



# INTRO TO PROGRAMMING

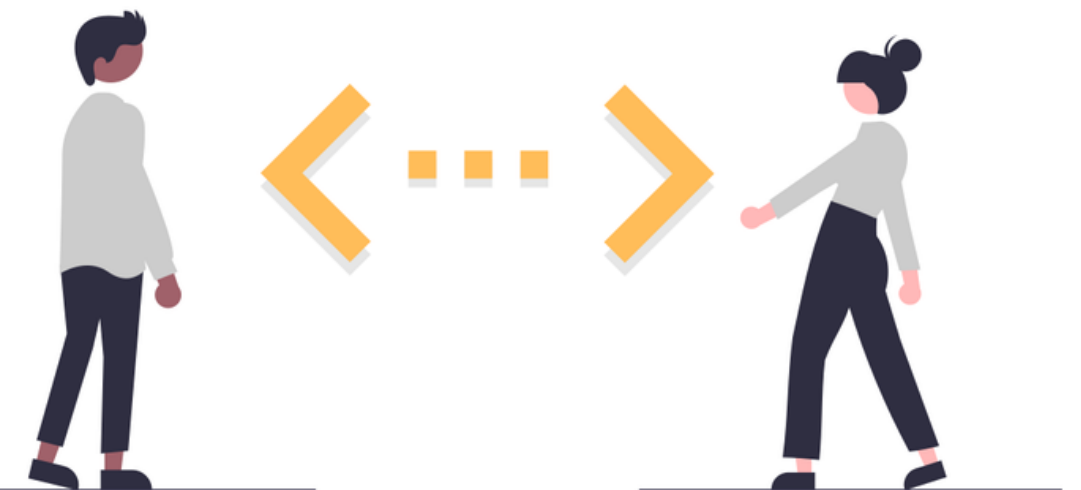
"ALGORITHM"

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



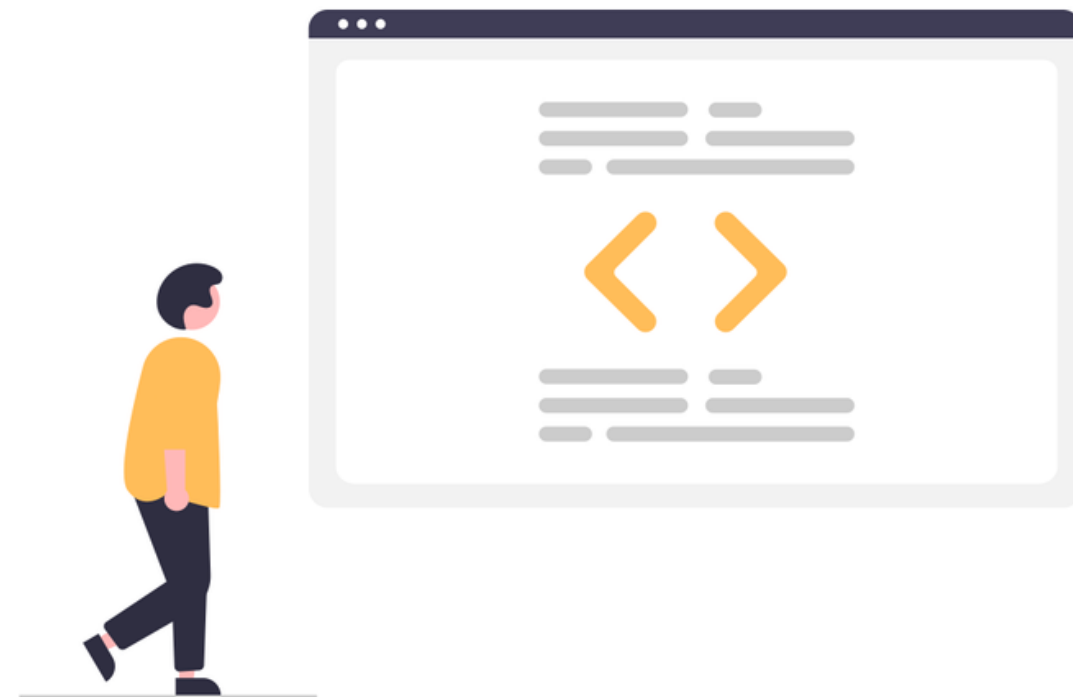
# Οι βασικές ερωτήσεις

ΤΟ ΠΡΟΒΛΗΜΑ, Η ΑΝΑΓΝΩΡΙΣΗ & Η ΕΠΙΛΥΣΗ





Τι είναι ένα πρόγραμμα  
aka program?





Γιατί να χρησιμοποιήσω  
πρόγραμμα?



Τι είναι ένα πρόγραμμα  
aka program?

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





Γιατί να χρησιμοποιήσω πρόγραμμα?

- Μια τακτική διαδικασία γίνεται πολύ πιο εύκολη
- Παρέχει ασφάλεια από σφάλματα χρήστη
- Παρέχει πρόσθετη ανάλυση δεδομένων



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

-JOHN JOHNSON

# Πως αναπτύσω ένα πρόγραμμα?

1 ANALYZE

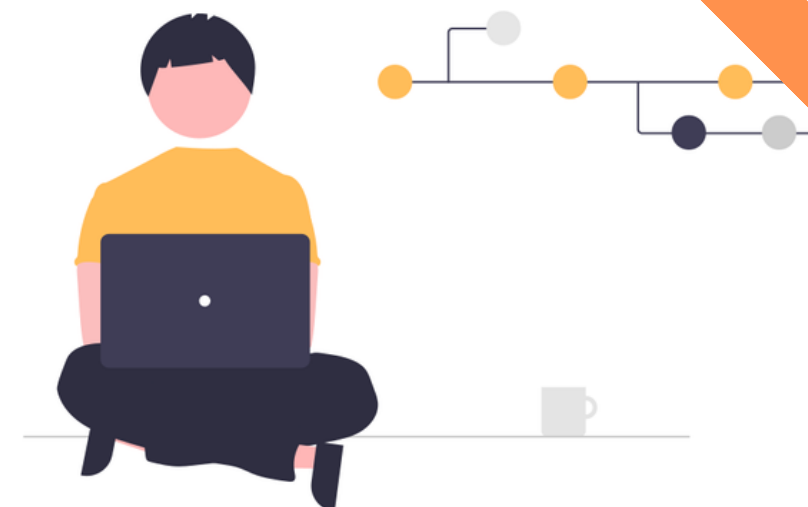
2

3

4

5

6





# Πως αναπτύσω ένα πρόγραμμα?

1

## ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

2

3

4

5

6



# Πως αναπτύσω ένα πρόγραμμα?

1

## ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

2

## DESIGN

3

4

5

6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

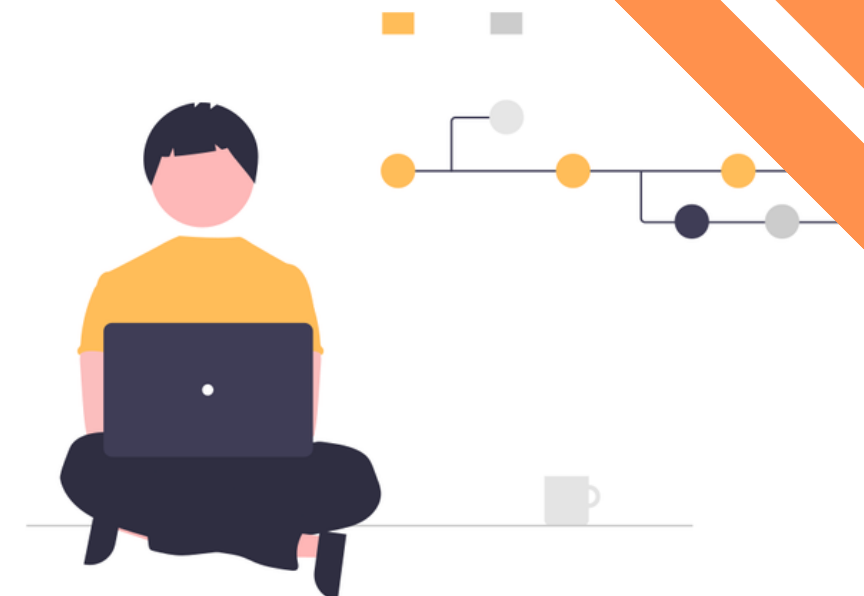
Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

3

4

5

6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

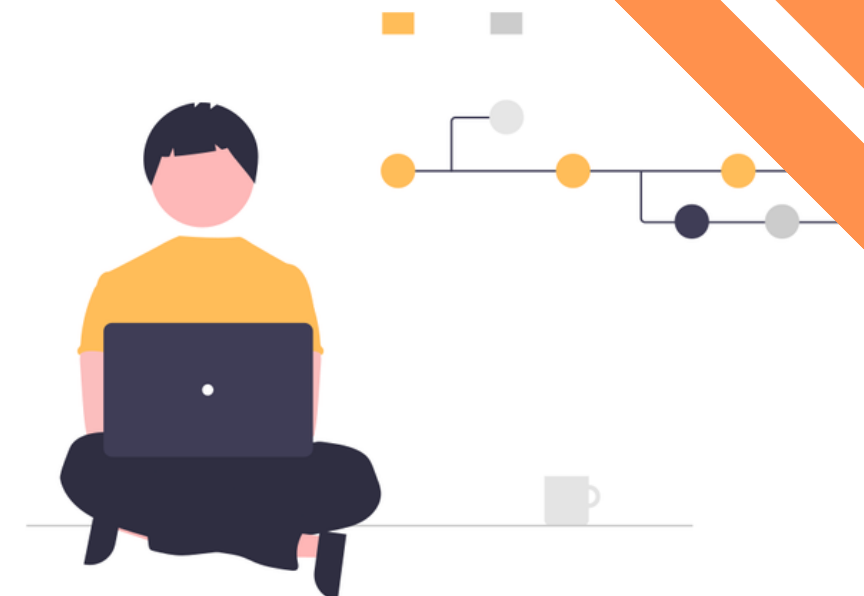
Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

4

5

6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

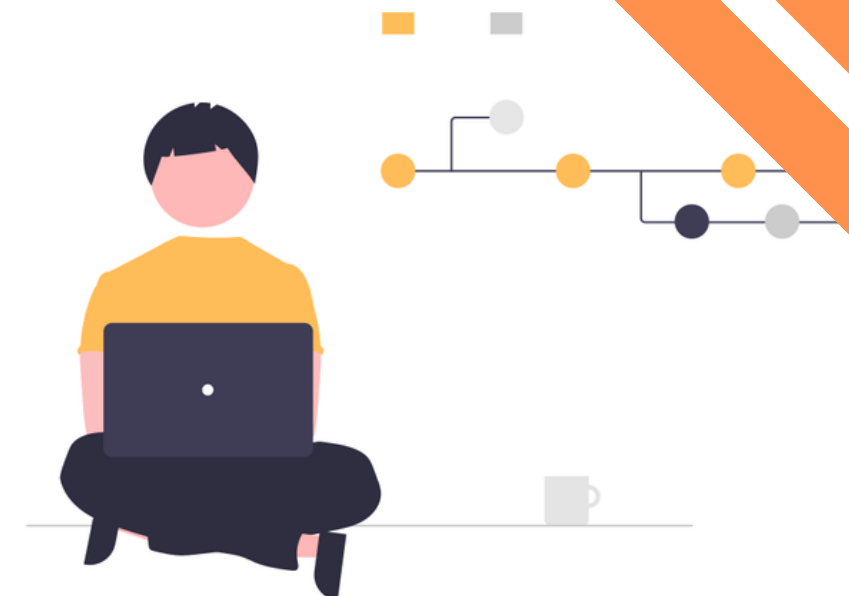
## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

4

5

6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

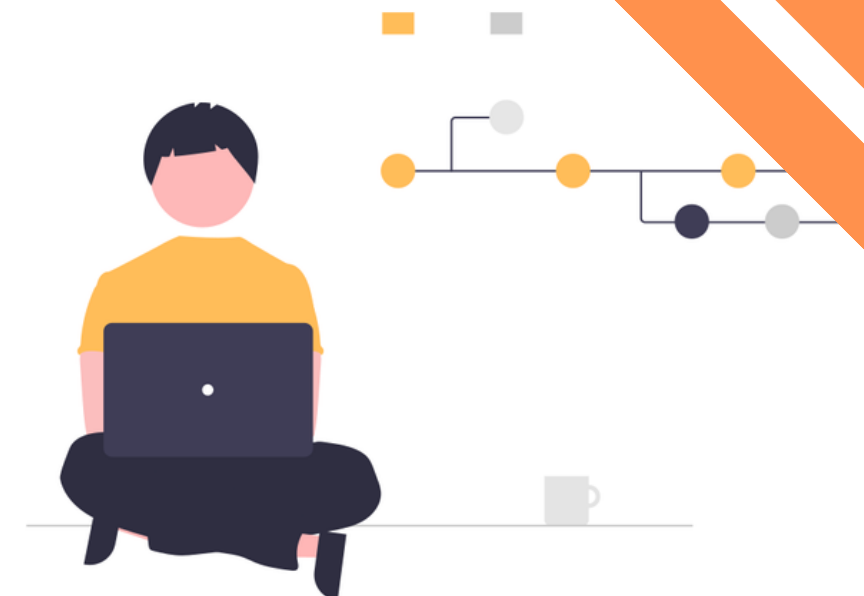
## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

5

6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

Αυτή είναι η διαδικασία εύρεσης των «σφαλμάτων» στον υπολογιστή. Τα σφάλματα είναι σημαντικό να βρεθούν επειδή επηρεάζουν τη σωστή λειτουργία ενός προγράμματος.

## 5

## 6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

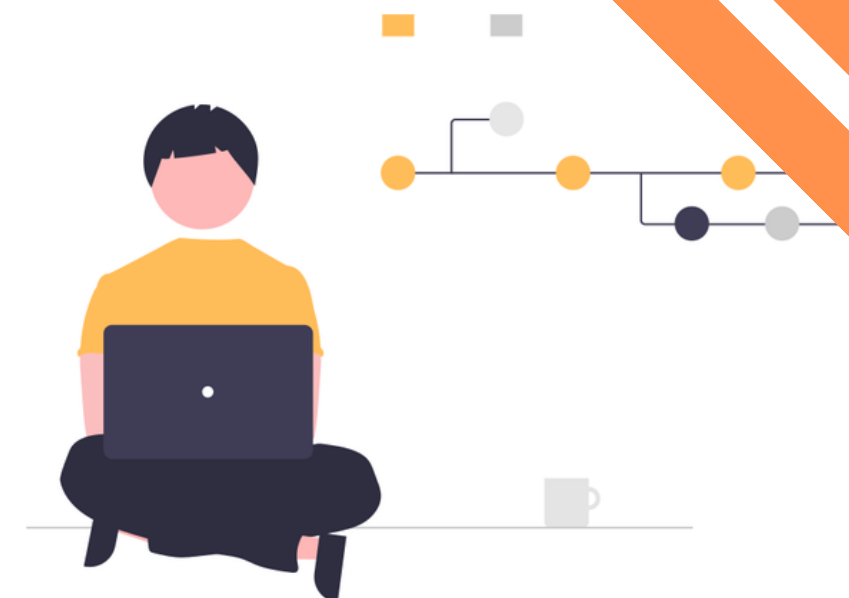
Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

Αυτή είναι η διαδικασία εύρεσης των «σφαλμάτων» στον υπολογιστή. Τα σφάλματα είναι σημαντικό να βρεθούν επειδή επηρεάζουν τη σωστή λειτουργία ενός προγράμματος.

## 5 FORMALIZE

## 6





# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

Αυτή είναι η διαδικασία εύρεσης των «σφαλμάτων» στον υπολογιστή. Τα σφάλματα είναι σημαντικό να βρεθούν επειδή επηρεάζουν τη σωστή λειτουργία ενός προγράμματος.

## 5 FORMALIZE

Κάποιος πρέπει να τρέξει το πρόγραμμα για να βεβαιωθεί ότι δεν υπάρχουν λάθη σύνταξης και λογικής. Τα συντακτικά είναι γραμματικά λάθη και τα λογικά σφάλματα είναι εσφαλμένα αποτελέσματα.

## 6



# Πως αναπτύσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

Αυτή είναι η διαδικασία εύρεσης των «σφαλμάτων» στον υπολογιστή. Τα σφάλματα είναι σημαντικό να βρεθούν επειδή επηρεάζουν τη σωστή λειτουργία ενός προγράμματος.

## 5 FORMALIZE

Κάποιος πρέπει να τρέξει το πρόγραμμα για να βεβαιωθεί ότι δεν υπάρχουν λάθη σύνταξης και λογικής. Τα συντακτικά είναι γραμματικά λάθη και τα λογικά σφάλματα είναι εσφαλμένα αποτελέσματα.

## 6 DOCUMENT



# Πως αναπτύσσω ένα πρόγραμμα?

## 1 ANALYZE

Ο χρήστης πρέπει να καταλάβει το πρόβλημα και, στη συνέχεια, να αποφασίσει πώς να επιλύσει το πρόβλημα – επιλέξτε ένα πρόγραμμα.

## 2 DESIGN

Ένα διάγραμμα ροής είναι σημαντικό έως απαραίτητο. Αυτό είναι ένα οπτικό διάγραμμα της ροής εντολών που θα περιέχει το πρόγραμμα.

## 3 CODE

Ο χρήστης χρησιμοποιεί τη γλώσσα προγραμματισμού για να γράψει τις γραμμές κώδικα. Ο κώδικας ονομάζεται καταχώριση ή πηγαίος κώδικας.

## 4 DEBUG

Αυτή είναι η διαδικασία εύρεσης των «σφαλμάτων» στον υπολογιστή. Τα σφάλματα είναι σημαντικό να βρεθούν επειδή επηρεάζουν τη σωστή λειτουργία ενός προγράμματος.

## 5 FORMALIZE

Κάποιος πρέπει να τρέξει το πρόγραμμα για να βεβαιωθεί ότι δεν υπάρχουν λάθη σύνταξης και λογικής. Τα συντακτικά είναι γραμματικά λάθη και τα λογικά σφάλματα είναι εσφαλμένα αποτελέσματα.

## 6 DOCUMENT

Η τεκμηρίωση εξηγεί το σκεπτικό που θα μπορούσε κάποιος, να κάνει μια αλλαγή στο πρόγραμμα ή πώς να γράψει ένα πρόγραμμα

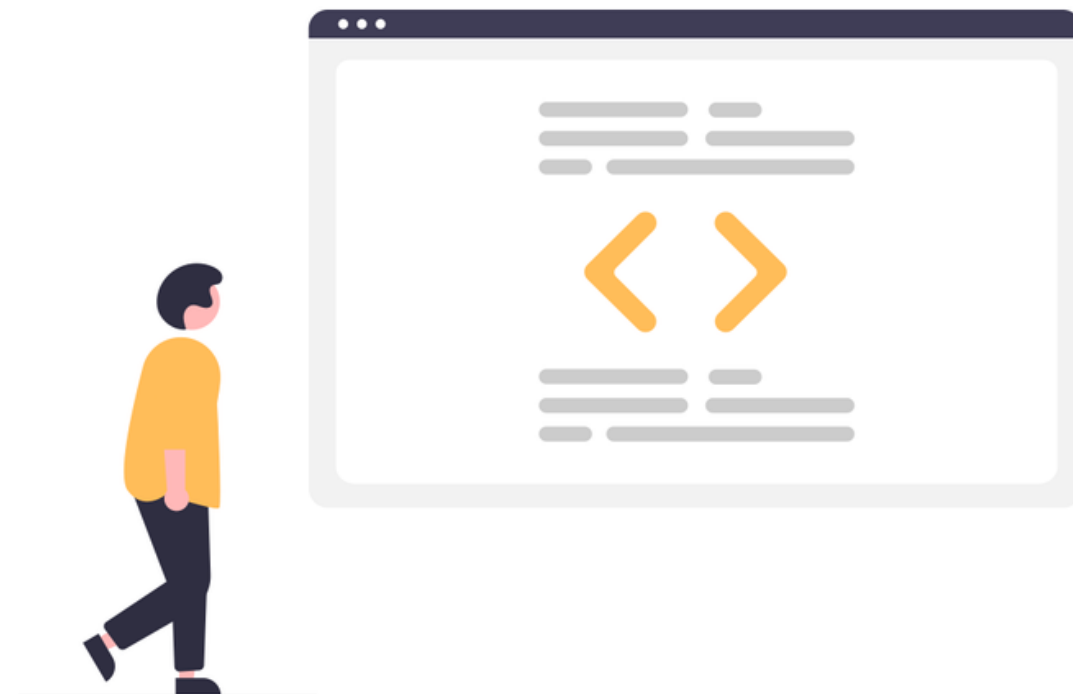




Ποια είναι η καλύτερη  
γλώσσα προγραμματισμού?



Πως διαλέγω την  
κατάλληλη γλώσσα  
προγραμματισμού?





Πως διαλέγω την  
κατάλληλη γλώσσα  
προγραμματισμού?

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("Hell
o 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";
```

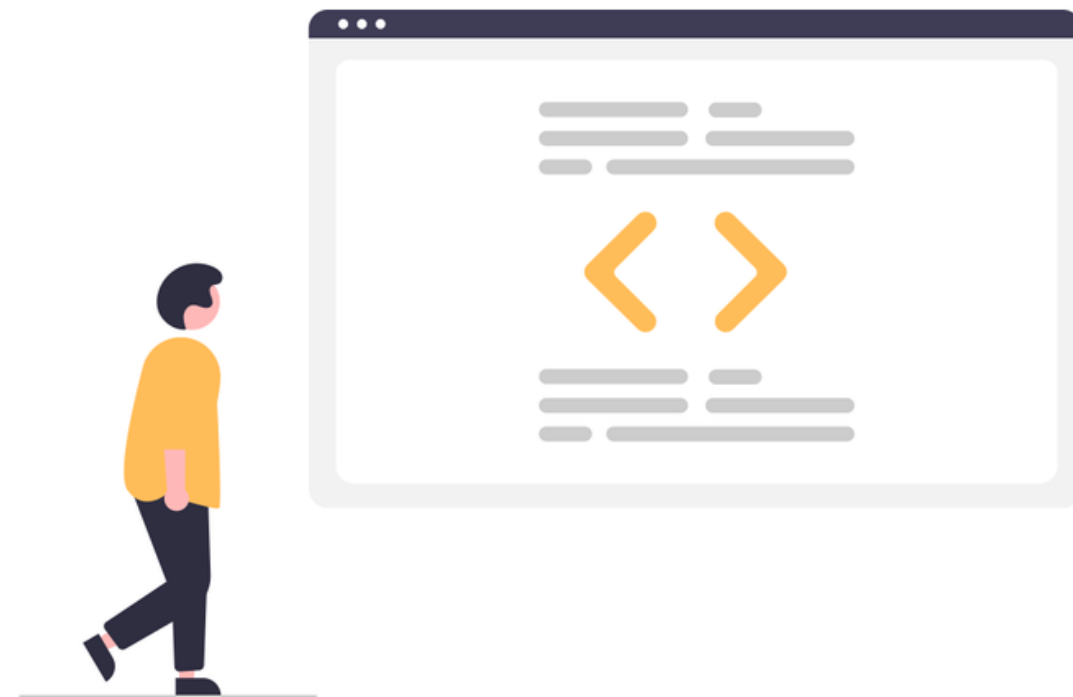
# Εισαγωγή στα στοιχεία προγραμματισμού







Από τι αποτελείται ένα πρόγραμμα?





Από τι αποτελείται ένα πρόγραμμα?

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



**STE(A)M PARTNERSHIPS**

**Education Resilience in Europe**

SCIENTIX  
The community for science education in Europe

CISCO

European Schoolnet

The STEAM Partnerships has been funded under the European Union's Horizon research and innovation programme - project SciEntia A, coordinated by European Schoolnet (ESN). The content of the document is the sole responsibility of the organizers and does not represent the opinion of the European Commission (EC), and the EC is not responsible for any use that might be made of information contained therein.