

---

# Ruby Monstas



---

## Session 20

---

# Agenda

---

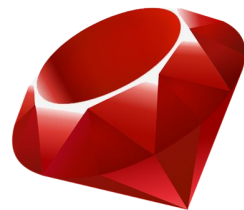
Recap

HTML Forms

HTTP POST

Development Tools

Exercises



---

# Recap

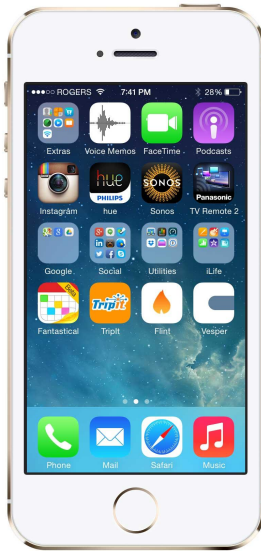
---

---

# HTTP: Computing model evolution

---

Client



Request

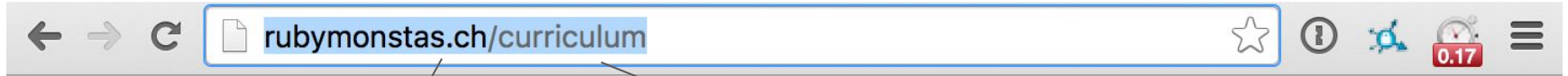
Server



Response

# HTTP: Uniform Resource Locator (URL)

---



DOMAIN NAME

rubymonstas.ch

RESOURCE

/curriculum

---

# HTTP: Method & Format

---



REQUEST METHOD

GET

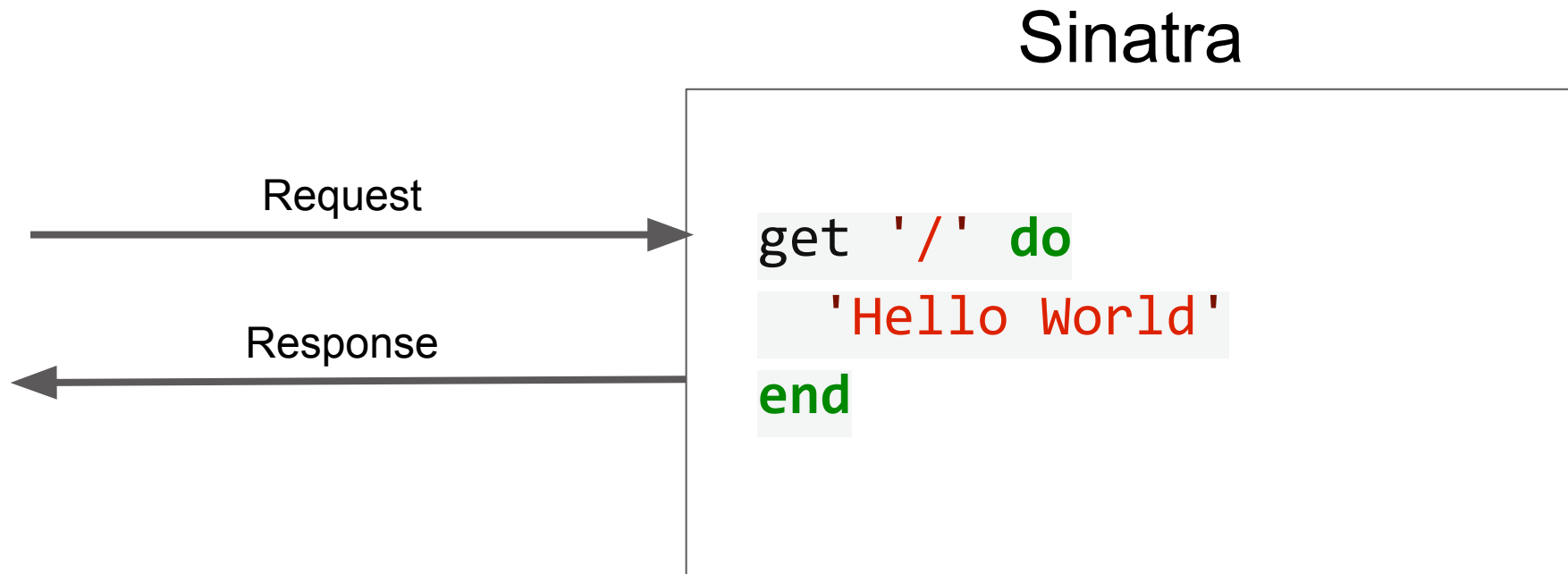
FORMAT

.HTML

---

# Sinatra: Map routes to code

---



---

# HTML Forms

---

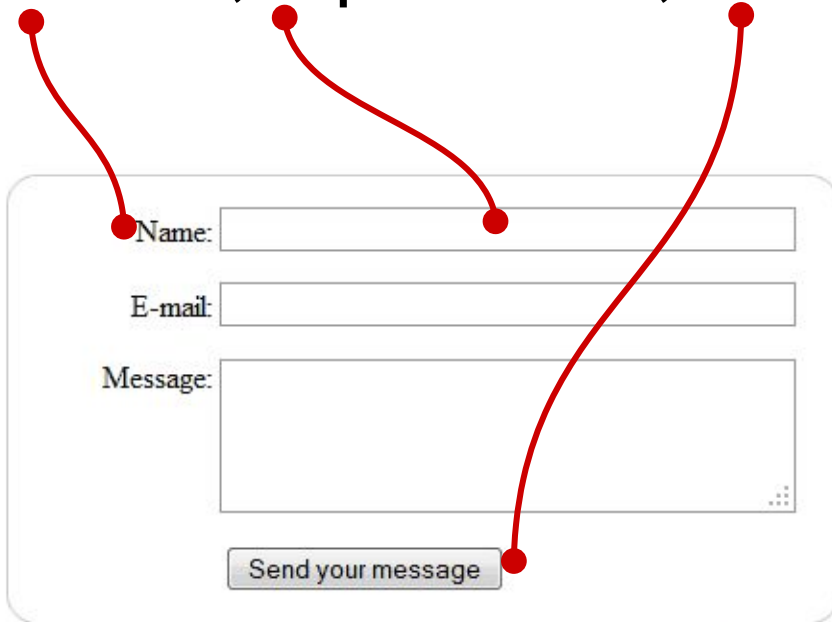
---



# HTML Form

---

labels, input fields, submit button



A screenshot of an HTML form designed for Windows 7. The form is enclosed in a light gray rounded rectangle. It contains three labels: 'Name:', 'E-mail:', and 'Message:'. Each label is followed by an input field. The 'Name:' and 'E-mail:' fields are single-line text boxes, while the 'Message:' field is a larger multi-line text area with a small scroll indicator in the bottom right corner. Below these fields is a 'Send your message' button with a gray gradient and rounded corners. Red dots are placed on the labels and the button, with red curved lines connecting them to the text 'labels, input fields, submit button' above. Specifically, a red dot on 'Name:' connects to 'labels', a red dot on the 'E-mail:' input field connects to 'input fields', and a red dot on the 'Send your message' button connects to 'submit button'.

Windows 7



A screenshot of the same HTML form designed for Mac OSX. The form is enclosed in a light gray rounded rectangle. It contains three labels: 'Name:', 'E-mail:', and 'Message:'. Each label is followed by an input field. The 'Name:' and 'E-mail:' fields are single-line text boxes, while the 'Message:' field is a larger multi-line text area with a small scroll indicator in the bottom right corner. Below these fields is a 'Send your message' button with a gray gradient and rounded corners. This form is identical to the Windows 7 version but lacks the red annotations.

Mac OSX

# HTML Form

---

Input 1 text:

Input 2 checkbox: ☐

Input 3 color:

Input 4 file:  No file selected.

Input 5 email:

Input 6 hidden:

Input 7 number:

Input 8 password:

Input 9 radio: ☐

---

# HTML Form

---

Input 1 text:

Input 4 file:  No file selected.

Input 7 number:

```
<form method="POST">
```

```
<p>
```

```
<label>Input 1 text:
```

```
<input type="text" name="input1" />
```

```
</label>
```

```
</p>
```

```
<p>
```

```
<label>Input 4 file:
```

```
<input type="file" name="input4" />
```

```
</label>
```

```
</p>
```

```
<p>
```

```
<label>Input 7 number:
```

```
<input type="number" name="input7" />
```

```
</label>
```

```
</p>
```

```
</form>
```

---

# HTML Form

---

Input 1 text:

Input 2 checkbox: ☐

Input 3 color:

Input 4 file:  Browse... No file selected.

Input 5 email:

Input 6 hidden:

Input 7 number:

Input 8 password:

Input 9 radio: ☐

```
<input type="XY" name="UNIQUE_NAME" />
```

```
<input type="text" name="some_unique_name1" />
```

```
<input type="checkbox" name="some_unique_name2" />
```

```
<input type="color" name="some_unique_name3" />
```

```
<input type="file" name="some_unique_name4" />
```

```
<input type="email" name="some_unique_name5" />
```

```
<input type="hidden" name="some_unique_name6" />
```

```
<input type="number" name="some_unique_name7" />
```

```
<input type="password" name="some_unique_name8" />
```

```
<input type="radio" name="some_unique_name9" />
```

# HTML Form

---

Let's check this in the browser:

[http://localhost:4567/example\\_form](http://localhost:4567/example_form)

---

---

# HTTP POST

---

---

# HTTP POST

---

We learned from GET so far

There are other verbs, like POST

Let's check this in the browser

---

# Developer Tools (HTTP GET)

The screenshot shows a web browser at `localhost:4567/example_form`. The browser's developer tools are open, with the 'Network' tab selected. A blue box highlights the form on the left, and another blue box highlights the network request in the developer tools. Blue arrows point from the form to the network request and from the network request to the text labels on the right.

Input 1 text:

Input 2 checkbox: ☐

Input 3 color:

Input 4 file:  No file selected.

Input 5 email:

Input 6 hidden:

Input 7 number:

Input 8 password:

Input 9 radio: ☐

Input 10 range:

Input 11 reset:

Input 12 search:

Input 13 tel:

Input 14 time:

Method	File	Domain	Type	Transferred	Size	Time
200 GET	example_form	localhost:4567	html	1.86 KB	1.86 KB	5 ms

- the request in the developer tools
- the page we requested (resource)
- the response rendered by the browser



# Developer Tools (HTTP POST)

The screenshot displays a web browser's developer tools interface. The address bar shows the URL `localhost:4567/example_form`. The left sidebar contains a list of input values: `Value input1:`, `Value input2:`, `Value input3: #000000`, `Value input4:`, `Value input5:`, `Value input6:`, `Value input7:`, `Value input8:`, `Value input9:`, `Value input10: 50`, `Value input11:`, `Value input12:`, `Value input13:`, and `Value input14: asdf`. The right sidebar has tabs for Inspector, Console, Debugger, Style Editor, Performance, Network, and Storage. The Network tab is active, showing a table of network requests. The first request is highlighted with a blue box, indicating a successful `200 POST` to `example_form` on `localhost:4567`. The request details show a status of `200`, a method of `POST`, a file of `example_form`, a domain of `localhost:4567`, a type of `html`, and transferred and size values of `0.78 KB`. The response time is `8 ms`.

✓	Method	File	Domain	Type	Transferred	Size	0 ms	80 ms
200	POST	example_form	localhost:4567	html	0.78 KB	0.78 KB	8 ms	

# Developer Tools (HTTP POST)

The screenshot displays a web browser's developer tools interface. The left pane shows the 'Console' tab with a list of input values: Value input1:, Value input2:, Value input3: #000000, Value input4:, Value input5:, Value input6:, Value input7:, Value input8:, Value input9:, Value input10: 50, Value input11:, Value input12:, Value input13:, and Value input14: asdf. The right pane shows the 'Network' tab with a list of requests. A request to 'example\_form' with a status of 200 and method POST is highlighted. The 'Headers' sub-tab is active, showing the following details:

- Request URL:** http://localhost:4567/example\_form
- Request method:** POST
- Remote address:** [::1]:4567
- Status code:** 200 OK
- Version:** HTTP/1.1

The response headers are expanded, showing:

- Connection:** "Keep-Alive"
- Content-Length:** "798"
- Content-Type:** "text/html; charset=utf-8"
- Date:** "Sun, 01 Nov 2015 21:09:52 GMT"
- Server:** "WEBrick/1.3.1 (Ruby/2.2.2/2015-04-13)"
- X-Frame-Options:** "SAMEORIGIN"
- X-XSS-Protection:** "1; mode=block"
- x-content-type-options:** "nosniff"

The request headers are also expanded, showing:

- Host:** "localhost:4567"
- User-Agent:** "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:43.0)"
- Accept:** "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8"
- Accept-Language:** "en-US,en;q=0.5"
- Accept-Encoding:** "gzip, deflate"
- Referer:** "http://localhost:4567/example\_form"
- Cookie:** "\_ga=GA1.1.1955854004.1434967852; lcl\_sessio...csr=(dir"
- Connection:** "keep-alive"

The request headers from the upload stream are also expanded, showing:

- Content-Type:** "application/x-www-form-urlencoded"
- Content-Length:** "121"

# Developer Tools (HTTP POST)

The screenshot shows a web browser at `localhost:4567/example_form`. The developer tools are open, and the Network tab is selected. A POST request to `example_form` is highlighted, showing a status of 200. The Params sub-tab is active, displaying the following form data:

Parameter	Value
input1	""
input3	"#000000"
input4	""
input15	""
input6	""
input7	""
input8	""
input10	"50"
input12	""
input13	""
input14	"asdf"
submit	"Submit"

On the left side of the image, there are labels for the input values:

- Value input1:
- Value input2:
- Value input3: #000000
- Value input4:
- Value input5:
- Value input6:
- Value input7:
- Value input8:
- Value input9:

# Developer Tools (HTTP POST)

The screenshot displays a web browser at `localhost:4567/example_form` with a form containing three inputs. The developer tools are open, showing the HTML structure of the form and the network request parameters.

**Form Inputs:**

- Input 1 text:
- Input 4 file:  Browse... No file selected.
- Input 7 number:

**HTML Code (Form Structure):**

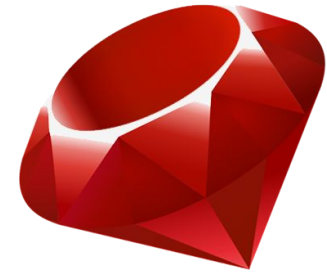
```
<form method="POST">
  <p>
    <label>Input 1 text:
      <input type="text" name="input1" />
    </label>
  </p>
  <p>
    <label>Input 4 file:
      <input type="file" name="input4" />
    </label>
  </p>
  <p>
    <label>Input 7 number:
      <input type="number" name="input7" />
    </label>
  </p>
</form>
```

**Network Request Parameters (Form data):**

Parameter	Value
input1	""
input3	"#000000"
input4	""
input15	""
input6	""
input7	""
input8	""
input10	"50"
input12	""
input13	""
input14	"asdf"
submit	"Submit"

---

# Time to practice



# Resources

---

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

<https://developer.mozilla.org/en/docs/Web/HTML/Element/form>

<https://developer.mozilla.org/en-US/docs/Tools>

<https://developer.mozilla.org/en-US/docs/Web/HTTP>

[https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

---