

Corso front-end BASE

Mirko Campari

4 capitoli e 4 test

Cosa affronteremo in questo capitolo

Angular

- Architettura e installazione
- Introduzione a Typescript

Node JS

- Cos'è Node.js
- NodeJS Command Line Interface

Test teorici

- Introduttivo e di conoscenza base
- Di fine capitolo

Pratica

- Primi script con TypeScript, JS e java
- Test pratico di gruppo

Quattro capitoli totali con 4 test pratici e teorici

Cosa sono back end e front end

I termini inglesi **front end** e **back end** in informatica denotano, rispettivamente, la parte visibile all'utente di un programma e con cui egli può interagire, nella maggior parte dei casi, un'interfaccia utente, e la parte che permette l'effettivo funzionamento di queste interazioni e limitando le possibilità la dove necessario.

Il **front end**, nella sua accezione più generale, è responsabile dell'acquisizione dei dati di ingresso e della loro elaborazione con modalità conformi a specifiche predefinite e invarianti, tali da renderli utilizzabili dal **back end**.

Il collegamento del **front end** al **back end** è un caso particolare di interfaccia.

Cosa sono back end e front end prt 2

Dopo la pura teoria ora andiamo a confrontare le differenze tra lo sviluppo back-end e quello front-end, le differenze di solito si presentano nelle seguenti categorie oltreché nelle specifiche tecniche del progetto:

- **Le aree di cui si occupano questi due tipi di sviluppo.**
- **La tecnologia e gli strumenti utilizzati da ciascuno.**
- **I linguaggi utilizzati nel backend rispetto al frontend.**
- **Requisiti e competenze professionali.**

Cos'è *HTML*?

In informatica l'HyperText Markup Language, cioè linguaggio a marcatori per ipertesti, comunemente noto con l'acronimo HTML, è un linguaggio di markup.

Nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web 1.0, oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web definita appunto dal markup e la sua rappresentazione, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C). È derivato dall'SGML, un metalinguaggio finalizzato alla definizione di linguaggi utilizzabili per la stesura di documenti destinati alla trasmissione in formato elettronico.

La versione attuale, la quinta, è stata rilasciata dal W3C nell'ottobre 2014.

Un esempio di HTML

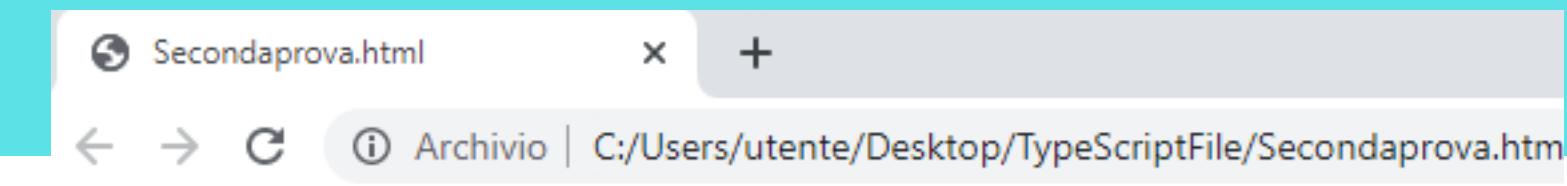
Andiamo ora a vedere un esempio di HTML

```
<!DOCTYPE html>
<html><body>

<h1 style="background-color:DodgerBlue;">Salve Aula!</h1>

<p style="background-color:Tomato;">
Lorem ipsum </p>

</body></html>
```



Quello che vedremo accadere dev'essere
qualcosa di simile a questo

----->

Salve Aula!

-----> **Lorem ipsum**

Cos'è il CSS?

Il CSS, sigla per Cascading Style Sheets, in italiano fogli di stile a cascata; è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML, ad esempio i siti web e relative pagine web. Le regole per comporre il CSS sono contenute in un insieme di direttive emanate da W3C.

L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine stesse sia per gli utenti, garantendo contemporaneamente anche il riutilizzo di codice ed una sua più facile manutenzione.

Cos'è Java Script?

In informatica JavaScript è un linguaggio di programmazione multi paradigma orientato agli eventi, comunemente utilizzato nella programmazione Web lato client (esteso poi anche al lato server) per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

Originariamente sviluppato da Brendan Eich della Netscape Communications con il nome di Mochan e successivamente di LiveScript, e poi stato rinominato "JavaScript" ed è stato formalizzato con una sintassi più vicina a quella del linguaggio Java

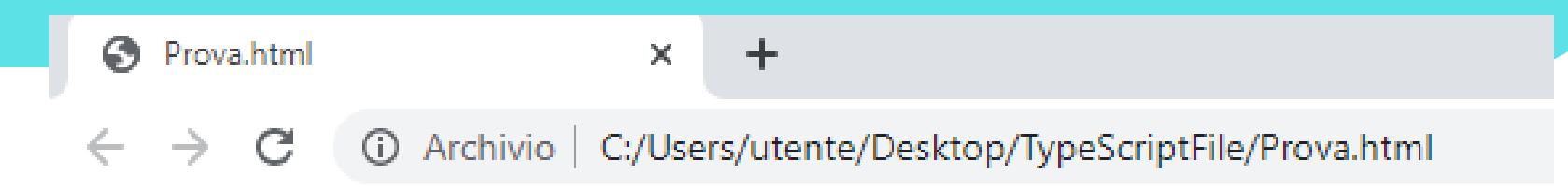
Un piccolo esempio di JS

Naturalmente useremo anche HTML in combinazione con JS

```
<!DOCTYPE html>
<html><body>
<h2>La nostra prima app</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()"
Clicca per vedere ora e data.</button>

<p id="demo"></p>
</body> </html>
```



Quello che vedremo accadere dev'essere
qualcosa di simile a questo



La nostra prima app

[Clicca per vedere ora e data.](#)

Mon Jan 09 2023 12:37:18 GMT+0100 (Ora standard dell'Europa centrale)

Cos'è AngularJS? e perchè partiamo da lui

AngularJS è un framework per applicazioni web open source, sviluppato da Google e dalla comunità di sviluppatori individuali, al fine di migliorare la quality of life dello sviluppo di applicazioni su singola pagina.

Per semplificare lo sviluppo e i test di queste applicazioni fornendo un framework lato client con architettura MVC (Model View Controller) e Model-view-viewmodel (MVVM) insieme a componenti usate nelle applicazioni RIA(Rich Internet application).

Il framework lavora leggendo prima la pagina HTML, che ha encapsulati degli attributi personalizzati addizionali, interagendo con questi attributi come delle direttive per legare le parti di ingresso e uscita della pagina al modello che è rappresentato da variabili standard JavaScript. Il valore di queste variabili può essere impostato manualmente nel codice o recuperato da risorse JSON

Cos'è Angular?

Angular è un framework open source per lo sviluppo di applicazioni web con licenza MIT, evoluzione del già citato AngularJS.

Sviluppato principalmente da Google, la sua prima release è avvenuta il 14 settembre 2016.

Perchè Angular?

Angular è stato completamente riscritto rispetto a AngularJS e le due versioni non sono compatibili.

Il linguaggio di programmazione usato per AngularJS è JavaScript mentre quello di Angular è TypeScript.

Recap post PRIMA lezione:

JS: è un linguaggio di programmazione multi paradigma orientato agli eventi

HTML: l'HyperText Markup Language, noto come HTML, è un linguaggio di markup

CSS: CSS ovvero Cascading Style Sheets è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML

AngularJS: è un framework per applicazioni web open source per lo sviluppo semplificato di applicazioni su singola pagina.

Recap Concettuale:

La differenza tra il back-end è il front-end può essere semplificata nel concetti di cosa viene mostrato all'utente e le interazioni che esso può eseguire FRONT e invece il BACK è il cosa, nella pratica, rende FUNZIONANTI le sue interazioni e che conserva dati e strutture.

Abbiamo visto la differenza tra AngularJS E ANGULAR, ovvero, la differenza nella programmazione del framework, le prestazioni e l'incompatibilità tra i due

Caratteristiche di Angular

Le applicazioni sviluppate in Angular vengono eseguite interamente dal web browser dopo essere state scaricate dal web server (elaborazione lato client). Questo comporta il risparmio di dover spedire indietro la pagina web al web-server ogni volta che c'è una richiesta di azione da parte dell'utente. Il codice generato da Angular gira su tutti i principali web browser moderni quali ad esempio Chrome, Microsoft Edge, Opera, Firefox, Safari ed altri.

Angular è stato progettato per fornire uno strumento facile per sviluppare app che girano su qualunque piattaforma .

Le applicazioni web in Angular in combinazione con il toolkit open source Bootstrap diventano responsive, ossia il design del sito si adatta in funzione alle dimensioni del dispositivo utilizzato.

È in corso di sviluppo un altro toolkit di design responsivo, Flex Layout, più semplice da usare rispetto a Bootstrap e concepito appositamente per Angular.

Altro toolkit che facilita la progettazione in Angular è Angular Material, una serie di componenti che permette di creare una pagina web molto velocemente: con l'utilizzo combinato di Flex Layout ed Angular Material si possono creare siti e app web responsive basate su Angular.

Tipizzazione cenni tecnici

Debole: *In informatica e programmazione, la tipizzazione dinamica è la politica di tipizzazione, ovvero di assegnazione di tipi alle variabili, in cui il controllo del tipo della variabile è effettuato a runtime piuttosto che in fase di compilazione.*

Forte: *In informatica, e in particolare in programmazione, la tipizzazione forte (strong typing) può essere usata per caratterizzare il tipo di regole che un determinato linguaggio di programmazione impone, a livello sintattico o semantico, circa la tipizzazione dei dati e all'uso dei dati in relazione al loro tipo. L'opposto della tipizzazione forte è la tipizzazione debole.*

L'implementazione della tipizzazione statica cambia molto da linguaggio a linguaggio, ma rimangono costanti alcune caratteristiche. Il programma mantiene in una tabella dei valori tutte le variabili dichiarate dal programmatore, insieme al loro tipo e al valore corrente. Grazie a ciò, un tentativo di assegnamento di un valore di tipo diverso da quello di una variabile causa un errore terminale.

I linguaggi a tipizzazione statica più comuni sono Java e C/C++.

Tipizzazione FORTE cenni tecnici

In un linguaggio fortemente tipizzato, il programmatore è tenuto a specificare il tipo di ogni elemento sintattico che durante l'esecuzione denota un valore (per esempio un valore costante, una variabile o un'espressione), e il linguaggio garantisce che tale valore sia utilizzato in modo coerente con il tipo specificato: per esempio, non è possibile eseguire una somma aritmetica su dati di tipo stringa. Questo concetto generale può applicarsi con diverse sfumature; a seconda del contesto. Dunque, l'espressione linguaggio fortemente tipizzato può riferirsi a cose leggermente diverse:

- **tipizzazione statica:** tutti i controlli sull'uso corretto dei valori rispetto al loro tipo vengono eseguiti durante la compilazione;
- **sicurezza rispetto ai tipi (type safety):** è garantito che vengano fatti controlli esaustivi sull'uso dei valori rispetto al loro tipo, ma non necessariamente durante la compilazione (un programma potrebbe fallire durante l'esecuzione a causa di violazioni di tipo)
- **impossibilità di eseguire conversioni di tipo;**
- **impossibilità di eseguire conversioni di tipo implicite;**

Tipizzazione FORTE cenni tecnici prt 2

Un esempio limite di linguaggio debolmente tipizzato è il linguaggio macchina, in cui un'area di memoria, rappresentata nel codice da un indirizzo di memoria, può essere usata indifferentemente per contenere valori di qualunque tipo, numeri interi, numeri con la virgola, caratteri, e così via. I linguaggi di programmazione ad alto livello tendono ad avere sistemi dei tipi più sicuri e quindi ad avvicinarsi all'ideale della forte tipizzazione, ma in genere rimangono disponibili meccanismi per trattare dati in modo flessibile che si prestano a essere utilizzati per violare il sistema dei tipi. Per esempio, il linguaggio C fornisce almeno tre meccanismi che lo qualificano come linguaggio debolmente tipizzato:

- **le operazioni di casting, che consentono di forzare l'interpretazione di un qualunque valore secondo un qualunque tipo (anche un tipo diverso da quello a cui il valore è stato precedentemente associato);**
 - **i puntatori a void, che godono di conversione di tipo implicita verso qualunque altro tipo puntatore;**
 - **le unioni, che consentono di interpretare una collezione di dati correlati secondo diverse attribuzioni di tipo indipendenti.**
-

Esempi pratici di tipizzazione statica

Java

```
public class EsempioTipizzazione {public static void main( String[] args ) {  
  
    int a; //dichiara la variabile intera a  
    a = 3.5; //ERRORE! Il valore 3.5 è numerico a virgola mobile, la variabile a invece è intera }  
}
```

C

```
int main() {  
    char c; /* dichiara una variabile di tipo carattere */  
    c = "Esempio";  
    /* ERRORE! "Esempio" è una stringa, non può essere assegnata a una variabile carattere */  
}
```

Esempi pratici di tipizzazione debole

Python

x = 12 # int || Integer || NUMERO INTERO

x = "parola alfanumerica" # String || Stringa || può contenere qualsiasi combinazione di lettere e numeri all'interno degli apici --> ""

x = False # Bool || Booleani || può contenere solo True o False

x = 123.23434 # Numero complesso FLOAT può contenere numeri con al virgola fino ad una determinata soglia

Print(x) // andrà in errore?

Esercitazione 1: Html e JS

Scrivere un documento HTML contenente una form contenente i seguenti campi:

- lavoro(casella di testo a scelta multipla)
- nome (casella di testo editabile lunga 30 caratteri) •
- email (casella di testo editabile lunga 30 caratteri)
- bottone di invio e bottone di reset
- una casella di testo per le note

Aggiungere al documento HTML JavaScript che esegua i seguenti controlli:

- verifica che il campo nome e l'email non siano vuoti ne troppo lunghi;
- verifica che la selezione del lavoro sia corretta.

Inoltre, fare si che le funzioni JS vengano eseguite solo quando l'utente invia la form.

Cos'è Type Script?

TypeScript è JavaScript con sintassi aggiunta per i tipi. Possiamo anche definirlo un superset sintattico di JavaScript che aggiunge tipizzazione statica, ciò significa sostanzialmente che TypeScript aggiunge la sintassi a JavaScript, consentendo agli sviluppatori di aggiungere tipi.
Quindi essere un "Superset sintattico" significa che condivide la stessa sintassi di base di JavaScript, ma aggiunge qualcosa ad esso.

JavaScript è un linguaggio vagamente tipizzato e è spesso molto difficile capire quali tipi di dati vengono trasmessi in JavaScript, i parametri e le variabili delle funzioni non hanno alcuna informazione utile alla tipizzazione.

TypeScript consente di specificare i tipi di dati che vengono passati all'interno del codice e ha la capacità di segnalare errori quando i tipi non corrispondono. Ad esempio, TypeScript segnalerà un errore quando si passa una stringa in una funzione che prevede un numero. JavaScript no.

TypeScript utilizza il controllo del tipo in fase di compilazione. Ciò significa che controlla se i tipi specificati corrispondono prima di eseguire il codice, non durante l'esecuzione del codice.

Come si usa TypeScript?

Un modo comune per utilizzare TypeScript consiste nell'utilizzare il compilatore TypeScript ufficiale, che transpila il codice TypeScript in JavaScript.

La sezione successiva mostra come ottenere la configurazione del compilatore per un progetto locale.

Alcuni popolari editor di codice, come Visual Studio Code, hanno il supporto TypeScript integrato e possono mostrare errori mentre scrivi il codice!

Come si usa TypeScript?

Per installare Type Script avremo bisogno del nostro NPM, esegui il seguente comando per installare il compilatore:

npm install typescript --save-dev

Che dovrebbe darti un output simile a:

```
□ Prompt dei comandi
Microsoft Windows [Versione 10.0.19045.2364]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\utente>npm install typescript --save-dev
added 1 package, and audited 4 packages in 13s

  found 0 vulnerabilities
  npm notice
  npm notice New major version of npm available! 8.19.2 -> 9.2.0
  npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.2.0
  npm notice Run npm install -g npm@9.2.0 to update!
  npm notice

C:\Users\utente>
C:\Users\utente>
```

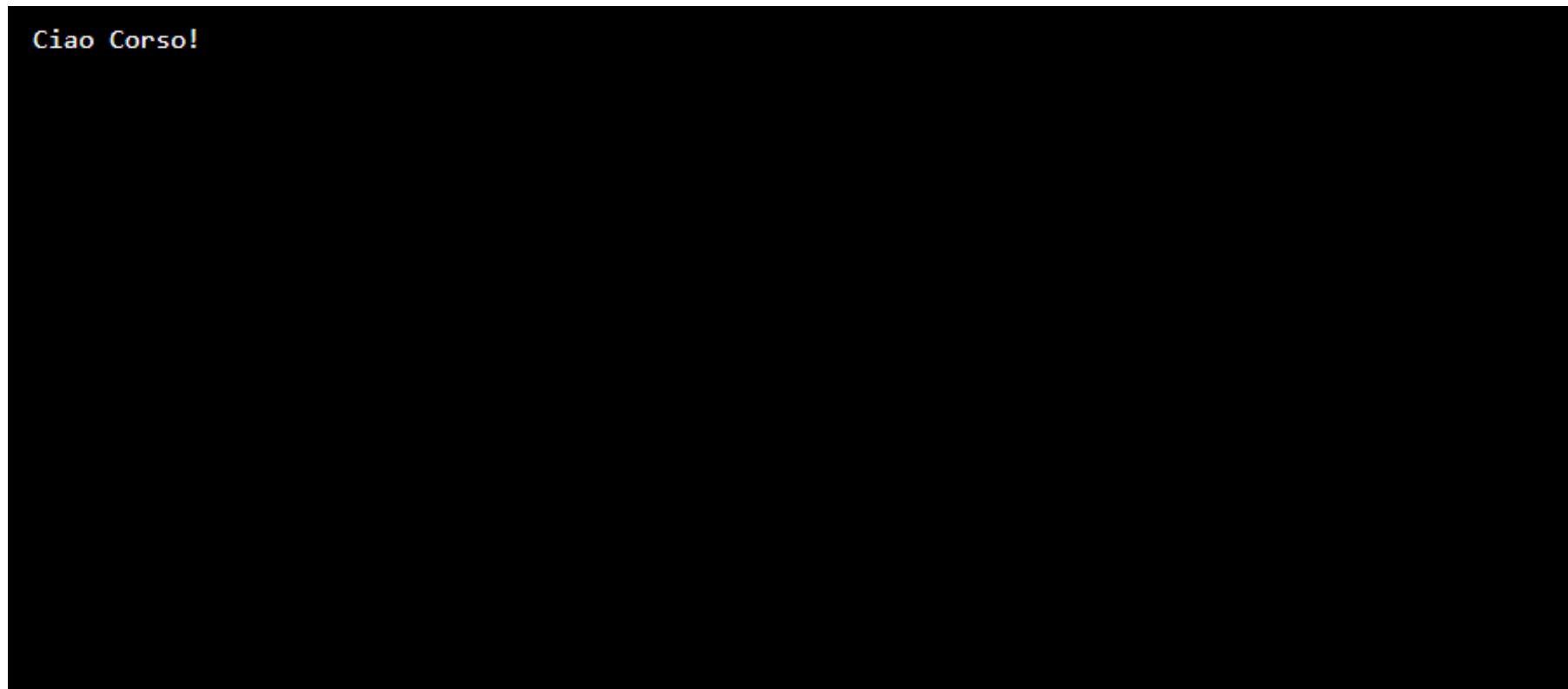
Il primo script in TypeScript

Andiamo ora a vedere il cosiddetto "hello word", ovvero il primo script che faremo girare, utilizzando type-script, faremo stampare una variabile tramite un semplice comando

```
let hellostring : string = "Ciao Corso!"; # tipizzazione esplicita
```

```
let hellostring = "Ciao Corso!";
console.log(hellostring); #tipizzazione vaga o implicita
```

**Dovrebbe darvi un output simile a questo
che vedete nell'immagine**



Ma cos'è NPM?

- **npm è la libreria software (registro) più grande del mondo**
- **npm è anche un software Package Manager e Installer**

Il registro contiene oltre 800.000 pacchetti di codice .

Il nome completo è Node Package Manager.

Gli sviluppatori open source usano npm per condividere il software.

Molte organizzazioni usano anche npm per gestire lo sviluppo privato.

L'uso di npm è gratuito

Puoi scaricare tutti i pacchetti software pubblici di npm senza alcuna registrazione o accesso proprio per via della sua vocazione open source.

Ma cos'è NPM?

npm è installato con Node.js

Ciò significa che dovremo installare Node.js per installare npm sul tuo computer.

Scarica Node.js dal sito web ufficiale di Node.js: <https://nodejs.org>

Il nome npm (Node Package Manager) deriva da quando npm è stato creato per la prima volta come gestore di pacchetti per Node.js.

Tutti i pacchetti npm sono definiti in file denominati package.json .

Il contenuto di package.json deve essere scritto in JSON .

Nel file di definizione devono essere presenti almeno due campi: name e version .

Cos'è Node.js?

Node.js è un runtime system open source multipiattaforma orientato agli eventi per l'esecuzione di codice JavaScript, costruito sul motore JavaScript V8 di Google Chrome.

Molti dei suoi moduli base sono scritti in JS, e gli sviluppatori possono scrivere nuovi moduli nello stesso linguaggio di programmazione. In origine, JS veniva utilizzato soprattutto lato client. In questo scenario gli script JavaScript, generalmente incorporati all'interno dell'HTML di una pagina web, vengono interpretati da un motore di esecuzione direttamente all'interno di un web browser.

Node.js consente invece di utilizzare JS anche per scrivere codice da eseguire lato server, ad esempio per la produzione del contenuto delle pagine web dinamiche, prima che la pagina venga inviata al browser dell'utente. Node.js in questo modo permette di implementare il paradigma "JavaScript everywhere", unificando lo sviluppo di applicazioni Web intorno a un unico linguaggio.

Node.js ha un'architettura orientata agli eventi che rende possibile l'I/O asincrono.

Questo design punta ad ottimizzare il throughput e la scalabilità nelle applicazioni web con molte operazioni di input/output, è inoltre ottimo per applicazioni web sistema real-time (ad esempio programmi di comunicazione in tempo reale o browser game).

Come controllare node.js

Per controllare se Node.js è installato, digitare node -v nel terminale.

**Se è tutto ok dovrebbe darvi
un output simile a questo che
vedete nell'immagine**

□ Prompt dei comandi

```
Microsoft Windows [Versione 10.0.19045.2364]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\utente>node -v
v18.12.1

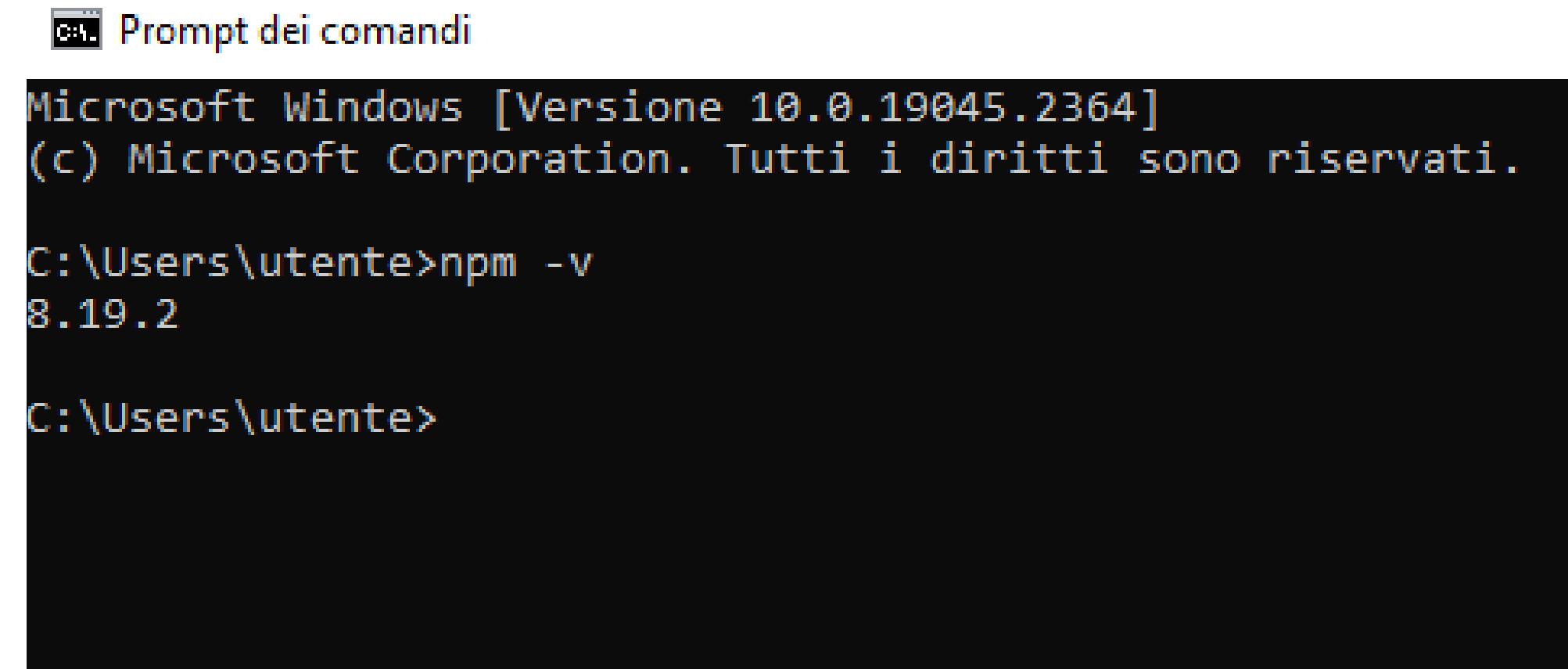
C:\Users\utente>
```

Come controllare NPM

Per controllare se NPM è installato, digitare npm -v nel terminale.

Se NPM è installato correttamente dovreste vedere il numero della versione

**Se è tutto ok dovrebbe darvi
un output simile a questo che
vedete nell'immagine**



A screenshot of a Windows command prompt window titled "Prompt dei comandi". The window shows the following text:
Microsoft Windows [Versione 10.0.19045.2364]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\utente>npm -v
8.19.2

C:\Users\utente>

Hello world in Node.js

Nel seguente esempio "hello world", molte connessioni possono essere gestite contemporaneamente. Ad ogni connessione, la richiamata viene attivata, ma se non c'è lavoro da fare, Node.js andrà in sospensione.

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Cenni tecnici basilari di Node.js

Gli elementi principali in NODE.JS sono i moduli e i nodi

Che cos'è un modulo in Node.js? Considera i moduli come le stesse librerie JavaScript o più tecnicamente potremmo anche dire l'insieme di tutte le funzioni che desideri includere nella tua applicazione.

Oltre tutto i moduli possono essere di diversi tipi

Moduli integrati

Node.js ha una serie di moduli integrati che puoi utilizzare senza ulteriori installazioni che consentono a n.js nativamente di funzionare.

https://www.w3schools.com/nodejs/ref_modules.asp

Cenni tecnici basilari di Node.js prt 2

**Per includere un modulo, bisogna utilizzare la KEYWORD require() funzione
che dovrà essere accompagnata con il nome del modulo:**

```
var http = require('http');
```

**Ora la nostra applicazione avrà accesso al modulo HTTP ed è in grado di
creare il server che poi noi andremo ad istanziare:**

```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end('Hello World!');  
}).listen(8080);
```

Cenni tecnici basilari di Node.js prt 3

Possiamo andare anche a creare dei moduli personali e includerli facilmente nelle tue applicazioni; Creiamo un modulo che restituisca la data e l'ora

```
exports.myDateTime = function () {  
    return Date();  
};
```

Utilizziamo la KEYWORD exports per rendere disponibili proprietà e metodi all'esterno del file del modulo.

Salviamo ora il codice sopra un file chiamato "primomodulo.js"

Cenni tecnici basilari di Node.js prt 4

Ora possiamo includere e utilizzare il modulo in uno qualsiasi dei nostri file Node.js. Usiamo il modulo "primomodulo.js" in un file Node:

```
var http = require('http');
var dt = require('./primomodulo');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("La data e l'ora al momento sono: " + dt.myDateTime());
  res.end();
}).listen(8080);
```

Si noti che utilizziamo ./ per individuare il modulo, ciò significa che il modulo si trova nella stessa cartella del file Node.js

Cenni tecnici basilari di Node.js prt 5

Salviamo ora il nostro codice sopra un file chiamato "modulo_demo.js" e avvia il file Avvia demo_module.js:

C:\Users\Your Name> node modulo_demo.js

SIAMO APPENA ANDATI AD ESEGUIRE UN NODO LOGICO TRAMITE I MODULI, O LE SU FUNZIONI, CHE ABBIAMO DECISO DI IMPORTARE AL SUO INTERNO

Esercitazione 2: HTML,js, CSS

HTML: Andare a creare una form con 4 parametri in ingresso dei tipi: Numerico, booleano, string e check type

si può accedere alla form solo dopo un check password che sarà: "AULA Front-end"

due bottoni invio dati e reset

Js: Controllo che tutti i campi siano correttamente tipizzati e che abbiano compilazione adeguata, dopo di che far si che vengano stampati a video sulla pagina con

CSS: due background color diversi, centrati e con grandezze diverse.

Cenni tecnici basilari di Node.js prt 5

HTTP è il modulo che consente a Node.js di trasferire dati tramite Hyper Text Transfer Protocol (HTTP).

Il modulo HTTP può creare un server HTTP che ascolta le porte del server e restituisce una risposta al client andiamo ora ad utilizzare il createServer() è un metodo che viene usato per creare un server HTTP:

```
var http = require('http');

http.createServer(function (req, res) {
  res.write('Serve c'è'); //Scriviamo quella che sarà la risposta del server
  res.end(); //Specifichiamo che il sistema da qui non avrà altri input client
}).listen(8080);
```

Cenni tecnici basilari di Node.js prt 6

Possiamo anche dividere una stringa di query e per farlo esistono moduli integrati per suddividere facilmente la stringa di query in parti leggibili, come il modulo URL.

```
var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(8080);
```

Cenni tecnici basilari di Node.js prt 7

**Ora andiamo a vedere il modulo FILE SYSTEM il 'suo uso comune passa da:
Leggere i file, Creare file, Aggiornare i file, Cancellare file e Rinominare i file**

```
var fs = require('fs');
```

Il *fs.readFile()* viene utilizzato per leggere i file sul tuo computer, creiamo ora un file HTML ,nella stessa cartella di Node.js:

```
<html>
<body>
<h1>My Header</h1>
<p>My paragraph.</p>
</body>
</html>
```

Cenni tecnici basilari di Node.js prt 7

Creiamo un file Node.js che legge il file HTML

```
var http = require('http');
var fs = require('fs');

http.createServer(function (req, res) {
  fs.readFile('demofile1.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

Cenni tecnici basilari di Node.js prt 8

Ora creiamo un vero e proprio nuovo file usando il metodo fs.appendFile():

```
var fs = require('fs');

fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Il modulo File System dispone di diversi metodi per la creazione di nuovi file:

fs.appendFile() fs.open() fs.writeFile()

Cenni tecnici basilari di Node.js prt 8

Il fs.appendFile() aggiunge il contenuto specificato a un file.

Se il file non esiste, il file verrà creato:

```
var fs = require('fs');

fs.appendFile('mynewfile1.txt', 'ciao a tutti!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Cenni tecnici basilari di Node.js prt 8

Il fs.open() accetta un "flag" come secondo argomento, se il flag è "w" per "scrittura", il file specificato viene aperto per la scrittura.

Se il file non esiste, viene creato un file vuoto:

```
var fs = require('fs');

fs.open('mynewfile2.txt', 'w', function (err, file) {
  if (err) throw err;
  console.log('Saved!');
});
```

Cenni tecnici basilari di Node.js prt 8

Il fs.writeFile() sostituisce il file e il contenuto specificati se esiste.

Se il file non esiste, verrà creato un nuovo file, contenente il contenuto specificato:

```
var fs = require('fs');

fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```

Cenni tecnici basilari di Node.js prt 8

Il modulo File System dispone di due metodi specifici per l'aggiornamento file:

fs.appendFile() fs.writeFile()

Il fs.appendFile() nello specifico aggiunge al file il contenuto specificato

```
var fs = require('fs');

fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {
  if (err) throw err;
  console.log('Updated!');
});
```

Cenni tecnici basilari di Node.js prt 8

**Il modulo File System dispone di un metodo per eliminare un file il fs.unlink()
elimina il file specificato:**

```
var fs = require('fs');

fs.unlink('mynewfile1.txt', function (err) {
  if (err) throw err;
  console.log('File deleted!');
});
```

Cenni tecnici basilari di Node.js prt 8

Il modulo File System per rinominare un file utilizza il fs.rename() che rinomina il file specificato:

```
var fs = require('fs');

fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {
  if (err) throw err;
  console.log('File Renamed!');
});
```

Cenni tecnici basilari di Node.js prt 9

C:\Users\Your Name>npm install upper-case

Una volta installato, il pacchetto è pronto per l'uso.Includi il pacchetto "maiuscolo" nello stesso modo in cui includi qualsiasi altro modulo:

```
var http = require('http');
var uc = require('upper-case');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(uc.upperCase("sono gigante!"));
  res.end();
}).listen(8080);
```

Esercitazione 3: un po di Node.js

Andiamo a creare un sistema CRUD(Create Rename Update Delate) che inserisca e rimuova elementi a vostra scelta, per tipo e valori, usando il modulo FS e facendolo stampare su un file di html Step1.5 css diviso per tipologia di azione

Dovrà farlo solo quando viene richiamato tramite http server

step2: far si che possa venir stampato partendo da un input da bottone e che sempre tramite un pulsante si possa salvare