

# Pug (Jade)

<https://www.npmjs.com/package/pug>

# Pug



A “high performance template engine” heavily influenced by HAML and implemented with JavaScript for node.js and browsers”.

It is a node package, found on npm, so installation is as simple as

**npm install pug**

```

<html>
  <head>This is the head of your page
    <title>Example HTML page</title>
  </head>
  <body>This is the body of your page.
    <script async="" src="//pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
    <!-- Homepage Leaderboard--><ins style="display:inline-block;width:728px;height:90px"
    <script>(adsbygoogle = window.adsbygoogle || []).push({});</script>
  </body>
</html>

```



No closing tags, instead, based on spacing

```

html
  head
    | This is the head of your page
    title Example HTML page
  body
    | This is the body of your page.
    script(async='', src="//pagead2.googlesyndication.com/pagead/js/adsbygoogle.js")
    // Homepage Leaderboard
    ins.adsbygoogle(style='display:inline-block;width:728px;height:90px', data-ad-client=
    script.
    (adsbygoogle = window.adsbygoogle || []).push({});

```

# Rendered on request

One line can be used to render the file with or without data sent inside of it.

In this case, data can be used directly inside of the pug file.

Can technically be compiled ahead of time and not rendered on demand so performance is not impacted, but I have not noticed poor performance based on rendering.

```
router.get('/myworkouts', function(req, res){
  req.visitor.pageview("/myworkouts").send();
  if(req.isAuthenticated() && req.user.pass != 'T'){
    var workouts;
    queries.get_workout({username: req.user.id}, function(err, result){
      workouts = result;
      var role = req.user.role;
      res.render('myworkouts.pug', { data_w: JSON.stringify(workouts)
    });
  } else {
    res.redirect('/');
  }
});
```

# Dynamic Data

Not only is it easier to write more rapidly, but it also allows for easy, powerful features, such as dynamically rendering based on data.

```
- if (typeof(data) !== 'undefined'){
table.table-bordered
  <thead>
    <tr>
      <th> Date
      <th> Hours Slept
      <th> Illness
      <th> Injury
      <th> Percent Health
      <th> Cycle
      <th> RPE
      <th> Time
      <th> Distance
      <th> Notes
    </th>
  </thead>
  <tbody>
    <tr>
      <td>
        <div>for workout in data</div>
        <div>
          <div>
            //this form holds the workout ID per workout as a hidden value, so if this
            //or edited, we know which workout to update/remove.
            form(name='form1', action='/myworkouts', method='post')
              input(type='hidden', value=workout['wid'], name='wID', required='')
              <div>td.datetag= workout['date']</div>
              <div>td.sleeptag= workout['sleep']</div>
              <div>td.illnesstag= workout['health']</div>
              <div>td.injurytag= workout['injury']</div>
            </div>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
}
```

You can see that Pug also allows iteration through blocks of objects

# Other features (that we haven't used)...

- Conditionals
- Case Statements
- Inheritance
- Filters