

# Δημιουργία Ιστοσελίδας

Η ιστοσελίδα που θα δημιουργήσουμε είναι απλή σελίδα HTML που θα περιέχει τα δεδομένα των μετρήσεων.

## ΤΕΣΤ Ιστοσελίδα

Η ιστοσελίδα είναι απλή HTML και έχει την ακόλουθη μορφή

```
<!DOCTYPE HTML>
<html>
<head>Meteors @ 1st Lyceum</head>
<body>
<h1>IoT Meteorologic Hub @ 1st Lyceum of Serres</h1>
<h2>This is a TEST Page</h2><br />
</body>
</html>
```

Και χρησιμοποιείται ως μια ΤΕΣΤ ιστοσελίδα για να ελέγξουμε αν λειτουργεί σωστά ο Arduino Server.

## Παρατηρήσεις

- Π.1.** Ό,τι περικλύεται μεταξύ των <html> και </html> είναι το περιεχόμενο της ιστοσελίδας.
- Π.2.** Ό,τι περικλύεται μεταξύ των <head> και </head> θα εμφανιστεί ως τίτλος στην καρτέλα του browser που χρησιμοποιούμε.
- Π.3.** Ό,τι περικλύεται μεταξύ των <body> και </body> είναι το κυρίως σώμα της ιστοσελίδας.
- Π.4.** Το κείμενο που περικλύεται μεταξύ των <h1> και </h1> θα εμφανιστεί αρκετά έντονα.
- Π.5.** Το κείμενο που περικλύεται μεταξύ των <h2> και </h2> θα εμφανιστεί έντονα (αλλά λιγότερο από το h1).

# Arduino Κώδικας της ΤΕΣΤ Ιστοσελίδας

Η παραπάνω ιστοσελίδα υλοποιείται με τον ακόλουθο Arduino Κώδικα

```
// Απαραίτητες Βιβλιοθήκες
#include <SPI.h> // Βιβλιοθήκη σειριακής επικοινωνίας με το Ethernet Module
#include <Ethernet.h> // Βιβλιοθήκη λειτουργιών του Ethernet

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC Address του Ethernet Module
byte ip[] = { 192, 168, 1, 177 }; // Διεύθυνση IP του Ethernet Module
EthernetServer server(80); // Ορισμός Server στο port 80

void setup() { // Αρχικοποίηση
  Serial.begin(9600); // Εκκίνηση της σύνδεσης στην οθόνη
  Ethernet.begin(mac, ip); // Εκκίνηση λειτουργίας του Ethernet Module
  server.begin(); // Εκκίνηση του Server
  Serial.print("server is at IP: "); // Μήνυμα στην οθόνη ότι ο Server λειτουργεί στη διεύθυνση IP:
  Serial.println(Ethernet.localIP()); // 192.168.1.177
  Serial.println("So, connect there!"); // Προτροπή για σύνδεση σε αυτήν τη διεύθυνση
}

void loop() { // Επαναλαμβανόμενο πρόγραμμα - Εντολές προς τον Server
  EthernetClient client = server.available(); // Περίμενε αιτήσεις για σύνδεση
  if (client) { // Αν υπάρξει αίτηση για σύνδεση από πελάτη (client) = browser
    Serial.println("new client"); // Τύπωσε στην οθόνη το μήνυμα "new client"
    boolean currentLineIsBlank = true; // κάθε αίτηση τελειώνει με μια κενή γραμμή, προς το παρόν αληθές
    while (client.connected()) { // όσο ο πελάτης είναι συνδεδεμένος
      if (client.available()) { // αν είναι διαθέσιμος
        char c = client.read(); // ξεκίνα να διαβάζεις την αίτηση του πελάτη γράμμα προς γράμμα
        Serial.write(c); // Γράψε την αίτηση στην οθόνη
        if (c == '\n' && currentLineIsBlank) { // Αν έφτασες σε κενή γραμμή
          client.println("HTTP/1.1 200 OK"); // Τύπος HTTP
          client.println("Content-Type: text/html"); // Το κείμενο που θα ακολουθήσει είναι μορφής
          // HTML, δηλαδή είναι ιστοσελίδα
          client.println("Connection: close"); // Η σύνδεση θα κλείσει μετά την προβολή της σελίδας
          client.println("Refresh: 5"); // Αυτόματη ανανέωση σελίδας κάθε 5 δευτερόλεπτα
          client.println(); // Κενή γραμμή
          // Εμφάνιση της Ιστοσελίδας που αναφέρθηκε πιο πάνω
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          client.println("<body>");
          client.println("<h1>IoT Meteorologic Hub @ 1st Lyceum of Serres</h1>");
          client.print("<h2> This is a TEST Page");
          client.println("</h2><br />");
          client.println("</body>");
          client.println("</html>");
          // Τέλος Ιστοσελίδας
          break; // Έξοδος από το βρόχο while
        }
      }
      if (c == '\n') { // Αν έφτασες σε αλλαγή γραμμής
        currentLineIsBlank = true; // Τότε θέσε τη μεταβλητή currentLineIsBlank αληθή, ώστε να εμφανιστεί η σελίδα
      } else if (c != '\r') { // Αλλιώς αν δεν έχει πατηθεί το enter
        currentLineIsBlank = false; // βρίσκεσε στην ίδια γραμμή, οπότε συνέχισε το διάβασμα της αίτησης
      }
    }
  }
  delay(1); // Δώσε χρόνο στον browser να λάβει τα δεδομένα
  client.stop(); // Κλείσε τη σύνδεση
  Serial.println("client disconnected"); // Τύπωσε στην οθόνη ότι η σύνδεση τερματίστηκε
}
```

# Ιστοσελίδα Μετρήσεων

Ο κώδικας Arduino για την εμφάνιση των μετρήσεων είναι ο ακόλουθος.

Είναι σημειωμένα μόνο τα επιλέον στοιχεία, τα οποία και σχολιάζουμε.

```
#include <SPI.h>
#include <Ethernet.h>
#include "DHT.h" // Βιβλιοθήκες χειρισμού του αισθητήρα DHT11
#include <Adafruit_Sensor.h> // Βιβλιοθήκες χειρισμού αισθητήρων της εταιρίας Adafruit
#include <Adafruit_BMP280.h> // Βιβλιοθήκες χειρισμού του αισθητήρα BMP280 της εταιρίας Adafruit

#define DHTPIN 6 // Ο DHT11 είναι συνδεδεμένος στο Pin 6 του Arduino
#define DHTTYPE DHT11 // Ο Τύπος DHT είναι ο DHT11
#define BMP_SCK 5 // Έξοδος SCK του αισθητήρα στο Pin 5 του Arduino
#define BMP_MISO 4 // Έξοδος SDO του αισθητήρα στο Pin 4 του Arduino
#define BMP_MOSI 3 // Έξοδος SDI του αισθητήρα στο Pin 3 του Arduino
#define BMP_CS 2 // Έξοδος CS του αισθητήρα στο Pin 2 του Arduino

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80);

DHT dht(DHTPIN, DHTTYPE); // Αντικείμενο Μετρήσεων του DHT11
Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK); // Αντικείμενο Μετρήσεων του BMP280

void setup() {
  dht.begin(); // Εκκίνηση λειτουργίας αισθητήρα DHT11
  bmp.begin(); // Εκκίνηση λειτουργίας αισθητήρα BMP280
  Serial.begin(9600);
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at IP: ");
  Serial.println(Ethernet.localIP());
  Serial.println("So, connect there!");
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n' && currentLineIsBlank) {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println("Refresh: 5");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");

          client.println("<body>");
          client.println("<h1>IoT Meteorologic Hub @ 1st Lyceum of Serres</h1>");

          client.print("<h2>Temperature: "); // Θερμοκρασία
          client.print(bmp.readTemperature()); // Τιμή Θερμοκρασίας
          client.print(" C"); // Μονάδα Μέτρησης (C)
          client.println("<br />"); // Αλλαγή γραμμής
        }
      }
    }
  }
}
```

```

client.print("Humidity: ");           // Υγρασία
client.print(dht.readHumidity());     // Τιμή Υγρασίας
client.print(" %");                  // %
client.println("<br />");              // Αλλαγή γραμμής

client.print("Pressure: ");           // Πίεση
client.print(int(bmp.readPressure()/100)); // Τιμή Πίεσης
client.print(" mbar");                // Μονάδα Μέτρησης (mbar)
client.println("</h2><br />");        // Αλλαγή γραμμής

client.println("</body>");
client.println("</html>");
break;
}
if (c == '\n') {
    currentLineIsBlank = true;
} else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
delay(1);
client.stop();
Serial.println("client disconnected");
}
delay(5000); // Αναμονή 5.000ms = 5s μέχρι την έναρξη της επόμενης μέτρησης
}

```