Loops

Loop Statements: while & for loop

Textbook Readings:
Chapter 4, Examples 4-5, 4-6, 4-7, 4-8, 4-9, 4-10, 4-11, 4-12, 4-13
Chapter 5, Examples 5-22, 5-23

Coding Train Videos:
4.1: while and for Loops https://youtu.be/cnRD9o6odjk
4.2: Nested Loops https://youtu.be/1c1_TMdf8b8
Which of the instruction sets is needed to create this?

(a) if (mouseY > 50 && mouseX > 50) {
    fill("#0000FF"); // blue
}

(b) if (mouseX > 50 && mouseY > 50) {
    fill("#FF0000"); // red
}

(c) if (mouseY > 50 && mouseX > 50) {
    noFill();
}

(d) if (mouseX > 50 && mouseY < 50) {
    fill("#FF0000"); // red
}

(e) if (mouseY < 50 && mouseX > 50) {
    fill("#0000FF"); // blue
}

(f) if (mouseX < 50 && mouseY > 50) {
    noFill();
}

The canvas size is 100 x 100.

A (a) (b) (c)  
B (d) (e) (f)  
C (a) (d) (f)  
D (b) (e) (f)  

NOTE: Find the best answer.
What is loop?

This is endless looped time.

Credit: https://www.youtube.com/watch?v=tZyUxqBvMBE (0:38 - 4:03)
What is loop?

- Loop refers to a process that repeats
Thus `draw()` is also a loop

```
function setup() {
  createCanvas(400, 400);
}

function draw() {
}
```
When does loop stop?

While Dormammu refuses, Dr. Strange says, “I have come to bargain!”

Dr. Strange dies.

1: Dormammu, I have come to bargain!

2: Dormammu, I have come to bargain!

3: Dormammu, I have come to bargain!
Loop repeats while condition is true

While Dormammu refuses,

Dr. Strange says, “I have come to bargain!”

Dr. Strange dies.
While Dormammu refuses,
Dr. Strange says, “I have come to bargain!”
Dr. Strange dies.
So two things to remember

- Loop...
  - **repeats** while condition is **true**
  - **stops** when condition becomes **false**
Thus a `while` loop statement is expressed as follows:

```
while(condition)
{
    code block
}
```
For example, we can express “endless looped time” as

```python
while(dormammu_refuses) {
    print(“I have come to bargain!”);
    dies();
}
```
Why use loop?

- It does repetitive tasks for us
*Untitled (1969), Donald Judd*
while (expression) {
    statements
}

Starter: https://editor.p5js.org/sanghosuh/sketches/NG7KcM0a

https://editor.p5js.org/sanghosuh/sketches/2GkYW5J7
Four Loop Questions

1. What do I want to repeat?

2. What do I want to change each time?

3. Where does it start, how does it change?

4. How long should it repeat?

while ( ) {

}

}
Four Loop Questions

1. What do I want to repeat?
   - a rect
2. What do I want to change each time?
   - the y position of the rect
3. Where does it start, how does it change?
   - start at y = 10, draw a rect every 20 pixels
4. How long should it repeat?
   - until it reaches the bottom of the canvas

while ( ) {

}

Which we express as follows:

```plaintext
y = 10;
while (y < height) {
    rect(50, y, 50, 10);
    y += 20;
}
```
See “10 Loops (trace)”
Chrome Debugger Demo

- Watch the loop execute using an “interactive debugger”

Guide to using Chrome Debugger
https://javascript.info/debugging-chrome

Code to debug (but run this code from Processing IDE):
https://editor.p5js.org/cs105/sketches/VAeK60UBn
**spaced-lines**

draw lines all the way across canvas with equal spacing (20) using:

- line position variable

Variations:

- margin
- right to left
- calculate spacing based on width to draw exact number of lines.

[https://editor.p5js.org/cs105/sketches/OmmqVprVz](https://editor.p5js.org/cs105/sketches/OmmqVprVz)
number-of-lines

draws an exact number of lines (7) with fixed spacing (20) using:

- counter variable
  (= "integer variable used to keep track of number of times piece of code is executed")

- line position variable

https://editor.p5js.org/cs105/sketches/ZyPAKsioE
draw() frames vs. while loop iterations

- the code block in `draw` is repeated every frame
- the code block in `while` is repeated every loop iteration

```javascript
function draw() {
  while (x < width) {
  }
}
```
What do you see after 5 frames?

```javascript
let x = 0;
function draw() {
  background(200);
  x = 20;
  while (x < width) {
    ellipse(x, 50, 20, 20);
    x = x + 20;
  }
}
```
What do you see after 5 frames?

```javascript
let x = 20;
function draw() {
  background(200);
  while (x < width) {
    ellipse(x, 50, 20, 20);
    x = x + 20;
  }
}
```
What do you see after 5 frames?

```javascript
let x = 0;
function draw() {
  background(200);
  x = 20;
  while (x < width) {
    ellipse(x, 50, 20, 20);
  }
  x = x + 20;
}
```

This is an infinite loop.

The answer is E, but why?
Common Logic Errors Leading to Infinite Loops

Adding a semicolon after the boolean expression means **no code block.** Without a code block, the loop variable won't update.

```cpp
x = 0;
while (x < width); {
    line(x, 10, x, height - 10);
    x = x + 10;
}
```

The while statement will never end, and the program will freeze (infinite loop).

Fix this by removing the semicolon after the boolean expression.

```cpp
x = 0;
while (x < width) {
    line(x, 10, x, height - 10);
    x = x - 10;
}
```

The loop variable is updated such that the loop condition will always be true.

The while statement will never end, and the program will freeze (infinite loop).

Fix this by ensuring that your code eventually makes the loop repetition expression false.

Code: [https://editor.p5js.org/sanghosuh/sketches/fBeFS3zw](https://editor.p5js.org/sanghosuh/sketches/fBeFS3zw) (Be careful! Running this program will result in your browser crashing)
Save your work because infinite loop can delete it

- Online editor or Processing IDE can crash if your program has infinite loop
- You will accidentally write infinite loops
  - Make sure to save your work frequently
Debugging Loops

- Test a single loop operation first (without a loop)
- Simplify the loop operation
- Slow down draw frames with `frameRate(1)`
- Use `print()`

```java
x = 0;
print("start loop"); // debug
while (x < width) {
    print("loop", x); // debug
    line(x, 10, x, height - 10);
    x = x + 10;
}
print("done loop"); // debug
```
gradient

create a vertical grayscale gradient

eexample of varying more than one variable

https://editor.p5js.org/cs105/sketches/U_fvzbRqx
(demos) convert while loop to for loop

```javascript
let i = 0;
function setup() {
  while (i < 10) {
    print("while loop ", i);
    i = i + 1;
  }
}

function setup() {
  for (let i = 0; i < 10; i++) {
    print("for loop ", i);
  }
}
```
Assignment Operator “Short Forms”

These all add 1 to x (they are all equivalent):

\[
x = x + 1;
\]
\[
x += 1;
\]
\[
x++;
\]

These both add 5 to x (they are both equivalent):

\[
x = x + 5;
\]
\[
x += 5;
\]

Other examples

\[
// same as x = x + 10 * y;
\]
\[
x += 10 * y;
\]
\[
// same as x = x + random(-2, 2);
\]
\[
x += random(-2, 2);
\]
More Assignment Operator “Short Forms”

x--;

x -= 10; // subtract 10 from x

x *= 10; // multiply x by 10

x /= 10; // divide x by 10
Anatomy of a For Loop

```
for (let y = 10; y < height; y += 20) {
    rect(50, y, 50, 10);
}

```

code block
When to use `for` or `while` loop?

`for` is a short form for `while`, and they’re interchangeable. (You can *always* use `while` if you want)

```javascript
y = 10;

while (y < height) {
  rect(50, y, 50, 10);
  y += 20;
}
```

```javascript
for (let y = 10; y < height; y += 20) {
  rect(50, y, 50, 10);
}
```
When to use *for* or *while* loop?

Both are functionally equivalent, but sometimes one is easier to use in certain cases:

- **Use *for* to loop an exact number of repetitions:**
  - I want 10 pacmen
  - I want 3 lines

- **Use *for* to update by same amount:**
  - I want to count by 10 from 0 to 100
  - I want lines spaced 10 pixels apart the width of the canvas

- **Use *while* to update by different amounts and you don’t need a predetermined number of repetitions:**
  - I want lines spaced randomly 2 to 10 pixels apart over the whole width of the canvas
When to use **for** or **while** loop?

- Do you know how many times the loop should repeat?
  - if no, then **while** loop
  - if yes, then **for** loop
number-of-lines-for

Use for to loop an exact number of times (a for loop version of number-of-lines)

"what do I want to change?" -> counter variable

```javascript
// position variable
x = 0;

for (let i = 0; i < num; i++) {
  line(x, 10, x, height - 10);
  x += spacing;
}
```

Starter: [https://editor.p5js.org/cs105/sketches/ZyPAKsioE](https://editor.p5js.org/cs105/sketches/ZyPAKsioE)

[https://editor.p5js.org/cs105/sketches/WWV7CJ-MI](https://editor.p5js.org/cs105/sketches/WWV7CJ-MI)
spaced-lines-for

Use for to update by same amount (a for loop version of spaced-lines)

“what do I want to change?” -> x position of lines

```javascript
for (let x = 0; x < width; x += spacing) {
    line(x, 10, x, height - 10);
}
```

Starter: [https://editor.p5js.org/cs105.sketches/OmmqVprVz](https://editor.p5js.org/cs105.sketches/OmmqVprVz)

[https://editor.p5js.org/cs105.sketches/MtjMGyIBD](https://editor.p5js.org/cs105.sketches/MtjMGyIBD)
Summary

- Two types of loops
  - while loop
  - for loop
- Debugging loops
- Demos on how to use while and for loop
- When to use for or while loop
  - they are interchangeable
Survey

▪ “Is there something we can do to help?”

▪ Bonus 1% for completing the survey by this Friday 11:59 pm: https://forms.gle/re3dcZmuhqqtYTg17

▪ It should take approximately 5 min. to complete
let i;
let x;

function draw() {
    background(200);
    i = 0;
    x = 20;
    while (i < 4) {
        ellipse(x, 50, 20, 20);
        i = i + 1;
        x = x + 20;
    }
}