3 Ideas for using Javascript:

Chenxin Zhang 3568978

1. True form validation :

```
var invalid_characters = /^[\w ]+$/;
var num = /\D/;
var badWords = ["shit", "shiit", "shiiit"];

if(!invalid_characters.test(x)) {
alert("Error: First name contains invalid characters!");
return false;
}

if(num.test(z)){
alert("Please only enter numeric characters for phone number ");
}
if(badWords.match(a)) {
alert("no bad words here!");
return false;
}
```

2. Investment simulator :

From https://stackoverflow.com/questions/13455042/random-number-between-negative-and-positive-value:

```
var ranNum = Math.ceil(Math.random() * 99) * (Math.round(Math.random()) ? 1 : -1)
//This will generator a random number between 1 and 99 with 50%
positive and 50% negative

x = 1000 // Beginning money

Button.onclick{
  for i = 0:100
      X+= Math.ceil(Math.random() * 99) * (Math.round(Math.random()) ? 1 : -1)
End
} // loop 100 times and x is the investor's money with random growth
or drop. Note real stock market is not random; but here is just a
simulation

Print(x)  // resulting money
```

3. A shopping cart :

Thanks to this website Showed me how to construct a shopping cart using JS: https://code-boxx.com/simple-vanilla-javascript-shopping-cart/

```
var cart = {
  // (A) PROPERTIES
  hPdt : null,      // html products list
  hItems : null,    // html current cart
  items : {},       // current items in cart
```

```
  iURL : "images/", // product image url folder

  // (B) LOCALSTORAGE CART
  // (B1) SAVE CURRENT CART INTO LOCALSTORAGE
  save : () => {
    localStorage.setItem("cart", JSON.stringify(cart.items));
  },

  // (B2) LOAD CART FROM LOCALSTORAGE
  load : () => {
    cart.items = localStorage.getItem("cart");
    if (cart.items == null) { cart.items = {}; }
    else { cart.items = JSON.parse(cart.items); }
  },

  // (B3) EMPTY ENTIRE CART
  nuke : () => { if (confirm("Empty cart?")) {
    cart.items = {};
    localStorage.removeItem("cart");
    cart.list();
  }},

  // (C) INITIALIZE
  init : () => {
    // (C1) GET HTML ELEMENTS
    cart.hPdt = document.getElementById("cart-products");
    cart.hItems = document.getElementById("cart-items");

    // (C2) DRAW PRODUCTS LIST
    cart.hPdt.innerHTML = "";
    let template = document.getElementById("template-
product").content,
        p, item, part;
    for (let id in products) {
      p = products[id];
      item = template.cloneNode(true);
      item.querySelector(".p-img").src = cart.iURL + p.img;
      item.querySelector(".p-name").textContent = p.name;
      item.querySelector(".p-desc").textContent = p.desc;
      item.querySelector(".p-price").textContent = "$" +
p.price.toFixed(2);
      item.querySelector(".p-add").onclick = () => { cart.add(id); };
      cart.hPdt.appendChild(item);
    }

    // (C3) LOAD CART FROM PREVIOUS SESSION
    cart.load();

    // (C4) LIST CURRENT CART ITEMS
    cart.list();
  },

  // (D) LIST CURRENT CART ITEMS (IN HTML)
  list : () => {
    // (D1) RESET
    cart.hItems.innerHTML = "";
    let item, part, pdt, empty = true;
    for (let key in cart.items) {
      if (cart.items.hasOwnProperty(key)) { empty = false; break; }
    }
```

```
    // (D2) CART IS EMPTY
    if (empty) {
      item = document.createElement("div");
      item.innerHTML = "Cart is empty";
      cart.hItems.appendChild(item);
    }

    // (D3) CART IS NOT EMPTY - LIST ITEMS
    else {
      let template = document.getElementById("template-cart").content,
          p, total = 0, subtotal = 0;
      for (let id in cart.items) {
        // (D3-1) PRODUCT ITEM
        p = products[id];
        item = template.cloneNode(true);
        item.querySelector(".c-del").onclick = () =>
{ cart.remove(id); };
        item.querySelector(".c-name").textContent = p.name;
        item.querySelector(".c-qty").value = cart.items[id];
        item.querySelector(".c-qty").onchange = function ()
{ cart.change(id, this.value); };
        cart.hItems.appendChild(item);

        // (D3-2) SUBTOTAL
        subtotal = cart.items[id] * p.price;
        total += subtotal;
      }

      // (D3-3) TOTAL AMOUNT
      item = document.createElement("div");
      item.className = "c-total";
      item.id = "c-total";
      item.innerHTML ="TOTAL: $" + total;
      cart.hItems.appendChild(item);

      // (D3-4) EMPTY & CHECKOUT
      item = document.getElementById("template-cart-
checkout").content.cloneNode(true);
      cart.hItems.appendChild(item);
    }
  },

  // (E) ADD ITEM INTO CART
  add : (id) => {
    if (cart.items[id] == undefined) { cart.items[id] = 1; }
    else { cart.items[id]++; }
    cart.save(); cart.list();
  },

  // (F) CHANGE QUANTITY
  change : (pid, qty) => {
    // (F1) REMOVE ITEM
    if (qty <= 0) {
      delete cart.items[pid];
      cart.save(); cart.list();
    }

    // (F2) UPDATE TOTAL ONLY
    else {
      cart.items[pid] = qty;
      var total = 0;
```

```
      for (let id in cart.items) {
        total += cart.items[id] * products[id].price;
        document.getElementById("c-total").innerHTML ="TOTAL: $" +
total;
      }
    }
  },

  // (G) REMOVE ITEM FROM CART
  remove : (id) => {
    delete cart.items[id];
    cart.save();
    cart.list();
  },

  // (H) CHECKOUT
  checkout : () => {
    // SEND DATA TO SERVER
    // CHECKS
    // SEND AN EMAIL
    // RECORD TO DATABASE
    // PAYMENT
    // WHATEVER IS REQUIRED
    alert("TO DO");

    /*
    var data = new FormData();
    data.append("cart", JSON.stringify(cart.items));
    data.append("products", JSON.stringify(products));

    fetch("SERVER-SCRIPT", { method:"POST", body:data })
    .then(res=>res.text()).then((res) => {
      console.log(res);
    })
    .catch((err) => { console.error(err); });
    */
  }
};
window.addEventListener("DOMContentLoaded", cart.init);
```