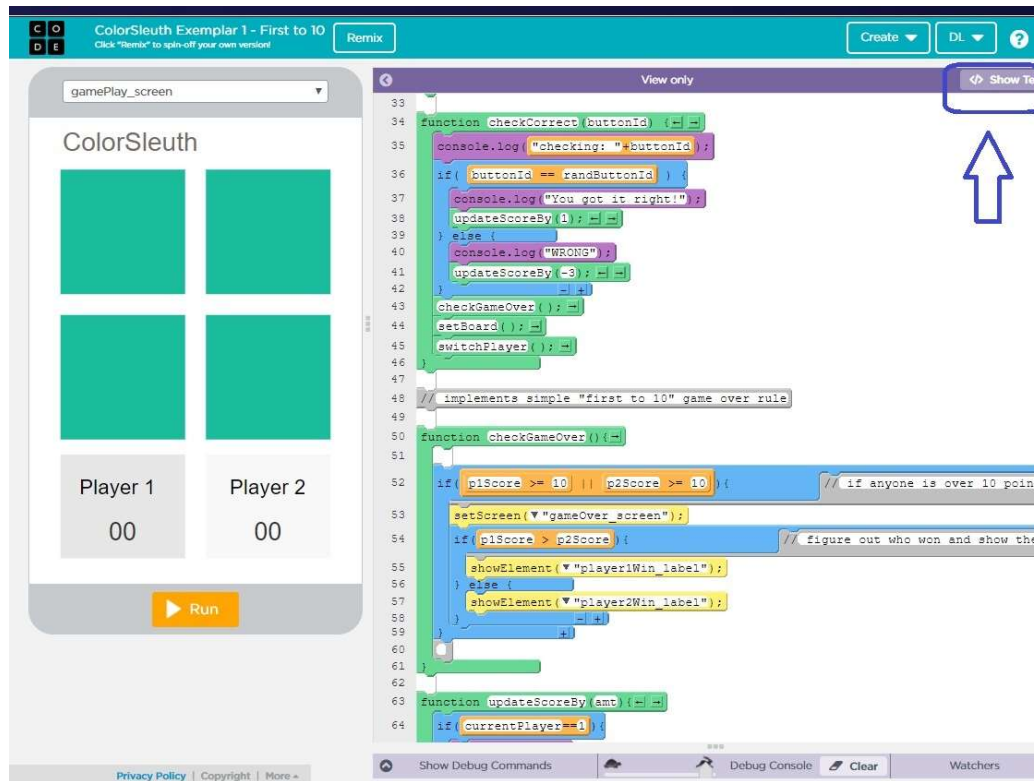


USING CODE PRINT

You must identify the Algorithm (function) and the Abstraction you will be discussing in your write up.

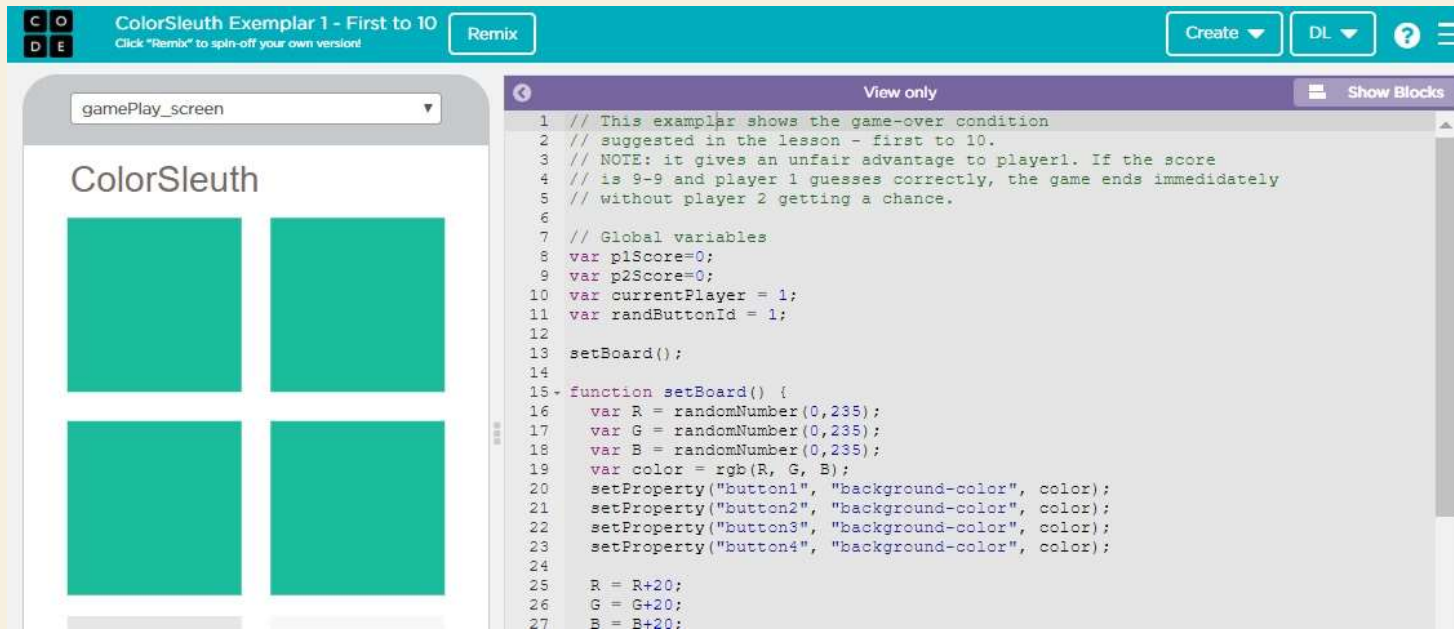
You will need to have your code in TEXT format in order to use the codePrint app.



MAKE SURE YOU ARE IN TEXT MODE

Click the SHOW TEXT button

THIS IS HOW IT SHOULD APPEAR IN TEXT MODE



```
1 // This exemplar shows the game-over condition
2 // suggested in the lesson - first to 10.
3 // NOTE: it gives an unfair advantage to player1. If the score
4 // is 9-9 and player 1 guesses correctly, the game ends immediately
5 // without player 2 getting a chance.
6
7 // Global variables
8 var p1Score=0;
9 var p2Score=0;
10 var currentPlayer = 1;
11 var randButtonId = 1;
12
13 setBoard();
14
15 function setBoard() {
16   var R = randomNumber(0,235);
17   var G = randomNumber(0,235);
18   var B = randomNumber(0,235);
19   var color = rgb(R, G, B);
20   setProperty("button1", "background-color", color);
21   setProperty("button2", "background-color", color);
22   setProperty("button3", "background-color", color);
23   setProperty("button4", "background-color", color);
24
25   R = R+20;
26   G = G+20;
27   B = B+20;
```

USE CTRL +A TO SELECT ALL

USE CTRL +C TO COPY ALL

ColorSleuth Exemplar 1 - First to 10

Click "Remix" to spin-off your own version

Remix

Create

DL

?

gamePlay_screen

ColorSleuth

Player 1

00

Player 2

00

View only

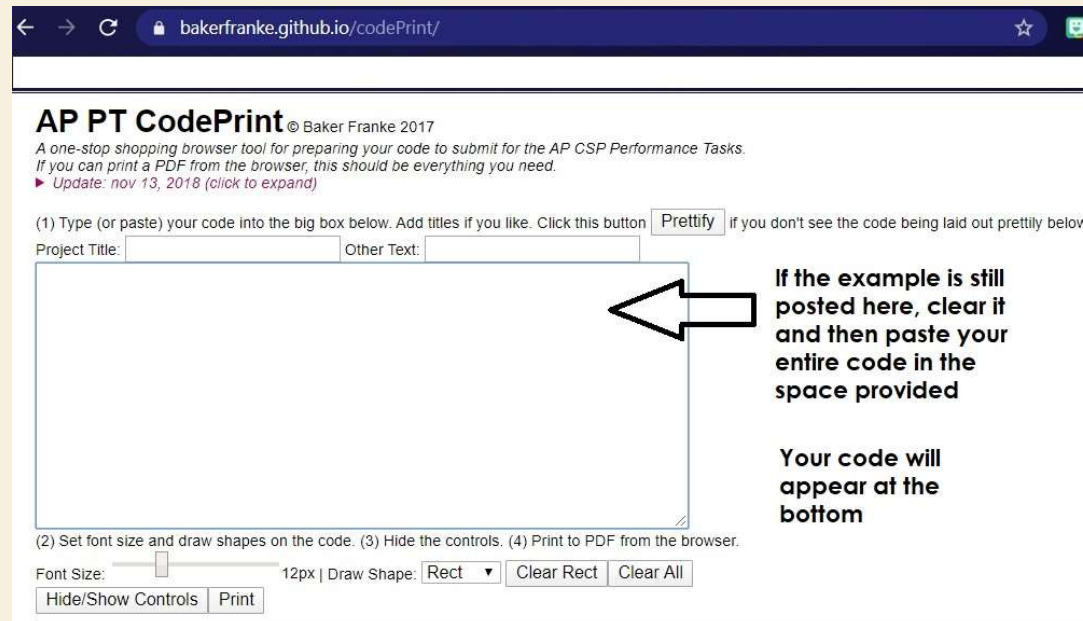
Show Blocks

```
1 // This exemplar shows the game-over condition.
2 // suggested in the lesson - first to 10.
3 // NOTE: it gives an unfair advantage to player1. If the score
4 // is 9-9 and player 1 guesses correctly, the game ends immediately
5 // without player 2 getting a chance.
6
7 // Global variables
8 var p1Score=0;
9 var p2Score=0;
10 var currentPlayer = 1;
11 var randButtonId = 1;
12
13 setBoard();
14
15 function setBoard() {
16   var R = randomNumber(0,235);
17   var G = randomNumber(0,235);
18   var B = randomNumber(0,235);
19   var color = rgb(R, G, B);
20   setProperty("button1", "background-color", color);
21   setProperty("button2", "background-color", color);
22   setProperty("button3", "background-color", color);
23   setProperty("button4", "background-color", color);
24
25   R = R+20;
26   G = G+20;
27   B = B+20;
28   var diffColor = rgb(R, G, B);
29   randButtonId = "button"+randomNumber(1,4);
30   setProperty(randButtonId, "background-color", diffColor);
31   console.log("correct button is: "+randButtonId);
32 }
33
```

Click CTRL+A to select all your code

Click CTRL + C to copy all your code

[HTTPS://BAKERFRANKE.GITHUB.IO/CODEPRINT/](https://bakerfranke.github.io/codePrint/) **CLEAR THE EXAMPLE CODE, CLICK INSIDE THE EMPTY BOX THEN CLICK CTRL +V TO PASTE YOUR CODE INTO THE APP**



YOU SHOULD NOW SEE YOUR CODE IN THE BOTTOM PANEL

```
(2) Set font size and draw shapes on the code. (3) Hide the controls. (4) Print to PDF from the browser.
Font Size:  | Draw Shape:   
 

1 // This exemplar shows the game-over condition
2 // suggested in the lesson - first to 10.
3 // NOTE: it gives an unfair advantage to player1. If the score
4 // is 9-9 and player 1 guesses correctly, the game ends immediately
5 // without player 2 getting a chance.
6
7 // Global variables
8 var p1Score=0;
9 var p2Score=0;
10 var currentPlayer = 1;
11 var randButtonId = 1;
12
13 setBoard();
14
15 function setBoard() {
16   var R = randomNumber(0,235);
17   var G = randomNumber(0,235);
18   var B = randomNumber(0,235);
19   var color = rgb(R, G, B);
20   setProperty("button1", "background-color", color);
```

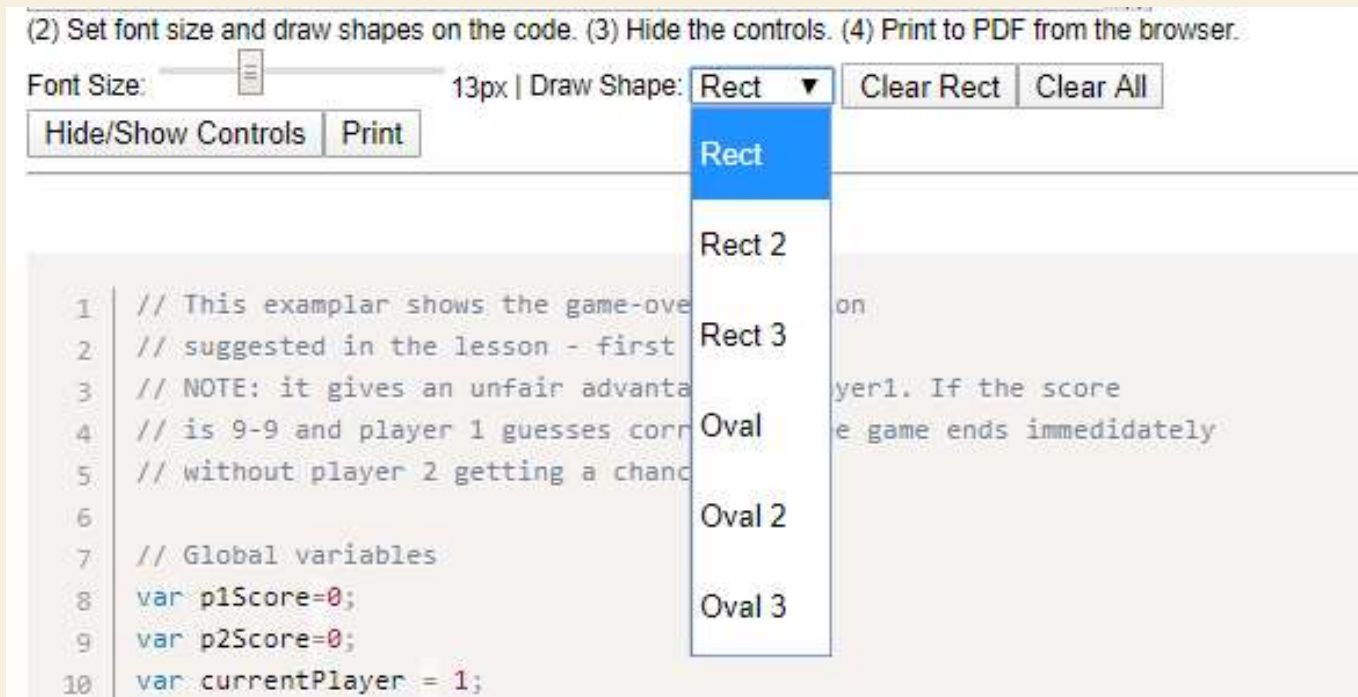
FONT SIZE AND RECT/OVAL CHOICES

(2) Set font size and draw shapes on the code. (3) Hide the controls. (4) Print to PDF from the browser.

Font Size: | Draw Shape: Rect ▾ Clear Rect Clear All

Hide/Show Controls Print

```
1 // This exemplar shows the game-over screen
2 // suggested in the lesson - first player to guess correctly wins
3 // NOTE: it gives an unfair advantage to player1. If the score
4 // is 9-9 and player 1 guesses correctly the game ends immediately
5 // without player 2 getting a chance to guess
6
7 // Global variables
8 var p1Score=0;
9 var p2Score=0;
10 var currentPlayer = 1;
```



LOCATE THE FUNCTION YOU WILL BE DISCUSSING IN 2C

After you have located it, select OVAL and encircle the complete function that includes the Parent function and where you called the child.

You should also select in your code for the CHILD function. Select Oval 2 and select your child function

```
31 console.log("correct button is: "+randButtonId);
32 }
33
34 function checkCorrect(buttonId) {
35     console.log("checking: "+buttonId);
36     if( buttonId == randButtonId ) {
37         console.log("You got it right!");
38         updateScoreBy(1);
39     } else {
40         console.log("WRONG");
41         updateScoreBy(-3);
42     }
43     checkGameOver();
44     setBoard();
45     switchPlayer();
46 }
47
48 // implements simple "first to 10" game over rule
49
50 function checkGameOver(){
51
52     if(p1Score >= 10 || p2Score >= 10){ //
53         setScreen("gameOver_screen");
54         if(p1Score > p2Score){ // +
55             showElement("player1Win_label");
56         } else {
57             showElement("player2Win_label");
58         }
59     }
60 }
61
62
63 function updateScoreBy(amt){
64     if(currentPlayer==1){
```


SELECT ABSTRACTION

Locate the abstraction you will discuss in your write up. Select RECT and place a rectangle around the abstraction
Then select RECT 2 and place it around the location that is called in your program.

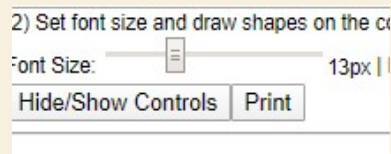
```
31 console.log("correct button is: "+randButtonId);
32 };
33
34 function checkCorrect(buttonId) {
35     console.log("checking: "+buttonId);
36     if( buttonId == randButtonId ) {
37         console.log("You got it right!");
38         updateScoreBy(1);
39     } else {
40         console.log("WRONG");
41         updateScoreBy(-3);
42     }
43     checkGameOver();
44     setBoard();
45     switchPlayer();
46 }
47
48 // implements simple "first to 10" game over rule
49
50 function checkGameOver(){
51
52     if(p1Score >= 10 || p2Score >= 10){ // if
53         setScreen("gameOver_screen");
54         if(p1Score > p2Score){ // fig
55             showElement("player1Win_label");
56         } else {
57             showElement("player2Win_label");
58         }
59     }
60 }
61
62
63 function updateScoreBy(amt){
64     if(currentPlayer==1){
65         p1Score += amt;
66     }
67     else{
68         p2Score += amt;
69     }
70     console.log("P1 Score: "+p1Score);
71     console.log("P2 Score: "+p2Score);
72     setText("score1_label", p1Score);
```

NOW YOU ARE READY TO PRINT TO PDF

Add your PROJECT TITLE at the top of the page

Title: 2020_Create_Code

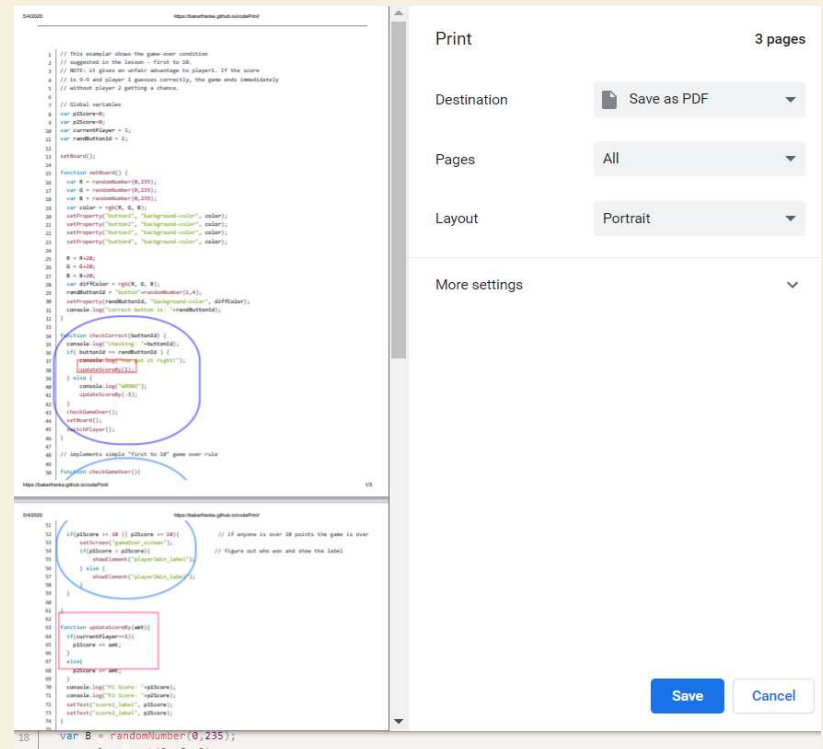
Click PRINT



Then make sure you have the destination as SAVE AS PDF

Click SAVE and it will want you to select where you want to save it. Save it in a location you will remember (like your desktop). Save your file as

2020_Create_Code



TWO LAST THINGS

When you complete your write up, you can insert a picture into your explanation so here is what I need you to do to help you.

Use your snipping tool and clip out your

Abstraction

(saved as **Abstraction.jpg**)

and your Function

(saved as **Parent-Child Function.jpg**)

Abstraction.jpg

```
36 // buttonId == randButtonId / {
37   console.log("you got it right!");
38   updateScoreBy(1);
39 } else {
40   console.log("WRONG");
41   updateScoreBy(-3);
42 }
43 checkGameOver();
44 setBoard();
45 switchPlayer();
46 }
47
48 // implements simple "first to 10" game
49
50 function checkGameOver(){
51
52   if(p1Score >= 10 || p2Score >= 10){
53     setScreen("gameOver_screen");
54     if(p1Score > p2Score){
55       showElement("player1win_label");
56     } else {
57       showElement("player2win_label");
58     }
59   }
60 }
61
62
63 function updateScoreBy(amt){
64   if(currentPlayer==1){
65     p1Score += amt;
66   }
67   else{
```

Parent-Child Function

```
31 // buttonId == randButtonId / {
32 }
33
34 function checkCorrect(buttonId) {
35   console.log("checking: "+buttonId);
36   if( buttonId == randButtonId ) {
37     console.log("you got it right!");
38     updateScoreBy(1);
39   } else {
40     console.log("WRONG");
41     updateScoreBy(-3);
42   }
43   checkGameOver();
44   setBoard();
45   switchPlayer();
46 }
47
48 // implements simple "first to 10" game over rule
49
50 function checkGameOver(){
51
52   if(p1Score >= 10 || p2Score >= 10){
53     setScreen("gameOver_screen");
54     if(p1Score > p2Score){
55       showElement("player1win_label");
56     } else {
57       showElement("player2win_label");
58     }
59   }
60 }
```

**END
CODEPRINT
VIDEO NOTES**

NOTES WILL BE AVAILABLE IN TEAMS