PixelMan Manual

Pixel art characters for Typst using CeTZ

Contents

1. Introduction

PixelMan is a Typst library that draws customizable pixel art characters using CeTZ (CeTZ, ein Typst Zeichenpaket). Instead of using external PNG images, characters are rendered as native vector graphics that scale perfectly.

2. Installation

Add to your `typst.toml` dependencies:

```
[dependencies]
pixelman = "0.1.0"
```

Or use locally:

```
#import "@local/pixelman:0.1.0": *
```

3. Quick Start

```
#import "@local/pixelman:0.1.0": *

// Basic usage
#bob() #alice() #eve()

// Custom size
#bob(size: 60pt)

// Custom colors
#alice(skin: brown, hair: red, shirt: blue)
```
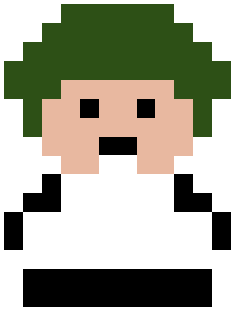
4. Character Gallery

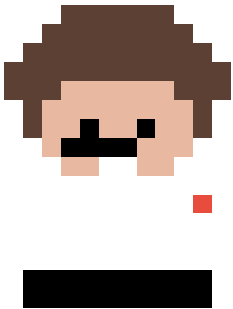## 4.1. Bob

Green hair, white shirt, black vest.


```
#bob()                          // Default
#bob(size: 60pt)                // Larger
#bob(hair: blue, shirt: red)    // Custom colors
```
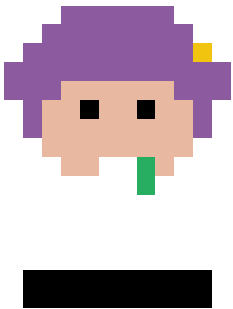
4.2. Alice

Brown hair, white shirt with red pocket, mustache.


```
#alice()
#alice(size: 50pt, hair: black)
```
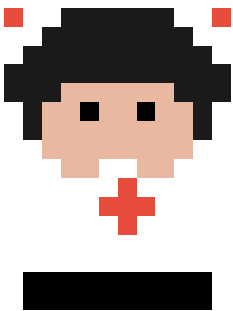
4.3. Christina

Purple hair with yellow accessory, white shirt with green tie.


```
#christina()
#christina(size: 55pt)
```

4.4. Mary

Black hair with red ribbons, white shirt with red bow.

```
#mary()
#mary(size: 45pt, hair: brown)
```

4.5. Eve

Red curly hair, green shirt.
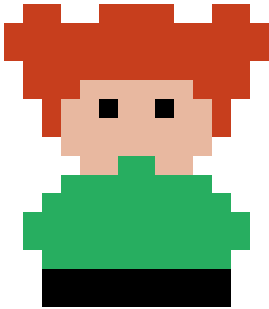

```
#eve()
#eve(size: 50pt, hair: pink)
```

5. All Characters
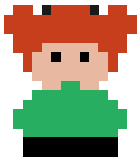
6. Customization

All characters accept these parameters:

- `size`: Overall size (default: 40pt)
- `skin`: Skin color
- `hair`: Hair color
- `shirt`: Shirt color
- `pants`: Pants color

7. Available Colors

Access the color palette:

```
#import "@local/pixelman:0.1.0": palette

#palette.skin        // Default skin tone
#palette.hair-brown  // Brown hair
#palette.shirt-blue  // Blue shirt
// etc.
```

8. Advanced Usage

8.1. Inline Characters

Use in text:

```
#bob() says hello to #alice()!
```

```
Result: (
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  .. (114 items omitted),
) says hello to (
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
```

```
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  (..) => ..,
  .. (114 items omitted),
)!
```

8.2. Different Sizes

```
#bob(size: 20pt)   // Small
#bob(size: 40pt)   // Default
#bob(size: 80pt)   // Large
```
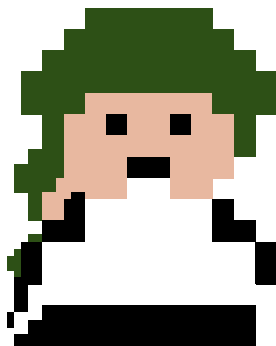

9. API Reference

9.1. Functions

Each character function has the same signature:

```
#let character-name(
  size: 40pt,
  skin: palette.skin,
  hair: palette.hair-color,
  shirt: palette.shirt-color,
  pants: palette.pants-color,
) = { ... }
```

Available characters: bob, alice, christina, mary, eve

10. License
MIT License - See repository for details.
```
#let character-name(
  size: 40pt,
  skin: palette.skin,
  hair: palette.hair-color,
  shirt: palette.shirt-color,
  pants: palette.pants-color,
) = { ... }
```