Pattern Day!

Day 6
Monday, August 4, 2014

What is the core element of any pattern?

What is the core element of any pattern?

REPETITION.

Repetition

You know how to draw a circle.

But if you want **50** circles...

```
ellipse(10, 10, 10, 10);
ellipse(10, 20, 10, 10);
ellipse(10, 30, 10, 10);
ellipse(10, 40, 10, 10);
ellipse(10, 50, 10, 10);
ellipse(10, 60, 10, 10);
ellipse(10, 70, 10, 10);
ellipse(10, 80, 10, 10);
ellipse(10, 90, 10, 10);
ellipse(10, 100, 10, 10);
ellipse(10, 110, 10, 10);
ellipse(10, 120, 10, 10);
ellipse(10, 130, 10, 10);
ellipse(10, 140, 10, 10);
```

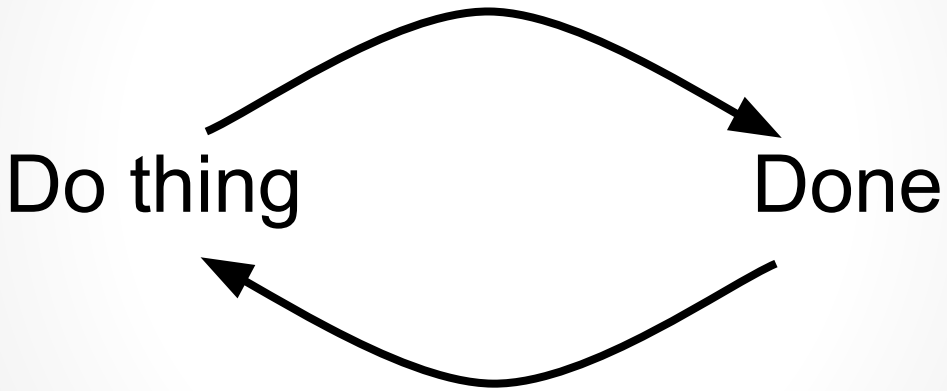
you get the point. It is super boring and annoying.

One awesome thing about computers

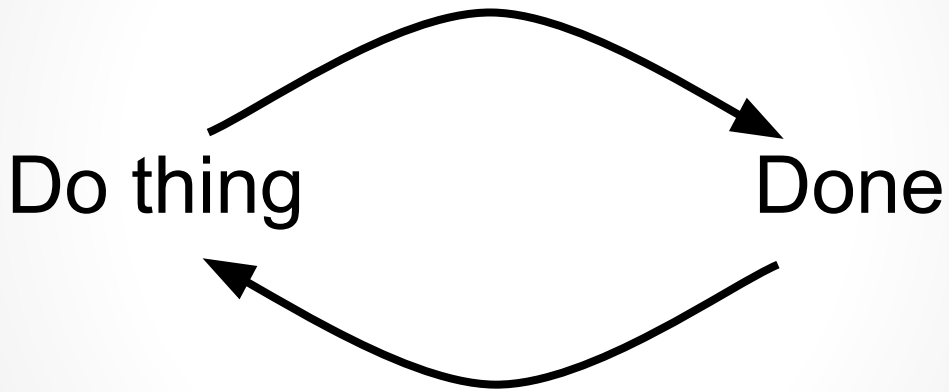
One awesome thing about computers

They never get bored

We want the computer to do this:



We want the computer to do this:



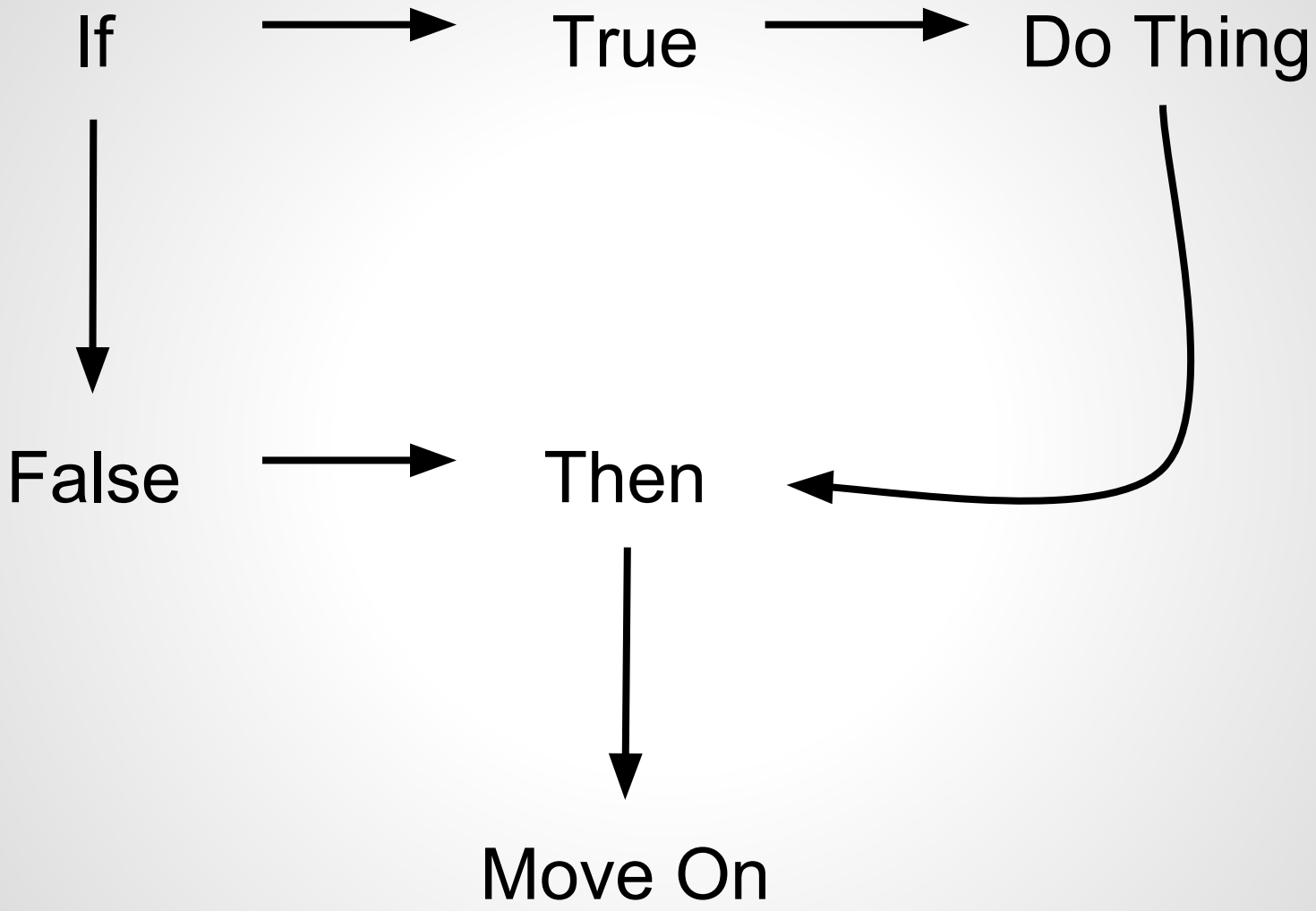
But when will it end? How will it start?
When does it run?

It sounds like we need a conditional statement.

Quick “if( )” conditional review

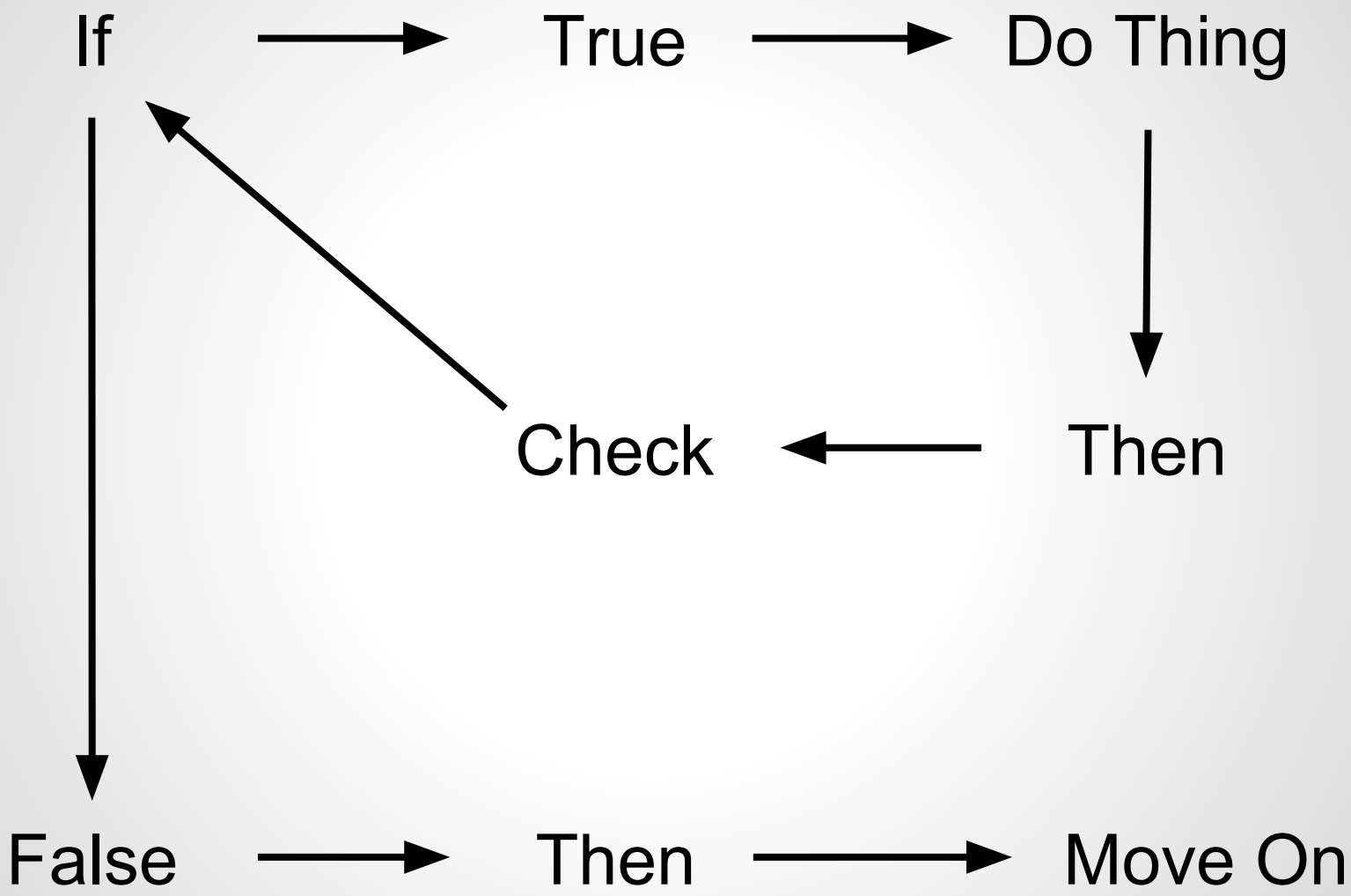
```
if (test) {
    stuff to do if true;
}
```

“if ( )”

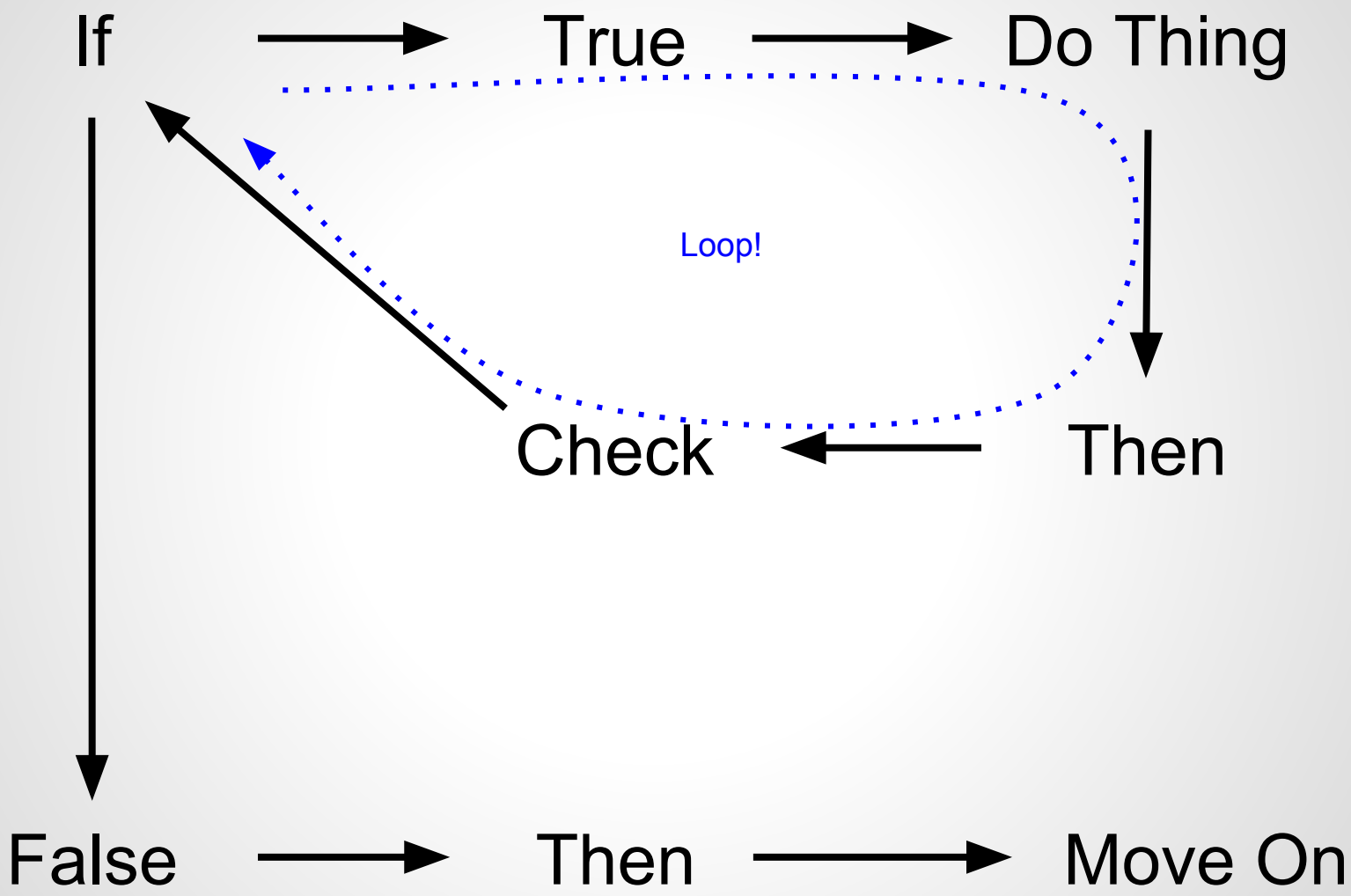


Let us mix the if () conditional with our earlier loop.

“while ( )”



“while ( )”



“while()” vs “if()”

```
if (test) {
    stuff to do if true;
}
```

```
while (test) {
    stuff to do if true;
}
```

“while()” vs “if()”

```
if (test) {
    stuff to do if true;
}
```

Happens only once, and then exits.

```
while (test) {
    stuff to do if true;
}
```

Repeats for the entire duration that the test is true.

“while()” vs “if()”

If test remains unchanged, program continues

```
if (test) {
    stuff to do if true;
}
```

Happens only once, and then exits.

If test remains unchanged, program loops until a change occurs.

```
while (test) {
    stuff to do if true;
}
```

Repeats for the entire duration that the test is true.


DANGER
WILL
ROBINSON!!

INTRODUCING:

INTRODUCING:

(theme music playing)

The for( ) Loop

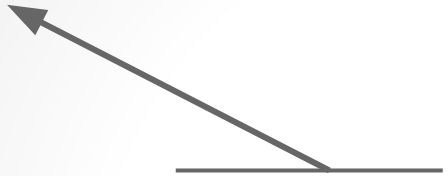
(theme music playing)

For Loops

```
for (int i = 0; i < 10; i++) {
    stuff to do while looping;
}
```

For Loops

initialize a variable to test.
This happens only once, when
the loop first begins.



```
for (int i = 0; i < 10; i++) {
    stuff to do while looping;
}
```

For Loops

initialize a variable to test.
This happens only once, when
the loop first begins.

Run the test.
This happens at the beginning
of every loop.

```
for (int i = 0; i < 10; i++) {
    stuff to do while looping;
}
```

For Loops

initialize a variable to test.
This happens only once, when the loop first begins.

Run the test.
This happens at the beginning of every loop.

Update the test variable.
This occurs at the end of the loop, preparing for the next test.

```
for (int i = 0; i < 10; i++) {
    stuff to do while looping;
}
```

For Loops

initialize a variable to test.
This happens only once, when the loop first begins.

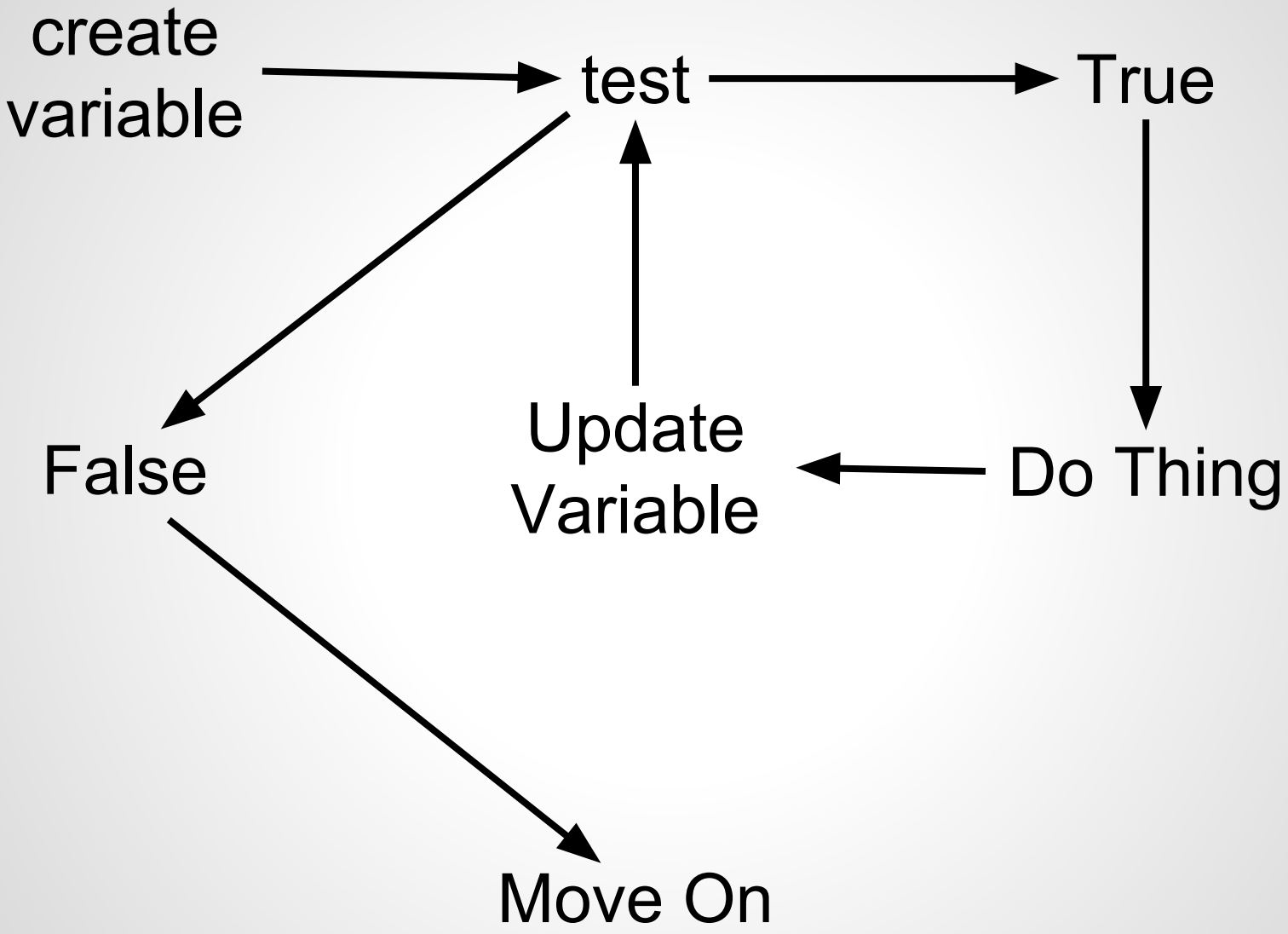
Run the test.
This happens at the beginning of every loop.

Update the test variable.
This occurs at the end of the loop, preparing for the next test.

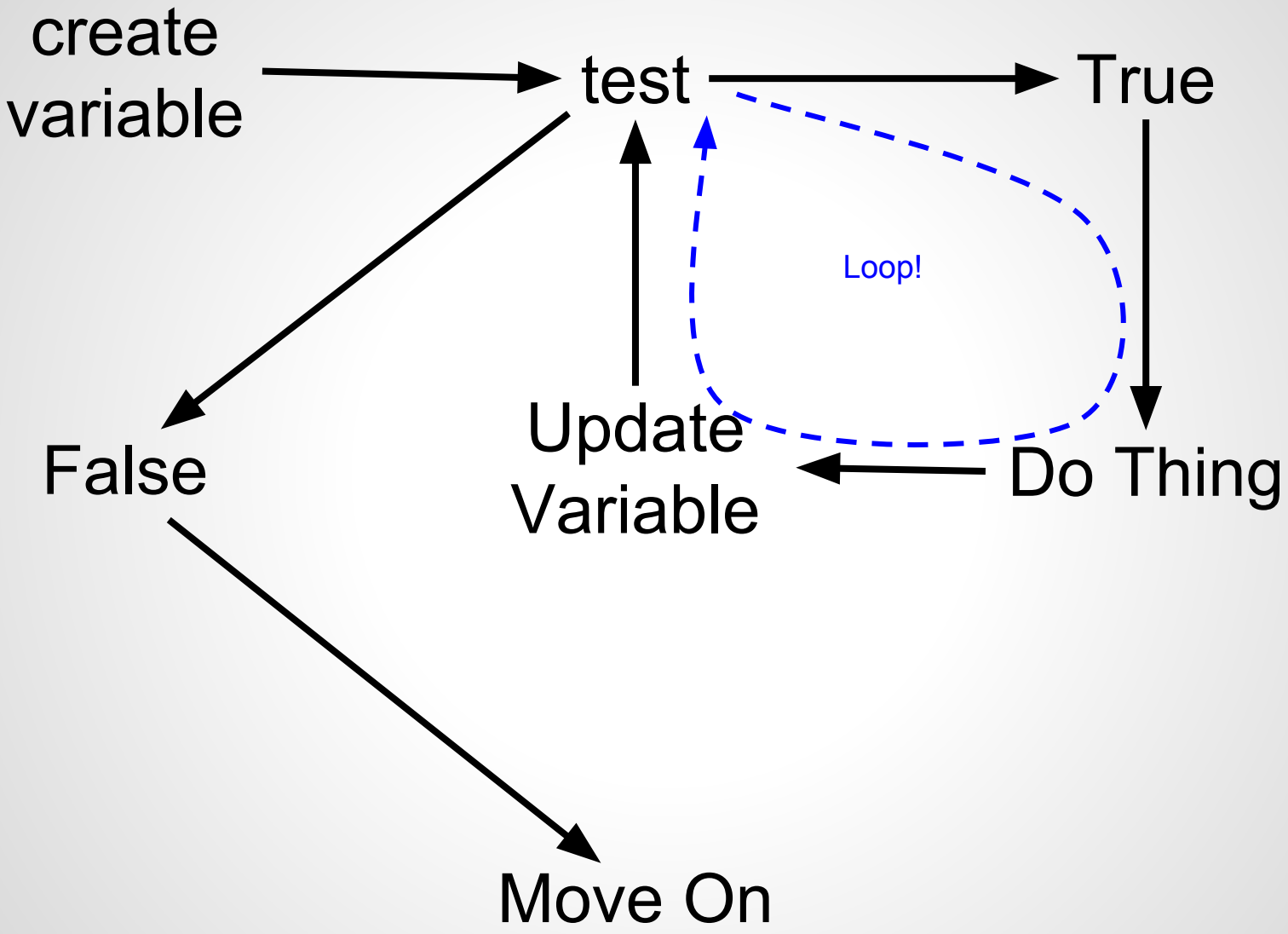
```
for (int i = 0; i < 10; i++) {
    stuff to do while looping;
}
```

local variable;
exists only within the for loop

“for ( )”


create variable
test
True
False
Update Variable
Do Thing
Move On

“for ( )”


create variable
test
True
Loop!
False
Update Variable
Do Thing
Move On

Common increment/decrement options

x++;	*means*	x = x + 1;
x--;	*means*	x = x - 1;
x+=2;	*means*	x = x + 2;
x*=3;	*means*	x = x * 3;

“while( )” vs “for( )”

```
initialize;
while (test) {
    stuff to loop;
    update;
}
```

```
for (initialize; test; update) {
    stuff to loop;
}
```

“while( )” vs “for( )”

```
initialize;
while (test) {
    stuff to loop;
    update;
}
```

Forget this and
crash the program.

```
for (initialize; test; update) {
    stuff to loop;
}
```

“while( )” vs “for( )”

Four lines of code.
Harder to read.

```
initialize;
while (test) {
    stuff to loop;
    update;
}
```

Forget this and
crash the program.

```
for (initialize; test; update) {
    stuff to loop;
}
```

“while( )” vs “for( )”

Four lines of code.
Harder to read.

```
initialize;
while (test) {
    stuff to loop;
    update;
}
```

Forget this and
crash the program.

```
for (initialize; test; update) {
    stuff to loop;
}
```

Two lines of code.
Simple and clean.

“while( )” vs “for( )”

Four lines of code.
Harder to read.

```
initialize;
while (test) {
    stuff to loop;
    update;
}
```

Forget this and
crash the program.

all condensed into one
easy to read line.

```
for (initialize; test; update) {
    stuff to loop;
}
```

Two lines of code.
Simple and clean.

Let’s Get An Example In Pseudocode

Let’s Shovel The Driveway!

THIS. IS. A. SHOVELING LOOP!

for (_____ ; _____ ; _____){

_____ ;

}

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; ____________________________ ;  ____________ ){

        ____________ ;

}
```

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength;  __________){

        __________;

}
```

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

        ____________;

}
```

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

    shovelSnow();

}
```

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

    shovelSnow();

}
```

WHAT IS WRONG WITH THIS PICTURE?

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

    shovelSnow();

}
```

Hint: It's this thing.

WHAT IS WRONG WITH THIS PICTURE?

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

    shovelSnow(parameter);

}
```

THIS. IS. A. SHOVELING LOOP!

```
for (int position = startOfDriveway; position < drivewayLength; stepForward){

    shovelSnow(position);

}
```

In Class Assignment

Using a for( ) loop, make a line of white circles, each of which turn black when the mouse hovers over it.

Homework

Experiment with for( ) loops to create three interesting and interactive patterns.