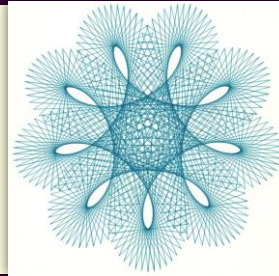


CS 105: Introduction to Computer Programming (using JavaScript)

Loops

Instructor: **Joel Castellanos**
e-mail: joel@unm.edu
Web: <http://cs.unm.edu/~joel/>
Office: Farris Engineering Center 2110



2/22/2018

Quiz: Khan Academy: Intro to While Loop

What does this JavaScript program display on the canvas?

```
fill(0, 0, 0);  
var message = "y is now ";  
var y = 10;  
while(y < 60)  
{  
    text(message + y, 30, y);  
    y += 15;  
}
```

a) y is now 10

b) y is now 10 25 40 55

c) y is now 30 10 30 25 30 40 30 55

d) y is now 10
y is now 25
y is now 40
y is now 55

2

Print Multiples of 3 from 3 to 30

```
1) fill(0,0,0);
2)
3) text( 3, 0, 10);
4) text( 6, 0, 10);
5) text( 9, 0, 10);
6) text(12, 0, 10);
7) text(15, 0, 10);
8) text(18, 0, 10);
9) text(21, 0, 10);
10) text(24, 0, 10);
11) text(27, 0, 10);
12) text(30, 0, 10);
```

Canvas:



3

Print Multiples of 3 from 3 to 30

```
1) fill(0, 0, 0);
2)
3) text( 3, 10, 10);
4) text( 6, 20, 10);
5) text( 9, 30, 10);
6) text(12, 40, 10);
7) text(15, 50, 10);
8) text(18, 60, 10);
9) text(21, 70, 10);
10) text(24, 80, 10);
11) text(27, 90, 10);
12) text(30, 100, 10);
```

Canvas:



4

Print Multiples of 3 from 3 to 30

```
1) fill(0, 0, 0);
2)
3) text( 3, 0, 10);
4) text( 6, 20, 10);
5) text( 9, 40, 10);
6) text(12, 60, 10);
7) text(15, 80, 10);
8) text(18, 100, 10);
9) text(21, 120, 10);
10) text(24, 140, 10);
11) text(27, 160, 10);
12) text(30, 180, 10);
```

```
1) fill(0, 0, 0);
2)
3) var x = 0;
4) var n = 3;
5) while(n <= 30)
6) {
7)   text(n, x, 10);
8)   n += 3;
9)   x += 20;
10) }
```

Canvas: 3 6 9 12 15 18 21 24 27 30

5

Quiz: What is displayed on the Canvas?

```
textSize(30);
fill(0,0,0);
var message = "y=";
var y = 100;
while(y < 200)
{
  text(message + y, 30, y);
  y += 25;
}
```

(a)

```
y=125
y=150
y=175
```

(b)

```
y=100
y=125
y=150
y=175
```

(c)

```
y=100
y=125
y=150
y=175
y=200
```

(d)

```
y=100, 30, 100
y=125, 30, 125
y=150, 30, 150
y=175, 30, 175
y=200, 30, 200
```

6

Print the Factors of an Integer

- Examples: num= 10 : Factors: 1, 10, 2, 5
- num = 12: Factors: 6, 2, 12, 1, 3, 4

- Algorithm:

var i = 1.

Loop while i is less than or equal to num.

if i divides evenly into num then print i.

Add 1 to i.

7

Modulus Operator

```
1) fill(0, 0, 0);
2) text( 7 % 3, 10, 10);
3) text( 8 % 3, 10, 30);
4) text( 9 % 3, 10, 50);
5) text(10 % 3, 10, 70);
```

| |
|---|
| 1 |
| 2 |
| 0 |
| 1 |

8

Quiz: What is displayed on the Canvas?

- 1) `fill(0, 0, 0);`
- 2) `text(22 % 5, 10, 10);`

(a)

2

(b)

4

(c)

4.4

(d)

4
10
10

(e)

4
5
10
10

9

Quiz: What is displayed on the Canvas?

- 1) `var brenda = 1;`
- 2) `fill(0, 0, 0);`
- 3) `while (brenda < 5)`
- 4) `{`
- 5) `text(7 % brenda, 10, 15*brenda);`
- 6) `brenda += 1;`
- 7) `}`

(a)

0
1
1
3

(b)

1
2
3
4

(c)

0
1
2
3

(d)

1
7

(e)

7

10

Print the Factors of an Integer

```
1) fill(0, 0, 0);  
2) var y = 10;  
3) var num = 50;  
4) var i = 1;  
5) while (i <= num)  
6) {  
7)     if (num % i === 0)  
8)     { text(i, 10, y);  
9)         y += 20;  
10)    }  
11)    i +=1;  
12) }
```

Sets font color to black.

In JavaScript, % is the modulus operator. It returns the remainder.

If a large font, is used, more than 20 will need to be added to y each time a new factor is found and displayed.

11

Getting Input form an HTML Page

HTML

```
<body>  
<h1>Enter a <i>positive integer</i> and I  
will display its factors:  
    <input type="text"  
          style="font-size: 100%"  
          id="inputNum"  
          value="100">  
</h1>
```

JavaScript

```
var inputNum =  
    document.getElementById("inputNum");  
var num = Number(inputNum.value);
```

12

Print the *Prime* Factors of an Integer

- 1 is NOT a prime number. The first 20 prime numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71.
- Examples: num= 10 : Prime Factors: 2, 5
num = 12: Prime Factors: 2, 2, 3
- Algorithm:
var **prime** = 2.
Loop while **prime** <= the ^{Why?}square root of **num**.
if **prime** divides evenly into **num** then
 Print **prime**.
 Change **num** to equal **num** divided by **prime**.
if **prime** does NOT divide evenly into **num** then
 add 1 to **prime**.

Why will this work? For example, if **num** = 16, then 4, which divides 16, yet is not a prime, should not (will not) get printed.

13

Print the Factors of an Integer

```
1) var y = 10;  
2) var num = 50;  
3) var prime = 2;  
4) while (prime <= sqrt(num))  
5) {  
6)     if (num % prime === 0)  
7)     { text(prime, 10, y);  
8)       y += 30;  
9)       num = num / prime;  
10)    }  
11)   else  
12)   { prime +=1;  
13)   }  
14) }  
14 15) if (num > 1) text(num, 10, y);
```

Note that each time a prime factor is found and printed (in line 7), **prime** is factored out of **num** (line 9: **num** = **num** / **prime**). Thus, each time a factor is found, the upper limit of the while loop gets smaller.

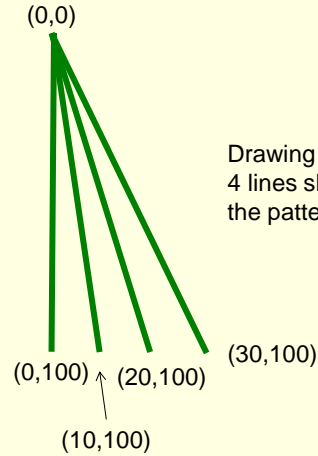
Contrariwise, each iteration when the current **prime** is NOT a factor of the remaining **num**, **prime** is increased by 1.

Why is this here?

14

What Pattern Does this Draw?

```
strokeWeight(2);  
stroke(0,200,0);  
line(0,0, 0, 100);  
line(0,0, 10, 100);  
line(0,0, 20, 100);  
line(0,0, 30, 100);  
line(0,0, 40, 100);  
line(0,0, 50, 100);  
line(0,0, 60, 100);  
line(0,0, 70, 100);  
line(0,0, 80, 100);  
line(0,0,100, 100);  
line(0,0,110, 100);  
line(0,0,120, 100);
```



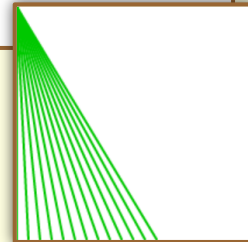
Drawing the first
4 lines shows
the pattern.

15

Use a Loop to Shorten Repeated Code

```
strokeWeight(2);  
stroke(0,200,0);  
line(0,0, 0, 200);  
line(0,0, 10, 200);  
line(0,0, 20, 200);  
line(0,0, 30, 200);  
line(0,0, 40, 200);  
line(0,0, 50, 200);  
line(0,0, 60, 200);  
line(0,0, 70, 200);  
line(0,0, 80, 200);  
line(0,0,100, 200);  
line(0,0,110, 200);  
line(0,0,120, 200);
```

```
strokeWeight(2);  
stroke(0,200,0);  
var x = 0;  
  
while (x <= 120)  
{  
  line(0,0, x, 200);  
  x += 10;  
}
```

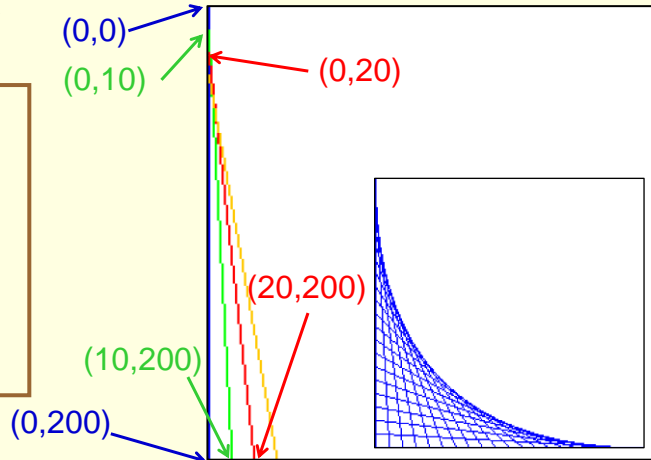


16

Deriving the Code to Draw a Pattern

```
8) var i = 0;  
9) while (i<=200)  
10) { line(0, i, i, 200); //x1,y1,x2,y2  
11)   i += 10;  
12) }
```

(x_1, y_1) to (x_2, y_2)
(0,0) to (0,200)
(0,10) to (10,200)
(0,20) to (20,200)
(0,30) to (30,200)



17

Pole: Internet Access

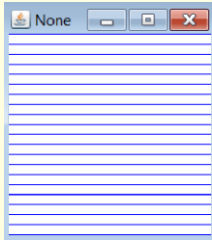
In order to access to the Internet within the United States borders, the government should:

- Require a national ID (or bio ID) and password be entered.
- Require a card with a unique ID chip be inserted into the computer and remain inserted while access is allowed.
- Hold hosting sites responsible for all content posted on those sites.
- Leave Internet access as it is with pseudo anonymity: casual users can be easily tracked by companies, and on a need-to-know basis, the government trace anything they want.
- Enable methods for true anonymity.

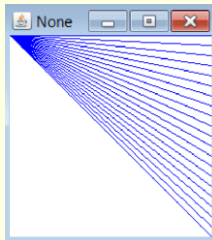
18

Quiz: Which Image is Drawn?

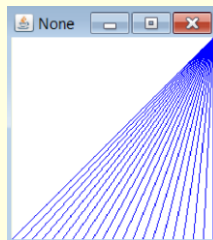
```
//Assume the canvas is 200x200 pixels
var i = 0;
while (i<=200)
{ line(200, 0, i, 200); //x1,y1,x2,y2
  i += 10;
}
```



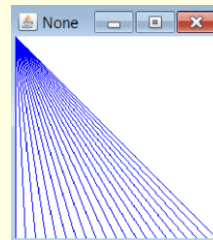
a)



b)



c)

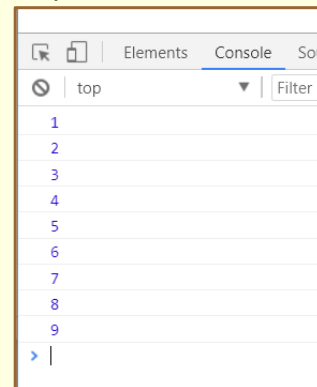


d)

19

For Loop

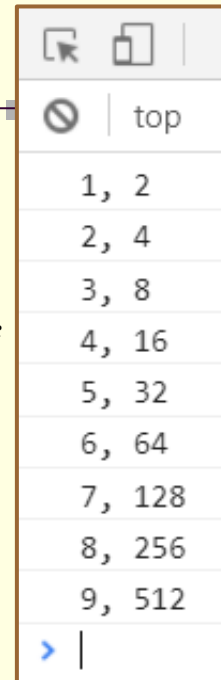
```
<script>
  function setup()
  {
    for (var i=1; i<10; i++)
    {
      console.log(i);
    }
  }
</script>
```



20

For Loop

```
var x = 1;
for (var i=1; i<10; i++)
{
    x = x * 2;
    console.log(i + ", " + x);
}
```



| top | |
|--------|--|
| 1, 2 | |
| 2, 4 | |
| 3, 8 | |
| 4, 16 | |
| 5, 32 | |
| 6, 64 | |
| 7, 128 | |
| 8, 256 | |
| 9, 512 | |
| > | |

21

For Loop

```
<script>
function setup()
{
    var x = 1;
    for (var i=1; i<=10; i++)
    {
        x = x * 2;
        console.log(i + ", " + x);
    }
}
```

22

