

# Creative Coding

Typography

COD 207 - Week 10 Class →



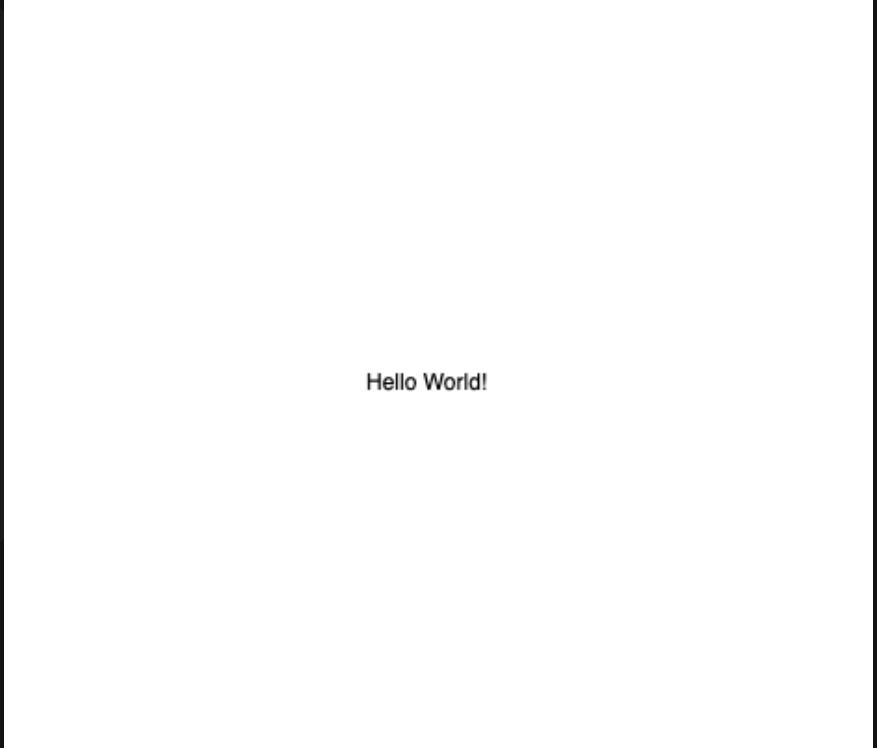
# Table of Contents

- 1. Creative Coding
- 2. Table of Contents
- 3. Display Text on Canvas
- 4. Custom style
- 5. Loading Custom Fonts
- 6. BREAK
- 7. Deconstruct / Reconstruct
- 8. Computational Poster Repository
- 9. Assignments

# Display Text on Canvas

text() function with default parameters

```
1  function setup() {  
2      createCanvas(600, 600);  
3      background(255);  
4  }  
5  
6  function draw() {  
7      background(255);  
8  
9      // Call text function  
10     // three parameters:  
11     // 1) text to display  
12     // 2-3) x,y to position text on the canvas  
13     text("Hello World!", width / 2, height / 2);  
14 }
```



Hello World!

# Custom style

textSize(), textAlign(), fill()

```
1  function setup() {  
2      createCanvas(600, 600);  
3      background(255);  
4  }  
5  
6  function draw() {  
7      background(255);  
8  
9      textSize(42); // Set text font-size  
10     textAlign(CENTER); // Align center  
11     fill(240,200,10); // Set color yellow  
12     text("Hello World!", width / 2, height / 2);  
13  
14     textSize(14); // Sub-header font-size  
15     textAlign(LEFT); // align to left  
16     fill(100); // fill with gray color  
17     text("This is sub title", 190, height / 2 + 15);  
18 }
```



Hello World!  
This is sub title

# Loading Custom Fonts

loadFont(), textFont()

```
1 let fontAkira; // font variable
2
3 function preload() {
4     fontAkira = loadFont("Akira Expanded Demo.otf"); // store font into variable
5 }
6
7 function setup() {
8     createCanvas(800, 800);
9 }
10
11 function draw() {
12     background(248, 249, 231);
13
14     textFont(fontAkira); // set text font defined and loaded on top
15     textSize(122); // set font size
16     textAlign(CENTER); // set font alignment
17
18     for (let i = 0; i < 7; i++) { // draw 7 lines of text underneath each other
19         let x = width * 0.5; // set x position of each text
20         let y = 180 + 88 * i; // set y position of each text
21         let op = map(i, 0, 7, 40, 255); // set the opacity of each font
22     }
23 }
```





# BREAK

10 mins.

# Deconstruct / Reconstruct

- Hansen's method
- Computational Graphic Design Repository → link

Phase	Step	Activity	Material	Domain
Deconstruction	1	<p><b>SELECT</b> Choose a pre-existing graphic design product specimen to be deconstructed from the sample set provided by the course instructor.</p>	Paper and pen	Graphic Design
	2	<p><b>DESCRIBE</b> Make detailed notes about immediately visible visual components, e.g., shapes, typography, colors, scaling, rotation, grids, rhythm, and repetitions.</p>		
	3	<p><b>ANALYZE</b> Identify and formalize invisible components, e.g., interconnections, math, logic and rules required to control the design system and make it behave as desired.</p>		
Reconstruction	4	<p><b>CONVERT</b> Convert notes from steps 2 and 3 into code that replicate the chosen specimen. Use the original specimen as a visual reference to guide and evaluate the process.</p>	Code	
	5	<p><b>EXPLORE</b> Modify the variables in the design system to create alternate versions of the original specimen.</p>		
	6	<p><b>TINKER</b> Modify (and possibly break) the code to create radical mutations of the original specimen.</p>		Computer Science

# Computational Poster Repository



*Figure 2: A selection of specimens suitable as material for the method.*

source ↗



*Figure 3: Poster by Enzo Mari (1963)*

# Assignments

1. Create a generative poster design.
2. Refer to the [pinterest repository](#)
3. Use randomness to create variations of your design. You can randomize colors, fonts, size of objects, position of objects, and so on...
4. Upload 3 different variations as jpg file from the same code.
5.  Submit the openprocessing link.
6.  Submit the sketch source code as zip file as well.