THE UNIVERSITY
of ADELAIDE

Web and Database Computing

adelaide.edu.au

Introduction to JavaScript: Client-Side JavaScript Fundamentals

Previously in WDC

We looked at how to build webpages with HTML & CSS

- Basic Syntax & structure.
- Style and Layout.

Static Web Pages

- They can't do much
  - Display content
- They can look nice
  - CSS
  - Media
  - Animations
- **They're Boring**

"A static website contains Web pages with fixed content. Each page is coded in HTML and displays the same information to every visitor"
- The Tech Terms Computer Dictionary

Dynamic Web Pages

- We can make our pages more useful by changing content as needed for the user.
- There are lots of reasons why we might want a web page to be dynamic
    - Different content depending on country of client (different languages)
    - Different content depending on browser, operating system, etc.
    - Shopping carts
    - Customised information
    - Respond to events e.g. mouse over menu, clicks, etc.

How do we make web pages dynamic?

- Server Side
    - Generate custom web page on a web server (in response to client actions) and send to client.
- Client Side
    - Execute code in client’s Browser
- We’re going to focus on **Client Side** this week

Why Client Side?

Why Client Side?

- Complex websites can put heavy load on the webserver providing them.
    - Client side code execution can reduce this load by moving it to the client
    - Clients have become much more powerful and are able to handle more complex tasks.
- Client side code can also reduce page loading times
    - Allow our pages to do interesting things without loading a new one.
    - Some tasks can be done without needing to communicate with the server.
- Better User Experience
    - Improved responsiveness for end user.
    - We will look at this more when we cover AJAX

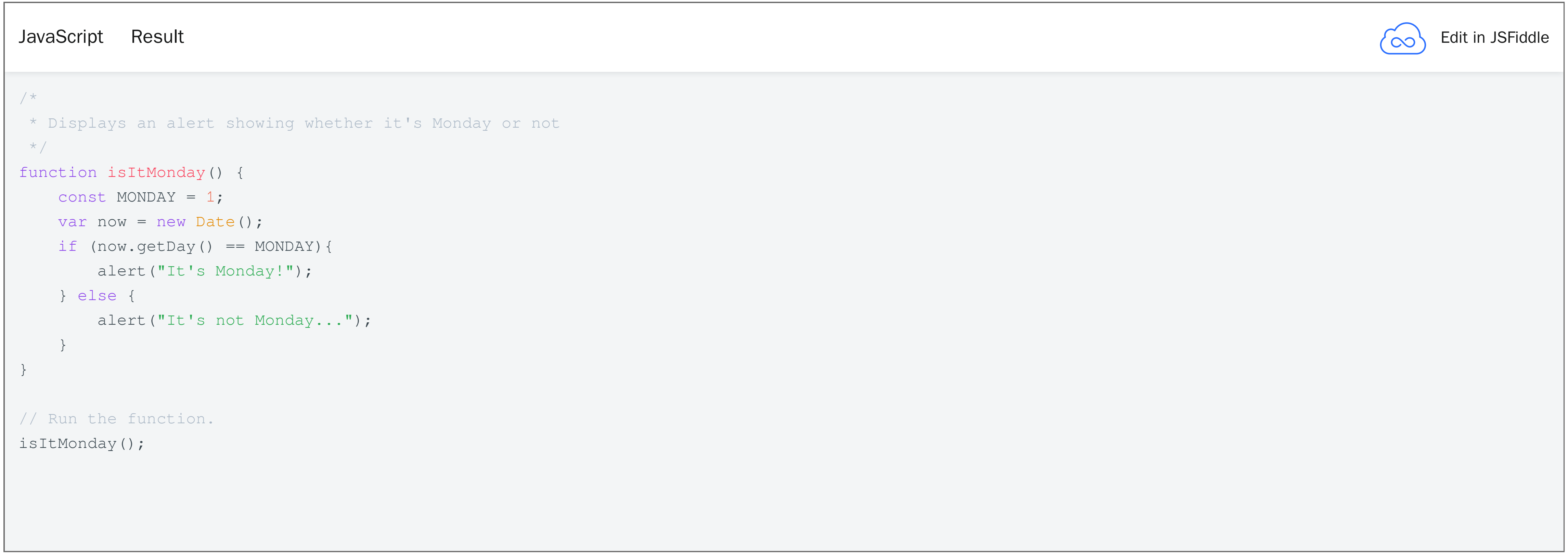
Enter ECMAScript

(aka JavaScript)

JavaScript is

- Dynamic
  - Code is interpreted as it is executed
- Weakly Typed
  - Types are associated with individual objects.
  - Can store ANY type in ANY variable/const.
  - Can pass ANY variable to ANY function (that accepts parameters)
- Object Oriented
- Event Driven
  - Concurrency can result in unexpected behaviour

An example:



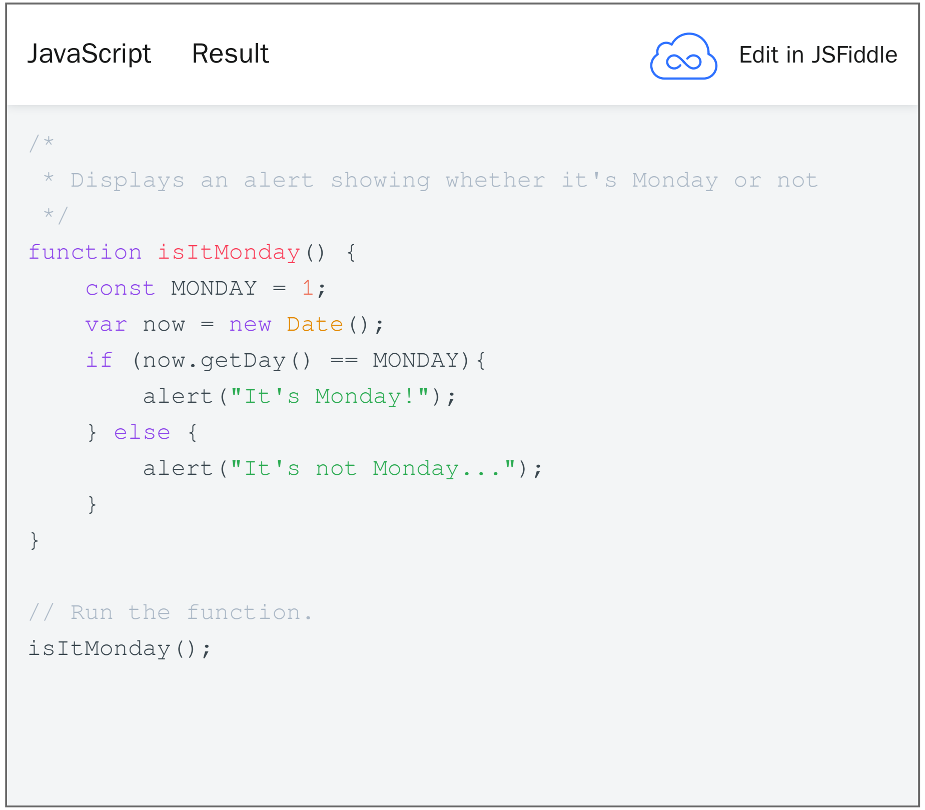
```
/*
 * Displays an alert showing whether it's Monday or not
 */
function isItMonday() {
    const MONDAY = 1;
    var now = new Date();
    if (now.getDay() == MONDAY){
        alert("It's Monday!");
    } else {
        alert("It's not Monday...");
    }
}

// Run the function.
isItMonday();
```

https://jsfiddle.net/ian_knight_uofa/51L0wq9u/4/

Syntax

C-Style Syntax:

- Lines end in `;`
- Code blocks enclosed in `{ }`
- Comments:
  - Single line comments start with `//`
  - Multiline comments wrapped in `/* */`
- Variables:
  - Variables are not typed, so anything can be passed anywhere
  - Scope depends on declaration
  - Undeclared variables scoped at **global** level
  - Declared using `var` scoped at **function** level
  - Declared using `let` scoped at **block** level
- Constants:
  - Declared using `const`, scoped at **function** level



```
/*
 * Displays an alert showing whether it's Monday or not
 */
function isItMonday() {
    const MONDAY = 1;
    var now = new Date();
    if (now.getDay() == MONDAY){
        alert("It's Monday!");
    } else {
        alert("It's not Monday...");
    }
}

// Run the function.
isItMonday();
```

Basic Control Structures

Conditionals (`if`/`else if`/`else`)

```
if (condition) {
  // Do something
} else if (condition) {
  // Do something else
} else {
  // Otherwise do this
}
```

`for` Loop

```
for (let i=0; i<5; i++) {
  // Do something
}
```

`while` Loop

```
while (condition) {
  // Do something
}
```

Defining Functions

Basic Function Definition

```
function addFunction(param1, param2) {
  var value = param1 + param2;
  return value;
}
```

Anonymous Function Definition

```
var addFunction = function(param1, param2) {
  var value = param1 + param2;
  return value;
}
```

ES6 Arrow Function Definition

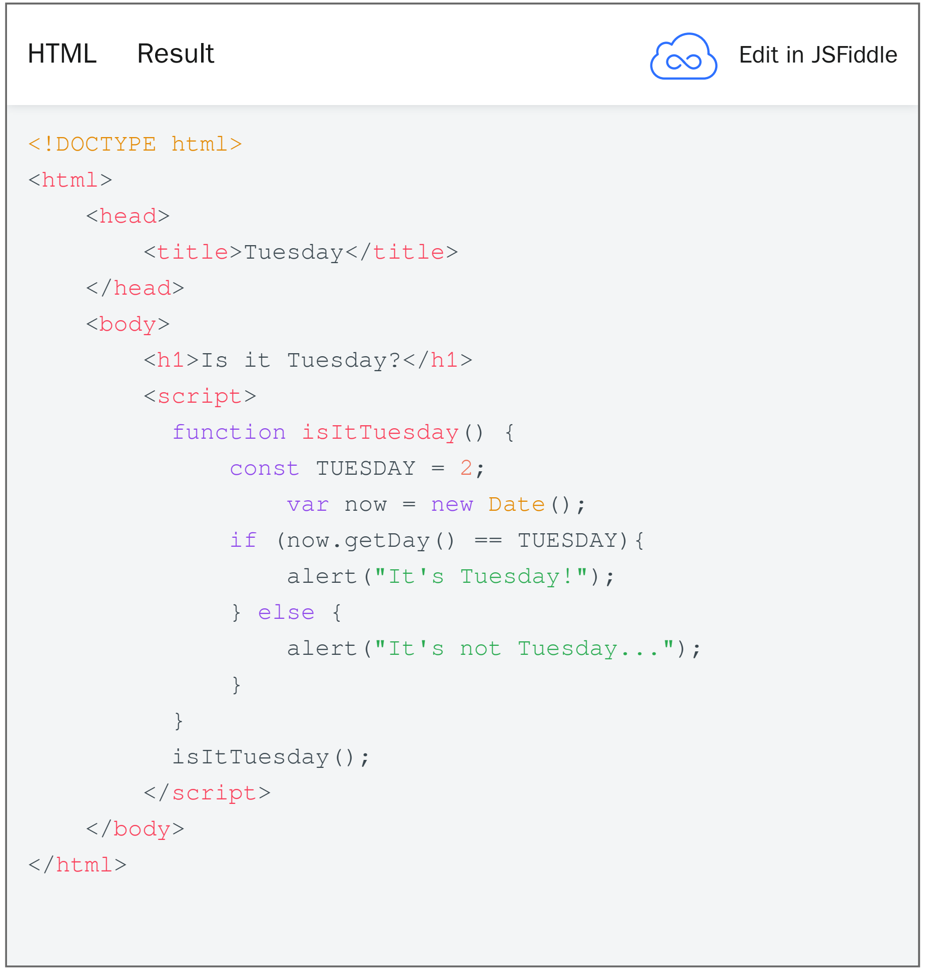
```
var addFunction = (param1, param2) => {
  var value = param1 + param2;
  return value;
}
```

Running our code

We use the `script` tag to run javascipt code in our webpages.

Scripts can be inline with the page content:

```
<script>
    // Some javascript...
</script>
```



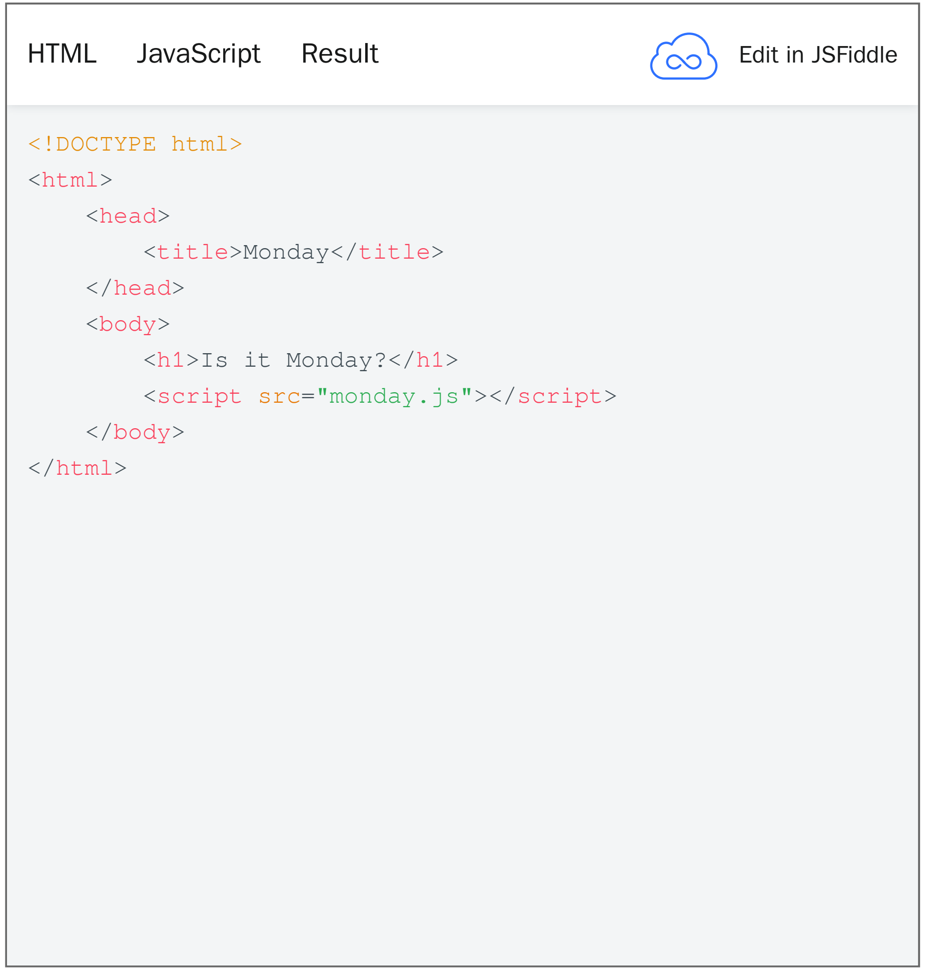
```
<!DOCTYPE html>
<html>
  <head>
    <title>Tuesday</title>
  </head>
  <body>
    <h1>Is it Tuesday?</h1>
    <script>
     function isItTuesday() {
       const TUESDAY = 2;
         var now = new Date();
       if (now.getDay() == TUESDAY){
         alert("It's Tuesday!");
       } else {
         alert("It's not Tuesday...");
       }
     }
     isItTuesday();
    </script>
  </body>
</html>
```

Running our example

or scripts can be loaded from external files:

```
<script src="file.js"></script>
```

(similar to CSS)



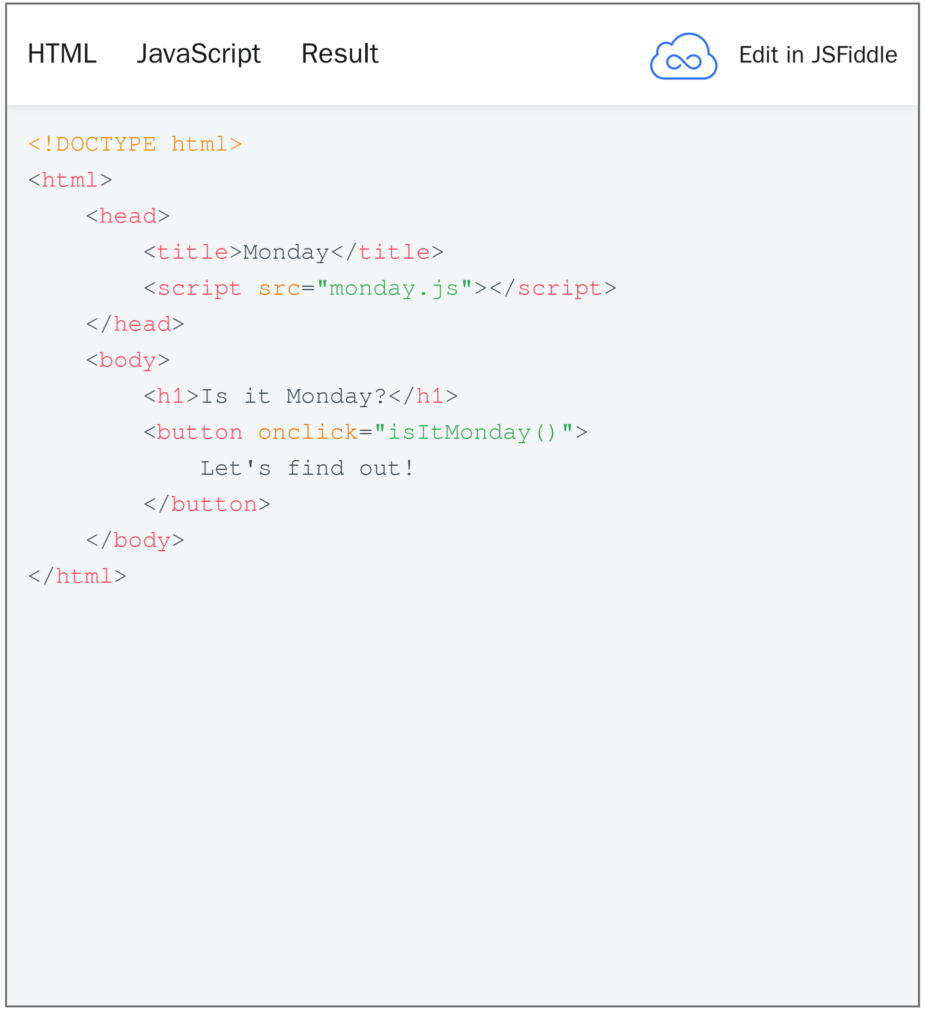
```
<!DOCTYPE html>
<html>
    <head>
        <title>Monday</title>
    </head>
    <body>
        <h1>Is it Monday?</h1>
        <script src="monday.js"></script>
    </body>
</html>
```

Running our example

code can be run using event attributes:

```
<button onclick="alert('Hello')">
    Let's find out!
</button>
```

or event attributes can be used to call parts of our scripts ▯

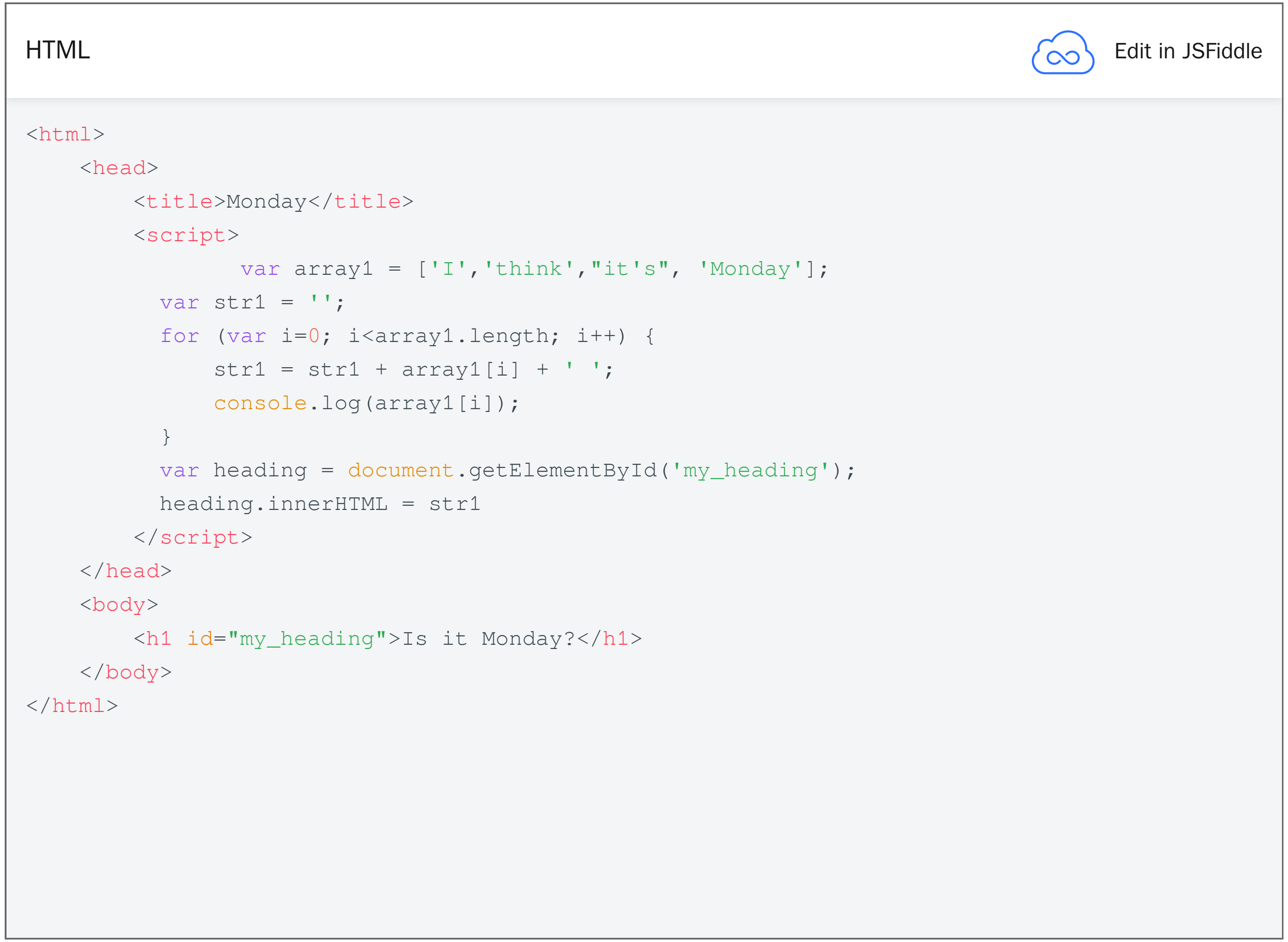


```
<!DOCTYPE html>
<html>
    <head>
        <title>Monday</title>
        <script src="monday.js"></script>
    </head>
    <body>
        <h1>Is it Monday?</h1>
        <button onclick="isItMonday()">
            Let's find out!
        </button>
    </body>
</html>
```

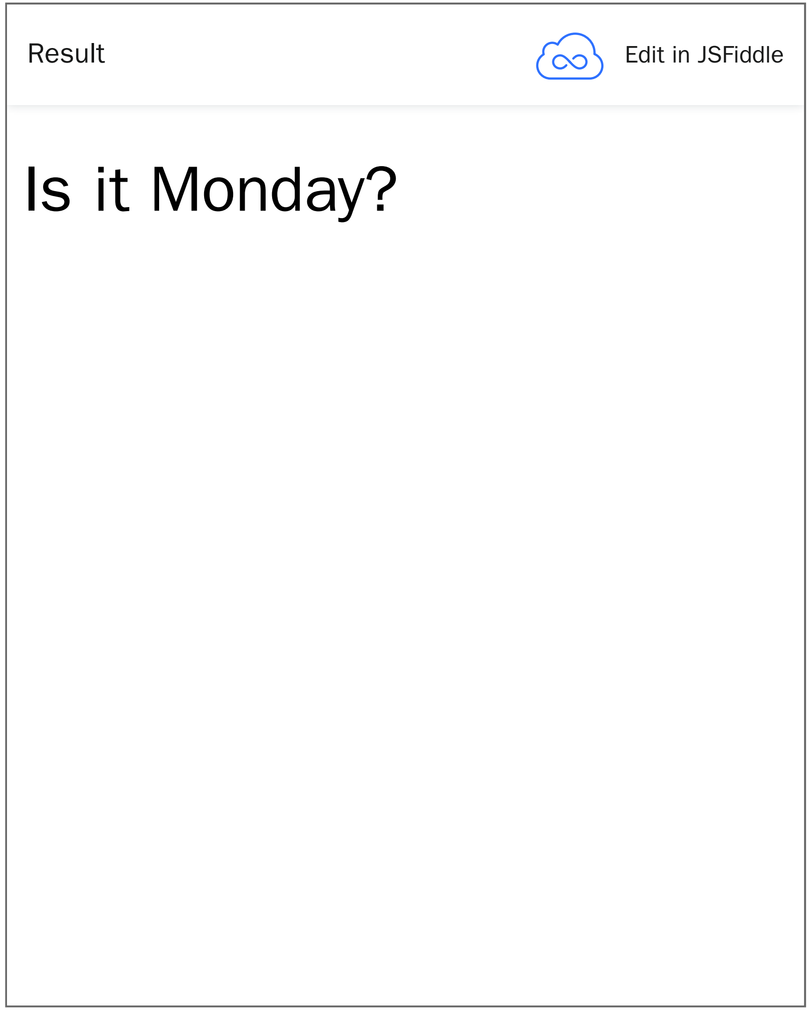
Take care!

Where you load your script matters

Another example (but it doesn't work)



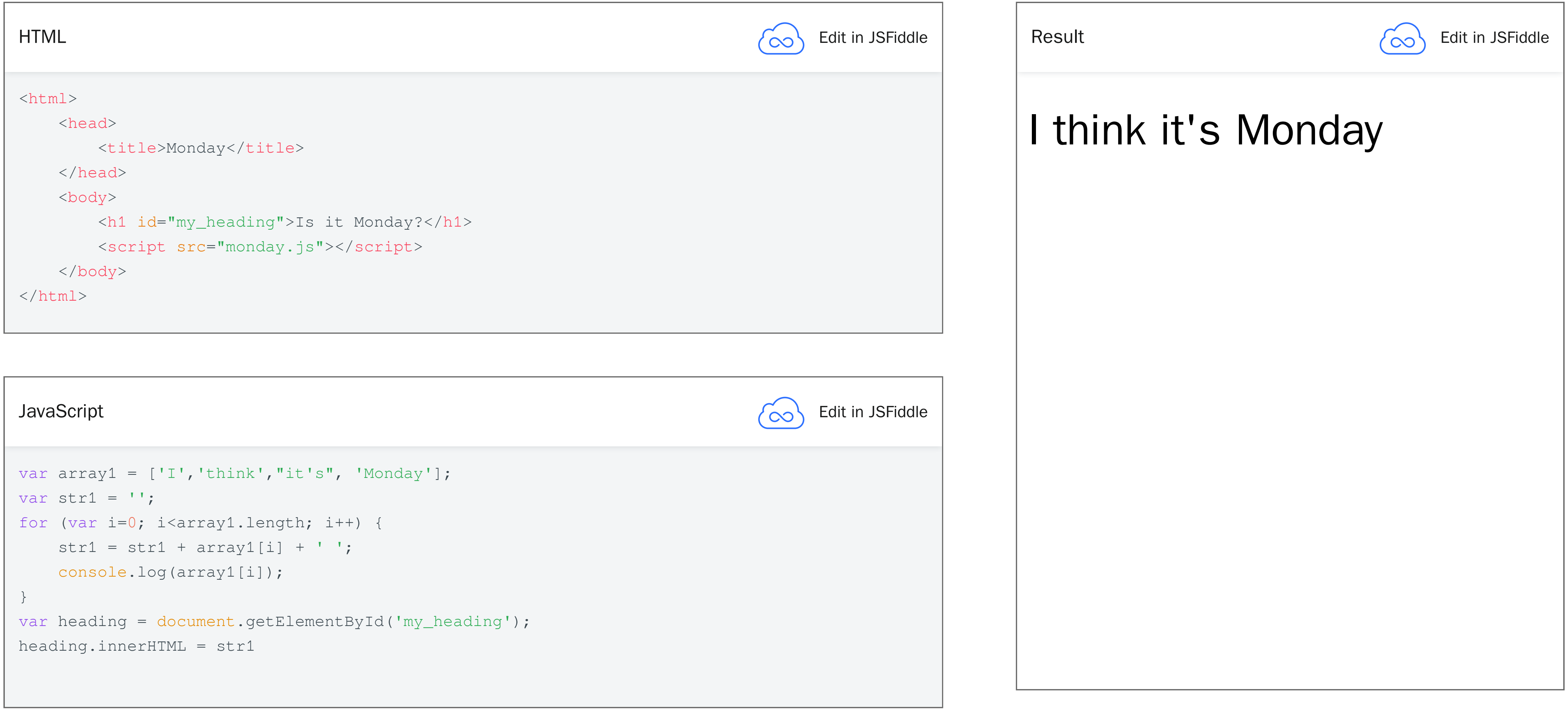
```
<html>
    <head>
        <title>Monday</title>
        <script>
                var array1 = ['I','think',"it's", 'Monday'];
          var str1 = '';
          for (var i=0; i<array1.length; i++) {
              str1 = str1 + array1[i] + ' ';
              console.log(array1[i]);
          }
          var heading = document.getElementById('my_heading');
          heading.innerHTML = str1
        </script>
    </head>
    <body>
        <h1 id="my_heading">Is it Monday?</h1>
    </body>
</html>
```



Is it Monday?

https://jsfiddle.net/ian_knight_uofa/g5ct3kLh/

Another example (that's better)



HTML

```
<html>
    <head>
        <title>Monday</title>
    </head>
    <body>
        <h1 id="my_heading">Is it Monday?</h1>
        <script src="monday.js"></script>
    </body>
</html>
```

JavaScript

```
var array1 = ['I','think',"it's", 'Monday'];
var str1 = '';
for (var i=0; i<array1.length; i++) {
    str1 = str1 + array1[i] + ' ';
    console.log(array1[i]);
}
var heading = document.getElementById('my_heading');
heading.innerHTML = str1
```

Result

I think it's Monday

https://jsfiddle.net/ian_knight_uofa/sqgf0y87/4/

SUB CRUCE LUMEN

THE UNIVERSITY of ADELAIDE