

The background features a silver laptop with a glowing screen, set against a dark blue background with abstract, glowing light trails in orange and blue. The text 'Nested Loops' is centered within a blue-bordered box.

# Nested Loops

# Plan for today

Green Screen

Single looping: a deeper look

Nested looping

Drawing grids



# Julia in the past



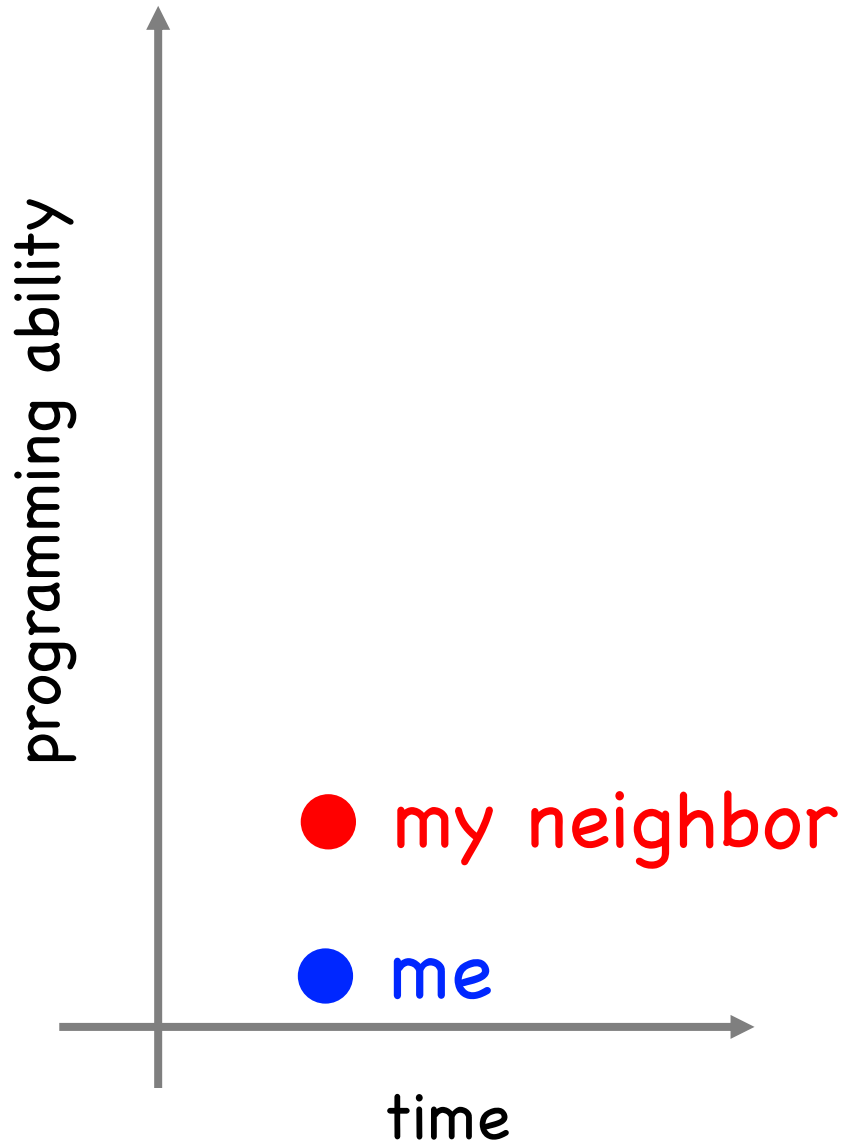


# Julia in the past





# The beginning of my journey



# The beginning of my journey



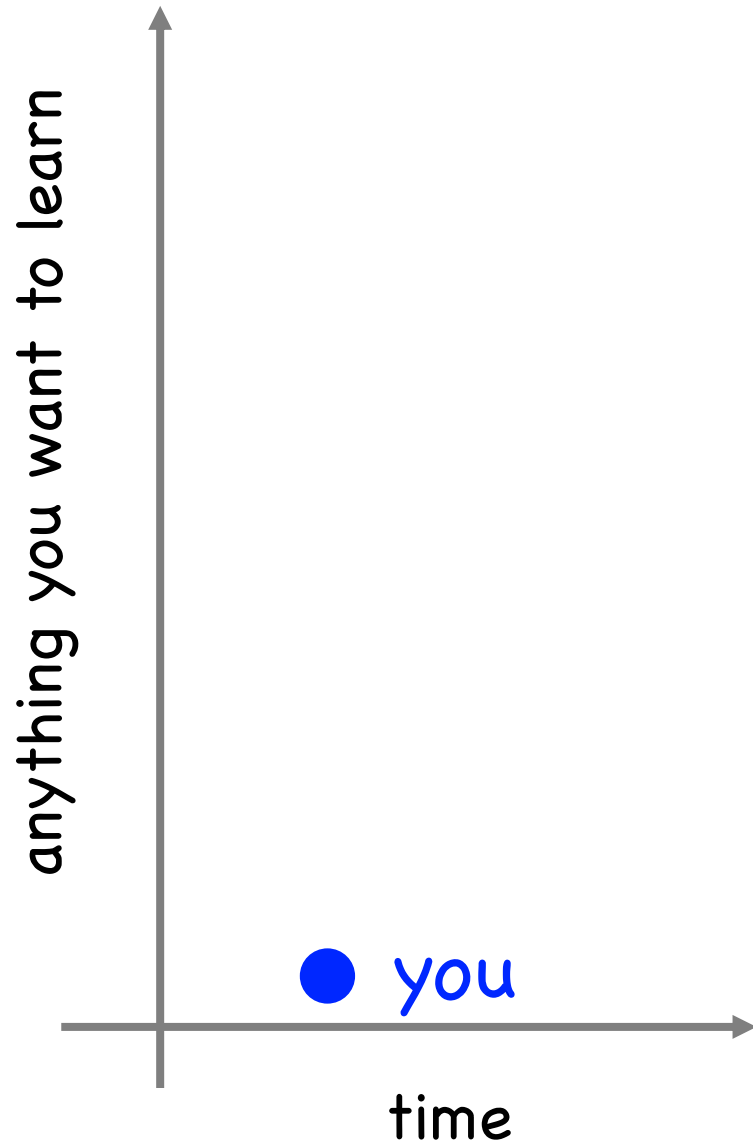
The beginning is hard  
for everyone.

We are learning  
an entirely new  
way to think!

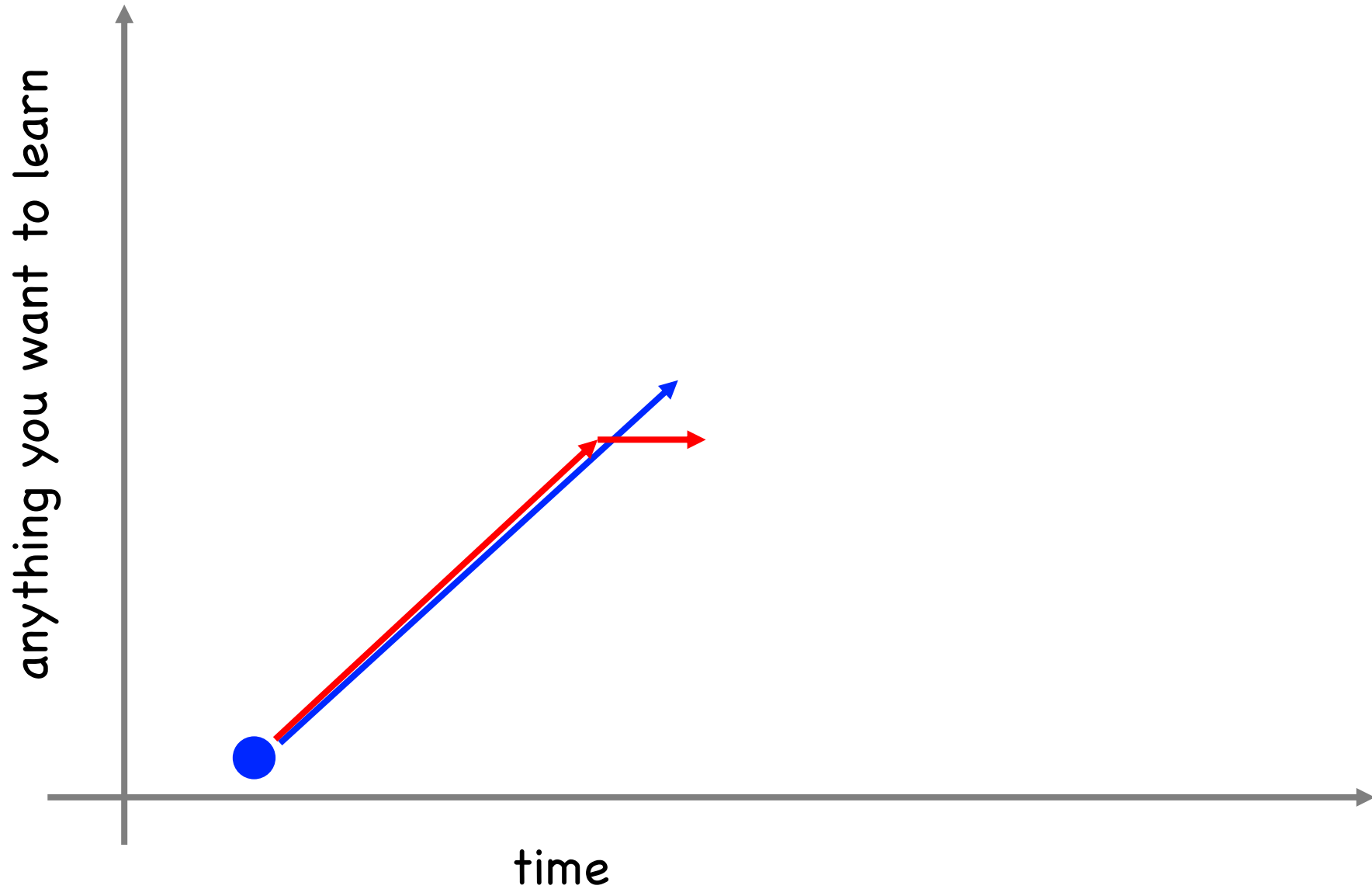
10x better than me?



# Think about yourself

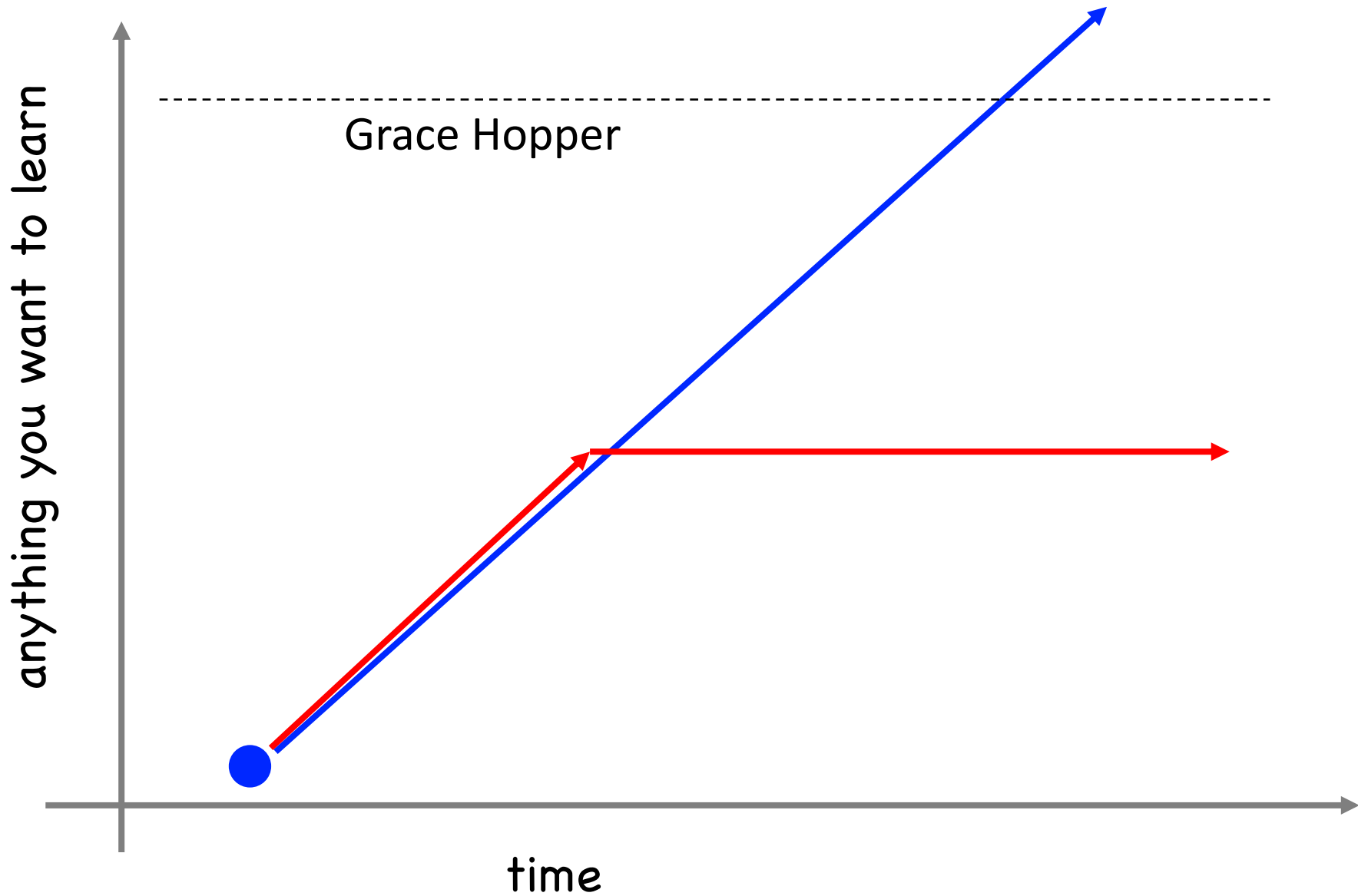


# There are many learning paths

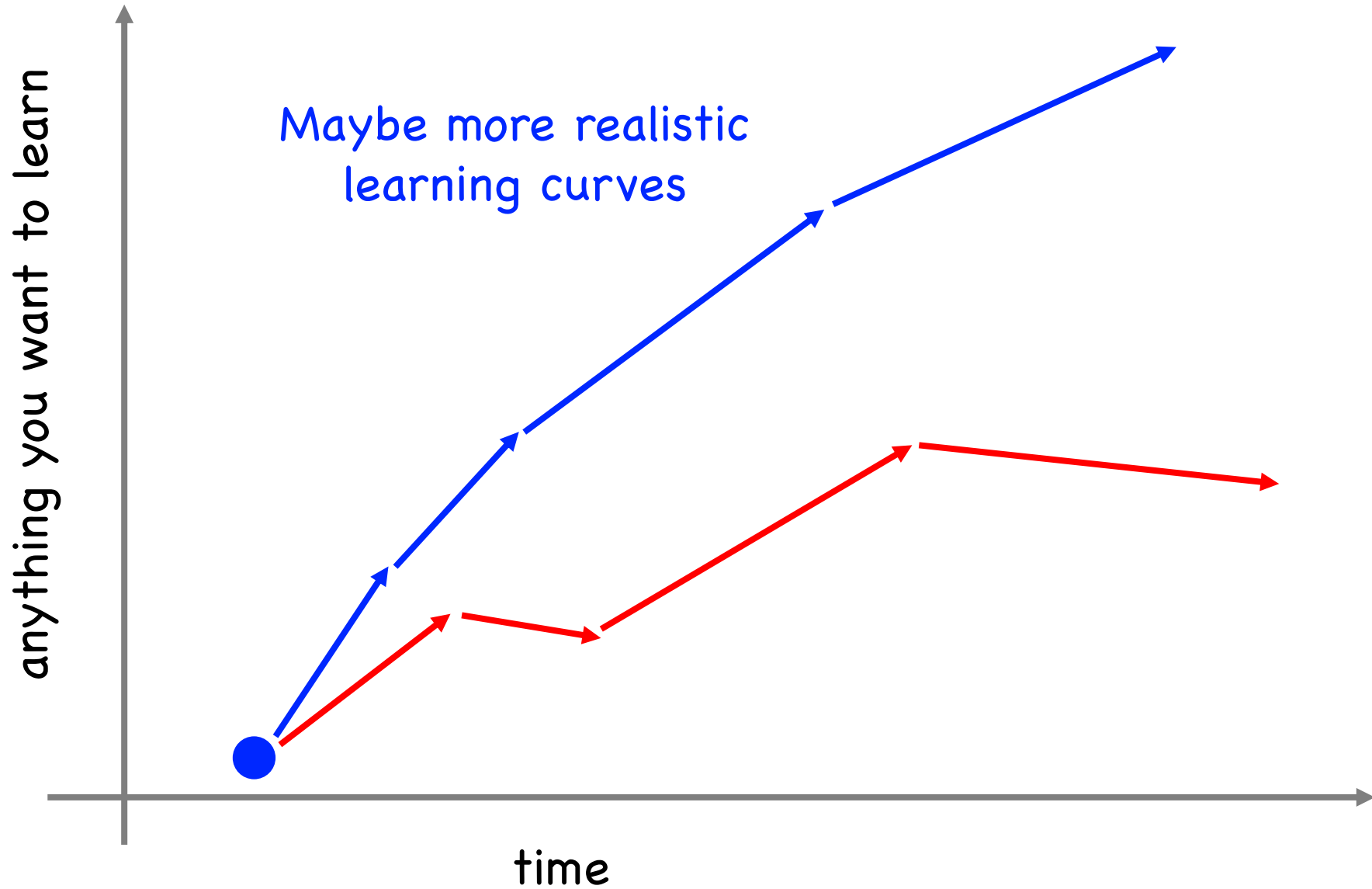




# There are many learning paths

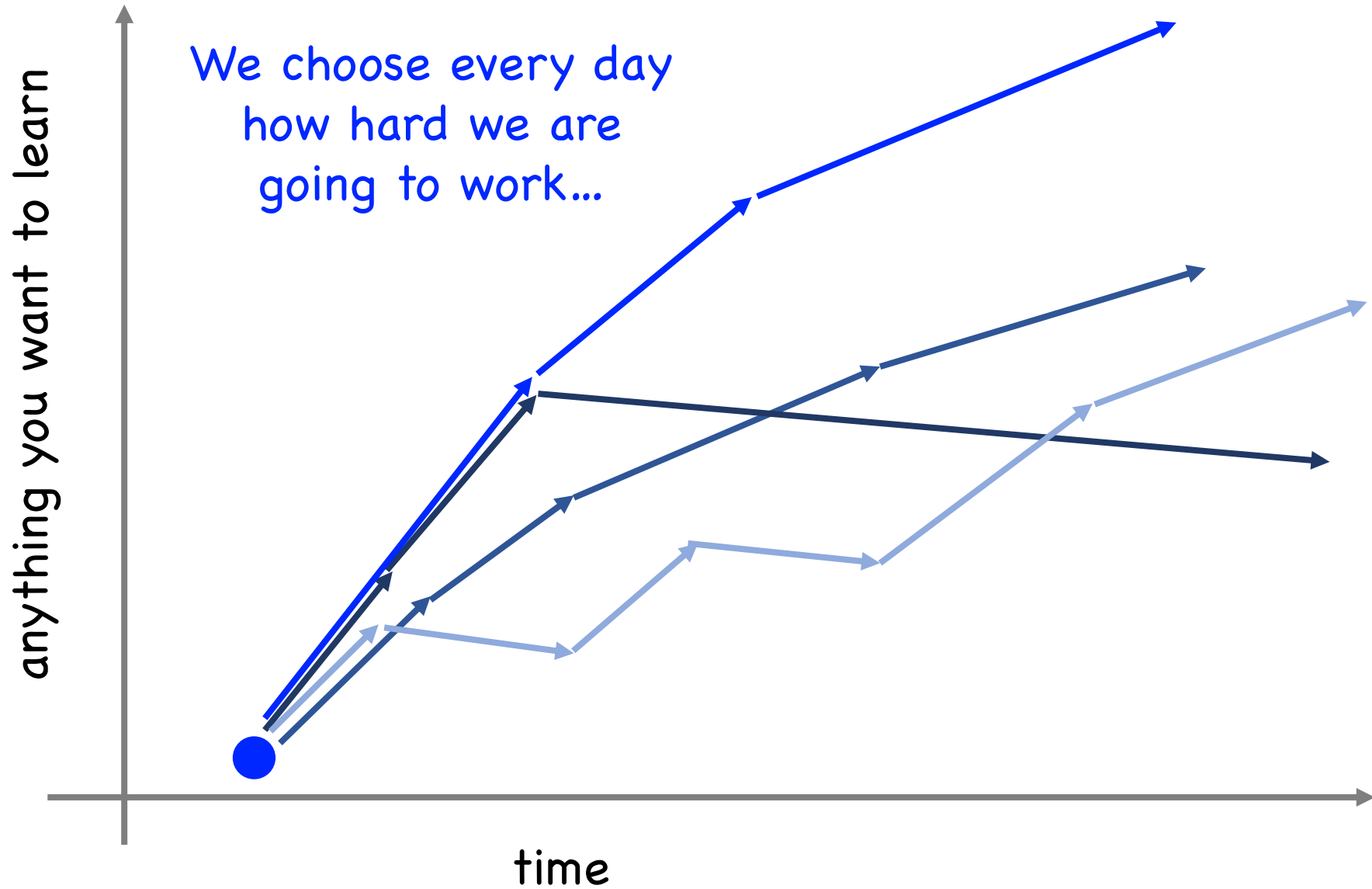


# There are many learning paths

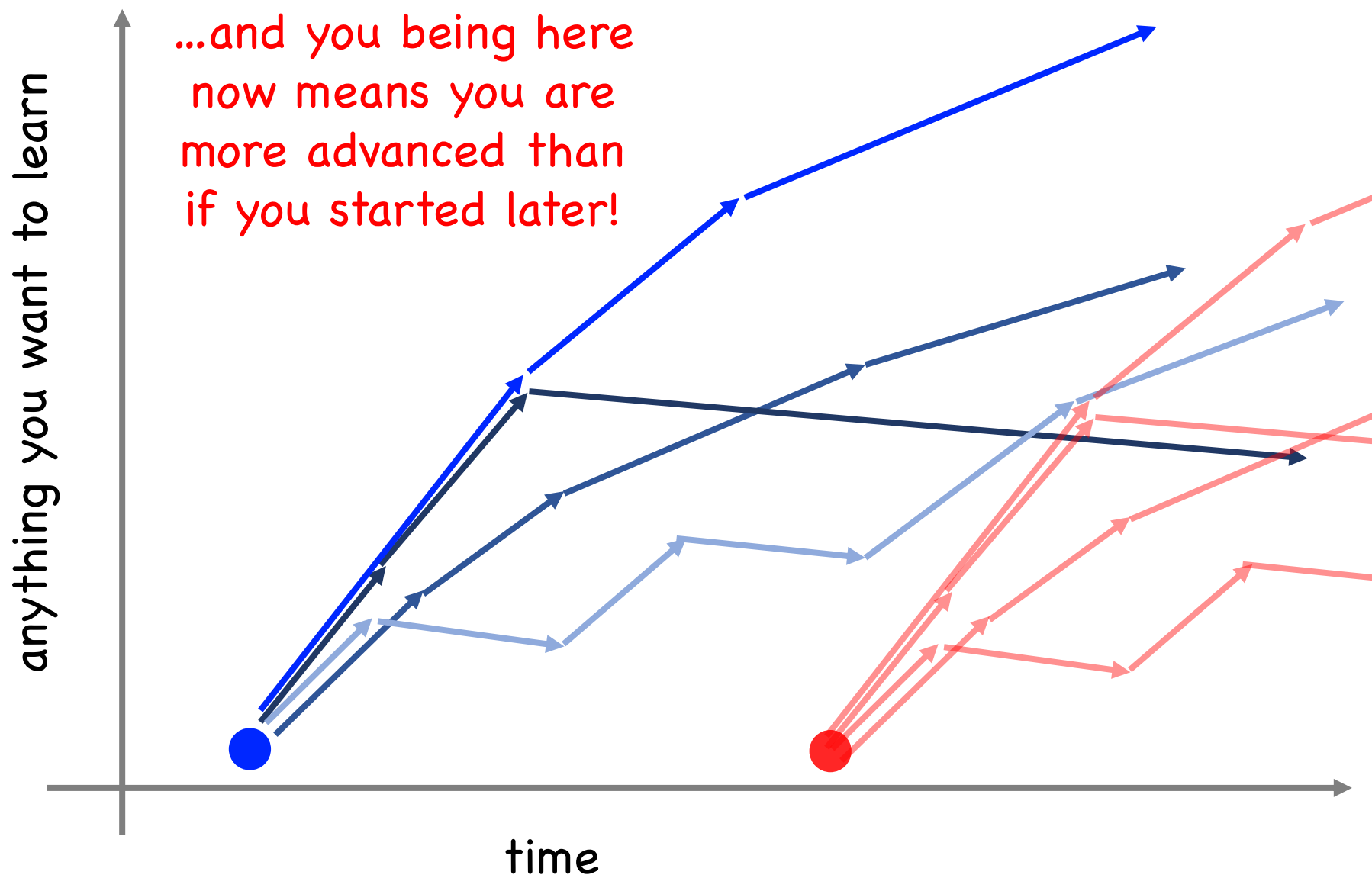




# There are many learning paths



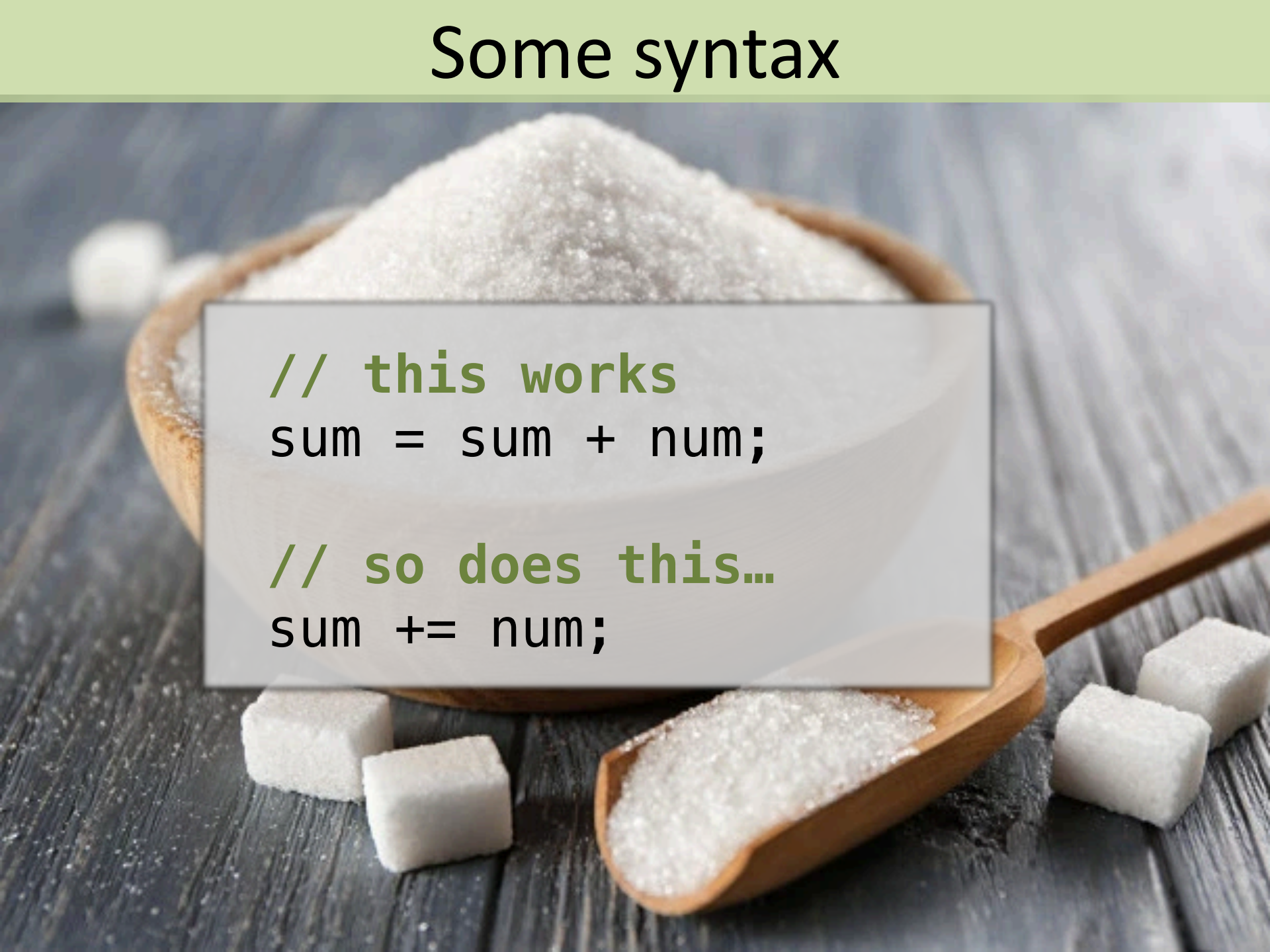
# There are many learning paths





If you want to do  
something difficult,  
what's important is how  
much you learn each day,  
not how much you know when  
you are 17.

# Some syntax



```
// this works  
sum = sum + num;  
  
// so does this...  
sum += num;
```

# Some syntax

```
// this works  
sum = sum + 1;
```

```
// so does this...  
sum += 1;
```

```
// and this does too  
sum++;
```



# Some syntax

```
// this works  
num = num - 1;
```

```
// so does this...  
num -= 1;
```

```
// and this does too  
num--;
```

How do you print “Czech this out!” 100 times?



# For loop

```
public void run() {  
    for(int i = 0; i < 100; i++) {  
        println("Czech this out!");  
    }  
}
```

# For loop

Executed once at  
the beginning  
of the loop

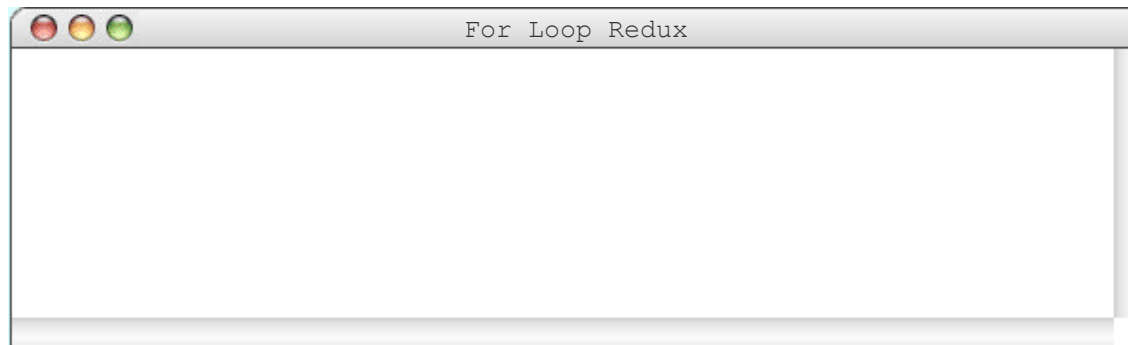
Run the body of  
the loop if  
this is true

Executed every  
time the loop  
finishes

```
for(int i = 0; i < 100; i++) {  
    println("Czech this out!");  
}
```

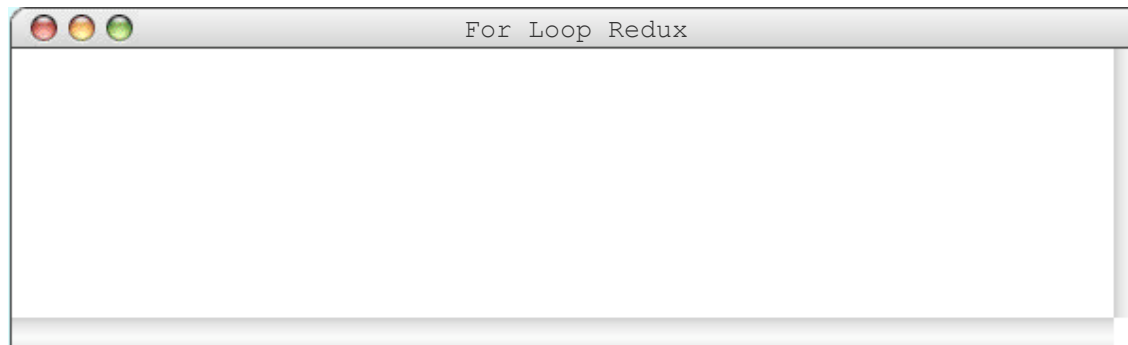
# For loop

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

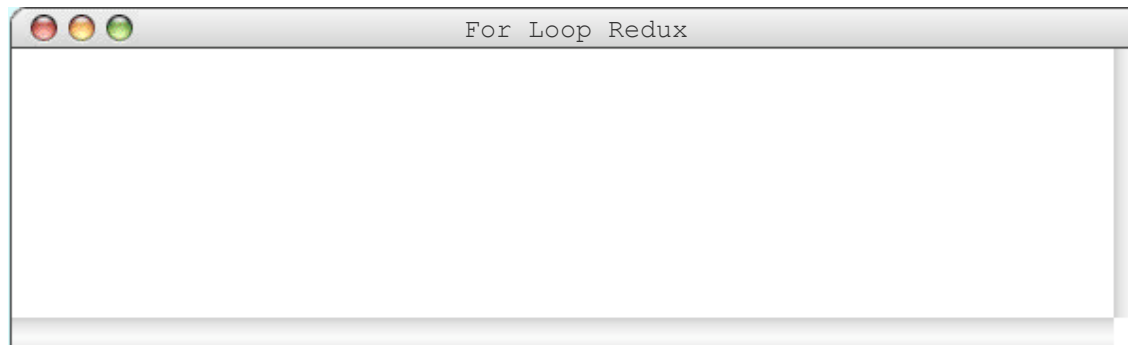
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 0

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```

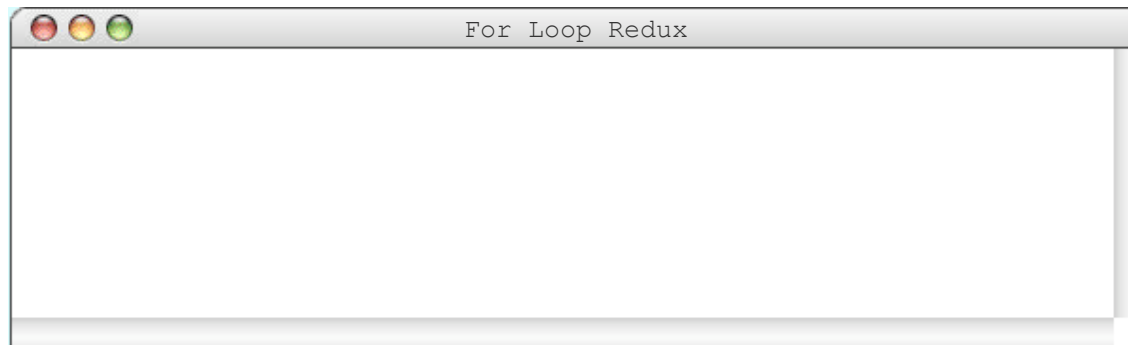




# For loop

i 0

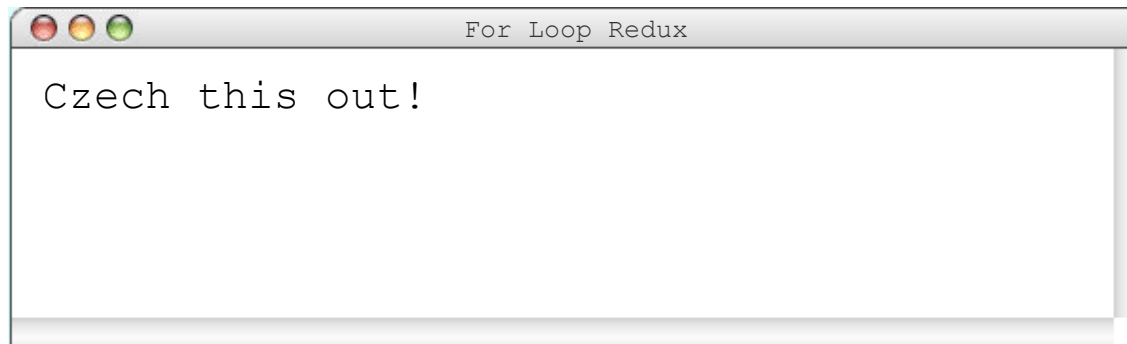
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 0

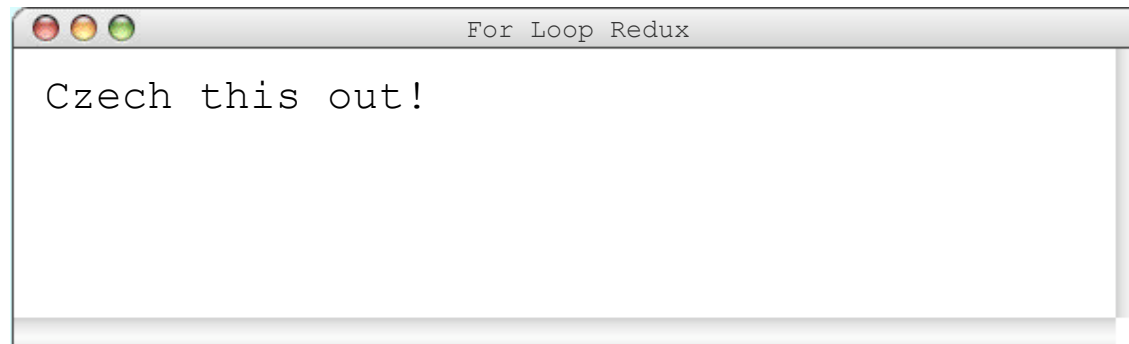
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 1

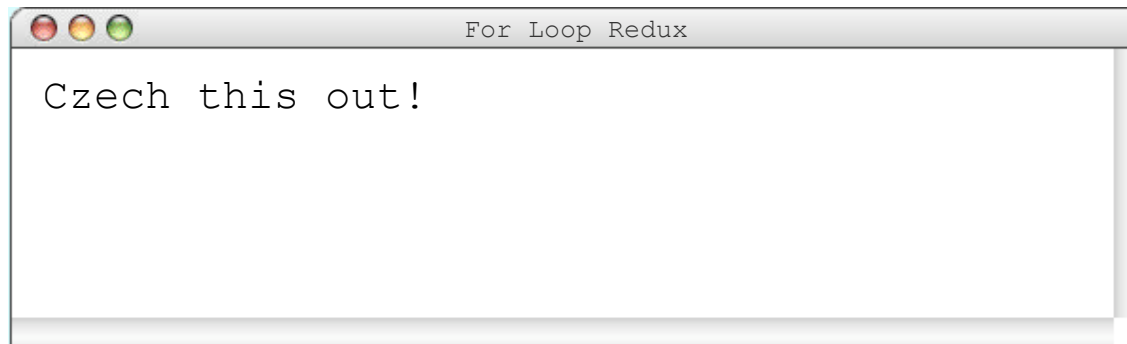
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 1

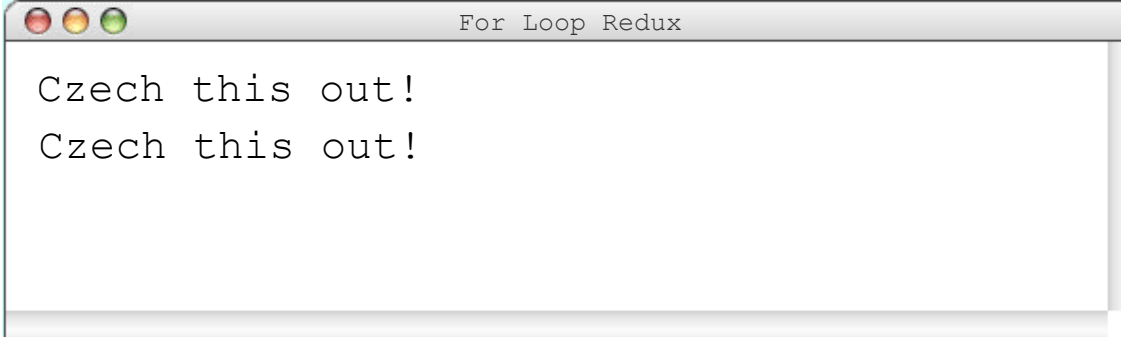
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 1

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



A terminal window titled "For Loop Redux" with three colored window control buttons (red, yellow, green) in the top-left corner. The window contains two lines of text: "Czech this out!" followed by a blank line, and then "Czech this out!" followed by a blank line.

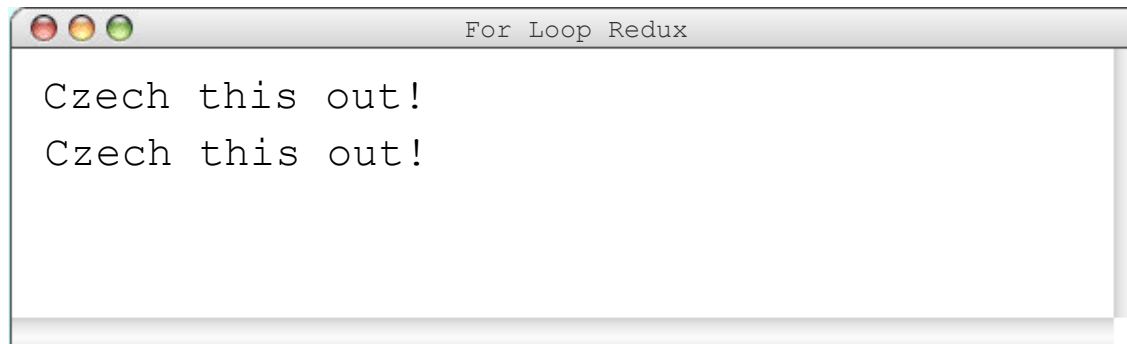
```
For Loop Redux  
Czech this out!  
Czech this out!
```



# For loop

i 2

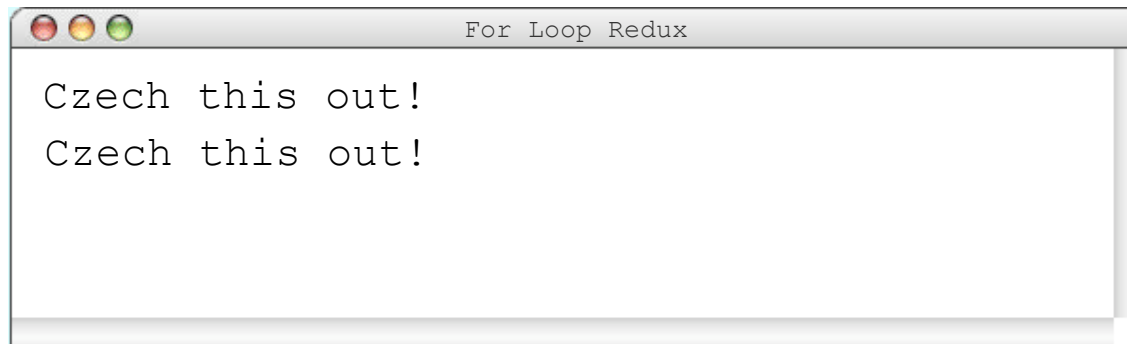
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 2

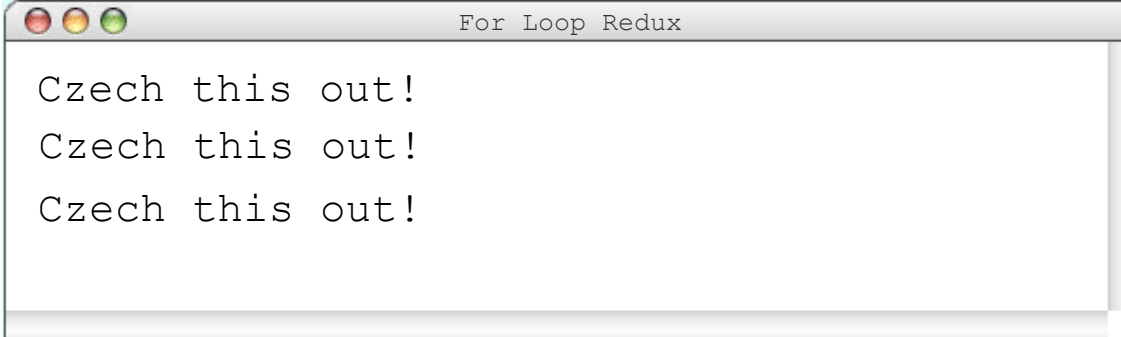
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



# For loop

i 2

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```

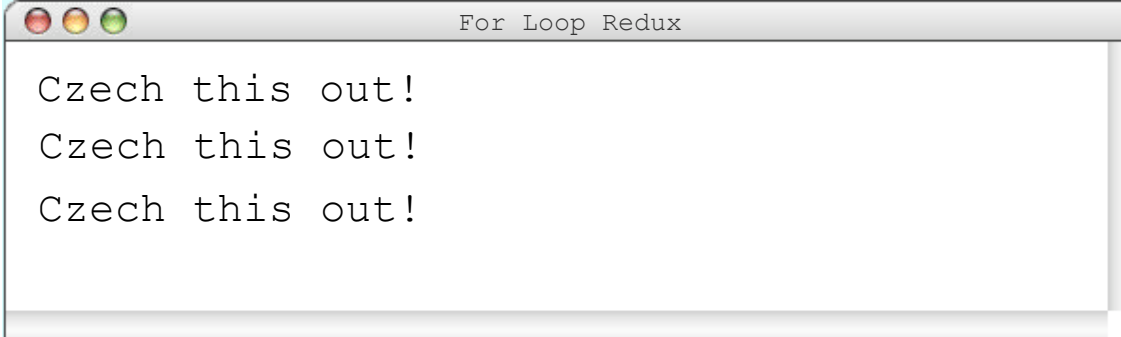


A terminal window titled "For Loop Redux" with three colored window control buttons (red, yellow, green) in the top-left corner. The window contains three lines of text, each on a new line: "Czech this out!".

# For loop

`i` 3

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```

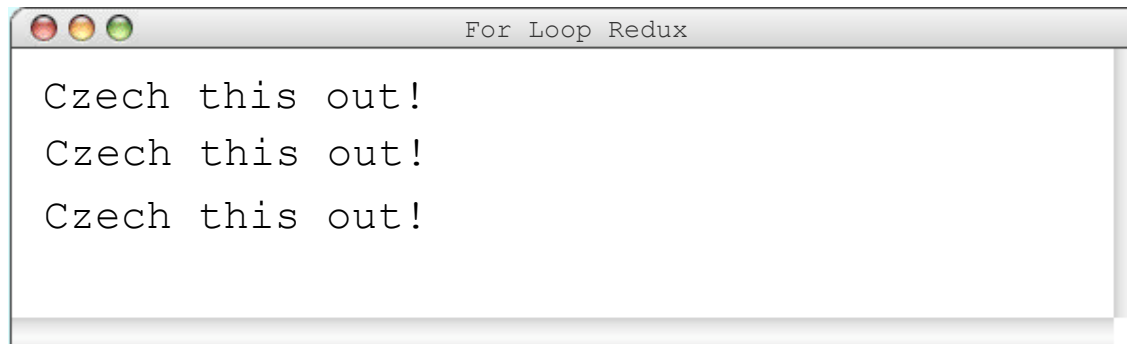


```
For Loop Redux  
Czech this out!  
Czech this out!  
Czech this out!
```

# For loop

i 3

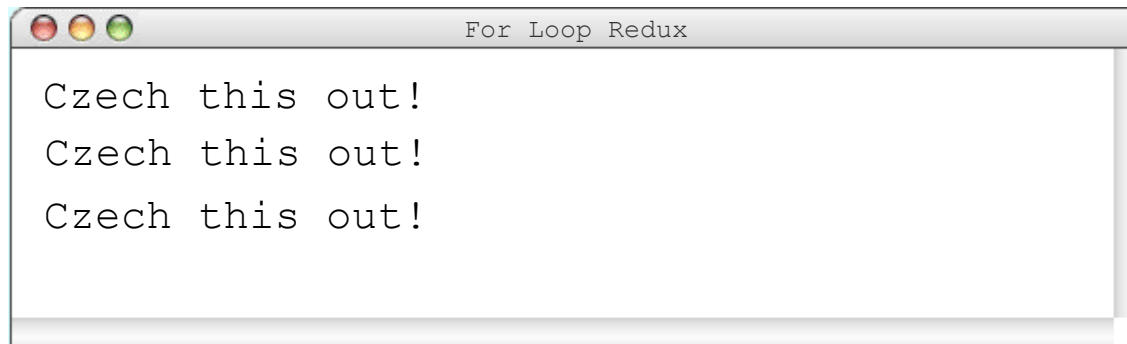
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



A terminal window titled "For Loop Redux" with three colored window control buttons (red, yellow, green) in the top-left corner. The window contains three lines of text, each on a new line: "Czech this out!".

# For loop

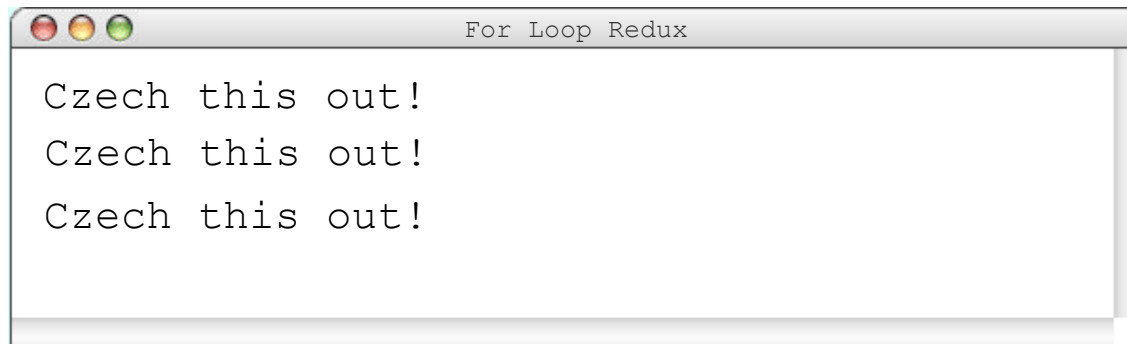
```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```





# For loop

```
for(int i = 0; i < 3; i++) {  
    println("Czech this out!");  
}
```



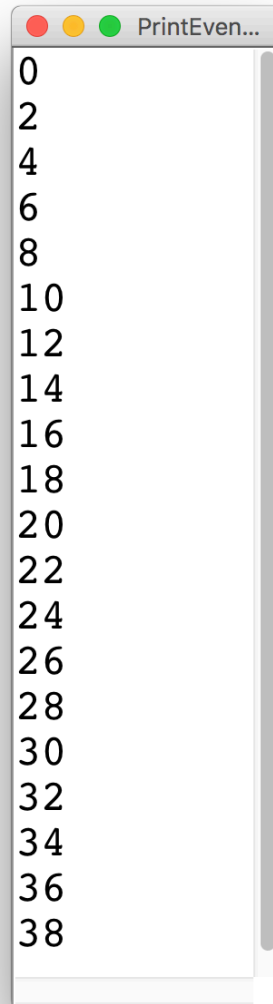
Think for a minute,  
then talk to the person next to you:

How would we print the first 100 even numbers?

# Use the loop variable!



# Printing even numbers

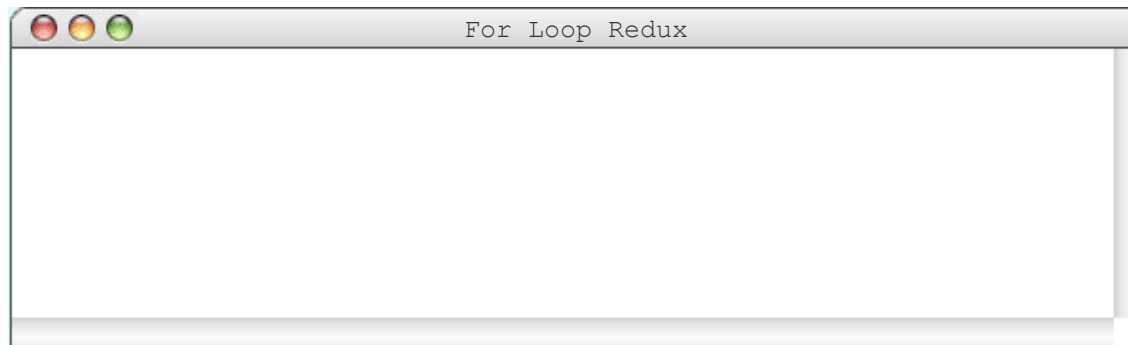


# Printing even numbers

```
for(int i = 0; i < NUMS; i++) {  
    println(i * 2);  
}
```

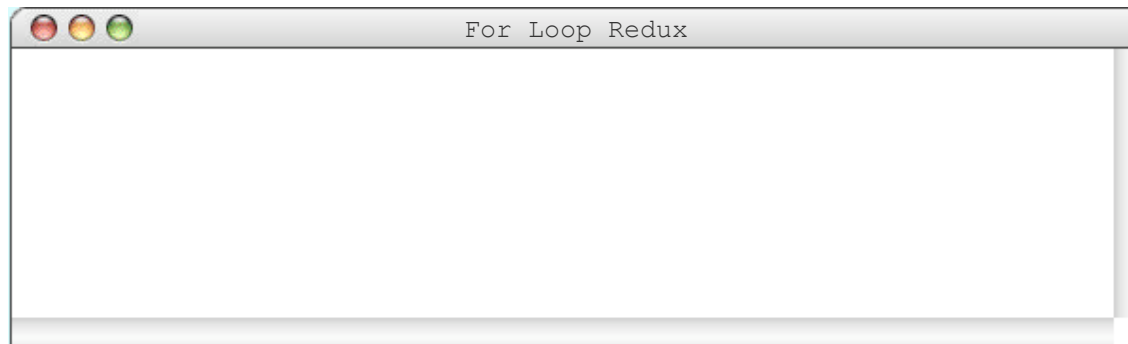
# Printing even numbers

```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

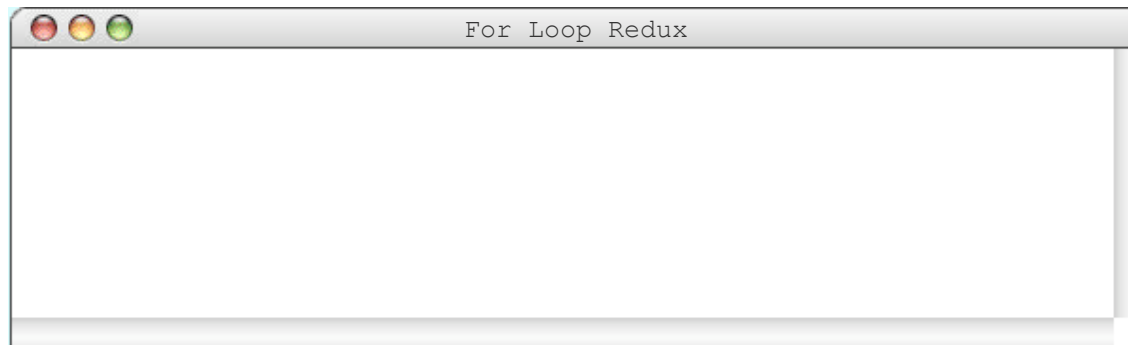
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 0

```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```

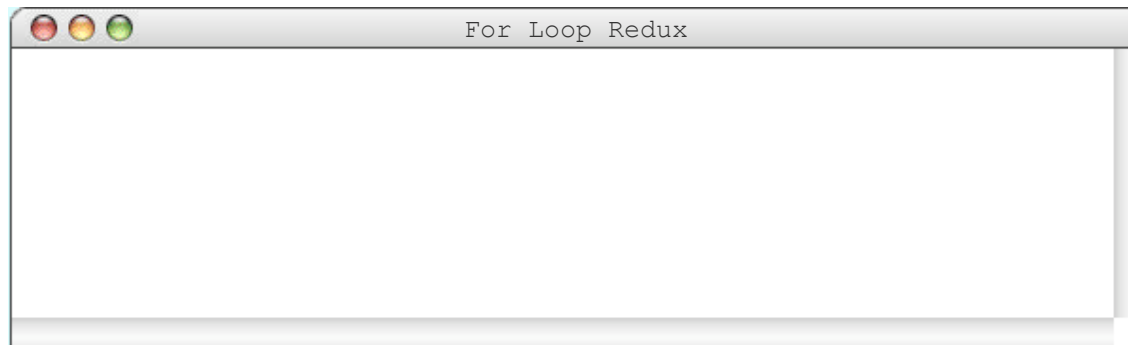




# Printing even numbers

i 0

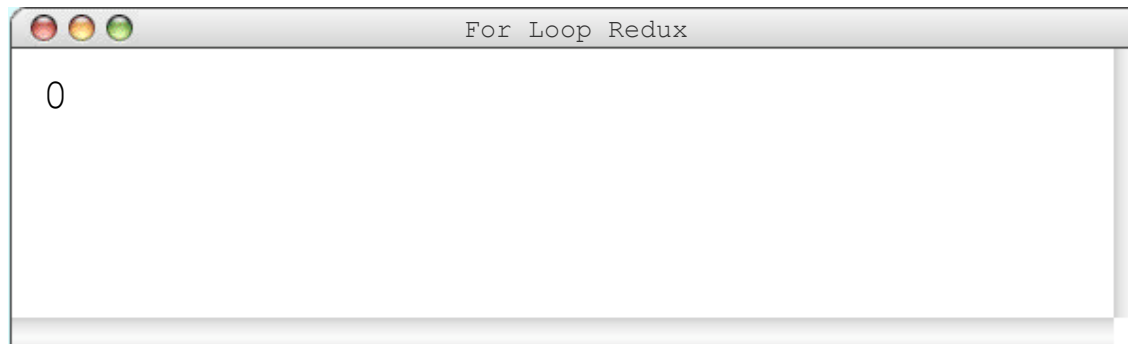
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 0

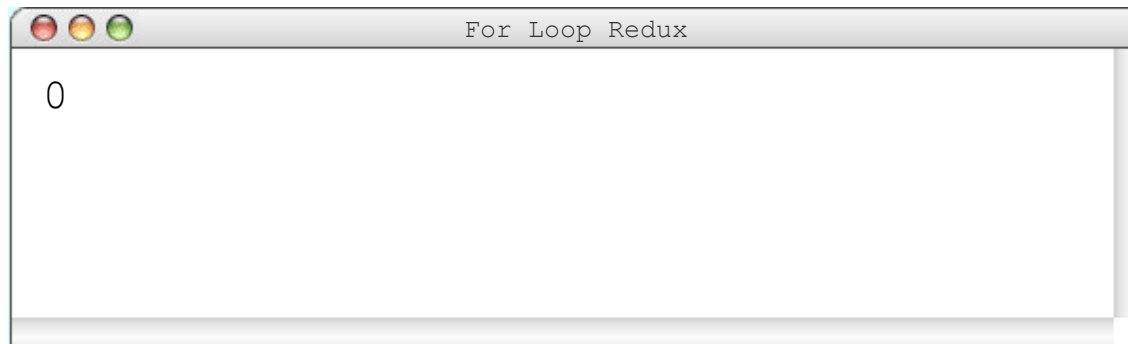
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 1

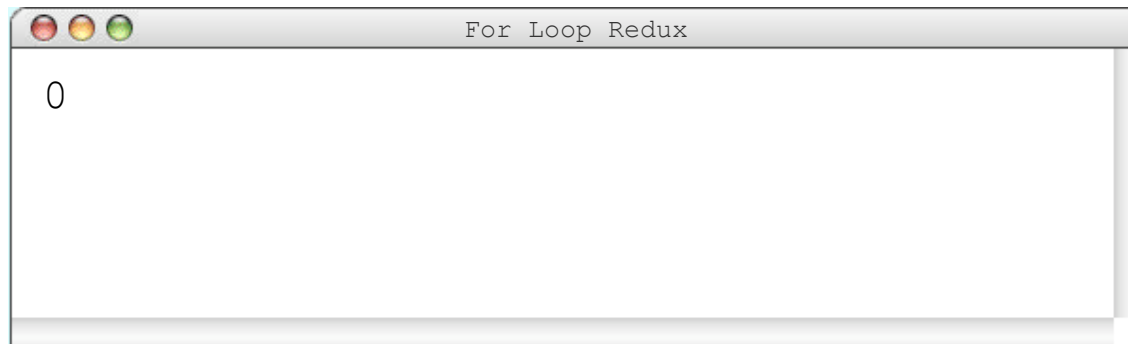
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 1

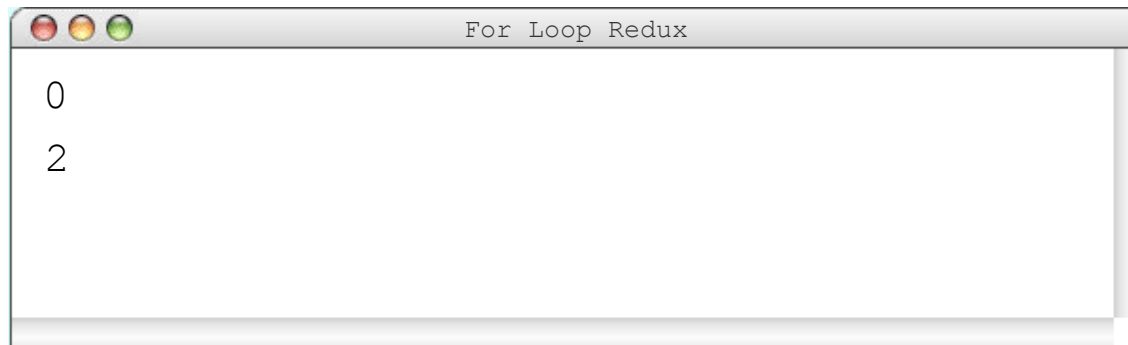
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 1

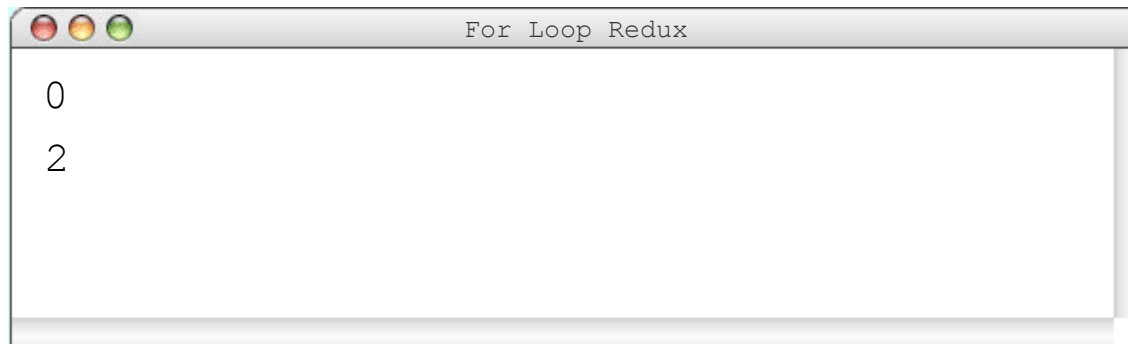
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 2

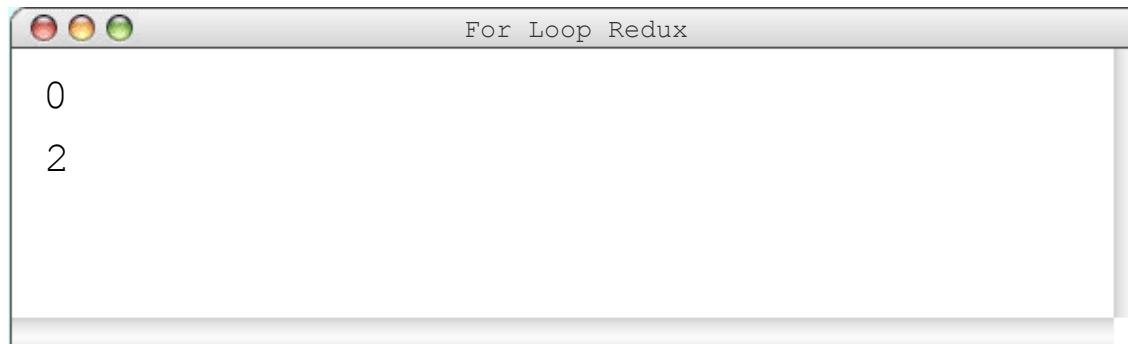
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 2

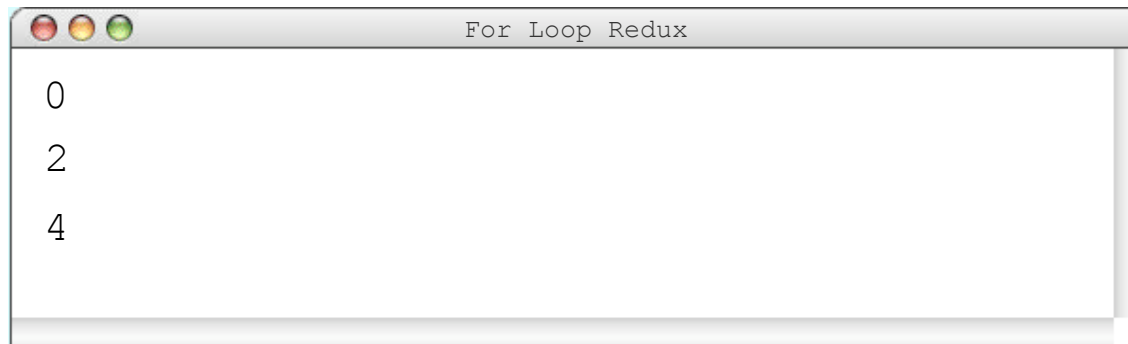
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 2

```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```

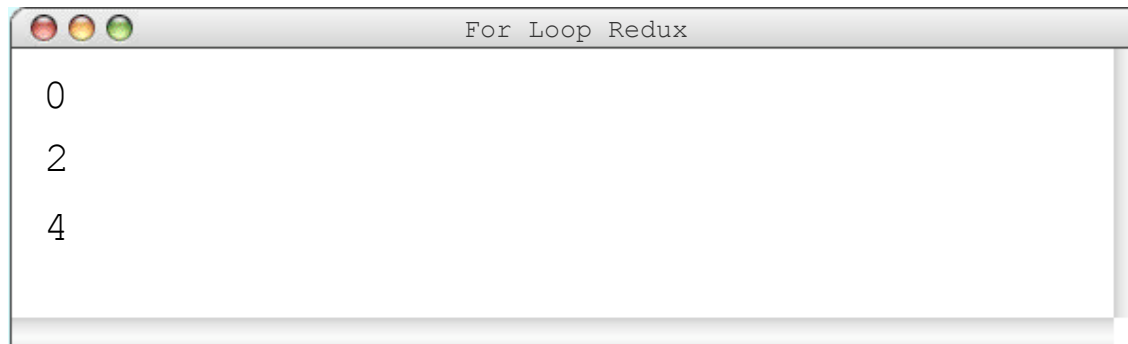




# Printing even numbers

i 3

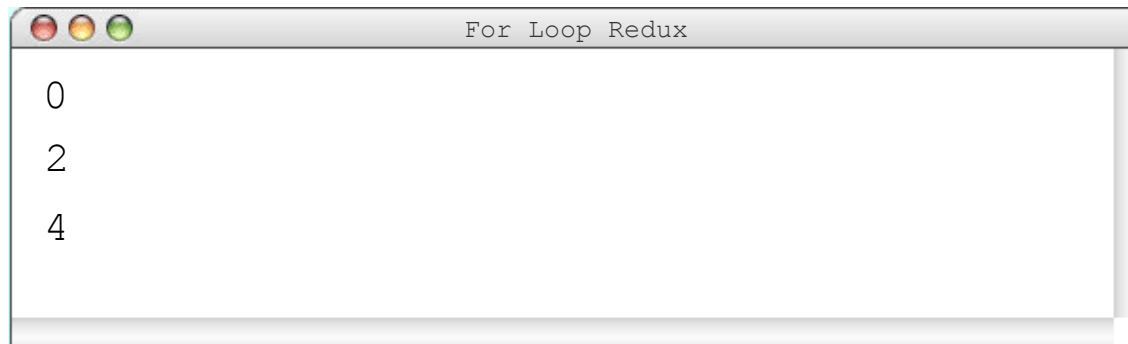
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

i 3

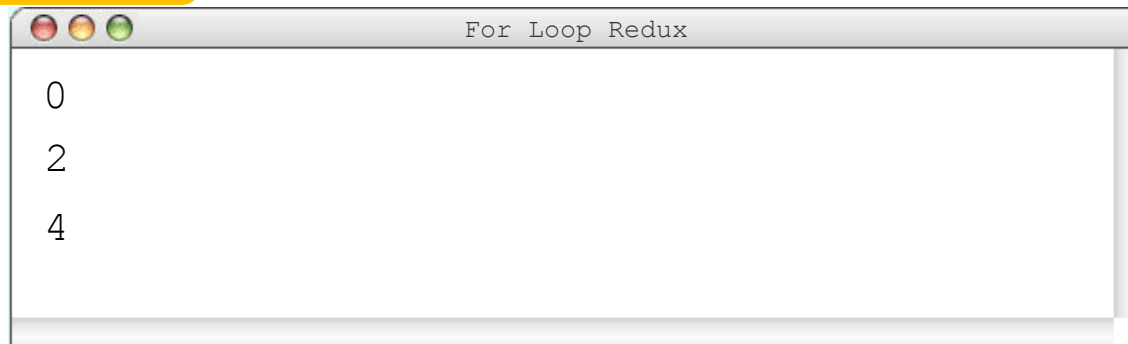
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

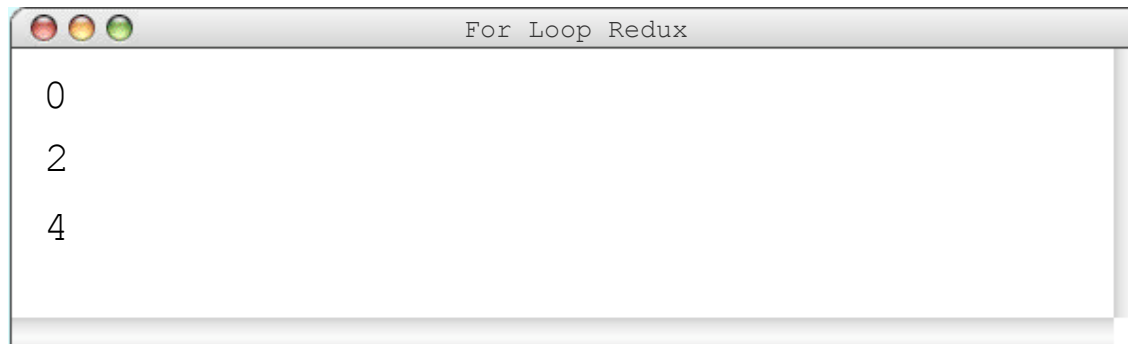
i 3

```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```



# Printing even numbers

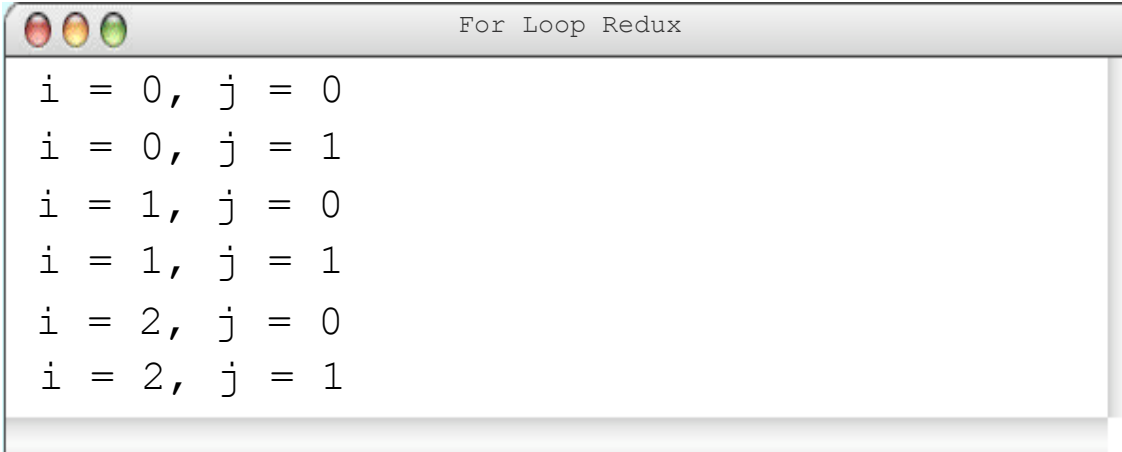
```
for(int i = 0; i < 3; i++) {  
    println(i * 2);  
}
```





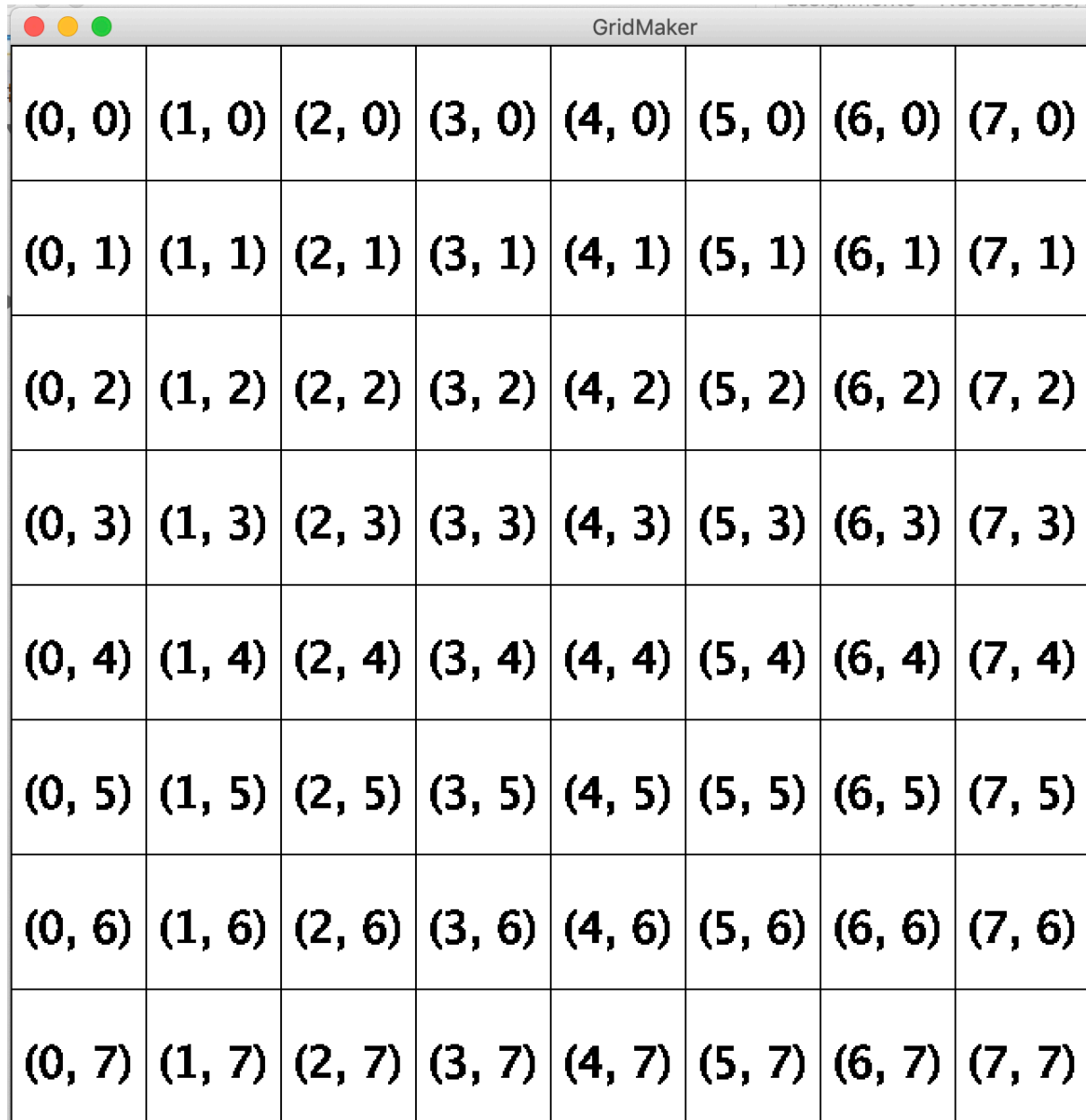
# Printing nested for loops

```
for(int i = 0; i < 3; i++) {  
    for(int j = 0; j < 2; j++) {  
        println("i = " + i + ", j = " + j);  
    }  
}
```



```
For Loop Redux  
i = 0, j = 0  
i = 0, j = 1  
i = 1, j = 0  
i = 1, j = 1  
i = 2, j = 0  
i = 2, j = 1
```

# Draw a grid



The image shows a window titled "GridMaker" with a standard macOS-style title bar (red, yellow, and green buttons). The window contains an 8x8 grid of cells. Each cell contains a coordinate pair in the format (x, y), where x and y are integers from 0 to 7. The grid is organized as follows:

(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)
(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)
(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)
(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)
(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)
(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)
(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)
(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)

So how does green screen work?



# Changing images

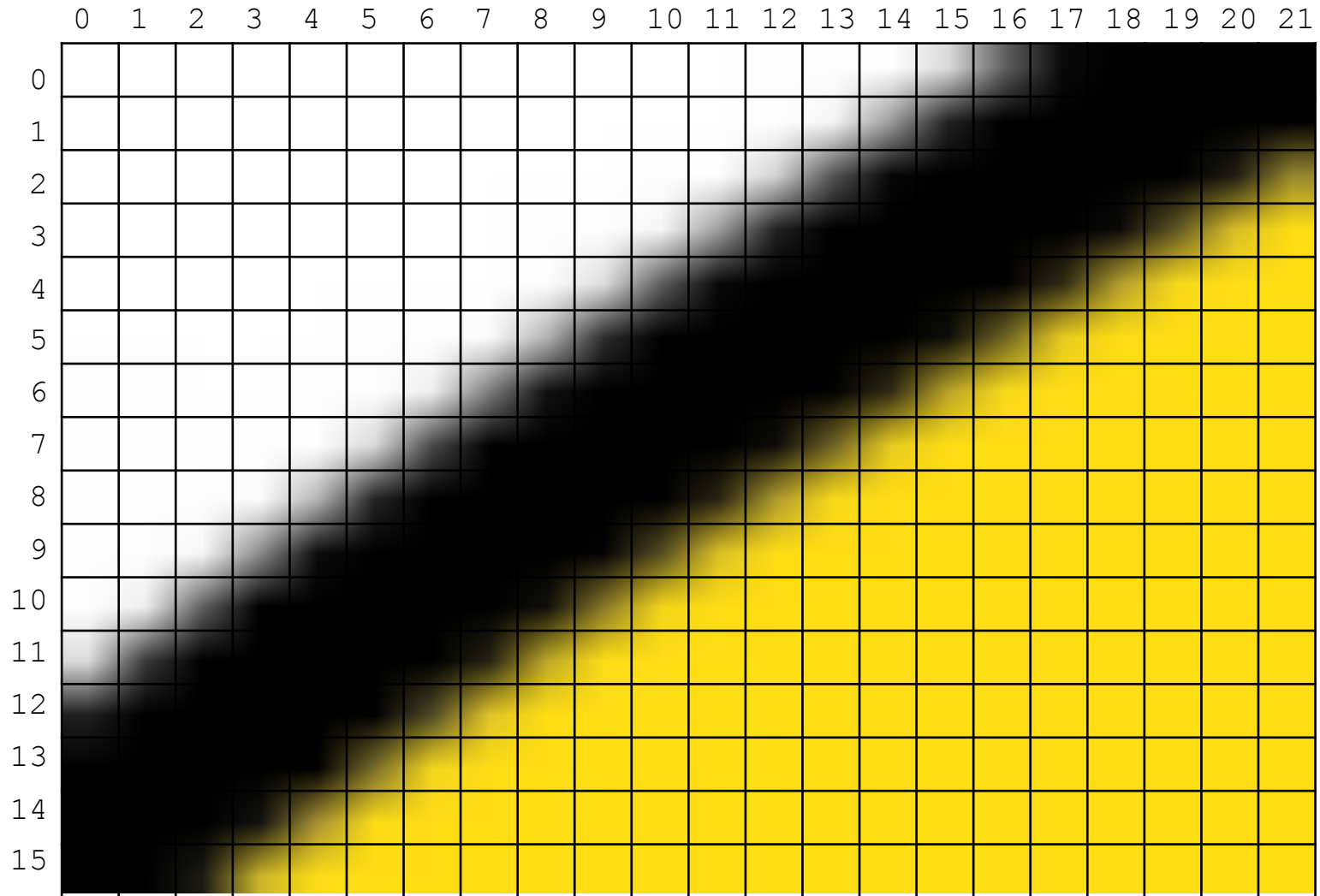


# Changing images

An image is made  
up of square  
pixels....

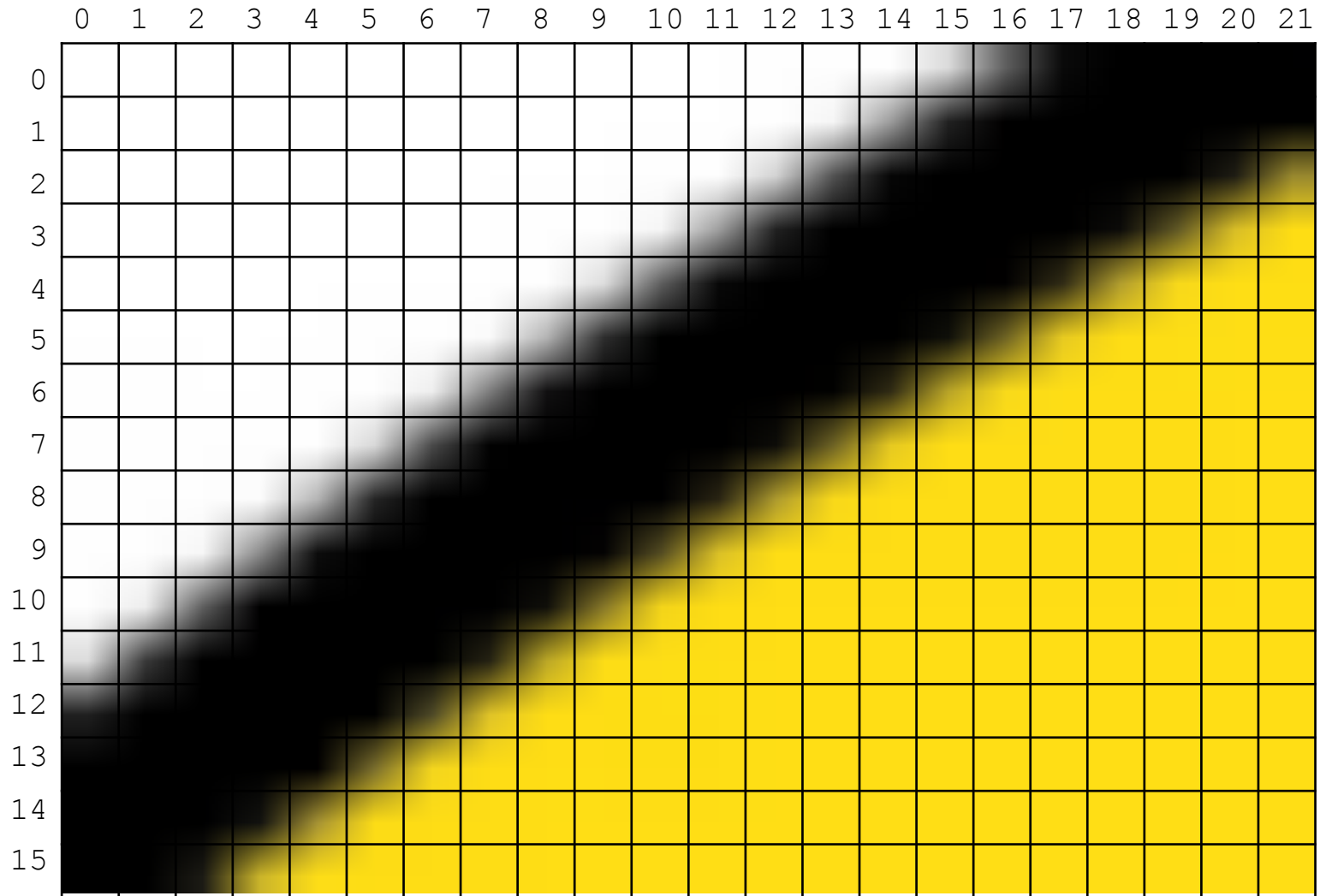


# Changing images



... which we can think of as a grid with rows and columns

# Changing images



We can look at each pixel like a box in a grid,  
and change the ones we want to change!

A photograph of the Golden Gate Bridge in San Francisco, California, taken during sunset. The bridge's iconic orange-red towers and suspension cables are silhouetted against a sky of soft pinks, oranges, and blues. The city lights are visible in the distance across the water. A red-bordered box is overlaid on the lower half of the image, containing the text.

**Přeji vám hezký víkend!**