

Taste of Code

RSS Feed Reader

:{) Codaisseur
@miriamtocino

What Do We Need

Editor

Atom, Notepad++, Vim

Browser

Chrome, Firefox

Slack

<https://nos-hq.slack.com/>



Outline Project

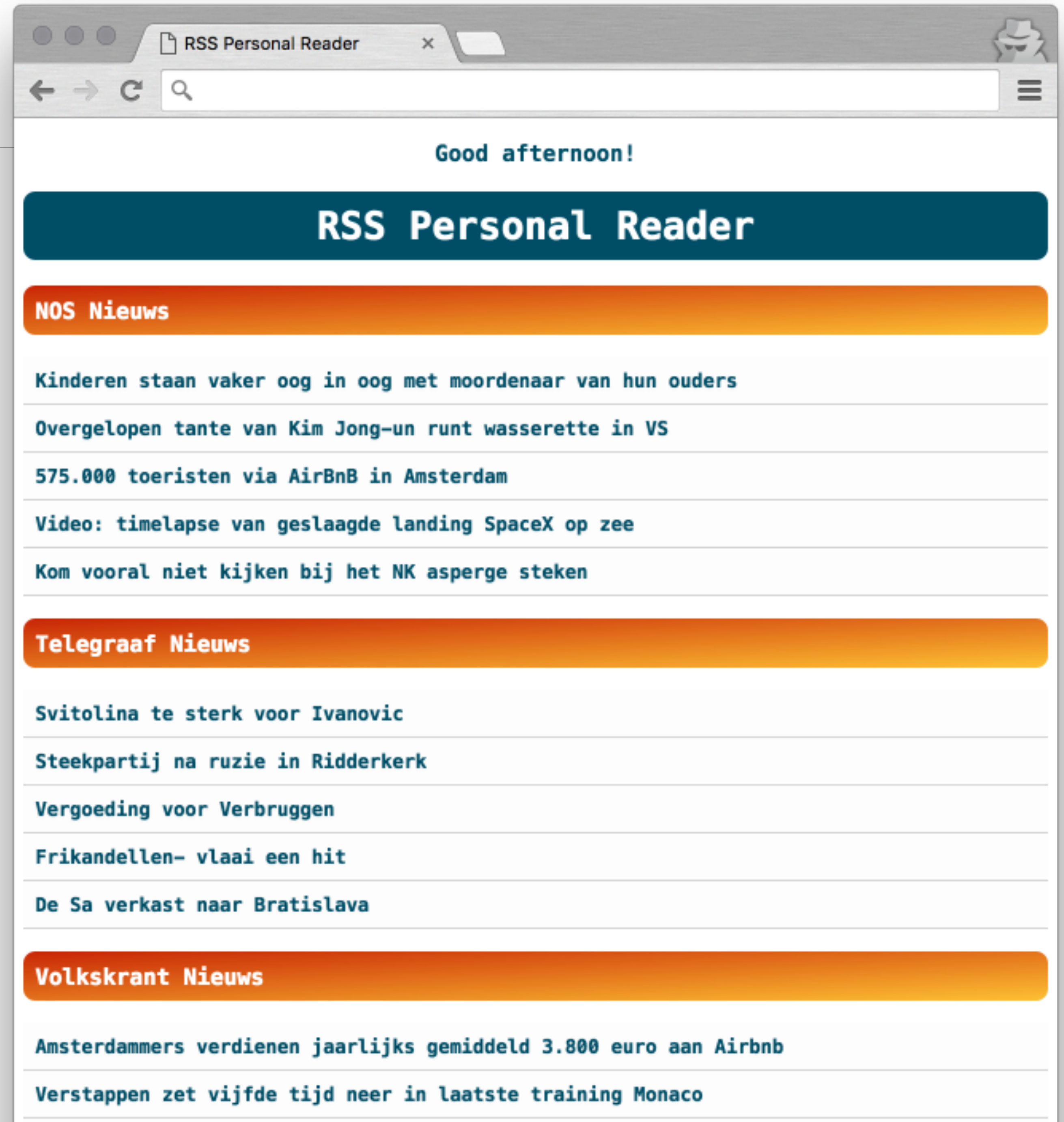
What You Will Build

It's always helpful to first outline
what you want your **project to do/achieve**.

What You Will Build

RSS Feed Reader

Using HTML, CSS
and JavaScript (jQuery)



What We Will Build

ToDo:

- 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





HTML

Content & Structure

HTML is the language of the web.



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 browser window mockup with a dark header bar containing window control buttons (red, yellow, green) and navigation icons (back, forward, search, star). 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

The browser window has a dark gray header bar with three colored window control buttons (red, yellow, green) on the left and navigation icons (back, forward, search, star) on the right. The main content area is white and contains the following text:

On the menu today:

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

TAGS open...

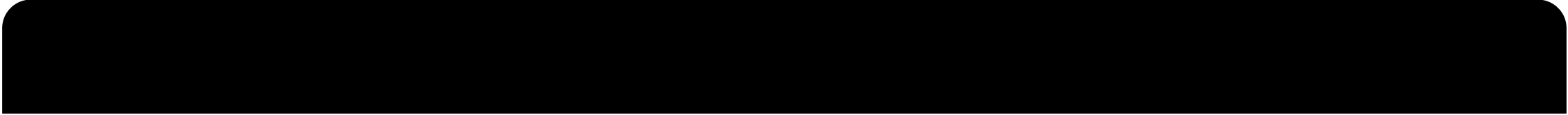


```
<h1>Agenda
```

Opening tag = starts heading



TAGS open and close



```
<h1>Agenda</h1>
```

Opening tag = starts heading
Closing tag = stops heading



Exercise

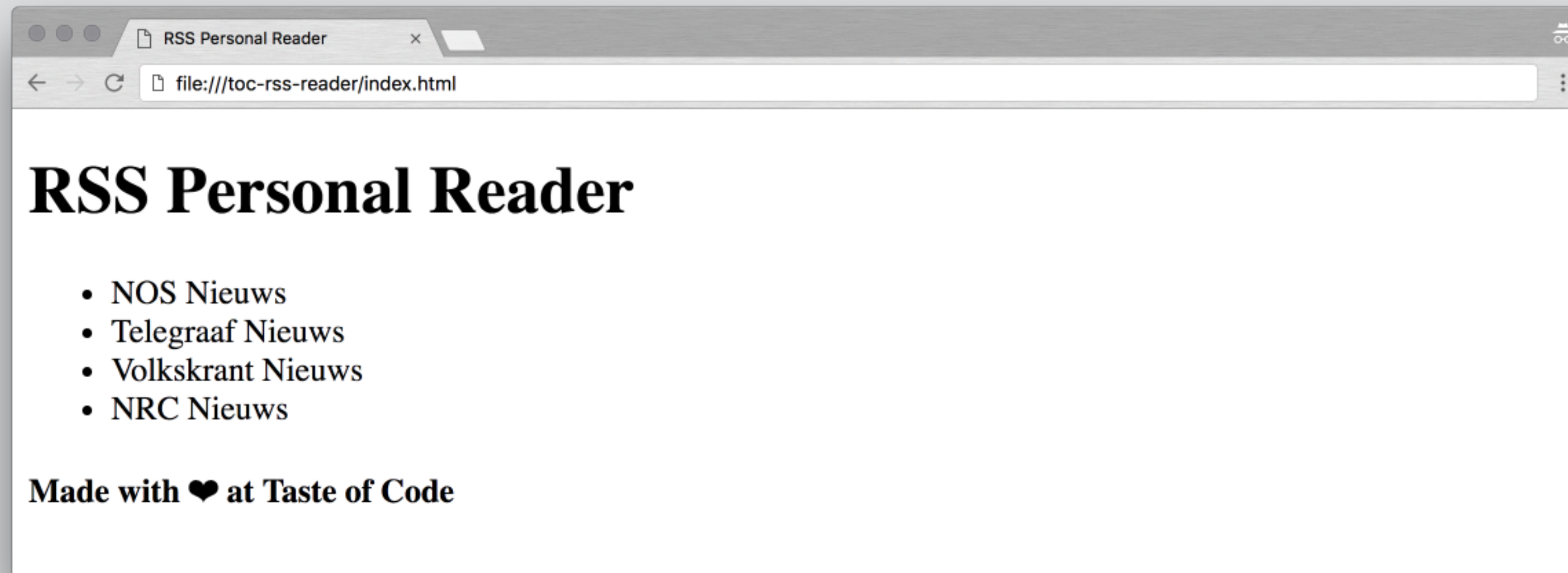
Your Very First HTML

Make a new folder for your project
and create an **index.html** file.

[Go to exercise](#)



Exercise





CSS

Style & Presentation

CSS describes **how HTML looks.**



What is CSS?

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

text color
fonts style
spacing between elements
background images



What is CSS?

color

sets the color of the text

index.html

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

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

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



What is CSS?

Agenda

On the menu today:

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



Exercise

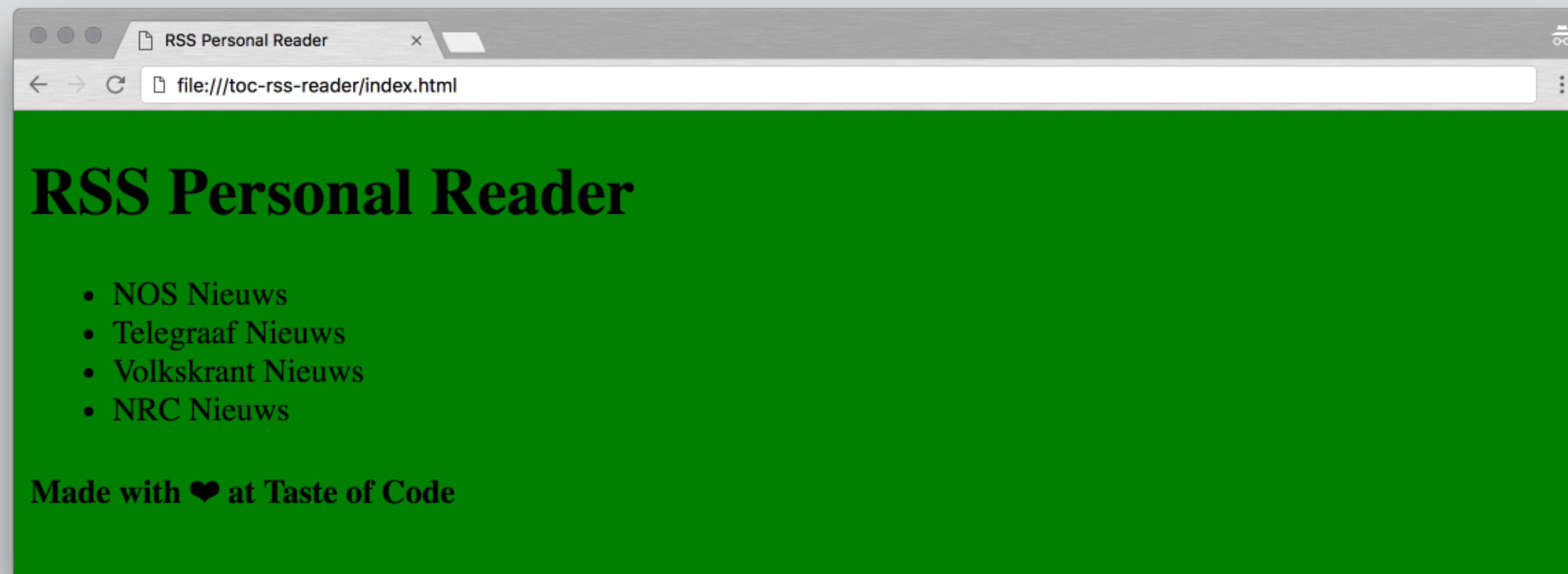
Page Background Color

Change the **background-color** of the web page to **green**.

[Go to exercise](#)



Exercise



CSS Selectors

Select element on your page
by its **class** name

index.html

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

    <style>
      .warning {
        color: red;
      }
    </style>

  </head>
  <body>
    <h1 class="warning">Agenda</h1>
  </body>
</html>
```



Exercise

Warning

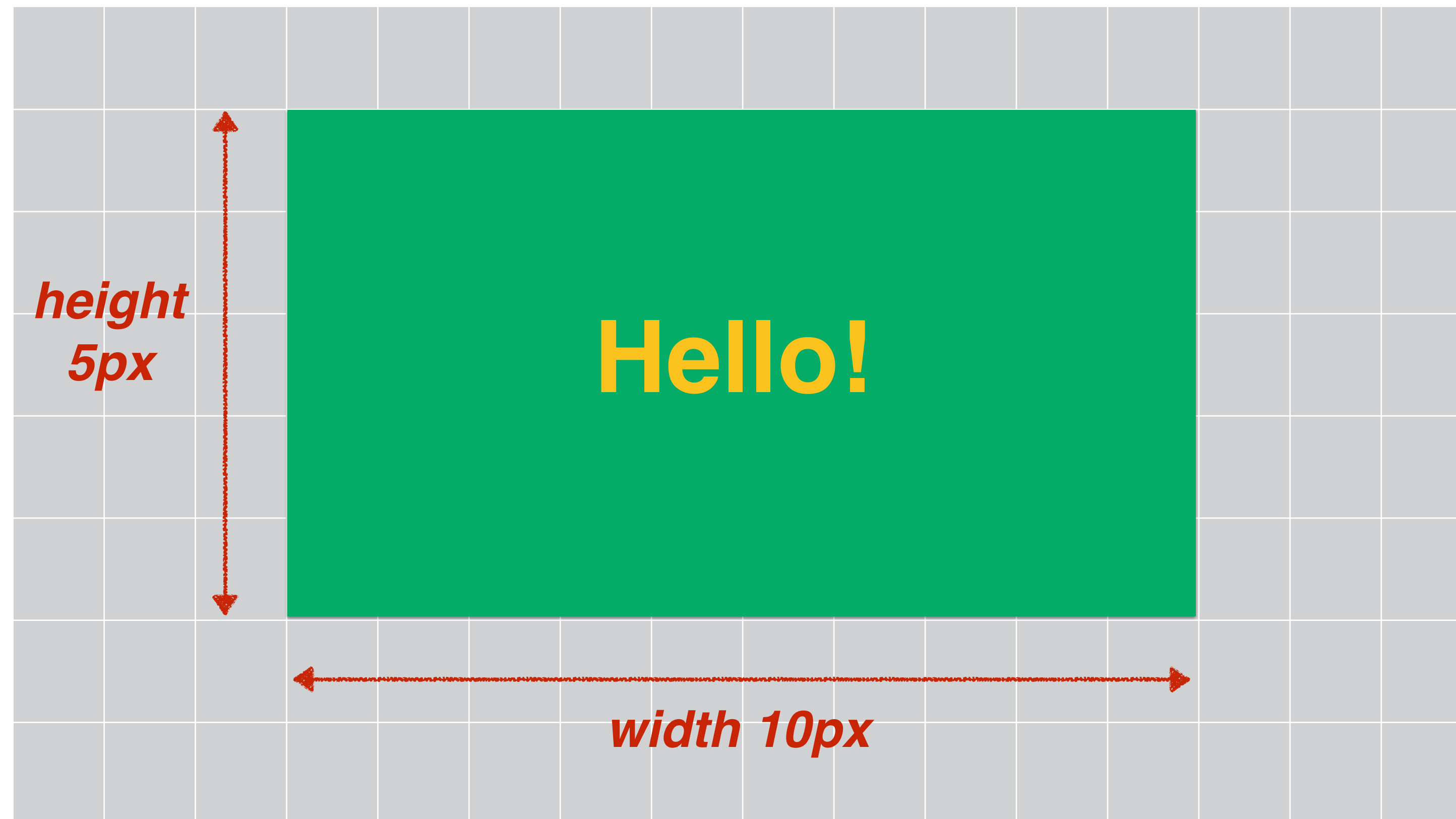
Apply the warning class to the main heading in your web page and change its color.

[Go to exercise](#)

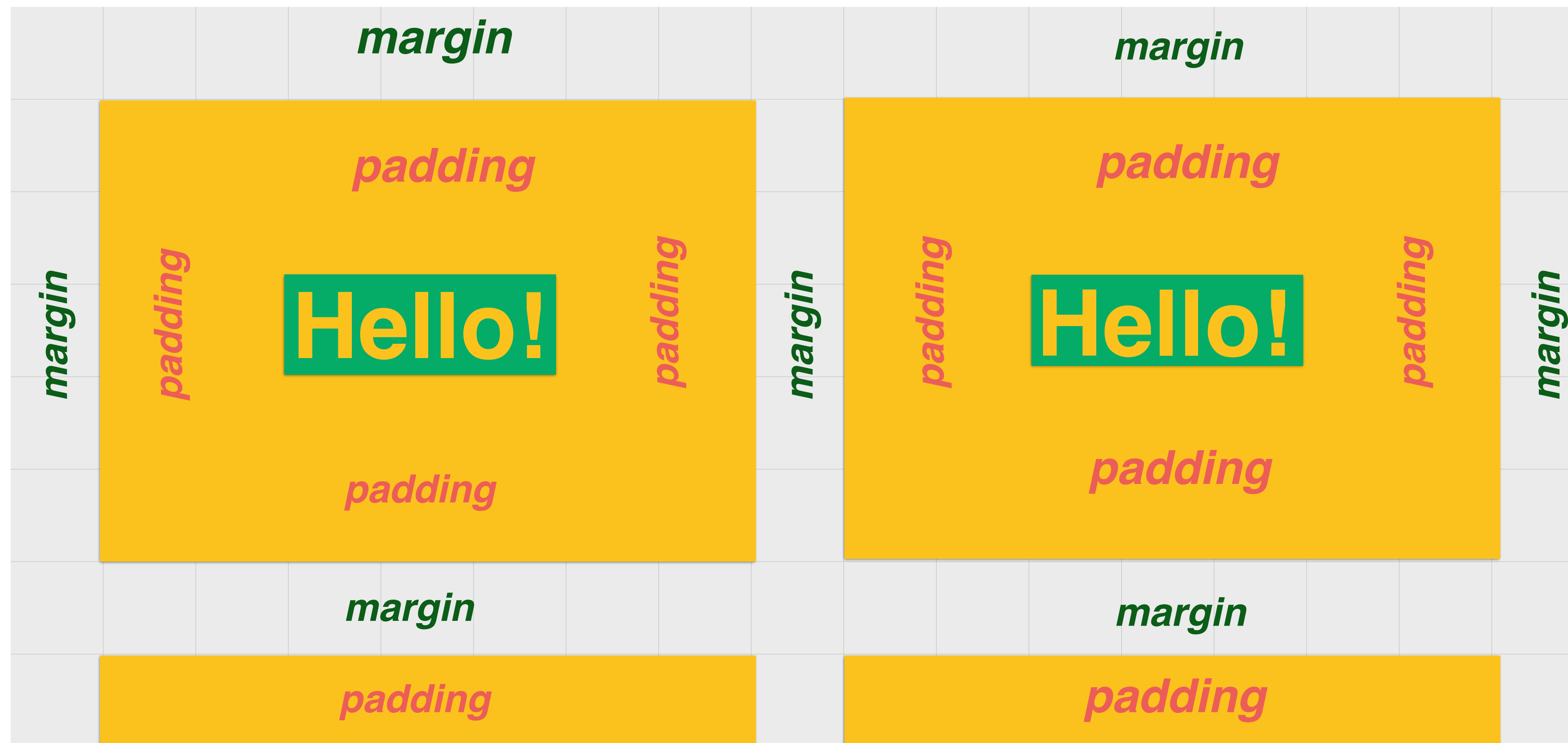


Padding & Margin in Pixels

A screen consists of pixels



Padding & Margin in Pixels



Padding & Margin in Pixels

css/styles.css

```
p {  
  padding: 30px;  
  margin: 10px;  
}  
  
.main-title {  
  padding: 10px;  
}  
  
#taste-of-code {  
  margin: 25px;  
}
```



Linking to CSS file from HTML page

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="css/styles.css">
  </head>
  <body>
    ...
  </body>
</html>
```

Exercise

Styling

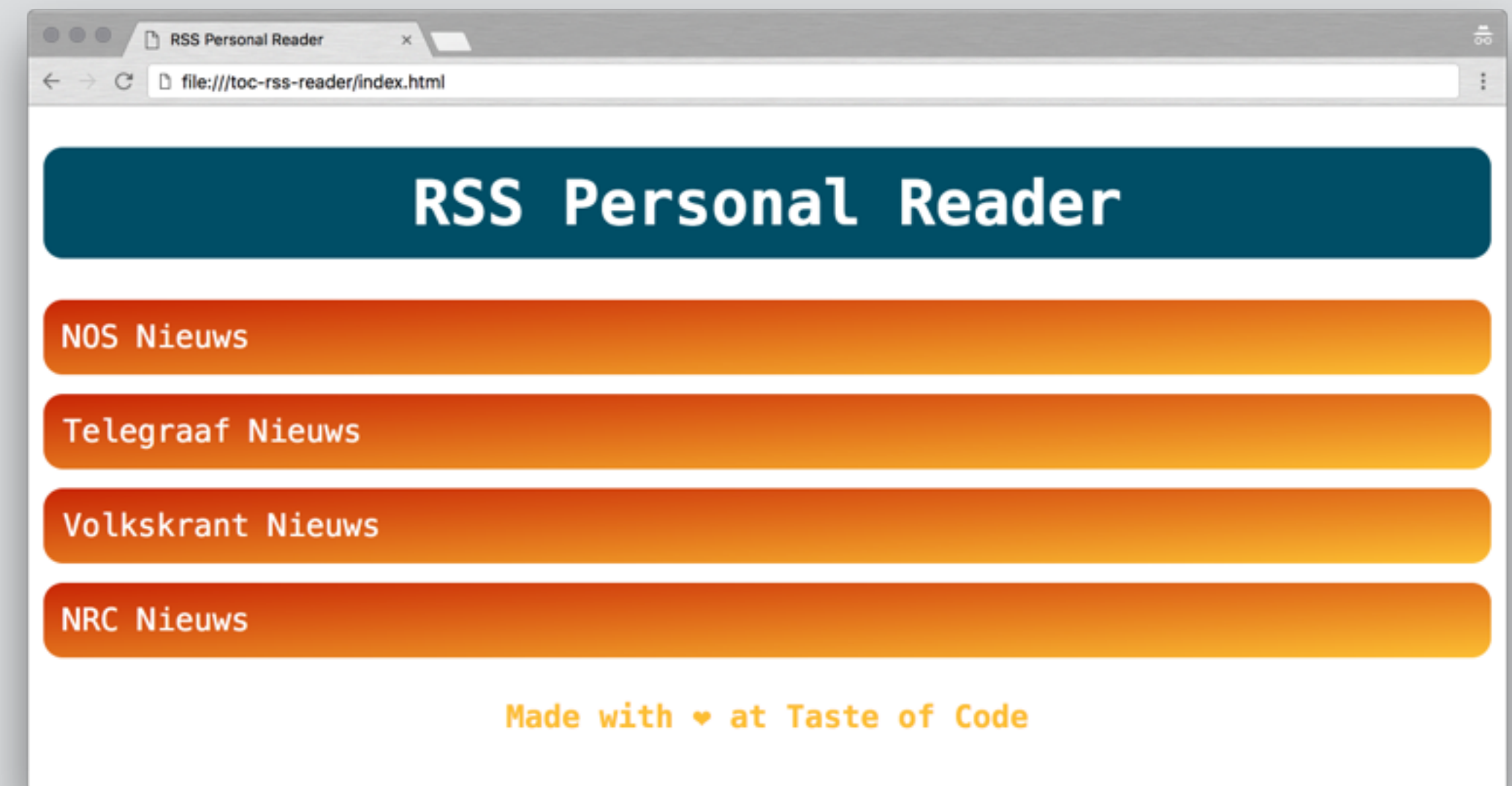
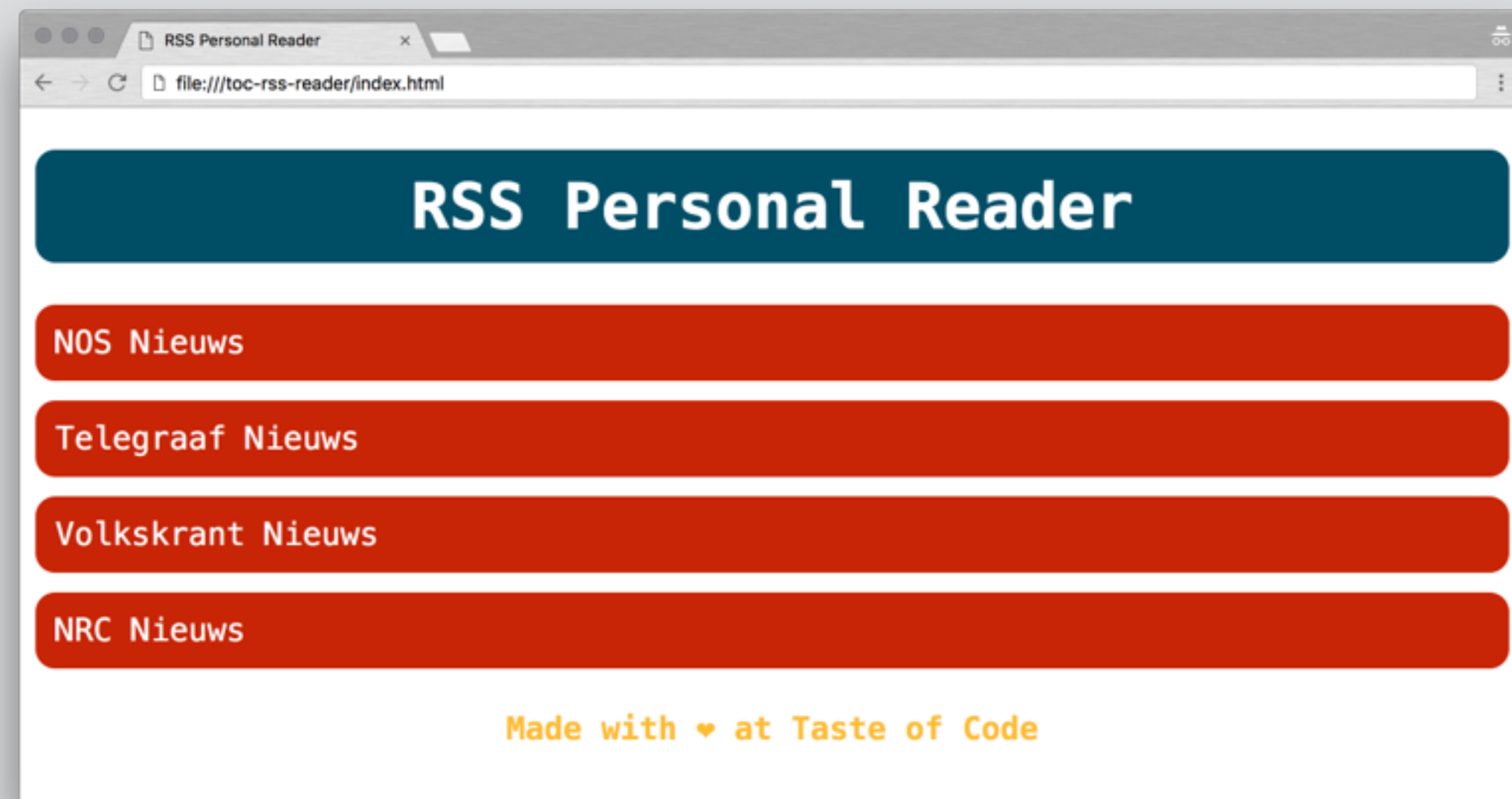
Move your CSS code into its own file.

Apply some margins and paddings to the elements in your web page.

[Go to exercise](#)



Exercise





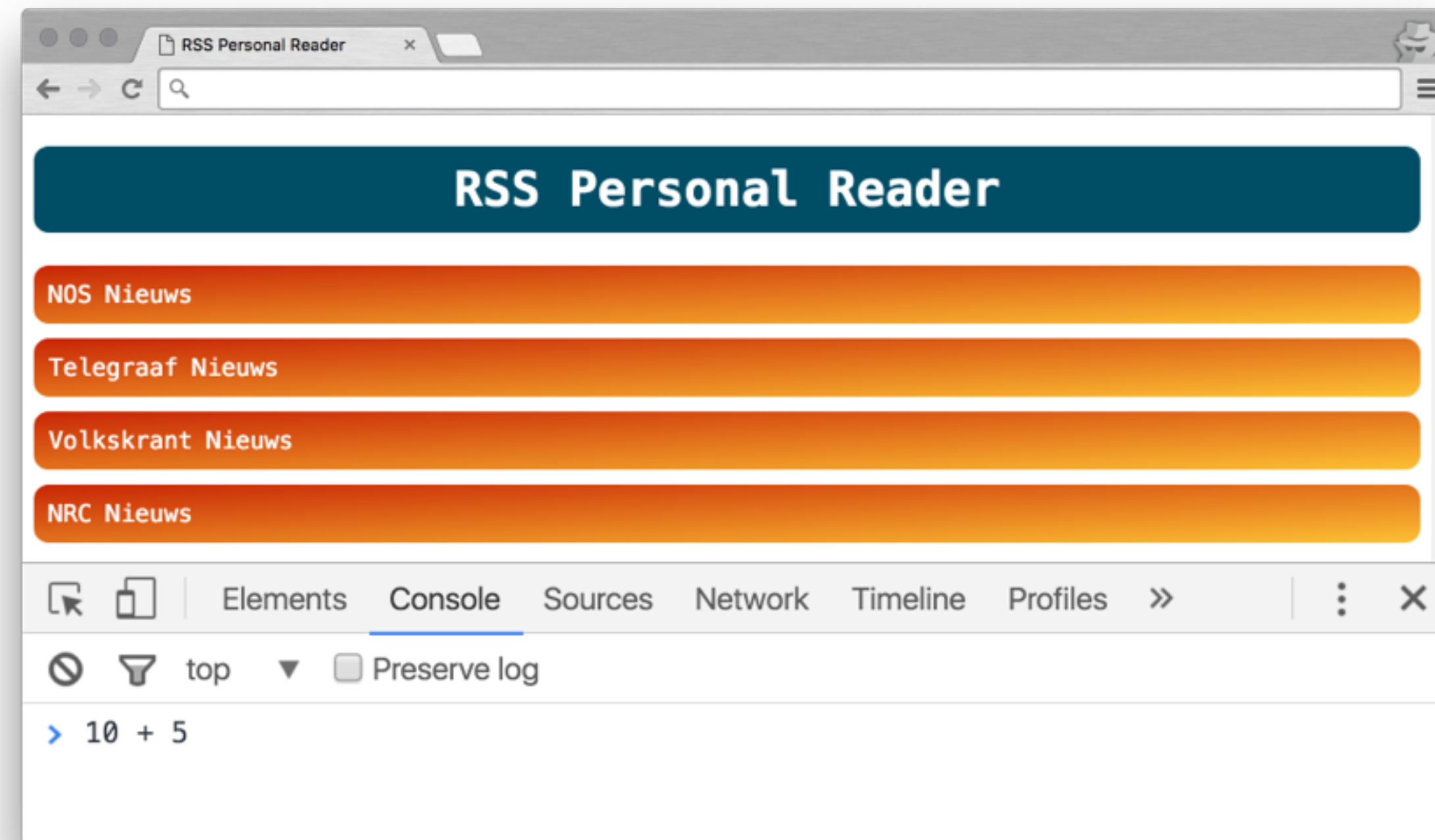
JavaScript

Behaviour layer

This is where we can change **how the page behaves**,
adding **interactivity**.



JavaScript lives in your browser



Open JavaScript Console

CMD + ALT + J (Mac)

Control + Shift + J (Windows/Linux)



Exercise

JavaScript Maths

Open the JavaScript console in your browser
and use it to do some calculations.

[Go to exercise](#)



How JavaScript makes pages more interactive

Access content

select elements that have a ``warning`` class

Modify content

add a paragraph of text after the first `<h1>` 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 a **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");
```



Exercise

Exploring the Console

Open up an alert box.

Inspect web page title and location.

Write some text on the web page.

[Go to exercise](#)



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)
```



Exercise

Storing feed urls in variables

Visit the site NEDERLANDSE RSS FEEDS.

Select some sites to follow.

Store each of the feed urls in a variable.

js/rss.js

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

[Go to exercise](#)



What is a script and how to create one

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.



Creating 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>')
```

Linking to JavaScript file from HTML page

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

Exercise

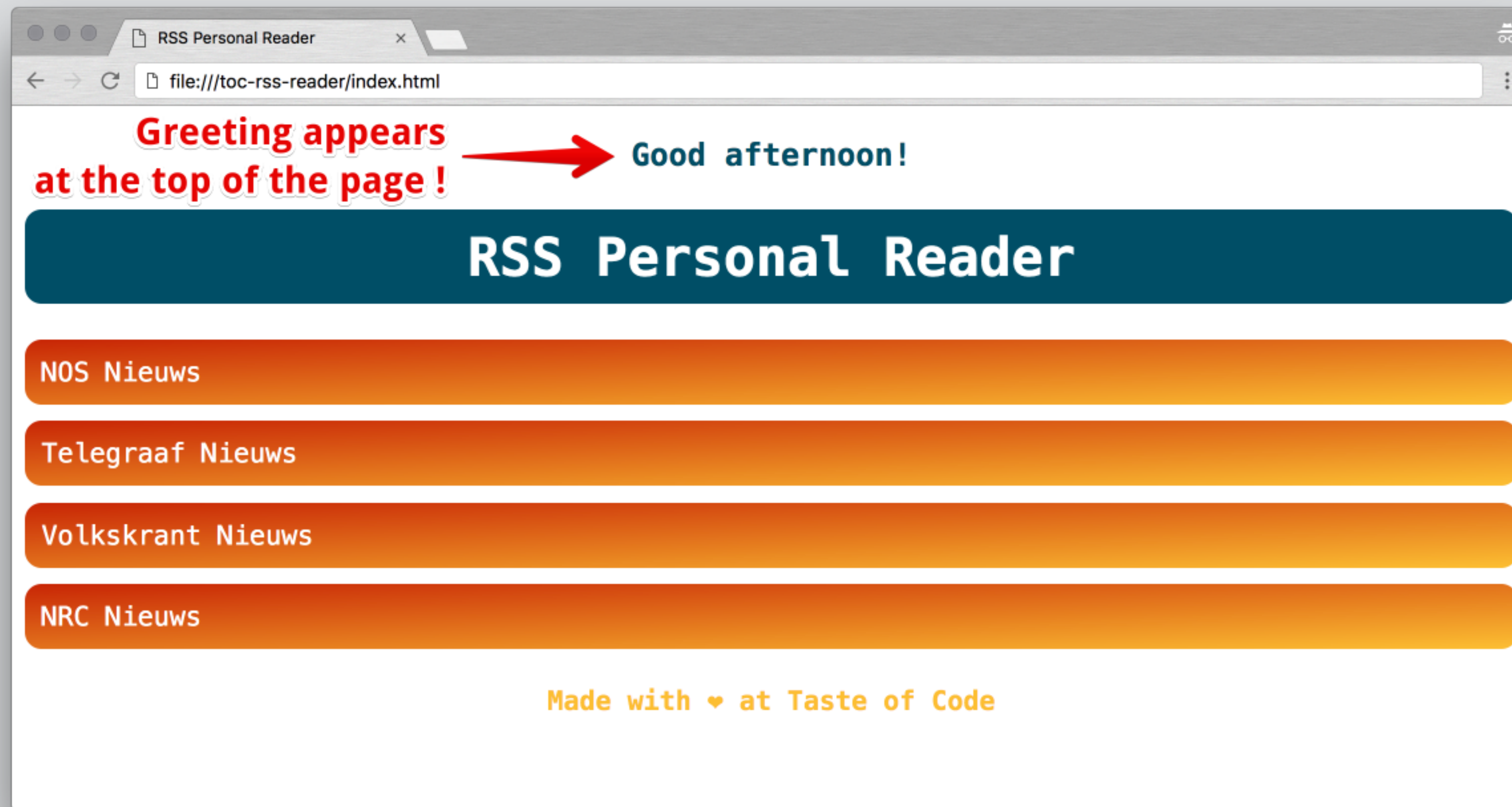
Greeting with JavaScript

Add any message you wish
at the top of your web page
depending on the day time.

[Go to exercise](#)



Exercise



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?

Example of 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



Access & update text / markup with innerHTML

You can select any element in the 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="js/rss.js"></script>
  </body>
</html>
```



Access & update text / markup with innerHTML

js/rss.js

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

var feedsLibrary = {
  nos: nosUrl
};

var ulNos = document.getElementById('nos');

ulNos.innerHTML = '<a href="' + feedsLibrary.nos + '">'
                  + feedsLibrary.nos + '</a>';
```



Exercise

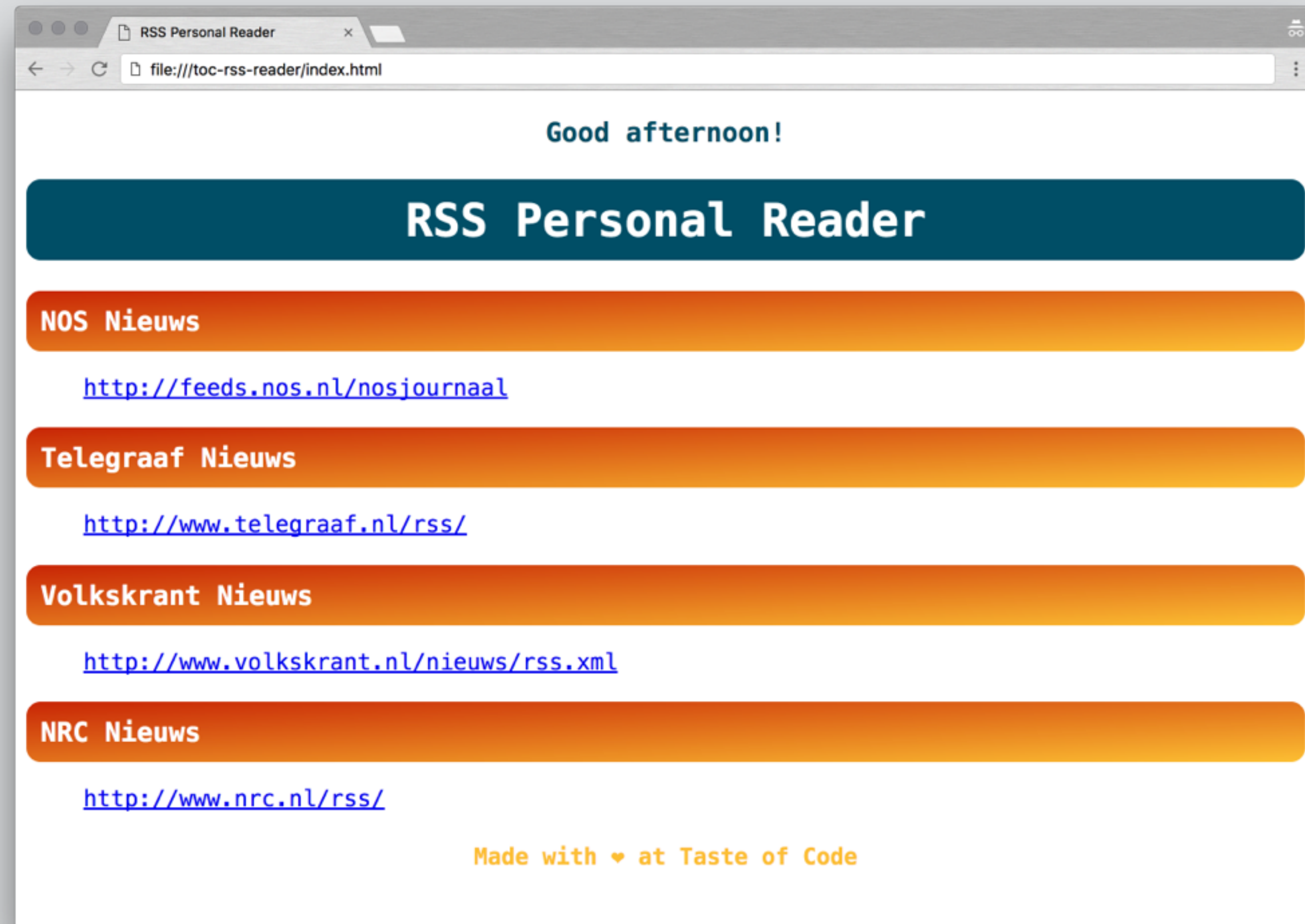
Rendering feed URLs with JavaScript

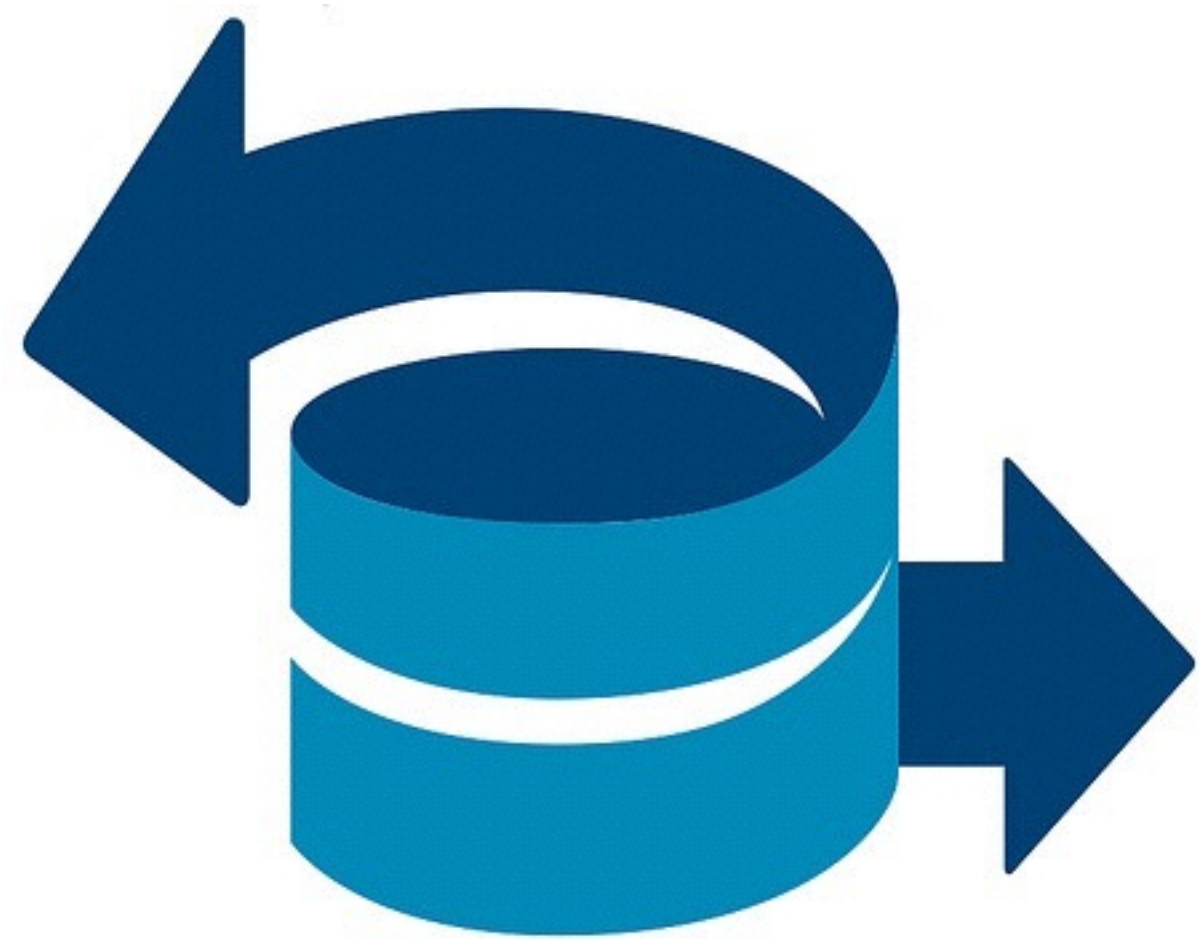
Show each of the feed URLs
stored in the **feedLibrary** object
under its corresponding unordered list.

[Go to exercise](#)



Exercise





YQL

Yahoo Query Language

Query, filter, and combine **data across the web**
through a single interface.

Exploring YQL

Our purpose

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

The screenshot shows the Yahoo! Developer Network (YDN) YQL interface. The browser address bar displays `https://developer.yahoo.com/yql/`. The page header includes the Yahoo! Developer Network logo, a search bar, and navigation links for Products, Events, My Apps, Sign In, and a language selector (US).

The main content area is titled "Sample YQL Response". On the left, a sidebar lists "Products" with "YQL" selected, and other links like "YQL Console", "Create an App", "Get Started", "Documentation", "Forum", and "FAQ".

The "Examples" section on the left lists several tasks, with "Extract HTML using XPath" highlighted. Other examples include "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 "Responses" section on the right shows the "JSON" format selected. The "YQL Query" input field contains the query: `SELECT * FROM feed WHERE url='http://feeds.nos.nl/nosjournaal' LIMIT 5`. A "Test" button is located to the right of the query input. A "YQL Console" link is also present.

The "Response" section displays the JSON output of the query:

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

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



Visiting the endpoint



```
{
  - 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
```



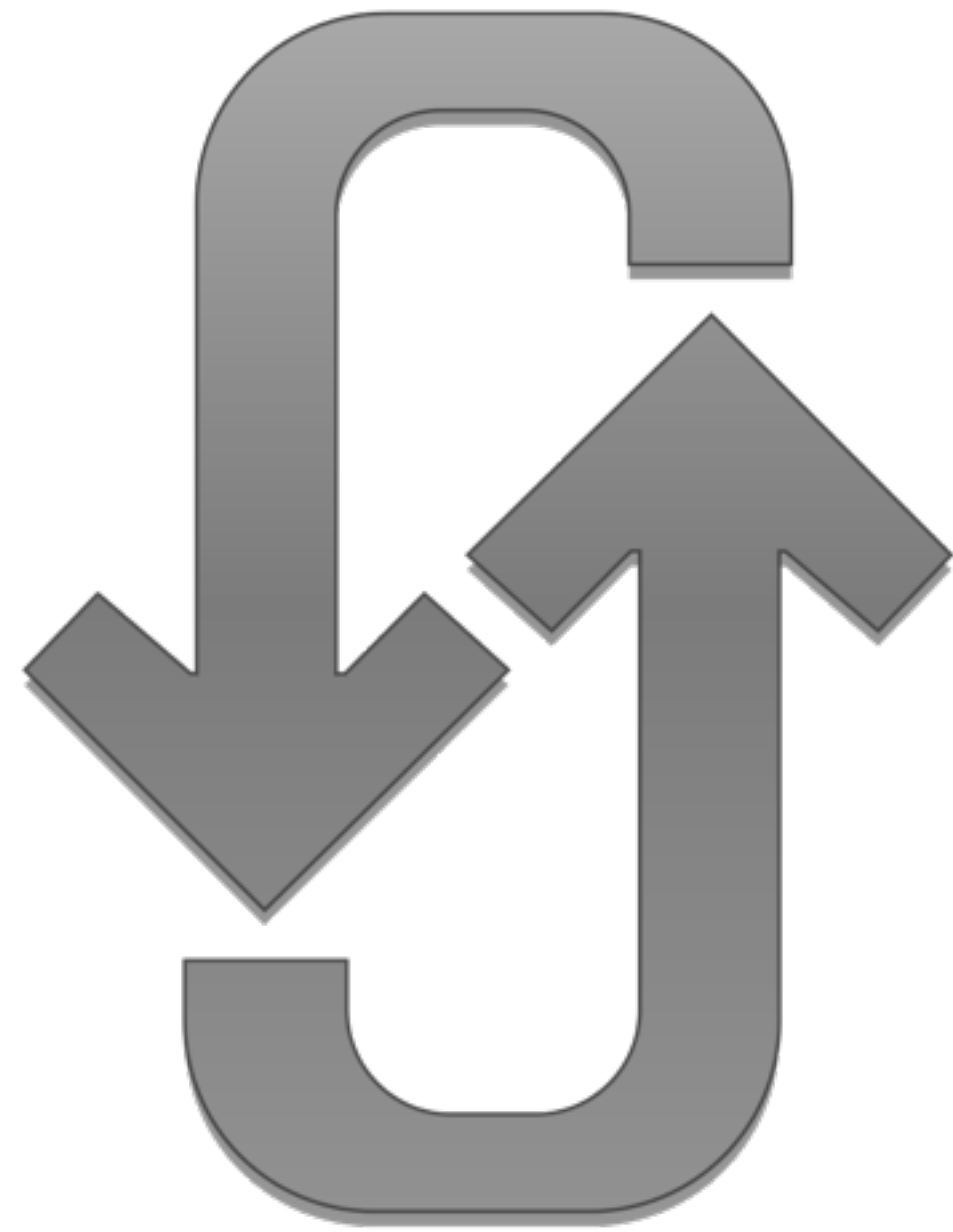
Exercise

Visiting the Endpoints

Check out the different endpoints for the sites that you stored under the **feedsLibrary**.

[Go to exercise](#)



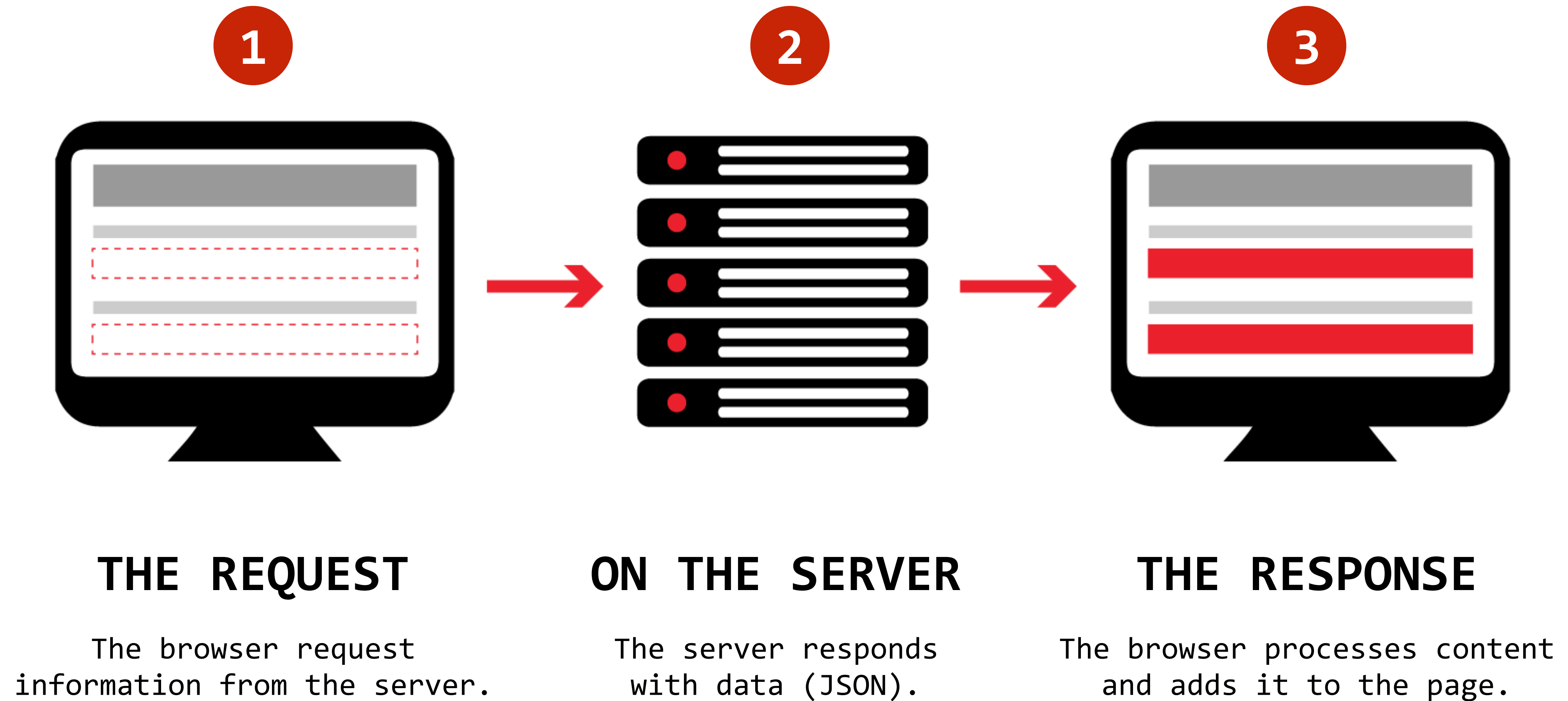


Ajax

Requesting Information

The **browser requests information**
from a web server using Ajax.

How Ajax Works



Include jQuery in your page

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>
    <script src="rss.js"></script>
  </body>
</html>
```



Building the path for the request

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

js/rss.js

```
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);
```



Exercise

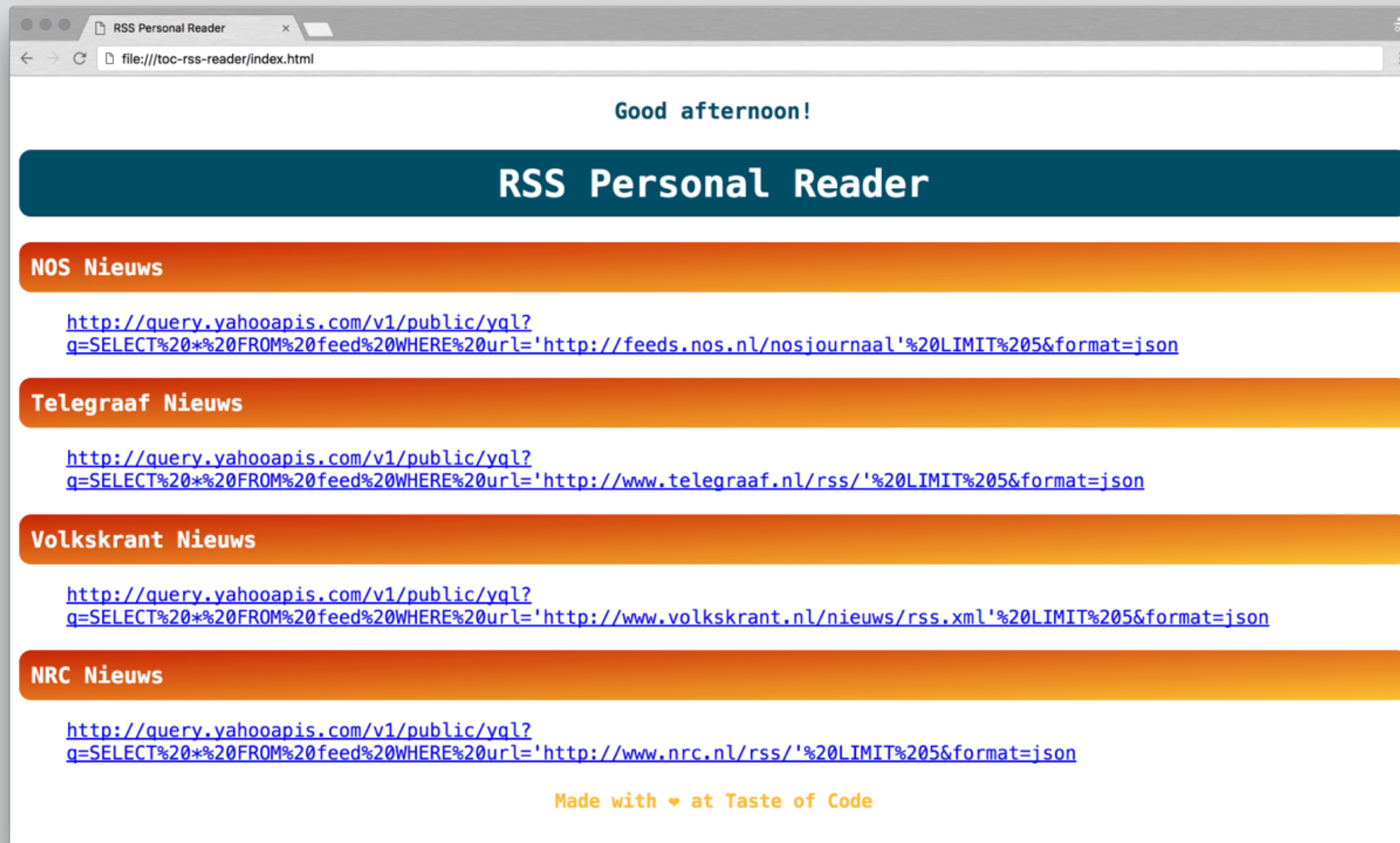
Printing out the rss feeds paths

Apply everything you know of YQL and Ajax to print out the path for each of the sites that you are following in the page.

[Go to exercise](#)



Exercise



Handling Ajax Requests & Responses

jQuery provides several methods that handle Ajax requests.

In this case we will need to load JSON data from a web server. The **.getJSON()** method is the one that helps us doing so.



Handling Ajax Requests & Responses

url = where the data is fetched from
callback = what needs to be done when data is returned

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



callback



Exercise

Printing out the response

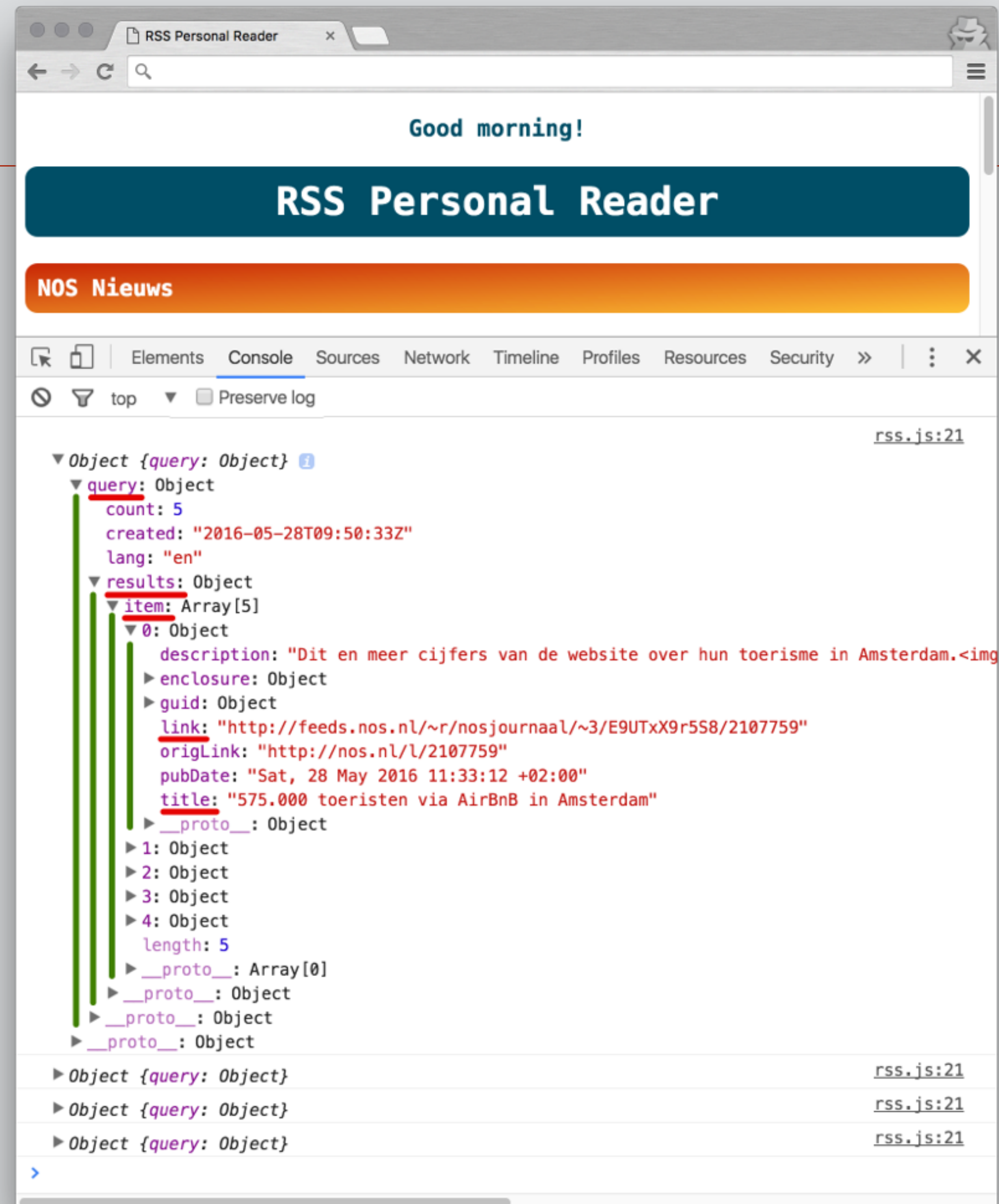
Using the `.getJSON()` method, print out the value of `response.query.results.item` from the response in the JavaScript console for each of the rss feeds paths.

[Go to exercise](#)



Exercise

Print out the value of
`response.query`
`response.query.results`
`response.query.results.item`



The screenshot shows a web browser window titled "RSS Personal Reader". The page content includes a greeting "Good morning!", a dark blue header with the text "RSS Personal Reader", and an orange bar with the text "NOS Nieuws". Below this is a developer console showing the output of a JavaScript script. The console displays a JSON object structure representing an RSS feed. The first log entry is expanded, showing the following structure:

```
Object {query: Object}
  query: Object
    count: 5
    created: "2016-05-28T09:50:33Z"
    lang: "en"
    results: Object
      item: Array[5]
        0: Object
          description: "Dit en meer cijfers van de website over hun toerisme in Amsterdam.<img"
          enclosure: Object
          guid: Object
          link: "http://feeds.nos.nl/~r/nosjournaal/~3/E9UTxX9r5S8/2107759"
          origLink: "http://nos.nl/l/2107759"
          pubDate: "Sat, 28 May 2016 11:33:12 +02:00"
          title: "575.000 toeristen via AirBnB in Amsterdam"
          __proto__: Object
        1: Object
        2: Object
        3: Object
        4: Object
        length: 5
        __proto__: Array[0]
      __proto__: Object
    __proto__: Object
  __proto__: Object
```

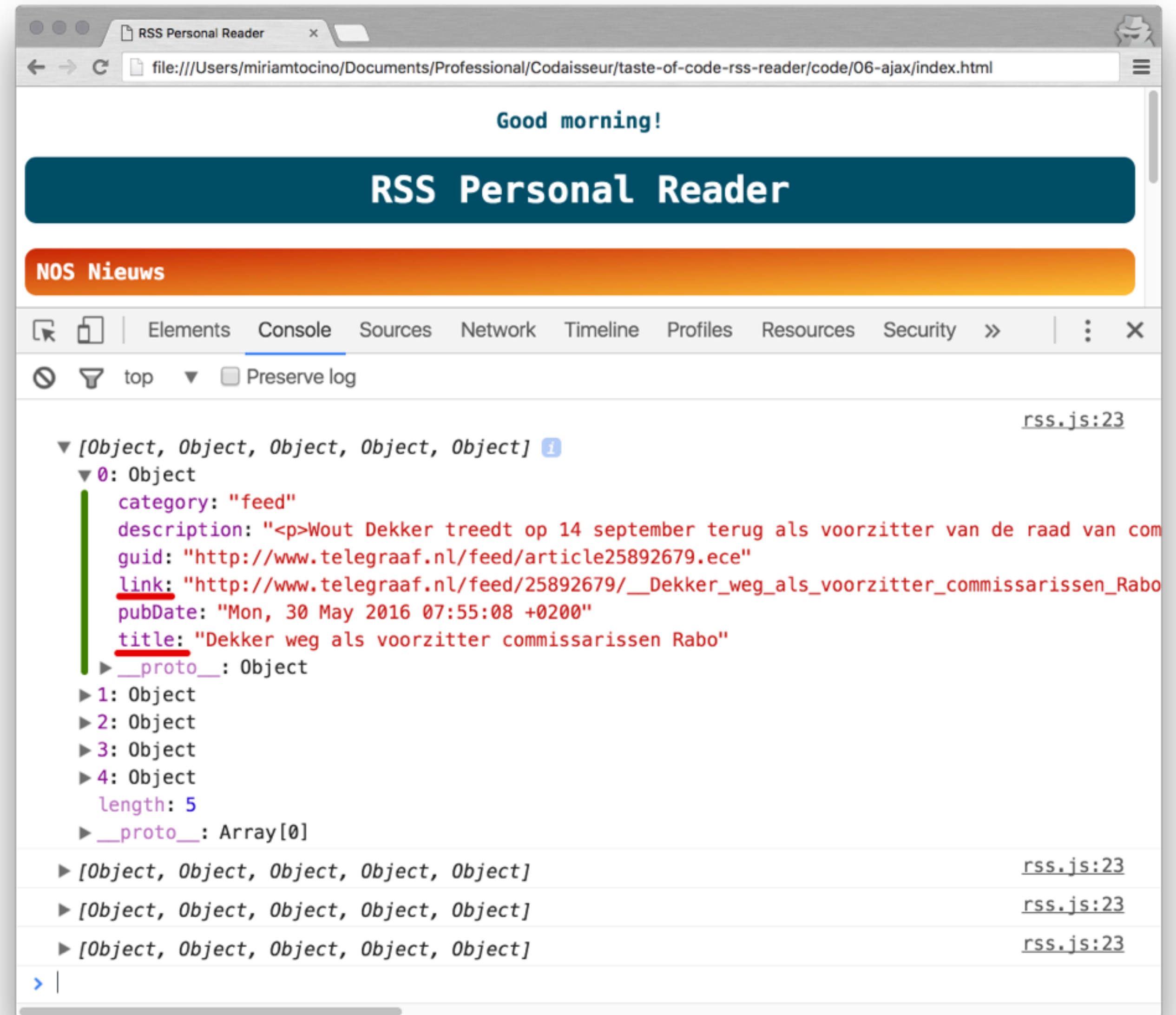
Below this, three more log entries are shown, each representing the same object structure, indicating that the script is logging the response multiple times.



Retrieving articles title and link

`response.query.results.item`
is an array of objects.

Arrays are just another way
of storing data as a
collection.



Retrieving articles title and link

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

js/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] ...`

are called **index**
and represents the element position
within the array



Exercise

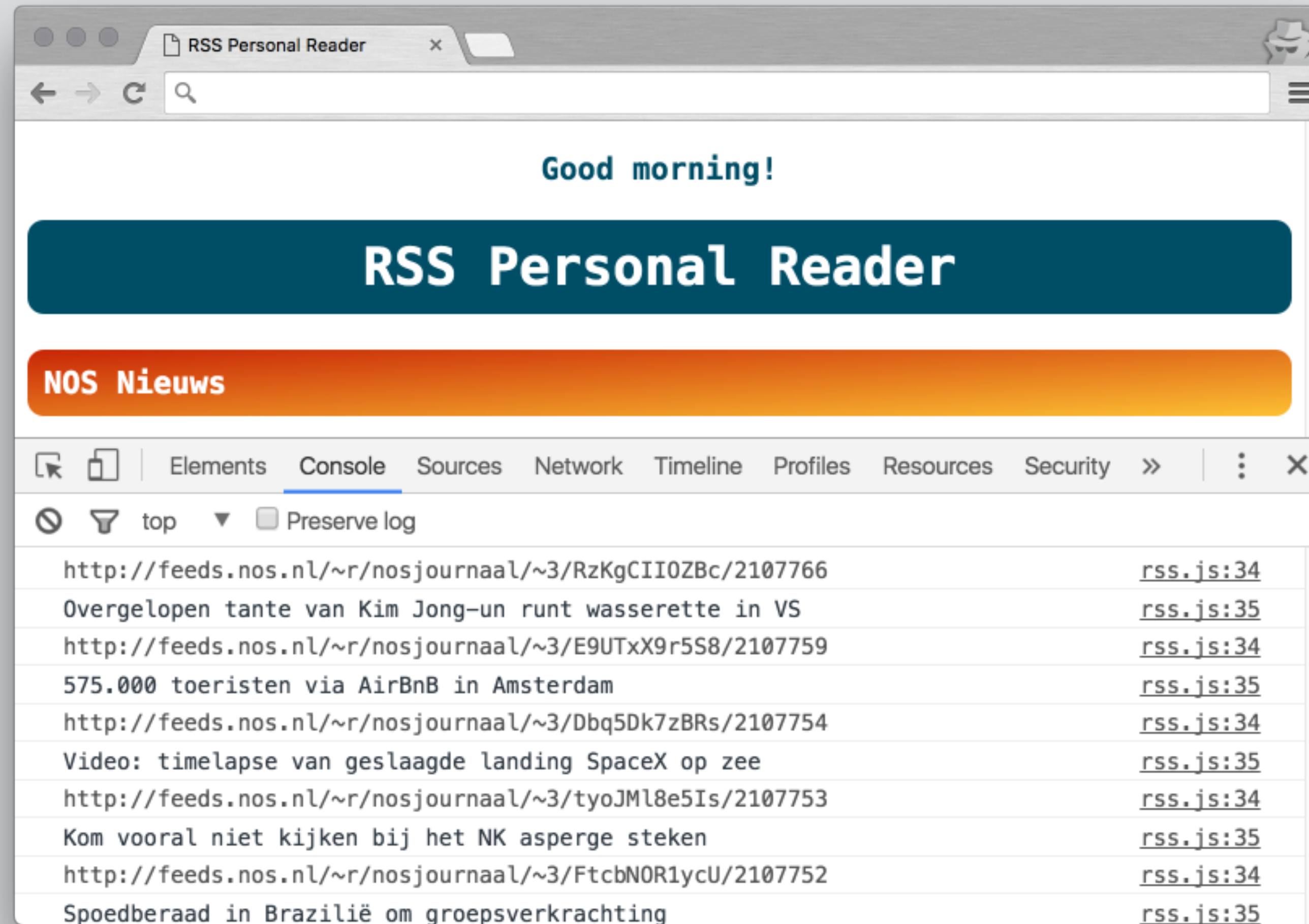
Printing out links and titles

Go ahead and print out each of the articles links and titles in the console.

[Go to exercise](#)



Exercise



Exercise

Rendering links and titles

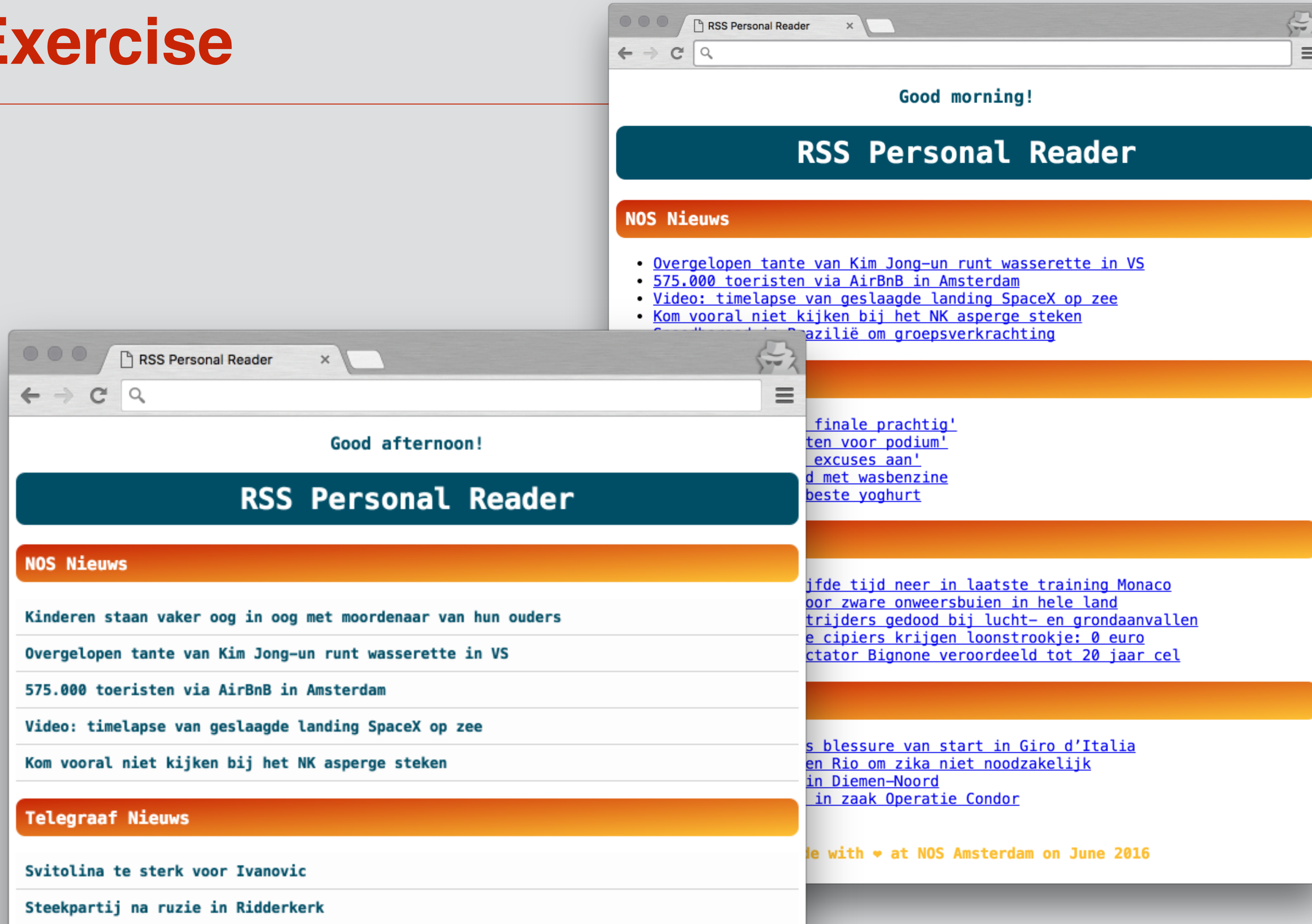
Render a list item including a link to each of the articles for every site site.

Make it so that every link opens in a new tab in your browser.

[Go to exercise](#)



Exercise



Good morning!

RSS Personal Reader

NOS Nieuws

- [Overgelopen tante van Kim Jong-un runt wasserette in VS](#)
- [575.000 toeristen via AirBnB in Amsterdam](#)
- [Video: timelapse van geslaagde landing SpaceX op zee](#)
- [Kom vooral niet kijken bij het NK asperge steken](#)
- [Gedetineerde in Brazilië om groepsverkrachting](#)

Good afternoon!

RSS Personal Reader

NOS Nieuws

- [Kinderen staan vaker oog in oog met moordenaar van hun ouders](#)
- [Overgelopen tante van Kim Jong-un runt wasserette in VS](#)
- [575.000 toeristen via AirBnB in Amsterdam](#)
- [Video: timelapse van geslaagde landing SpaceX op zee](#)
- [Kom vooral niet kijken bij het NK asperge steken](#)

Telegraaf Nieuws

- [Svitolina te sterk voor Ivanovic](#)
- [Steekpartij na ruzie in Ridderkerk](#)

le with ♥ at NOS Amsterdam on June 2016

