Topic 1 Your development environment

Created	@October 10, 2022 1:33 PM
Class	CM1005-Introduction-to-Programming-I
Type	Lecture
Materials	
Reviewed	☐
URL	https://www.coursera.org/learn/uol-introduction-to-programming-1/home/week/1
Week	22 Oct Week 1

1.0 Introduction

Tutors:

Simon

Edward

Growth mindset.

Technical tools:

- P5.JS
- Brackets
- JavaScript programming language

Other students are recommending the VS Code instead of Brackets with a P5.VSCode and a Live Server extension, so I will be using that throughout the programme.

1.1 Introduction to P5.JS

Commands

> Smallest building blocks of a program, a single instruction for a computer to perform a task.

Program

> Collection of commands put together in the right order to solve a problem, complete a task, or user interaction.
>
> Also referred as Source Code or just code.

A **code editor** helps with writing a problem by *making suggestions and spotting errors*.

Running a program is called *executing*.

Programming language is a dictionary of commands that a language provides.

Interpreter translates them to *binary instructions* that a specific hardware understands.

We will use **JavaScript** with the P5.JS library to avoid reinventing the wheel.

P5.js homepage

https://p5js.org/

P5.js reference page

https://p5js.org/reference/

P5.js examples

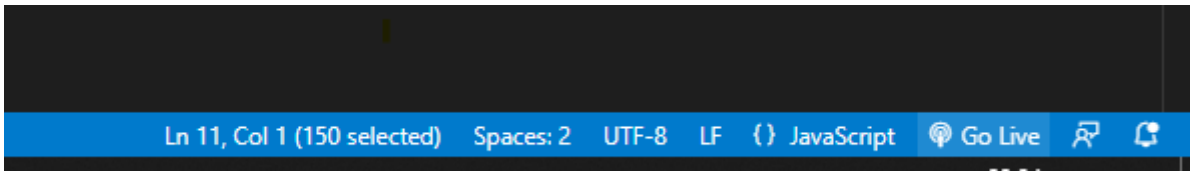
https://p5js.org/example

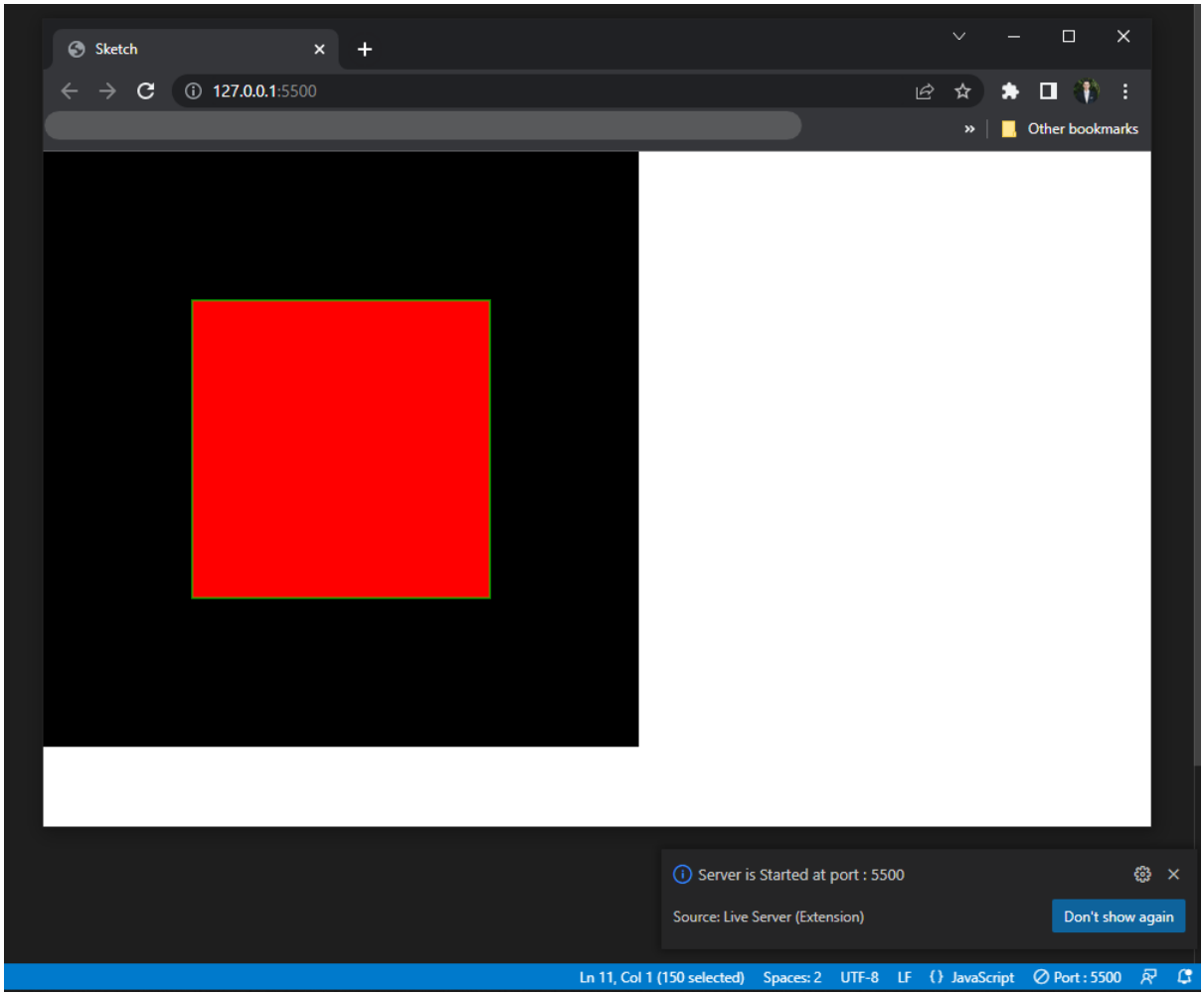
Create a new project in VSCode with the P5.VSCode addon pressing **<Ctrl>+<Shift>+<P>** and select *Create P5.JS project*.

The **sketch.js** contains the JavaScript source code that we are going to edit.

```
function setup() {
  createCanvas(500, 500);
}

function draw() {
  background(0,0,0);
  fill(255,0,0);
  stroke(0,255,0);
  rect(125,125,250,250);
}
```

We have to press the Go Live button in the taskbar of VSCode to trigger the live view and render index.html:





Learn by hacking allows us to discover what a code does just by doing changes.

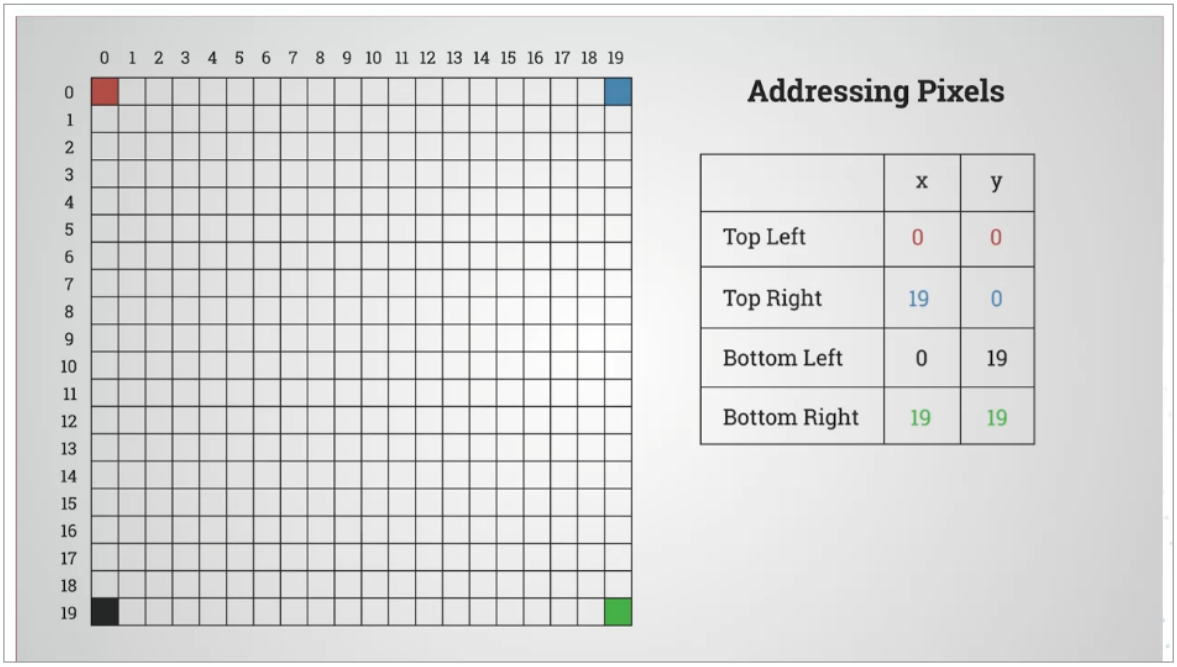
Pixels are dots of the screen and are arranged into a **grid**, which we call a **canvas** and is indexed from 0.

It is called a **Cartesian coordinate** system.

```
createCanvas(500,400); // width, height
```

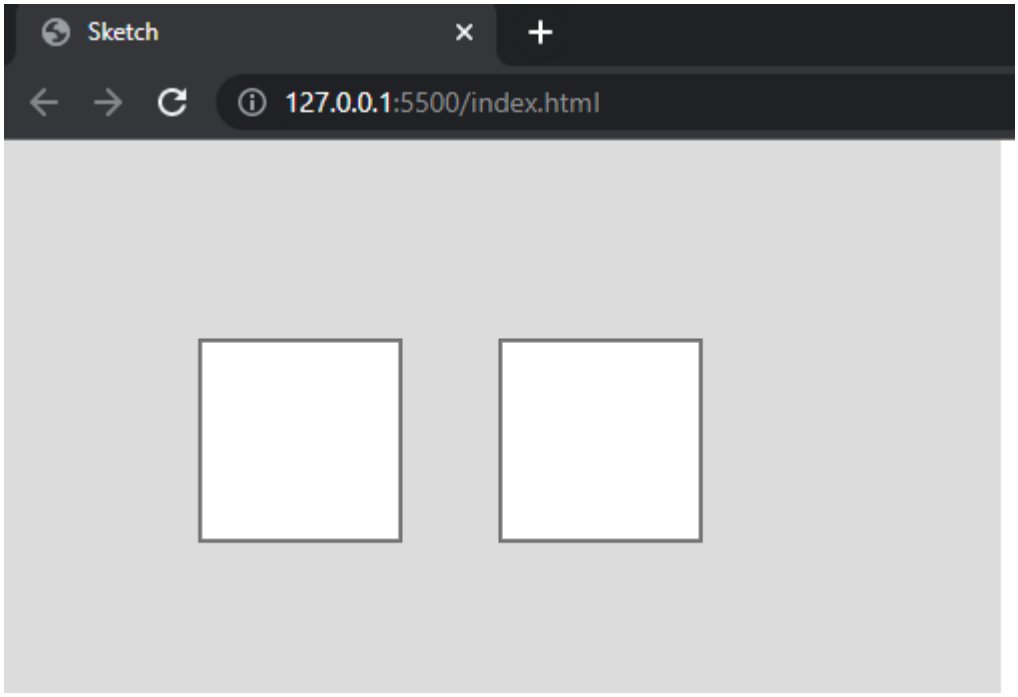
2D coordinate system

Notes Discuss



Drawing two rectangles in p5.js after creating a canvas:

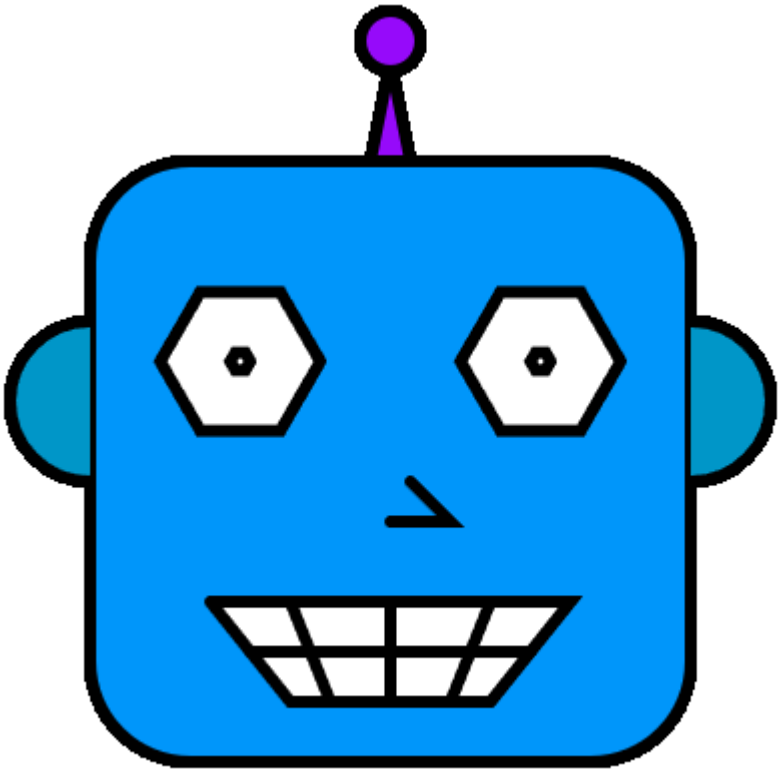
```
function setup() {
  createCanvas(500, 500);
}

function draw() {
  background(220);
  rect(100,100,100,100);
  rect(250,100,100,100);
}
```



Change the robot task:

```
function setup()
{
  //create a canvas for the robot
  createCanvas(500, 500);
}

function polygon(x, y, radius, npoints)
{
  let angle = TWO_PI / npoints;
  beginShape();
  for (let a = 0; a < TWO_PI; a += angle) {
    let sx = x + cos(a) * radius;
    let sy = y + sin(a) * radius;
    vertex(sx, sy);
  }
  endShape(CLOSE);
}

function draw()
{
  strokeWeight(6);

  //robots head
  fill(0,150,250);
  rect(100, 100, 300, 300, 50);


  //robots antenna
  fill(150, 10, 250);
  triangle(240,100,260,100,250,50)
  ellipse(250, 40, 30, 30);
```

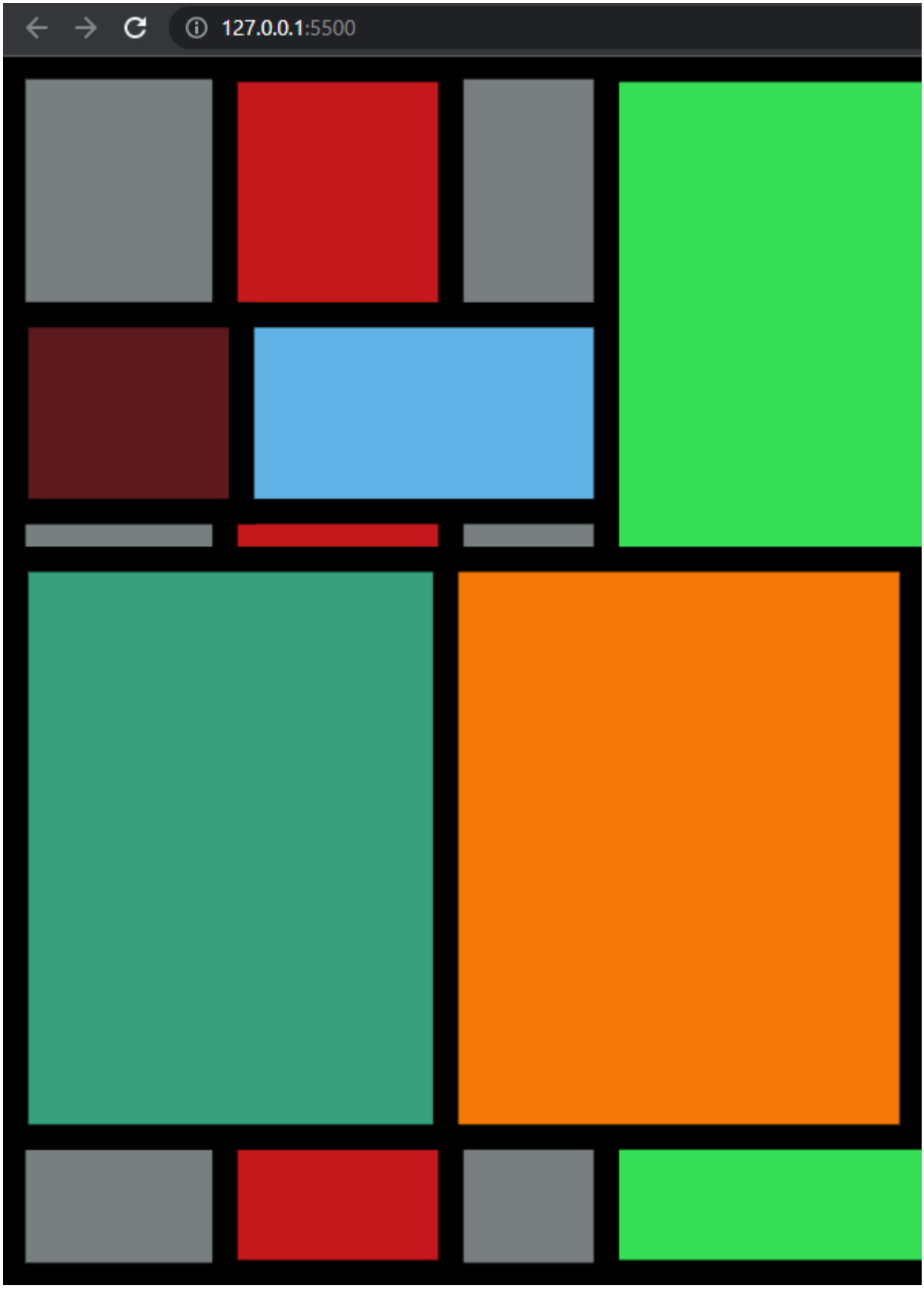
```
  //robots eyes
  fill(255);
  polygon(175,200,40,6);
  polygon(175,200,5,6);
  polygon(325,200,40,6);
  polygon(325,200,5,6);

  //robots nose
  noFill();
  beginShape();
  vertex(260, 260);
  vertex(280, 280);
  vertex(250, 280);
  endShape();

  //robots ears
  fill(0, 150, 200);
  arc(100, 220, 80, 80, HALF_PI, PI+HALF_PI);
  arc(400, 220, 80, 80, PI+HALF_PI, HALF_PI);

  //robots mouth
  fill(255);
  beginShape();
  vertex(160, 320);
  vertex(200, 370);
  vertex(300, 370);
  vertex(340, 320);
  vertex(160, 320);
  endShape();
  line(180, 345, 320, 345);
  line(200, 320, 220, 370);
  line(250, 320, 250, 370);
  line(300, 320, 280, 370);
}
```



De Stijl task:

```
function setup()
{
  width = 600;
  height = 800;
  //create a large square canvas
  createCanvas(width, height);

  s1 = random(5)+10;
  s2 = random(15)+10;

  // frame
  fill(random(255), random(255), random(255));
  strokeWeight(s1);
  rect(0+s1/2, 0+s1/2, width-s1, height-s1);

  generateCross();
  generateCross();
}

function generateCross()
{
  xy1 = random(height/2-100)+100;
  xy2 = xy1 + random(height/2)-50;
  xy3 = random(width/2-50)+50;
  xy4 = xy3 + random(width/2)-25;
```

```
  // vertical rectangle
  strokeWeight(s2);
  fill(random(255), random(255), random(255));
  rect(xy2, 0+s2/2, xy3-s2, height-s2);

  // full horizontal rectangle
  fill(random(255), random(255), random(255));
  rect(0+s2/2, xy1+s2/2, width-s2, xy2-s2);

  // smaller rectangle
  fill(random(255), random(255), random(255));
  rect(0+s2/2, xy1+s2/2, xy3-s2, xy2-s2);
}

function draw()
{
}
```


127.0.0.1:5500