Exercise:
SKILL

Prof. Dr. P. Fischer

Lehrstuhl für Schaltungstechnik und Simulation
Uni Heidelberg

Exercise A ('warmup'):

- Calculate 1 + 1
- What is Sinus (1)?   ;sin 1 (radian)
- Find a way to get π

- Define two variables x and y, set them to 2 and 3 and multiply them

- Let p be a *list* of two float values
- Get the first and second element of the list p

- Add the first and second element directly (in one command)

- Define a function (Double ..) which outputs the double of a (float) argument

- Calculate the area of a rectangle (a list of two points)

- Define a function **area** which does this

For
- **(setq A list(1:2 3:4))**

your function
- **(area A)**

should return
- **4**

Exercise C: Sum of Elements

- Define a function **sumup** which calculates the sum of the elements of a list
  - Assume all elements are numbers

- Write 2 different versions at least:
  - Using **foreach**
  - Explicitly picking the **nth** element (i.e. using an index)

- You could try two more versions:
  - Stepping through the list with **car** / **cdr**
  - A recursive version

- Create a new layout. Create a rectangle on *metal1* and a path on *poly* by hand.
- From the CIW store the ID of the view in `myID`
- What is `myID~>shapes` ?
- Examine the two objects

- In the CIW: Assign a new value to the bounding box of the rectangle and check what happens in the open cell view.

- Create a new rectangle using the command from the lecture
- Search in the manual how to place a circle or a path.

- Write a routine to create a circle (quadrant) from many small rectangular stripes

  - Fix the layer to `'("ME2" "drawing")`
  - Fix the origin at (0,0)
  - Input parameters are
    - the outer `radius`
    - the (x) `step` size of the strips
  - It is sufficient to generate a quadrant, i.e. We define a procedure '`quadrant`' and call `(quadrant radius step)`
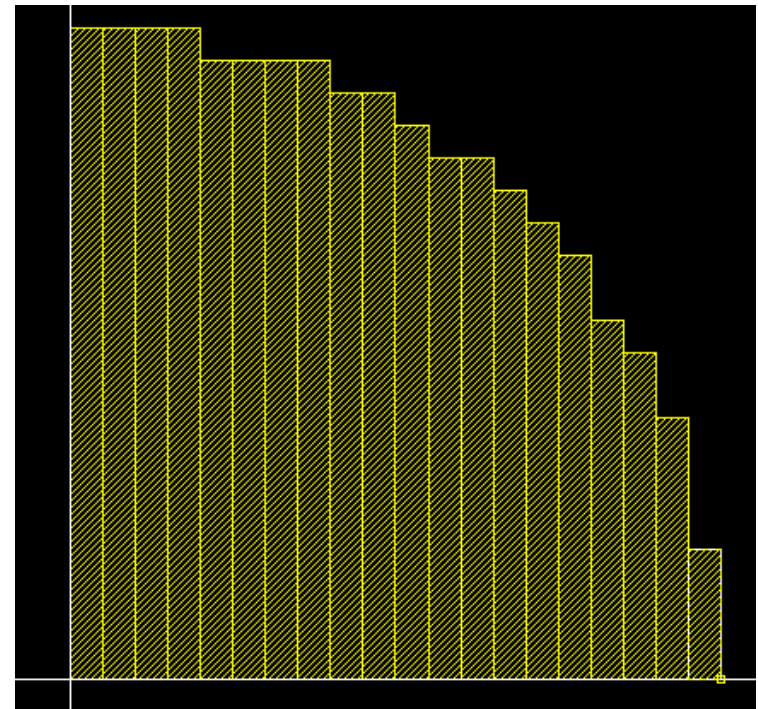
- For a start assume that `radius` is an integer multiple of `step`
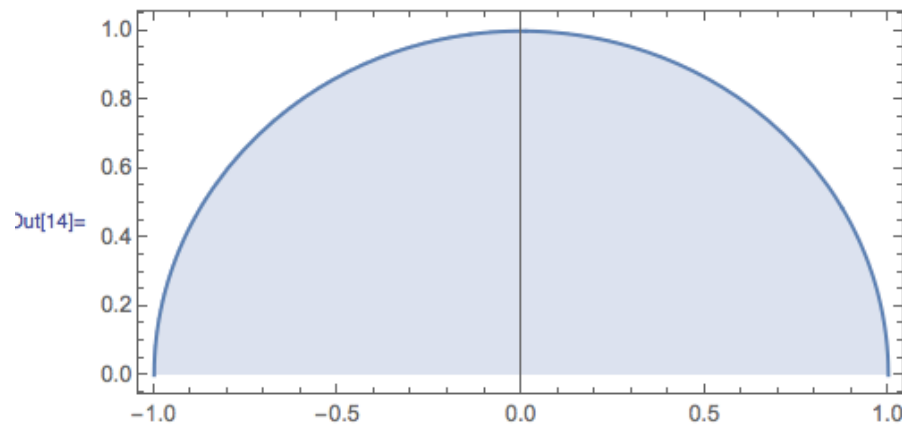- Make sure the y-coordinates are on 'some' grid, for instance multiples of `step`

- The formula for the 'ideal' area of a slice is obtained by integration:

$\text{In[1]:=} \quad \mathbf{f[R\_, x\_] = \sqrt{R^2 - x^2}}$

$\text{Out[1]=} \quad \sqrt{R^2 - x^2}$

$\text{In[14]:=} \quad \mathbf{Plot[f[1, x], \{x, -1, 1\}, AspectRatio \rightarrow 1/2, Frame \rightarrow True, Filling \rightarrow Axis]}$

$\text{In[11]:=} \quad \mathbf{\$Assumptions = R > 0 \&\& x1 > -R \&\& x1 < R \&\& x2 > -R \&\& x2 < R \&\& x1 < x2;}$

$\text{In[12]:=} \quad \mathbf{Integrate[f[R, x], \{x, x1, x2\}]}$

$$\text{Out[12]=} \quad \frac{1}{2}\left(-x1\sqrt{R^2 - x1^2} + x2\sqrt{R^2 - x2^2} + R^2\left(-\text{ArcTan}\left[\frac{x1}{\sqrt{R^2 - x1^2}}\right] + \text{ArcTan}\left[\frac{x2}{\sqrt{R^2 - x2^2}}\right]\right)\right)$$

$\underbrace{\qquad}_{\text{x1s}} \qquad \underbrace{\qquad}_{\text{x2s}}$

- Check in your .cdsinit file how the shortcuts for changing the grid are defined
  - Change the message that is printed
  - Add another key for a new grid

- Define a function which places a rectangle (1x1 um^2) at the cursor position
  - You get the cursor position with
    (hiGetPoint (hiGetCurrentWindow))
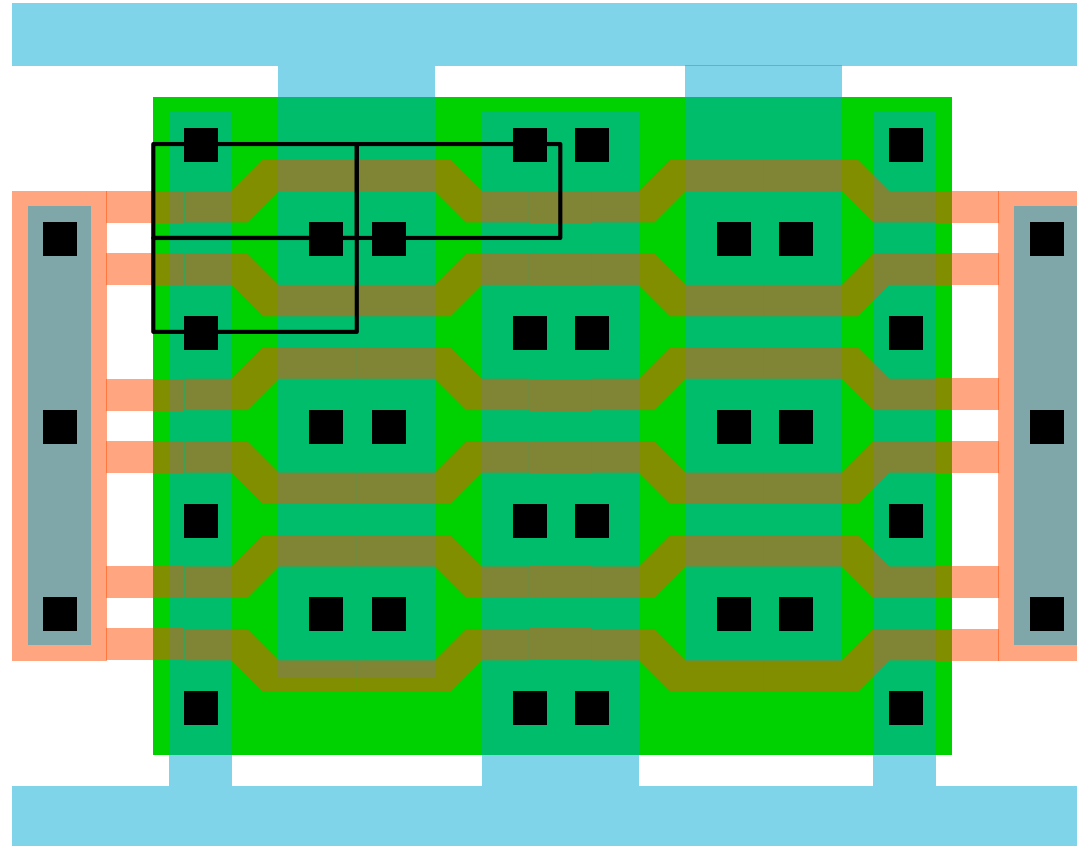- Associate the function to a bind key

- Create a PCELL which generates the quarter circle from exercise E

- Make a PCELL which creates a Waffle transistor like this

- Make small unit layout cells (core/leftright/top/bottom).
- The PCELL just composes the layout from these elements