



INTRO TO PROGRAMMING

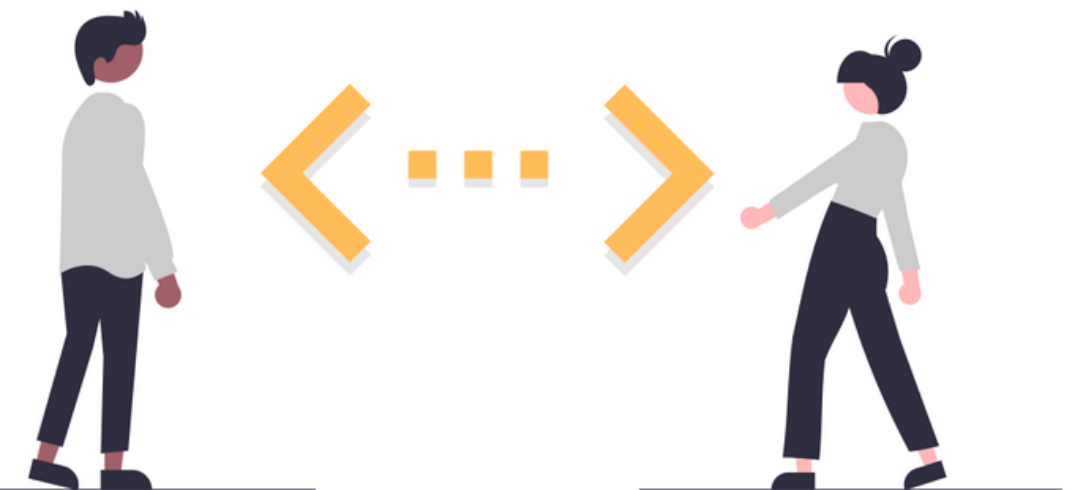
"ALGORITHM"

WORD USED BY PROGRAMMERS WHEN
THEY DON'T WANT TO EXPLAIN
WHAT THEY DID



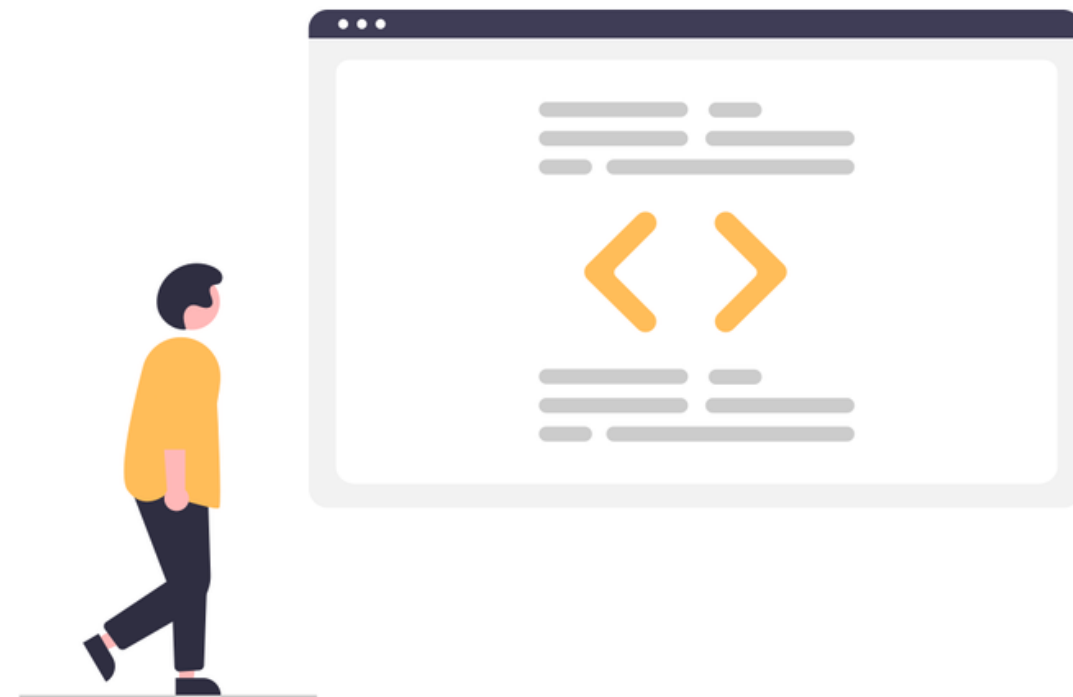
The main questions

THE PROBLEM, THE RECOGNITION & THE SOLUTION





What is a program?





Why should I use a
program?



Why should I use a program?

A computer program is a sequence or set of instructions in a programming language for a computer to execute.





Why should I use a program?

- A regular procedure becomes much easier
- Provides security from user errors
- Provides additional data analysis



“FIRST, SOLVE THE
PROBLEM. THEN, WRITE
THE CODE.”

-JOHN JOHNSON

How do I develop a program?

1

ANALYZE

2

3

4

5

6



How do I develop a program?

1

ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2

3

4

5

6



How do I develop a program?

1

ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2

DESIGN

3

4

5

6



How do I develop a program?

1

ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2

DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3

4

5

6



How do I develop a program?

1

ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2

DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3

CODE

4

5

6

How do I develop a program?

1

ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2

DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3

CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4

5

6

How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

5

6



How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

This is the process of finding "bugs" in the computer.
Errors are important to find because they affect the proper functioning of a program.

5

6



How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

This is the process of finding "bugs" in the computer.
Errors are important to find because they affect the proper functioning of a program.

5 FORMALIZE

6



How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

This is the process of finding "bugs" in the computer.
Errors are important to find because they affect the proper functioning of a program.

5 FORMALIZE

Someone needs to run the program to make sure there are no syntax and logic errors.
Syntax errors are grammatical errors and logic errors are incorrect results.

6



How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

This is the process of finding "bugs" in the computer.
Errors are important to find because they affect the proper functioning of a program.

5 FORMALIZE

Someone needs to run the program to make sure there are no syntax and logic errors.
Syntax errors are grammatical errors and logic errors are incorrect results.

6 DOCUMENT



How do I develop a program?

1 ANALYZE

The user must understand the problem and then, decide how to solve the problem – choose a program.

2 DESIGN

A flowchart is important to essential.
This is a visual diagram of the command flow that the program will contain.

3 CODE

The user uses the programming language to write the lines of code.
The code is called an entry or source code.

4 DEBUG

This is the process of finding "bugs" in the computer.
Errors are important to find because they affect the proper functioning of a program.

5 FORMALIZE

Someone needs to run the program to make sure there are no syntax and logic errors.
Syntax errors are grammatical errors and logic errors are incorrect results.

6 DOCUMENT

The documentation explains the rationale that one could, make a change to the program or how to write a program

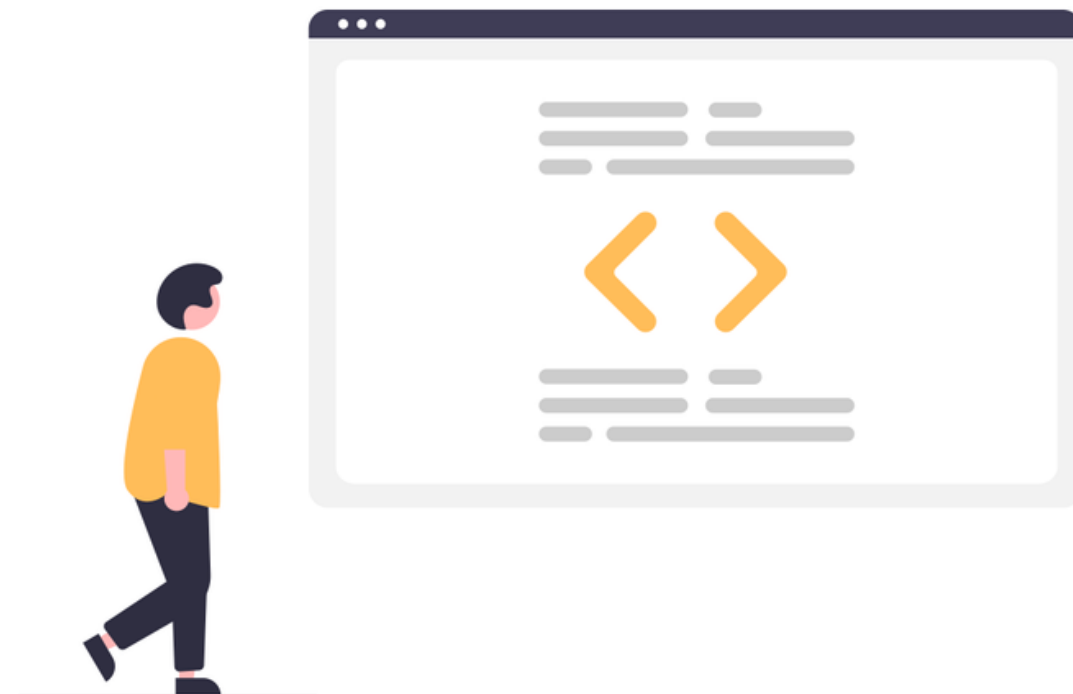




What is the best
programming language?



Πως διαλέγω την
κατάλληλη γλώσσα
προγραμματισμού?





How do I choose the
right programming
language?

Criteria

Application

Resources & Limitations

Support & Documentation

Scalability and performance





C

```
int main(void)
{
    printf("hello, world\n");
}
```

JAVA

```
class HelloWorldApp {
    public static void
main(String[] args) {

    System.out.println("Hello
World!");
    }
}
```

PYTHON

```
print("Hello World")
```

C++





```
int main()
{
    std::cout << "Hello,
world!\n";
    return 0;
}
```

JAVASCRIPT

```
console.log("Hello World!");
```

PHP

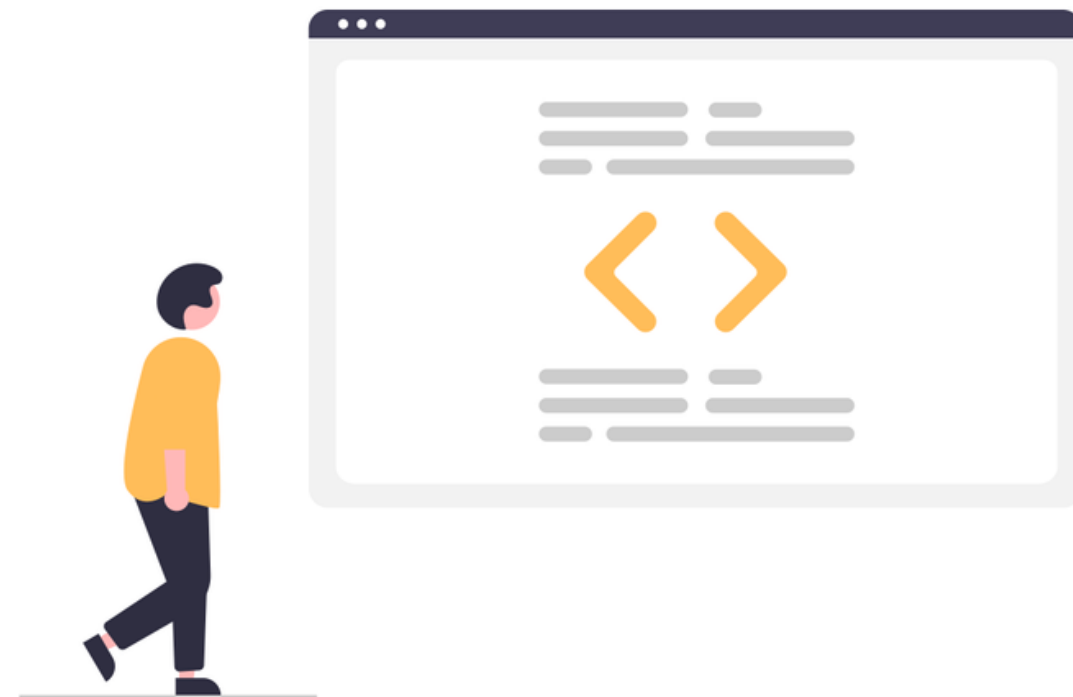
```
<?php echo "Hello, World";
```



Introduction to the elements of programming



What does a program
consist of?





What does a program
consist of?

Values & Data types

Variables & Operations

Functions

Conditions & loops

Libraries

Values & Data types

```
#Data Types
```

```
int: 1
```

```
float: 3.14
```

```
char: "S"
```

```
string: "STEAMing the Future"
```

```
bool: True, False --> (1,0)
```

```
list: [1,2,3], ["STEAMing", "the", "Future"],  
      [True, False, False], [[1,2],[3,4]]
```

```
dict: {"Day":3, "Month":12, "Year":2022}
```

```
tuple: (1.618,2.718)
```

Variables & Operations

```
#Variables
```

```
x = 1  
y = 2
```

```
x = 1.732  
y = 1.414
```

```
Name = "Mike"  
Surname = "Wheeler"
```

```
Full_Name = "Mike Wheeler"
```

```
teamX = ["Andreas", "Ilias", "Eugenia", "Nefeli", "Rafaelia"]
```

```
date = (3, 12, 2022)
```

Variables & Operations

#Operators

```
a = 1 + 2
```

```
b = 2 * 2
```

```
c = 4 / 2
```

```
d = 1.732 ** 2
```

```
e = 5 % 2
```

```
f = 10 // 3
```

```
g = "Mike" + " " + "Wheeler"
```

```
a += 1
```

```
b /= 2
```

Functions

```
# Display something on the console.  
print("Hello world!")
```

```
# Summarize some values.  
sum([1,2,3,4,5])
```

```
# Ask the user to type something.  
input("Please enter your name: ")
```

```
# Find the minimum and maximum value.  
max(1,2,3)  
min("a", "b", "c")
```

```
# Absolute value.  
abs(-1)
```

```
# Raise to a power.  
pow(2,2)
```

```
# Find the length of a list.  
len([1,2,2,-1])
```

```
# Convert to int.  
int(2.2)
```

```
# Convert to string.  
str(1)
```

Functions

```
def my_function(a,b):  
    print("Initial Values:")  
    print(a,b)  
    c = a  
    a = b  
    b = c  
    print("Final Values:")  
    print(a,b)  
  
my_function(3,4)
```

Conditions & loops

```
#Conditions
```

```
list1 = [1,5,7,5]
```

```
if list1[0] == list1[1]:  
    print("The values are equal")
```

```
else:
```

```
    print("The values are not equal")
```

```
if list1[0] < list1[1]:  
    print(list1[1])
```

```
elif list1[0] > list1[1]:  
    print(list1[0])
```

```
else:
```

```
    print("Equal")
```


Conditions & loops

```
#Loops
```

```
for i in range(10):  
    x = pow(i,3)
```

```
    if x == 64:  
        print(i)
```

```
for i in teamX:  
    print("Here is " + i)
```

```
for i in range(10):  
    for j in range(100):  
        print(i**j)
```

Conditions & loops

```
#Loops
```

```
temperature = 0
```

```
while temperature < 20:  
    print("I'm staying inside.")
```

```
    temperature += 1
```

```
print("Ok, now I can go out.")
```

```
while True:  
    print("Wakanda forever")
```

Libraries

```
#Libraries

import math

a = 3
b = 4
c = math.sqrt(a**2 + b**2)
print(c)

from math import *

c = sqrt(a**2 + b**2)
```

QUESTIONS

