



Welcome to Algorismo!

Unlock the World of Algorithms and Coding

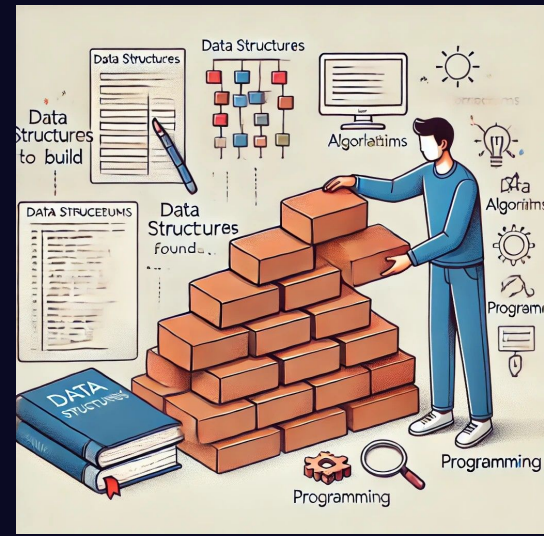
Purpose of the Series



Purpose of the Series



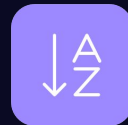
A stylized illustration in a flat, modern style. A central figure, a person in a dark blue suit, stands with their back to the viewer, reaching up to touch a large orange circle. Inside this circle is a white document icon with three dots. To the left of the person is a tall stack of books in blue and orange, with a small plant at the base. To the right is another stack of books. Various icons are scattered around: a clock, a checkmark in a blue circle, a document with a red circle, and a document with a red checkmark. The background is white with small blue and orange dots.



What We Will Cover

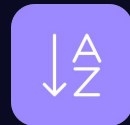
- Start with basic tasks:

Searching for a number, **sorting** names



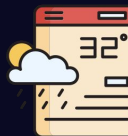
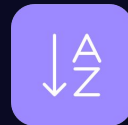
What We Will Cover

- Start with basic tasks:
Searching for a number, **sorting** names
- Move to more **complex organization**:
To-do lists, file organization



What We Will Cover

- Start with basic tasks:
Searching for a number, **sorting** names
- Move to more **complex organization**:
To-do lists, file organization
- Tackle **advanced problems**: City
maps, weather forecast



What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



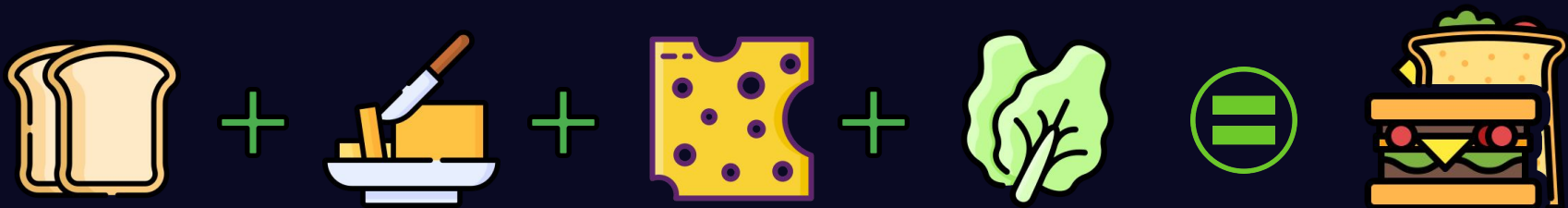
What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



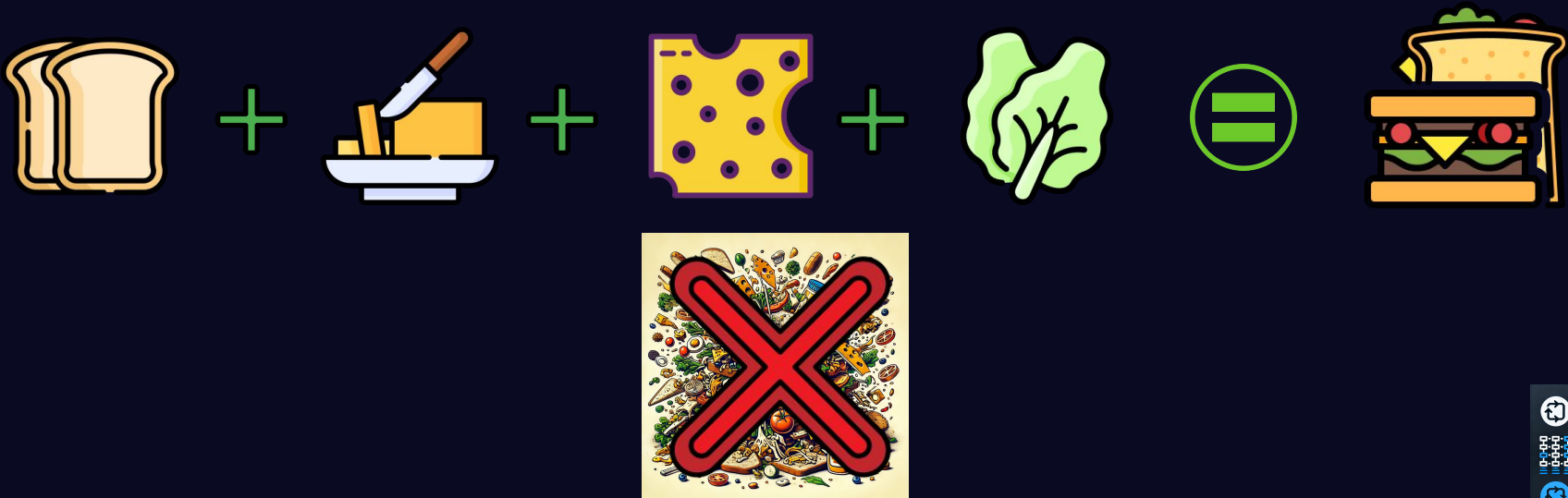
What is an Algorithm?

- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.

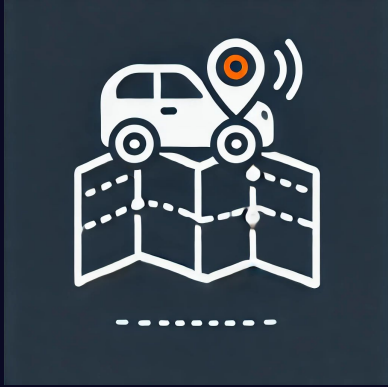


What is an Algorithm?

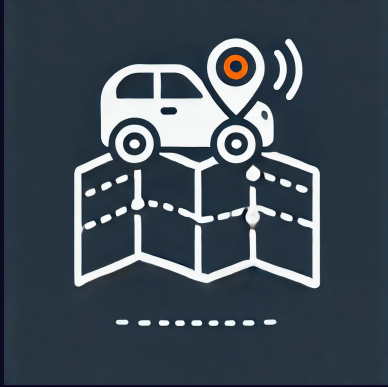
- An algorithm is a **step-by-step** procedure for solving a problem.
 - Think of it as a **recipe** for your computer.



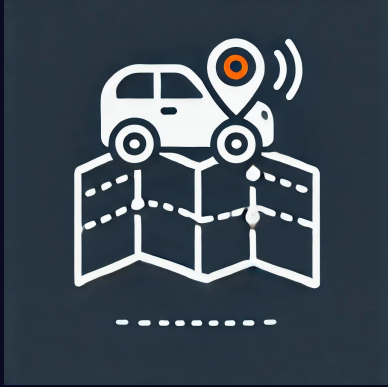
Why Algorithms Matter?



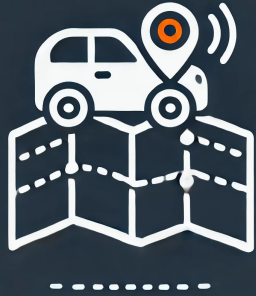
Why Algorithms Matter?



Why Algorithms Matter?



Why Algorithms Matter?



Key Traits of a Good Algorithm

- **Correct** – It should give you the right answer.
- **Efficient** – It should use resources wisely.



Key Traits of a Good Algorithm

- **Correct** – It should give you the right answer.



- **Efficient** – It should use resources wisely.



- **Finite** – It should end.



- **Clear** – It's easy to understand.



- **General** – It solves a wide range of problems.



Basic Components of Algorithms

Input



Basic Components of Algorithms

Input

Output



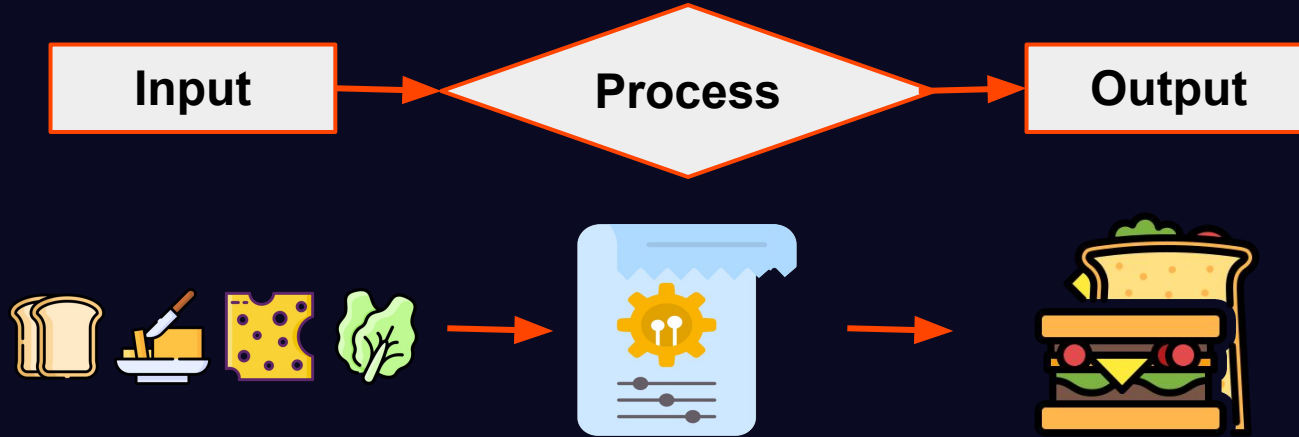
Basic Components of Algorithms



Basic Components of Algorithms



Basic Components of Algorithms



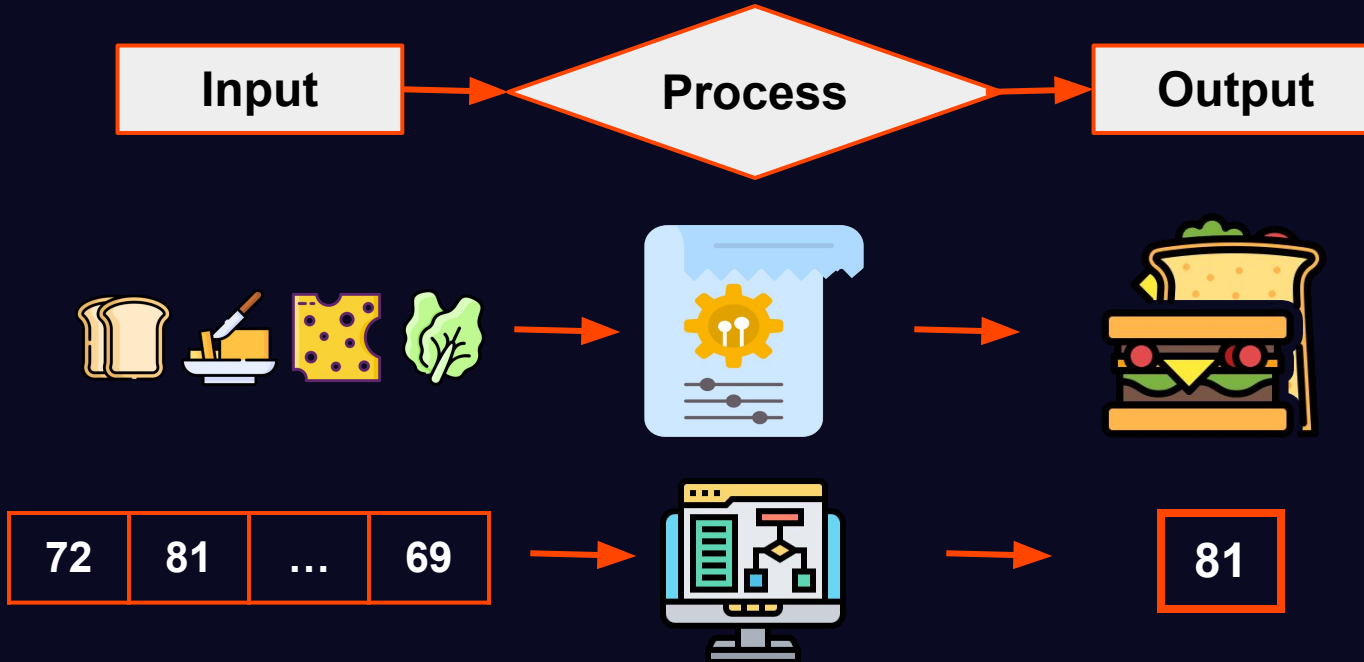
Basic Components of Algorithms



Basic Components of Algorithms



Basic Components of Algorithms



Algorithm in Action: Finding the Max

Input

5	3	9	1	6
---	---	---	---	---



Algorithm in Action: Finding the Max

Input

5	3	9	1	6
---	---	---	---	---

Output

9



Algorithm in Action: Finding the Max

Input



max = 5

Output



Algorithm in Action: Finding the Max

Input



max = 5

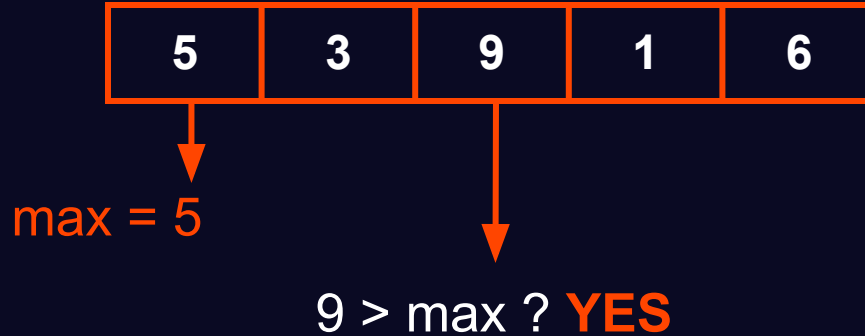
3 > max ? NO

Output



Algorithm in Action: Finding the Max

Input

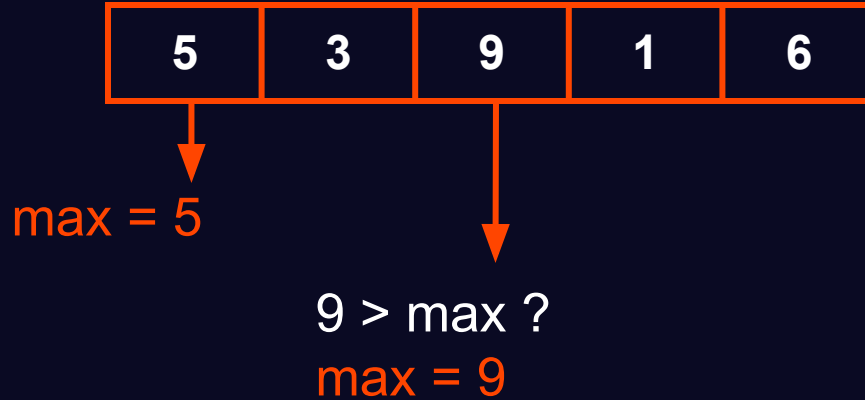


Output



Algorithm in Action: Finding the Max

Input



Output

9



Algorithm in Action: Finding the Max

Input



max = 9

1 > max ? **NO**

Output



Algorithm in Action: Finding the Max

Input



max = 9

6 > max ? **NO**

Output



Algorithm in Action: Finding the Max

Input

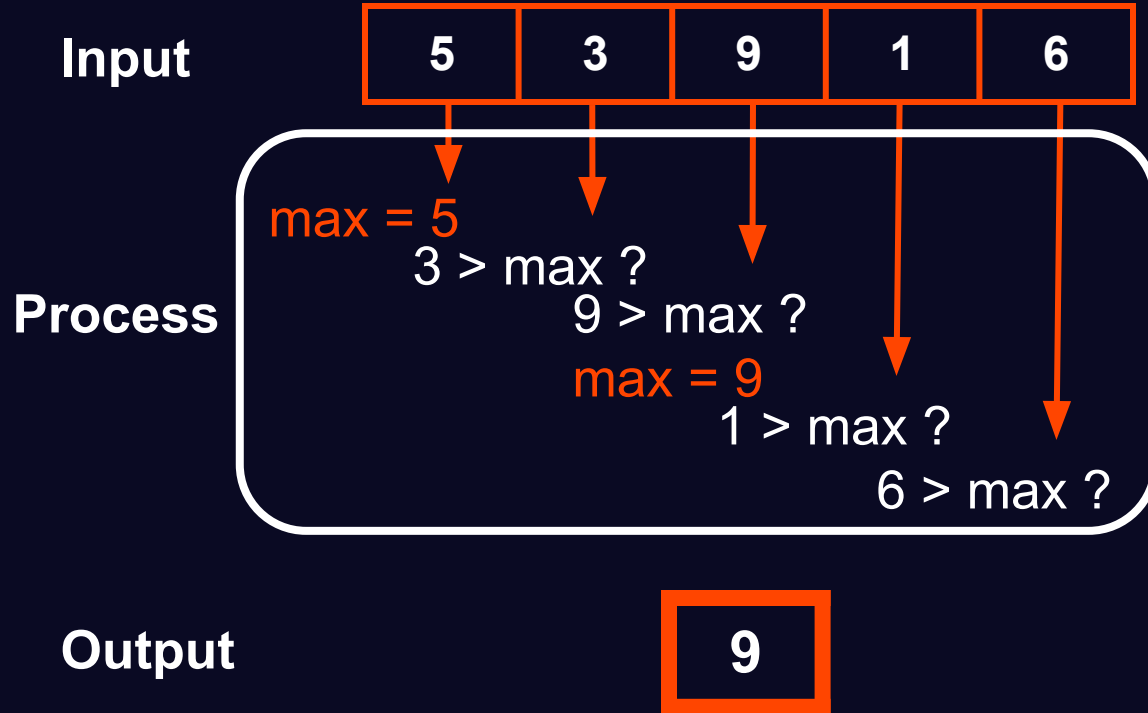


max = 9

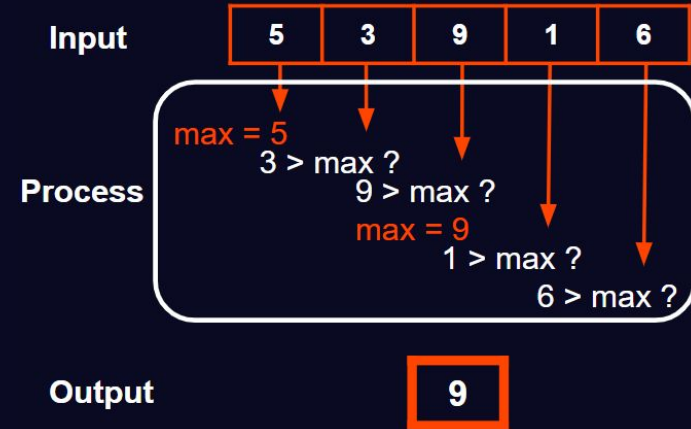
Output



Algorithm in Action: Finding the Max



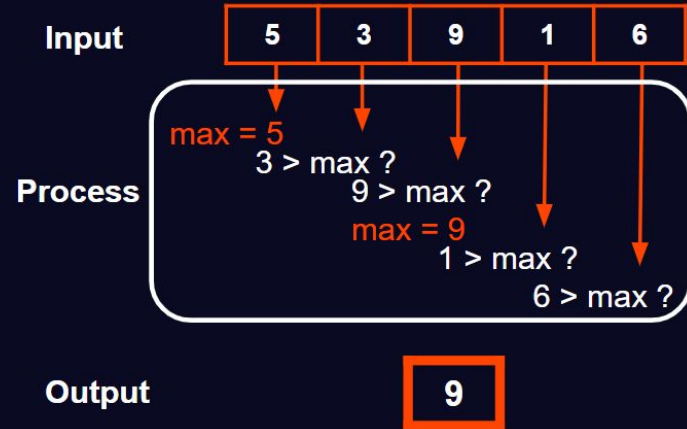
Describing Algorithm: Pseudocode



Describing Algorithm: Pseudocode

Step1:

Start with the **first** number, call it **max**.



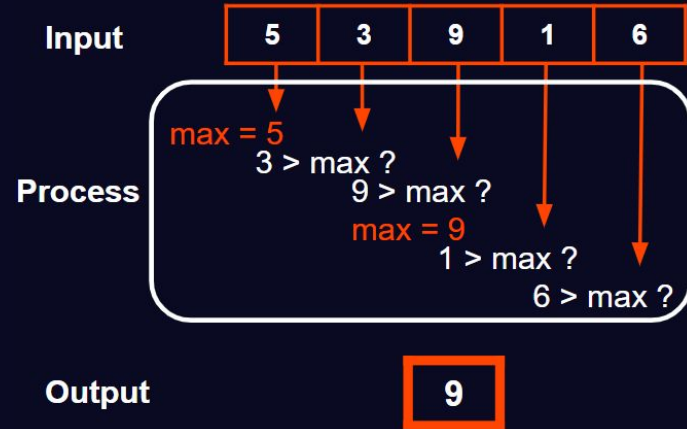
Describing Algorithm: Pseudocode

Step1:

Start with the **first** number, call it **max**.

Step2:

Compare it with **each** number in the list.



Describing Algorithm: Pseudocode

Step1:

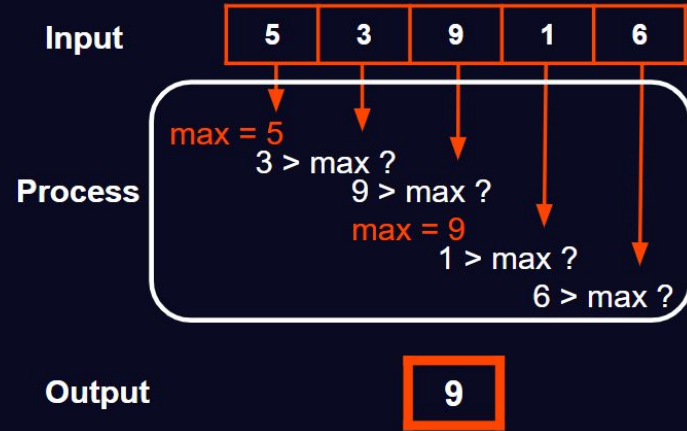
Start with the **first** number, call it **max**.

Step2:

Compare it with **each** number in the list.

Step3:

If you find a **bigger** number, **update max**.



Describing Algorithm: Pseudocode

Step1:

Start with the **first** number, call it **max**.

Step2:

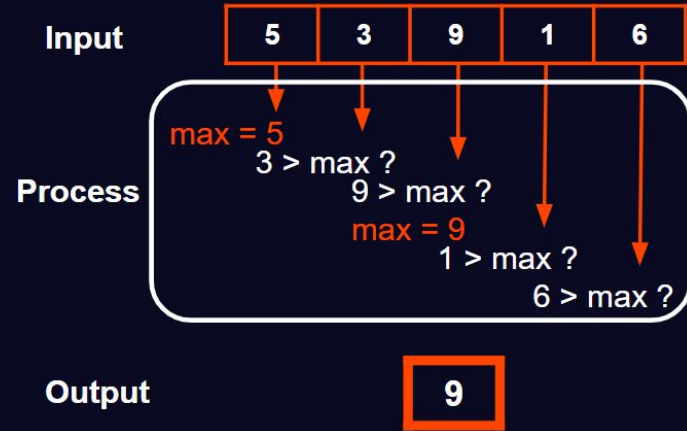
Compare it with **each** number in the list.

Step3:

If you find a **bigger** number, **update max**.

Step4:

At the **end**, **max** is the **biggest** number.



Formalizing Algorithm: Pseudocode



Formalizing Algorithm: Pseudocode

Algorithm FindMax

Input: A list of numbers

Output: The maximum number in the list



Formalizing Algorithm: Pseudocode

Algorithm FindMax

Input: A list of numbers

Output: The maximum number in the list

Begin

`max ← list[0]`

→ Step1: Start with the **first** number, call it **max**.



Formalizing Algorithm: Pseudocode

Algorithm FindMax

Input: A list of numbers

Output: The maximum number in the list

Begin

$\text{max} \leftarrow \text{list}[0]$

Step1: Start with the **first** number, call it **max**.

for each number in list **do**
 if number > max **then**
 $\text{max} \leftarrow \text{number}$
 end if
end for

Step2: **Compare** it with **each** number in the list.

Step3: If you find a **bigger** number, **update max**.



Formalizing Algorithm: Pseudocode

Algorithm FindMax

Input: A list of numbers

Output: The maximum number in the list

Begin

$\text{max} \leftarrow \text{list}[0]$

Step1: Start with the **first** number, call it **max**.

for each number in list do

if number > max then

$\text{max} \leftarrow \text{number}$

end if

end for

return max



Step4: At the **end**, **max** is the **biggest** number.

End



Implementing Algorithm in Python

```
numbers = [5, 3, 9, 1, 6]
```



Implementing Algorithm in Python

```
numbers = [5, 3, 9, 1, 6]
```

```
max_number = numbers[0]
```

Step1: Start with the **first** number, call it **max_number**.

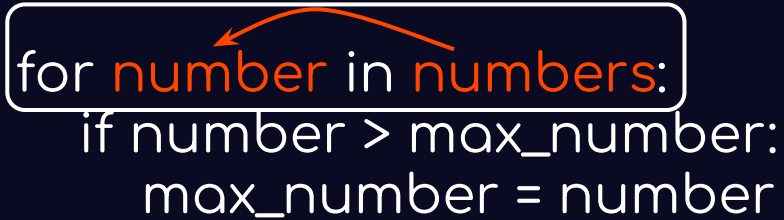


Implementing Algorithm in Python

```
numbers = [5, 3, 9, 1, 6]
```

```
max_number = numbers[0]
```

```
for number in numbers:  
    if number > max_number:  
        max_number = number
```



Step1: Start with the **first** number, call it **max_number**.

Step2: **Compare** it with **each** number in the list.

Step3: If you find a **bigger** number, **update max_number**.



Implementing Algorithm in Python

```
numbers = [5, 3, 9, 1, 6]
```

```
max_number = numbers[0]
```

```
for number in numbers:
```

```
    if number > max_number:  
        max_number = number
```

Step1: Start with the **first** number, call it **max_number**.

Step2: **Compare** it with **each** number in the list.

Step3: If you find a **bigger** number, **update** max_number.



Implementing Algorithm in Python

```
numbers = [5, 3, 9, 1, 6]
```

```
max_number = numbers[0]
```

Step1: Start with the **first** number, call it **max_number**.

```
for number in numbers:  
    if number > max_number:  
        max_number = number
```

Step2: **Compare** it with **each** number in the list.

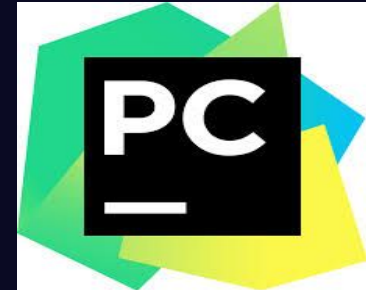
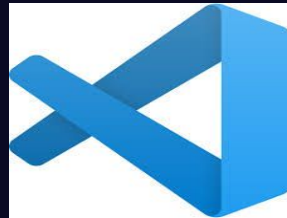
Step3: If you find a **bigger** number, **update max_number**.

```
print("The maximum number is:", max_number)
```

Step4: At the **end**, **max_number** is the **biggest** number.



Running our Python Code



Running our Python Code

<https://pythontutor.com/python-compiler.html>



Converting Algorithm to Function



Converting Algorithm to Function

```
numbers = [5, 3, 9, 1, 6]
max_number = numbers[0]
for number in numbers:
    if number > max_number:
        max_number = number
print("The maximum number is:", max_number)
```



Converting Algorithm to Function

~~numbers = [5, 3, 9, 1, 6]~~

```
max_number = numbers[0]
for number in numbers:
    if number > max_number:
        max_number = number
```

~~print("The maximum number is:", max_number)~~



Converting Algorithm to Function

~~numbers = [5, 3, 9, 1, 6]~~

max_number = numbers[0]

for number in numbers:

if number > max_number:

max_number = number

~~print("The maximum number is:", max_number)~~

sample_numbers = [5, 3, 9, 1, 6]

print("The maximum number is:", max_number)



Converting Algorithm to Function

```
def find_max(numbers):
```

```
    max_number = numbers[0]  
    for number in numbers:  
        if number > max_number:  
            max_number = number
```

```
sample_numbers = [5, 3, 9, 1, 6]  
print("The maximum number is:", max_number)
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
```

```
sample_numbers = [5, 3, 9, 1, 6]
print("The maximum number is:", max_number)
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

```
sample_numbers = [5, 3, 9, 1, 6]
```

```
print("The maximum number is:", max_number)
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

```
sample_numbers = [5, 3, 9, 1, 6]
```

```
print("The maximum number is:", find_max(sample_numbers))
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

```
sample_numbers = [5, 3, 9, 1, 6]
print("The maximum number is:", find_max(sample_numbers))
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Parameters

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

Arguments

```
sample_numbers = [5, 3, 9, 1, 6]
print("The maximum number is:", find_max(sample_numbers))
```



Converting Algorithm to Function

```
def find_max(numbers):
```

Indented Block

```
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

```
sample_numbers = [5, 3, 9, 1, 6]
print("The maximum number is:", find_max(sample_numbers))
```



Converting Algorithm to Function

```
def find_max(numbers):  
    max_number = numbers[0]  
    for number in numbers:  
        if number > max_number:  
            max_number = number  
    return max_number
```



**Function
Definition**

```
sample_numbers = [5, 3, 9, 1, 6]  
print("The maximum number is:", find_max(sample_number))
```



Converting Algorithm to Function

```
def find_max(numbers):
```

```
    max_number = numbers[0]
```

```
    for number in numbers:
```

```
        if number > max_number:
```

```
            max_number = number
```

```
    return max_number
```

Function
Definition

Function
Call

```
sample_numbers = [5, 3, 9, 1, 6]
```

```
print("The maximum number is:", find_max(sample_number))
```



Summary and What's Next

- Summary
 - Understanding Algorithms
 - Pseudocode
 - Python Implementation
 - Functions



Summary and What's Next

- Summary
 - Understanding Algorithms
 - Pseudocode
 - Python Implementation
 - Functions
- Exercise
 - An exercise to count the steps it takes to find the maximum number.
 - Solution is provided
- Link to examples, exercises, and solutions
 - <https://github.com/rmateeq/algorismo>



Summary and What's Next

- Summary

- Understanding Algorithms
- Pseudocode
- Python Implementation
- Functions

- **Next Topic**

- **Linear Search**

- Exercise

- An exercise to count the steps it takes to find the maximum number.
- Solution is provided

- Link to examples, exercises, and solutions

- <https://github.com/rmateeq/algorismo>





Thank You for Watching!