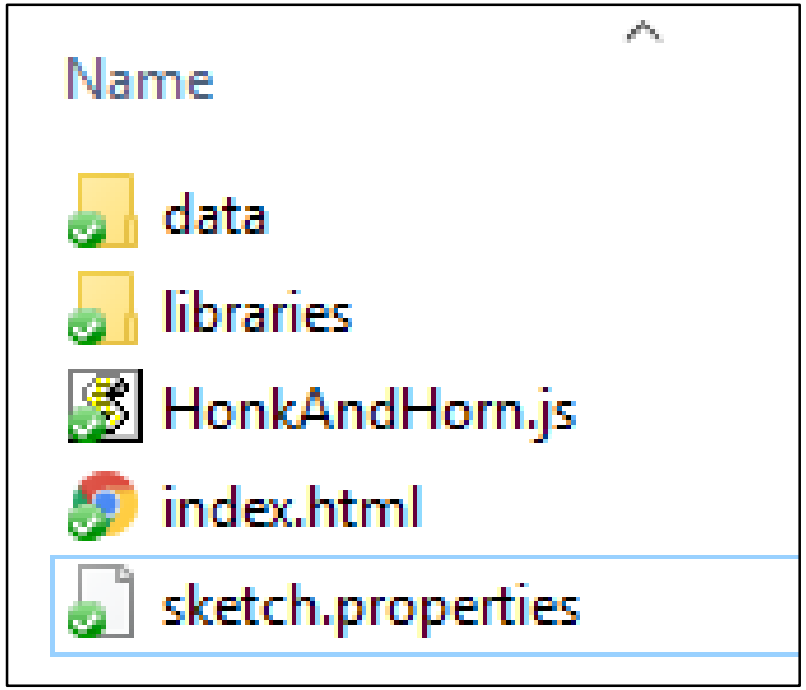
JavaScript p5 and the DOM

- To date we have always used the “canvas”
- Everything is drawn to the canvas
- Use createCanvas() to create a canvas
  - Note the “create”
- Let’s do things outside the canvas
  - Using other “create”, which we will show shortly
  - Need a library called “p5.dom”

Review of the Processing IDE Editor

- The IDE manages:
  - Your JavaScript p5 code (in this screenshot is HonkAndHorn.js)
  - Any other JavaScript p5 tabs you create (such as in the Trading Cards assignment)
  - index.html
  - A configuration file for the sketch properties
  - The data directory
  - The p5.min.js library and possible other libraries such as p5.sound.min.js
- The “IDE” refers to Integrated Development Environment, meaning it integrates these files in a folder for you



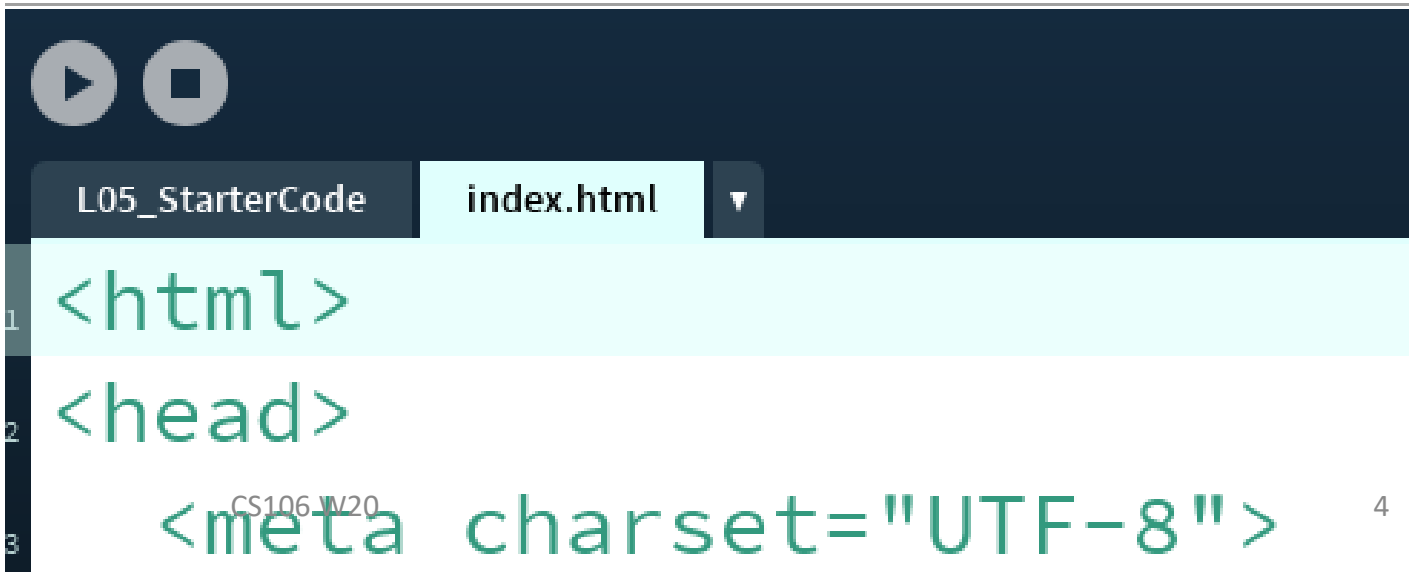
Processing IDE has a “sketch.properties” file

- This is only needed by the Processing IDE.
- In CS106, we never change this file.

```
1 mode=p5.js
2 mode.id=processing.mode.p5js.p5jsMode
3
```

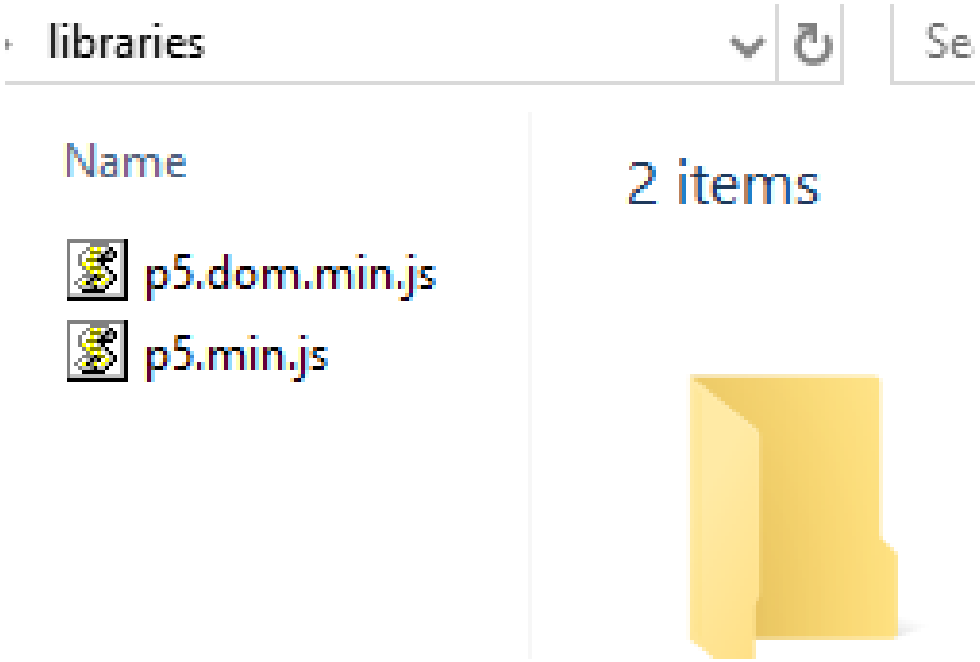
Processing IDE has an “Index.html” tab

- index.html contains, among other things, the contents of your webpage
- You can add content to index.html directly (we don’t do this in CS106)
- You can use JavaScript p5 to add content and interactivity to your webpage (we do this in CS106)

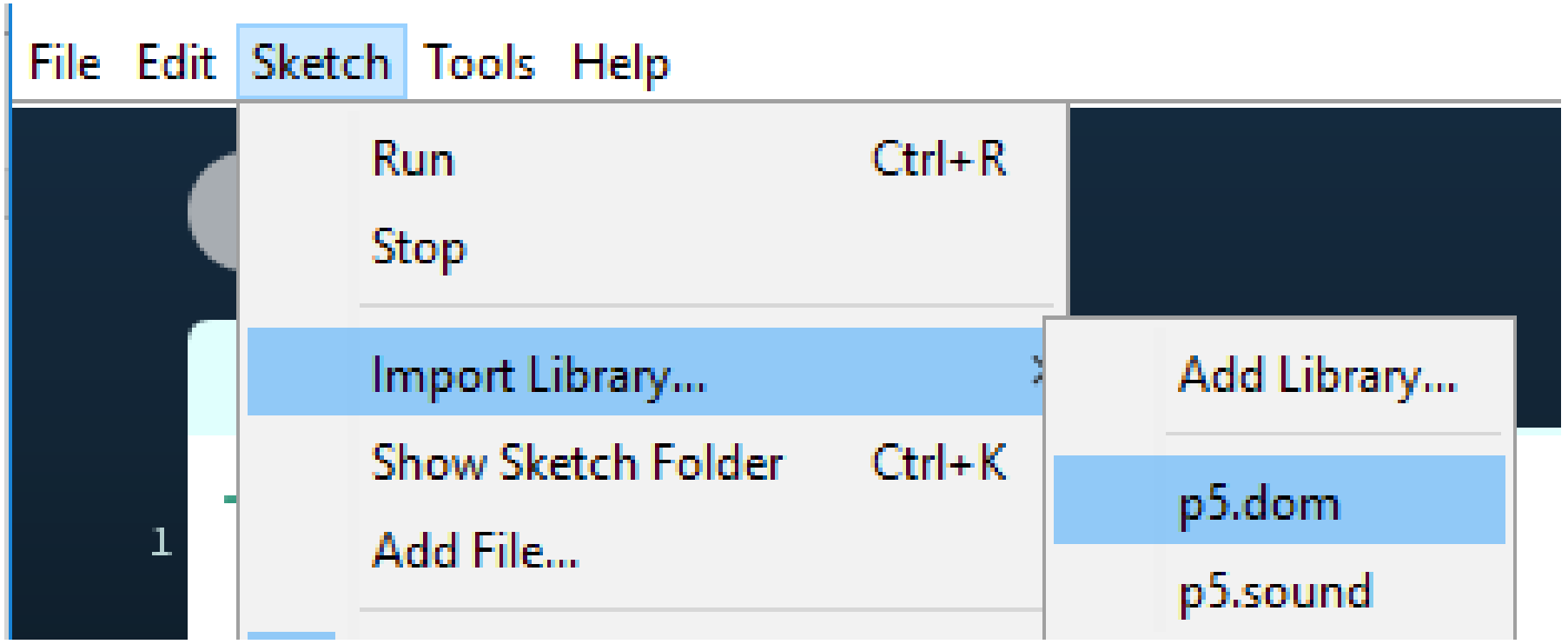


Processing IDE has a “libraries” directory

- It contains any p5 libraries that you have loaded
- In the example to the right, it has two libraries.
  - The main one is “p5.min.js”.
  - The example to the right also includes “p5.dom.min.js” which is a library we will use starting today.

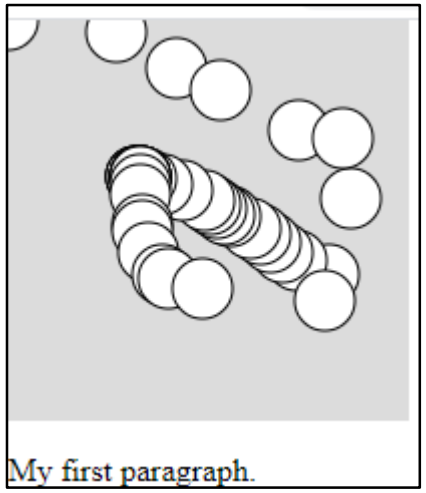


Add the “p5.dom.min.js” library



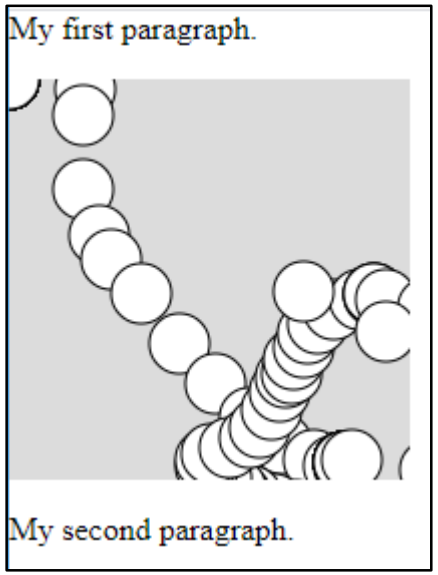
Create a Paragraph Below the Canvas

```
function setup() {
    createCanvas(200, 200);
    background(220);
    createP("My first paragraph.");
}
function draw() {
    ellipse(mouseX, mouseY, 30, 30);
}
```



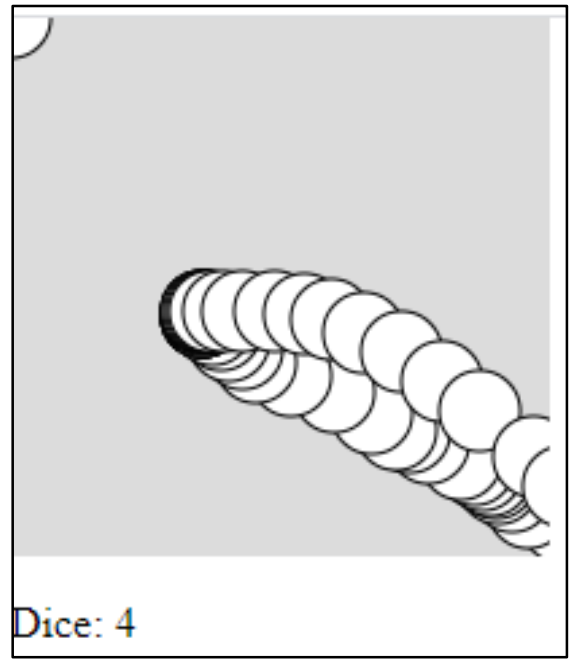
Create Paragraphs Above/Below the Canvas

```
function setup() {
    createP("My first paragraph.");
    createCanvas(200, 200);
    background(220);
    createP("My second paragraph.");
}
function draw() {
    ellipse(mouseX, mouseY, 30, 30);
}
```



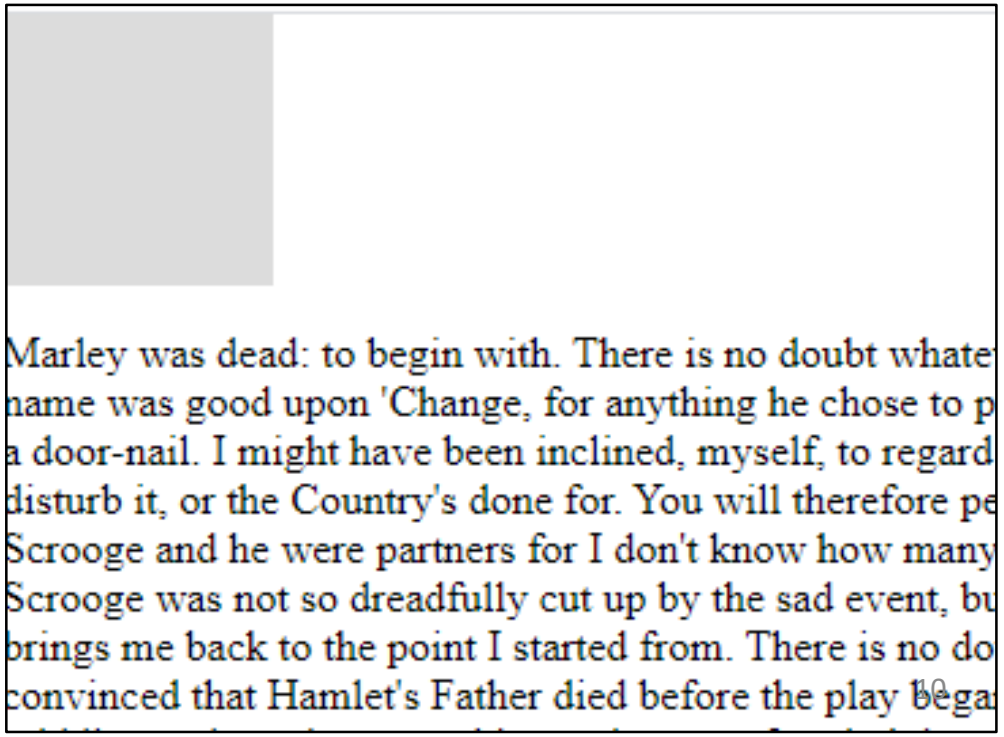
Use Variables

```
let dice;
function setup() {
  createCanvas(200, 200);
  background(220);
  dice = floor(random(1, 7));
  createP("Dice: " + dice);
}
function draw() {
  ellipse(mouseX, mouseY, 30, 30);
}
```



Read and Display Data from a Text File

```
let lines = [];
function preload(){
  lines = loadStrings("data/marley.txt");
}
function setup() {
  createCanvas(200, 200);
  background(220);
  createP(lines);
}
```




iclicker


iclicker


iclicker

List of DOM elements

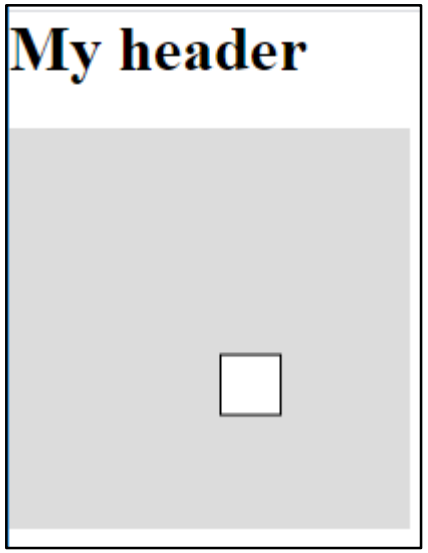
- To see the JavaScript p5 list of available DOM elements:
  - Go to: https://p5js.org/reference/#group-DOM
  - Scroll down to “DOM”
  - There are many
- CS106 will cover some of these including:
  - createP()
  - createElement()
  - createButton()
  - createSlider()
  - createInput()
  - createRadio()

Create a Header “h1” Element

```
let myHeader;

function setup() {
  myHeader = createElement("h1", "My header");
  createCanvas(200, 200);
}

function draw() {
  background(220);
  rect(mouseX, mouseY, 30, 30);
}
```



Modify an Element using “html”

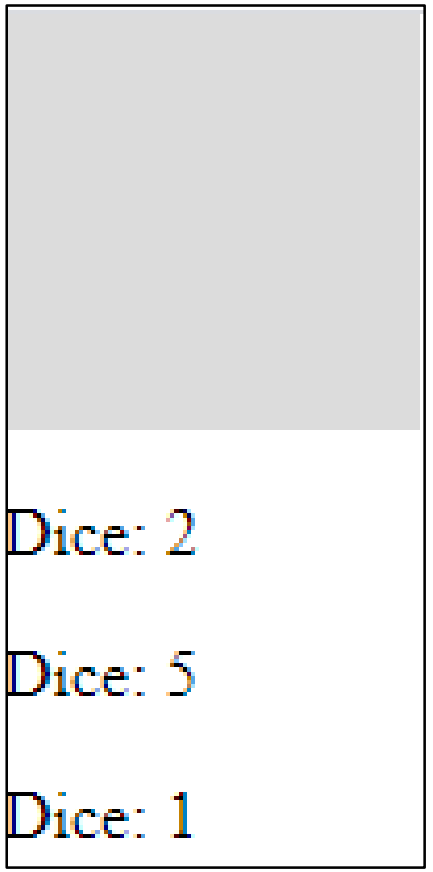

```
let myHeader;

function setup() {
  myHeader = createElement("h1", "My header");
  createCanvas(200, 200);
  background(220);
}

function mousePressed() {
  myHeader.html("New Header !!!");
}
```

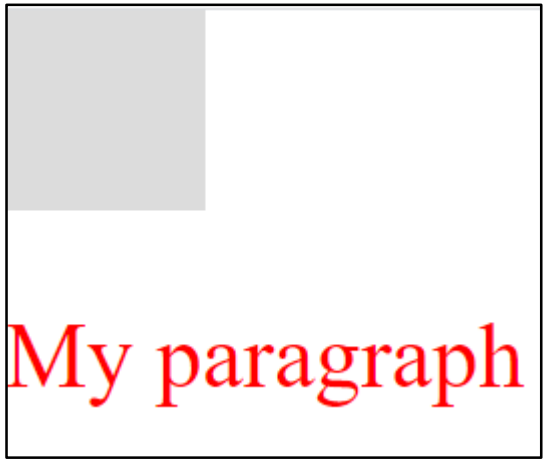
Create Many Paragraphs

```
let dice;
function setup() {
  createCanvas(100, 100);
  background(220);
}
function mousePressed() {
  dice = floor(random(1, 6));
  createP("Dice: " + dice);
}
```



Style an Element

```
let myPara;
let size = 18;
function setup() {
  createCanvas(100, 100);
  background(220);
  myPara = createP("My paragraph");
  myPara.style('color', 'red');
}
function mousePressed() {
  size = size + 2;
  myPara.style('font-size', size);
}
```



JavaScript p5 uses CSS Style Tags

- To see a list of CSS tags:
  - Go to a list: https://www.w3schools.com/cssref/
- CS106 will cover only ‘color’ and ‘font-size’:
  - `myPara.style('color', 'red');`
  - `myPara.style('font-size', size);`


iclicker


iclicker


iclicker

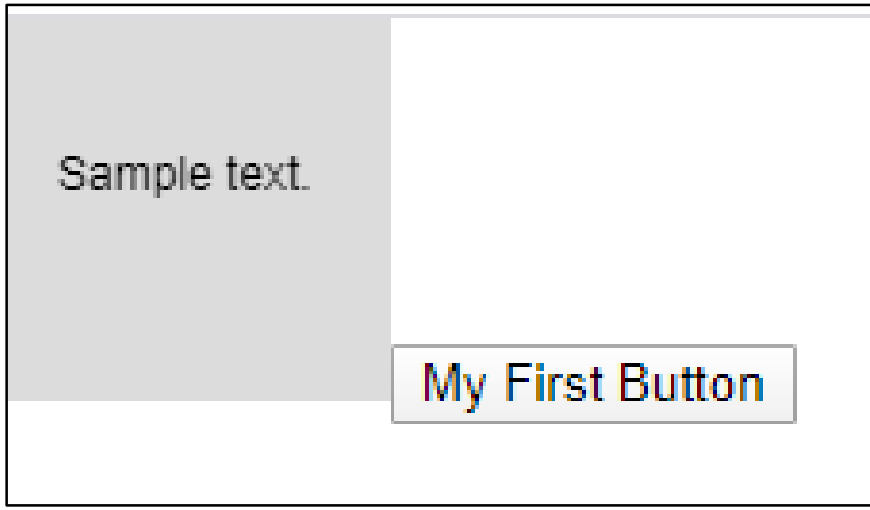
CreateButton() and Events

- Create interactive elements
  - createButton()

Create a Button (it does nothing)

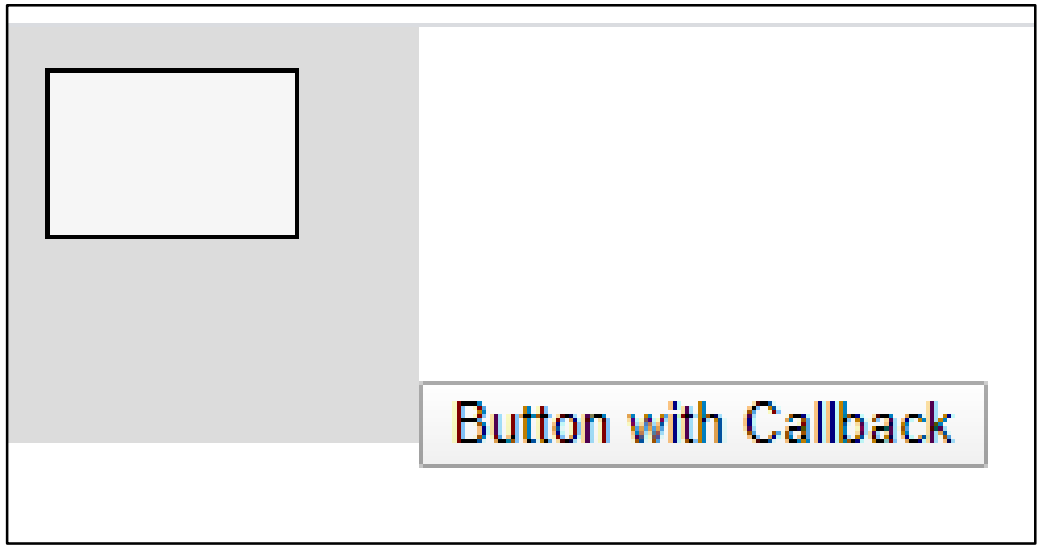
```
function setup() {
    createCanvas(100, 100);
    createButton("My First Button");
  }

  function draw() {
    background(220);
    text("Sample text.", 10, 40);
  }
```



Button With a Callback Function “changeColor”

```
let myButton;
function setup() {
  createCanvas(100, 100);
  myButton = createButton("Button with Callback");
  myButton.mouseClicked(changeColor);
}
function changeColor() {
  fill(random(256));
}
function draw() {
  background(220);
  rect(10, 10, 60, 40);
}
```




iclicker


iclicker


iclicker

List of DOM elements

- To see the JavaScript p5 list of available DOM elements:
  - Go to: https://p5js.org/reference/#/p5.Element
  - Look at all the methods for every p5 element
- CS106 will cover some of these including:
  - mousePressed()
  - mouseOver()
  - mouseOut()

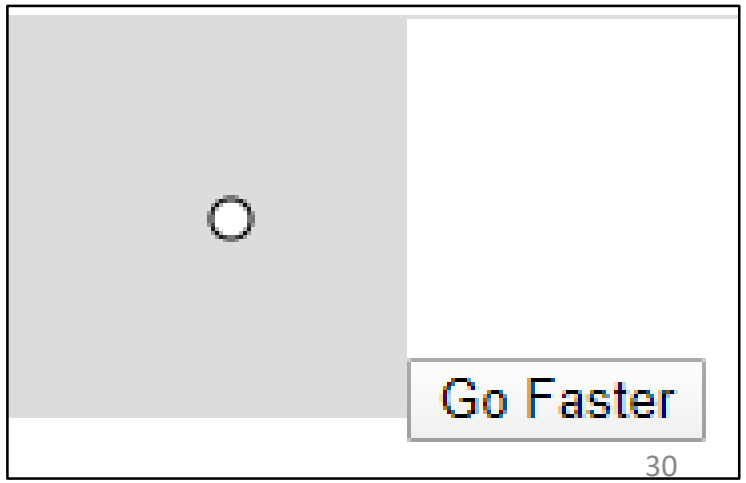
Use a Button to Control the Ball Speed

```
let ballX = 0;
let myButton;

function setup() {
  createCanvas(100, 100);
  myButton = createButton("Go Faster");
  myButton.mouseClicked(increaseSpeed);
}

function increaseSpeed() {
  speed ++;
}
```

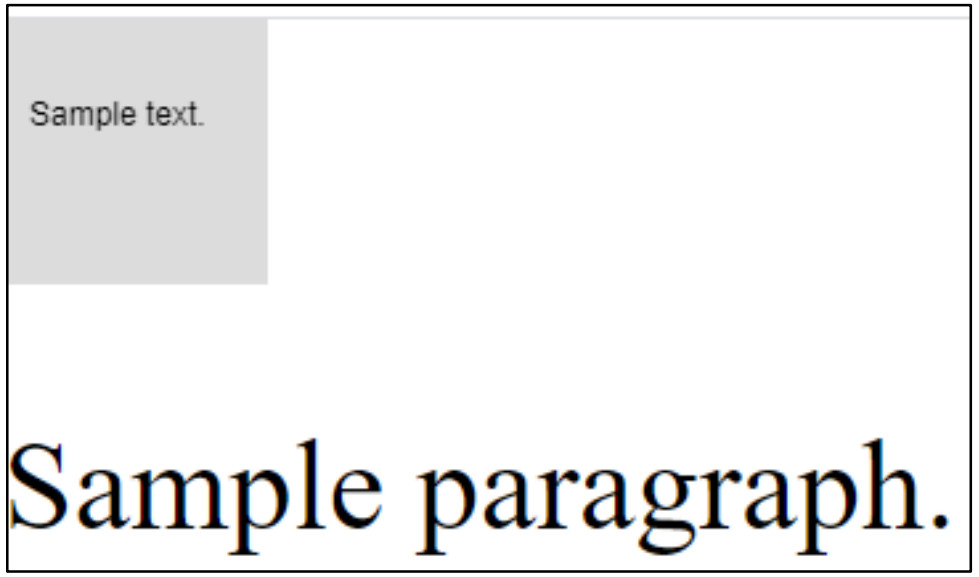
```
function draw() {
  background(220);
  ballX = ballX + speed;
  ellipse(ballX, 50, 10, 10);
  if (ballX > width) {
    ballX = 0;
  }
}
```



Events on a Paragraph

```
let p;
function setup() {
  createCanvas(100, 100);
  p = createP("Sample paragraph.");
  p.mouseOver(showBigText);
  p.mouseOut(showSmallText);
}
function showBigText() {
  p.style('font-size', 48);
}
function showSmallText() {
  p.style('font-size', 20);
}
```

```
function draw() {
  background(220);
  text("Sample text.", 10, 40);
}
```



An Annoying Webpage (part 1 of 2)
Honk and Horn

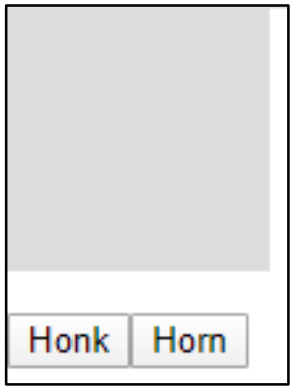
```
let honkSound;
let hornSound;

function preload() {
  honkSound = loadSound("data/honk.wav");
  hornSound = loadSound("data/horn.wav");
}
```

An Annoying Webpage (part 2 of 2)
Honk and Horn

```
function setup() {
  background(220);
  createP(" ");
  honkButton = createButton("Honk");
  honkButton.mouseClicked(playHonk);
  hornButton = createButton("Horn");
  hornButton.mouseClicked(playHorn);
}
```

```
function playHorn() {
  hornSound.play();
}
function playHonk() {
  honkSound.play();
}
```



Sliders

```
let mySlider;

function setup() {
  createCanvas(600, 400);
  createP(" ");
  mySlider = createSlider(0, 100, 20);
}
```

```
function draw() {
  background(220);
  textSize(mySlider.value());
  text("Sample text.", 10,height/2);
}
```

Text Input Box (slide 1 of 2)

```
let myInput;
let greeting;
let button;

function setup() {
  createCanvas(100, 100);
  background(220);
  greeting = createElement('h2', 'what is your name?');
  myInput = createInput();
  button = createButton('submit');
  button.mouseClicked(greet);
}
```

Text Input Box (slide 2 of 2)

```
function greet() {
  let name = myInput.value();
  greeting.html('Hello ' + name + '!');
  myInput.value('');
}
```

Radio Control (slide 1 of 3)

```
let tileset;
let myRadio;

function preload() {
  tileset = loadImage("data/Tiles.png");
}
```

Radio Control (slide 2 of 3)

```
function setup() {
  createCanvas(640, 576);
  noSmooth();
  createP(" ");
  createP("Choose night or day.");
  myRadio = createRadio();
  myRadio.option("grassland");
  myRadio.option("winter");
  myRadio.option("tropical");
  myRadio.option("autumn");
  myRadio.value("grassland");
}
```

Radio Control (slide 3 of 3)

```
function draw() {
  if (myRadio.value() === "grassland") {
    copy(tileset, 16, 16, 160, 144, 0, 0, 160 * 4, 144 * 4);
  } else if (myRadio.value() === "winter") {
    copy(tileset, 240, 16, 160, 144, 0, 0, 160 * 4, 144 * 4);
  } else if (myRadio.value() === "tropical") {
    copy(tileset, 16, 240, 160, 144, 0, 0, 160 * 4, 144 * 4);
  } else if (myRadio.value() === "autumn") {
    copy(tileset, 240, 240, 160, 140, 0, 0, 160 * 4, 144 * 4);
  }
}
```


iclicker

```
}
```


iclicker


iclicker

}

The End