Free software for...
Learning
Desktop publishing
repeat 500 [ circle 10 + repcount/10 fd 5 + repcount/10 lt 10 ]
Turtle vector graphics in LibreOffice

- Modern Logo programming environment
  - Printing measurements, shapes and shape grouping, zoom and turtle-tracing, Python base and integration, localized commands (now English and Hungarian)

- Quality graphics, storage and printing
  - Interactive vector graphics, anti-aliasing, transparence, ISO OpenDocument format, PDF and SVG export, Graphite font technology

- LibreOffice Writer Extension Toolbar
  - Turtle forward, back, turn left and right, program start and stop, home, clear screen, fast command line
Education and LibreLogo

- Turtle graphics, algorithms in elementary schools
- Using office suites (LibreOffice): handling pictures, page settings, PDF-export etc.
- Practical programming knowledge: Python list, tuple, dict, set data structures etc.
- New motivations: art, desktop publishing, open source code of LibreLogo (thousand lines in Python/PyUNO)
# Logo and LibreLogo

<table>
<thead>
<tr>
<th>Logo</th>
<th>Differences</th>
<th>LibreLogo</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>turnright 90 = rt 90</code></td>
<td><strong>optional clock positions</strong> ► (suitable for the lower grades)</td>
<td><code>turnright 90° = rt 90 = turnright 3h</code></td>
</tr>
<tr>
<td><code>forward 1 = fd 90</code></td>
<td><strong>DTP point, inch, cm, mm</strong> ► ◄ pixel</td>
<td><code>forward 1pt = fd 1 = fd 1in/72 = fd 2.54cm/72</code></td>
</tr>
<tr>
<td><code>fill (flood-fill, need position)</code></td>
<td><strong>vector graphics</strong> ► ◄ raster graphics</td>
<td><code>fill (close and fill actual shape)</code></td>
</tr>
<tr>
<td>&quot;word [string]&quot;</td>
<td><strong>text notation</strong> ◄ <strong>writing standard</strong> ► ◄ <strong>formal (LISP)</strong></td>
<td>“string” (orthography, Writer), 'string' (Python), &quot;word, &quot;word&quot;</td>
</tr>
<tr>
<td><code>lists [] (eg. 1-line instruction list)</code></td>
<td><strong>Python in Logo turtle shell</strong> ◄ <strong>functional programming language</strong></td>
<td>blocks [ ] (need space or line break) and lists [], eg. repeat 5 [ ellipse [5, 10] ]</td>
</tr>
</tbody>
</table>
Checkerboard

- Filled complex shape (with a single line, see illustration)

to checkerboard size x y ; 2x × 2y sq.
repeat x [
    fd size\*y*2 rt 90 fd size rt 90
    fd size\*y*2 lt 90 fd size lt 90
] fd size\*y*2 lt 90
repeat y [
    fd size\*x*2 lt 90 fd size lt 90
    fd size\*x*2 rt 90 fd size rt 90
] fd size\*x*2 fill
end

checkerboard 1cm 2 2
to line pattern
  for i in pattern [
    pu fd 10 pd fillcolor "white"
    rectangle [10, 10]
    fillcolor "black"
    if i = "x" [ circle 8 ]
  ]
  pu rt 90 fd 10 lt 90
  back 10 * count pattern
end

rt 90
line " x "
line "  x"
line "xxx"
Pencils

- Pencils are different pictures (grouped shapes)

```plaintext
to triangle size color
    repeat 3 [ fd size lt 120 ] fc color fill
end

to box size f
    repeat 2 [ fd size*10 rt 90 fd size*f rt 90 ]
end

to pencil size color
    box size 1 fc color fill
    box size 2/3 box size 1/3
    close rt 150 triangle size "chocolate"
    fd size*0.75 triangle size/4 color
    back size*0.75 lt 150
end

pensize 2 rt 90
for color in ["red", "orange", "yellow", "lime", ~
    "skyblue", "navy", "violet"] [ pic [ pencil 30 color ]
    pu rt 90 fd 45 lt 90 pd ]
```
Pieces

- Unicode characters (♔♚♕♛♖♜♗♝♘♞♙♟)
- In “invisible” squares for manual positioning
- White pieces: transparent white Unicode pieces combined with dark pieces in white
- Grouping pieces with their background