

Year 12 Information Technology

IT Major Project

Student ID: 8401271

Teacher: Mr Segaran

---

Pokédex Project

Web Site Design

Major Project

Technical Manual

## Contents

Pokédex Project Technical Manual .....	4
Authors: .....	4
Version history .....	4
1.0: .....	4
1.1: .....	4
1.2: .....	4
Copy right and acknowledgment .....	4
Functions list .....	5
Search bar function .....	5
Overview .....	5
The code .....	5
Known Bugs and errors .....	6
Sort alphabetically function .....	7
Overview .....	7
The code .....	7
Bugs and errors .....	8
Sort reverse alphabetically function .....	9
Overview .....	9
The code .....	9
Bugs and errors .....	10
Sort numerically function .....	11
Overview .....	11
The code .....	11
Bugs and errors .....	12
Sort reverse numerically function .....	13
Overview .....	13
The code .....	13
Bugs and errors .....	14
Dropdown menu functions .....	15
Overview .....	15
The code .....	15
Bugs and errors .....	16

Redirect functions.....	16
Overview .....	16
The code.....	16
Bugs and errors .....	16
Site map .....	17
Data (class) dictionary.....	18
Update procedure.....	18
System requirements.....	18
Known bugs & errors .....	<b>Error! Bookmark not defined.</b>

# Pokédex Project Technical Manual

## Authors:

Thomas Withers

## Version history

1.0:

Home page complete set up, with search function enabled and sort options.

1.1:

First 15 Pokémon added, fixed search function bug, home page redesigned.

1.2:

Fixed search function drop down bug, Pokémon pages updated, added more options feature.

## Copy right and acknowledgment

This report includes references and information about the Pokémon franchise series including terminology, images, and game information. This all the intellectual property of Nintendo and GAMEFREAK and is protected by copyrights and trademarks. The use of this intellectual property is for a fan reference and is covered by fair use and that the report will be significantly impaired without this information.

## Functions list

### Search bar function

#### Overview

This function takes what a user types in and searches through the list of Pokémon to check and see if anything matches the types result, it is not case sensitive but it does require correct spelling.

#### The code

```
<div class="search" id="filter">
```

```
    <input type="text" id="Search" onkeyup="searchfunction()" placeholder="Search..."/>
```

```
</div>
```

This portion covers the html part, it essentially just creates a text box to type in and sends an alert to the Javascript to start its function which covers the search & filtration aspect.

```
</script>
```

```
    <script id =FILTER>
```

```
    function searchfunction() {
```

```
        var input, filter, table, tr, td, i;
```

```
        input = document.getElementById("Search");
```

```
        filter = input.value.toUpperCase();
```

```
        table = document.getElementById("Pokelist");
```

```
        tr = table.getElementsByTagName("tr");
```

```
        for (i = 1; i < tr.length; i++) {
```

```
            // Hide the row initially.
```

```
            tr[i].style.display = "none";
```

```
            td = tr[i].getElementsByTagName("td");
```

```
            for (var j = 0; j < td.length; j++) {
```

```
                cell = tr[i].getElementsByTagName("td")[j];
```

```
                if (cell) {
```

```
                    if (cell.innerHTML.toUpperCase().indexOf(filter) > -1) {
```

```
                        tr[i].style.display = "";
```

```
        break;
    }
}
}
}
}
</script>
```

This is the javascript part of the function, it covers the actual searching part. Starting by declaring the table that it needs to be filtering which in this case is the list of Pokémon, the columns its filtering through which is all of them and finally what to do with the searches that don't match, which is to hide them.

#### Known Bugs and errors

There is currently a known bug in which the search function includes searching through the image link which is considered minor but can create confusion or skewed results.

## Sort alphabetically function

### Overview

This is one of the sort option types, it takes the list of Pokémon and puts them into alphabetical order by their name.

### The code

```
<button onclick=AZfunction() class=drp>Sort alphabetically</button>
```

This is the html section which just creates a button to trigger the JavaScript's function.

```
<script id=AZ_sort>

    function AZfunction() {

        var table, rows, switching, i, x, y, shouldSwitch;

        table = document.getElementById("Pokelist");

        switching = true;

        /* Make a loop that will continue until
        no switching has been done: */
        while (switching) {

            // Start by saying: no switching is done:
            switching = false;

            rows = table.rows;

            /* Loop through all table rows (except the
            first, which contains table headers): */
            for (i = 1; i < (rows.length - 1); i++) {

                // Start by saying there should be no switching:
                shouldSwitch = false;

                /* Get the two elements you want to compare,
                one from current row and one from the next: */
                x = rows[i].getElementsByTagName("td")[2];
                y = rows[i + 1].getElementsByTagName("td")[2];

                // Check if the two rows should switch place:
                if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {

                    // If so, mark as a switch and break the loop:
```

```
        shouldSwitch = true;

        break;
    }
}

if (shouldSwitch) {

    /* If a switch has been marked, make the switch
    and mark that a switch has been done: */

    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);

    switching = true;

}

}

}

</script>
```

This is the Javascript aspect of the code, It takes the table its designed for, in this case the list of Pokémon and once triggered sorts through the list by comparing rows alphabetically within the declared column which is [2] since names are the third row (count starts at 0).

#### Bugs and errors

There are no known bugs or errors.



## Sort reverse alphabetically function

### Overview

This is one of the sort option types, it takes the list of Pokémon and puts them into reverse alphabetical order by their name.

### The code

```
<button onclick=ZAfunction() class=drp>Sort reverse alphabet</button>
```

This is the html section which just creates a button to trigger the JavaScript's function.

```
<script id=ZA_sort>
```

```
function ZAfunction() {  
    var table, rows, switching, i, x, y, shouldSwitch;  
    table = document.getElementById("Pokelist");  
    switching = true;  
    /* Make a loop that will continue until  
    no switching has been done: */  
    while (switching) {  
        // Start by saying: no switching is done:  
        switching = false;  
        rows = table.rows;  
        /* Loop through all table rows (except the  
        first, which contains table headers): */  
        for (i = 1; i < (rows.length - 1); i++) {  
            // Start by saying there should be no switching:  
            shouldSwitch = false;  
            /* Get the two elements you want to compare,  
            one from current row and one from the next: */  
            x = rows[i].getElementsByTagName("td")[2];  
            y = rows[i + 1].getElementsByTagName("td")[2];  
            // Check if the two rows should switch place:  
            if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {  
                // If so, mark as a switch and break the loop:
```

```
        shouldSwitch = true;

        break;
    }
}

if (shouldSwitch) {
    /* If a switch has been marked, make the switch
    and mark that a switch has been done: */
    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);

    switching = true;
}
}
}

</script>
```

This is the Javascript aspect of the code, It takes the table its designed for, in this case the list of Pokémon and once triggered sorts through the list by comparing rows alphabetically (then placing them in reverse) within the declared column which is [2] since names are the third row (count starts at 0).

#### Bugs and errors

There are no known bugs or errors.

## Sort numerically function

### Overview

This is one of the sort option types, it takes the list of Pokémon and puts them into numerical order by their name. This is the default sort so it only has an effect once another sort option has been triggered

### The code

```
<button onclick=countupfunction() class=drp>Sort numerically</button>
```

This is the html section which just creates a button to trigger the JavaScript's function.

```
<script id=increase>
```

```
function countupfunction() {  
    var table, rows, switching, i, x, y, shouldSwitch;  
    table = document.getElementById("Pokelist");  
    switching = true;  
    /*Make a loop that will continue until  
    no switching has been done:*/  
    while (switching) {  
        //start by saying: no switching is done:  
        switching = false;  
        rows = table.rows;  
        /*Loop through all table rows (except the  
        first, which contains table headers):*/  
        for (i = 1; i < (rows.length - 1); i++) {  
            //start by saying there should be no switching:  
            shouldSwitch = false;  
            /*Get the two elements you want to compare,  
            one from current row and one from the next:*/  
            x = rows[i].getElementsByTagName("TD")[1];  
            y = rows[i + 1].getElementsByTagName("TD")[1];  
            //check if the two rows should switch place:  
            if (Number(x.innerHTML) > Number(y.innerHTML)) {  
                //if so, mark as a switch and break the loop:
```

```
        shouldSwitch = true;

        break;
    }
}

if (shouldSwitch) {

    /*If a switch has been marked, make the switch
    and mark that a switch has been done:*/

    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);

    switching = true;

}

}

}

</script>
```

This is the Javascript aspect of the code, It takes the table its designed for, in this case the list of Pokémon and once triggered sorts through the list by comparing rows numerically within the declared column which is [1] since numbers are the second row (count starts at 0).

#### Bugs and errors

There are no known bugs or errors.

## Sort reverse numerically function

### Overview

This is one of the sort option types, it takes the list of Pokémon and puts them into reverse numerical order by their name.

### The code

```
<button onclick=countdownfunction() class=drp>sort reverse numerically</button>
```

This is the html section which just creates a button to trigger the JavaScript's function.

```
<script id=decrease>
```

```
function countdownfunction() {  
    var table, rows, switching, i, x, y, shouldSwitch;  
    table = document.getElementById("Pokelist");  
    switching = true;  
    /*Make a loop that will continue until  
    no switching has been done:*/  
    while (switching) {  
        //start by saying: no switching is done:  
        switching = false;  
        rows = table.rows;  
        /*Loop through all table rows (except the  
        first, which contains table headers):*/  
        for (i = 1; i < (rows.length - 1); i++) {  
            //start by saying there should be no switching:  
            shouldSwitch = false;  
            /*Get the two elements you want to compare,  
            one from current row and one from the next:*/  
            x = rows[i].getElementsByTagName("TD")[1];  
            y = rows[i + 1].getElementsByTagName("TD")[1];  
            //check if the two rows should switch place:  
            if (Number(x.innerHTML) < Number(y.innerHTML)) {  
                //if so, mark as a switch and break the loop:
```

```
        shouldSwitch = true;

        break;
    }
}

if (shouldSwitch) {
    /*If a switch has been marked, make the switch
    and mark that a switch has been done:*/
    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
    switching = true;
}
}
}

</script>
```

This is the Javascript aspect of the code, It takes the table its designed for, in this case the list of Pokémon and once triggered sorts through the list by comparing rows numerically (then placing them in reverse order) within the declared column which is [1] since numbers are the second row (count starts at 0).

#### Bugs and errors

There are no known bugs or errors.

## Dropdown menu functions

### Overview

The drop down menu's function is to hold all of the sorting options, this compiles them all safely and makes the site look cleaner and makes it easier to navigate.

### The code

```
<div class=sort id=sort>

    <button onclick=sortfunction() class="dropbtn">sort options</button>

    <div id="dropdown" class="sort-content">

        <button onclick=countupfunction() class=drp>Sort numerically</button>

        <button onclick=countdownfunction() class=drp>sort reverse numerically</button>

        <button onclick=AZfunction() class=drp>Sort alphabetically</button>

        <button onclick=ZAfunction() class=drp>Sort reverse alphabet</button>

    </div>
```

The HTML section divides each button and declares them all as a part of the drop down menu.

```
.dropbtn {
    background-color: red;
    padding: 10px;}

.sort-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: 160px;
    overflow: auto;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;}

.sort-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;}
```

```
.sort a:hover {background-color: #ddd;}
```

```
.show {display: block;}
```

The CSS section sends alerts as to when to show the dropdown menu's contents and how it should appear on the page itself.

#### Bugs and errors

Drop down menu moves the entire screen to the left slightly, not breaking or damaging but slightly annoying.

## Redirect functions

### Overview

Alerts as to when the page needs to be changed to a specifically desired page. A really simple function that essentially just reroutes the user to another page.

### The code

```
onclick="document.location = 'X' "
```

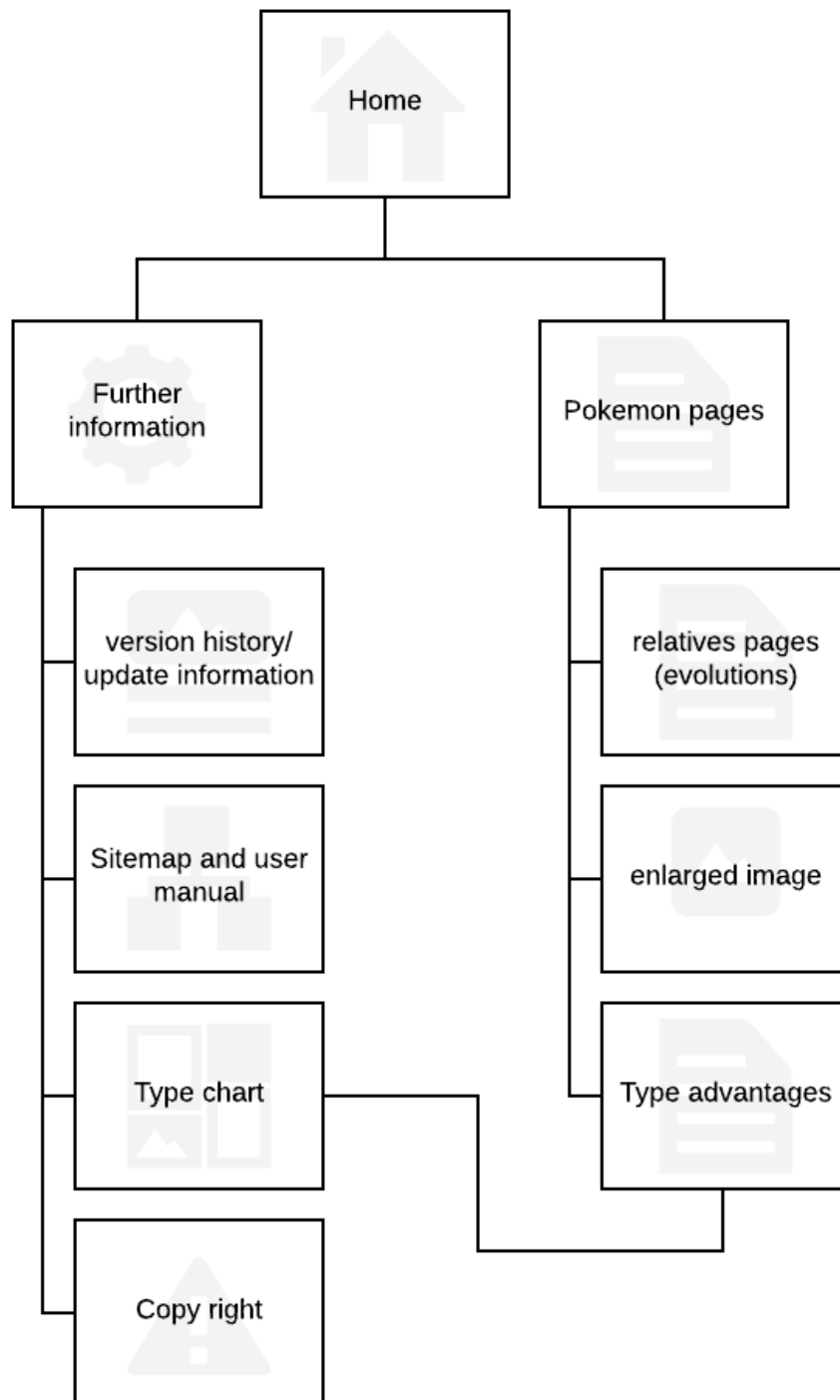
The change function simply needs the above code on the desired linking element with X = desired link.

### Bugs and errors

There are no known bugs or errors.



## Site map



## Data (class) dictionary

Term used	Definition
Grid	Used to start out the grid and set up the layout of the website. Used within div elements and sorts out where its contents lies in the page
Sort	Refers to the sorting element used on the homepage, includes all of the options and the dropdown menu
Dropbtn	Used in conjunction with the sorting elements to ensure they're all hidden until the user wants to use them.
Extras	The word used to declare the extra information code sections.
Index	Refers to the list of all Pokémon presently in use.
Initial	The area on the Pokémon pages that holds the basic information.
Base-stats	The broad section for the base statistics that the Pokémon has
Base	A more narrow version referring directly to the numbers used for the base stats.
Sprites	The first two images that includes the standard sprite as well as the shiny
Evolutions	The evolutions of the Pokémon that run down the left side of the screen.
entry	The area in the centre of the pages that is used for the Pokédex entries.
Further	The detailed information that runs down the right side of the screen.
Locations	The area at the bottom of the screen which explains what locations the Pokémon can be found in.
Comment	The text area in the further information/help section of the website.
User	Used for the user manual section of the help page
Tech	Used for the technical manual section of the help page
credit	The acknowledgement sections of the website.

## Update procedure

1. Request permission to edit through the provided interface on the website
2. Download the necessary files if approved (sent through email)
3. Submit any changes or adjustments for testing and checking

## System requirements

2Gb ram

Core I3 or better

64 bit operating system

Or

IOS 1 or better