



Day three

Wednesday, 21st January

Day three, Wednesday 21st

- HTTP & Sinatra routes & redirects
- Sinatra views & ERB templating

But first... recap!

- Intro to unit testing with RSpec
- The marvelous world of TDD
- Refactoring & test doubles
- Open class

Got it. So now...

Fun
stuff
today!



Day three, Wednesday 21st

- HTTP & Sinatra routes & redirects
- Sinatra views & ERB templating (!)

About HTTP

Tell me stuff

- States for *HyperText Transfer Protocol* (HTTP)
- A protocol is a set of rules that determine how information is transferred and treated
- First proposal in 1989 (>25 years already!) by Sir Tim Berners-Lee



Tell me (more) stuff

- Stateless!
- Content-independent
- Based on communication between nodes

Request & response

- The **client** performs **requests**
- And the **server** answers with **responses**
- That's it

A graphic way



Request ironhack.com



Response with the site content



Verbs

- They indicate the desired action for the request
- Their purpose is semantic, in order for the server to answer properly
- Predetermined, a total of 9 exist

(a list of) verbs

- GET
- POST
- PUT
- DELETE
- HEAD
- TRACE
- OPTIONS
- CONNECT
- PATCH

Status codes

- They indicate how the request went
- They are composed by both a number and a phrase
- Standard, so communication is better and more efficient

(a table of) status codes

- **1xx:** Message received, doing stuff
- **2xx:** Everything done
- **3xx:** Go to this other place
- **4xx:** You fucked up
- **5xx:** I fucked up

A raw example (header)

```
⌘ curl -v example.com
* Rebuilt URL to: example.com/
* Hostname was NOT found in DNS cache
*   Trying 93.184.216.34...
* Connected to example.com (93.184.216.34) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.37.1
> Host: example.com
> Accept: */*
>
< HTTP/1.1 200 OK
< Accept-Ranges: bytes
< Cache-Control: max-age=604800
< Content-Type: text/html
< Date: Tue, 20 Jan 2015 14:52:31 GMT
< Etag: "359670651"
< Expires: Tue, 27 Jan 2015 14:52:31 GMT
< Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
* Server ECS (fll/0761) is not blacklisted
< Server: ECS (fll/0761)
< X-Cache: HIT
< x-ec-custom-error: 1
< Content-Length: 1270
<
```

A raw example (body)

```
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
body {
  background-color: #f0f0f2;
  margin: 0;
  padding: 0;
  font-family: "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
div {
  width: 600px;
  margin: 5em auto;
  padding: 50px;
  background-color: #fff;
  border-radius: 1em;
}
a:link, a:visited {
  color: #38488f;
  text-decoration: none;
}
@media (max-width: 700px) {
  body {
    background-color: #fff;
  }
  div {
    width: auto;
    margin: 0 auto;
    border-radius: 0;
    padding: 1em;
  }
}
```

Quick intro

Web frameworks

- A set of tools for a specific language that abstract common actions related to web...
- ...so you don't have to implement it yourself
- They ease routing, HTTP responses, templating... you name it

Sinatra routing



This is Sinatra

- Named after a New-York-based, Mafia-near, awesome-singer, smiley guy
- It's a minimal web framework: listens through a port, and answers stuff

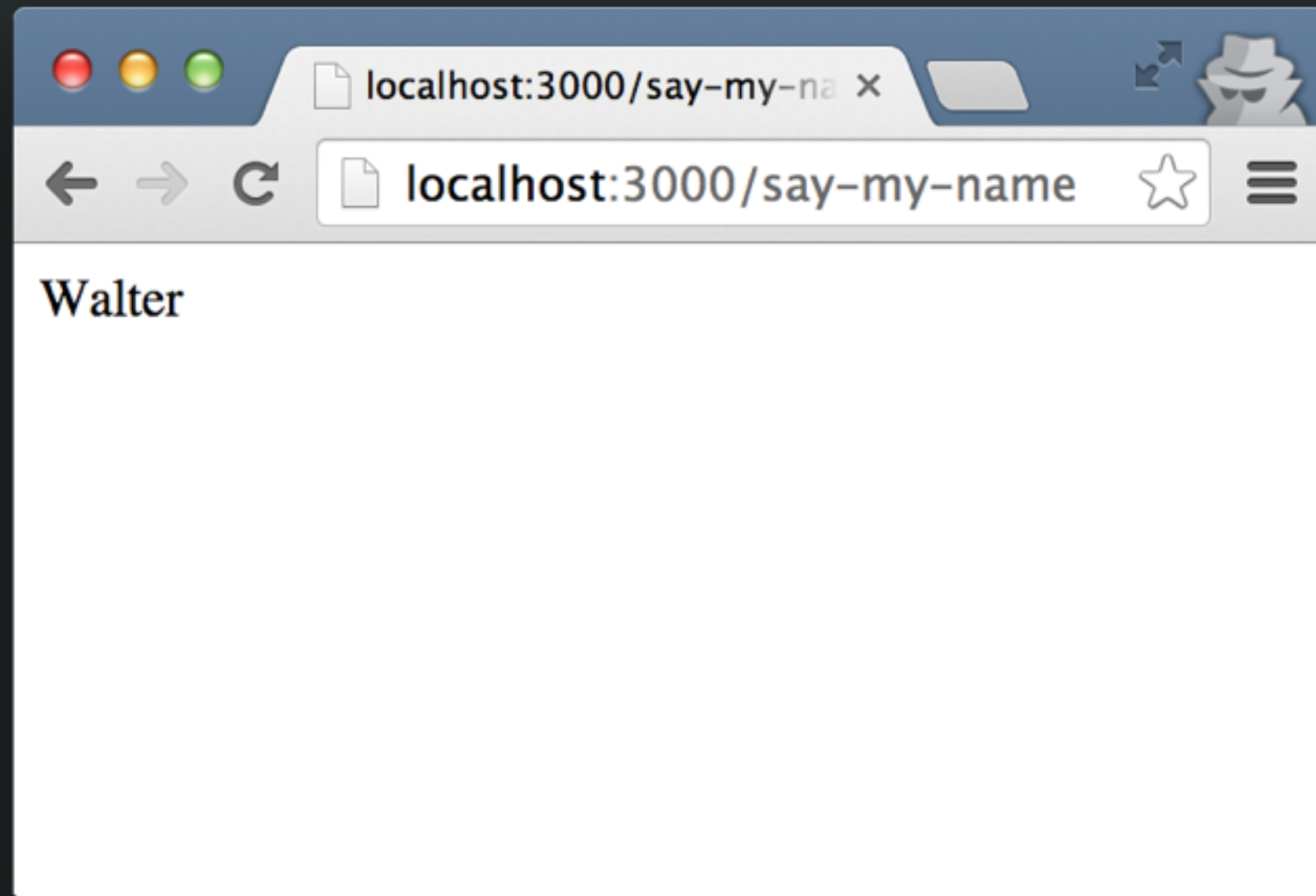
“Say my name”

```
require 'sinatra'

set :port, 3000
set :bind, '0.0.0.0'

get '/say-my-name' do
  'Walter'
end
```

THE NAME IS SAID!!!!



Reload me, baby

gem install sinatra-contrib

```
require 'sinatra'
require 'sinatra/reloader'

set :port, 3000
set :bind, '0.0.0.0'

get '/say-my-name' do
  erb :index
end
```

(please reload the app)

More Sinatras...

sinatrarb.com

github.com/sinatra/sinatra

And a tool: curl

```
brew install curl
```

(Linux has it already)



Not these ones :(

```
🍏 curl localhost:3000/say-my-name  
Walter%
```

more info: `man curl`

LIVE CODING FROM BCN!!!!!!!

Let's say hi from Sinatra

LET'S GET READY TO RUMBLE

Exercises SL6 & SL7

ERB templating

(and dynamic templating in general)

```

1 div.list-wrapper id="#{is_inbound ? 'inbound' : 'outbound'}"
2   - way = is_inbound ? 'inbound' : 'outbound'
3   = render 'shared/book_entries/import_modal', way: way if can?(:create, BookEntry)
4
5   .form-inline.row-fluid
6     = search_form_for @q, :url => target, :html => { :class => "inline input-filer" } do |f|
7     = hidden_field_tag :book_entries_sort, @order_by
8     = hidden_field_tag :book_entries_direction, @direction
9
10    .row-fluid.header.page-header
11      h1.pull-left
12        = t(is_inbound ? 'inbound' : 'outbound', :scope => 'common.invoicing')
13      hr
14    .row-fluid.page_subheader
15      .calendar.span5
16        = render 'shared/book_entries/daterangepicker', is_inbound: is_inbound, target: target
17      .span7#sum_header
18        = render "shared/book_entries/sum", is_inbound: is_inbound
19      = render 'shared/book_entries/graphics', is_inbound: is_inbound
20      = render "shared/book_entries/filters", is_inbound: is_inbound, way: way
21
22    .total_results
23      span=' @total_count
24      = pluralize(@total_count, t('.element')).gsub(/#{@total_count} /, '')
25
26    - if book_entries.any?
27      .legend-filter
28        ul
29          - unless is_inbound
30            li.budget
31              = link_to search_path_for_type(:budgets) do
32                .filter-square
33                  = t('budget', scope: 'common.status')
34          - %w{ payment_pending paid unpaid }.each do |state|
35            li class=state
36              - state_for_translation = state
37              - state_for_translation = 'charged' if !is_inbound && state == 'paid'
38              = link_to search_path_for_status(is_inbound, state) do
39                .filter-square
40                  = t(state_for_translation, scope: 'common.status')
41
42          - if is_inbound && owner.managed?
43            - { verified: 'icon-ok', refused: 'icon-remove' }.each do |status, clazz|
44              li
45                i class=clazz
46                = link_to t(status, scope: 'common.status'), search_path_for_status(true, status)
47
48    .table-list
49      table.table#book-entries-table
50        - if book_entries.any?
51          = render 'shared/book_entries/headers', is_inbound: is_inbound
52          tbody
53            = render :partial => "shared/book_entries/tr", :collection => book_entries, :as => :book_entry,
54              :locals => { :lister => lister, :is_inbound => is_inbound }, :inline => true
55      = paginate @paginated_book_entries

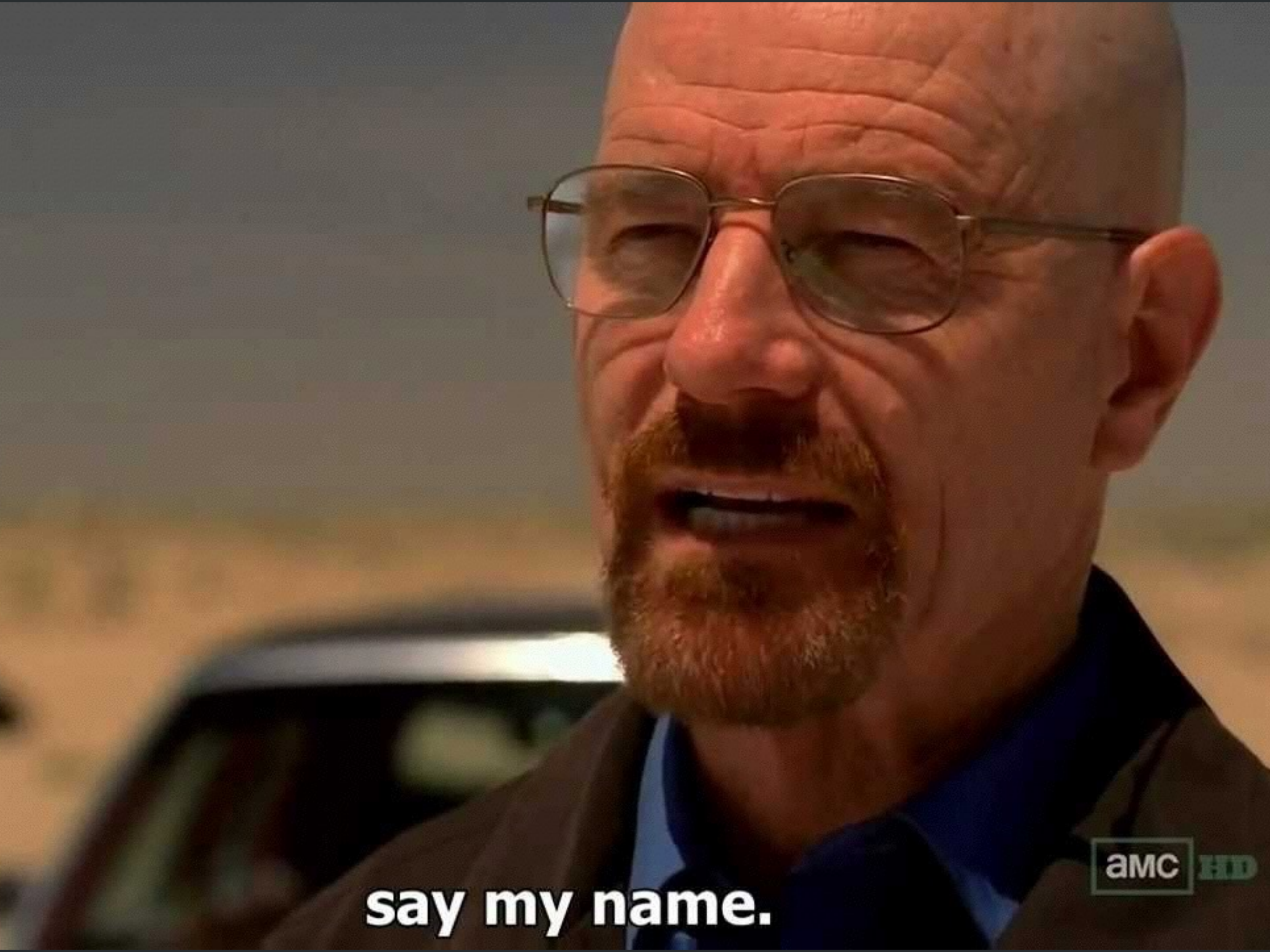
```




Now tell me about it

- A way of creating HTML files dynamically
- Variables, arrays, conditionals... you name it
- Use composition to rock it hard

So let's improve it



say my name.

The .rb file

```
require 'sinatra'
require 'sinatra/reloader'

set :port, 3000
set :bind, '0.0.0.0'

get '/say-my-name' do
  @name = 'Walter'
  erb :index
end
```


And the .erb file

```
<html>  
  <body>  
    I said the name and is...  
    <%= @name %>  
  </body>  
</html>
```

Curl it!

```
❯ curl localhost:3000/say-my-name
<html>
  <body>
    I said the name and is...
    Walter
  </body>
</html>
```




You're goddamn right.

LIVE CODING FROM BCN!!!!!!!

Let's template some stuff

LET'S GET READY TO RUMBLE

Exercises SL8 & SL9