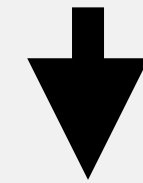


																			
																			
																			
																			
																			
	WEEK FIVE - ACTION																		
																			
																			
																			
																			
																			
																			

# THE BUILDING BLOCKS OF CODING

[You are a programmer!](#)

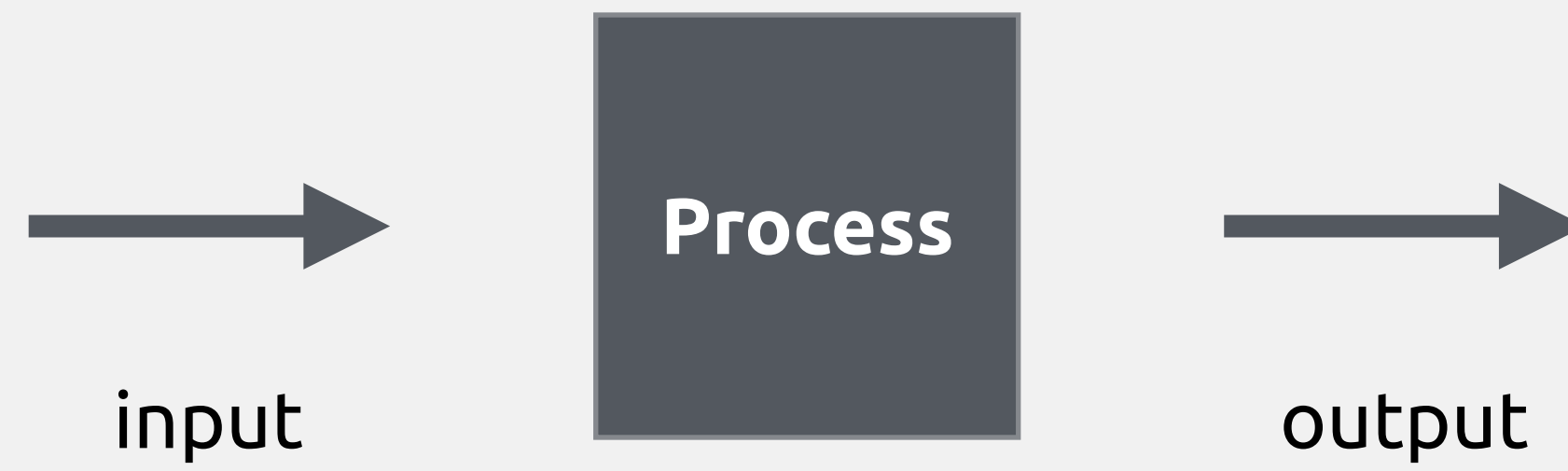


MEMORY    DECISION    REPETITION    **ACTION**

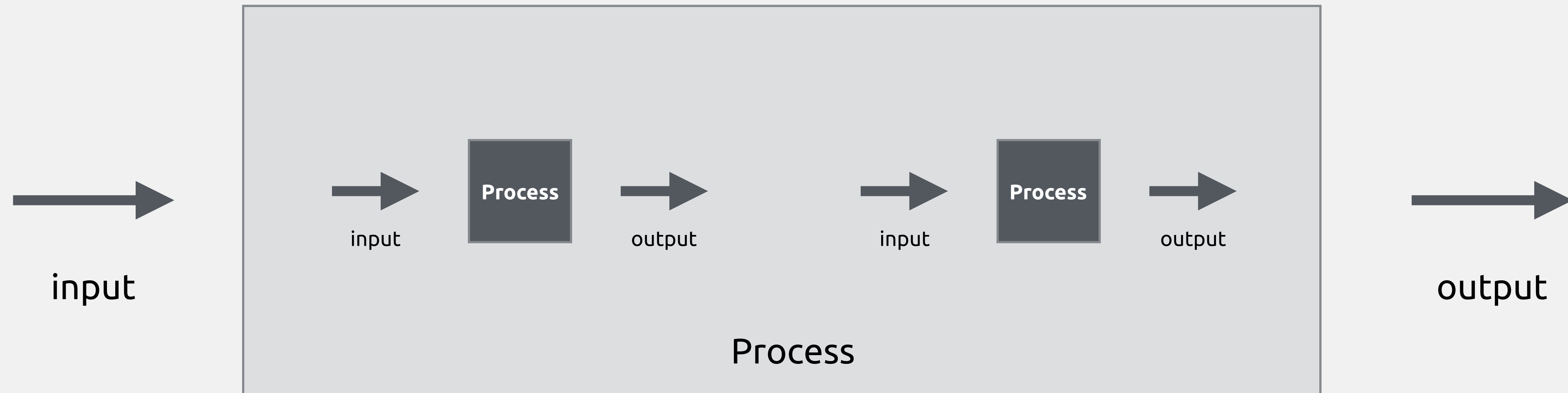
# ACTION

- Functions are the building blocks of computer programs
- A Function is a named block of code that can be invoked.
- Functions may have one or more parameters
- Functions may return a single value

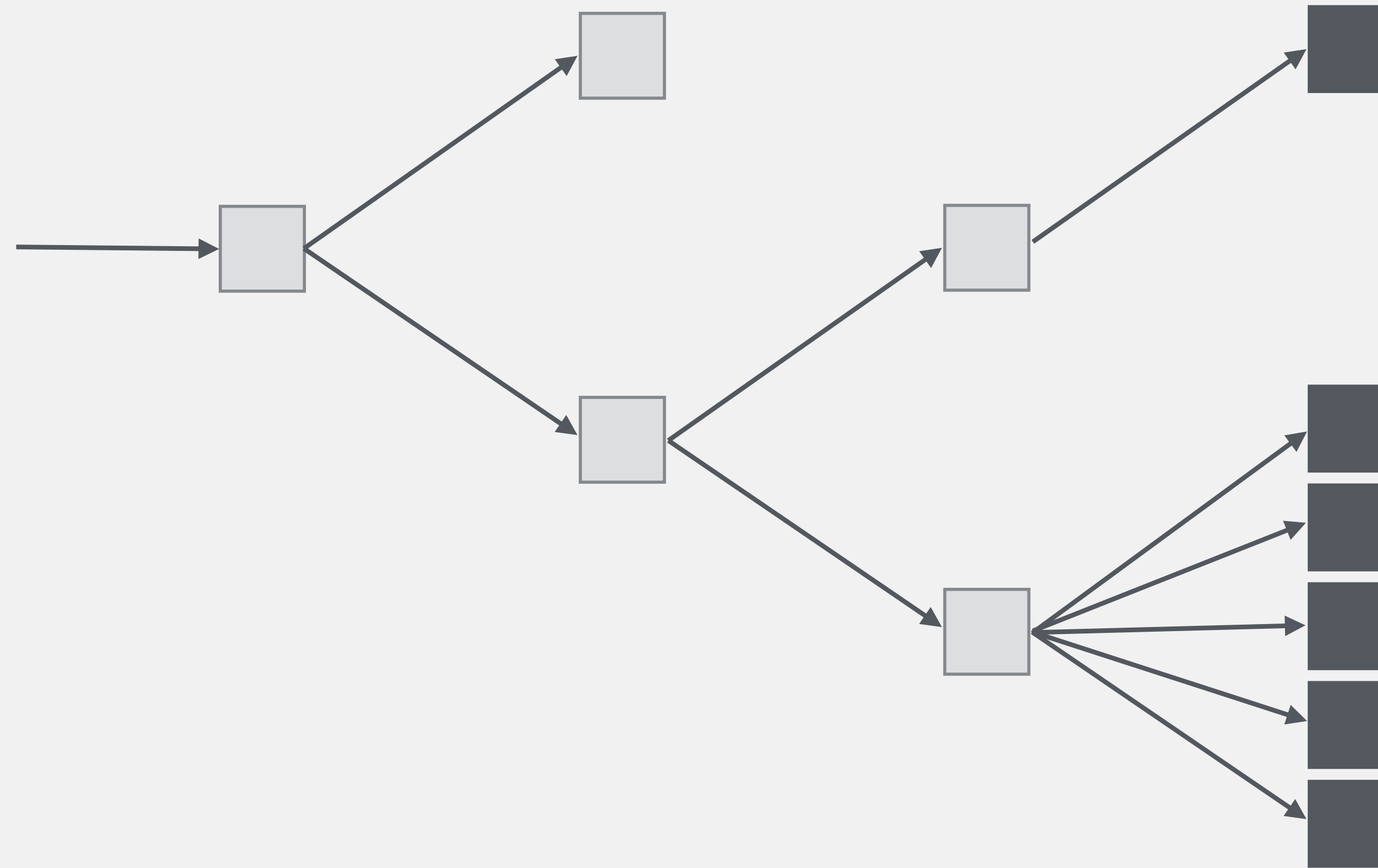
# ENCAPSULATION



# ENCAPSULATION

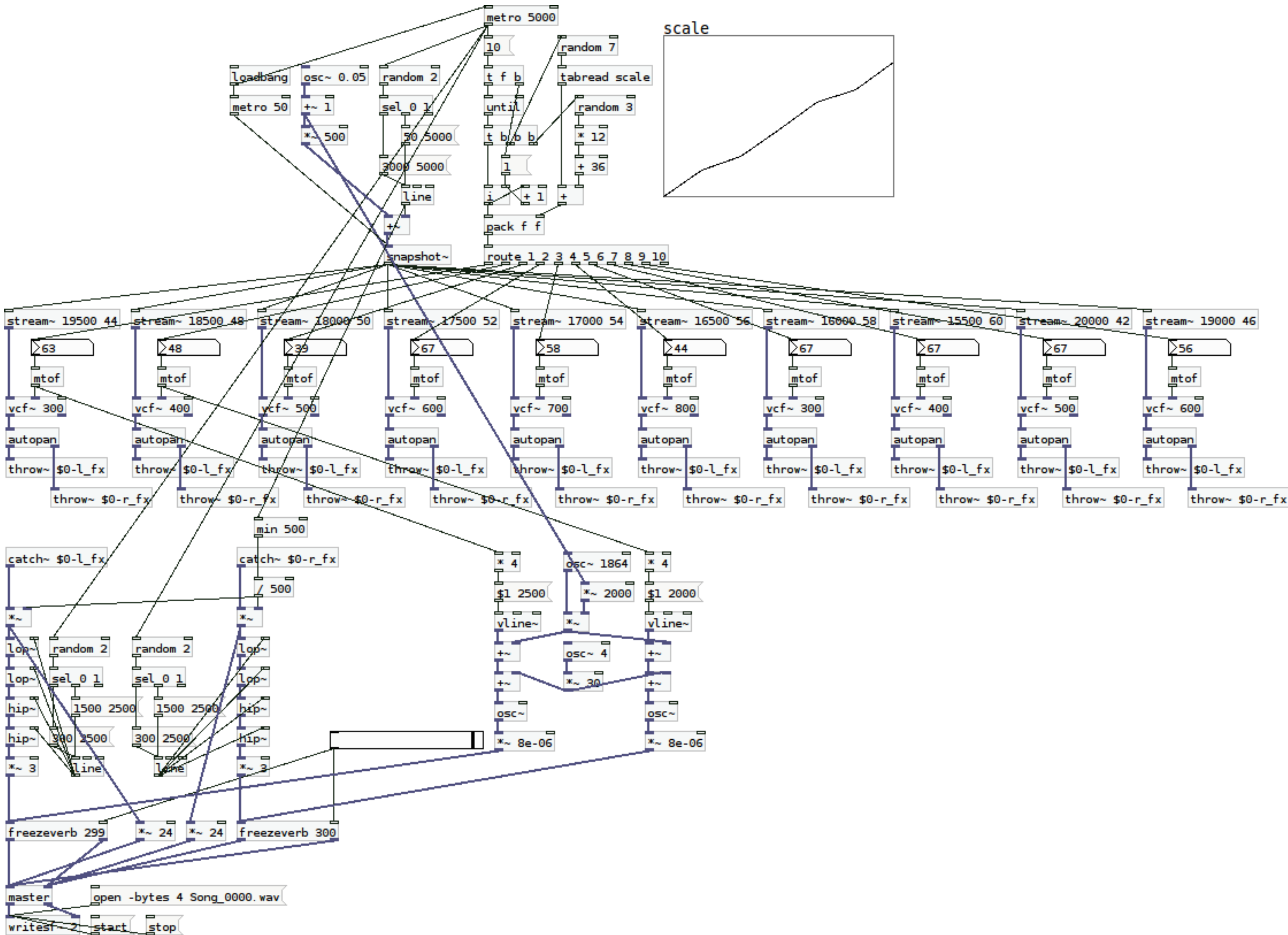


# BRANCHING LOGIC



The combination of conditions, loops and functions allows us to  
express ANY algorithm

# DATA FLOW PROGRAMMING



# PARAMETERS

```
line(0, 0, 100, 100);
```

**Parameters** are bits of information which the function should use

(input)



## RETURN VALUE

```
var num = random( -10, 10 );
```



The **Return value** is what replaces the function call inline  
(output)

# FUNCTIONAL PROGRAMMING

```
1  V = frc(V) +
2      sgn(
3      stp(
4      vec4(0.2) ,
5      md(
6      T4.rrgr,
7      vec4(.88,.88,0,0)
8      )
9      )
10 ) *
11 (
12   vec4(0.78, .0, 1.,0) +
13   (
14     (T4.rrgg - T4.bgrb) *
15     vec4(1., 1., 0.,0)
16   ) -
17   frc(V.rrbr)
18 ) * vec4(1., .9, 1., 0);
```

[https://en.wikipedia.org/wiki/Functional\\_programming](https://en.wikipedia.org/wiki/Functional_programming)

## A FUNCTION DEFINITION

```
sayHello();  
  
// ...  
  
function sayHello(){  
    print("Hello!");  
}
```

no Parameters, no Return value

## A FUNCTION DEFINITION

```
var rto = getMouseRatio();  
  
// ...  
  
function getMouseRatio(){  
    return (mouseX / mouseY);  
}
```

Return value, no Parameters,

## A FUNCTION DEFINITION

```
drawAnXAt(mouseX, mouseY);  
  
// ...  
  
function drawAnXAt(x, y){  
    line(x-10, y-10, x+10, y+10);  
    line(x-10, y+10, x+10, y-10);  
}
```

Parameters, no Return value

## A FUNCTION DEFINITION

```
var a = testMouseXPos(200);  
  
// ...  
  
function testMouseXPos(xPos) {  
    if(mouseX >= xPos){  
        return true;  
    }  
    return false;  
}
```

Parameters and Return value

## MODULO

$(40 / 10) \gg 4$  **(0)**  $\rightarrow (40 \% 10) \gg 0$

$(42 / 10) \gg 4$  **(2)**  $\rightarrow (42 \% 10) \gg 2$



MODULO returns the **remainder** when dividing two numbers

## MODULO

```
if(currentFrame % 10 == 0) {  
    // do something once every 10 frames  
}
```

MODULO is useful for testing divisions and increments



## MODULO

```
for(var i = 0; i < 100; i++){  
    strokeWeight(1);  
    if(i % 5 == 0) strokeWeight(5);  
    line(i*10, 0, i*10, height);  
}
```

every fifth line will be thicker

# HOMEWORK #5

## Defiant Drawing app

Please create a p5 sketch that uses mouse / button / keyboard input for creating an original drawing. Your app needs to have a will of its own. It should manipulate user input in surprising ways. The challenge is to find a sweet spot between choice and chance.

For next week, save and print out your favorite sketch result in A3 format  
(and hang in the room next door)