Basic Scripts

Image changer

This snippet enables you to replace an image with another one when the user clicks on it and switches back to the original image when the user clicks on it again.

```
cont img = document.querySelector('img');

img.onclick = function() {
  const src = img.getAttribute('src');
  if (src === 'image1.jpg') {
    img.setAttribute('src', 'image2.jpg');
  } else {
    img.src = 'image1.jpg';
  }
};
```

Prompt and store user input

This snippet prompts the user to enter their name and stores it in a variable. We also add an initialization code underneath the function definition to display the user's name in the `h1` element.

```
function setName() {
    const name = prompt('Please enter your name');
    localStorage.setItem('name', name);
    document.querySelector('h1').textContent = `Hello ${name}`;
}
if(!localStorage.getItem('name')) {
    setName();
} else {
    const storedName = localStorage.getItem('name');
    document.querySelector('h1').textContent = `Hello ${storedName}`;
}
```

Recursive prompt to avoid null input

This snippet prompts the user to enter their name and stores it in a variable. If the user enters a null value, the prompt will appear again.

```
function setName() {
    const name = prompt('Please enter your name');
    if (!name) {
        setName();
    } else {
```

```
        localStorage.setItem('name', name);
        document.querySelector('h1').textContent = `Hello ${name}`;
    }
}
```

Display the real-time

This snippet displays the current time in the `h1` element and updates it every second.

```
function tick() {
  const t = new Date()
  const clock = document.getElementById("clock")
  clock.innerHTML = `${t.getHours()}:${t.getMinutes()}:${t.getSeconds()}`
}

function setTimer() {
  setInterval(tick, 1000);
}

// Run the setTimer function when the page loads
window.addEventListener("load", setTimer)
```

One liner conditions to translate directions of wind

This snippet translates the direction of the wind from coordinates to North, South, East, and West.

```
function translateWindDirection(deg) {
  if (deg > 0 && deg <= 22.5) return "N";
  if (deg > 22.5 && deg <= 67.5) return "NE";
  if (deg > 67.5 && deg <= 112.5) return "E";
  if (deg > 112.5 && deg <= 157.5) return "SE";
  if (deg > 157.5 && deg <= 202.5) return "S";
  if (deg > 202.5 && deg <= 247.5) return "SW";
  if (deg > 247.5 && deg <= 292.5) return "W";
  if (deg > 292.5 && deg <= 337.5) return "NW";
  if (deg > 337.5 && deg <= 360) return "N";
  else return "N/A";
}
```

Advanced Scripts

Fetch data from an API

This snippet fetches data from an API and displays it in the console.

```
function getWeather() {
    let latitude="38.67"
    let longitude="-9.32"
    let
fields="precipitation,cloudcover,winddirection_10m,apparent_temperature,temperatur
e_2m,relativehumidity_2m,windspeed_10m"
    // Create the api url based on the variables
    let url = `https://api.open-meteo.com/v1/forecast?
latitude=${latitude}&longitude=${longitude}&current_weather=true&hourly=${fields}`
;
    // Create a promise to fetch the data from the api
    fetch(url)
    .then(response => response.json())
    .then(data => {
        console.log(data);
    });
}
```

Display the return of an API line by line

This snippet gets the data from an API and displays it in a list with each item on a new line. We also slice the data to use the `.slice()` method and the `idx` element of the `.forEach()` method.

```
function listHourlyWeather(data){
  let h = new Date().getHours();
  let listElement = document.getElementById("hourly-weather");
  // We only look at the next 24 hours
  let hours = data.hourly.time.slice(h, h + 24);

  // Let us loop through the next 24 hours
  hours.forEach((hour, idx) => {
    // Create a list item
    let li = document.createElement("li");
    // Define the class of the list item
    li.className = "houritem";
    let div1 = document.createElement("div");
    // Keep only the time part of the date
    div1.innerHTML = hour.substring(11);
    let div2 = document.createElement("div");
    div2.innerHTML = data.hourly.temp[idx];
    // Append the divs to the list item
    li.appendChild(div1);
    li.appendChild(div2);
    // Append the list item to the list
    listElement.appendChild(li);
  });
}
```

Visual display of a ranking

This snippet displays a ranking as x/5 stars with the amount of reviews in parentheses.

```
function createStars(rating, number) {
    const div = document.createElement("DIV")
    for (let i = 0; i < 5; i++) {
        const star = document.createElement("SPAN")
        star.innerText = "★"
        star.classList.add("star")
        // Make it yellow if it is less than the rating
        if (i + 1 <= rating)
            star.classList.add("yellowstar")
        div.append(star)
    }
    const numberSpan = document.createElement("SPAN")
    // Add the number of reviews in parentheses
    numberSpan.innerText = `(${number})`
    div.append(numberSpan)
    return div
}
```

Create a dropdown menu

This snippet creates a dropdown menu with the options "Option 1", "Option 2", and "Option 3".

- We want to achieve the following HTML:

```
<form action="http://someservice.to" method="POST" _target="">
  <Label for="email">Email: </Label>
  <input type="text" id="email" name="email">
  <select name="period" id="subscription">
      <option value="daily">Daily</option>
      <option value="weekly" selected>Weekly</option>
      <option value="monthly">Monthly</option>
  </select>
  <input type="submit" value="Subscribe">
</form>
```

- If we want to just create the dropdown menu, we can use the following JavaScript:

```
function createDropdown(options, selected) {
    const select = document.getElementById("subscription");
    options.forEach(option => {
        const opt = document.createElement("option");
        if (option === selected) {
            opt.selected = true;
        }
        opt.value = option.toLowerCase();
        opt.textContent = option;
```

```
            select.appendChild(opt);
        });
    }
```