

LOOPS I.

// TODO.

A Reminder:

Test 1 covers lectures 1 - 6 and lab 1

Moving forward:

Slowly start introducing new topics

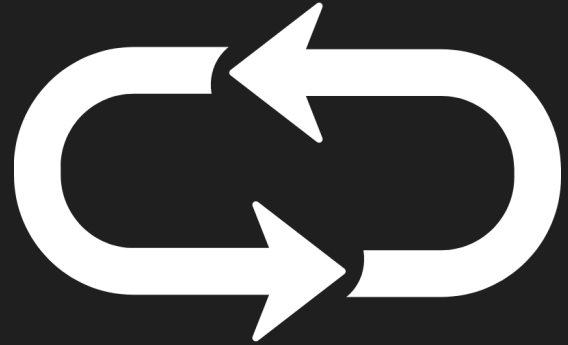
LOOPS.

What are loops?

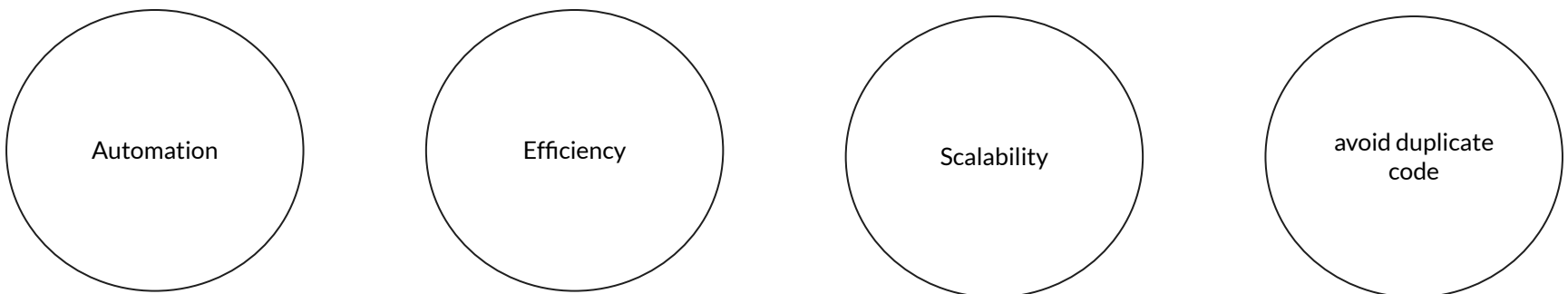
A repetition structure causes a statement or set of statements to execute repeatedly.

They provide a way to automate repetitive tasks and process data efficiently.

These structures are known as "Loops".



Why?



Automation

Efficiency

Scalability

avoid duplicate
code

Where are loops used?



Algorithm
Design

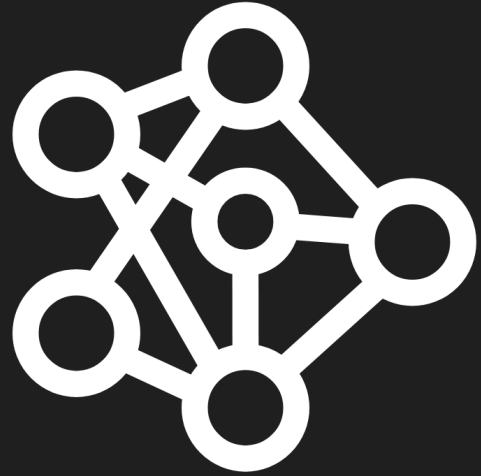
Data Processing

User Interfaces

Artificial
Intelligence

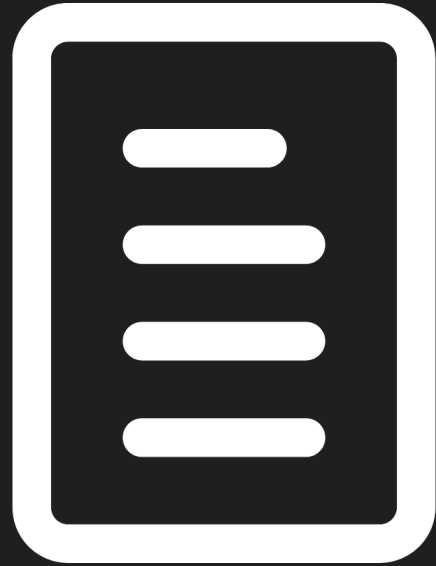
Algorithm Design.

Loops are fundamental to many algorithms for tasks such as searching and sorting on complex datasets.



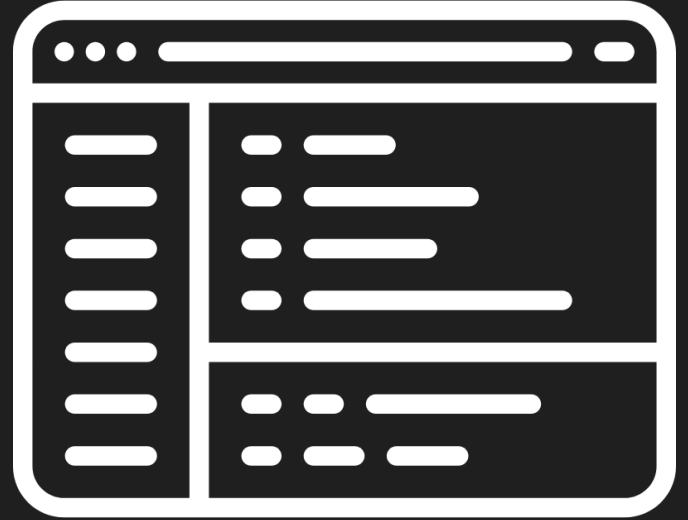
Data Processing.

Loops are used for processing and manipulating data in various formats, such as text processing, image processing, audio processing, and video processing.



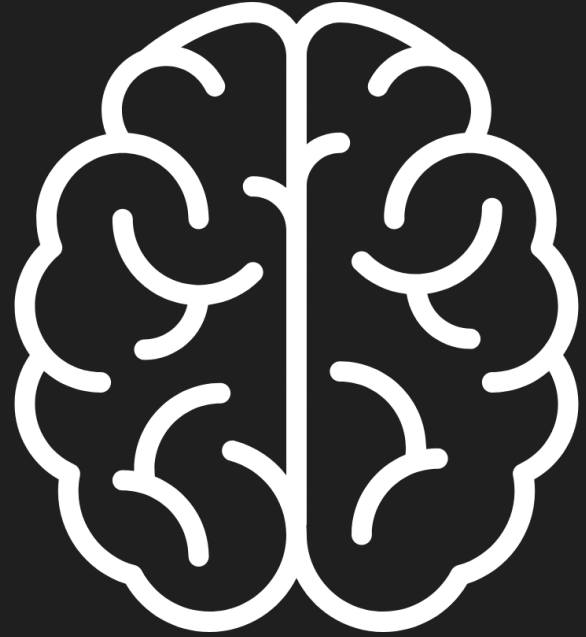
User Interfaces.

Loops are used in graphical user interfaces to continuously monitor user input, update the display, and handle events.



Artificial Intelligence.

Loops are used for training machine learning models.



... and so much more !!

WHILE LOOP.

If statements.

When I introduced an "if statement", the block of code is executed once, only if the condition evaluates to true.

```
if (condition) {  
    // some code  
}
```

While loop.

However, what happens if we want the block of code to be repeatedly executed? This is the idea behind the "While" loop.

The above block of code is repeatedly executed as long as the condition evaluates to true

```
while(condition) {  
    // some code  
}
```

What does this look like in
Flowgorithm?

Infinite loops.

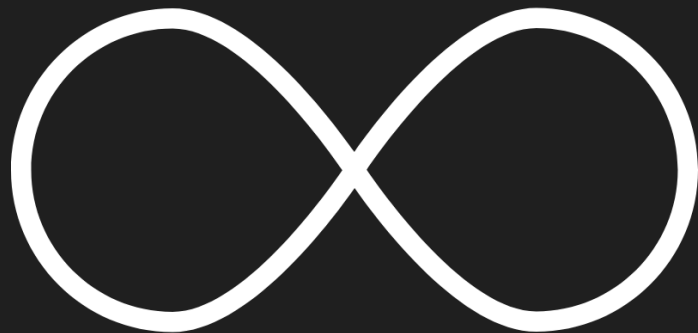
Be careful when writing a loop! It is important to make sure that the loop can eventually terminate and not execute forever.

Very often in loops, we will do 3 things:

Step 1) Declare and initialize a variable before the loop.

Step 2) Check a condition before each iteration.

Step 3) Perform some update step at the end of each iteration.



Building a (bad) program.

First, let's think about how this would look if we only used "if statements".

How would this look in Flowgorithm?

What about C#?

"output a welcoming message 5 times."

Building a (better) program.

Let's try this now with ITERATION :^)

```
"output a welcoming message 5 times."
```

Building a (slightly) better program.

Using ITERATION !!

```
"Guess the number between 1 and 1000."  
"Fail, and the game will never end."
```

EXERCISE 1.

Exercise 1.

Write the following idea in flowgorithm.

"If I am playing rock, paper, scissors with a computer"

"I should only be allowed to play a valid move"

INPUT VALIDATION.

Input Validation.

Input validation is the process of ensuring that user input is valid.

While loops are a popular choice used to perform input validation.

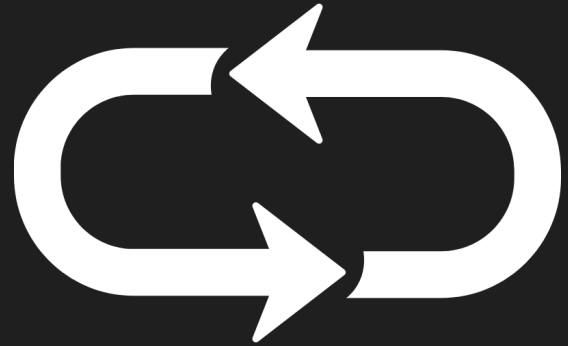
```
int.TryParse(Console.ReadLine(), out number);
```

FOR LOOPS.

For loops.

A for loop is a while loop with a built-in counter.

The while loop and the for loops are equivalent.



These mean the same things.

```
int i = 0;  
while (i < 5) {  
    // some code  
    i++;  
}
```

```
for (int i = 0; i < 5 ; i++) {  
    // some code  
}
```

As a general rule:

Use **for** loops when the condition depends on the value of an integer, and the number of iterations is fixed or easily computable.

- when you need a counter, might as well use built in functionality

Use **while** loops when the number of iterations is indefinite.

- input validation
- secret number guessing

```
int i = 0;  
while (i < 5) {  
    // some code  
    i++;  
}
```

```
for (int i = 0; i < 5 ; i++) {  
    // some code  
}
```

```
for (int i = 0; i < 5 ; i++) {  
    // some code  
}
```

Building the best program.

What would this look like in Flowgorithm using **for** loops?

What about C#?

```
"output a welcoming message 5 times."
```

Building the best program.

What would this look like in Flowgorithm?

What about C#?

```
"A computer should output all even numbers between 1 and 20."
```

QUESTIONS?