

☆ p5 Cheat Sheet ☆

Canvas

p5 lets you draw on a “canvas.”

They can be any size:

```
createCanvas(width,height)
```

To create a canvas that is the size of the window:

```
createCanvas(windowWidth>windowHeight)
```

Sketches

Every sketch has two built-in functions:

```
function setup() {  
  // This runs ONCE each time a sketch is run  
}  
  
function draw() {  
  // This is a LOOP that runs forever, top to bottom  
}
```

Rectangles

```
rect(x,y,width,height)
```

By default, `x` and `y` are the top-left corner of the rectangle:

If you want the rectangle to start from the center, type `rectMode(CENTER)` before your rectangle.

Ellipses

```
ellipse(xCenter,yCenter,width,height)
```

Lines

```
line(xStartingPoint,yStartingPoint,xEndingPoint,yEndingPoint)
```

Variables

Variables are used to store values. You can change the values of variables to affect your code.

Anytime you want something in your code to change – a location, a color, a size – you will use a variable.

Declaring Variables

Variables must be declared, and then assigned values. Use `var` to create a new variable. You can assign any name to a variable as long as it is not a variable name that is already built into `p5.js`.

```
var x; //Declares the variable  
x = 12; //Assigns a value to the variable
```

This can be shortened by declaring a variable and assigning it a value in one line of code:

```
var x = 12;
```

Use the variables in your code where you would like to use that value.:

```
ellipse(x,100,50,50)  
//Draws the ellipse at (12,100,50,50)
```

Moving things with variables

`function draw()` runs repeatedly, updating your variables each cycle, which allows you to animate your code. You can move a shape by creating a variable for its x or y location and writing an equation that adds or subtracts from that variable.

```
var x = 0;
//Declare the variable
function setup() {
  createCanvas(windowWidth, windowHeight);
}

function draw() {
  background('white');
  //Without a background, every circle that is drawn is visible
  //A background will cover up all but the current shape
  x = x + 1;
  //This equation adds 1 to x every time the draw function repeats
  //As x increases by 1, the ellipse moves to the right
  ellipse(x, 200, 50, 50);
}
```

Conditionals

Conditionals allow you to ask a question to your code and decide between different actions based on whether the answer to that question is true or false.

`If` statements provide a condition under which to perform a certain action. The condition is written in parentheses, and the action is written in brackets. If the condition is met (or “true”), the program will perform the action written in the brackets.

```
function setup(){
  createCanvas(windowWidth,windowHeight);
}
function draw(){
  if(mouseIsPressed){
    background('black')
    //If the mouse is pressed, the background will turn black
  }
}
```

Adding `else{ }` provides an alternative action to be performed while the condition is not met.

```
function setup(){
  createCanvas(windowWidth,windowHeight);
}
```

```
function draw(){
  if(mouseIsPressed){
    background('black')
  }
  //If the mouse is pressed, the background will turn black
  else{
    background('white')
  }
  //If the mouse is not pressed, the background will be white
}
```

If two conditions need to be met before an action is performed, the two conditions can be joined inside the parentheses with `&&` , which is the symbol for “and.”

```
if(mouseIsPressed && mouseX<windowWidth/2){
  background('black')
  //The background will turn black only if the mouse is pressed
  //and if the mouse is on the left side of the canvas.
}
```

If either of the two conditionals needs to be met before an action is performed, the conditionals can be joined with `||` , which is the symbol for “or.”

```
if(mouseIsPressed || mouseX<windowWidth/2){
  background('black')
  //The background will turn black either if the mouse is
  //pressed, or if the mouse is on the left side of the canvas.
}
```