PAT 204/504 (Fall 2025)

Creative Coding

Lecture 1: Introduction

Instructor: Hao-Wen Dong

UNIVERSITY OF MICHIGAN

Welcome! Tell Us about Yourself!

- Name
- Pronouns
- Program/year
- What is your **most familiar instrument** (if any)?
- Have you ever coded? Which **programming language**?

About Me

- Hao-Wen (**Herman**) Dong
- Pronouns: he/him
- Email: **hwdong**@umich.edu
- Office: **Stearns 131** (15 min walk to the north from Moore)
- Office hours: By appointments (link on the course website)
- Research areas: Generative AI for music, audio, and video

Creative Coding

Creative **Coding**

Creative Coding

What is this course all about?

An introduction to principles and practices of **computer programming for musical applications**. Emphasis is on **creative and artistic uses of code**.

Processing

Max

Learning Objectives

- Gain an understanding of **programming for music and multimedia**
- Learn to **use programming languages** commonly used in artistic projects
- Gain hands-on experience through **implementing audiovisual systems** using creative coding
- Gain a critical comprehension of common concepts and theories employed in creative coding practices

Intro to Processing

What is Processing?

- A free programming language built for electronic arts, new media art and visual design
- Easy to get started with!
- Based on Java
  - 3rd most popular programming language on GitHub
  - Can leverage the power of Java

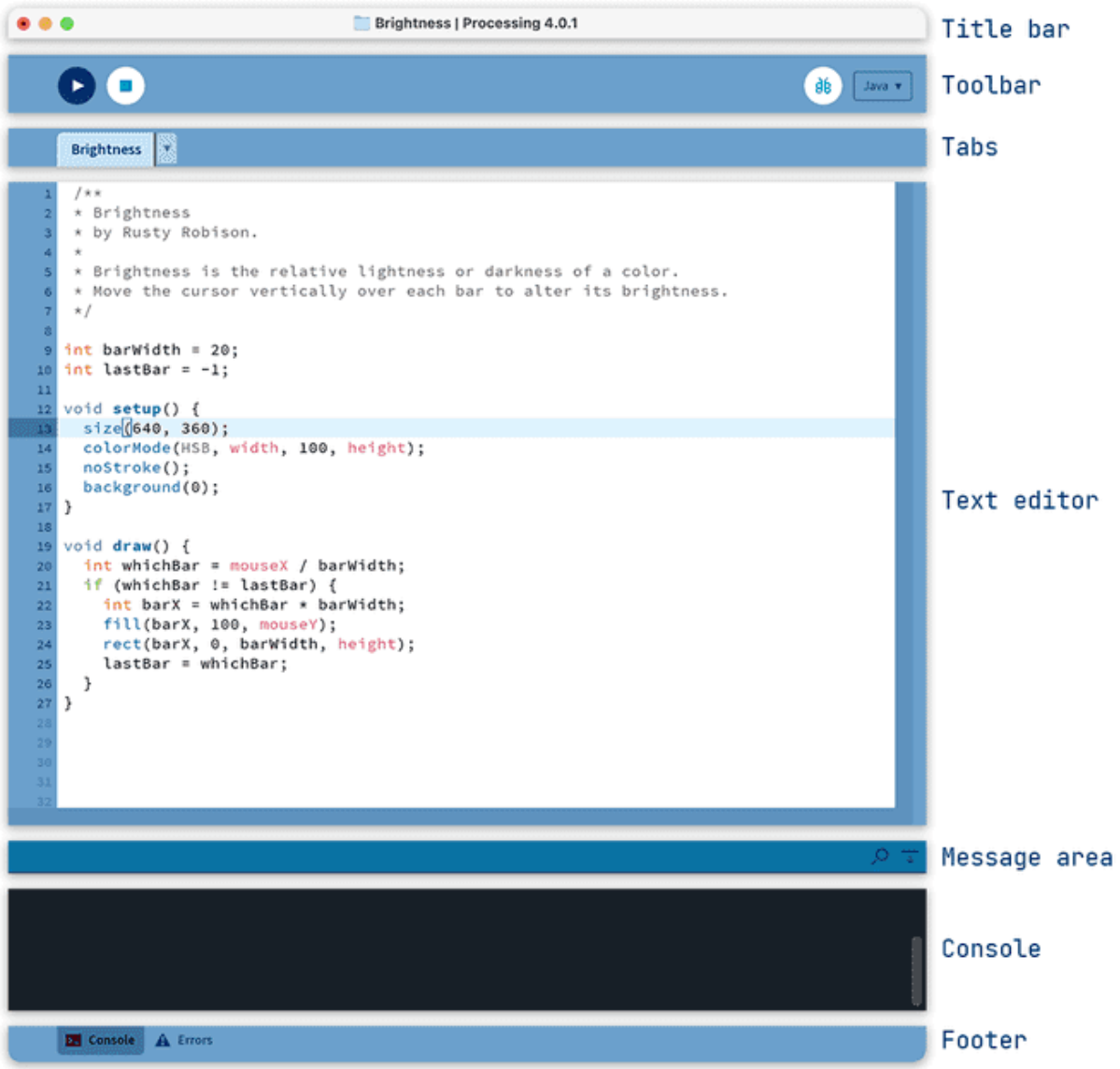
Radiohead – House of Cards (2007)

github.com/dataarts/radiohead

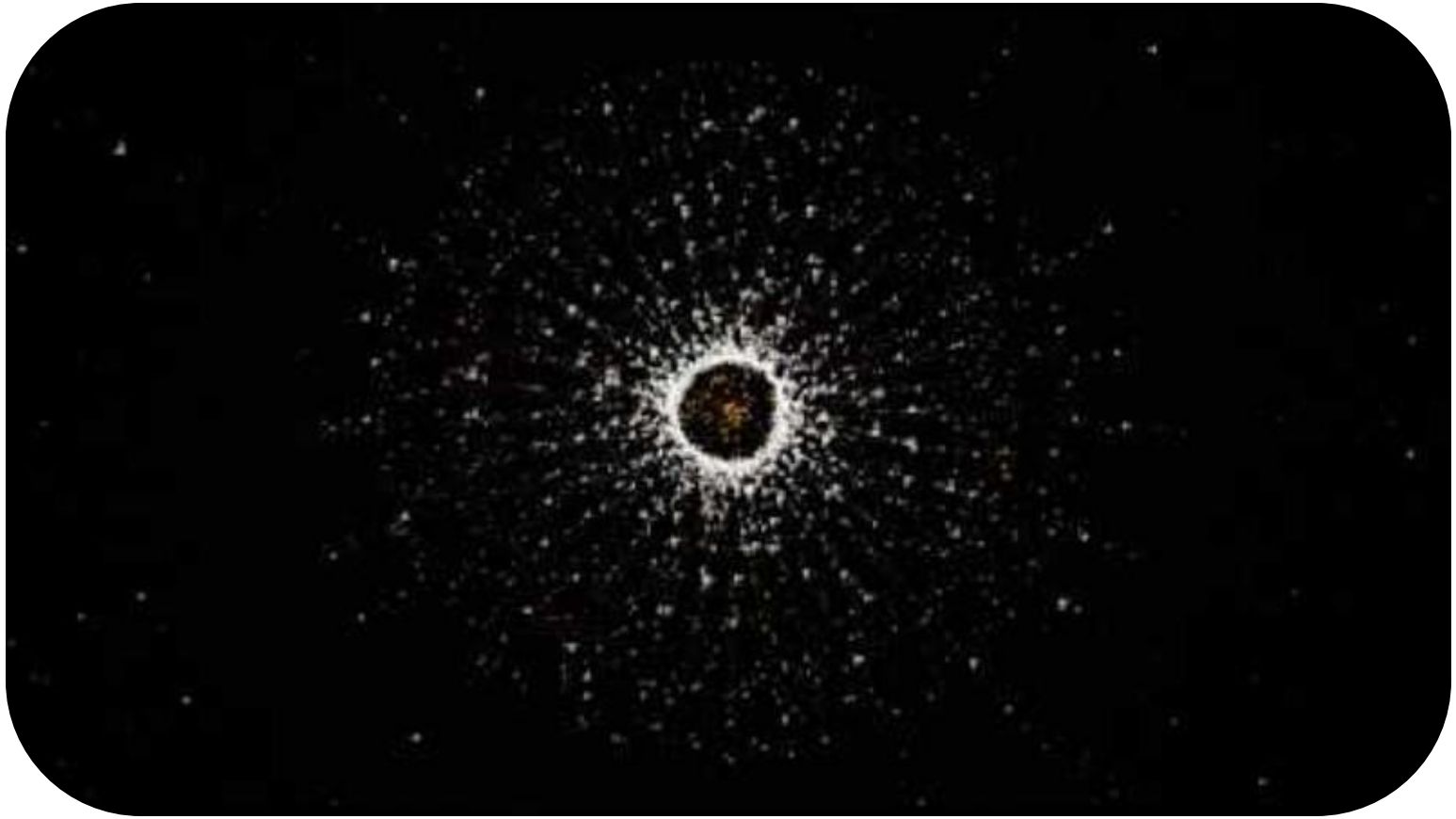
A Processing Sketch

- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window



Music Visualizer



youtu.be/283rmgvFDE0 & pastebin.com/JtAn1mV5

Generative Portraits

youtu.be/ZzNO3FvkTJM

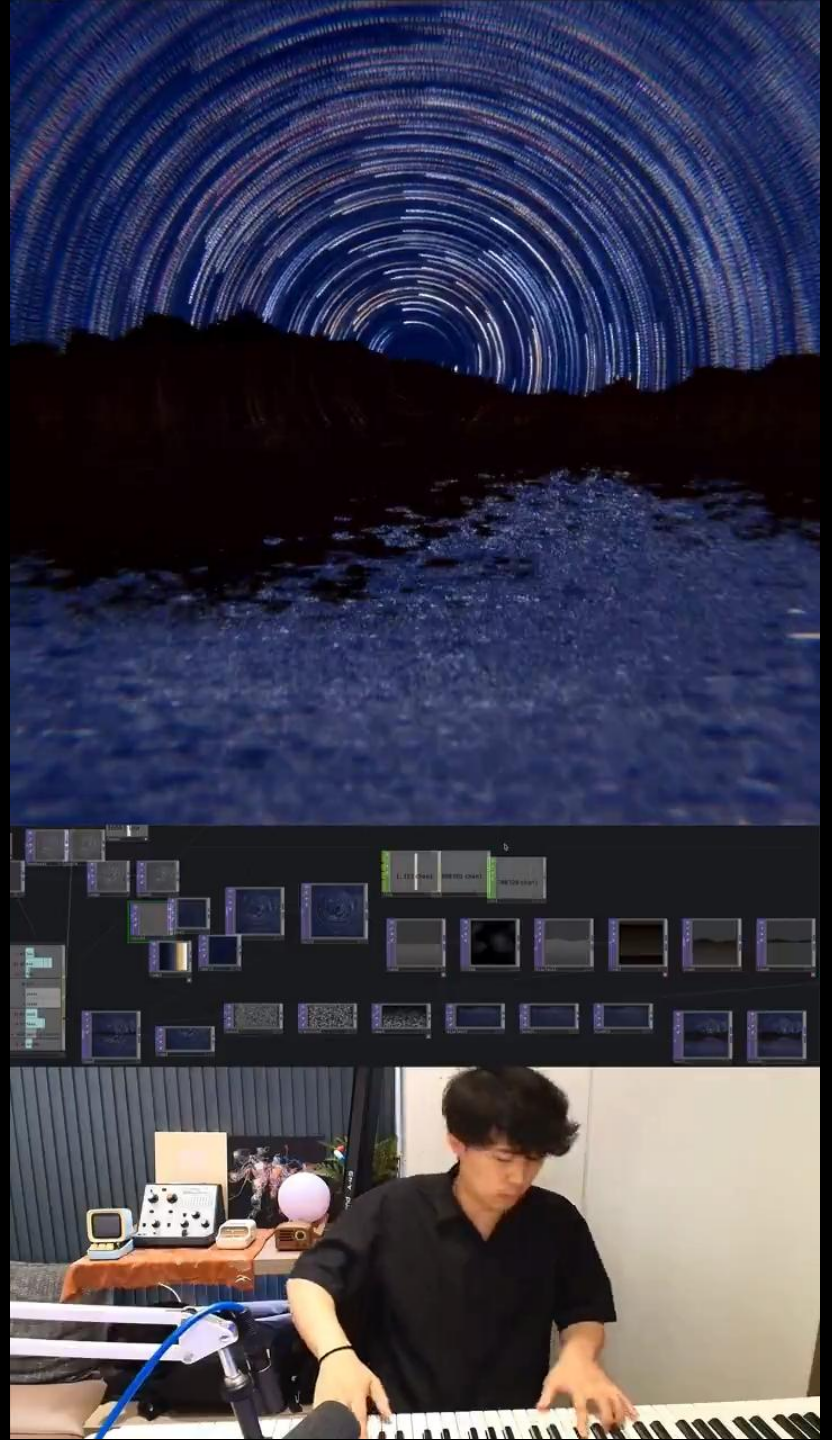
Chaos Dancer by Che-Yu Wu

openprocessing.org/sketch/1217113

Galaxy Reflection by Che-Yu Wu

Created with
TorchDesigner
(not Processing)

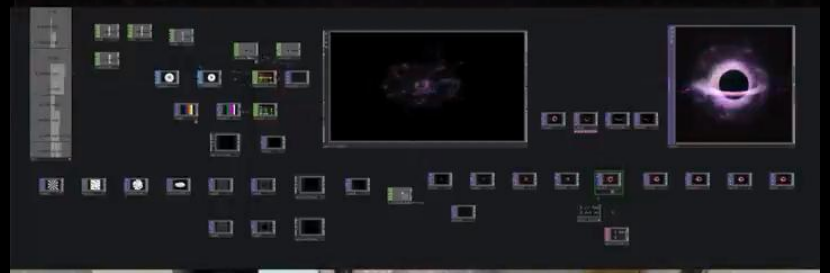
x.com/cheyuwu345/status/1922257576760385681



Cornfield Chase (from “Interstellar”) by Che-Yu Wu

Created with
TorchDesigner
(not Processing)

x.com/cheyuwu345/status/1911768654351671553



Course Logistics

Communications

- **Course website**: Syllabus, schedule, readings, recordings, etc.
- **Email**: Announcements
- **Google Chat**: Q&A

hermandong.com/teaching/pat204_504

Prerequisites

- **None!**
- We'll learn to code in Processing and Max **from scratch**!

Assignments (tentative)

- **Homework**
  - Bouncing “Hello, World!”
  - Paddle ball game
  - Spectrum visualizer
  - MIDI keyboard
  - Polyphonic FM synthesizer
  - Drum machine
- **Midterm assignment**
  - Build your own music visualizer

Assignment Policies

- All assignments must be **completed on your own**
  - You are welcome to exchange ideas with your peers, but this should be in the form of concepts and discussion, not in the form of writing and code.
- **Please turn in your work even if it's incomplete!**
- Due at **11:59pm ET** on the date specified
- Late submissions: **up to a week**, **1 point deducted per day**

Project

- **Open-ended project**
- Group size: 1–3
- **Milestones** (tentative)
  - **Group forming** Oct 29
  - **Pitch** Nov 5
  - **Video presentation** Dec 15
  - **Final report** Dec 15
- Video presentations will be published on the course website
- Deliverables due at **11:59pm ET** on the date specified
- **No late submissions!** Submit your work early and update it later.

Grading

- All **grading** and **regrade requests** will be handled on Gradescope
- **Homework** (**40%**)
- **Midterm assignment** (**20%**)
- **Project** (**40%**)
  - Presentation (20%)
  - Final report (20%)

Resources

- **Processing**
  - Free software with an LGPL license available at processing.org/download
  - **Documentation**: processing.org/reference
  - Official **tutorials**: processing.org/tutorials
  - Official **examples**: processing.org/examples
- **Max**
  - Licensed software with a 30-day trial available at cycling74.com/shop
  - **Documentation**: docs.cycling74.com
- Both Processing and Max are available at the **Music Tech Lab**

Optional Reading

- "*Processing: A Programming Handbook for Visual Designers and Artists*" by Casey Raes and Ben Fry
- "*The Nature of Code*" by Daniel Shiffman
- "*Electronic Music and Sound Design: Theory and Practice with Max/MSP*" by Alessandro Cipriani and Maurizio Giri
- "*The Theory and Technique of Electronic Music*" by Miller Puckette

Policies: Attendance

- In-person attendance is strongly encouraged
- Course lectures will be recorded and made available on the course website
- Please **attend in-person** for the **midterm assignment showcase** and **project pitch**

Policies: Generative AI Usage

- Feel free to use GenAI tools (U-M GPT, ChatGPT, Stable Diffusion, DALLE, etc.) in your workflow. However, **you must disclose your usage of GenAI services in your write-ups**.
- **You take full responsibility for AI-generated materials as if you had produced them yourself**: ideas should be attributed and facts should be true.

Policies: Academic Integrity

- Plagiarism and cheating violate SMTD's Academic Code of Conduct. **All plagiarism, cheating and other academic misconduct cases will be reported to SMTD's Office of Academic and Student Affairs.**
- **All assignments must be completed on your own**. You are welcome to exchange ideas with your peers, but this should be in the form of concepts and discussion, not in the form of writing and code.
- You must **provide proper citations/references for any external resources** you use in your writing and code.

Any Questions on the Syllabus?

hermandong.com/teaching/pat204_504

Let’s Get Started with Processing!

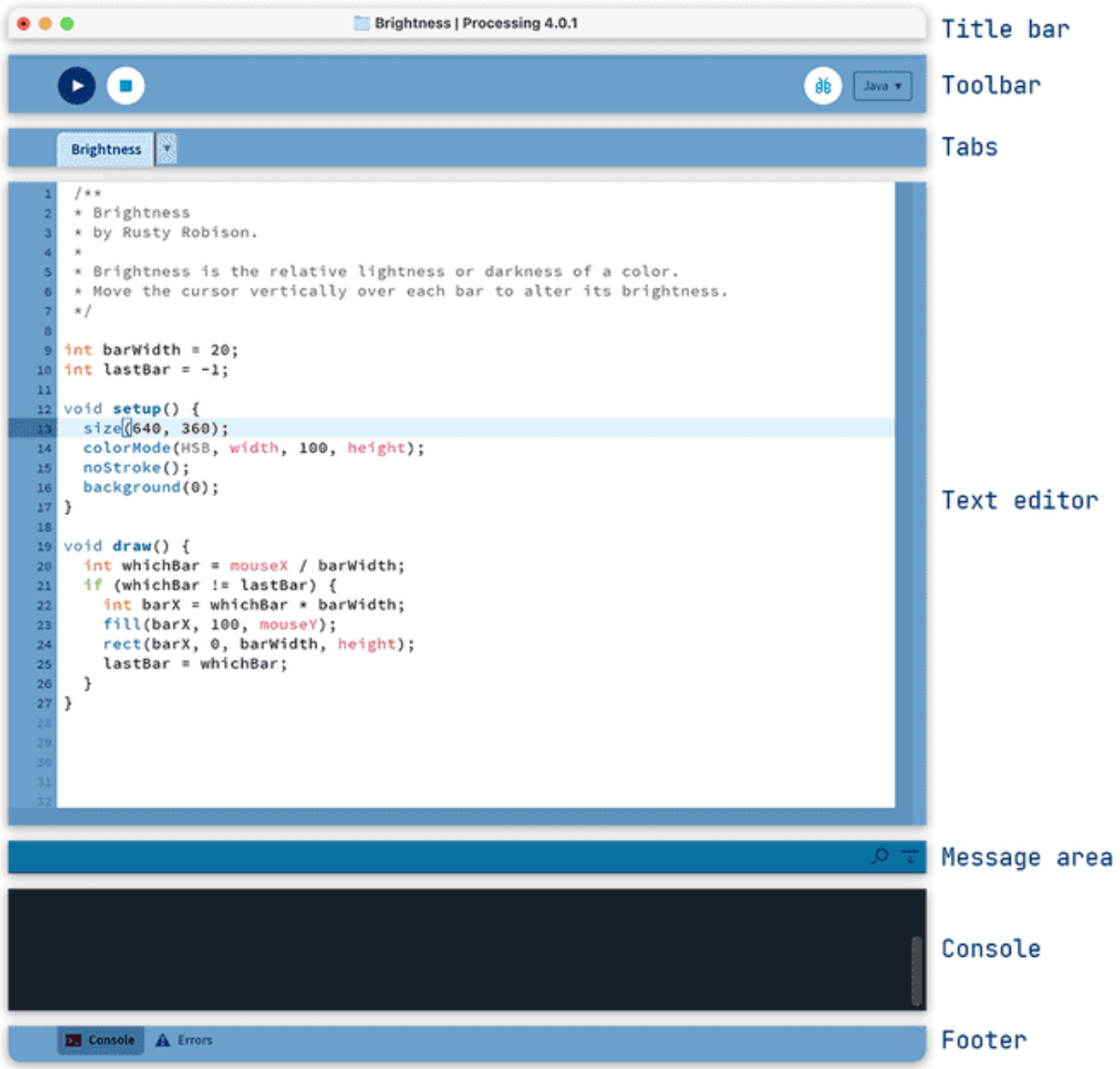
Accessing Processing

- Processing can be downloaded at **processing.org/download**
- Available at the **Music Teach Lab**

A Processing Sketch

- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window



The Canvas

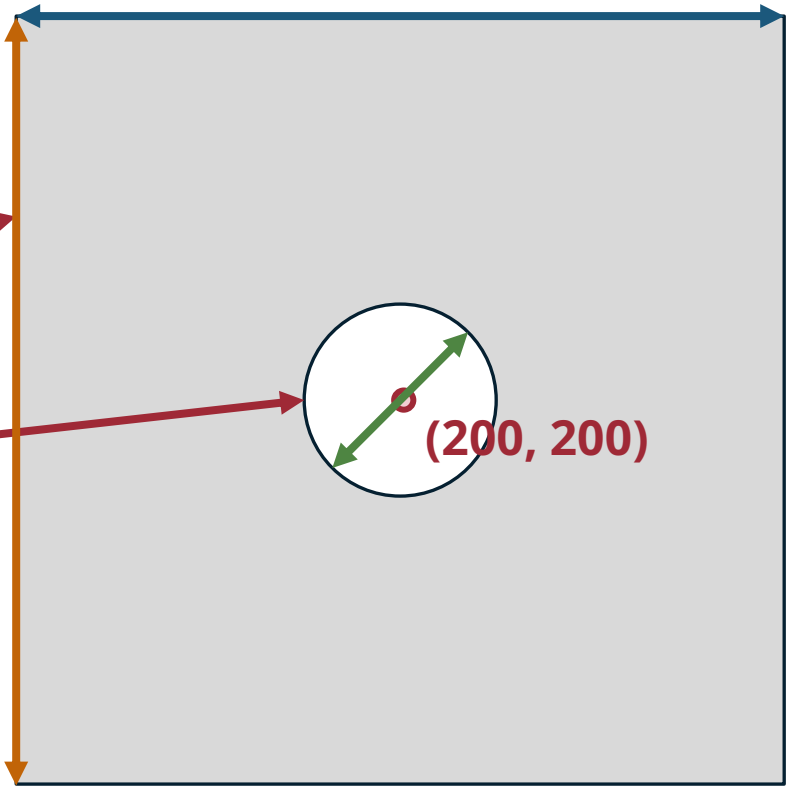
Coordinate System

Our First Processing Sketch

```
size(400, 400);
circle(200, 200, 100);
```

width height

x y diameter



Our First Processing Sketch

Parentheses

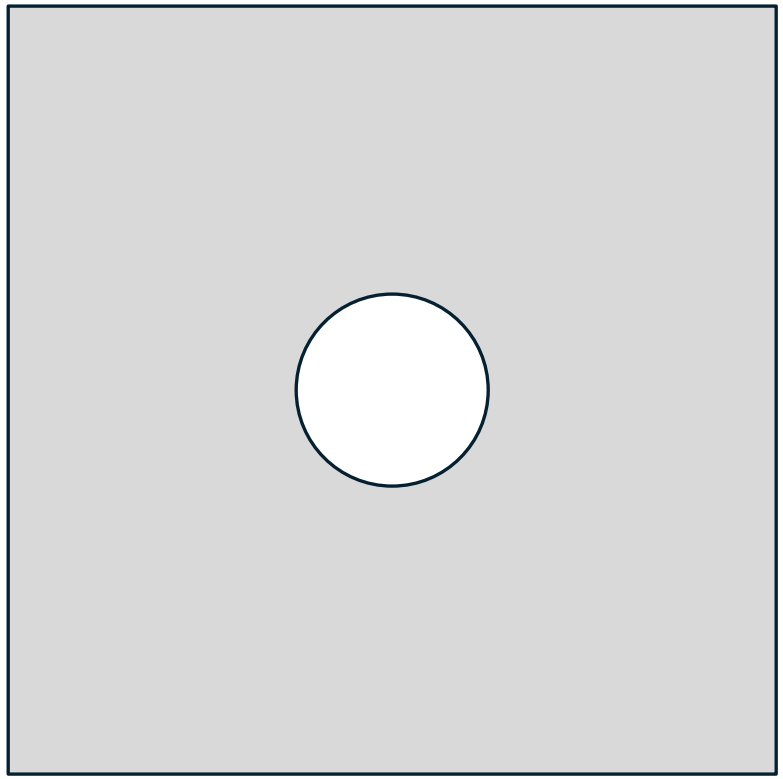
Function name

Arguments (parameters)

```
size(400, 400);
```

Semicolon!

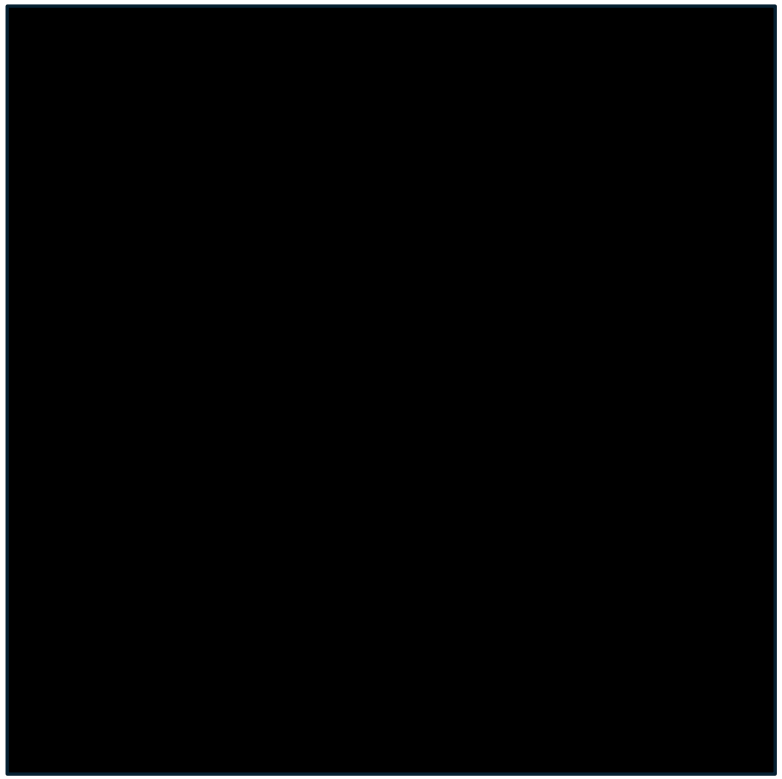
```
circle(200, 200, 100);
```



Colors

Background Color

```
size(400, 400);
background(0);
```

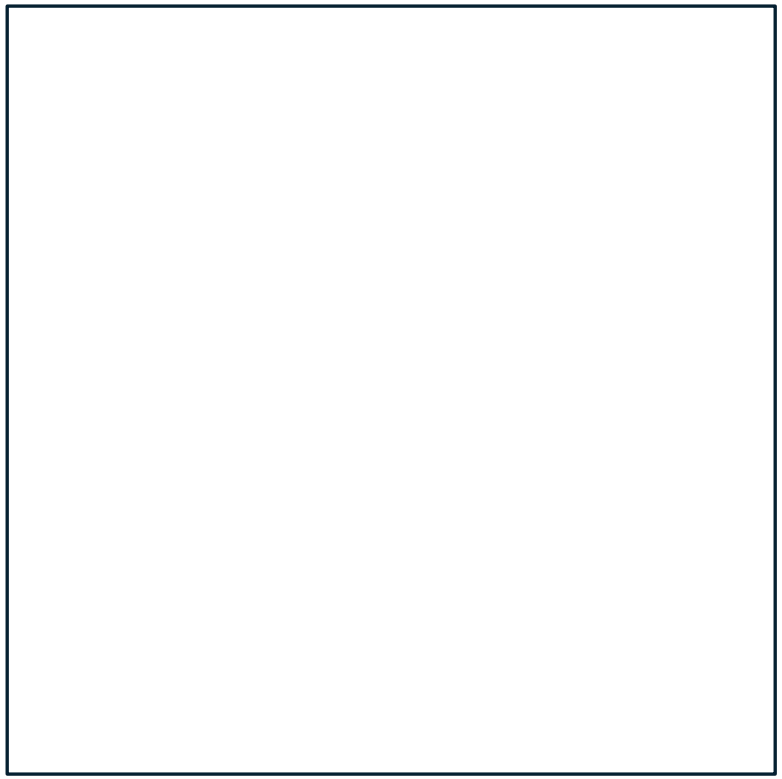


Background Color

```
size(400, 400);
background(255);
```

Range: **0–255**

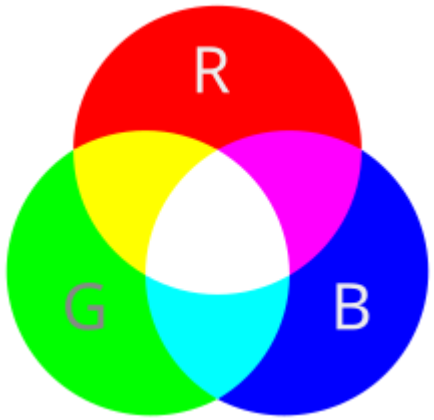
Why?



Background Color

```
size(400, 400);
background(0, 39, 76);
```

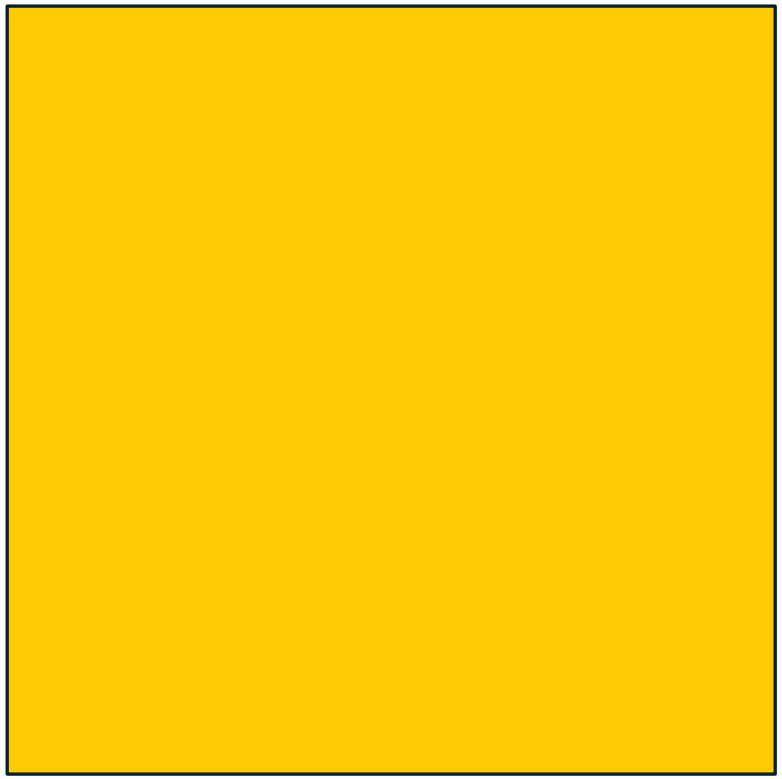
R G B



Background Color

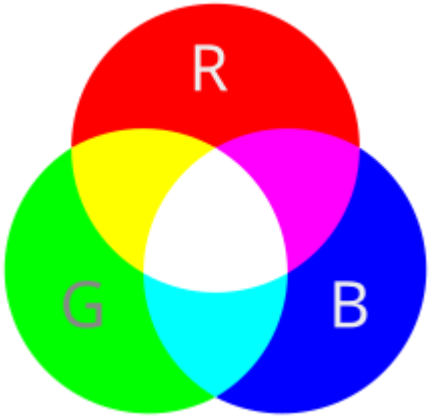
```
size(400, 400);
background(#FFCB05);
```

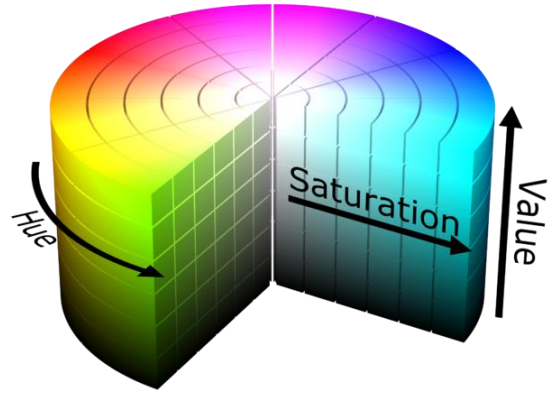
R G B



Colors in Processing

- **Grayscale**: `0–255`
- **RGB color**: `(255, 203, 5)`
- **Hex code**: `#00274C`
- **HSB color**
  - **H**: Hue
  - **S**: Saturation
  - **B**: Brightness
  - Enable with `colorMode(HSB)`



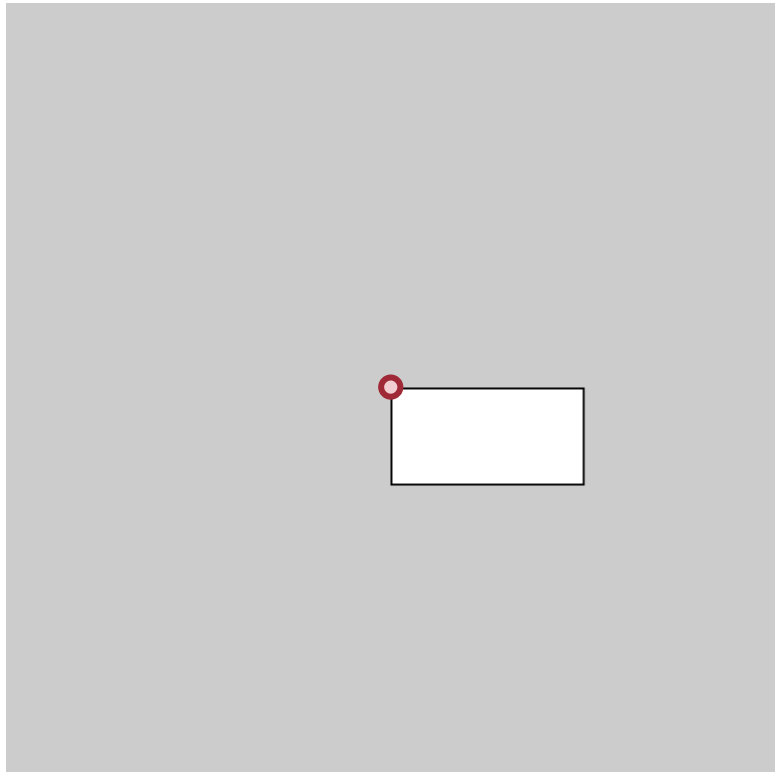


Shapes

Creating a Rectangle

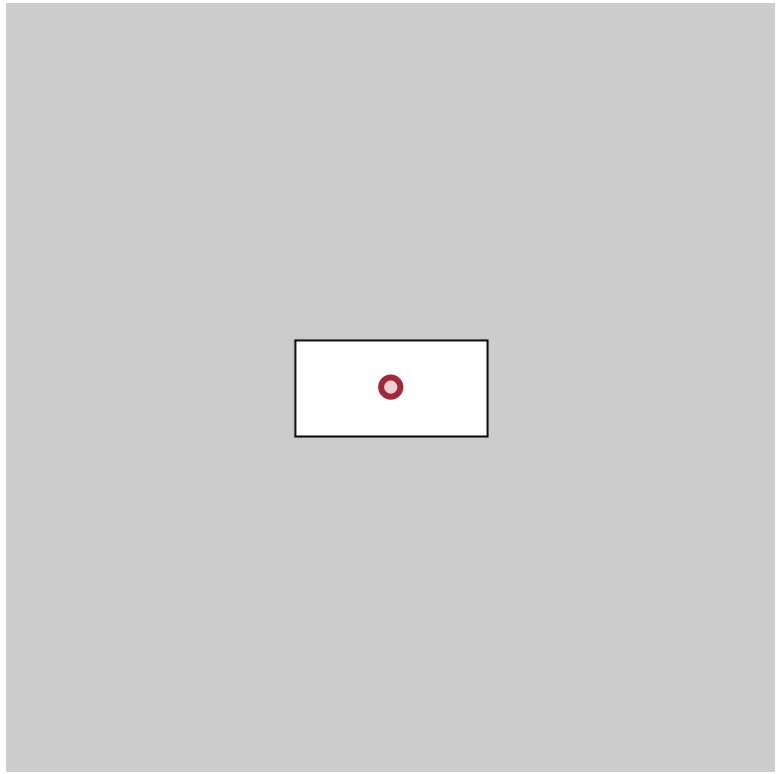
```
size(400, 400);
rect(200, 200, 100, 50);
```

x y width height



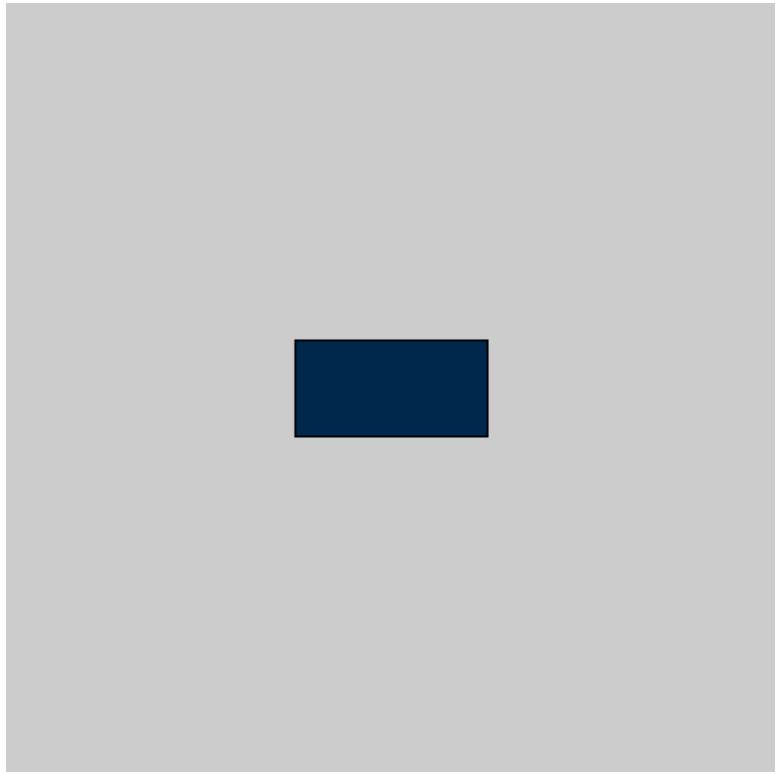
Setting the Anchor Points of Rectangles

```
size(400, 400);
rectMode(CENTER);
rect(200, 200, 100, 50);
```



Changing the Colors of Shapes

```
size(400, 400);
rectMode(CENTER);
fill(#00274C);
rect(200, 200, 100, 50);
```



More Shapes

- Circle `circle(x, y, diameter)`
- Ellipse `ellipse(x, y, width, height)`
- Square `square(x, y, width)`
- Rectangle `rect(x, y, width, height)`
- Point `point(x, y)`
- Line `line(x1, y1, x2, y2)`
- Triangle `triangle(x1, y1, x2, y2, x3, y3)`
- Quadrilateral `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

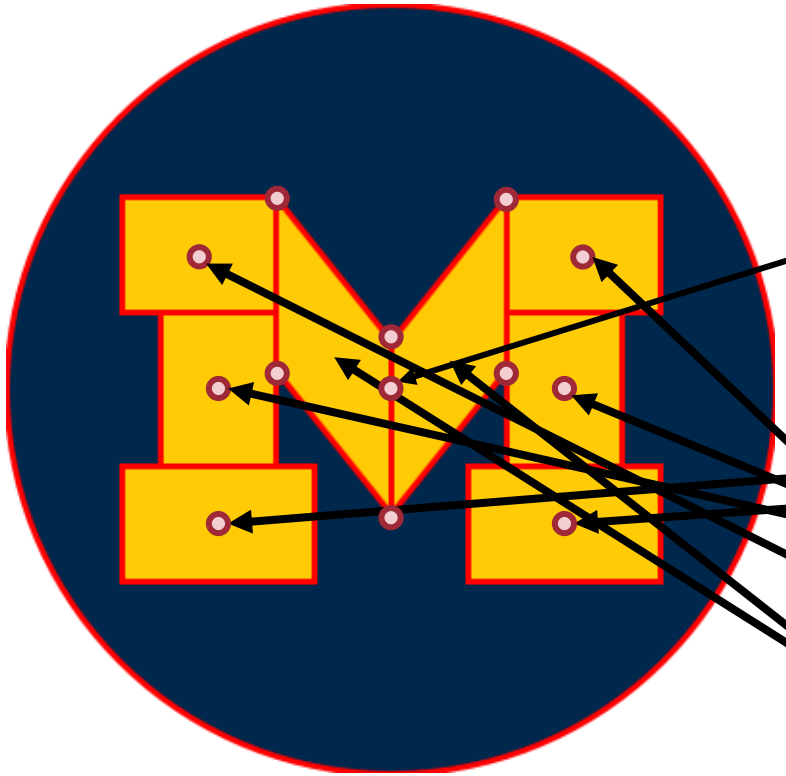
Essential Shortcuts

- **Cmd+R**: Run
- **Esc**: Stop
- **Cmd+Shift+F**: Search in the online reference
- **Cmd+/**: Comment/uncomment
- **Cmd+T**: Auto format

Can you recreate the Block M in Processing?

- Michigan **Blue**: **#00274C**
- Michigan **Maize**: **#FFCB05**

My Version



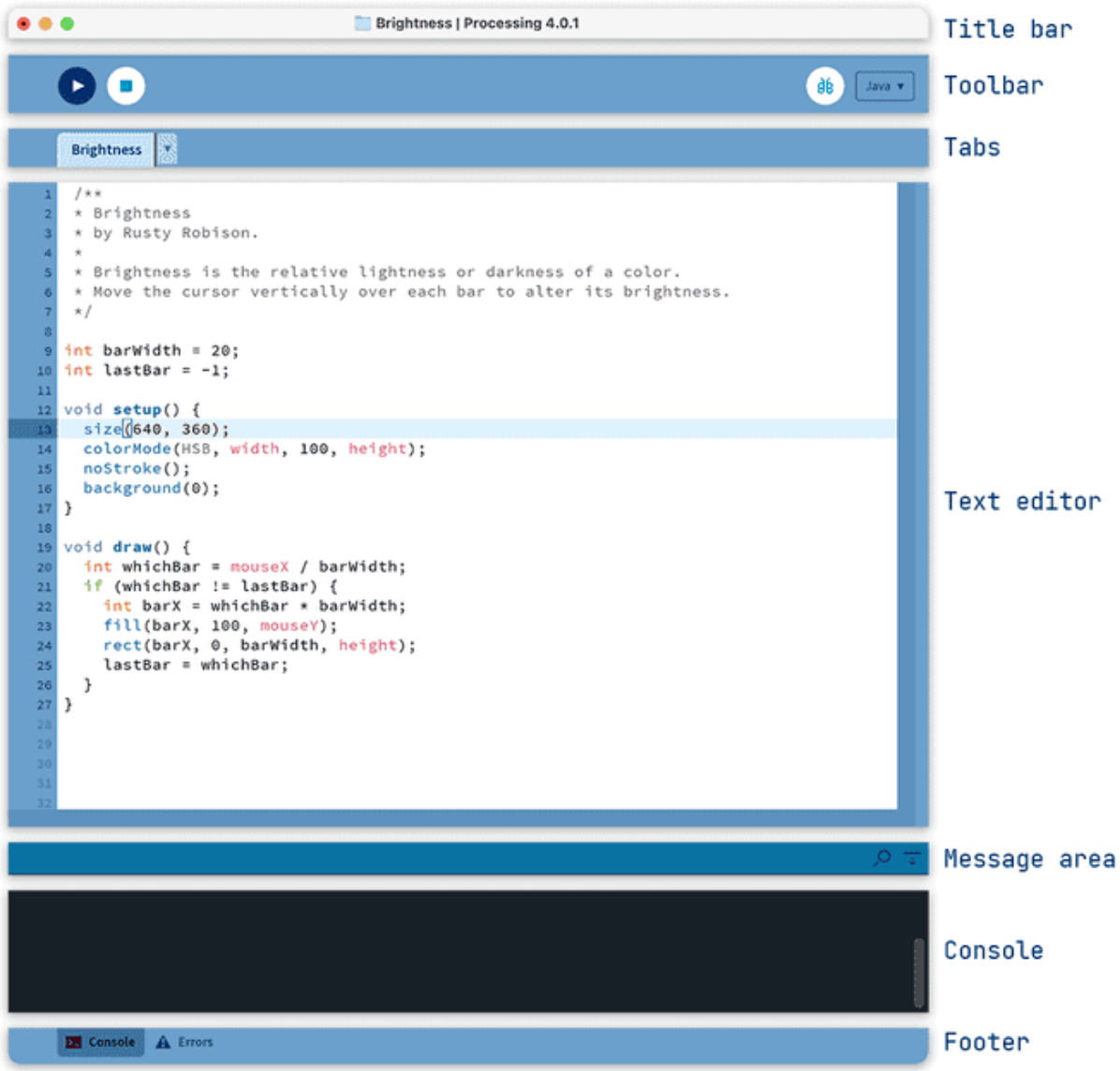
```
void setup() {
  // Create a 400x400 canvas
  size(400, 400);
}

void draw() {
  // Set the background color to white
  background(255);

  // Draw the shapes without outlines
  noStroke();

  // Draw the blue circle at the back
  fill(#00274C);
  circle(200, 200, 400);

  // Set the anchor point of rectangles to the center
  rectMode(CENTER);

  // Set up the yellow text color
  fill(#FFCB05);

  // Draw the feet
  rect(110, 270, 100, 60);
  rect(290, 270, 100, 60);

  // Draw the columns
  rect(110, 210, 60, 150);
  rect(290, 210, 60, 150);

  // Draw the caps
  rect(100, 130, 80, 60);
  rect(300, 130, 80, 60);

  // Draw the "V"
  quad(140, 100, 140, 190, 200, 265, 200, 175);
  quad(260, 100, 260, 190, 200, 265, 200, 175);
}
```

Recap

A Processing Sketch

- Processing comes with an IDE
  (Integrated Development Environment)
  - A **text editor**
  - A **console**
  - A **display window** (when you click the *run* button)

Display window

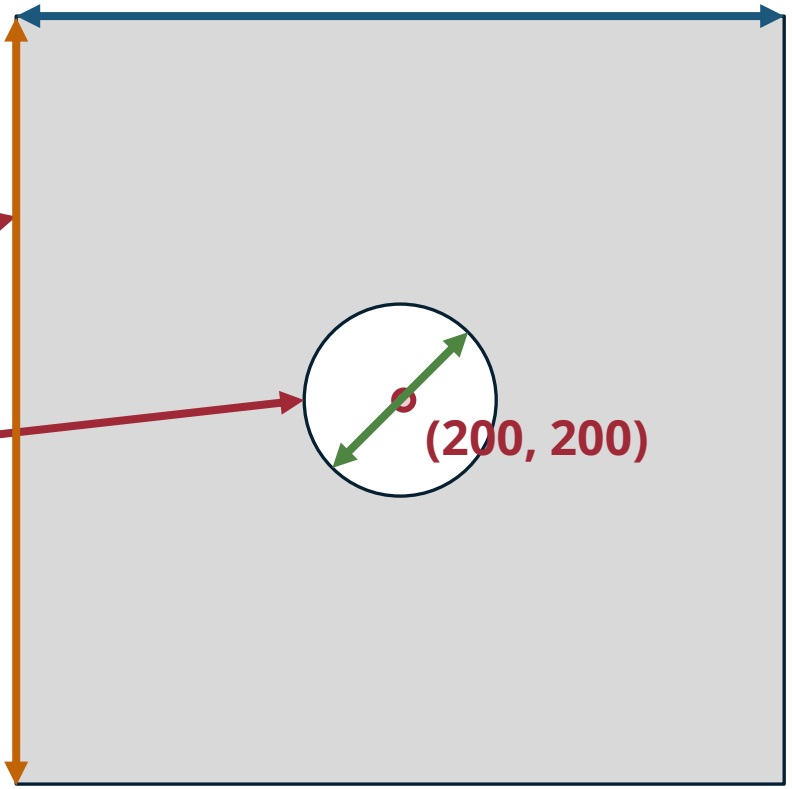


Our First Processing Sketch

```
size(400, 400);
circle(200, 200, 100);
```

width height

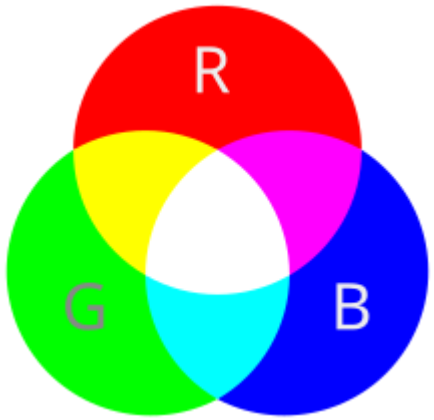
x y diameter



Background Color

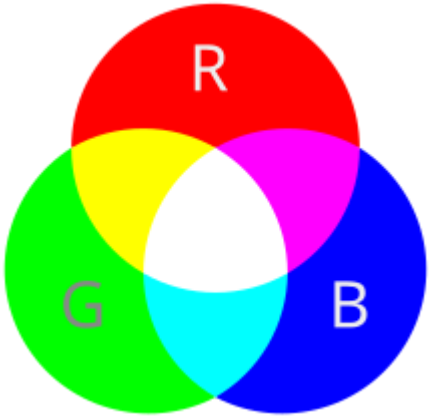
```
size(400, 400);
background(0, 39, 76);
```

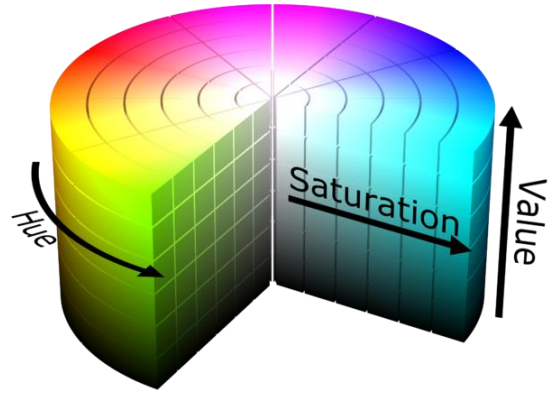
R G B



Colors in Processing

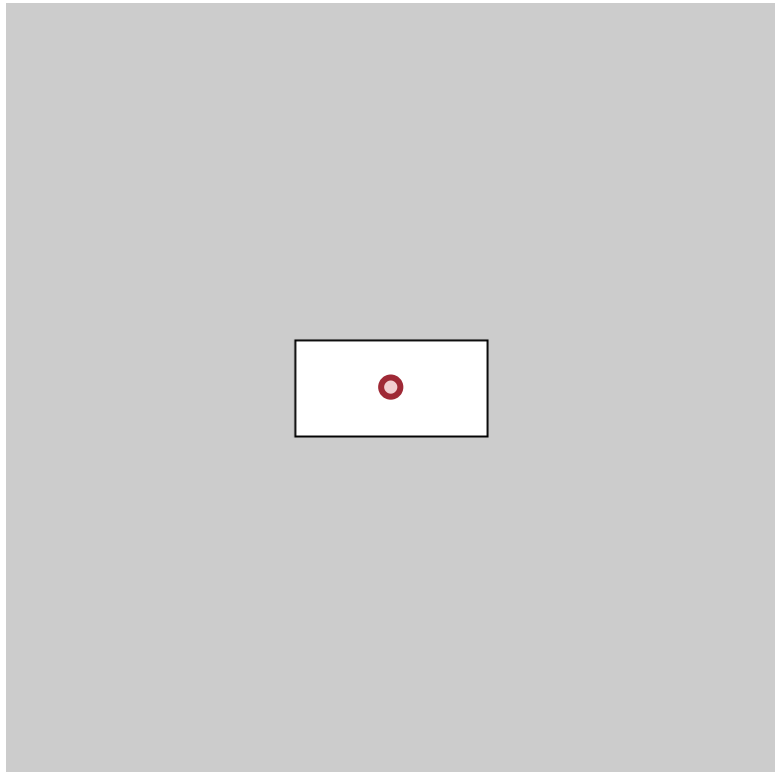
- **Grayscale**: `0–255`
- **RGB color**: `(255, 203, 5)`
- **Hex code**: `#00274C`
- **HSB color**
  - **H**: Hue
  - **S**: Saturation
  - **B**: Brightness
  - Enable with `colorMode(HSB)`





Setting the Anchor Points of Rectangles

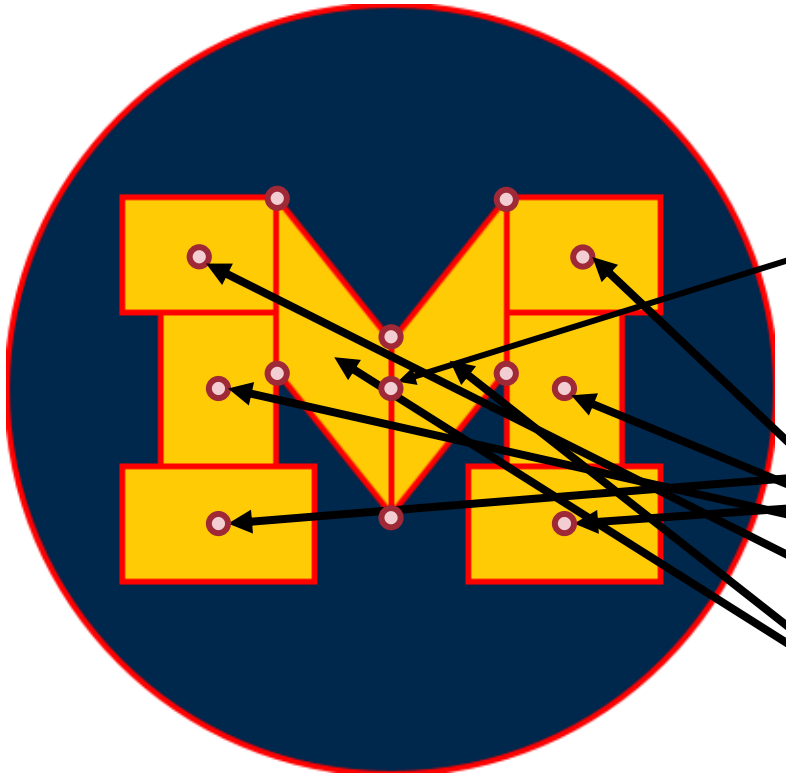
```
size(400, 400);
rectMode(CENTER);
rect(200, 200, 100, 50);
```



More Shapes

- Circle `circle(x, y, diameter)`
- Ellipse `ellipse(x, y, width, height)`
- Square `square(x, y, width)`
- Rectangle `rect(x, y, width, height)`
- Point `point(x, y)`
- Line `line(x1, y1, x2, y2)`
- Triangle `triangle(x1, y1, x2, y2, x3, y3)`
- Quadrilateral `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

My Version



```
void setup() {
  // Create a 400x400 canvas
  size(400, 400);
}

void draw() {
  // Set the background color to white
  background(255);

  // Draw the shapes without outlines
  noStroke();

  // Draw the blue circle at the back
  fill(#00274C);
  circle(200, 200, 400);

  // Set the anchor point of rectangles to the center
  rectMode(CENTER);

  // Set up the yellow text color
  fill(#FFCB05);

  // Draw the feet
  rect(110, 270, 100, 60);
  rect(290, 270, 100, 60);

  // Draw the columns
  rect(110, 210, 60, 150);
  rect(290, 210, 60, 150);

  // Draw the caps
  rect(100, 130, 80, 60);
  rect(300, 130, 80, 60);

  // Draw the "V"
  quad(140, 100, 140, 190, 200, 265, 200, 175);
  quad(260, 100, 260, 190, 200, 265, 200, 175);
}
```

Next Lecture

Processing Basics