

RSS Personal Reader

Taste of Code

with Miriam Tocino

 **Codaisseur**

Are You Ready?

Editor

Atom, Notepad++, Vim

Browser

Chrome, Firefox

Slack

<https://kpntasteofcode.slack.com/>

What Are You Building?

RSS Feed Reader

Using HTML, CSS
and JavaScript (jQuery)

- ✦ Display a list of some news sites
- ✦ Display links to the last 5 articles for each of these sites
- ✦ Open articles in a different window when clicked on



Intro to HTML

Content & Structure



What is HTML

HTML is a **markup language**
for writing web pages.

Any website you ever visited
was **made with html**.

Each new version of HTML is
adapted to new requirements of the web.

What is HTML

index.html

<DOCTYPE>

defines doc type

<html>

describes HTML doc

<head>

provides doc info

<title>

provides doc title

<body>

describes page content

<h1>

describes a heading

<p>

describes a paragraph

```
<!DOCTYPE html>
<html>
  <head>
    <title>Taste of Code</title>
  </head>
  <body>
    <h1>Agenda</h1>

    <p>On the menu today:</p>

    <ul>
      <li>Structure with HTML</li>
      <li>Styling with CSS</li>
      <li>Happiness with Lunch</li>
    </ul>
  </body>
</html>
```

What is HTML

A stylized illustration of a web browser window. The window has a dark gray header bar with three colored window control buttons (red, yellow, green) on the left, a search bar in the center, and a magnifying glass and star icon on the right. The main content area is white and contains the following text:

Agenda

On the menu today:

- Structure with HTML
- Styling with CSS
- Happiness with Lunch

On the menu today:

- Structure with HTML
- Styling with CSS
- Happiness with Lunch

Looking closer at tags

Most HTML elements **open** and **close**.

<h1>

Taste of Code

</h1>



Looking closer at tags

HTML elements can have **attributes**.



<p lang="nl"> Text in Dutch. **</p>**



Looking closer at tags

Some HTML elements are **self-closing**.

They don't need a closing tag.

```

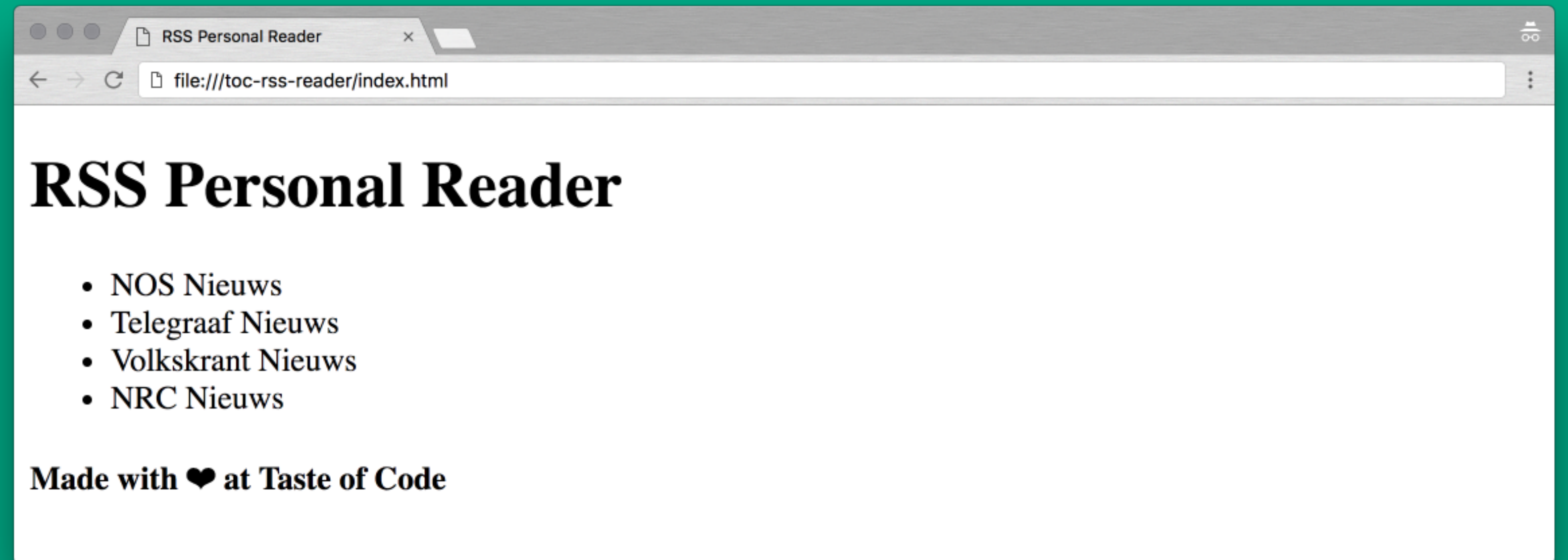
```

└─┬─┐ ATTRIBUTE
NAME
└─┬─┐
└─┬─┐ ATTRIBUTE
VALUE
└─┬─┐

└────────────────────────────────┬────────────────────────────────┐
SELF-CLOSING TAG

Intro to HTML

Exercises



[Link to Github](#)



Intro to CSS

Style & Presentation



What is CSS

CSS is a **stylesheet language** that describes the presentation of an HTML document.

- ✦ **text color**
- ✦ **fonts style**
- ✦ **background images**
- ✦ **spacing between elements**

Your First CSS

index.html

color

sets the color of the text

```
<!DOCTYPE html>
<html>
  <head>
    <title>Taste of Code</title>

    <style>
      h1 {
        color: red;
      }
    </style>

  </head>
  <body>
    <h1>Agenda</h1>
    ...
  </body>
</html>
```

Your First CSS

Agenda

On the menu today:

- Structure with HTML
- Styling with CSS
- Happiness with Lunch

Looking closer at CSS rules

CSS associates **style rules** with HTML tgs.

SELECTOR

p {

font-family: Helvetica;

}

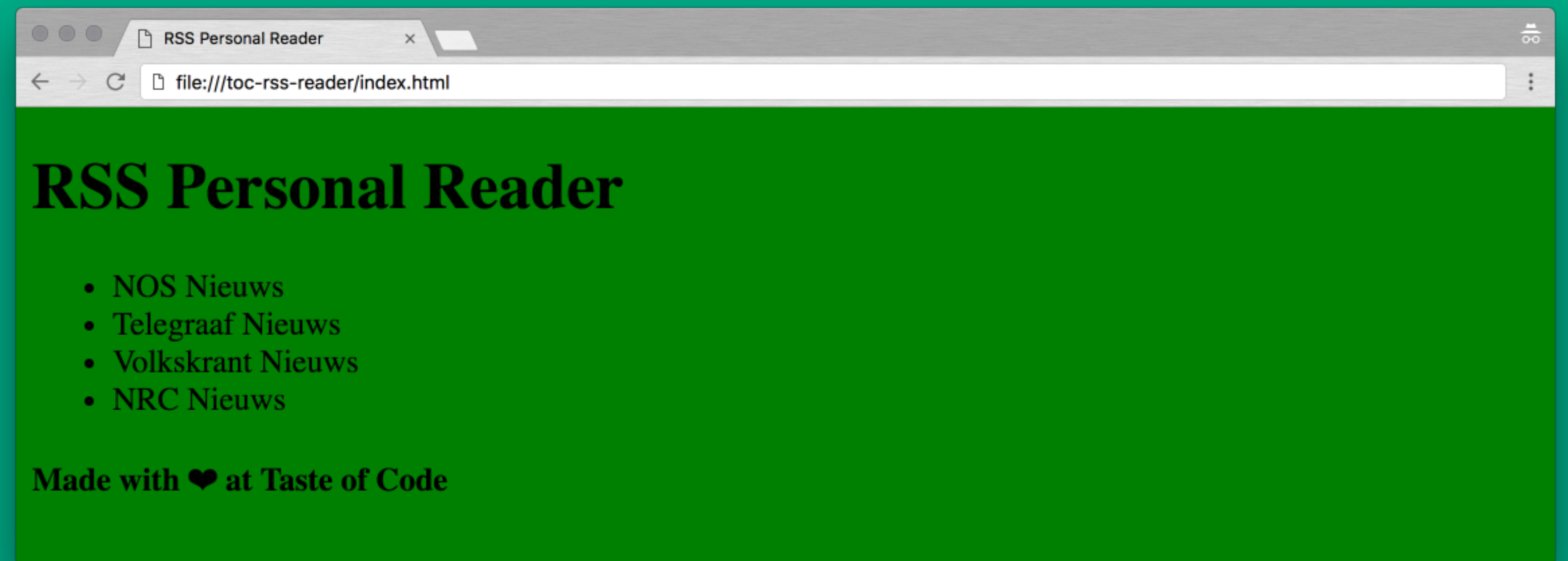
PROPERTY

VALUE

CSS DECLARATION

Intro to CSS

Exercises



[Link to Github](#)



CSS Selectors

How to target specific HTML tags



CSS Selectors

index.html

Select any HTML element
on your page by its
class name

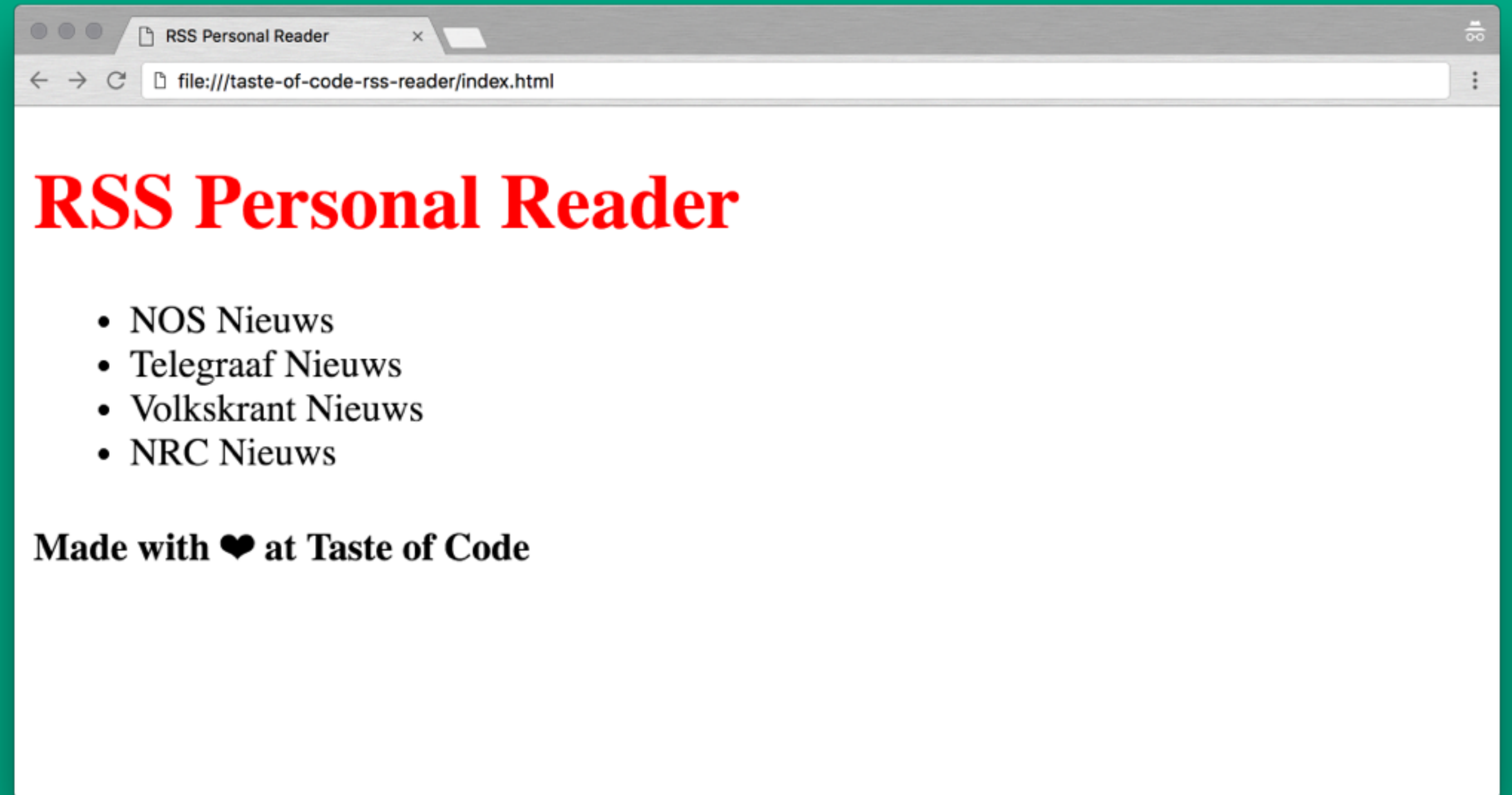
```
<!DOCTYPE html>
<html>
  <head>
    <title>Taste of Code</title>

    <style>
      .main-title {
        color: red;
      }
    </style>

  </head>
  <body>
    <h1 class="main-title">Agenda</h1>
  </body>
</html>
```

CSS Selectors

Exercises



[Link to Github](#)



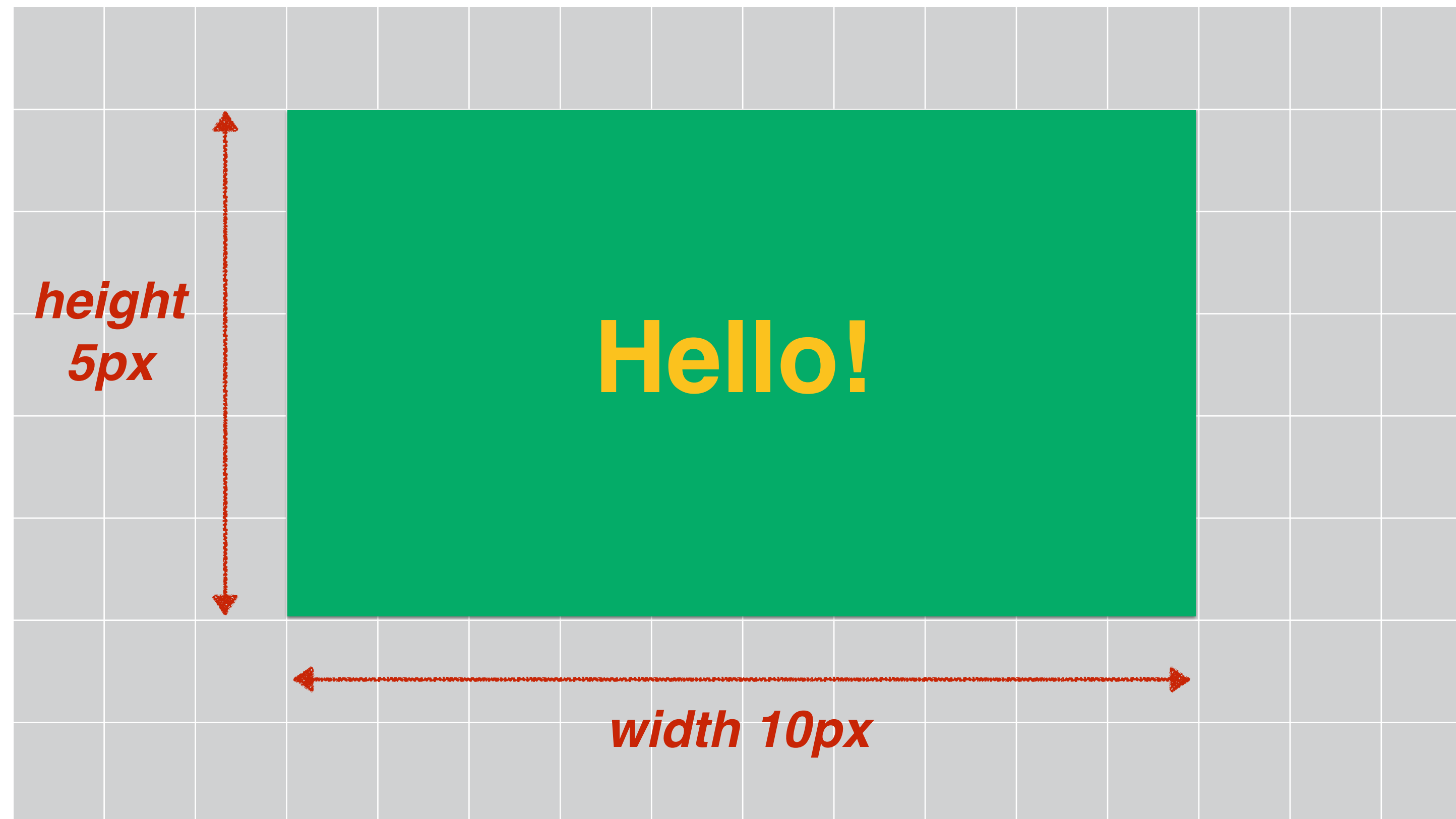
CSS Advanced Styling

Bringing space between elements

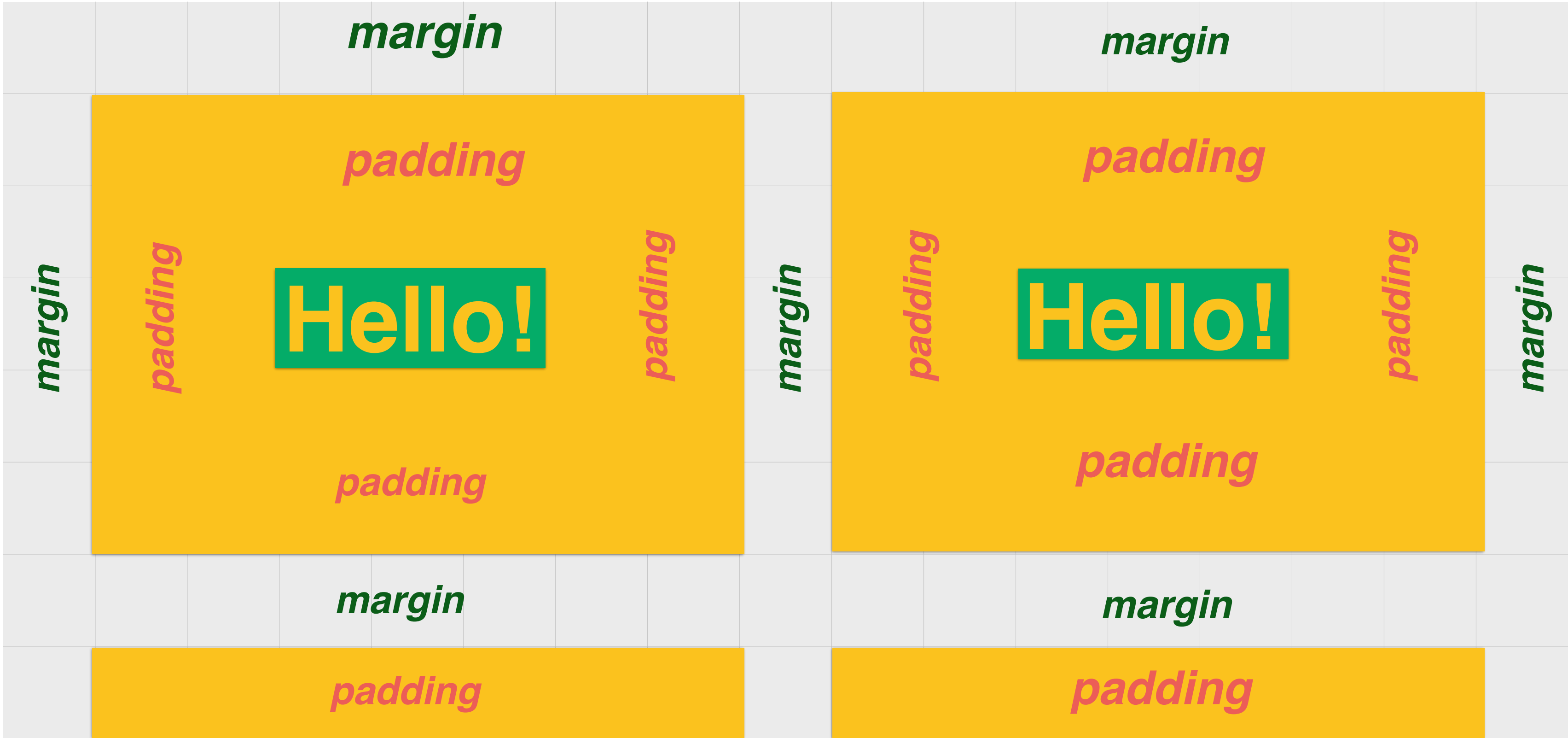


Height & Width

A screen consists of pixels.



Padding & Margin



Width, Height, Padding & Margin

css/styles.css

```
p {  
  padding: 30px;  
  margin: 10px;  
}  
  
.main-title {  
  padding: 10px;  
}  
  
.taste-of-code {  
  width: 200px;  
}
```


Using an external CSS file

Use the HTML
<link> tag

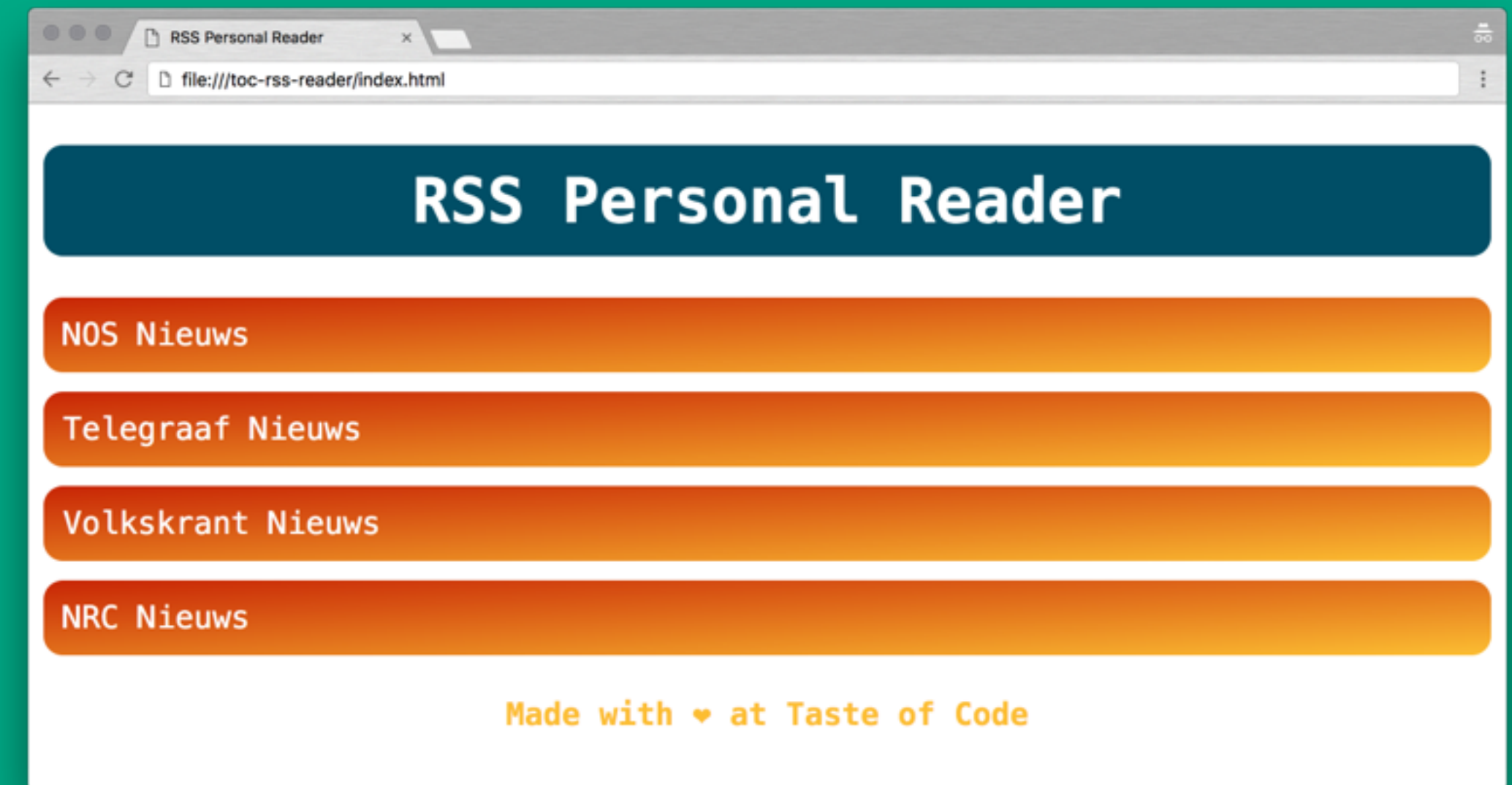
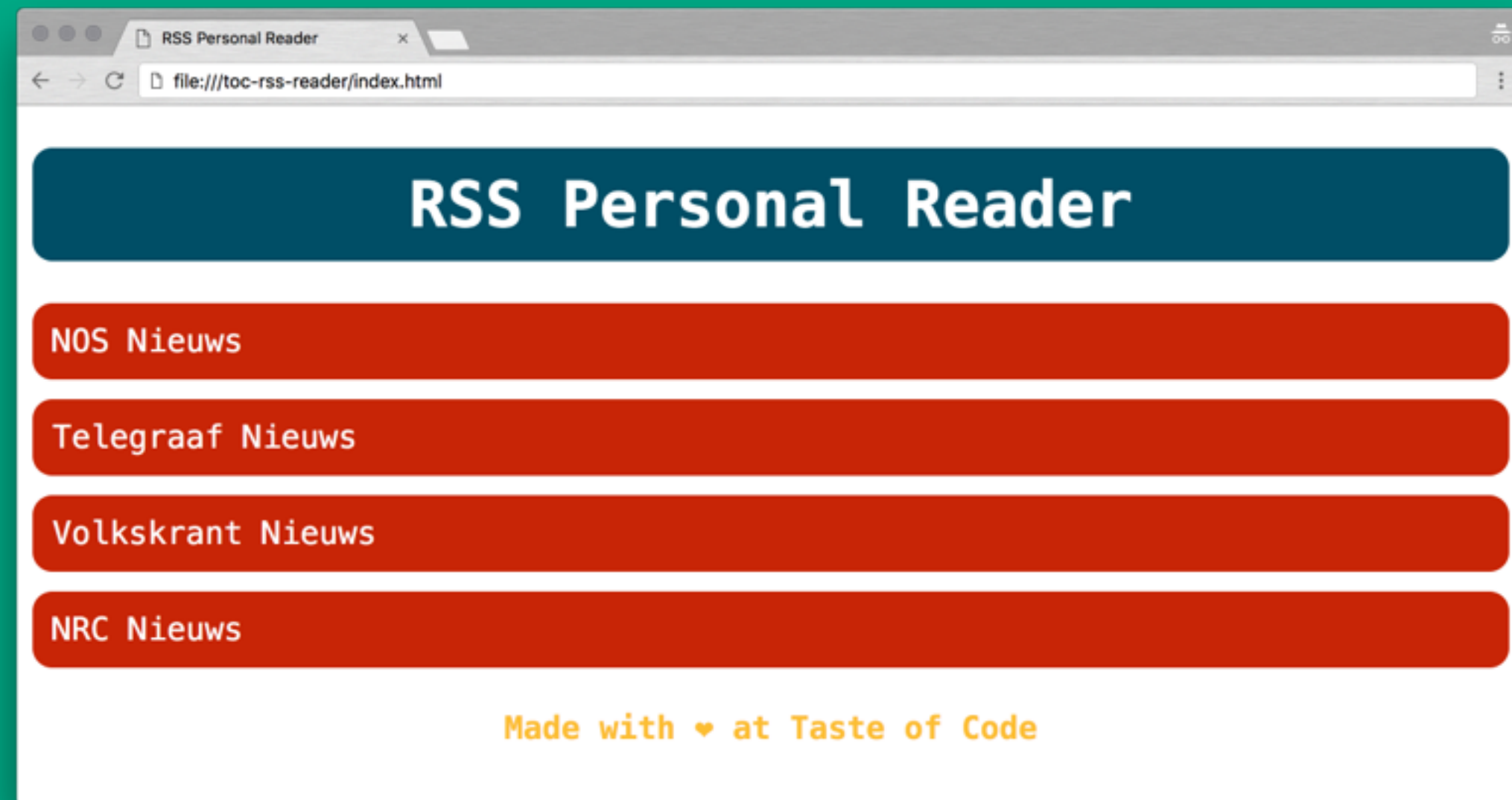
Its **href** attribute tells the browser
where the CSS file is stored.

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>RSS Personal Reader</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
    ...
  </body>
</html>
```

CSS Advanced Styling

Exercises



[Link to Github](#)



Intro to JavaScript

Making your web page interactive

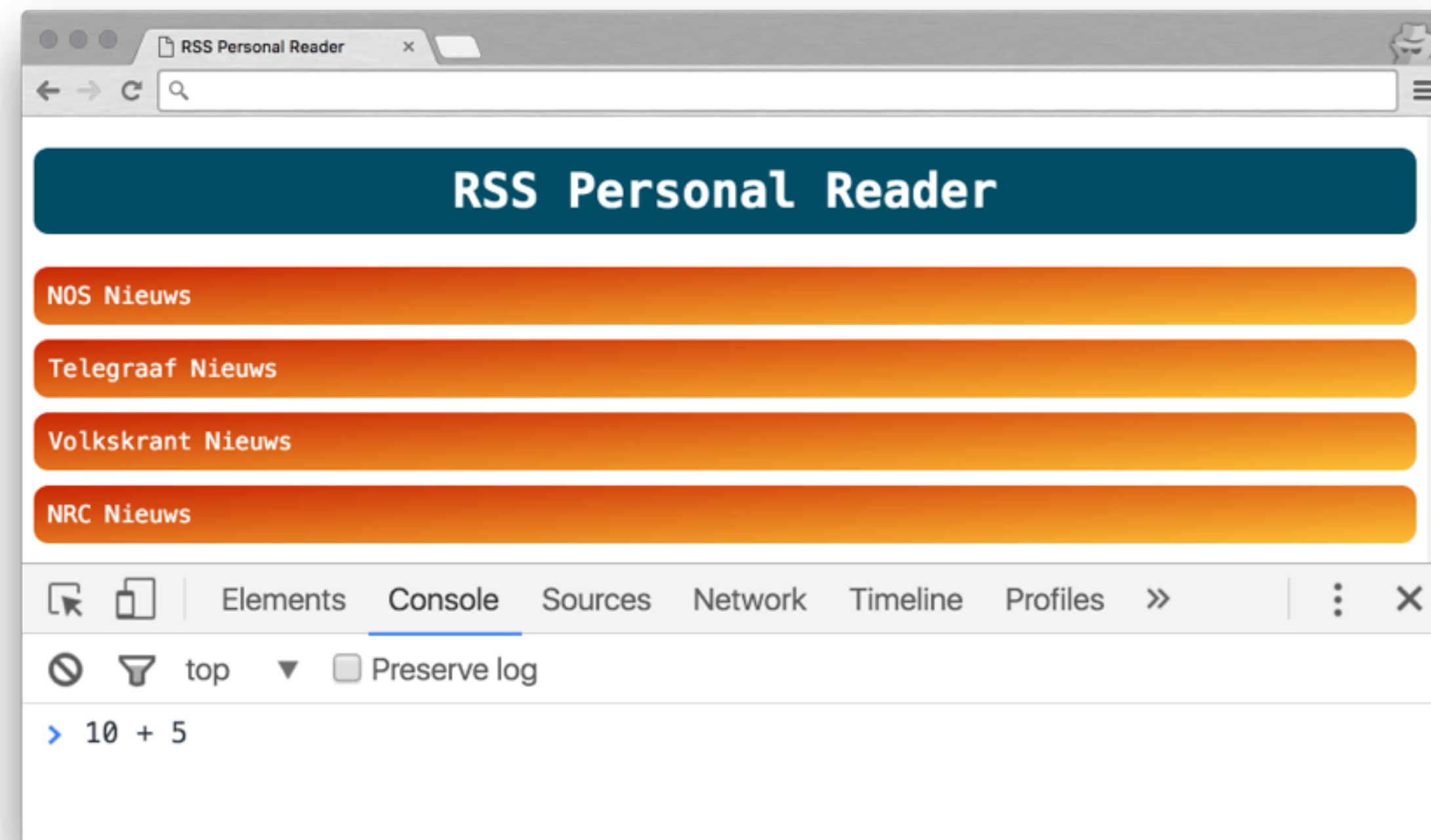


JavaScript lives in your browser

Open JavaScript Console

CMD + ALT + J (Mac)

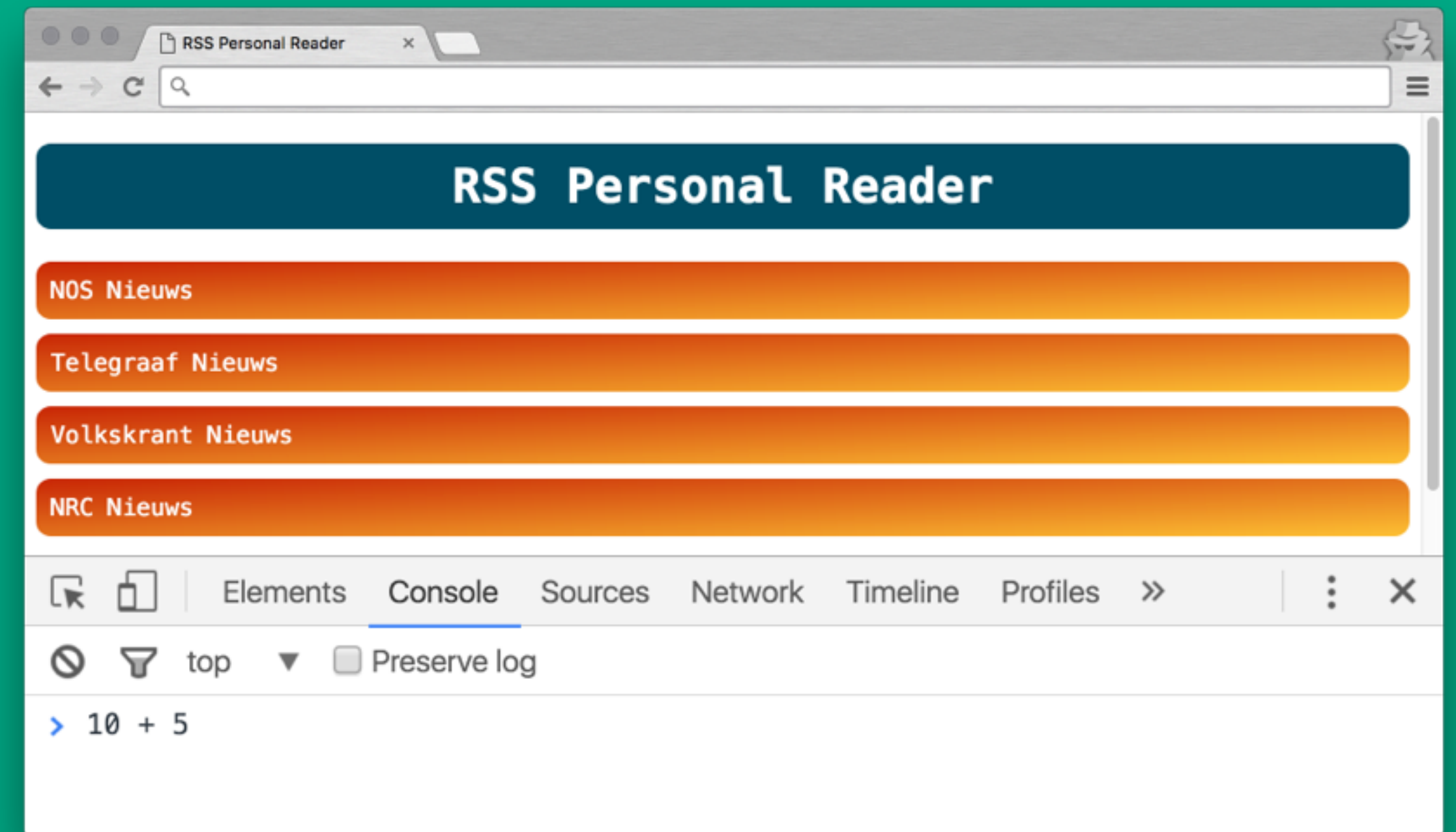
Control + Shift + J (Windows/Linux)



Intro to JavaScript

Exercises

[Link to Github](#)



JavaScript Interactivity

Making your web page interactive



How JavaScript makes your pages more interactive

Access content

select elements that have a `warning` class

Modify content

add a paragraph of text after the first `

` element

Program Rules or Instructions

write a script that writes some content depending on the day time

React to events

specify that a script should be run when a button is clicked

Objects (things) & Properties (characteristics)

In computer programming,
each thing in the world
can be represented as an object.

Each object can have
its own **properties** and **methods**.

Objects (things) & Properties (characteristics)

Window Object

The browser represents each window or tab using the **window** object.

javascript console

```
window.location
```

```
window.alert("We are building a RSS Feed Reader!")
```

Objects (things) & Properties (characteristics)

Document Object

The current web page loaded into each window is modelled using the **document** object.

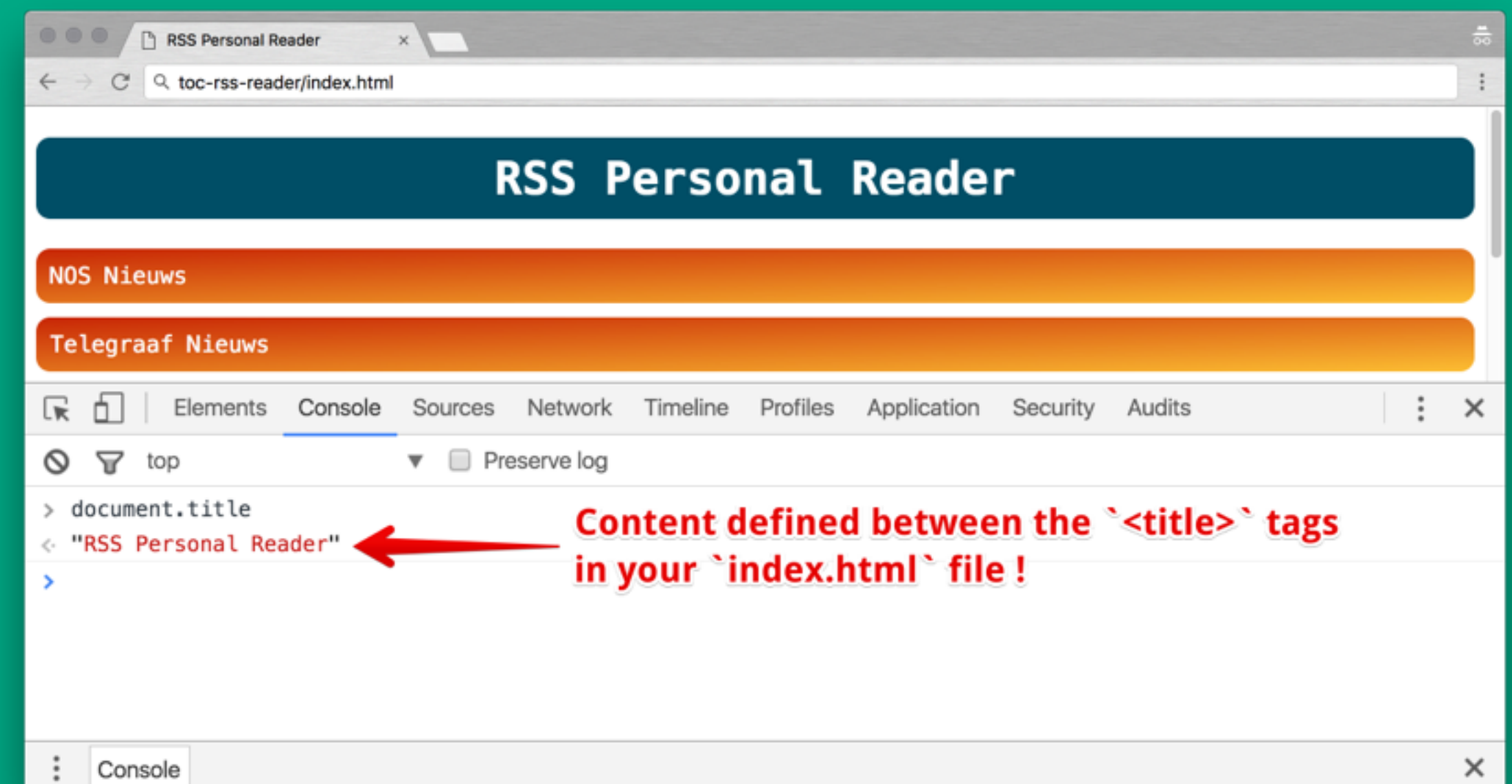
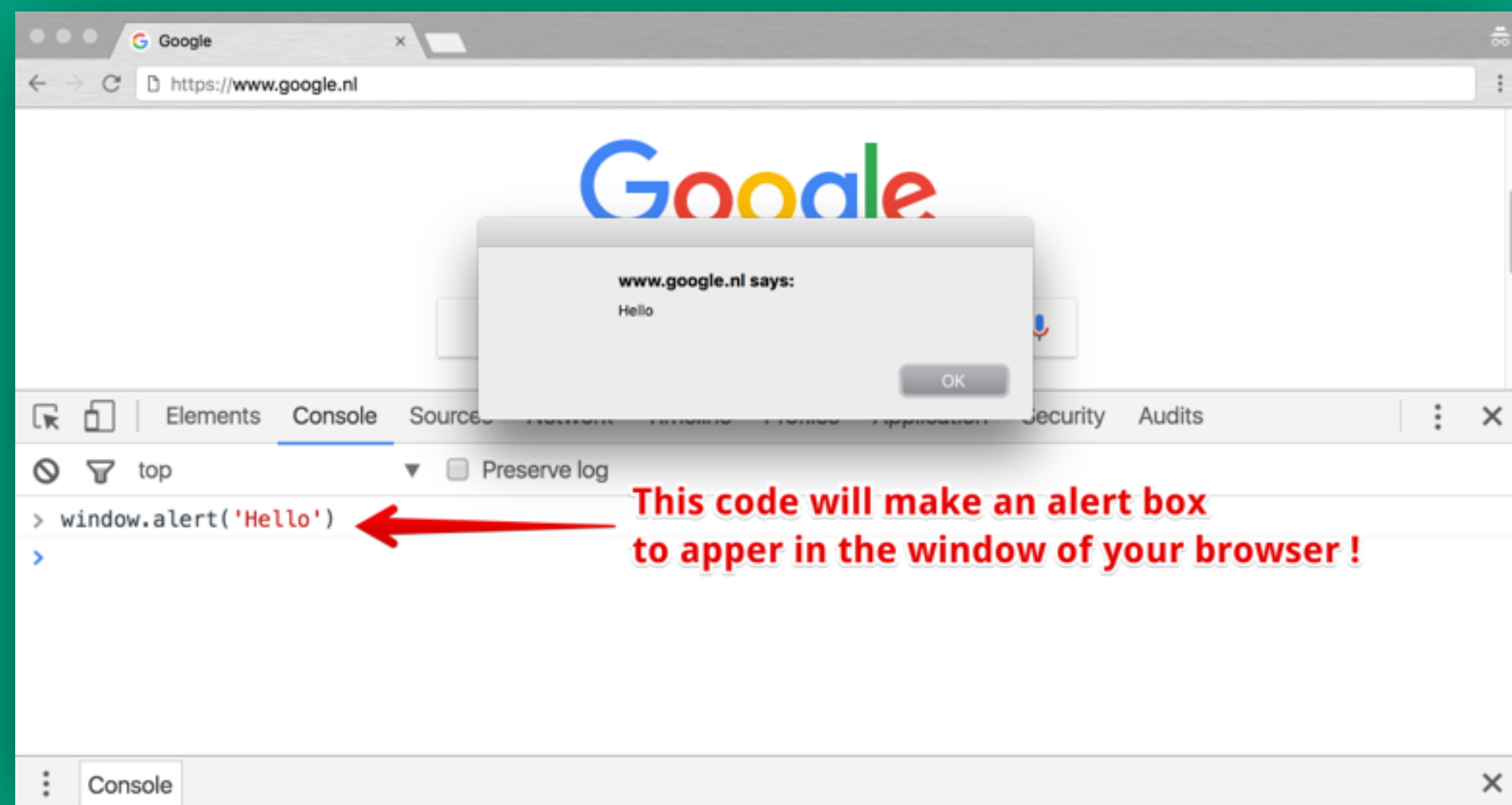
javascript console

```
document.title
```

```
document.write("It is gonna be awesome")
```

JavaScript Interactivity

Exercises



[Link to Github](#)



JavaScript Variables

Storing data

::{)

What is a Variable?

Variables can be thought of
as **named containers**.

You can place data into these containers
and refer to that data simply by
naming the container.

What is a Variable?

javascript console

```
// Store a string
var text = "It is gonna be awesome"
document.write(text)

// Store a number
var sum = 10 + 5
document.write(sum)

// Store a boolean (true/false)
var buildingFeedReaderAtTOC = true
document.write(buildingFeedReaderAtTOC)
```



JavaScript Variables

Exercises

js/rss.js

```
var nosUrl = "http://feeds.nos.nl/nosjournaal"
```

[Link to Github](#)



JavaScript Scripts

Giving Instructions

::{)

What is a Script?

A script is a **series of instructions** that a computer can follow step-by-step to achieve a goal.

First **state your goal** and then **list the tasks** that need to be completed to achieve it.

Your First Script

Greeting with JavaScript

This script adds a
greeting into an
HTML page.

The greeting changes
depending on the time.

greeting.js

```
var today = new Date()
var hourNow = today.getHours()

if (hourNow > 18) {
    var greeting = "Good evening!"
} else if (hourNow > 12) {
    var greeting = "Good afternoon!"
} else if (hourNow > 0) {
    var greeting = "Good morning!"
} else {
    var greeting = "Welcome!"
}

document.write("<h3>" + greeting + "</h3>")
```

External JavaScript

Use the HTML

`<script>` tag

Its **src** attribute tells the browser where the JavaScript file is stored.

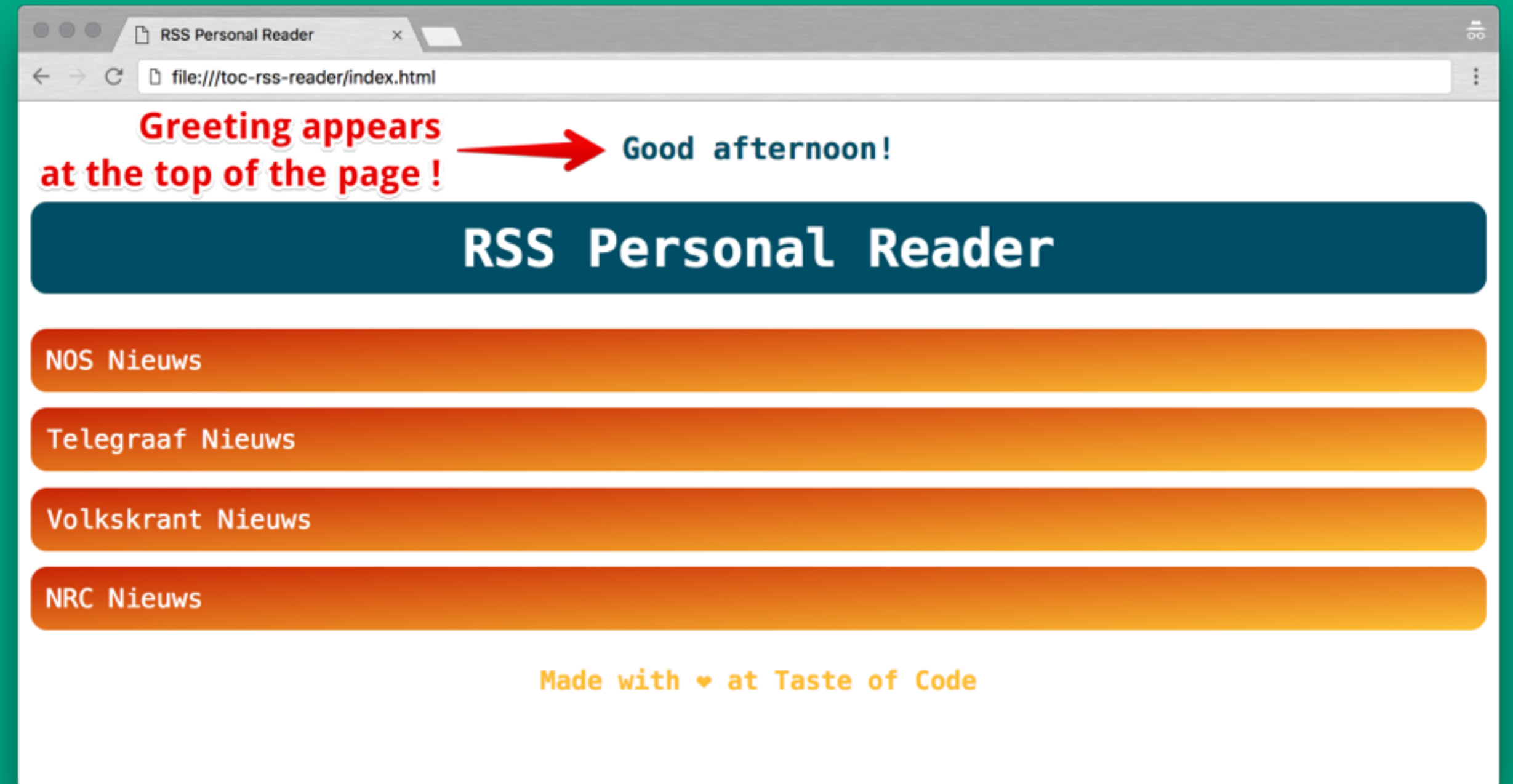
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>RSS Personal Reader</title>
    <link rel="stylesheet" type="text/css"
      href="css/styles.css">
  </head>
  <body>
    <script src="greeting.js"></script>

    <h1 class="main-title">
      RSS Personal Reader
    </h1>
  </body>
</html>
```

JavaScript Scripts

Exercises



[Link to Github](#)



JavaScript Objects

Modelling the world around you

`::{)`

Including jQuery

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>RSS Personal Reader</title>
    <link rel="stylesheet" type="text/css" href="css/styles.css">
  </head>
  <body>
    <!-- ... -->
    <script src="http://code.jquery.com/jquery-1.12.4.min.js"></script>
  </body>
</html>
```

What is an Object?

Objects group together a **set of variables** to **create a model** of something you would recognize from the real world.

In an object,
variables become known as **properties**.

What is an Object?

An object could be a library where a **collection of feed urls** are stored, categorized by site.

```
var feedsLibrary = {  
  nos: "http://feeds.nos.nl/nosjournaal"  
}
```

rss.js

key



value

Manipulating HTML with JavaScript

You can select any element in HTML by selecting its **class** or **id** attribute

★ How to show the **nosUrl** value within the **nos** unordered list?

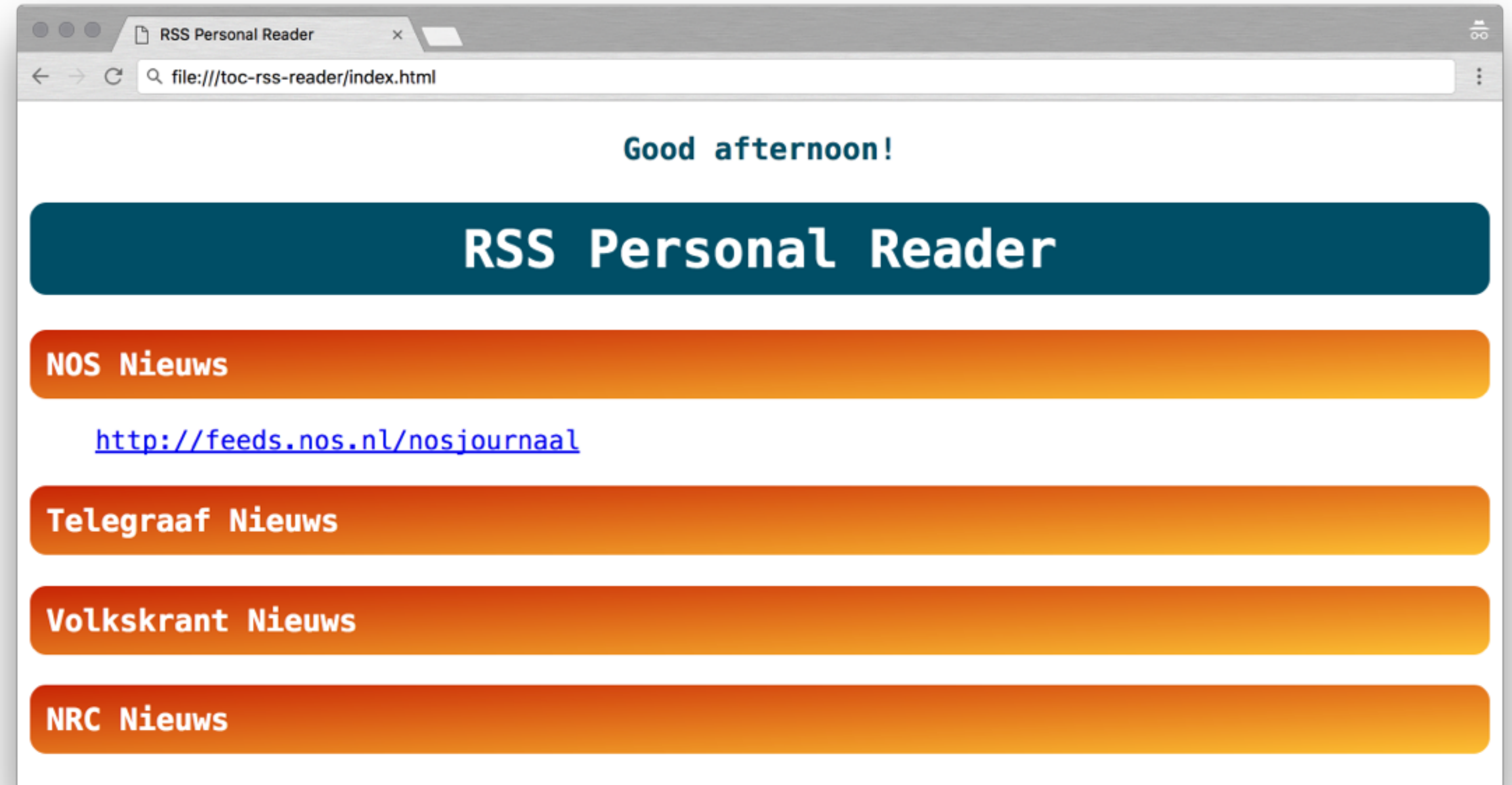
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>RSS Personal Reader</title>
    <link rel="stylesheet" type="text/css"
      href="css/styles.css">
  </head>

  <body>
    ...
    <ul id="nos"></ul>
    <script src="http://code.jquery.com...">
    <script src="js/rss.js"></script>
  </body>
</html>
```

Manipulating HTML with JavaScript

★ How to show the **nosUrl** value within the **nos** unordered list?



Manipulating HTML with JavaScript

js/rss.js

```
var nosUrl = "http://feeds.nos.nl/nosjournaal"

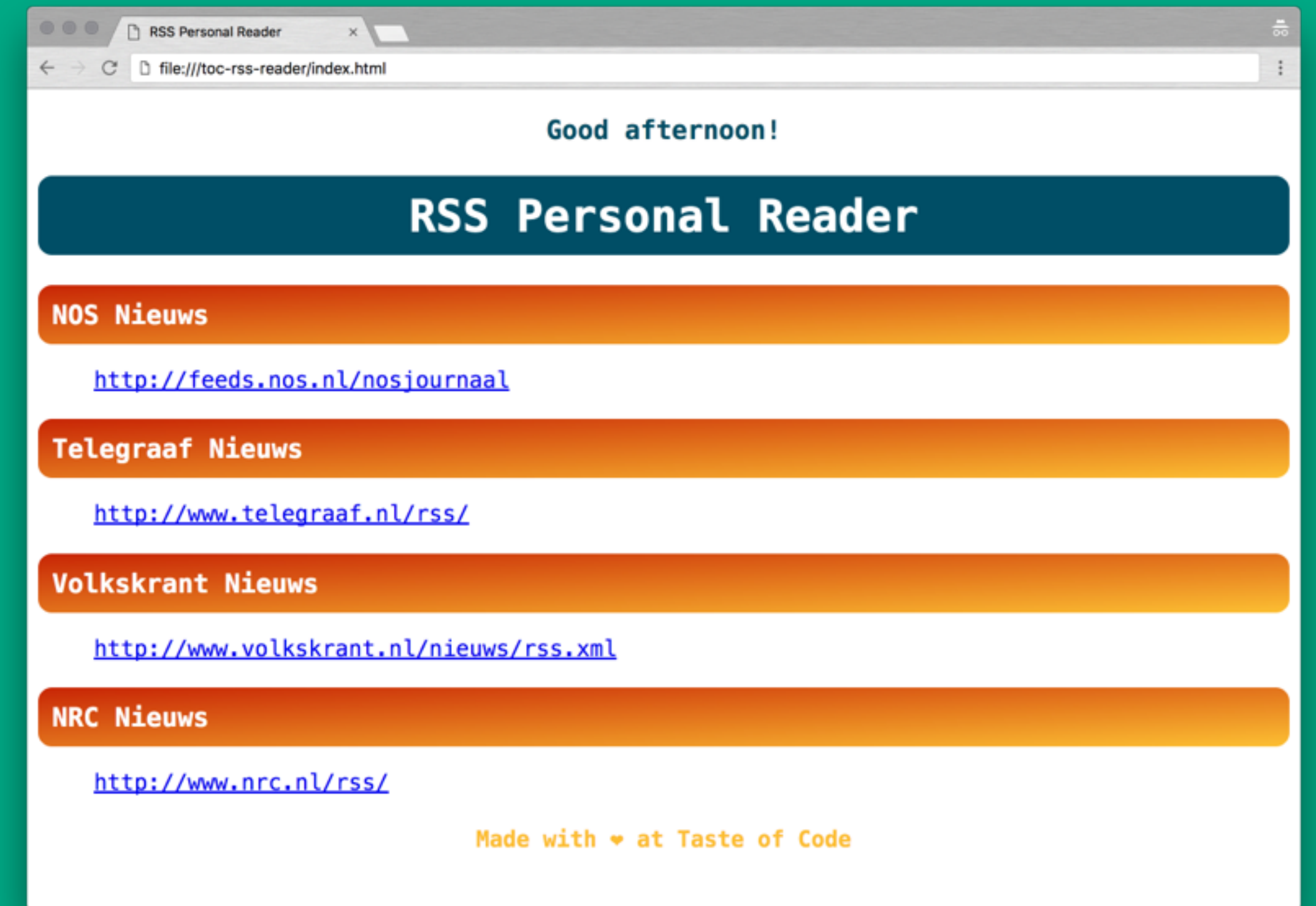
var feedsLibrary = {
  nos: nosUrl
}

var nosList = $('#nos')

nosList.append('<a href="' + feedsLibrary.nos + '">' + feedsLibrary.nos + '</a>')
```

JavaScript Objects

Exercises



[Link to Github](#)



Intro to YQL

Query, filter & combine data across the web



Exploring YQL (Yahoo Query Language)

✦ Our goal

Grab the RSS feed for each of the sites,
passing them through the **query string**.

We will be using the **JSON** form of the data.

Exploring YQL (Yahoo Query Language)

The screenshot shows the Yahoo! Developer Network (YDN) YQL console. The page title is "Sample YQL Response". The left sidebar contains a "Products" menu with links to "YQL", "YQL Console", "Create an App", "Get Started", "Documentation", "Forum", and "FAQ". The main content area is divided into two columns. The left column, titled "Examples", lists several tasks: "Extract HTML using XPath" (highlighted with a blue arrow), "Aggregate and filter multiple rss feeds", "Convert CSV to JSON or XML", "Extract HTML via CSS Selectors", "Get AppLinks meta data", and "Parse any XML source". The right column, titled "Responses", has radio buttons for "JSON" (selected) and "XML". Below this is a "YQL Query" input field containing the query: `SELECT * FROM feed WHERE url='http://feeds.nos.nl/nosjournaal' LIMIT 5`. A "Test" button is to the right of the query. Below the query is a "Response" section displaying a JSON object:

```
{
  "query": {
    "count": 44,
    "created": "2016-05-28T08:31:40Z",
    "lang": "en",
    "results": {
      "a": [
        {
          "class": "image",
          "href": "/wiki/File:Yahoo!_logo.svg",
          "img": {
```

 At the bottom, an "Endpoint" section shows the URL: `https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20html%20where%20u`.

YQL - Yahoo Developer Network

Search YDN

Products Events My Apps Sign In

Products

YQL

YQL Console

Create an App

Get Started

Documentation

Forum

FAQ

Sample YQL Response

Examples

- Extract HTML using XPath
- Aggregate and filter multiple rss feeds
- Convert CSV to JSON or XML
- Extract HTML via CSS Selectors
- Get AppLinks meta data
- Parse any XML source

Responses

☒ JSON ☐ XML

YQL Query ? [YQL Console](#)

SELECT * FROM feed WHERE url='http://feeds.nos.nl/nosjournaal' LIMIT 5

Test

Response

```
{
  "query": {
    "count": 44,
    "created": "2016-05-28T08:31:40Z",
    "lang": "en",
    "results": {
      "a": [
        {
          "class": "image",
          "href": "/wiki/File:Yahoo!_logo.svg",
          "img": {
```

Endpoint

https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20html%20where%20u

Visiting the endpoints



```
{
  - query: {
    count: 5,
    created: "2016-05-28T08:30:01Z",
    lang: "en",
    - results: {
      - item: [
        - {
          title: "Man in bijzijn van zoontje geliquideerd in Diemen-Noord",
          link: "http://feeds.nos.nl/~r/nosjournaal/~3/mqNQvZg8J58/2107729",
          - enclosure: {
            type: "image/jpeg",
            url: "http://nos.nl/data/image/2016/05/28/285742/m.jpg"
          },
          description: "De man werd onder vuur genomen toen hij in zijn auto zat. Hij was een bekende van de politie.<img src='http://feeds.feedburner.com/~r/nosjournaal/~4/mqNQvZg8J58' height='1' width='1' alt='' />",
          pubDate: "Sat, 28 May 2016 07:42:54 +02:00",
          - guid: {
            isPermaLink: "false",
            content: "http://nos.nl/1/2107729"
          },
          origLink: "http://nos.nl/1/2107729"
        },
        + { ... },
        + { ... },
        + { ... },
        + { ... }
      ]
    }
  }
}
```

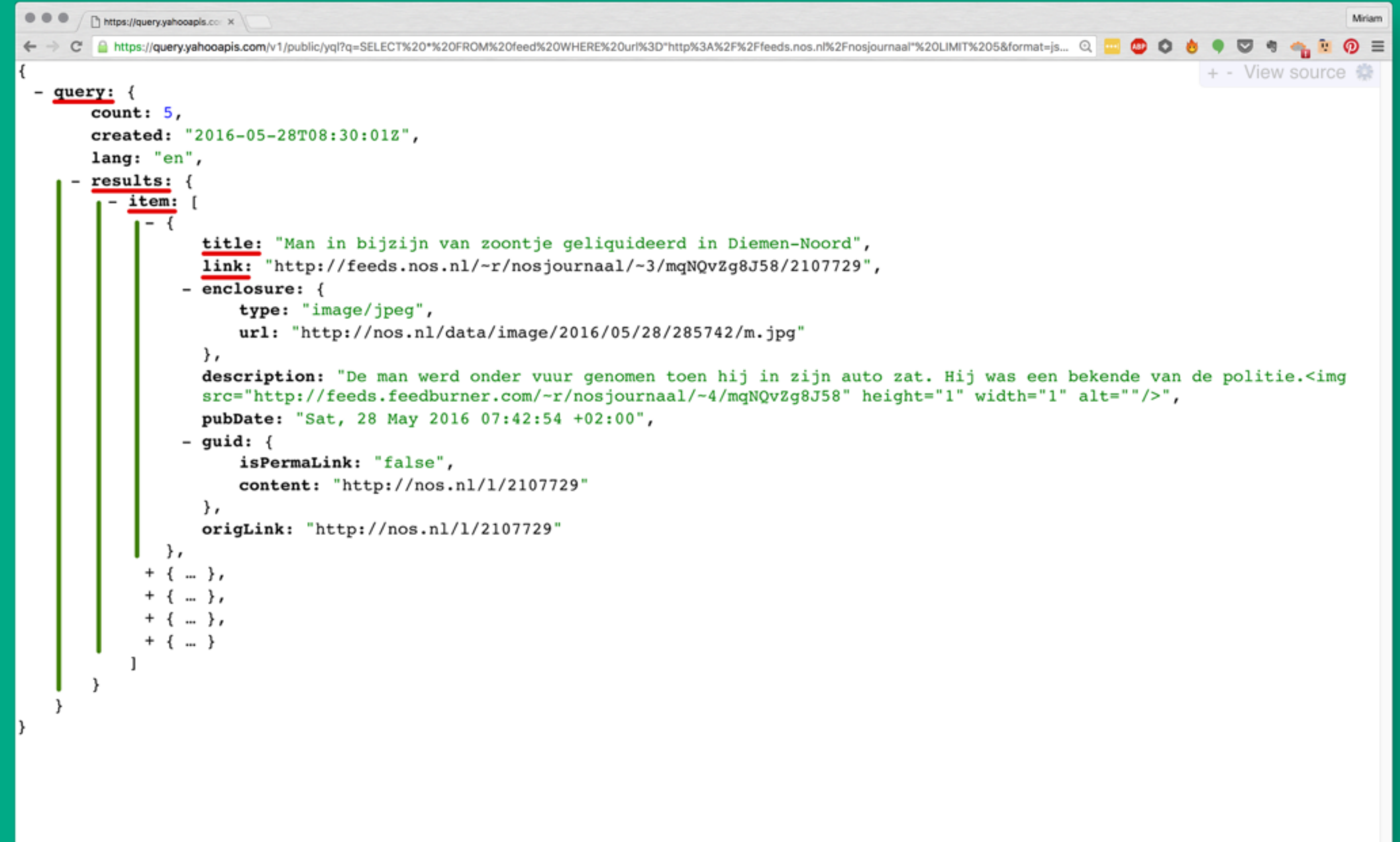

Building up the query string

yahoo sql language

```
SELECT * FROM feed WHERE url="path/to/rss/feed" LIMIT 5
```

Intro to YQL

Exercises



The screenshot shows a web browser window with the address bar displaying a YQL query URL. The main content area shows a JSON response from the query. The JSON is formatted with syntax highlighting and a tree view on the left. The response includes a 'query' object with 'count' and 'created' fields, and a 'results' array containing one 'item' object. The 'item' object has fields for 'title', 'link', 'enclosure' (with 'type' and 'url'), 'description', 'pubDate', 'guid' (with 'isPermaLink' and 'content'), and 'origLink'.

```
{
  - query: {
    count: 5,
    created: "2016-05-28T08:30:01Z",
    lang: "en",
  - results: {
    - item: [
      - {
        title: "Man in bijzijn van zoontje geliquideerd in Diemen-Noord",
        link: "http://feeds.nos.nl/~r/nosjournaal/~3/mqNQvZg8J58/2107729",
        - enclosure: {
          type: "image/jpeg",
          url: "http://nos.nl/data/image/2016/05/28/285742/m.jpg"
        },
        description: "De man werd onder vuur genomen toen hij in zijn auto zat. Hij was een bekende van de politie.<img src='http://feeds.feedburner.com/~r/nosjournaal/~4/mqNQvZg8J58' height='1' width='1' alt='' />",
        pubDate: "Sat, 28 May 2016 07:42:54 +02:00",
        - guid: {
          isPermaLink: "false",
          content: "http://nos.nl/1/2107729"
        },
        origLink: "http://nos.nl/1/2107729"
      },
      + { ... },
      + { ... },
      + { ... },
      + { ... }
    ]
  }
}
```

[Link to Github](#)

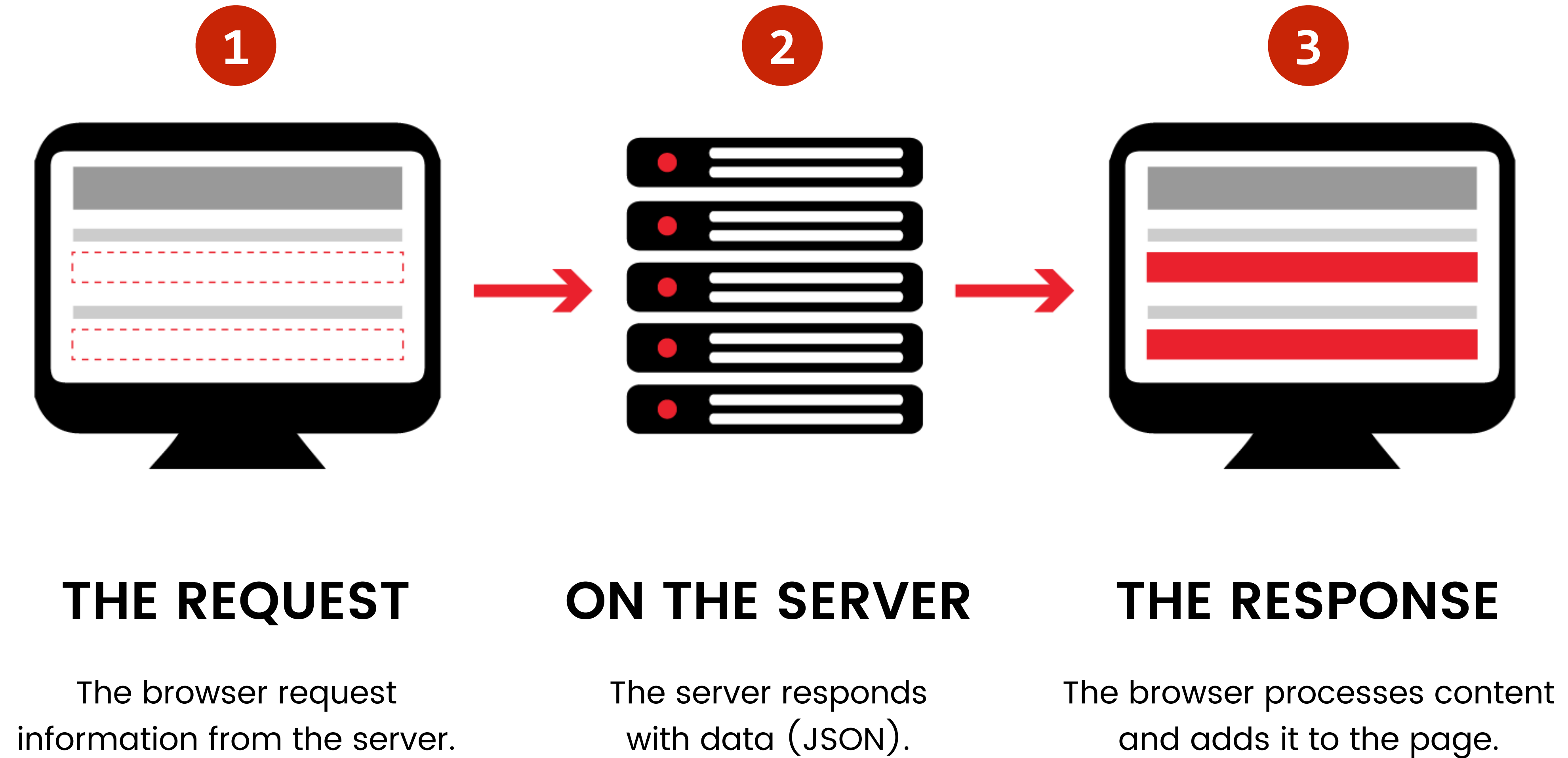


Intro to AJAX

Getting Information from others



How AJAX works



Building up the path to make the request

https://query.yahooapis.com/v1/public/yql?q=SELECT%20*%20FROM%20feed%20WHERE%20url%3D%27http%3A%2F%2Ffeeds.nos.nl%2Fnosjournaal%27%20LIMIT%205&format=json

rss.js

```
var nosUrl = "http://feeds.nos.nl/nosjournaal"

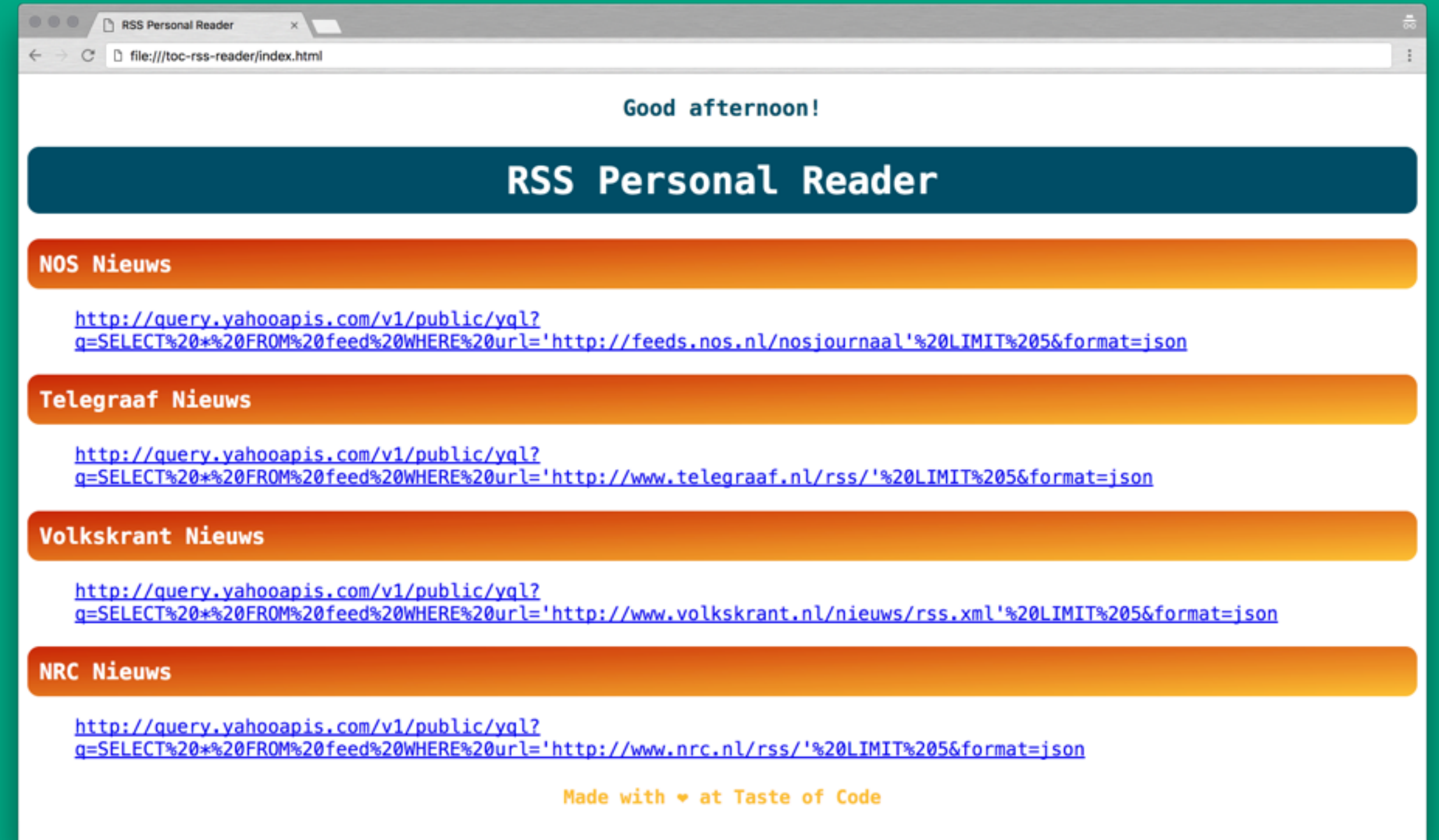
var baseUrl = "http://query.yahooapis.com/v1/public/yql?q="

var queryString = encodeURIComponent
    ("SELECT * FROM feed WHERE url='" + nosUrl + "' LIMIT 5")

var format = "&format=json"
var rssFeedPath = baseUrl.concat(queryString, format)
```

Intro to AJAX

Exercises



[Link to Github](#)



AJAX Requests

Asking for desired data



Making an AJAX request

jQuery provides several methods to make AJAX requests and handle responses.

To get JSON data from an external web server, you can use the `.getJSON()` method.

Making an AJAX request

✨ **url** = where to make the request to get the data

✨ **callback** = what needs to be done when data is returned

javascript

```
$.getJSON(url, function(response) {  
    // Do something with response and data returned  
});
```

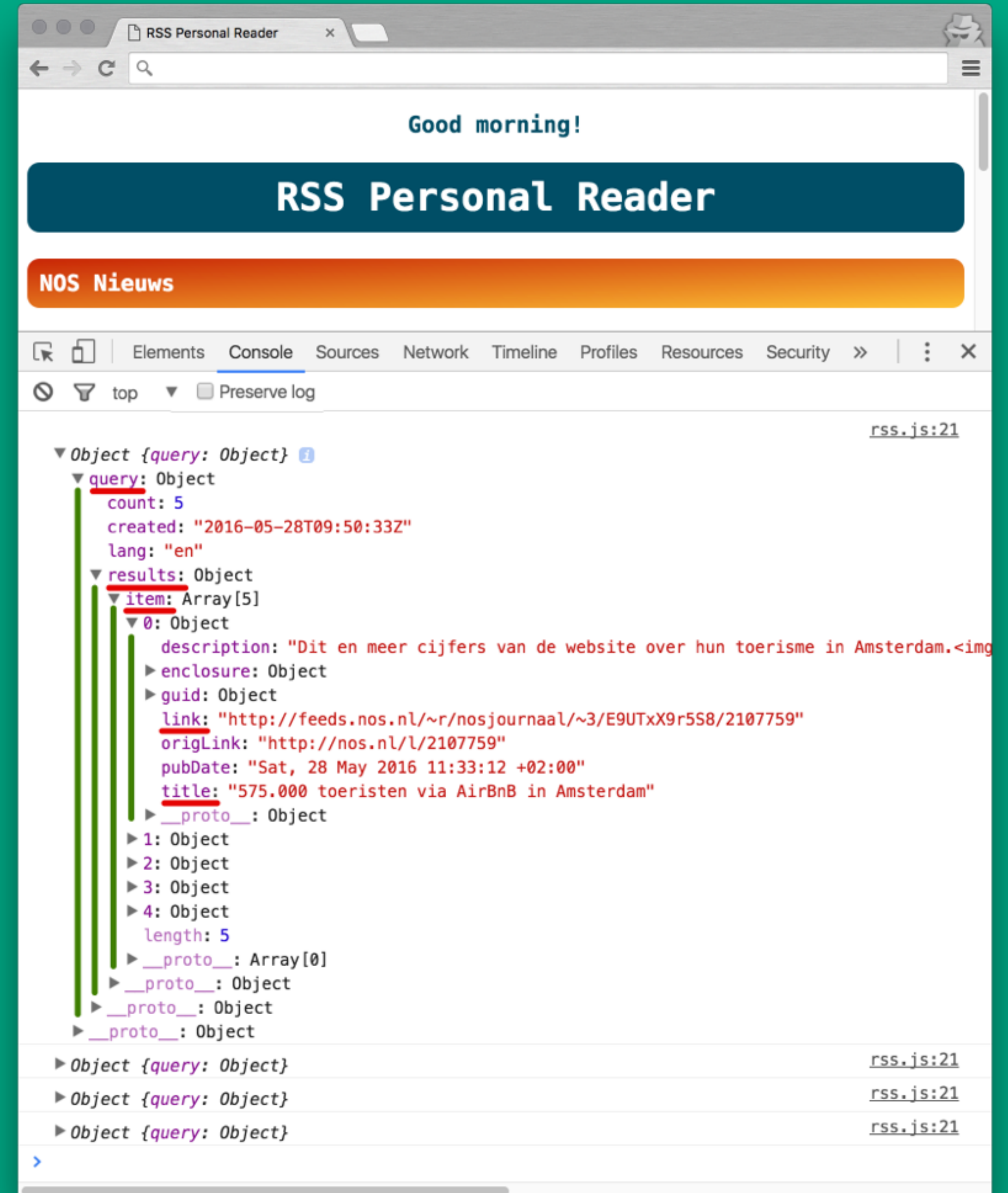
callback



AJAX Requests

Exercises

[Link to Github](#)



AJAX Responses

Using the returned data

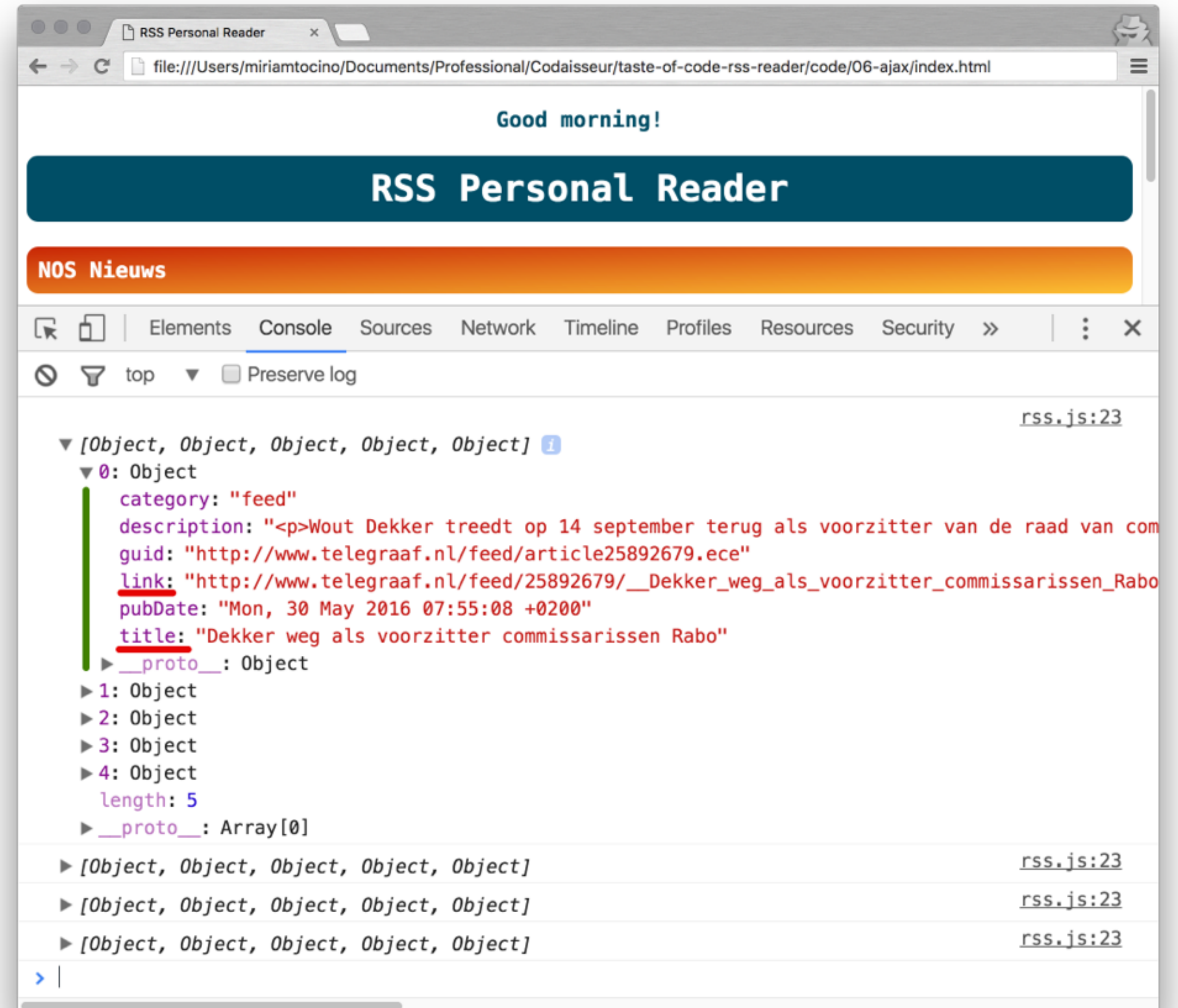


Handling an AJAX response

`response.query.results.item`

is an array of objects

★ **Arrays** are another way of storing data as a collection



Handling an AJAX response

`response.query.results.item` stores a collection of articles

rss.js

```
var feedItems = response.query.results.item
```

```
// First item data
```

```
link1 = feedItems[0].link
```

```
title1 = feedItems[0].title
```

```
// Second item data
```

```
link2 = feedItems[1].link
```

```
title2 = feedItems[1].title
```

```
// ...
```

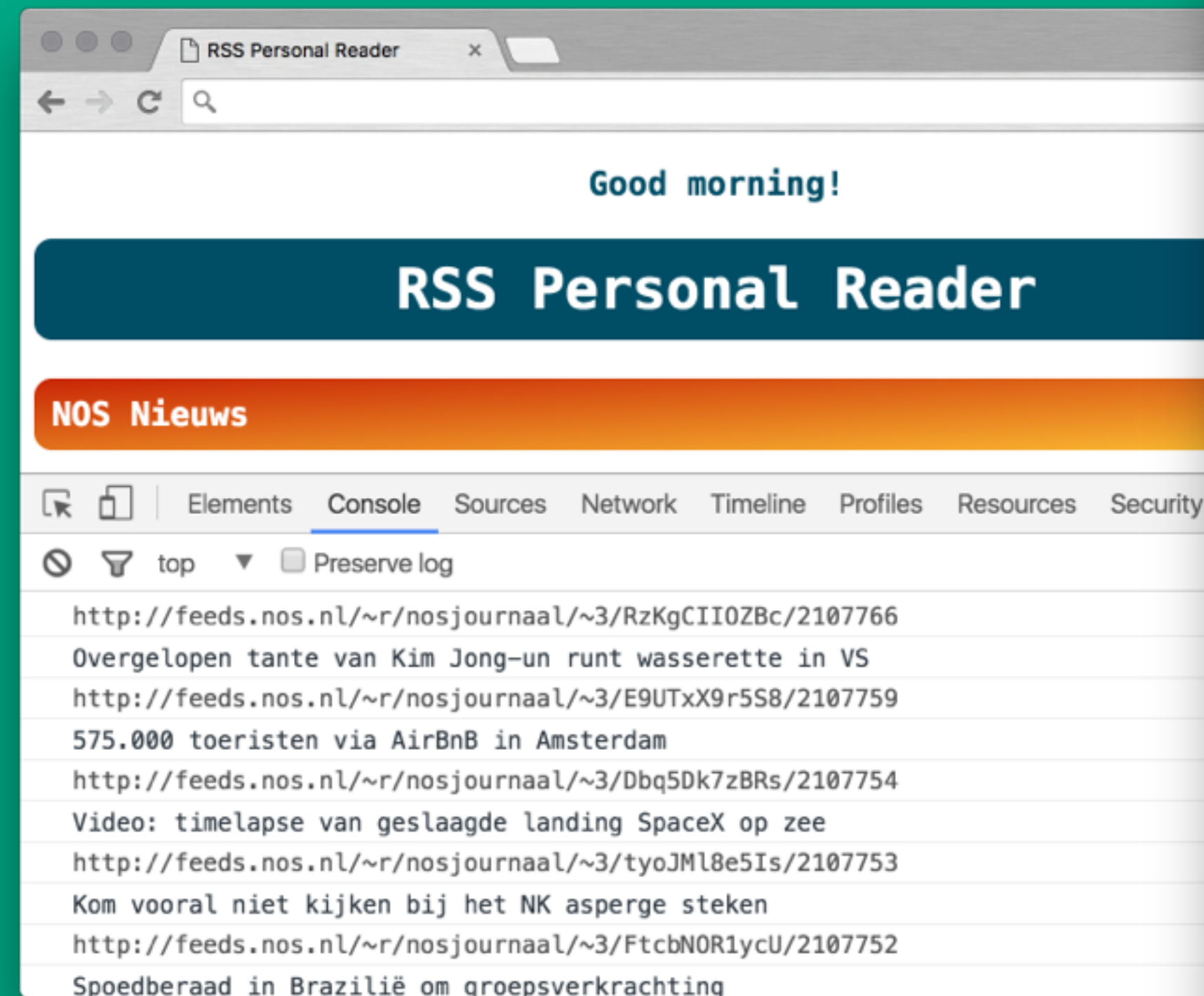
[0] [1] [2] ...

✦ The **index** represents the element position within the array

AJAX Responses

Exercises

[Link to Github](#)



Fine-tuning

