**Le espressioni sono codice Javascript inserito in doppie parentesi graffe:**

**{{ espressione javascript }}.**

**Foglio base HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

**Per aggiungere Angular JS, si inserisce il seguente link all’interno dell’head:**

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

**La nostra pagina diventerà quindi:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Prima pagina AngularJS</title>

</head>

<body>

</body>

</html>

**Per inserire AngularJS all’interno di un div, si inserisce ng-app nel tag:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Prima pagina AngularJS</title>

</head>

<body>

    <div id="container" ng-app>

    </div>

</body>

</html>

**Accanto ad ng-app si inizializzano le variabili tramite il comando angular ng-init:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Prima pagina AngularJS</title>

</head>

<body>

    <div id="container" ng-app ng-init="quantita=1; prezzo=1">

        Quantita <input type="text" name = "quantity" ng-model="quantita">

        Prezzo <input type="text" name="price" ng-model="prezzo">

    </div>

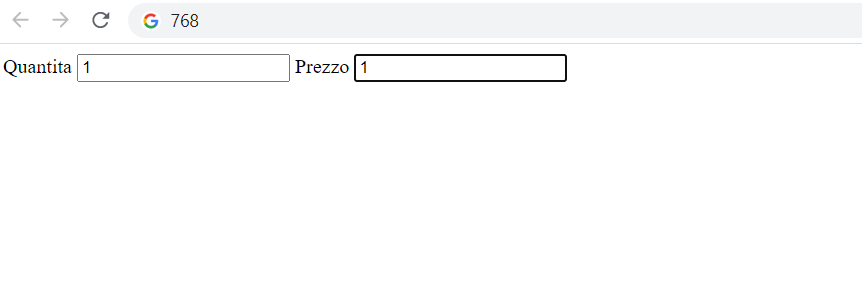
</body>

</html>

**Da quanto si evince dal codice precedente, le variabili js vengono inizializzate tramite il comando ng-init.**

**Esse verranno poi riprese tramite il comando ng-model all’interno degli input.**

**BROSWER**

****

**Consideriamo un altro esempio. Questa volta, utilizziamo i dati inseriti negli input.**

**Supponiamo di voler scrivere il nome completo di una persona.**

**Il codice sarà il seguente:**

**CODICE HTML, JS,ANGULAR:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Prima pagina AngularJS</title>

</head>

<body>

    <div id="container" ng-app ng-init="nome=''; cognome=''">

        Quantita <input type="text" name = "name" ng-model="nome">

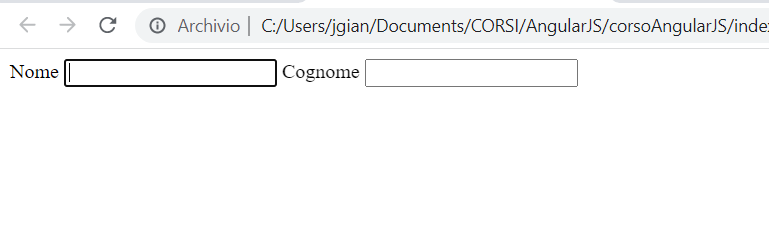
        Prezzo <input type="text" name="surname" ng-model="cognome">

    </div>

</body>

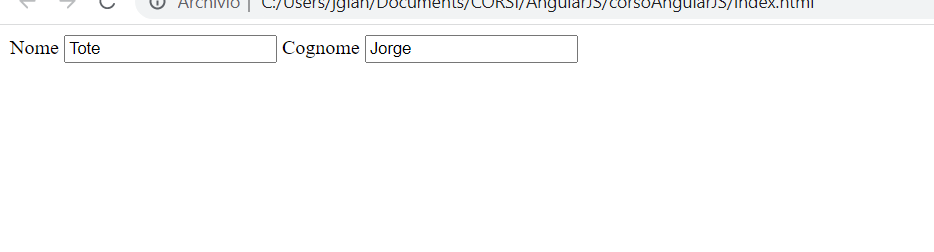
</html>

**BROSWER:**

****

**Negli input Nome e Cognome non è presente alcun testo. Questo perché le due variabili sono inizializzate con stringa vuota.**

**Inserendo il nome e il cognome, vedremo che essi saranno scritti all’interno degli input.**

****

**Vediamo ora come far comparire il testo in un altro punto della pagina web.**

**Supponiamo di inserire un paragrafo sotto al div nel quale sono presenti gli input, e supponiamo di voler far comparire in esso il nome completo (concatenazione del nome e del cognome).**

**Il codice HTML sarà il seguente:**

<body>

    <div id="container" ng-app ng-init="nome=''; cognome=''">

        <div ng-init="nome=''; cognome=''">

            Nome <input type="text" name="name" ng-model="nome">

            Cognome <input type="text" name="surname" ng-model="cognome">

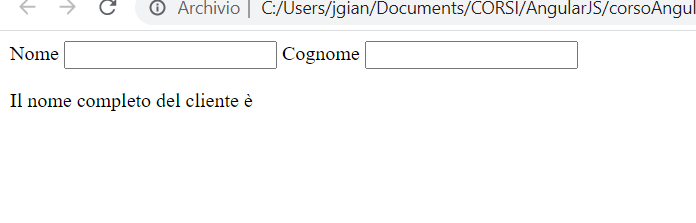
        </div>

        <p>Il nome completo del cliente è {{nome}} {{cognome}}</p>

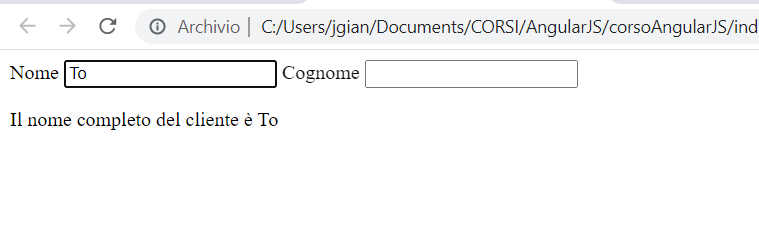
    </div>

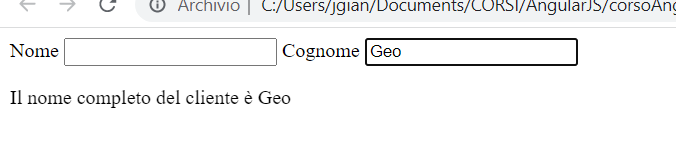
</body>

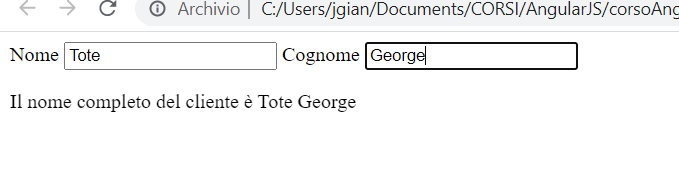
**BROSWER**

****

**Inserendo negli input il nome e il cognome:**

****

****

****

**Proviamo ora ad inserire un altro div all’interno del container (nel quale, come sappiamo, possiamo scrivere in angular tramite l’attributo ng-init).**

**Creiamo ora una semplice calcolatrice:**

    <div id="container"  ng-app ng-init="nome=''; cognome=''">

        <div ng-init="nome=''; cognome=''">

            Nome <input type="text" name="name" ng-model="nome">

            Cognome <input type="text" name="surname" ng-model="cognome">

        </div>

        <p>Il nome completo del cliente è {{nome}} {{cognome}}</p>

        <div id="calcolatrice" style= "margin-top:100px;"  ng-init="numero1=1 ;numero2=1;numero3=1">

            Inserisci qui il primo numero: <input type="number" name="nr1" ng-model="numero1"><br><br>

            Inserisci qui il secondo numero: <input type="number" name="nr2" ng-model="numero2"><br><br>

            Inserisci qui il terzo numero: <input type="number" name="nr3" ng-model="numero3"><br><br>

            <p>La somma tra i tre numeri inseriti è: {{numero1 + numero2 + numero3}}</p>

            <p>La differenza tra i tre numeri inseriti è: {{numero1 - numero2 - numero3}} </p>

            <p>Il prodotto tra i tre numeri inseriti è: {{numero1 \* numero2 \* numero3}} </p>

            <p>Il rapporto tra i tre numeri inseriti è: {{numero1 / numero2 / numero3}} </p>

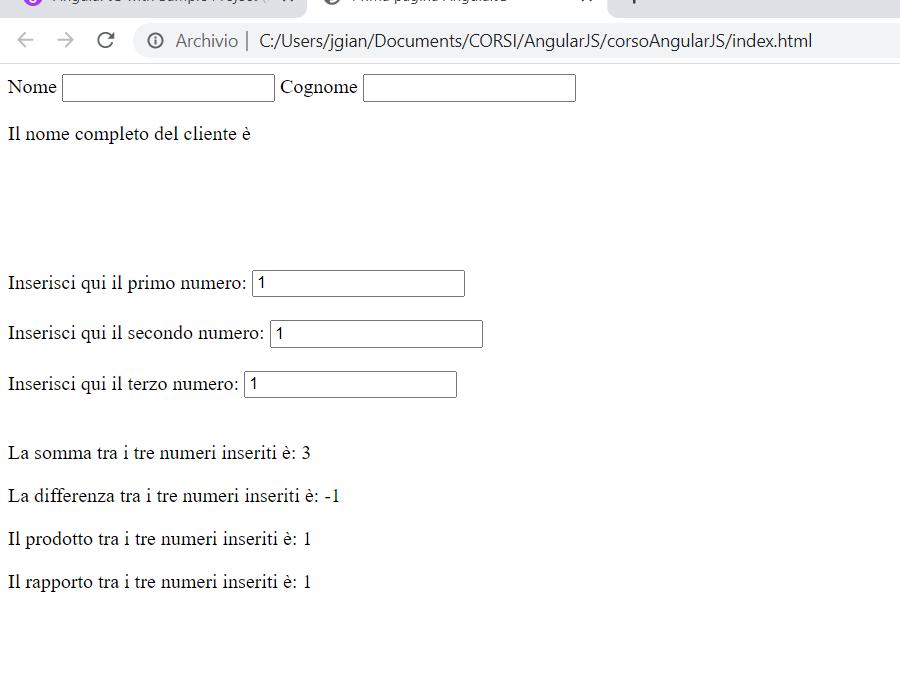
        </div>

    </div>

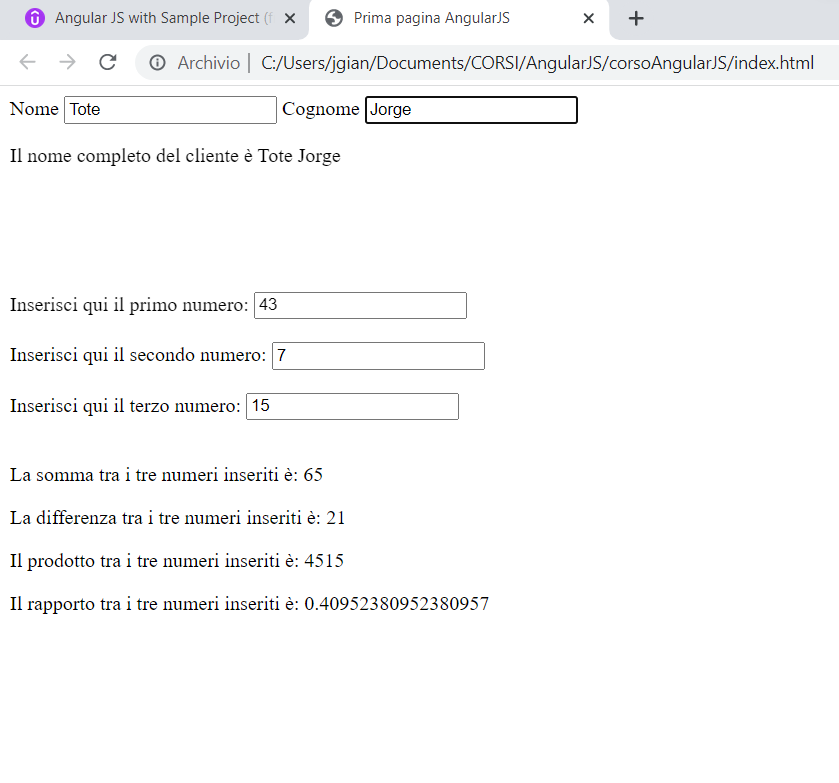
</body>

**BROSWER:**

**Riempiendo i campi:**

****

**Modificando i numeri all’interno degli input:**

****

**Ricapitoliamo.**

* **Nel div container si inserisce la proprietà ng-app, in modo tale che il codice Angular possa essere inserito all’interno del div container**
* **Per immagazzinare i valori degli input e poterli utilizzare altrove, si crea un div all’interno del container e si inizializzano le variabili js utilizzando l’attributo ng-init.**
* **All’interno di ciascun input, si inserisce la proprietà ng-model e si inseriscono al loro interno le variabili create in precedenza**
* **Si richiamano tali variabili in qualsiasi tag desiderato tramite la notazione {{}}.**

**Finora abbiamo inserito variabili. Al loro posto avremmo potuto anche inserire un oggetto.**

**Vediamo un esempio di oggetto js:**

persona1={

    nome:"Tote",

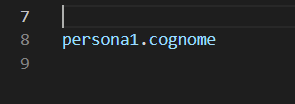
    cognome:"George",

    luogoDiNascita:"Rio"

}

**Per estrarre un elemento da un oggetto, è sufficiente scrivere nomeOggetto.nomeProprietà.**

**Considerando l’esempio precedente:**

****

**Riprendiamo quindi l’esempio della calcolatrice, e al suo interno, inizializziamo un oggetto chiamato numeri:**

**CODICE HTML:**

    <div id="container"  ng-app ng-init="nome=''; cognome=''">

        <div id="calcolatrice"  ng-init="numeri={numero1:1,numero2:1,numero3:1}">

            Inserisci qui il primo numero: <input type="number" name="nr1" ng-model="numeri.numero1"><br><br>

            Inserisci qui il secondo numero: <input type="number" name="nr2" ng-model="numeri.numero2"><br><br>

            Inserisci qui il terzo numero: <input type="number" name="nr3" ng-model="numeri.numero3"><br><br>

            <p>La somma tra i tre numeri inseriti è: {{numeri.numero1 + numeri.numero2 + numeri.numero3}}</p>

            <p>La differenza tra i tre numeri inseriti è: {{numeri.numero1  - numeri.numero2  - numeri.numero3 }} </p>

            <p>Il prodotto tra i tre numeri inseriti è: {{numeri.numero1 \* numeri.numero2  \* numeri.numero3 }} </p>

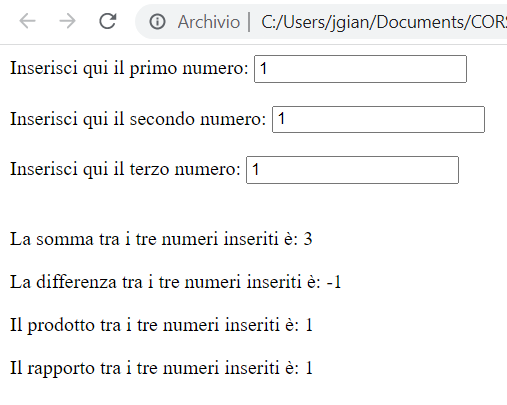
            <p>Il rapporto tra i tre numeri inseriti è: {{numeri.numero1  / numeri.numero2  / numeri.numero3 }} </p>

        </div>

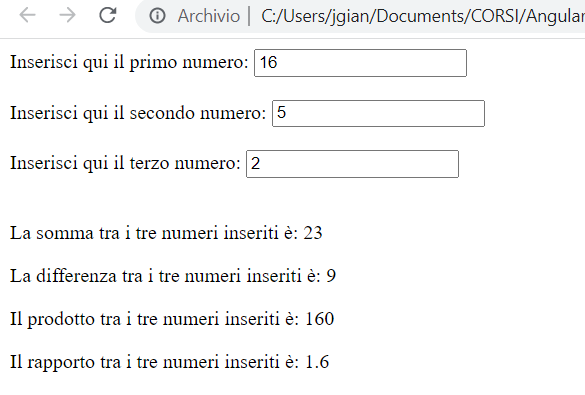
    </div>

</body>

**BROSWER:**

****

**E modificando i numeri all’interno degli input:**

****

**Abbiamo finora lavorato con variabili e con oggetti. Lavoriamo ora con gli array:**

**CODICE HTML:**

<body>

    <div id="container"  ng-app ng-init="nome=''; cognome=''">

        <div id="calcolatrice"  ng-init="numeri=[5,4,6]">

            Inserisci qui il primo numero: <input type="number" name="nr1" ng-model="numeri[0]"><br><br>

            Inserisci qui il secondo numero: <input type="number" name="nr2" ng-model="numeri[1]"><br><br>

            Inserisci qui il terzo numero: <input type="number" name="nr3" ng-model="numeri[2]"><br><br>

            <p>La somma tra i tre numeri inseriti è: {{numeri[0] + numeri[1] + numeri[2]}}</p>

            <p>La differenza tra i tre numeri inseriti è: {{numeri[0]  - numeri[1]  - numeri[2] }} </p>

            <p>Il prodotto tra i tre numeri inseriti è: {{numeri[0] \* numeri[1]  \* numeri[2] }} </p>

            <p>Il rapporto tra i tre numeri inseriti è: {{numeri[0]  / numeri[1] /numeri[2]}} </p>

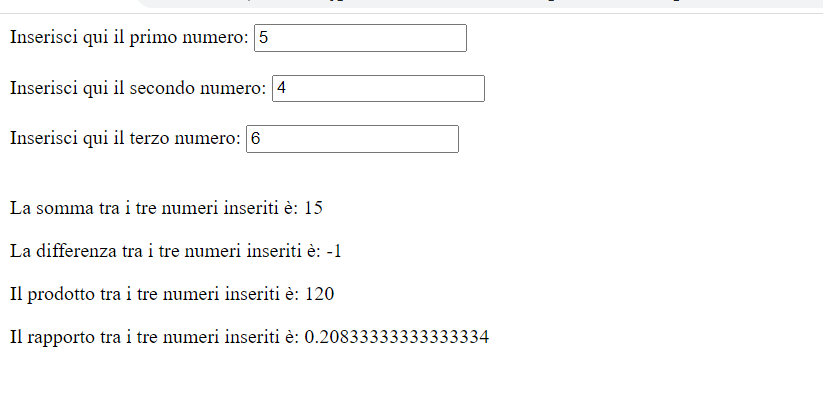
        </div>

    </div>

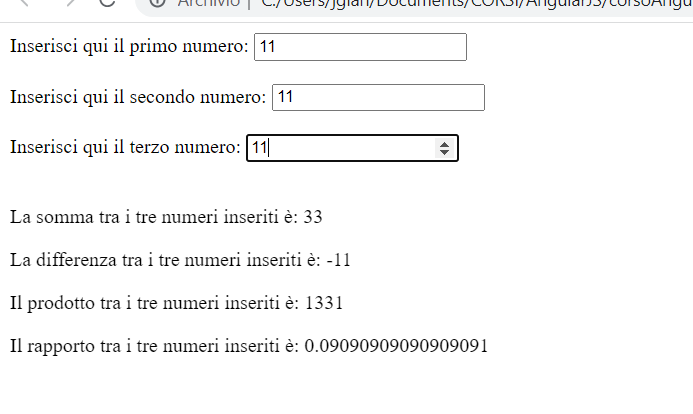
    <script src="prova.js"></script>

</body>

**BROSWER**

****

**Modificando i numeri negli input:**

****

**Studiamo ora un array di oggetti. Consideriamo i seguenti oggetti js:**

persona1={

    nome:'Tote',

    cognome:'George',

    luogoDiNascita:'Rio'

}

persona2={

    nome:'Amilcare',

    cognome:'Genovesi',

    luogoDiNascita:'Torino'

}

Persona3={

    nome:'Guendalina',

    cognome:'Panini',

    luogoDiNascita:'Brescia'

}

**Inseriamo quindi un array contenente questi oggetti nell’ng-init in un div. Per farlo, dobbiamo togliere i nomi degli oggetti e dare un nome all’array. Nel nostro caso l’array si chiamerà persone:**

persone= [{    nome:'Tote',

    cognome:'George',

    luogoDiNascita:'Rio'

},

{

    nome:'Amilcare',

    cognome:'Genovesi',

    luogoDiNascita:'Torino'

},

{

    nome:'Guendalina',

    cognome:'Panini',

    luogoDiNascita:'Brescia'

}

]

**Dato che lavoriamo con un array, per accedere agli elementi si utilizza il relativo indice. Questo l’abbiamo visto anche in precedenza, nel caso di un array di numeri.**

**Il primo oggetto si trova in posizione 0, il secondo in posizione 1, il terzo in posizione 2…. L’ultimo in posizione n-1.**

**Supponiamo quindi di voler estrarre il nome e il cognome del primo, del secondo e del terzo oggetto dell’array, e supponiamo di voler descrivere la loro mansione al lavoro.**

**Per farlo, utilizziamo il seguente codice HTML:**

<body>

    <div id="container" ng-app>

        <div id="people" ng-init="persone=

    [{

        nome:'Tote',

        cognome:'George',

        luogoDiNascita:'Rio'

    },

    {

        nome:'Amilcare',

        cognome:'Genovesi',

        luogoDiNascita:'Torino'

    },

    {

        nome:'Guendalina',

        cognome:'Panini',

        luogoDiNascita:'Brescia'

    }

    ]

    ">

           <p> I signori {{persone[0].nome}}  {{persone[0].cognome}} e {{persone[1].nome}}  {{persone[1].cognome}}  hanno raggiunto il livello quadro.<p><br>

            <p>La signora {{persone[2].nome}} {{persone[2].cognome}} ha raggiunto il terzo livello</p>

            <br><br>

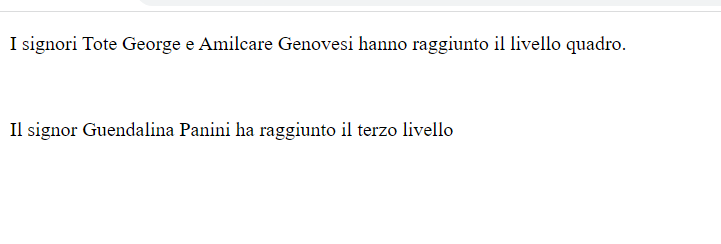
        </div>

    </div>

    <script src="prova.js"></script>

</body>

**Broswer:**

****

**NB. Finora abbiamo inizializzato le variabili. Cosa succede se non le inizializziamo?**

**Le variabili non inizializzate sono considerate null.**

**Vediamo un esempio.**

**CODICE HTML:**

  <div ng-init="nome = 'Tote'">

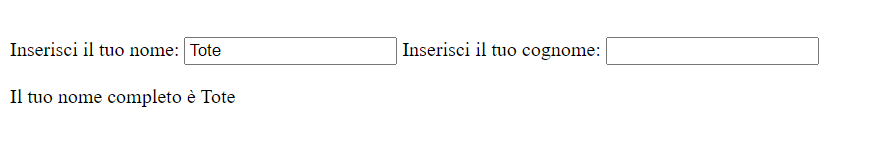
            Inserisci il tuo nome:  <input type="text" name="name" ng-model="nome">

            Inserisci il tuo cognome: <input type="text" name="surname" ng-model="cognome">

            <p>Il tuo nome completo è {{nome}} {{cognome}}</p>

        </div>

**BROSWER**

****

**Il cognome non esiste, a meno che non si inserisca nell’input.**

**In questo caso avremo la concatenazione tra la stringa Tote e null. Il risultato sarà Tote.**

**Se invece lavoriamo con dati di tipo numerico, avremo che le sottrazioni e le addizioni conterranno solo il numero inizializzato. Per quanto riguarda moltiplicazione e divisione avremmo NaN, perché un numero moltiplicato o diviso per null non restituirà un numero.**

**OPERATORE TERNARIO**

**In AngularJS non esistono i control statements. Questo perché la logica si inserisce nel controller attraverso metodi scritti in Javascript. L’unico controllo che esiste nel front-end è l’operatore ternario. Vediamo un esempio.**

**CODICE HTML:**

<body>

    <div id="container" ng-app>

        <div ng-init="salari=[{salario:1500},{salario:800}]">

            Mario ha un salario mensile pari a:  <input type="number" name="salario1" ng-model="salari[0].salario">

            Tote ha un salario mensile pari a:  <input type="number" name="salario2" ng-model="salari[1].salario">

        </div>

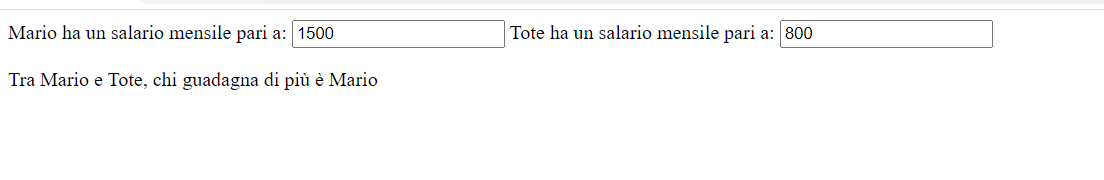
        <p>Tra Mario e Tote, chi guadagna di più è {{salari[0].salario > salari[1].salario ? 'Mario' : 'Tote'}}</p>

    </div>

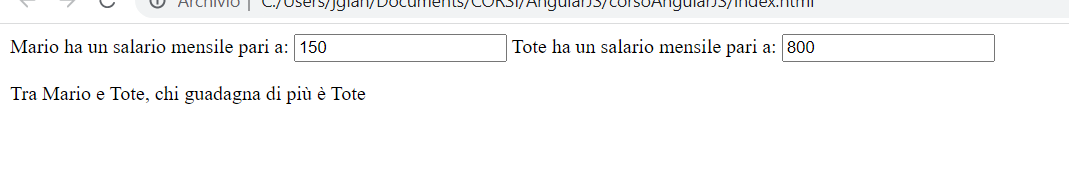
    <script src="prova.js"></script>

</body>

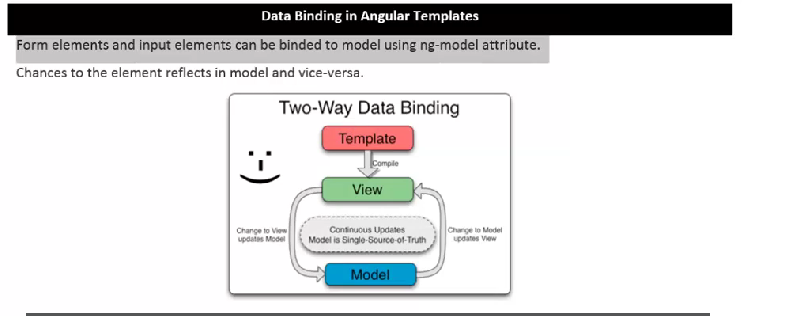
**BROSWER:**

****

**Modificando gli input, se mettiamo a Mario un salario inferiore a quello di Tote, vedremo che il paragrafo sottostante cambierà:**

****

**DATA-BINDING**

****