What is programming?

Programming = giving a computer instructions to achieve a specific goal

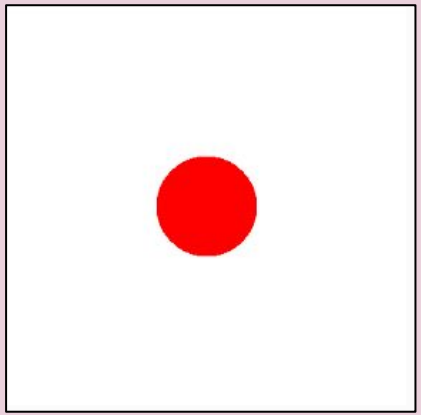
The problem is broken down into singular steps which are executed sequentially. This sequence is called an **algorithm**.

Goal: Draw a red circle in the center of the screen.

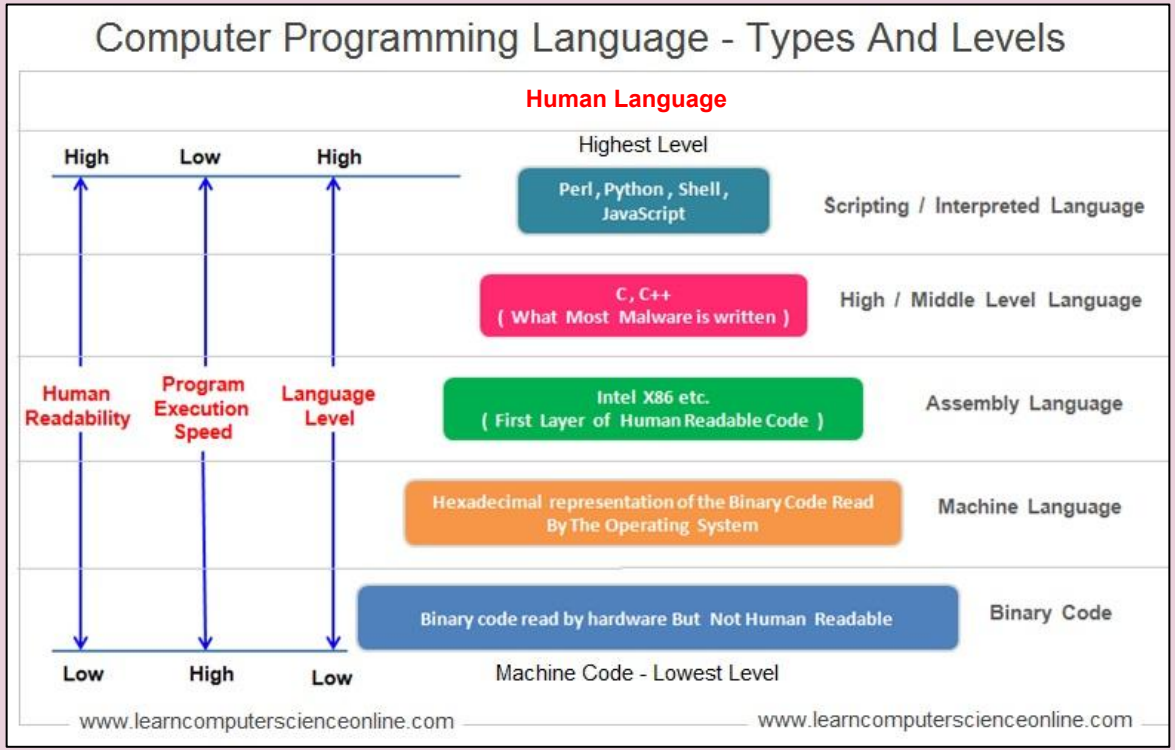
How does the computer know that these 3 lines of code should draw a red circle? (a computer doesn't know the words "circle" or "fill")

→ In using a language the computer understands: a **programming language**

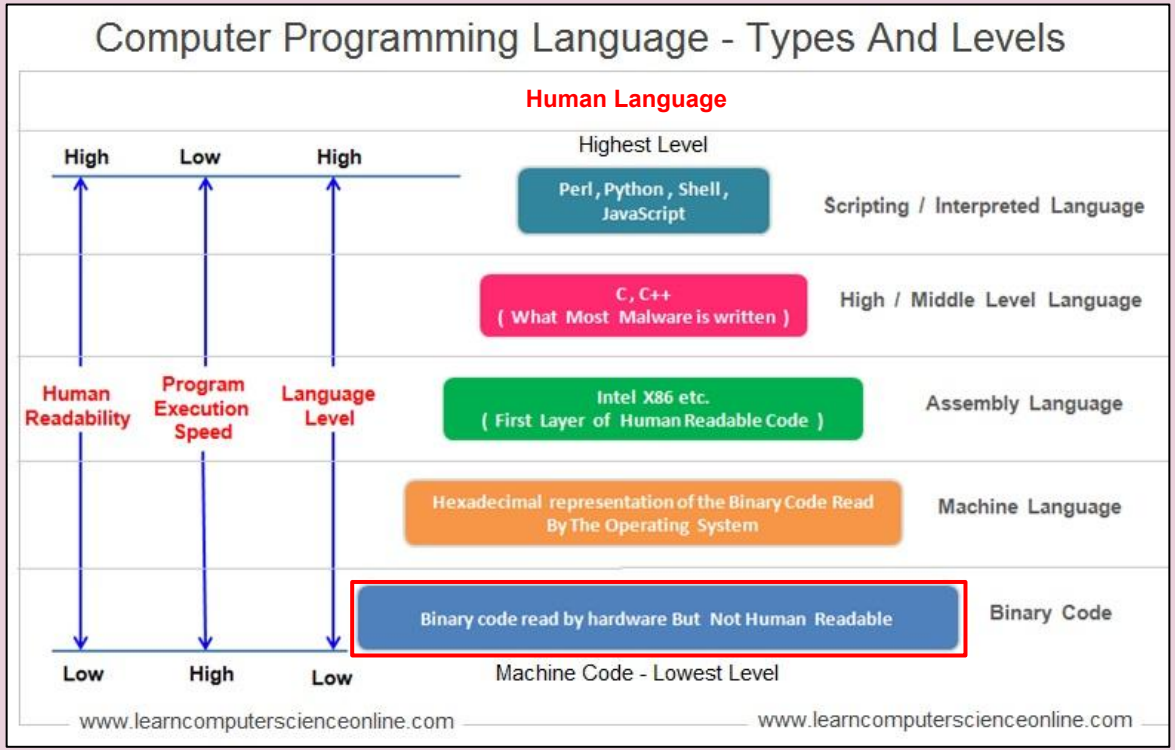
```
void draw() {
    fill(255, 0, 0);
    noStroke();
    ellipse(width/2, height/2, 100, 100);
}
```



Programming languages serve as a **layer of abstraction** between the hardware and a human. Different programming languages are thereby closer to the hardware (less abstracted) or human language (highly abstracted).



Programming languages serve as a **layer of abstraction** between the hardware and a human. Different programming languages are thereby closer to the hardware (less abstracted) or human language (highly abstracted).

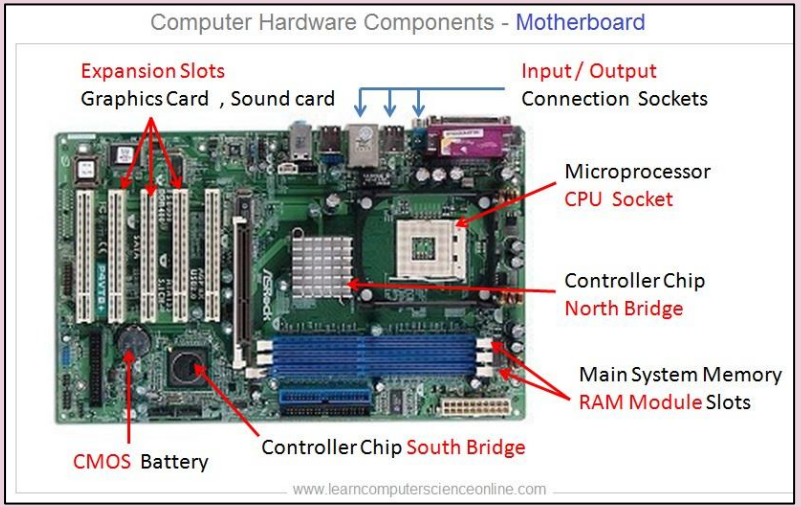


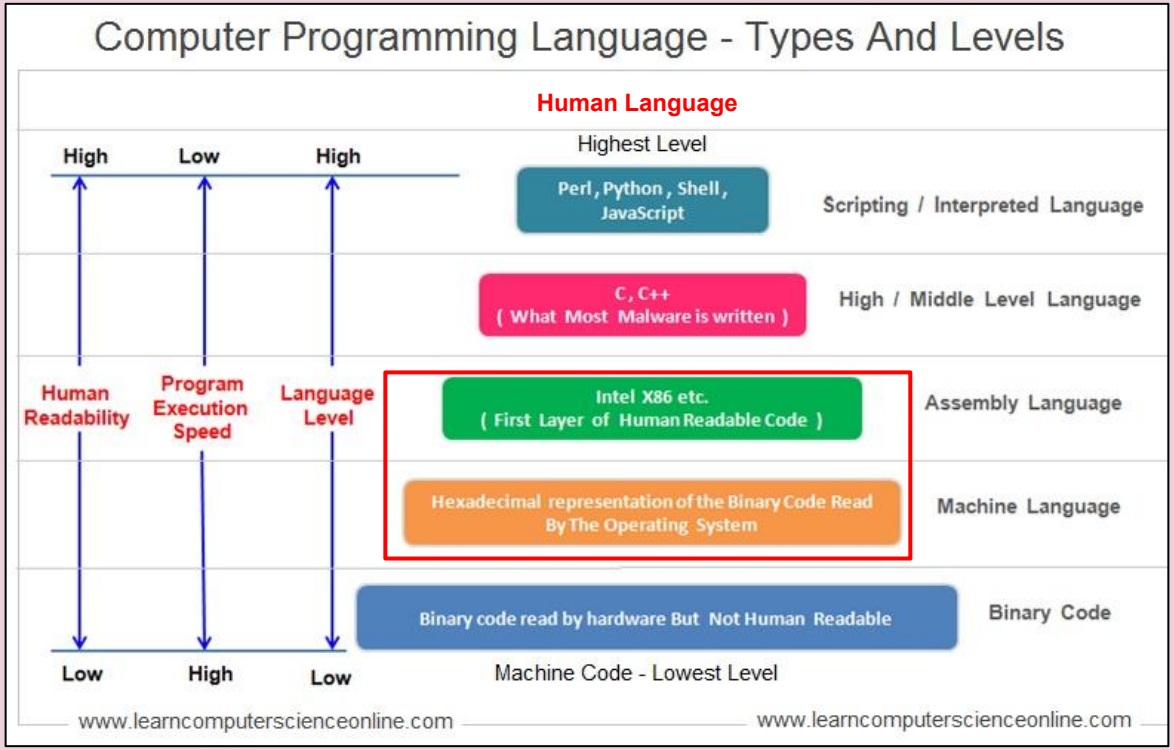
Lowest level (Hardware):

Different components like CPU, memory etc. communicate via voltages. This communication can be represented as **binary code**. Existing voltage is encoded as 1, no voltage as 0.

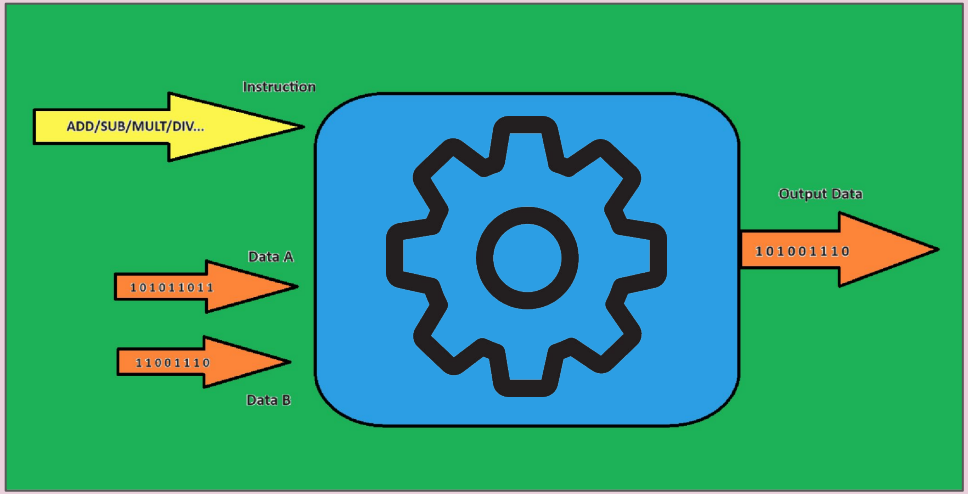
The whole sequence of how component-interactions is a **program**. All the 1s and 0s which are passed from one component to another is the **data**. By passing data around, it is manipulated and finally stored back into **memory**.

Computer = Data storage and data manipulator




Computer Programming Language - Types And Levels
Human Language
Highest Level
High
Low
High
Perl, Python, Shell, JavaScript
Scripting / Interpreted Language
C, C++
(What Most Malware is written)
High / Middle Level Language
Human Readability
Program Execution Speed
Language Level
Intel X86 etc.
(First Layer of Human Readable Code)
Assembly Language
Hexadecimal representation of the Binary Code Read By The Operating System
Machine Language
Binary code read by hardware But Not Human Readable
Binary Code
Low
High
Low
Machine Code - Lowest Level
www.learncomputerscienceonline.com
www.learncomputerscienceonline.com

Normal instructions				Compressed	
00000	SUB	10000	CMP	000	SUB
00001	AND	10001	TEST	001	AND
00010	ADD	10010	LW	010	ADD
00011	OR	10011	SW	011	CMP
00100	XOR	10100	LH	100	LW
00101	LSR	10101	SH	101	SW
00110	LSL	10110	LB	110	LDI
00111	ASR	10111	SB	111	MOV
01000	BREV	11000	LDI	Reserved for FPU	
01001	LDILO	11001			
01010	MPYUHI	Special Insn		11010	FPADD
01011	MPYSHI			11011	FPSUB
01100	MPY	11100	BREAK	11100	FPMPY
01101	MOV	11101	LOCK	11101	FPDIV
01110	DIVU	11110	SIM	11110	FPI2F
01111	DIVS	11111	NOOP	11111	FPF2I

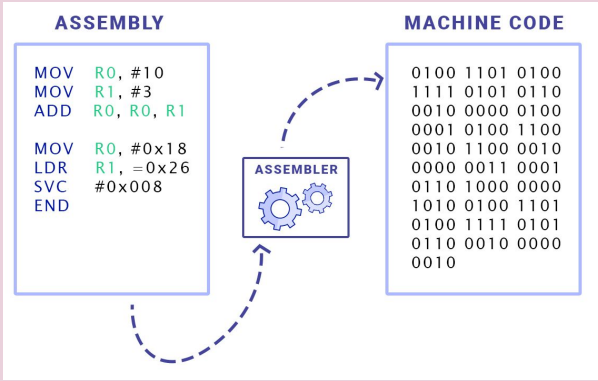


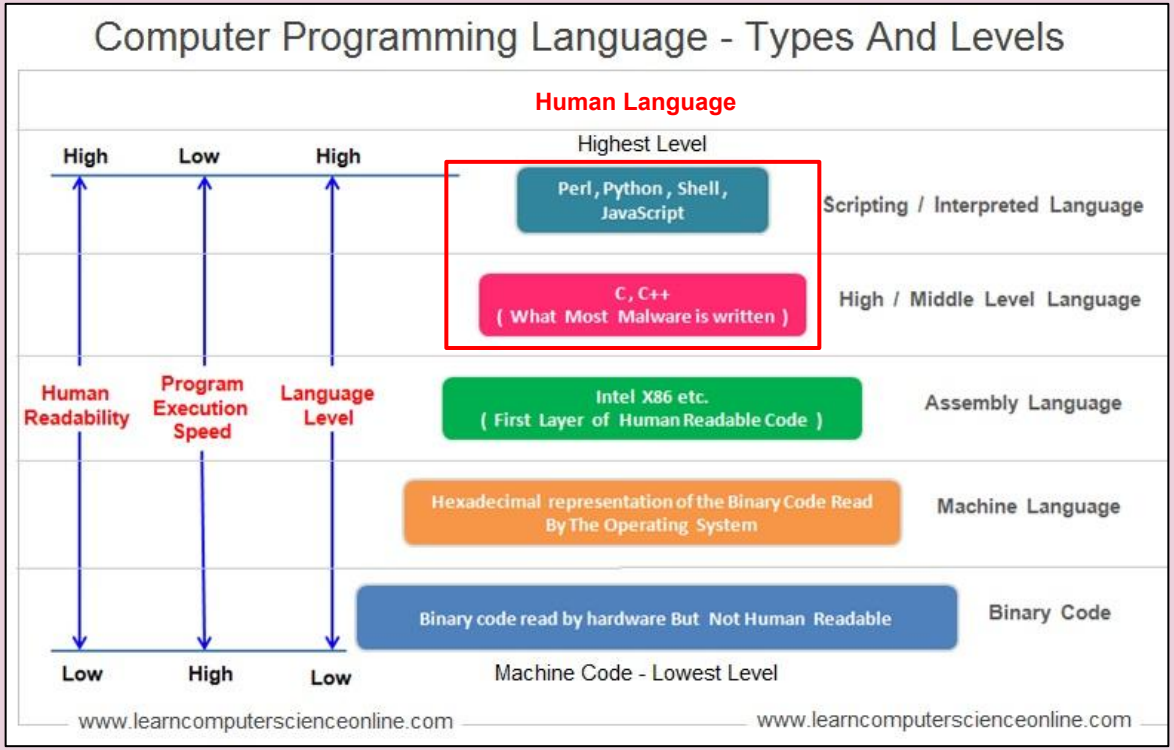
Second level (machine code):

Which operations are executed on the data is controlled via different control commands.

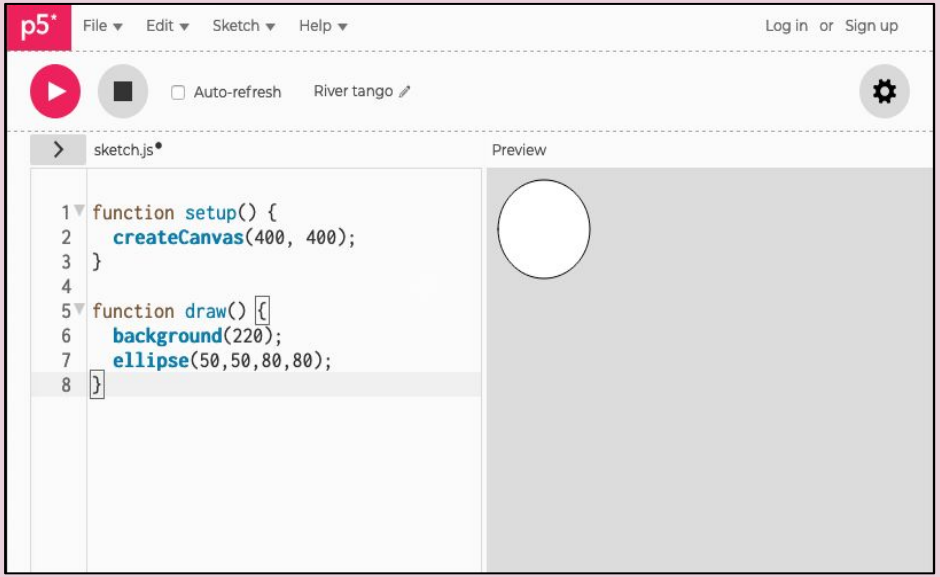
Third level (Assembly):

Human-readable representation of machine code.




Computer Programming Language - Types And Levels
Human Language
Highest Level
High
Low
High
Perl, Python, Shell, JavaScript
Scripting / Interpreted Language
C, C++
(What Most Malware is written)
High / Middle Level Language
Human Readability
Program Execution Speed
Language Level
Intel X86 etc.
(First Layer of Human Readable Code)
Assembly Language
Hexadecimal representation of the Binary Code Read By The Operating System
Machine Language
Binary code read by hardware But Not Human Readable
Binary Code
Low
High
Low
Machine Code - Lowest Level
www.learncomputerscienceonline.com
www.learncomputerscienceonline.com

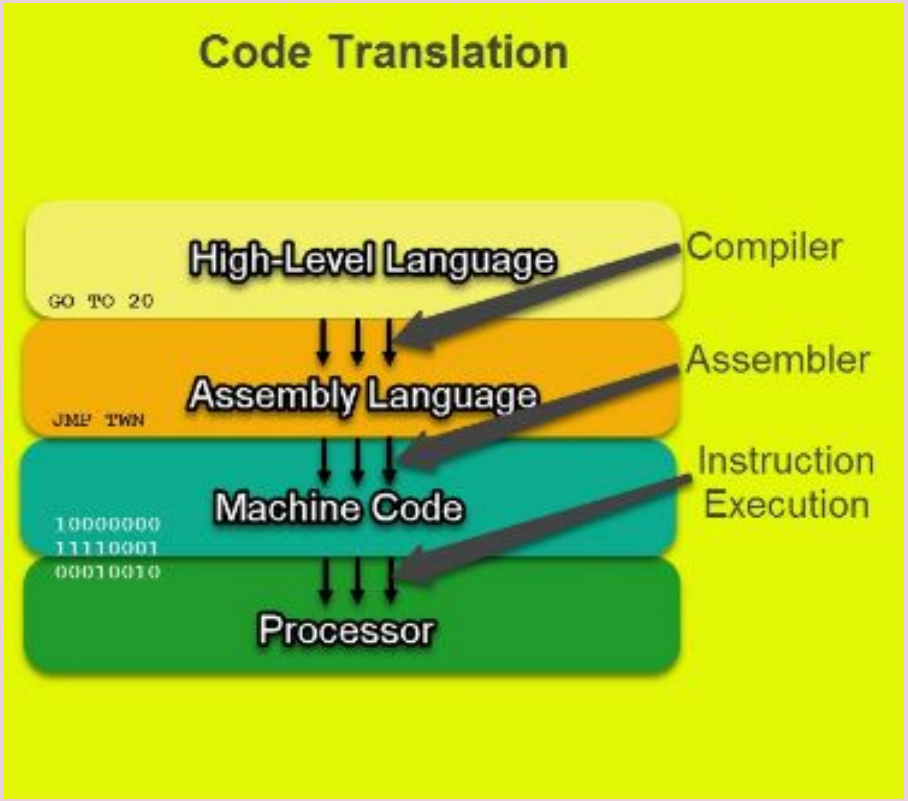
```c
#include <stdio.h>
#include <stdlib.h>
int main(){
    int *ptr =NULL;
    int  n =5;
    ptr = (int*)malloc(sizeof(int)*5);  //memory allocated for 20 bytes
    if(ptr == NULL)
    {
        printf("Failed to allocated");
    }
    else
    {
        for(int i=0;i<n;i++)
        {
            *(ptr + i) = i;
        }
    }
    for(int i=0;i<n;i++)
    {
        printf("%d ",*ptr+i);
    }
    free(ptr);
    return 0;
}
```



Fourth/Fifth level (high-level languages):

"Easily" readable. Code highly abstracted from the hardware.

In using different steps, the code is translated back to the hardware level.



*“p5.js is a JavaScript **library** for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else!”*

→ highly abstracted, therefore easy to learn

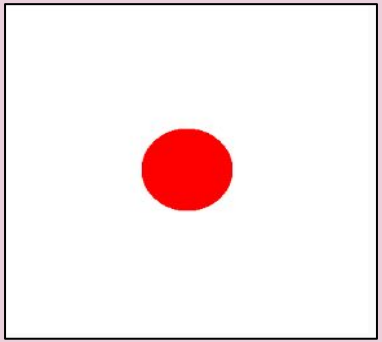
→ interactive learning experience as code directly produces visible results

```java
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Graphics;
import java.awt.Color;
import java.awt.event.MouseMotionListener;

public class HoverCircle extends JFrame {
    public HoverCircle() {
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Hover to Draw Circle");
        JPanel panel = new JPanel() {
            int x = -50;
            int y = -50;
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                g.setColor(Color.RED);
                g.fillOval(x - 50, y - 50, 100, 100);
            }
            {
                addMouseMotionListener(new MouseMotionListener() {
                    @Override
                    public void mouseMoved(java.awt.event.MouseEvent e) {
                        x = e.getX();
                        y = e.getY();
                        repaint();
                    }
                    @Override
                    public void mouseDragged(java.awt.event.MouseEvent e) {}
                });
            }
        };
        add(panel);
        setVisible(true);
    }
    public static void main(String[] args) {
        new HoverCircle();
    }
}
```

```processing
void setup() {
  size(400, 400);
  background(255);
}

void draw() {
  background(255);
  fill(255, 0, 0);
  ellipse(mouseX, mouseY, 100, 100);
}
```



- Webeditor
- functions z.B. createCanvas(), background()
- setup() vs. draw()
- draw primitive shapes, fill+color, stylings
- variables, built-in variables (mouseX/mouseY, width/height)
- random(), map()
- mouseClicked(), keyPressed()
- if/else statements