# Pocketmods mit LaTeX und Python

https://github.com/UweZiegenhagen/TalksAndArticles/2021-Dante-Herbst

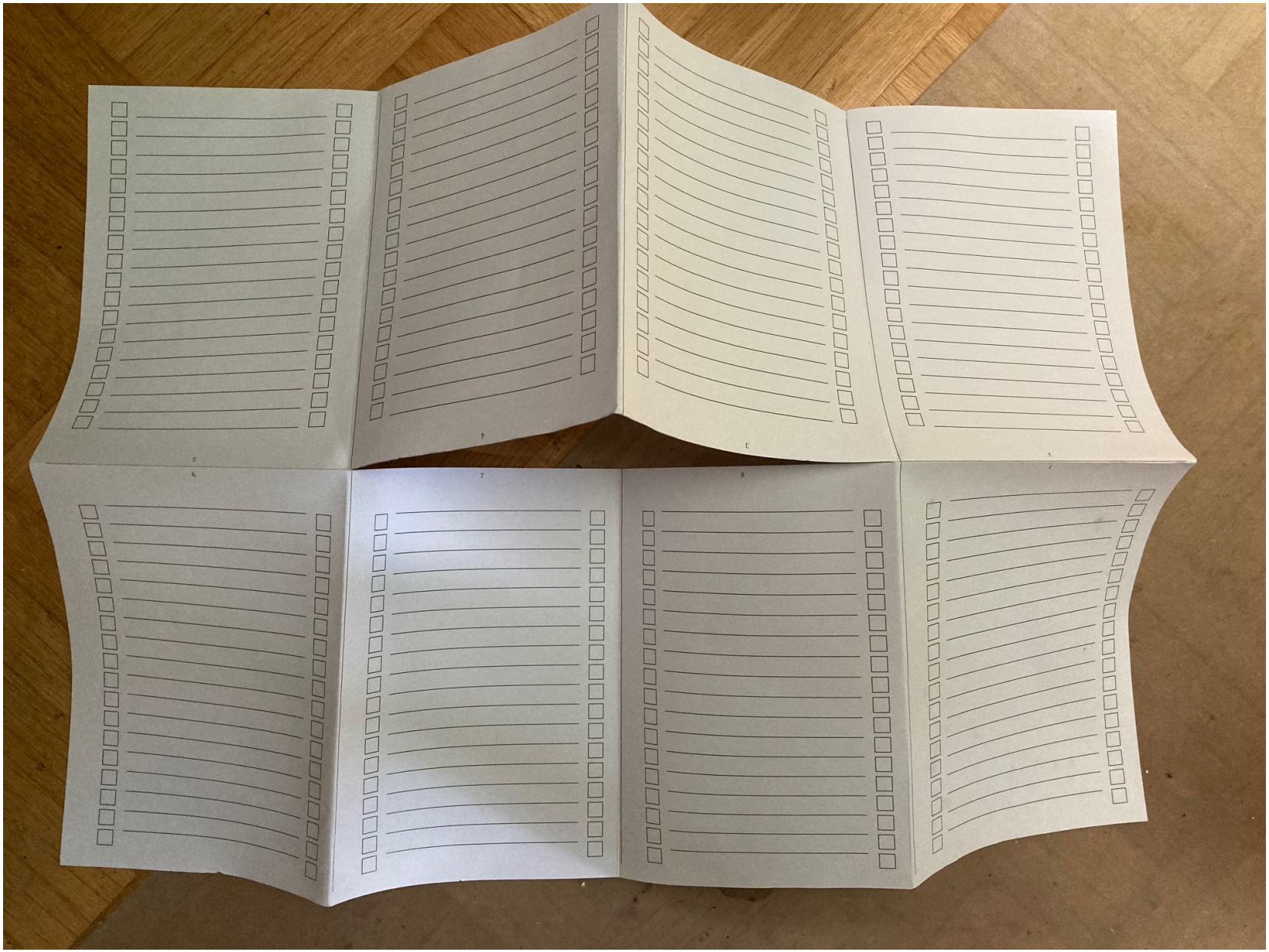
Uwe Ziegenhagen

18. September 2021

# Über mich

- Geboren im „Speckgürtel“ Berlins
- Seit 2008 in Köln
- Business Analyst bei verschiedenen Finanzdienstleistern
- Seit 2020 bei der Toyota Kreditbank im Bereich *Business Intelligence & Treasury*
- TEXnisches Interesse: Satzautomatisierung (mit Python)

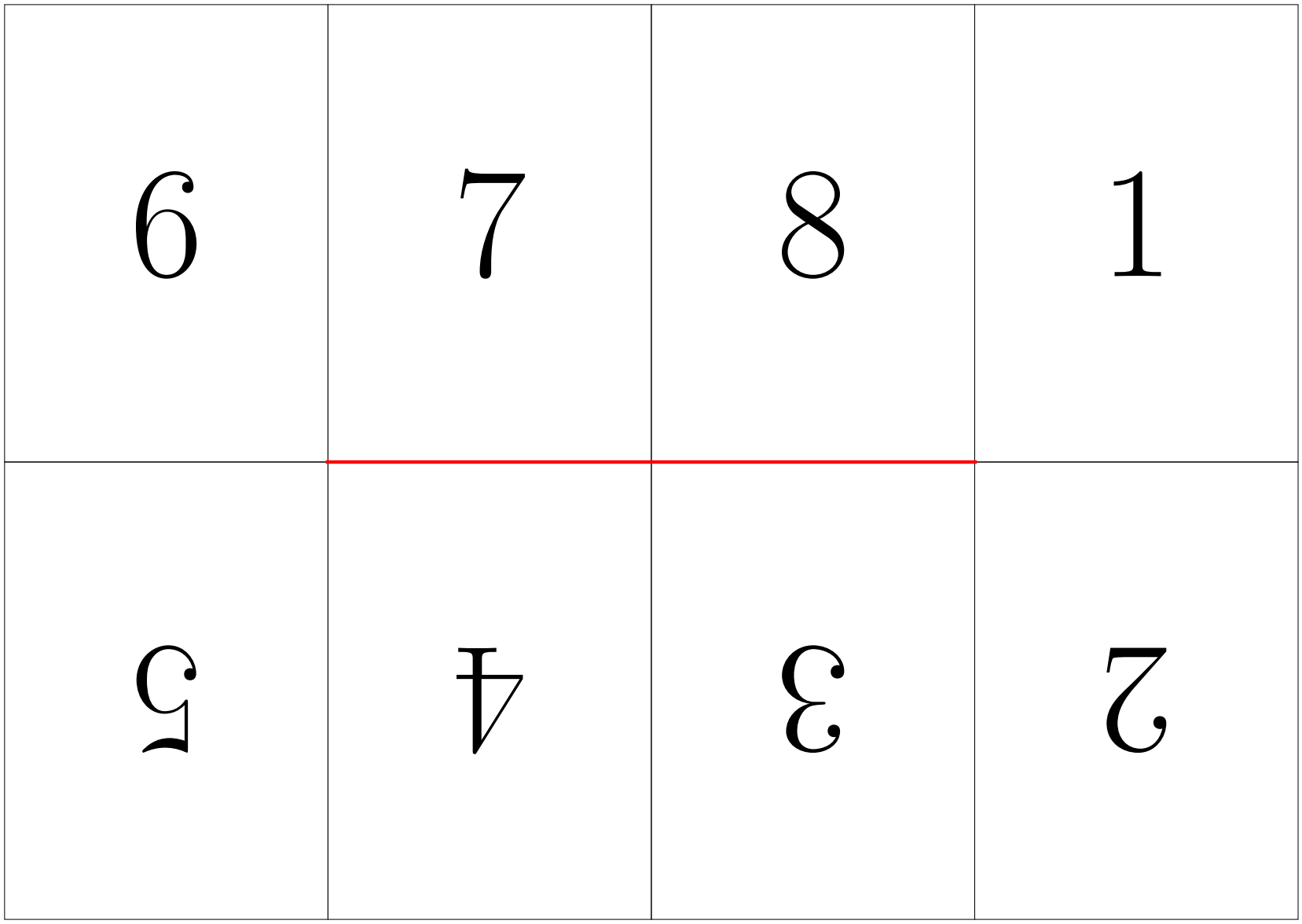
# Pocketmod



# Über Pocketmods

- Pocketmod = cleveres Faltsystem
- Macht aus einer DIN A4 oder DIN A3 Seite Kalender bzw. Notizbuch
- Online-Generator unter pocketmod.com
- Falt-Tutorial unter https://www.youtube.com/watch?v=FHD-01Sc9dM
- Mein erster Artikel zum Thema in DTK 3/2010

# Faltschema



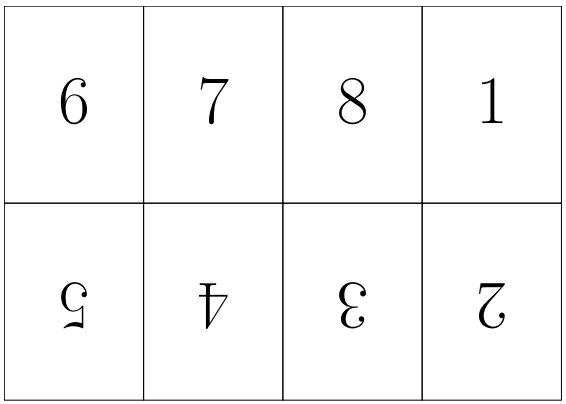
# pgfpages: Physische und Logische Seiten

- `pgfpages` ist Teil von TikZ
- Greift in den Ausgabemechanismus ein
- „Schubst“ Seiten beliebig umher
- Erlaubt z.B. `\pgfpagesuselayout{2 on 1}`
  $\Rightarrow$ 2 Seite auf 1 Seite darstellen

# Beispiel für Seite Nr. 1

```
\pgfpagesphysicalpageoptions{%
logical pages=8,%
physical height=\pgfpageoptionheight,%
physical width=\pgfpageoptionwidth,%
current logical shipout=\pgfpageoptionfirstshipout%
}

\pgfpageslogicalpageoptions{1}{%
border shrink=\pgfpageoptionborder,%
resized width=.25\pgfphysicalwidth,%
border code=\pgfusepath{stroke},%
resized height=0.5\pgfphysicalheight,%
center=\pgfpoint{.875\pgfphysicalwidth}{.75\pgfphysicalheight}%
}%
```



# Pocketmods mit TikZ – TODO Liste

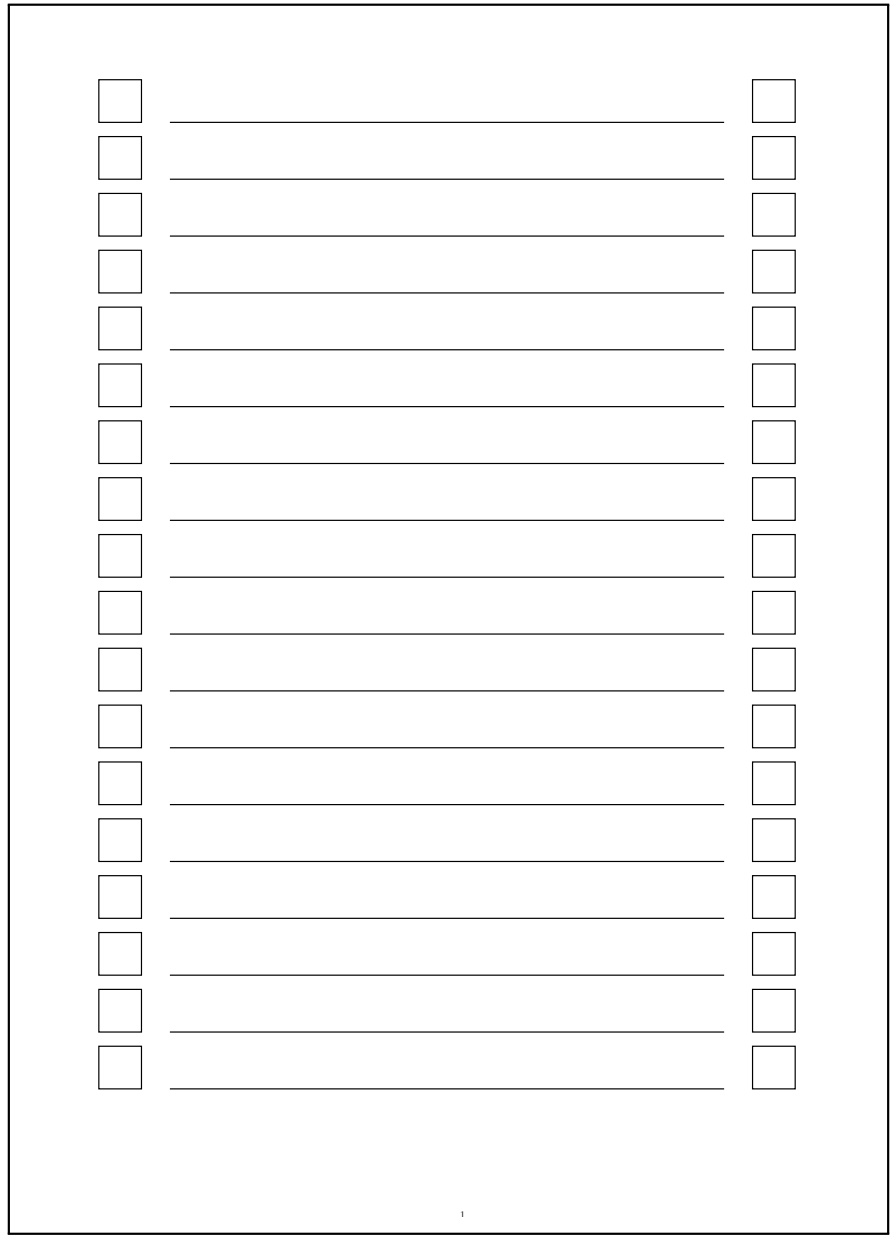
```
\begin{tikzpicture}
\draw[very thick](0,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                            -- ++(0,-1.5)--cycle;
\draw[very thick](2.5,\i*2)-- (22,\i*2);
\draw[very thick](23,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                             -- ++(0,-1.5)--cycle;
\end{tikzpicture}
```

# Pocketmods mit TikZ – 1. Schleife

Eine Schleife, um eine Seite zu erzeugen.

```
\begin{tikzpicture}
\foreach \i in {0,...,-17}{%
\draw[very thick](0,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                            -- ++(0,-1.5)--cycle;
\draw[very thick](2.5,\i*2)-- (22,\i*2);
\draw[very thick](23,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                             -- ++(0,-1.5)--cycle;
}
\end{tikzpicture}
```

# Ergebnis der ersten Schleife



# Pocketmods mit TikZ – 2. Schleife

Zweite Schleife für die Erstellung der acht Seiten.

```
\forloop{ct}{1}{\value{ct} < 9}{%
\begin{tikzpicture}
\foreach \i in {0,...,-17}{%
\draw[very thick](0,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                            -- ++(0,-1.5)--cycle;
\draw[very thick](2.5,\i*2)-- (22,\i*2);
\draw[very thick](23,\i*2)-- ++(0,1.5)-- ++(1.5,0)
                             -- ++(0,-1.5)--cycle;
}
\end{tikzpicture}
\clearpage
}
```

$\Rightarrow$ PDF mit 8-Seiten TODO-Liste

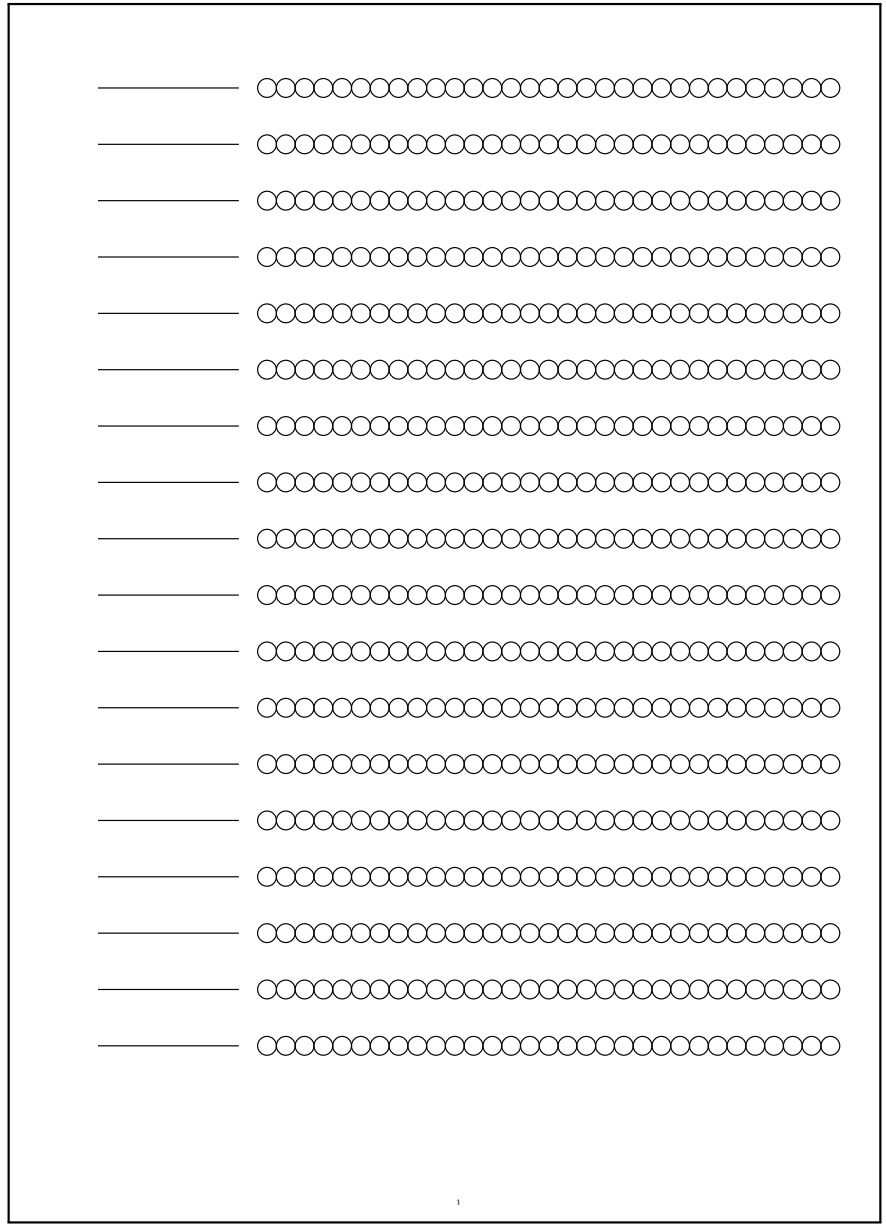
# Kombination mit pgfpages Code

# Aktivitäts-Tracker I

- Aktivitäten wie „Katzenklo gesäubert?“, „2 Liter Wasser getrunken?“, etc
- pro Tag ein Pünktchen

```
\begin{tikzpicture}
\foreach \i in {0,...,-17}{%
  \draw[very thick](0,\i*2)-- (5,\i*2);
    \foreach \j in {0,...,30}{%
          \draw[very thick](\j/1.5+6,\i*2) circle (9.5pt);
        }
}
\end{tikzpicture}
```

# Aktivitäts-Tracker II



# Kombination mit Python

Idee: a) Monatskalender erzeugen und b) mit Events aus Google Calendar befüllen $\Rightarrow$ Python nutzen, um LaTeX zu erzeugen

Python:

- eine Skriptsprache, erfunden in den Niederlanden
- leicht erlernbar, kompakt, aber sehr mächtig
- erlaubt objektorientierte, funktionale und Batch-Programmierung
- Standard im Bereich ML und KI
- Fokus auf lesbaren Code

# Monatskalender Teil 1

Ziel: 2-dimensionales Array mit Wochen und Tagen,
Beispiel für den September 2021:

```
[['  ', '  ', '01', '02', '03', '04', '05'],
 ['06', '07', '08', '09', '10', '11', '12'],
 ['13', '14', '15', '16', '17', '18', '19'],
 ['20', '21', '22', '23', '24', '25', '26'],
 ['27', '28', '29', '30', '  ', '  ', '  ']]
```

# Monatskalender Teil 1

```
def prep_Cal(year, month):
    first = date(year, month, 1)
    last = first + relativedelta(months=+1, seconds=-1)
    month = []
    for i in range(5):
        month.append(['␣␣' for i in range(7)])

    if last.weekday() < 3:
        month.append(['␣␣' for i in range(7)])

    week_counter = 0
    daterange = pd.date_range(first, last)

    for i in daterange:
        weekday_before = (i + relativedelta(days=-1)).weekday()
        if i.weekday() > weekday_before:
            month[week_counter][i.weekday()] = str(i.day).zfill(2)
        else:
            week_counter = week_counter + 1
            month[week_counter][i.weekday()] = str(i.day).zfill(2)
    return month
```

# Monatskalender Teil 2

```
def as_table(month):
    print(r'\begin{tabular}{cccccccc}')
    print(r'Mo & Di & Mi & Do & Fr & Sa & So \\')
    for week in month:
        print(' & '.join(week), r'\\')
    print(r'\end{tabular}')
```

| Mo | Di | Mi | Do | Fr | Sa | So |
|---|---|---|---|---|---|---|
| | | 01 | 02 | 03 | 04 | 05 |
| 06 | 07 | 08 | 09 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | | | |

# Monatskalender Teil 3 - TikZ Version

- Schönere Kalender mit Tikz
- Einzelne nodes mit Position und Inhalt
- Formatierung über Stildefinition

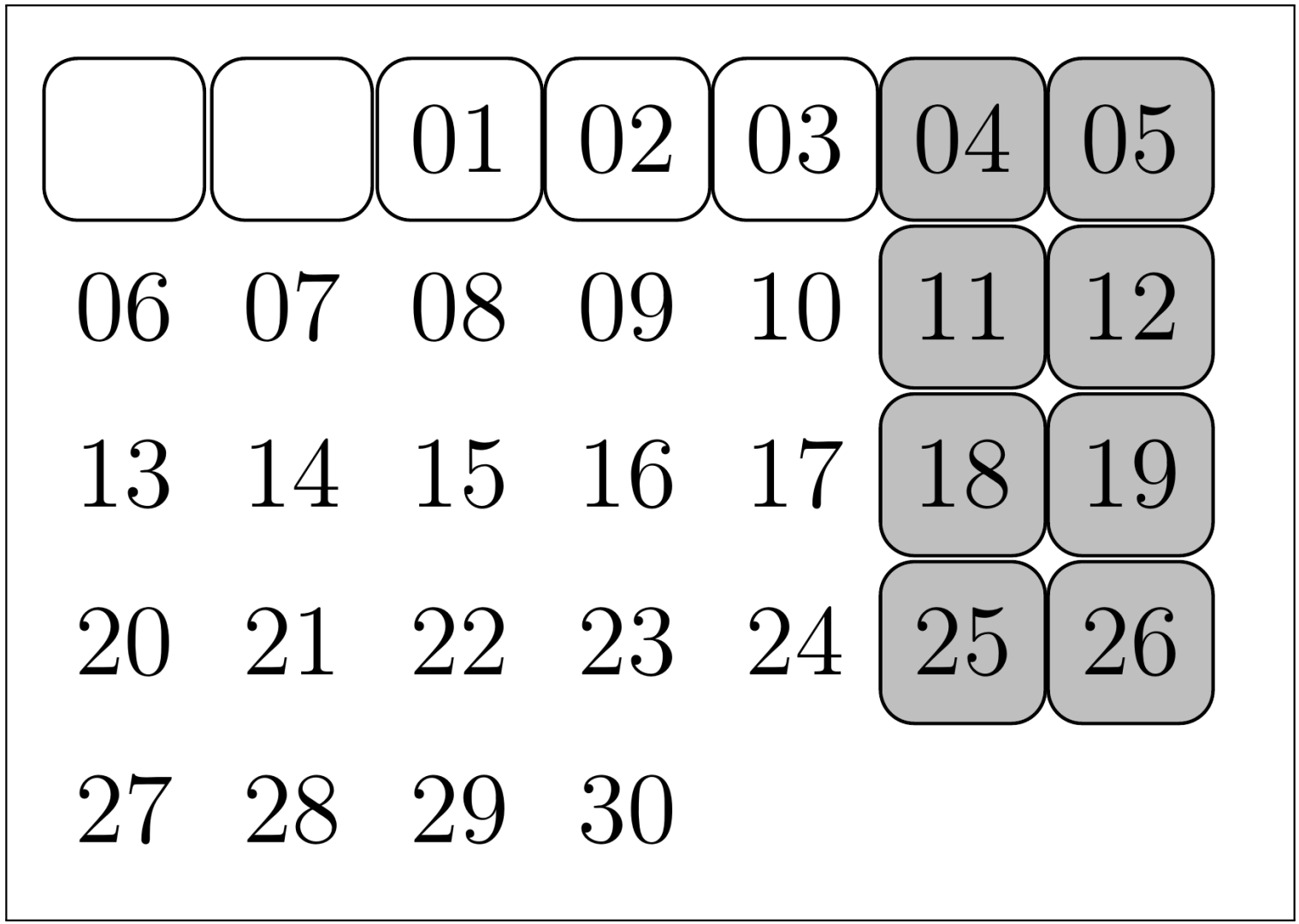
```
\node at (3,0) {02};
```

```
def as_tikz_table(month):
    for cntw, week in enumerate(month):
        for cntd, day in enumerate(week):
            print(f'\\node␣at␣({cntd},{-cntw})␣{{{day}}};')
```

# Monatskalender Teil 3 - TikZ Version

```
def as_tikz_table_wknd(month):
    for cntw, week in enumerate(month):
        for cntd, day in enumerate(week):
            stil = '[wknd]' if cntd in [5,6] and \
            day != '␣␣' else ''
            print(f'\\node␣{stil}␣at␣({cntd},{-cntw})␣{{{day}}};')
```

# Monatskalender Teil 3 - TikZ Version Ergebnis



# Google Calendar API

API für Google Calendar

- Account unter `https://console.cloud.google.com`, Projekt erzeugen, Anmeldedaten erzeugen, OAuth 2.0-Client-ID erzeugen, Credentials.json herunterladen
- `https://developers.google.com/calendar/api/quickstart/python`
-
-
-
-
-