

Tutorial de PUG

Andres Iporre Medrano
Jose Manuel Vargas Cruz

Pug es un motor de plantilla de Node.js. con el que seremos capaces de escribir código HTML de una sintaxis mucho más sencilla, clara y directa, tanto a la hora de escribir como de leer y modificar.

Instalación de PUG, antes de la instalación necesitamos de NODE.JS, después de la instalación de node, con el comando `npm i -g pug-cli`, instalaremos PUG

```
C:\Users\Andres Eloy Iporre>npm i -g pug-cli
npm WARN deprecated core-js@2.6.12: core-js@<3 is no longer maintained and not recommended for usage due to the number of
f issues. Please, upgrade your dependencies to the actual version of core-js@3.
C:\Users\Andres Eloy Iporre\AppData\Roaming\npm\pug -> C:\Users\Andres Eloy Iporre\AppData\Roaming\npm\node_modules\pug-
cli\index.js

> core-js@2.6.12 postinstall C:\Users\Andres Eloy Iporre\AppData\Roaming\npm\node_modules\pug-cli\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

+ pug-cli@1.0.0-alpha6
added 76 packages from 141 contributors in 38.976s
```

Escribimos el código para un formulario de registro en index.pug

```
doctype html
html(lang='en')
  head
    meta(charset='UTF-8')
    meta(name='viewport' content='width=device-width, initial-scale=1.0')

    title Formulario HTML
  body
    h1#titulo Formulario de Registro
    form(name='form1')
      label(for='nombres') Nombres:
      br
      input#nombres.controls(type='text' placeholder='Ingresa su nombre' required='')
      br
      label(for='apellidos') Apellidos:
      br
      input#apellidos.controls(type='text' placeholder='Ingresa sus apellidos' required='')
      br
      label(for='ci') Carnet de Identidad:
      br
```

```
input#ci.controls(type='number' placeholder='Ingrese su ci' required='')
br
label(for='mail') Email:
br
input#mail.controls(type='email' placeholder='Ingrese su Email' required='')
br
label(for='password') Contraseña:
br
input#password.controls(type='password' placeholder='Ingrese su Contraseña'
required='')
br
label Sexo:
br
input(type='radio' name='sexo')
| Masculino
br
input(type='radio' name='sexo')
| Femenino
br
label Lenguajes de Programación favoritos:
br
input(type='checkbox')
| Java
br
input(type='checkbox')
| Python
br
input(type='checkbox')
| C++
br
input(type='checkbox')
| C
br
input(type='checkbox')
| Haskell
br
input(type='checkbox')
| Pascal
br
label Pais:
br
select
  option Argentina
  option Bolivia
  option Chile
  option Uruguay
br
label Color Favorito
```

```
br
input(type='color')
br
label Fecha de Nacimiento:
br
input(type='date')
br
label Hora actual
br
input(type='time')
br
label Seleccione un archivo
br
input(type='file')
br
label Limpiar Registro
br
input(type='reset')
br
label Escribe una breve descripción sobre ti:
br
textarea(cols='30' rows='10')
br
br
input.type='submit' value='REGISTRAR')
```

Con el comando `pug -w . -o ./html -P` traducimos el archivo `index.pug` a `index.html`

```
C:\WINDOWS\system32\cmd.exe - pug -w . -o ./html -P
Microsoft Windows [Versión 10.0.19041.685]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Manuel>cd Documents/pug examples
C:\Users\Manuel\Documents\pug examples>pug -w . -o ./html -P
watching index.pug
rendered C:\Users\Manuel\Documents\pug examples\html\index.html
```

Atributos:

Los atributos de las etiquetas son similares a los de HTML

- **Atributo de Interpolación:**

- `res.render (ruta, variables)` // Busca un archivo pug para representarse en la ruta "ruta", y le pasa "variables"
- `#{variable}` // Interpola "variable" en línea con el código de Jade circundante, después de evaluar "variable"
- `! {variable}` // Interpola "variable" en línea con el código de Jade circundante, sin evaluar "variable".
- `# [elemento]` // Interpola "elemento" dentro de un elemento HTML Pug existente.

La sintaxis de los elementos HTML interpolados es idéntica a la de los elementos HTML normales.

- **Atributos Booleanos:**

Se aceptan valores booleanos (verdadero y falso). Cuando no se especifica ningún valor, se asume verdadero.

Ejemplo:

```
input(type='checkbox' checked)
|
|
input(type='checkbox' checked=true)
```

- **Atributos de Estilo:**

El atributo de estilo puede ser una cadena, como cualquier atributo normal; pero también puede ser un objeto, lo que resulta útil cuando JavaScript genera estilos.

Ejemplo:

```
a(style={color: 'red', background: 'green'})
```

- **Atributo de Clase:**

El atributo de clase puede ser una cadena, como cualquier atributo normal; pero también puede ser una matriz de nombres de clases, que es útil cuando se genera desde JavaScript.

Ejemplo:

```
- var classes = ['foo', 'bar', 'baz']
a(class=classes)
|
```

|

Clase Literal:

Las clases se pueden definir usando una sintaxis `.classname`

Ejemplo:

a.button *en caso de que fuera un div* → **.content**

ID Literal:

ID's se definen usando la sintaxis: **#idName**

&Atributos:

La sintaxis de &atributos se puede utilizar para descomponer un objeto en atributos de un elemento.

Ejemplo:

div#foo(data-bar="foo")&attributes({'data-foo': 'bar'})