Web Components Icons font - Guidelines

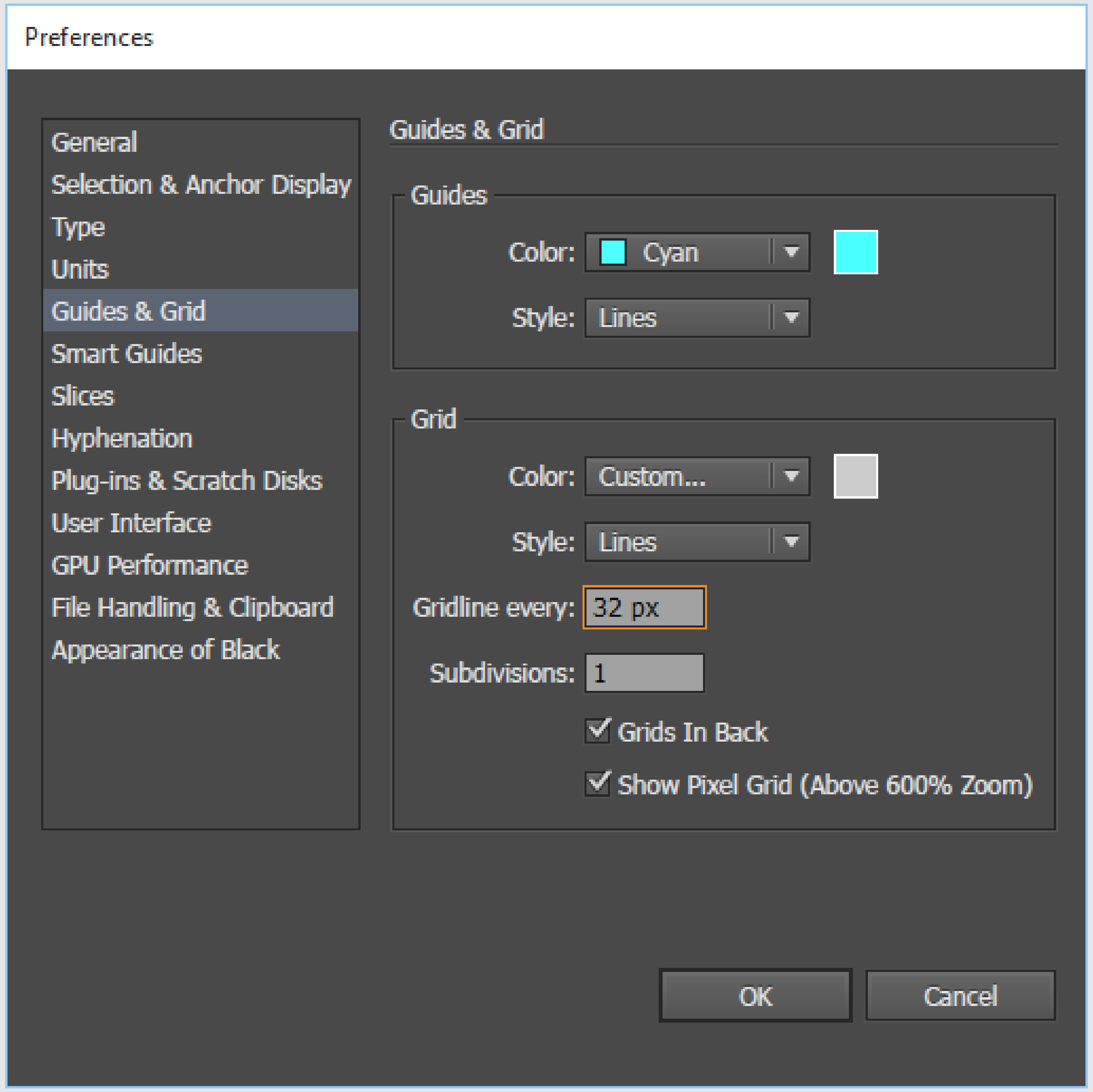
Progress®

1. Grid, proportion, and style

1.1. ADOBE ILLUSTRATOR PREFERENCES / GUIDES & GRID

- The Artboard size /width x height/ is 512px X 512px

- Gridline every: 32px



Preferences

Guides & Grid

Guides

Color: 🟦 Cyan

Style: Lines

Grid

Color: Custom...

Style: Lines

Gridline every: 32 px

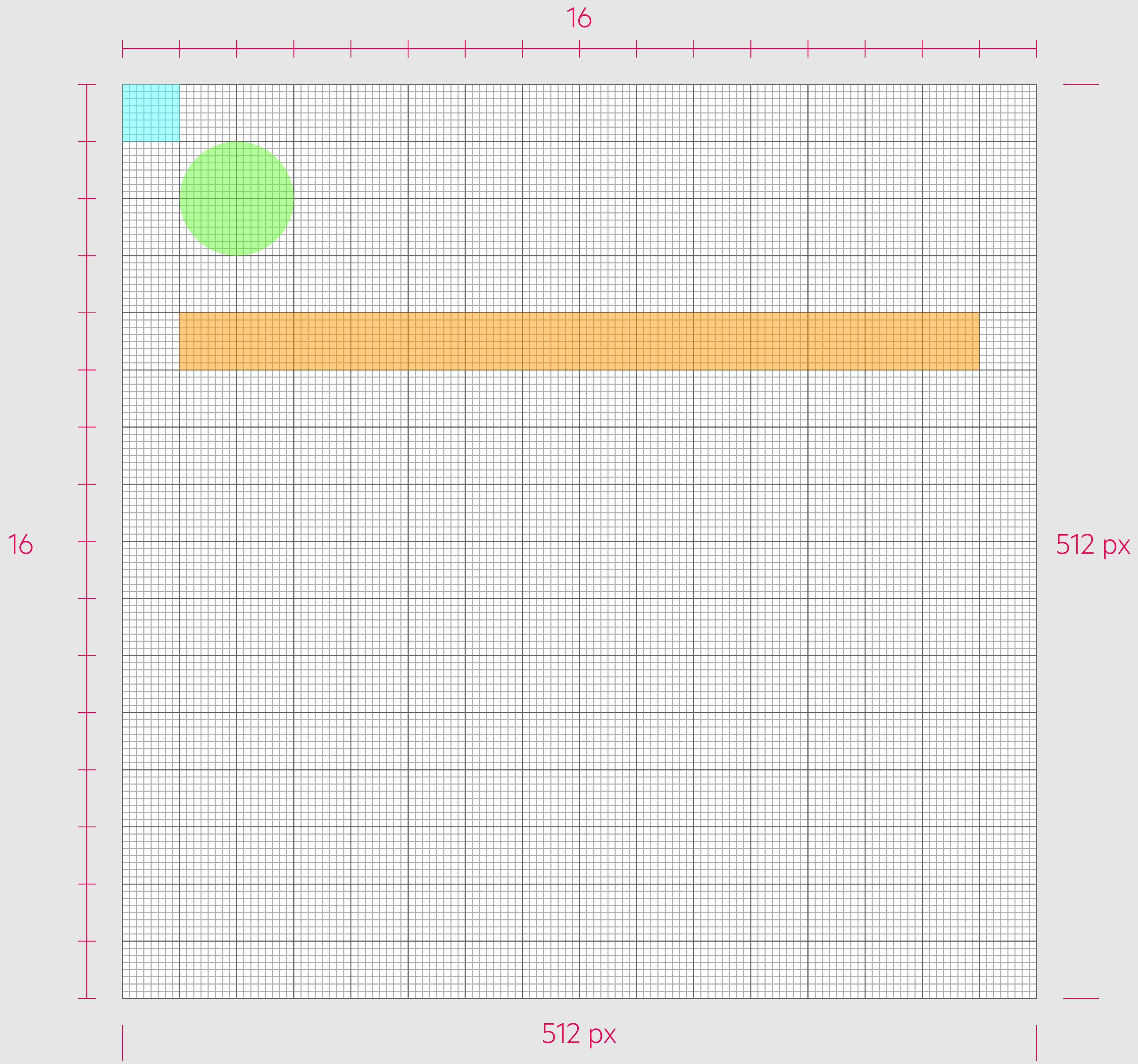
Subdivisions: 1

☑ Grids In Back

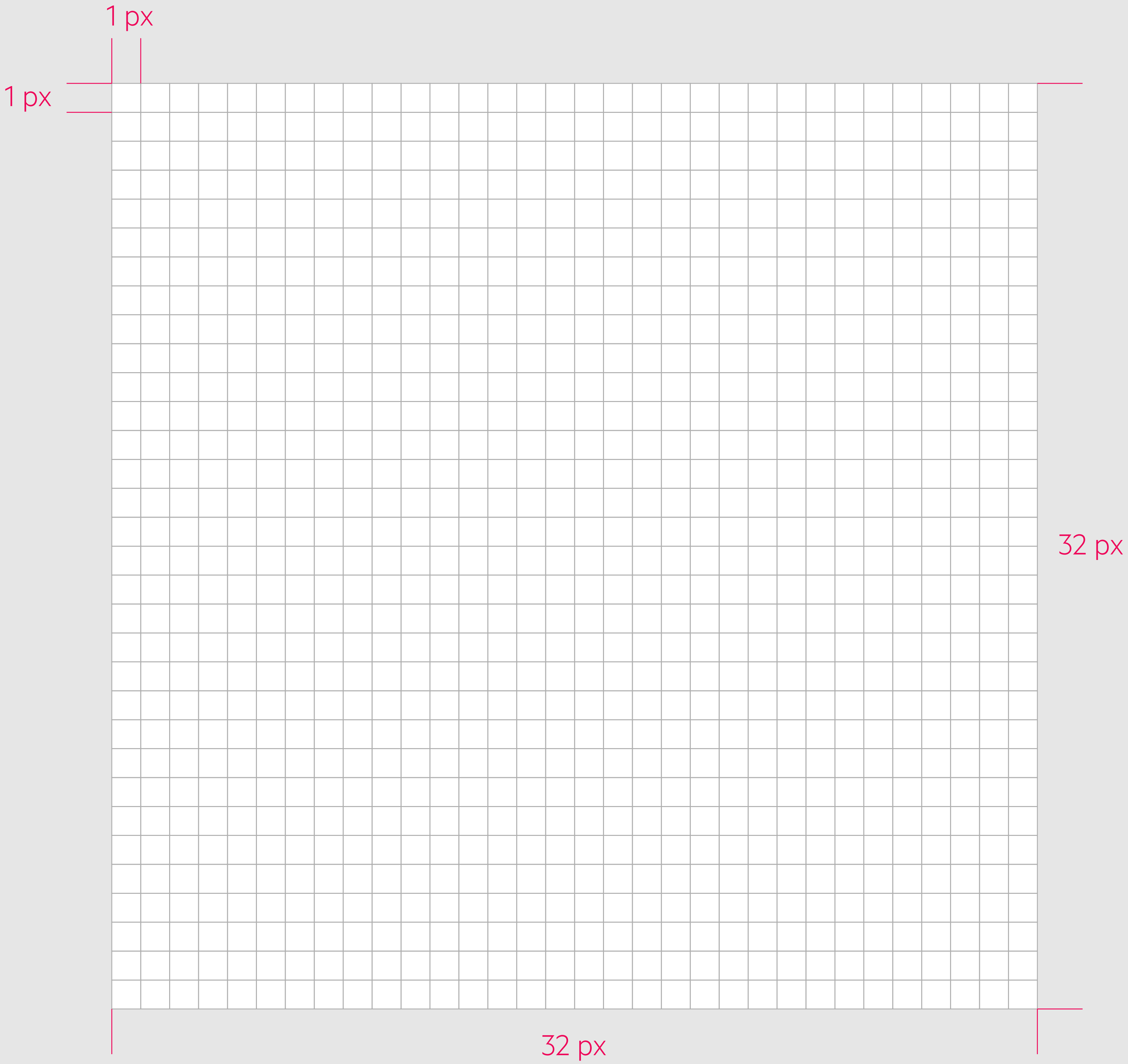
☑ Show Pixel Grid (Above 600% Zoom)

1.2. UNIT GRID

- Unit grid constructed to be 16 x 16 unit /a/, with edges at 1.
- 🟦 = a;   16a = 512 px;   a= 32 px
- R = 2a
- W = a

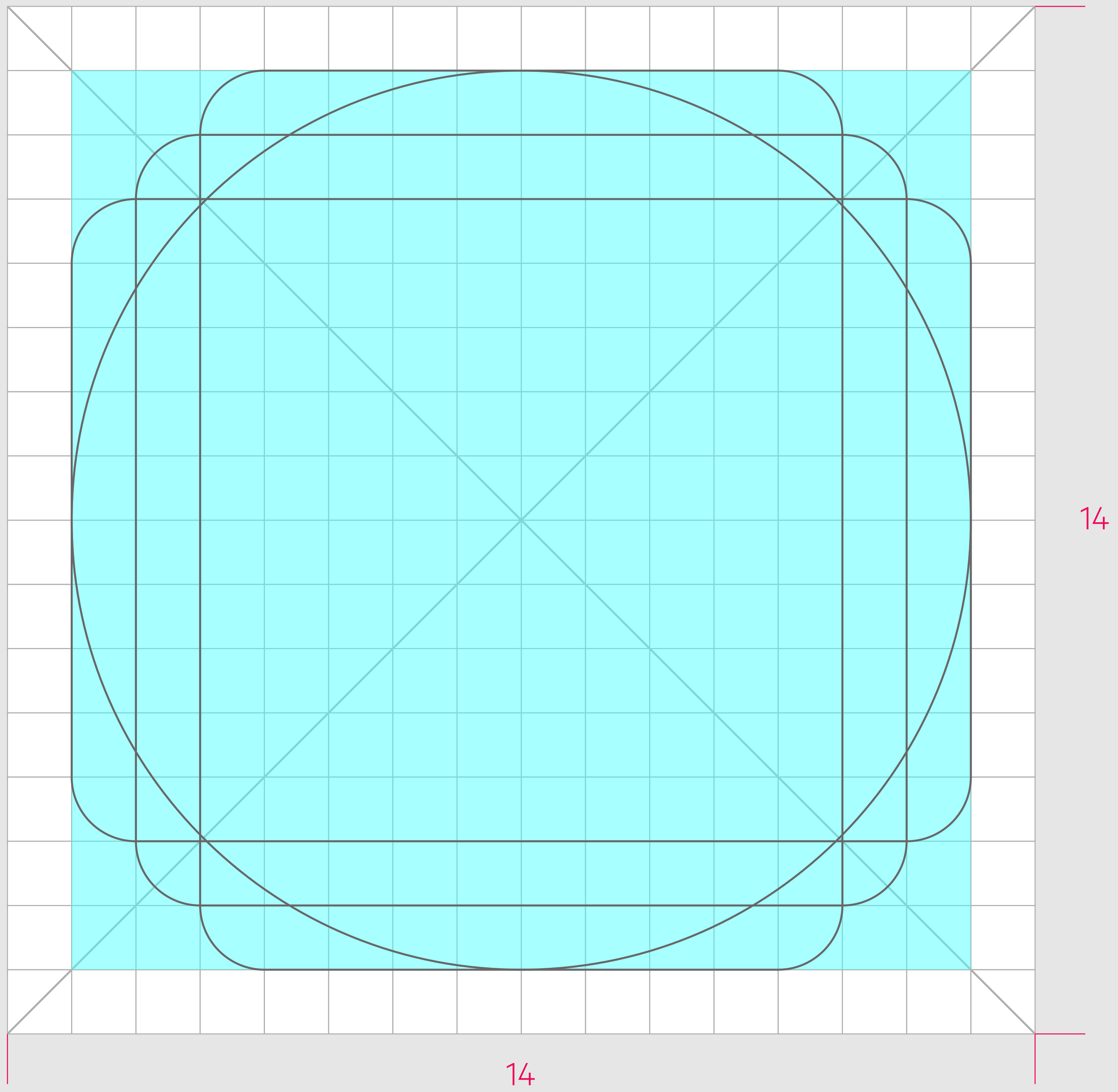


1.3. GRID



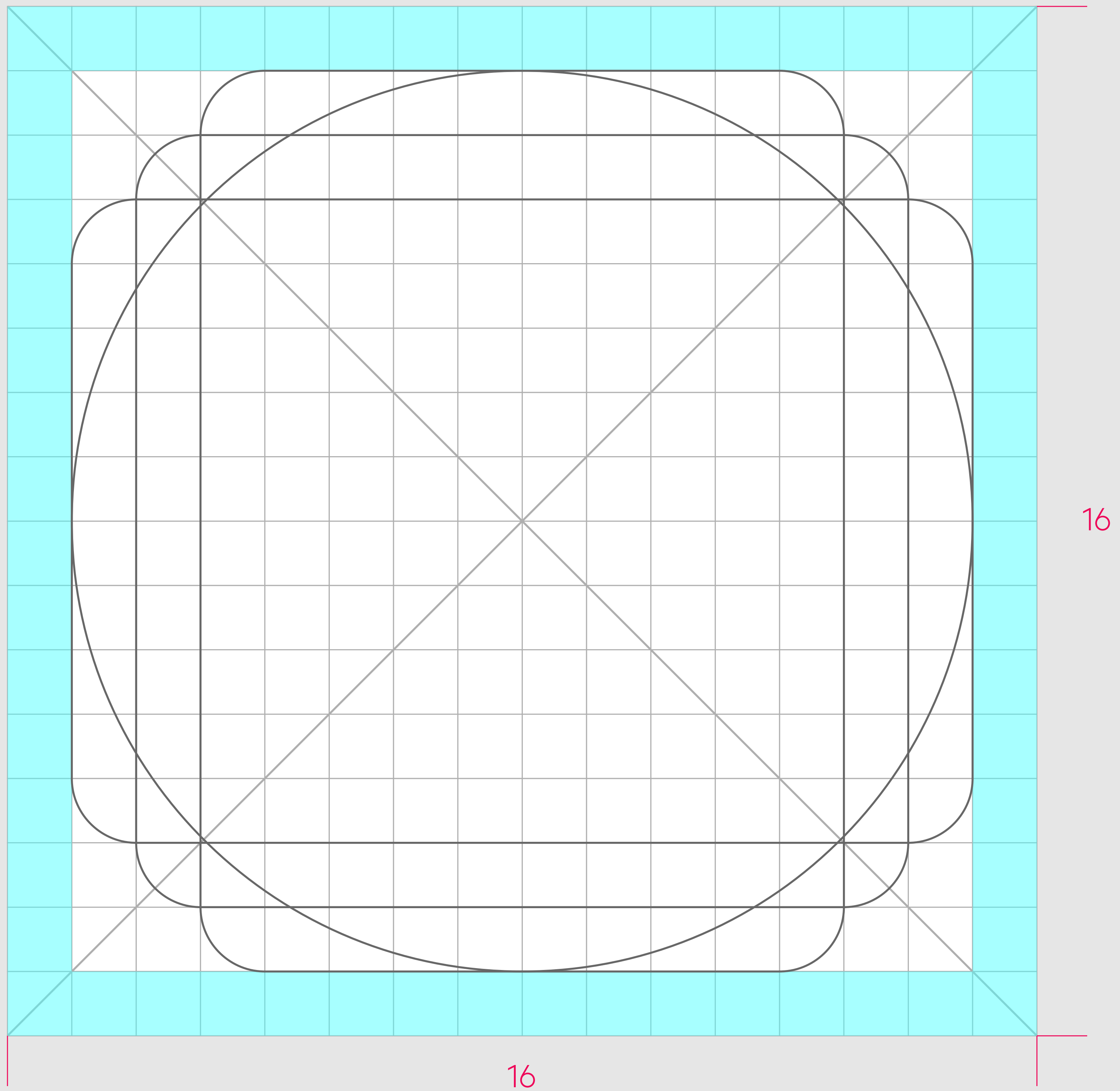
1.4. LIVE AREA

- Icon content is limited to the 14 x 14 live area, with 2 of trim around the perimeter.
- If the icon contains a lot of details, then live area should be increased to 16x16.
- The icon always should be placed in the middle of the live area.
- If the icon cannot be placed in the middle of the live area (because of the shapes specifics), then the preffered place is a top left corner of the live area.



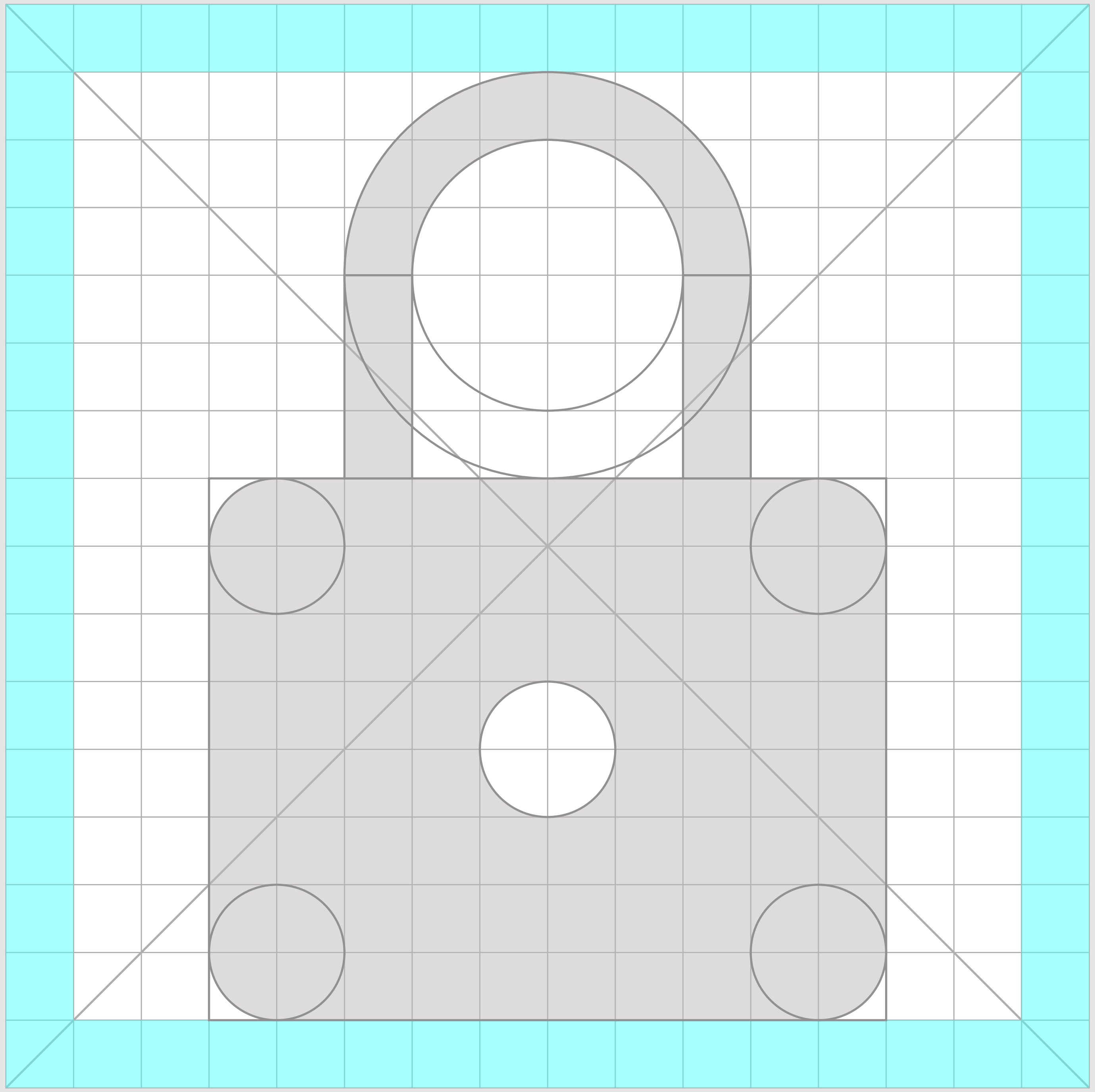
1.2. TRIM GRID

- 2 units of empty space make up the trim area surrounding the 14 x 14 live area.



1.6. STYLE

- Flat and graphic to match the brand look and feel.
- Outlined rather than filled icons to relate back to the brandmark symbol.
- Pure geometric shapes to be used: straight lines and circles. No organic shapes.
- Preset standards have been determined for specific keylines: circle, square, rectangle, orthogonals, and diagonals. This small palette of universal and simple elements has been developed to unify icons and systemize their placement on the grid.
- All lines should be either vertical, horizontal or at a 45-degree angle (relating to symbol angles).
- Before final save of the SVG file, you should unify shapes and make them to compound path.



2. Font Structure, Set Name & Icon Name

2.1. FONT STRUCTURE

The Glyph Font consists of separated sets with similar elements. /eg. Navigation and Layout, Actions, Media, AlertNotification, Toggle, Image, Editor, etc./

The separations of the sets allows to have right positioning and consistent hexadecimal code for icons in the same set.

The structure of each set is: 1. Code-SetName

1.1. AI folder

1.2. SVG folder

1.2.1. icon-name.svg

1.2.2. icon-name-small.svg

2.2. SET & ICON NAME

- the folder's name has to contain "Hexadecimal Code", "-", and letters, without empty spaces;
- icon name should contain only small letters, number and "-" to separate different words, without empty spaces;
- icon name should be detailed, descriptive and intuitive;
- for small icon /the same icon with smaller size/, there should be added suffix "-small";
- for all icons with direction should be used:

-- "up", "right", "down" and "left" instead "n", "e", "s", "w";

-- "up-right", "up-left", "down-right" and "down-left" instead of "ne, nw, se, sw";

-- "horizontal" instead of "h";

-- "vertical" instead of "v";

/the words from icon name are used for search tags/

3. Tool "ImgenUI" - run, import, load & take your new/updated font

3.1. TOOL DESCRIPTION

Tool is starting by double click on "ImgenUI.exe".

In essence, the tool reads the *.metadata.json & input.json files, loops through SVG folders, reads and extracts svg data and creates the output.json and output.css files.

3.2. FILES DESCRIPTION & REQUIREMENTS

• ***.metadata.json** - the file contains metadata objects of the generated font.

-- Object "metadata" (name, url, license, etc) - specific information about the licence & copyrights;

-- Object "preferences" - contains some subobjects:

--- Object "fontPref" (icon prefix, metadata, selector & classSelector).

--- Object "imagePref" (prefix, classSelector) - defines the data for the demo of the font.

--- gridSize - defines the optimized size for the font. The step of increase is a grid size. For example: if the grid size (unit grid) is 16, than the the generated font is optimized for a font size 16px, 32px, 48px etc.

--- Object "iconSets" (prevSize) - defines the size of the generated font in preview mode.

• **input.json** - the file contains the name of the project, name of the font, info about icon size (width, height), grid & arrays (sorted lists) of the icon sets.

For each icon set, there needs to be defined:

-- "name" - the folder’s name to contain “Hexadecimal Code”, “-”, and letters, without empty spaces;

-- "startingPoint" - “Hexadecimal Code”.

-- "icons" - the arrangment of the elements in each array provides a specific place (hexadecimal code) for the icon in the font. Icons are separated by comma ",".

--- Ligature - for each icon we can give more than one (original) name of the icon. Ligature are separated with "|".

- **outputlog.txt** - the tool has an option for validating that all svg files are in the correct format/dimensions and have the ability to either throw, or just warn.
- **output.json** - the output contains a composite json object:
  - -- meta data (extracted from *.metadata.json);
  - -- set of icons:
    - --- set data (programmatically assigned, incremental);
    - --- icon:
      - ---- meta data (extracted from svg file name and path).
      - ---- char data (programmatically assigned, incremental).
      - ---- icon svg path data (extracted from svg file content).
- **output.css** - a list of class names based on the *.metadata.json file.

3.3. CREATE FONT

Finaly when the tool's work is done, the output.json file is ready to be imported as a new project in IcoMoon app.

3.4. UPDATE FONT

If you want to add a new icon in the exisiting font, you must add a new icon name in the input file (input.json) to the end of the choosen set & run the tool.