

jQuery



jQuery or JS?



- **Fast and concise** JavaScript library
- Simplifies
 - HTML document traversing
 - Event handling
 - Animations
- Some instructions start with “\$”

jQuery - Files

Needs to be imported in the **<head>** tag

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"  
integrity="sha256-/xUj+3OJU5yExlq6GSYGGSHk7tPXikynS7ogEvDej/m4="  
crossorigin="anonymous"></script>
```

It should go **before** the **<script>** tag, and before any other library that uses jQuery.

jQuery - DOM traversing

- ready and onload

```
$( function() { ... })  
$.on( "ready", function() { ... })  
$( document ).on("ready" function() { ... })  
$( "img" ).first().on("ready", function() { ... })  
  
//onload: when everything is loaded (img, iframes, etc)  
$( window ).on( "load", function() { ... })
```

jQuery - DOM traversing

- Shorter functions, more intuitive

```
// getElementById is now just $( "#name" )  
// .css( "property" , "value" )  
$( "#container" ).css( "color" , "blue" );  
  
// getElementsByTag is $( "tag" ), and getElementsByClass is $(  
".class" )  
// .forEach(function()): loop each of the elements in an array  
$( "p" ).forEach( function() {  
    $( "this" ).parent().addClass( "text" );  
});
```

jQuery - Event handling

```
$( "#divMouse" ).on( "mouseenter mouseleave", function() {  
    console.log( "mouse hovered over or left a div" );  
});  
  
/*****/  
$( "#divMouse" ).on({  
    mouseenter: function() {  
        console.log( "hovered over a div" );  
    },  
    mouseleave: function() {  
        console.log( "mouse left a div" );  
    },  
    click: function() {  
        console.log( "clicked on a div" );  
    }  
});
```

jQuery - Event handling

```
$( "button" ).forEach(function() {  
    $("this").click(function() {  
        console.log( "yay you clicked me" );  
    });  
});  
  
$( "#name-input" ).change(function() {  
    console.log( "focus, change, lost focus" );  
});  
  
$( "#name-input" ).keydown(function() {  
    console.log( "each time a key is pressed inside this input" );  
});  
  
$( ".first-fieldset" ).first().hover(function() {  
    console.log( "this are not the droids you were looking for" );  
});
```

jQuery - Animations

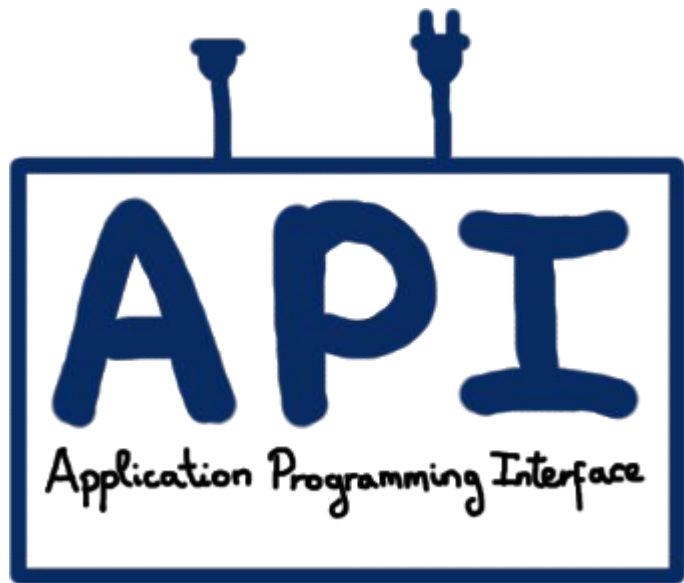
JS: using **setInterval** function

jQuery: you set the result in the animate method

```
$( "#block" ).animate({  
    width: "70%",  
    opacity: 0.4,  
    fontSize: "3em",  
    borderWidth: "10px"  
}, 1500 );  
  
/*****/  
  
$( ".block" ).animate( { "left": "+=50px" }, "slow" );
```


jQuery and CSS animations





AXIOS

AXIOS?

- **HTTP Client**
- To **send and retrieve data** from a server **asynchronously** (in the background), without interfering with the display and behaviour of the webpage

API

- Application Programming Interface
- Functions and procedures that allow, using applications, the **communication between services** like web servers, OS, etc.
- Gives access to **features and data**

API

<https://www.pokeapi.co/api/v2/pokemon/549/>

<https://swapi.co/api/planets/1/>

<https://swapi.co/api/people/1/?format=json>

<http://steamspy.com/api.php?request=top100in2weeks>

<http://steamspy.com/api.php?request=genre&genre=Early+Access>

JSON

- JavaScript Object Notation
- Syntax for storing and send/receive data
- Text, written in JavaScript Object notation
 - Data is name/value pairs, separated by commas
 - Curly braces ({}) hold objects
 - Square braces ([]) hold arrays

```
[{"employees": [  
  { "firstName": "John", "lastName": "Doe" },  
  { "firstName": "Anna", "lastName": "Smith" },  
  { "firstName": "Peter", "lastName": "Jones" }  
]]
```

JSON

- Convert string to object

```
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';  
var myObj = JSON.parse(myJSON);
```

- Convert object to string

```
var myObj = { "name":"John", "age":31, "city":"New York" };  
var myJSON = JSON.stringify(myObj);
```

HTTP

- **Hypertext Transfer Protocol:** to enable communication between client and server. It uses name/value pairs.
- **Methods**
 - GET : parameters in the URL

```
https://www.mywebsite.com/main_form.php?name1=value1&name2=value2
```

- POST: parameters are hidden (forms, etc.)

```
/** Package sent when submitting a form**/  
POST /main_form.php HTTP/1.1  
Host: mywebsite.com  
name1=value1&name2=value2
```


AXIOS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Axios in JavaScript</title>
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
    <script src="script.js"></script>
  </head>
  <body>
    <div>
      <h1>Users</h1>
      <ul></ul>
    </div>
  </body>
</html>
```

AXIOS - GET

```
axios.get('/api/employees')  
  .then( response => {  
    console.log(response.data)  
  } )  
  .catch( err => {  
    console.log(err)  
  } )
```

```
try {  
  const response = axios.get('/api/employees')  
  console.log(response.data)  
} catch (error) {  
  console.log(error)  
}
```

AXIOS - POST

```
axios.post('/api/employees', { employee } )  
  .then( response => {  
    console.log(response.data)  
  } )  
  .catch( err => {  
    console.log(err)  
  } )
```

AXIOS - PUT/PATCH

```
axios.put(`/api/employees/${id}`, { employee })  
  .then( response => {  
    console.log(response.data)  
  } )  
  .catch( err => {  
    console.log(err)  
  } )
```

AXIOS - DELETE

```
axios.delete(`/api/employees/${id}`)  
  .then( response => {  
    console.log(response.data)  
  } )  
  .catch( err => {  
    console.log(err)  
  } )
```