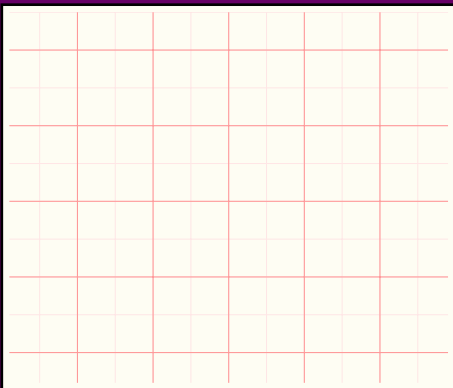
```
geobib:stopBild()  
\end{luacode*}
```



# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

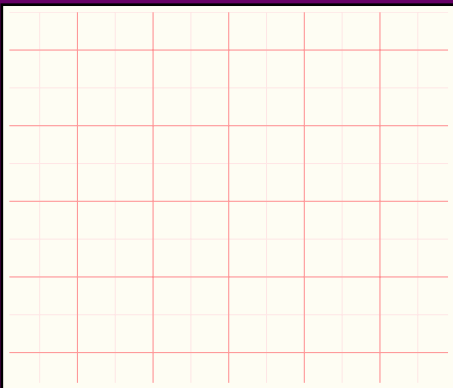
```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
```

```
geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

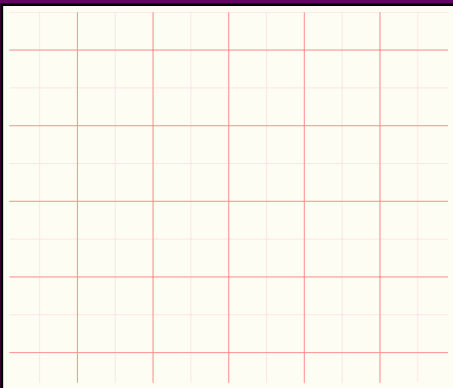
```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
```

```
geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

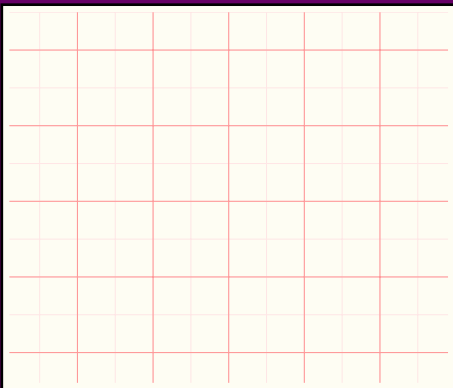
```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
```

```
geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)

geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen



A

## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()

geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen

A

B

## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()
B:MalMich()

geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen

A

C

B

## L<sup>A</sup>T<sub>E</sub>X-Dokument

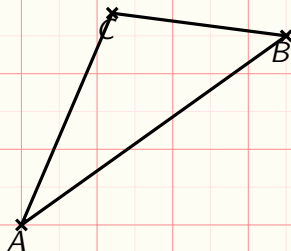
```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()
B:MalMich()
C:MalMich()

geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()
B:MalMich()
C:MalMich()
P:MalMich()

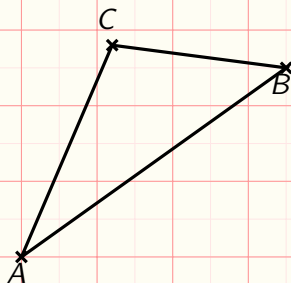
geobib:stopBild()
\end{luacode*}
```



# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

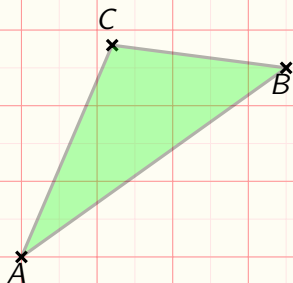
```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()
B:MalMich()
C.Position=[[above]]
C:MalMich()
P:MalMich()

geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen



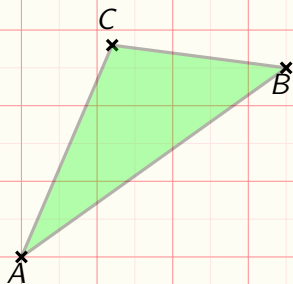
## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
require("lua/geometrie.lua")
geobib:startBild()
A = Punkt(-2,-1)
B = Punkt(1.5,1.5)
C = Punkt(-0.8,1.8)
P = Polygon(A,B,C)
A:MalMich()
B:MalMich()
C.Position=[[above]]
C:MalMich()
P.Fuellfarbe=[[green]]
P:MalMich()
geobib:stopBild()
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
```

```
...
```

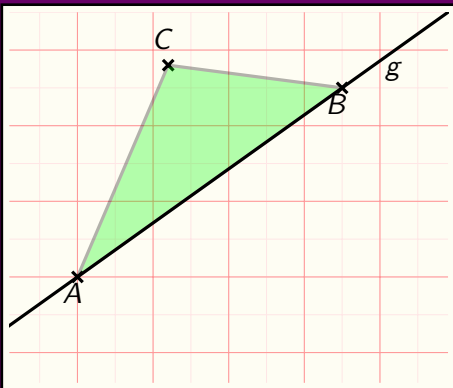
```
g = Gerade(A,B)
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
```

```
...
```

```
g = Gerade(A,B)
```

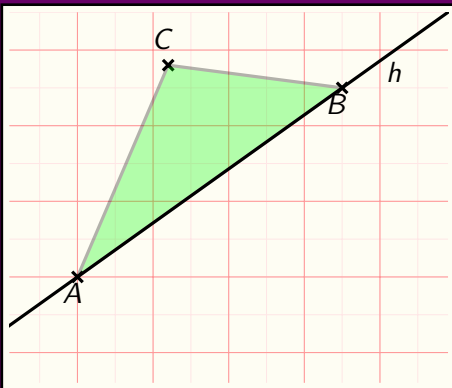
```
g:MalMich()
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
```

```
...
```

```
g = Gerade(A,B)
```

```
g.Bezeichnung=[[h]]
```

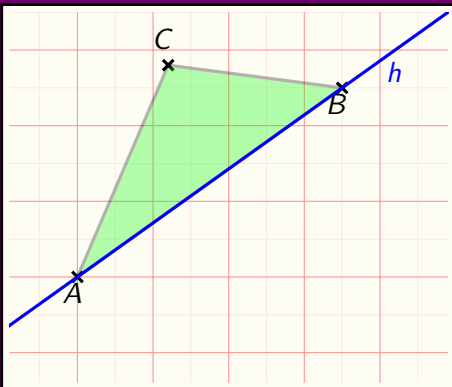
```
g:MalMich()
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

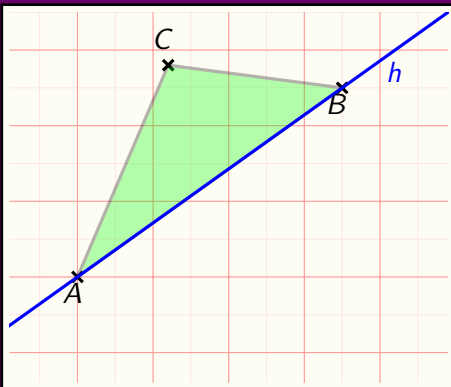
```
\begin{luacode*}
  ...
  g = Gerade(A,B)
  g.Bezeichnung=[[h]]
  g.Farbe=[[blue]]
  g:MalMich()
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

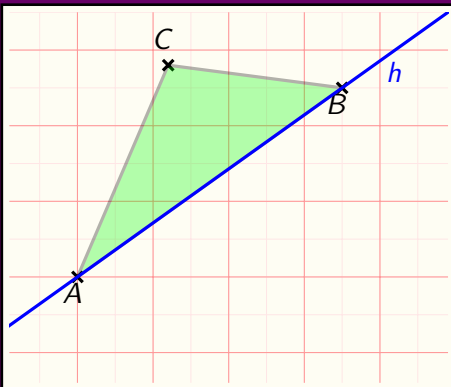
```
\begin{luacode*}
    ...
    g = Gerade(A,B)
    g.Bezeichnung=[[h]]
    g.Farbe=[[blue]]
    g:MalMich()
    sigma = Achsenspiegelung(g)
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
    ...
    g = Gerade(A,B)
    g.Bezeichnung=[[h]]
    g.Farbe=[[blue]]
    g:MalMich()
    sigma = Achsenspiegelung(g)
    Ps = sigma:Bewegen(P)

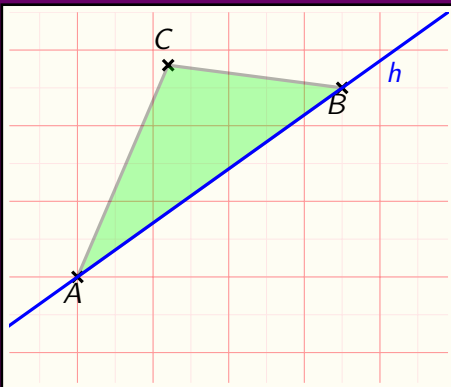
\end{luacode*}
```



# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

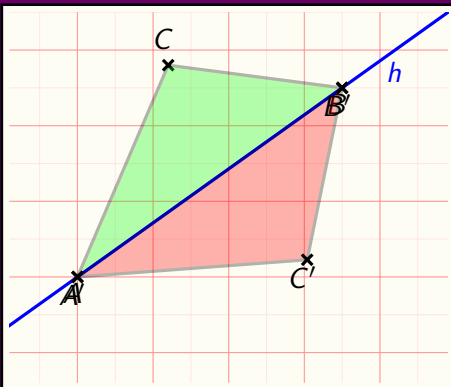
```
\begin{luacode*}
    ...
g = Gerade(A,B)
g.Bezeichnung=[[h]]
g.Farbe=[[blue]]
g:MalMich()
sigma = Achsenspiegelung(g)
Ps = sigma:Bewegen(P)
Ps.Fuellfarbe=[[red]]

\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

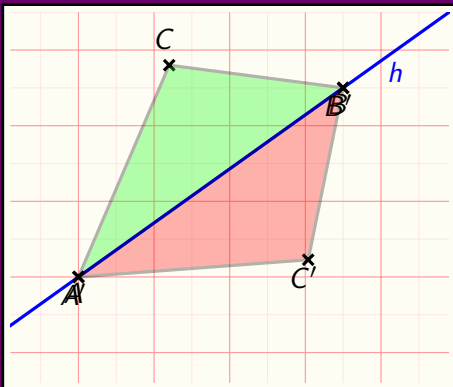
```
\begin{luacode*}
    ...
    g = Gerade(A,B)
    g.Bezeichnung=[[h]]
    g.Farbe=[[blue]]
    g:MalMich()
    sigma = Achsenspiegelung(g)
    Ps = sigma:Bewegen(P)
    Ps.Fuellfarbe=[[red]]
    Ps:MalMich()

\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
```

```
...
```

```
A = Punkt(-2,-1)
```

```
B = Punkt(1.5,1.5)
```

```
C = Punkt(-0.8,1.8)
```

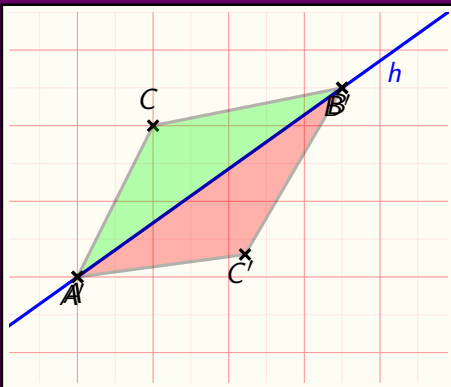
```
...
```

```
\end{luacode*}
```

# TikZ und Lua

## Idee: Geometrische Objekte

- ▶ Mit Lua erzeugen
- ▶ Mit tikz malen
- ▶ In Bewegung versetzen



## L<sup>A</sup>T<sub>E</sub>X-Dokument

```
\begin{luacode*}
```

```
...
```

```
A = Punkt(-2,-1)
```

```
B = Punkt(1.5,1.5)
```

```
C = Punkt(-1,1)
```

```
...
```

```
\end{luacode*}
```

# TikZ und Lua

## Vorteile

Mit TikZ und Lua können die Vorteile von TikZ bei der Erstellung geometrischer Zeichnungen genutzt werden, ohne die Nachteile (wie eine komplexe Syntax oder lange Rechenzeiten) von TikZ in Kauf zu nehmen.

## PS:

Die von mir in Lua entwickelte Bibliothek `geometrie.lua` steht frei zur Verfügung und kann (zusammen mit weiteren Beispielen und einer unter Windows leicht installierbaren Version von TeXLive) über den in der [Zusammenfassung](#) stehenden Link herunter geladen werden.

## Arbeitsblatt Geometrie

- ▶ Automatische Generierung von Aufgabenblättern
- ▶ Automatische Generierung von Musterlösungen

# Arbeitsblatt Geometrie

## Voraussetzung

Gegeben sei ...

## Zeichenfläche

Hier kann die Schülerin mit Zirkel und Lineal arbeiten.

## Aufgaben

(a) Konstruiere ...

# Arbeitsblatt Geometrie

**Voraussetzung**  
Gegeben sei

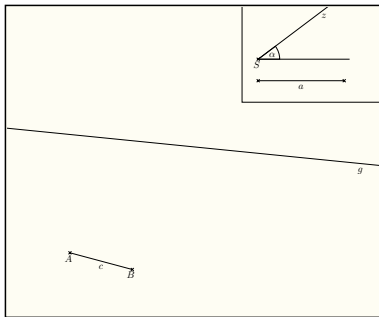
**Zeichenfläche**  
Hier kann  
Zirkel und Lineal

**Aufgaben**  
(a) Konstruiere

## Konstruktionen Dreieck

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3 \text{ cm}$ ,  $a = 4 \text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = \|\angle(C, A, B)\| = 37^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.



# Arbeitsblatt Geometrie

**Voraussetzung**  
Gegeben sei

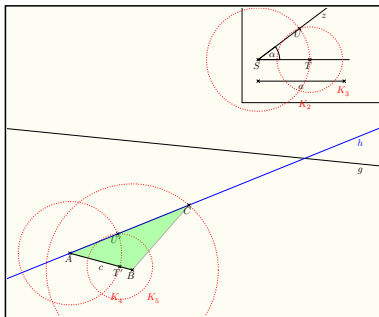
**Zeichenfläche**  
Hier kann  
Zirkel und Li

**Aufgaben**  
(a) Konstrui

## Konstruktionen Dreieck (Musterlösung)

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 4\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = \|\angle(C, A, B)\| = 37^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

# Arbeitsblatt Geometrie

**Voraussetzung**  
Gegeben sei

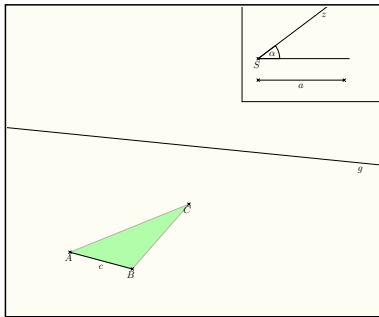
**Zeichenfläche**  
Hier kann  
Zirkel und Lineal

**Aufgaben**  
(a) Konstruiere

## Konstruktionen Dreieck (Musterlösung)

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 4\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = \|\angle(C, A, B)\| = 37^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

# Arbeitsblatt Geometrie

**Voraussetzung**  
Gegeben sei

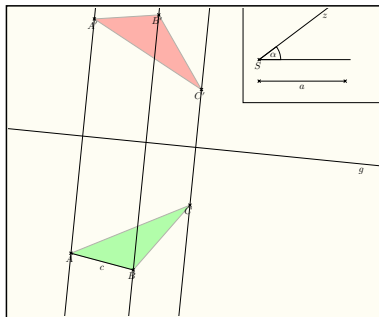
**Zeichenfläche**  
Hier kann  
Zirkel und Li

**Aufgaben**  
(a) Konstrui

## Konstruktionen Dreieck (Musterlösung)

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 4\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = \|\angle(C, A, B)\| = 37^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

# Arbeitsblatt Geometrie

Voraussetzung  
Gegeben sei

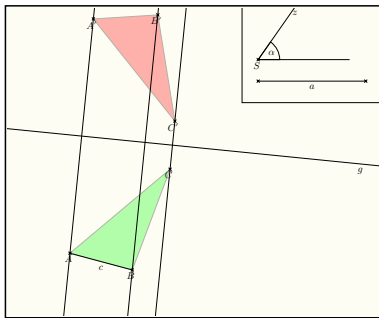
Zeichenfläche  
Hier kann  
Zirkel und Li

Aufgaben  
(a) Konstrui

## Konstruktionen Dreieck (Musterlösung)

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 5\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = |\angle(C, A, B)| = 55^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.

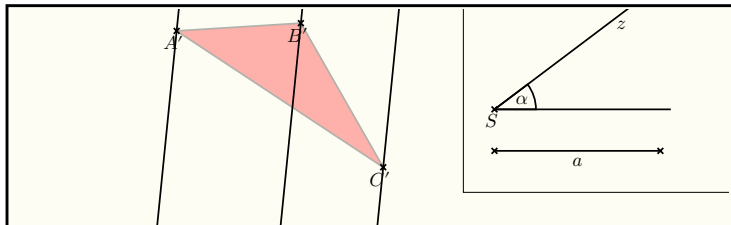


### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 4\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = |\angle(C, A, B)| = 37^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



## Aufgaben

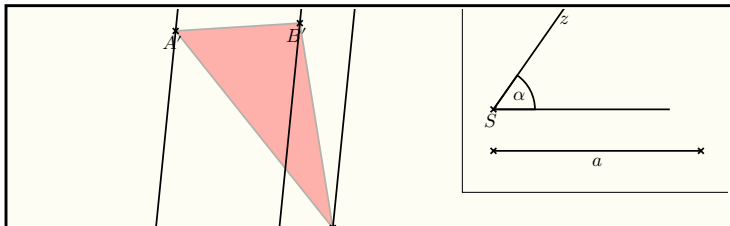
### (a) Konstruiere

#### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

### Voraussetzung

Zu einem Dreieck  $\triangle ABC$  seien die Seitenlängen  $c = 3\text{ cm}$ ,  $a = 5\text{ cm}$  und das Maß des Winkels  $\angle(C, A, B)$  bekannt,  $\alpha = |\angle(C, A, B)| = 55^\circ$  bekannt. In der Konstruktionsvorlage sind die Punkte  $A$  und  $C$ , sowie die Länge der Seite  $a$  und das Maß des Winkels vorgegeben. Weiterhin sei eine Gerade  $g$  vorhanden.



## Aufgaben

### (a) Konstruiere

#### Aufgabe

- Konstruiere das Dreieck  $\triangle ABC$  über der Strecke  $\overline{AC}$  zu den gegebenen Maßen.
- Konstruiere das an der Gerade  $g$  gespiegelte Dreieck  $\triangle A'B'C'$ .
- Zeichne die Geraden  $AA'$ ,  $BB'$  und  $CC'$ . In welcher Beziehung stehen die Geraden zueinander?
- Begründe, warum die in (c) gezeichneten Geraden in der angegebenen Beziehung stehen.

## Zusammenfassung

- ▶ Mit Lua erhält eine einfache Skriptsprache Einzug in die Arbeitswelt von  $\text{\LaTeX}$ .
- ▶ TikZ in Verbindung mit Lua belebt die Geometrie auf Arbeitsblättern.
- ▶ Mit  $\text{\TeX}$ Live ist ein einfacher Einstieg in Lua $\text{\LaTeX}$  möglich.

### Link

- ▶ [Zip-Archiv](#) (Win-Zielverzeichnis: `c:\lua $\text{\LaTeX}$` )  
Enthält:  $\text{\TeX}$ Live 2013, Maxima,  $\text{\TeX}$ Studio  
Ermöglicht: Einfaches Arbeiten mit Lua $\text{\LaTeX}$  mit vielen Beispielen