

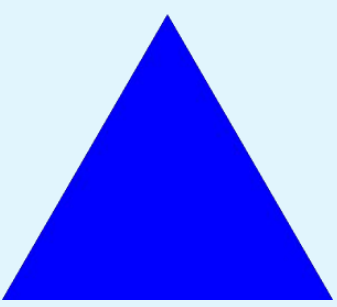


The Toolbox of Functions: Teaching Code Reuse in Schools

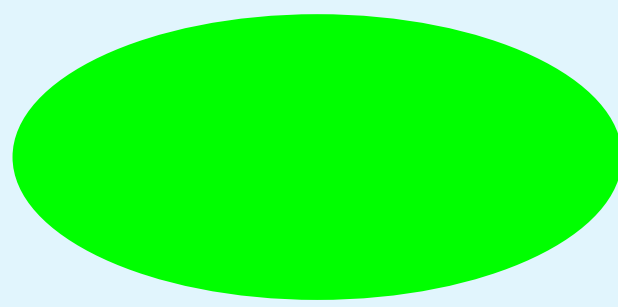
Luca Chiodini, Joey Bevilacqua, Matthias Hauswirth



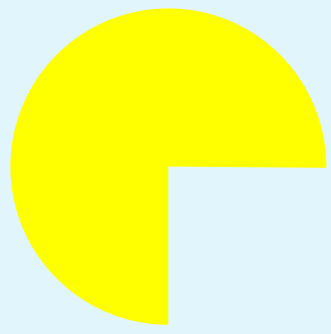
rectangle(200, 100, red)



triangle(100, 100, 60, blue)



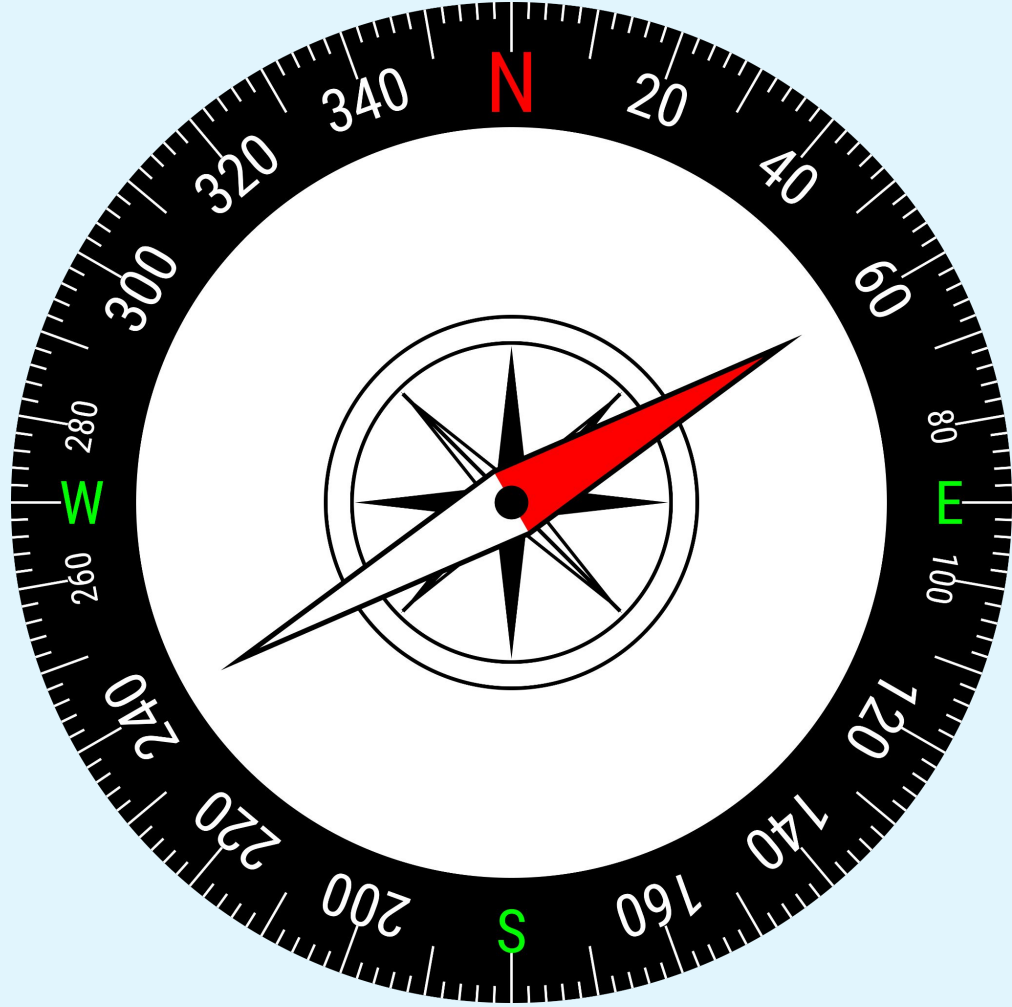
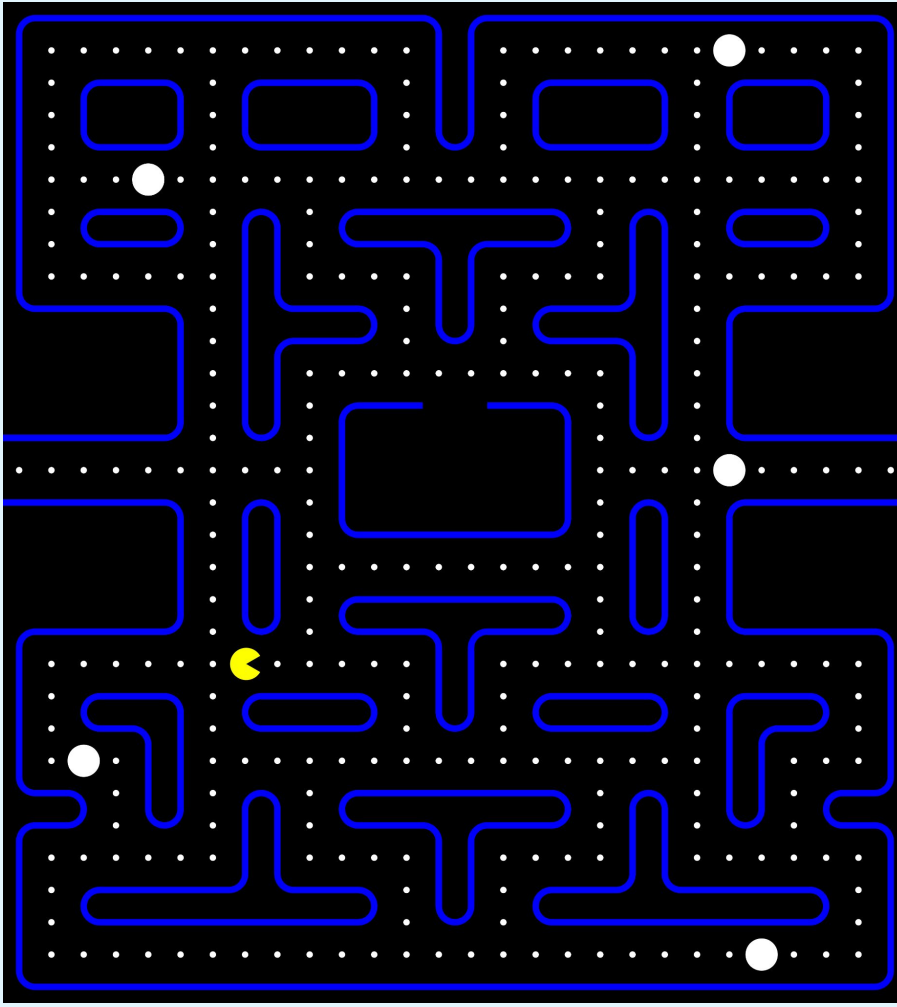
ellipse(300, 150, green)



circular_sector(100, 270, yellow)

text("PyTamaro", "Fira Sans", 20, black)

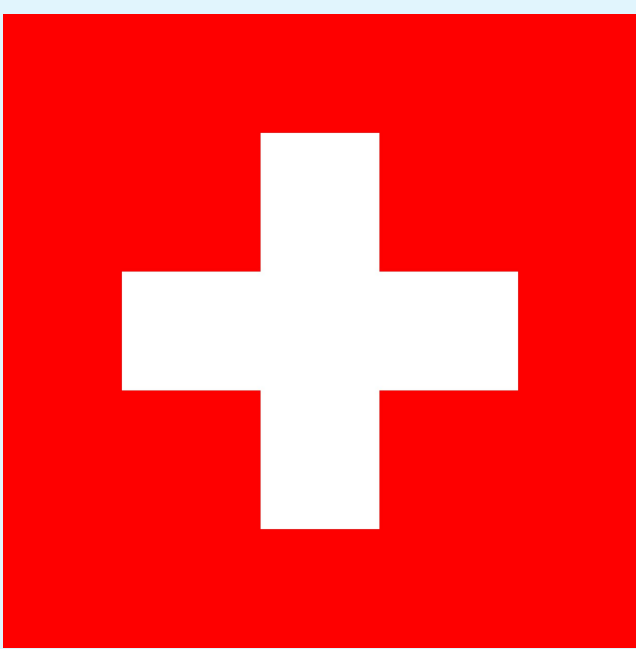
PyTamaro



1 Program the Flags

```
1 from pytamaro import (above, compose, overlay, pin, rectangle, red,
2                       rgb_color, rotate, show_graphic, top_left, white)
3 from funtools import reduce
4
5 # Program the Swiss Flag
6 swiss_cross_arm = rectangle(200, 60, white)
7 swiss_cross = overlay(swiss_cross_arm, rotate(90, swiss_cross_arm))
8 swiss_flag = overlay(swiss_cross, rectangle(320, 320, red))
9 show_graphic(swiss_flag)
10
11 # Program the Greek Flag
12 greek_cross_arm = rectangle(100, 20, white)
13 greek_cross = overlay(greek_cross_arm, rotate(90, greek_cross_arm))
14 blue = rgb_color(0x0D, 0x5E, 0xAF)
15 greek_cross_with_bg = overlay(greek_cross, rectangle(100, 100, blue))
16 blue_stripe = rectangle(270, 20, blue)
17 white_stripe = rectangle(270, 20, white)
18 stripes = reduce(above, [white_stripe, blue_stripe] * 4, blue_stripe)
19 greek_flag = compose(pin(top_left, greek_cross_with_bg),
20                     pin(top_left, stripes))
21 show_graphic(greek_flag)
```

Code Clones



2 Define a Function

```
1 def cross(arm_width: float, arm_height: float, color: Color) -> Graphic:
2     """Colored cross given the horizontal arm width and height"""
3     cross_arm = rectangle(arm_width, arm_height, color)
4     return overlay(cross_arm, rotate(90, cross_arm))
5
6 # Program the Swiss Flag
7 swiss_cross = cross(200, 60, white)
8 ...
9
10 # Program the Greek Flag
11 greek_cross = cross(100, 20, white)
12 ...
```

Abstraction & Code Reuse

Handy in the Future?
Add the Function to the Toolbox Library!

3 Save the Function to Your Toolbox

Implementation

- 📘 Clean up your code before saving a function to your toolbox. Some tips:
- Remove any code not needed by the function you want to save
 - Remove code that produces output (such as calls to `show_graphic` or `print`)
 - Remove unnecessary imports: move all of them to the top, combine them if possible, and delete unused ones

```
1 from pytamaro import overlay, rectangle, rotate, Color, Graphic
2
3 def cross(arm_width: float, arm_height: float, color: Color) -> Graphic:
4     """Colored cross given the horizontal arm width and height"""
5     cross_arm = rectangle(arm_width, arm_height, color)
6     return overlay(cross_arm, rotate(90, cross_arm))
```

Function to save

Function
cross

Colored cross given the horizontal arm width and height

Document what your function does

Example that uses your function

```
1 from toolbox import cross
2 from pytamaro import green, show_graphic
3
4 show_graphic(cross(100, 40, green))
```

DONE ✓



SAVE TO TOOLBOX

4 Explore Your Toolbox

PyTamaro | Activit

Function from library toolbox

cross(arm_width: float, arm_height: float, color: Color) -> Graphic

MY TOOLBOX

- cross
- iso_trapezoid
- circle
- equilateral_triangle
- factorial
- clock
- is_even
- twice
- square

arm_width

arm_height

color

cross

Graphic

Colored cross given the horizontal arm width and height

VIEW DETAILS

Usage

```
1 from toolbox import cross
2 from pytamaro import green, show_graphic
3
4 show_graphic(cross(100, 40, green))
```

DONE ✓

5 Easily Import From Your Toolbox

```
1 from pytamaro import overlay, rgb_color, rotate, show_graphic
2 from toolbox import circle, cross
3
4 red = rgb_color(0xC1, 0x12, 0x1C)
5 blue = rgb_color(0x15, 0x48, 0x89)
6 sign_cross = rotate(45, cross(160, 19, red))
7 sign_base = overlay(circle(69, blue), circle(96, red))
8 sign = overlay(sign_cross, sign_base)
9 show_graphic(sign)
```



1400+

Functions Added to Their Toolbox by 800+ Users

Mostly in Swiss Schools Over a Year

