

While Loops & Lists

CSCI 185: Repeating Code

Outline

1. Review (Sample Quiz 3 questions)
2. Intro to while loops
3. Exercises: While loops
4. Intro to arrays (also called lists) & objects
5. Aside: using random numbers

Outline

1. **Review (Sample Quiz 3 questions)**
2. Intro to while loops
3. Exercises: While loops
4. Intro to arrays (also called lists) & objects
5. Aside: using random numbers

Announcements

1. [Project Proposal](#) due **tonight** at midnight. Looking forward to reading them!
2. [Homework 6](#) due next Monday (4/10) at midnight. We'll take a look at that in class on Wednesday.
3. Tutorial 9 due this Friday (4/8) on looping, art, and animation. Will post it by Wednesday.
4. Class cancelled 2 Fridays from now (4/14). Sarah will be out of town at an academic conference.

Studying for Quiz 3

[Quiz 3](#) is all about programming, so I want to make sure that you're reviewing and understanding the concepts as you go. Given what we have covered, you should be able to answer the following questions now (review the past few lectures if you're confused on these concepts or come to office hours)...

Arithmetic Operators

Consider the following:

```
let result = 9 % 2;
```

1. What is the VALUE stored in result?
2. What is the DATA TYPE of the value stored in result?

Arithmetic Operators

Consider the following:

```
let result = 9 / 2;
```

1. What is the VALUE stored in result?
2. What is the DATA TYPE of the value stored in result?

Arithmetic Operators

Consider the following:

```
let a = 4 ** 2;  
let result = a / 2;
```

1. What is the VALUE stored in result?
2. What is the DATA TYPE of the value stored in result?

Functions

```
function f1(a, b) {  
    return a * b;  
}
```

```
function f2(a, b) {  
    return b - a;  
}
```

```
let x = f1(2, 3);  
let y = f2(1, x);  
let z = f2(y, f1(4, 5));
```

What is stored in x, y and z?

Comparison Operators

Consider the following:

```
let a = 50;  
let b = 50;  
let result = b < a;
```

1. What is the VALUE of the value stored in result?
2. What is the DATA TYPE of the value stored in result?

Logical Operators

Consider the following:

```
let c = true;  
let d = false;  
let result = c && d;
```

1. What is the VALUE of the value stored in result?
2. What is the DATA TYPE of the value stored in result?

Logical Operators

Consider the following:

```
let a = false;  
let b = true;  
let c = false;  
let result = !a && (b || c);
```

1. What is the VALUE of the value stored in result?
2. What is the DATA TYPE of the value stored in result?

Logical Operators

Consider the following:

```
let c = true;  
let d = false;  
let result = c || d;
```

1. What is the VALUE of the value stored in result?
2. What is the DATA TYPE of the value stored in result?

Conditional Logic

Given the following code block, what prints to the console?

```
let a = true;
let b = true;
let c = false;
let result;

if (!a) {
  result = 'horse';
} else if (a && c) {
  result = 'donkey';
} else if (b || c) {
  result = 'mule';
} else {
  result = 'llama';
}
console.log(result);
```

Conditional Logic

What prints to the console?

```
function moveAvatar(buttonA, buttonB) {  
  if (buttonA) {  
    console.log("kick");  
  } else if (buttonB) {  
    console.log("jump");  
  } else if (buttonA && buttonB) {  
    console.log("jump kick");  
  }  
}
```

```
moveAvatar(true, true);    ← team 3: what prints to the screen?  
moveAvatar(true, false);  ← team 4: what prints to the screen?  
moveAvatar(false, true);  ← team 1: what prints to the screen?
```

Outline

1. Review (Sample Quiz 3 questions)
- 2. Intro to while loops**
3. Exercises: While loops
4. Intro to arrays (also called lists) & objects
5. Aside: using random numbers

Repeating Tasks

Once you teach your computer to do something once, it is easy to get it to do the same thing many times. Repeating tasks is one of the things a computer does really well.

- What are some examples of tasks you repeat over and over?

While Loops Metaphors

→ `if (there is a pringle in the can) {
 // eat a pringle
}`



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```



But do you really want to eat just one?

While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}  
if (there is a pringle in the can) {  
    // eat a pringle  
}  
if (there is a pringle in the can) {  
    // eat a pringle  
}  
if (there is a pringle in the can) {  
    // eat a pringle  
}
```



But what if you don't know how many pringles are in the can?

While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
→ while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
→ while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
→ while (there is a pringle in the can) {  
    // eat a pringle  
}
```



While Loops Metaphors

```
if (there is a pringle in the can) {  
    // eat a pringle  
}
```

```
// because no one can eat just one  
while (there is a pringle in the can) {  
    // eat a pringle  
}
```



...until they're gone :(



Tweets
4Following
16Followers
2Likes
9Lists
0Moments
0

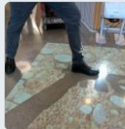
Edit profile

Sarah

@vanwars

Joined June 2017

Photos and videos



Who to follow · Refresh · View all

Followed by Anne Jonas



janelle scott @janelletscott

Follow

Followed by Anne Jonas



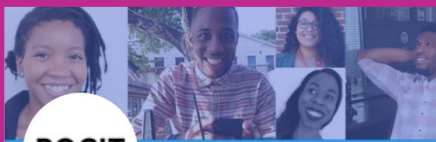
Berkeley School of Infor...

Follow

Followed by the sarahjozket



Somed Like A Girl @sarahjozket



POCIT

PEOPLE OF COLOR IN TECH

Following

PeopleOfColorInTech ✓

@pocitech

The stories of 'people of color' in tech /
Jobs 📄 pocitjobs.com / By
[@michaelberhane_](#)



Medium ✓

@Medium

Words matter. Learn more at



Tamyra Walker

@tamyraintech

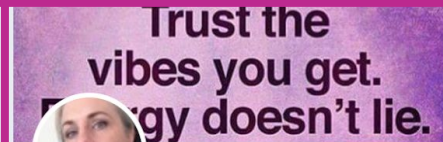
#TechEquity as a stance, #STEM
inclusion as a platform, #Advocacy as a
vehicle of change.



CSUMB Women's Soccer

@CSUMBWSoccer

Everything women's soccer at Cal State



the sarahjozket 🔒

@ketron_sarah Follows you



Stuart Geiger ✓

@staeiou

computational ethnographer and

Home

Moments

Notifications

Messages

Search Twitter

Tweet



Tweets

Following

Followers

Likes

Lists

Moments

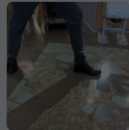
Edit profile

Sarah

@vanwars

Joined Jun

Photos and



Who to follow

Follow



jane

Follow

Followed by Anne Jonas



Berkeley School of Infor...

Follow

Followed by the sarahjoket

1. What do you want to repeat?
2. How long to you want to repeat it?
3. What do you want to change each time?



Medium ✓

@Medium

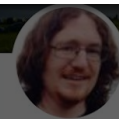
Words matter. Learn more at



CSUMB Women's Soccer

@CSUMBWSoccer

Everything women's soccer at Cal State



Stuart Geiger ✓

@staeiou

computational ethnographer and



6

Tweets
4Following
16Followers
2Likes
9Lists
0Moments
0

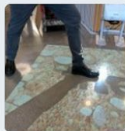
Edit profile

Sarah

@vanwars

Joined June 2017

Photos and videos



3

Who to follow · Refresh · View all

Followed by Anne Jonas



janelle scott @janelletscott

Follow

Followed by Anne Jonas



Berkeley School of Infor...

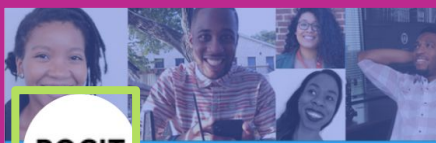
Follow

Followed by the sarahjozket



the sarahjozket

6



POCIT

PEOPLE OF COLOR IN TECH

Following

PeopleOfColorInTech ✓

@pocitech

The stories of 'people of color' in tech /
Jobs pocitjobs.com / By
[@michaelberhane](https://twitter.com/michaelberhane)



M

Medium ✓

@Medium

Words matter. Learn more at

Following



Tamyra Walker

@tamyraintech

#TechEquity as a stance, #STEM
inclusion as a platform, #Advocacy as a
vehicle of change.

Following

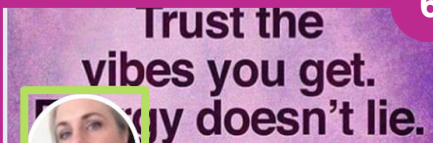
MB
WOMEN'S
SOCCER

CSUMB Women's Soccer

@CSUMBWSoccer

Everything women's soccer at Cal State

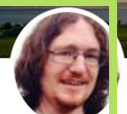
Following



the sarahjozket

@ketron_sarah Follows you

Following



Stuart Geiger ✓

@staeiou

computational ethnographer and

Following

6

32

Repeating Tasks

We are going to examine several ways of repeating code:

- While Loops
- For Loops

Some common scenarios

How do I...

1. Keep playing the beat over and over again?
2. Animate my creature?
3. Search through the data for keywords?
4. Render all of the photos in a list?
5. Draw 1,000 creatures?
6. Play all the notes in a list?
7. Find the biggest number in a list?

While Loops

Pseudocode

```
while (some condition is true) {  
    // execute code block  
}
```

Print the Statement

How do print this sentence over and over again?

```
console.log('Hey there! Hope you\'re doing OK!');
```

Print the Statement

Solution: How do print this sentence over and over again?

```
// using the node.js interpreter:  
while (true) { // the condition is always true  
    console.log('Hey there! Hope you\'re doing OK!')  
}
```

How do I make it print 10 times and then stop?

To do this, you have to answer two questions:

1. How do we figure out how many times the sample has printed?
2. How do we tell the interpreter to break out of the loop when it's printed 10 times?

How would you do it?

I'm going to run the program again, and you're going to figure out when it's printed the sentence 10 times...

...how did you do it?

Using variables to track **state**

- In computer science, the “state” of a program refers to its current values or contents
- If we want to know how many times something has happened, we have to use a variable to track the state of the program
- Each time the loop block executes, we can increment the variable so that we know how long it’s been iterating

Using variables to track state

What needs to change to get it to only print 10 times?

```
while (true) { // the condition is always true
  console.log('Hey there! Hope you\'re doing OK!')
}
```

Using variables to track state

What needs to change to get it to stop after 10 beats?

```
let counter = 0; ]———— Initialize iteration variable
```

```
while (counter < 10) { ]———— Specify halt condition
```

```
  console.log('Hey there! Hope you\'re doing OK!');
```

```
  counter += 1; ]————
```

```
}
```

Recall: If Statement Syntax

BLOCK

If the condition evaluates to True, the block executes. Otherwise, the block is skipped.

```
if (condition) {  
    statement 1  
    statement 2  
    ...  
}
```

CONDITION

Boolean expression that evaluates to True or False.

While Loop Syntax

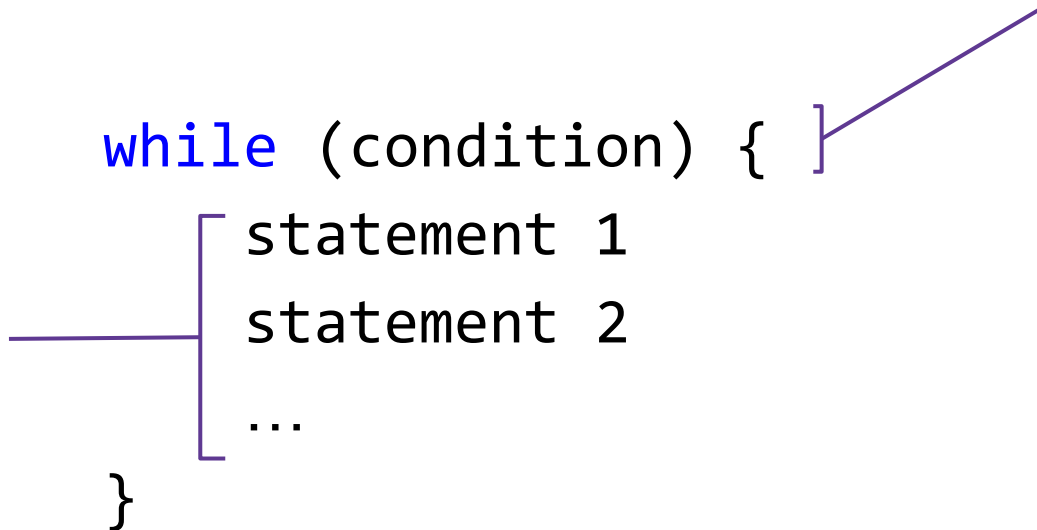
CONDITION

Boolean expression that evaluates to True or False.

BLOCK

While the condition evaluates to True, the block will continue to execute over and over. Otherwise, the block is skipped.

```
while (condition) {  
    statement 1  
    statement 2  
    ...  
}
```



Recall: Comparison Operators Evaluate to Either true or false

Comparison operators compare two operands according to a comparison rule and evaluate to either True or False (boolean)

Operator	Description
<code>==</code> <code>===</code>	If the values of two operands are equal, then the condition becomes true.
<code>!=</code>	If values of two operands are not equal, then condition becomes true.
<code>></code>	If the value of left operand is greater than the value of right operand, then condition becomes true.
<code><</code>	If the value of left operand is less than the value of right operand, then condition becomes true.
<code>>=</code>	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.
<code><=</code>	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.

Recall: Logical Operators Also Evaluate to Either True or False

Logical operators provide additional ways to determine whether something is true or false:

Operator	Description
&&	If both operands are true then the expression evaluates to true. Otherwise, the expression evaluates to false
	If either or both operands are true then the expression evaluates to true. If both operands are false, the expression evaluates to false
!	If the operand is false then the expression evaluates to true (and vice versa)

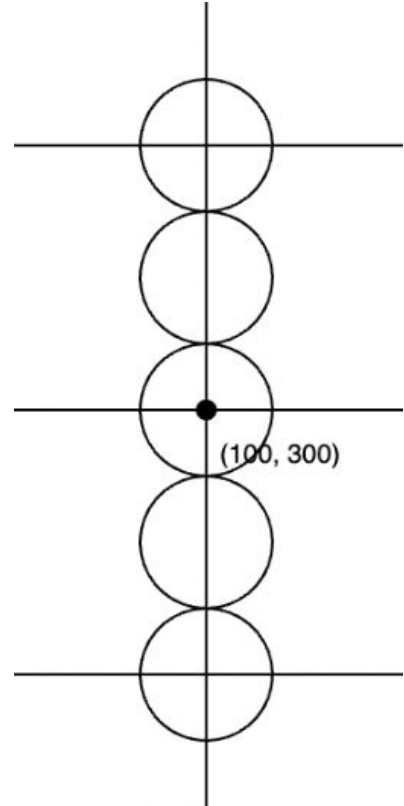
Outline

1. Review (Sample Quiz 3 questions)
2. Intro to while loops
- 3. Exercises: While loops**
4. Intro to arrays (also called lists) & objects
5. Aside: using random numbers

While Loop Exercise: Circles

01-circles

1. Open circles
2. See if you can use a while loop to recreate this functionality, where there is only one makeCircle function call that is repeated within a while loop.
3. Enhancements
 - a. Make the column span the entire screen
 - b. Make multiple columns
 - c. Change the size and the position of the circles



Animations are also loops!

- Animation involves drawing slightly different pictures at regular intervals
- In p5.js, any statement written inside the draw() function will be put inside a continuous animation loop (**02-animation-simple**)
- “Under the hood,” p5.js is probably doing something like this:

```
while(true) {  
    draw();  
    sleep(20);  
}
```

03-animation-while-loop

Outline

1. Review (Sample Quiz 3 questions)
2. Intro to while loops
3. Exercises: While loops
- 4. Intro to arrays (also called lists) & objects**
5. Aside: using random numbers

Lists (AKA “Arrays”)

```
const myList = ["charlie", "freddie", "lucy"]  
//access individual items:  
console.log(myList[0]);  
console.log(myList[1]);  
console.log(myList[2]);  
console.log(myList.length); //get length of list  
myList.push("jimena");      // append item to bottom  
myList.pop();               // remove item from bottom
```

0	"charlie"
1	"freddie"
2	"lucy"

Examples of Lists

1. A list of image urls (list of strings)
2. A list of names (list of strings)
3. A list of Tweets (list of objects)
4. A list of Songs (list of objects)

Objects: Syntax

What is an object?

Objects are a way to group variables and functions into a convenient package.

- Objects have “keys” (variables) and associated values.
- You can access the values stored in “keys” using “dot” notation.
- You can also assign object “keys” to new values using the assignment operator (=).

Example:

```
const ball1 = {  
  key — [ x: 300, ] — value  
          y: 150,  
          size: 130  
}
```

Objects: Example

```
const ball1 = {  
  x: 300,  
  y: 150,  
  size: 130  
}
```

```
const ball2 = {  
  x: 40,  
  y: 3450,  
  size: 50  
}  
console.log(ball1.x);  
console.log(ball2.x);  
console.log(ball2.size);  
  
ball2.size += 0.5;  
console.log(ball2.size);
```

Lists of Objects

It is also common to store data in lists of objects. In the following example, what do you think will be printed to the screen?:

```
const circleData = [  
  {x: 500, y: 50, d: 100, speedX: 3, speedY: 3},  
  {x: 300, y: 150, d: 50, speedX: 2, speedY: 0},  
  {x: 400, y: 400, d: 25, speedX: 0, speedY: 1},  
  {x: 40, y: 520, d: 80, speedX: 2, speedY: 0},  
  {x: 140, y: 120, d: 150, speedX: 0, speedY: -0.5},  
  {x: 350, y: 350, d: 70, speedX: 1, speedY: 1}  
];  
console.log(circleData[0]);  
console.log(circleData[0].speedX);  
console.log(circleData[2].d);  
console.log(circleData[4].x);
```

Exercises

List of Strings

- Open the **04-list-of-strings** folder in VS Code
- How could you print all of the names using a while loop?

List of Images

- Open the **05-photos** folder in VS Code
- How can you output all of the images?

List of Cards

- Open the **07-list-of-objects** folder in VS Code
- How can you output a card for each game player?

Code From Class

- **Section 1:** Game mechanics: <https://t.ly/H6bA>

Introduction to Homework 6

Creating a Photo Carousel

- Download the starter files
- Discuss:
 - how would you implement the forward() function?
 - How would you attach the event handler?
 - How would you implement the showImage() function

Outline

1. Review (Sample Quiz 3 questions)
2. Intro to while loops
3. Exercises: While loops
4. Intro to arrays (also called lists) & objects
- 5. Aside: using random numbers**

The Built-In Random Function

One important concept in computer science is the notion of randomness, which allows a programmer to simulate a number of different possible inputs quickly. Some applications:

- Position a bunch of shapes randomly on a screen
- Pick a note out of scale to simulate a jazz riff
- Simulate a probability distribution
- Make leaves on a tree have slightly different sizes, shapes, and angles to simulate the randomness of nature

Random Function Examples

Q: How do I generate a random number between 0 and 10?

```
const myNum = Math.floor(Math.random() * 11);
```

Q: How do I generate a random number between 5 and 15?

```
const myNum = Math.floor(Math.random() * 10) + 6;
```

Q: Can't I just make a function to do this?

```
function randomInt(min, max) {  
  // min and max included  
  return Math.floor(Math.random() * (max - min + 1) + min);  
}
```

Exercise 08-randomness

Open 08-randomness and take a look at **sketch.js** (ignore p5.js and utilities.js). Then, see if you can modify the code as follows:

1. Instead of always using the same fill color, randomly select a color from the colors array above.
2. Use a loop to draw many random circles.
3. Consider animating the drawing by putting your logic in the draw() function instead of the setup() function.
4. Experiment with drawing other shapes (squares, lines, triangles, etc.).

Exercise 09-circle-animation: Working with data

1. Study the circle animation. How does it work?
2. Convert the 5 variables that control the state of the circle (x, y, d, xSpeed, and ySpeed) into a “circle” object.
 - a. This enables us to more conveniently organize our variables if we have lots of variables.
3. Use a while loop to animate multiple circles by reading the **circleData** array.
4. Implement a click event handler that adds new circles to the animation.