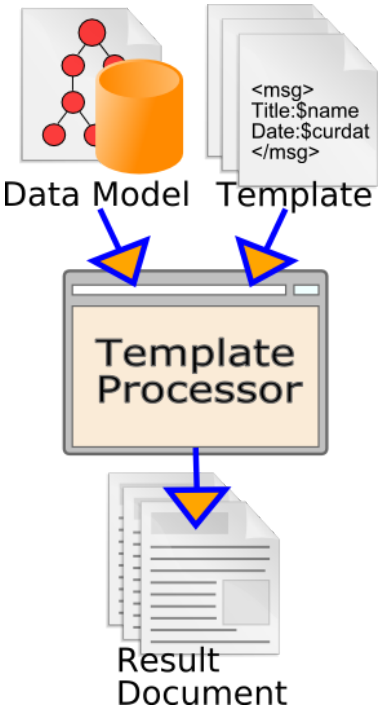
PUG

TEMPLATE ENGINE



Prima di cominciare a usare PUG, dobbiamo chiarire cos'è un template engine (o template processor).
Le pagine web che andremo a costruire saranno costituite dalla fusione di due elementi:

- I dati (presenti tipicamente su un DB)
- La grafica (le nostre pagine html + css)

I template engine servono proprio a mettere insieme questi due elementi in modo semplice e veloce.

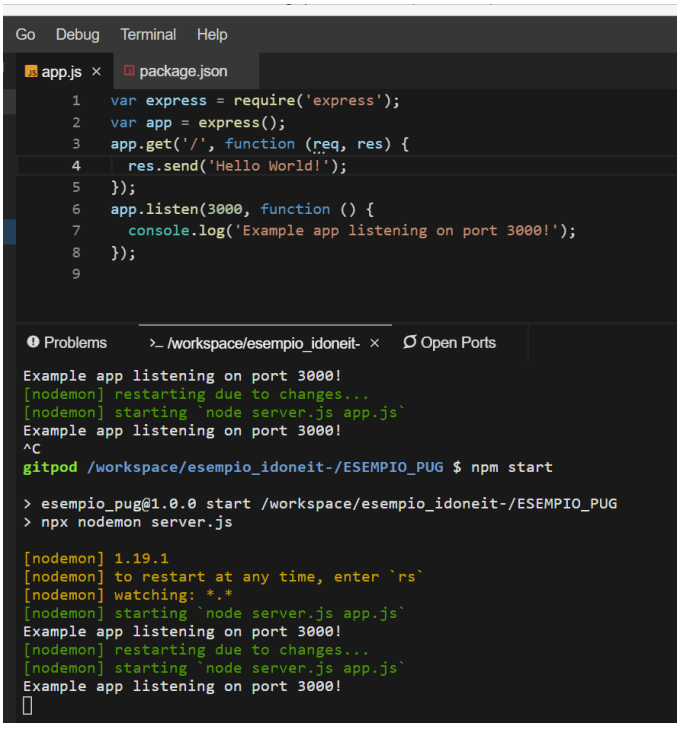
Creare un progetto express che usa PUG come motore di Template

Avvio del progetto

- Crea una cartella chiamata ESEMPIO_PUG
- Crea il file package.json (`npm init`)
- Aggiungi un repository al progetto (`git remote add`) se non è già stato fatto (ad esempio usando gitpod)
- Installa Express (`npm install express --save)`
- Crea il file app.js

```
var express = require('express');
var app = express();
app.get('/', function (req, res) {
  res.send('Hello World!');
});
app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

Verifica che il progetto funzioni.



Installazione di nodemon

Per velocizzare lo sviluppo e il testing, è possibile installare nodemon (riavvia automaticamente il server dopo ogni modifica)

```
npm install nodemon --save
```

Nel file package.js copia il seguente codice evidenziato tra gli script

```
"scripts": {
   ...
   "start": "npx nodemon app.js"
```

```
},
```

Avvia il progetto con

```
npm start
```

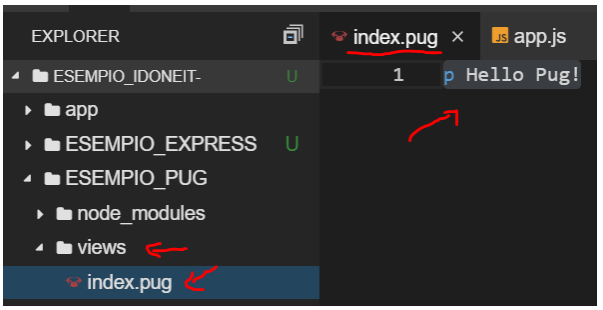
A questo punto non sarà più necessario riavviare il server node ad ogni modifica

INSTALLARE PUG

Digita il comando

```
npm install pug --save
```

- Aggiungi alle cartelle del progetto una cartella chiamata views.
- All'interno della cartella views crea un file chiamato index.pug
- In questo file scrivi: `p Hello Pug!`



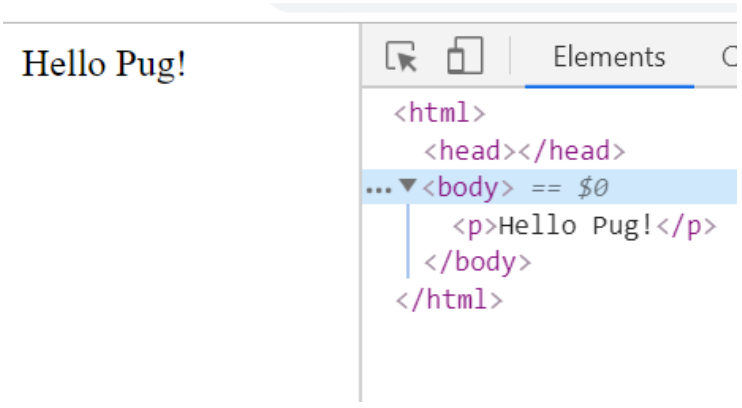
Modifica il codice dell'app.js come segue

```
var express = require('express');
var app = express();

app.set('view engine', 'pug');   //Dico a express di usare pug come motore di template

app.get('/', function (req, res) {
 //res.send('Ciao Mondo');
 res.render('index');  //Dico a express di processare e inviare la pagina index.pug
});
app.listen(3000, function () {
 console.log('Example app listening on port 3000!');
});
```

Avvia il progetto e verifica che funzioni.
Controlla anche il codice HTML creato da PUG con il code inspector del browser



UN SGUARDO RAPIDO A PUG

Pug è insolito rispetto agli altri motori di template, in quanto non usa Tag HTML. Pug ha un approccio piuttosto minimalista, usa nomi di tag, indentazione e un metodo ispirato ai CSS per definire la struttura dell'HTML.

NOTA I template Pug devono essere indentati con spazi, non tab.

Ogni riga del template deve iniziare con il nome di un tag HTML (se ometto il nome del tag, viene inserito un div in automatico).

Il seguente frammento di codice mostra un semplice esempio di un modello di Pug:

```
#banner .page-header
  h1 La mia pagina
  p.lead Benvenuto nella mia pagina
```

Il risultato del codice definito sopra dopo la fase di processing è il seguente:

```
<div id = "banner" class = "page-header">
  <h1> La mia pagina </h1>
  <p class = "lead"> Benvenuto nella mia pagina </p>
</div>
```

Dalle prime righe di input e output è possibile notare che:

- Senza specificare il nome del tag, viene creato un .
- #banner in Pug diventa id = "banner" in HTML.
- .page-header in Pug diventa class = "page-header" in HTML.
- h1 diventa <h1>
- p.lead diventa <p class=”lead”>

Si noti inoltre che il rientro in Pug è importante, in quanto definisce la nidificazione dell’Output HTML. Ricorda che il rientro deve essere fatto con spazi, non tabulazioni!

Una prima pagina PUG

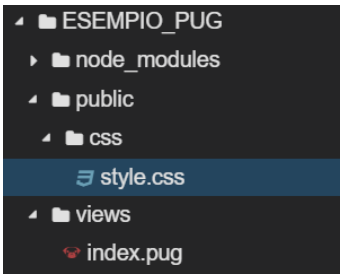
Modifica la pagina index.pug con il seguente codice e verifica di ottenere il risultato riportato sotto

```
doctype html
html
 head
   title Ciao Mondo
   link(rel='stylesheet', href='/css/style.css')
   meta(name="viewport" content="width=device-width, initial-scale=1")
 body
   header
         h1 ciao mondo
   article Questo sito parla del mondo
```

```
<!doctype html>
<html>
▶<head>…</head>
▼<body>
  ▼<header>
     <h1>ciao mondo</h1>
   </header>
   <article>Questo sito parla del mondo</article>
 </body>
•</html> == $0
```

Se osservi la console, troverai un errore perché Node non riesce a trovare il file per il css. Questo per due motivi:

1. Dobbiamo creare il file
2. Dobbiamo dire ad Express dove recuperare i file statici.



Crea una cartella public ed una cartella css. All'interno di questa crea un file chiamato style.css.

Copia questo css nel file style.css

https://github.com/ayoisaiah/node-website-starter-files/blob/master/public/css/style.css

Modifica il codice dell'app.js come segue per dire ad Express dove trovare i file statici

```
var express = require('express');
var app = express();

app.set('view engine', 'pug');   //Dico a express di usare pug come motore di template

app.use(express.static(__dirname + '/public')); // Dico ad express dove recuperare i file statici

app.get('/', function (req, res) {
 //res.send('Ciao Mondo');
 res.render('index');  //Dico a express di processare e inviare la pagina index.pug
});
app.listen(3000, function () {
 console.log('Example app listening on port 3000!');
});
```

Riprova a lanciare il server, ora non dovrebbero esserci più errori.

Aggiungere le classi CSS

Se guardi il file style.css puoi vedere come siano presenti una classe header ed una classe container

```
/* Header
  ============================================================================ */
.header {
  ....
}

/* Homepage Container
  ============================================================================ */
.container {
....
}
```

Aggiungi infine le classi al file index.pug e verifica che venga applicato il css

```
doctype html
html
 head
   title Ciao Mondo
   link(rel='stylesheet', href='/css/style.css')
   meta(name="viewport" content="width=device-width, initial-scale=1")
 body
   header.header
         h1 ciao mondo
   article.container Questo sito parla del mondo
```

Domande a cui rispondere

1. Cos'è nodemon? e come si usa?
2. In quale cartella vanno inserite le viste PUG?
3. Quale estensione hanno i file che contengono le viste PUG?
4. Come si imposta un engine module in Express?
5. Qual è il comando per dire a express di elaborare una specifica pagina PUG e inviarla al client?
6. Cosa succede se in una riga di un template PUG non inserisco il nome di un tag (es: .ciao)

7. Come si specifica un id css in PUG?
8. Come si specifica una classe css in PUG?
9. Come si innesta un tag dentro un altro? (es: ciao)
10. Come si dice a Express quale cartella conterrà tutti i file statici?
11. In cosa viene tradotto il seguente codice PUG
    a. *article.contenitore saluti al mondo*