# Example 1

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Quiz Project</title>

<style>

.box {

display: inline-block;

padding: 5px;

border: 1px solid #ddd;

border-radius: 10px;

margin: 5px;

min-width: 100px;

text-align: center;

}

.boxCursor {

cursor: pointer;

}

.boxCursor:hover {

background-color: azure;

}

.que {

font-size: 1.8em;

margin-bottom: 15px;

}

.btn {

display: block;

width: 50%;

margin: auto;

margin-top: 20px;

border-top: 1px solid black;

padding: 10px;

}

</style>

</head>

<body>

<h1>JavaScript Quiz</h1>

<div class="output"></div>

<button class="btn">Start Game</button>

<script src="app7.js"></script>

</body>

</html>

const id = '1H2xq8J5uZKDABgu1JlvV-nVWKqZoV8\_UpB\_D\_snOrLo';

const url = 'https://spreadsheets.google.com/feeds/list/'+id+'/1/public/values?alt=json';

const questions = [];

const output = document.querySelector('.output');

const btn = document.querySelector('.btn');

let cur = 0;

const player = {

score: 0,

answers: []

}

const holder = [];

const totalOutput = document.querySelector('h1');

btn.addEventListener('click', (e) => {

if (cur >= questions.length) {

let html = `<hr><h1>Score = ${player.score}</h1>`;

player.answers.forEach((el) => {

let bg = el.correct ? 'green' : 'red';

html += `<div style="background:${bg}">Question : ${capitalizeText(el.question)}? <br>`;

html += `Response : ${el.response} (${el.correctAnswer})<br>`;

html += `Result : ${el.correct} </div><br>`;

})

output.innerHTML = html;

} else {

newQuestion();

}

btn.style.display = 'none';

})

window.addEventListener('DOMContentLoaded', () => {

//console.log('ready');

loadQuestions();

})

function capitalizeText(str) {

return str.charAt(0).toUpperCase() + str.slice(1);

}

function newQuestion() {

updateScore();

const el = questions[cur];

el.options.sort(() => {

return 0.5 - Math.random()

});

console.log(cur);

console.log(questions.length);

console.log(questions[cur]);

output.innerHTML = '';

const que1 = document.createElement('div');

que1.classList.add('que');

let strOutput = capitalizeText(el.question);

console.log(strOutput);

const ans1 = document.createElement('div');

que1.textContent = strOutput + '?';

holder.length = 0;

el.options.forEach((ans) => {

const div = document.createElement('div');

holder.push(div);

div.correctAnswer = el.correct;

div.textContent = ans.response;

div.classList.add('box');

div.classList.add('boxCursor');

div.correct = ans.correct;

div.addEventListener('click', selOption);

ans1.append(div);

})

output.append(que1);

output.append(ans1);

}

function selOption(e) {

//track the progress

console.log(e);

const tempObj = {

"question": questions[cur].question,

"response": e.target.textContent,

"correctAnswer": e.target.correctAnswer

}

endTurn();

if (e.target.correct) {

player.score++;

updateScore();

tempObj.correct = true;

e.target.style.backgroundColor = 'green';

} else {

e.target.style.backgroundColor = 'red';

tempObj.correct = false;

}

player.answers.push(tempObj);

e.target.style.color = 'white';

nextBtn();

console.log(player);

}

function updateScore() {

totalOutput.innerHTML = `${cur+1} out of ${questions.length} Score: ${player.score}`;

}

function endTurn() {

holder.forEach((el) => {

el.removeEventListener('click', selOption);

el.style.backgroundColor = '#ddd';

el.classList.remove('boxCursor');

})

}

function nextBtn() {

btn.style.display = 'block';

cur++;

if (cur >= questions.length) {

btn.textContent = 'See Score';

} else {

btn.textContent = 'Next Question';

}

}

function loadQuestions() {

fetch(url).then(rep => rep.json())

.then((data) => {

//console.log(data.feed.entry);

data.feed.entry.forEach(el => {

//console.log(el.content['$t']);

let mainTemp = {options:[]};

for(let key in el){

if(key.substring(0,3) == 'gsx'){

let header = key.slice(4);

let val = el[key]['$t'];

if(header=='question'){

mainTemp.question = val;

}else if(header=='correct'){

mainTemp.correct = val;

let temp = {

"response": val,

"correct": true

};

mainTemp.options.push(temp);

}else{

if(val.length>0){

let temp = {

"response": val,

"correct": false

};

mainTemp.options.push(temp);

}

}

//console.log(header);

//console.log(el[key]['$t']);

}

}

questions.push(mainTemp);

});

console.log(questions);

//document.write(JSON.stringify(questions));

})

}

# Example 2

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Quiz Project</title>

<style>

.box {

display: inline-block;

padding: 5px;

border: 1px solid #ddd;

border-radius: 10px;

margin: 5px;

min-width: 100px;

text-align: center;

}

.boxCursor {

cursor: pointer;

}

.boxCursor:hover {

background-color: azure;

}

.que {

font-size: 1.8em;

margin-bottom: 15px;

}

.btn {

display: block;

width: 50%;

margin: auto;

margin-top: 20px;

border-top: 1px solid black;

padding: 10px;

}

</style>

</head>

<body>

<h1>JavaScript Quiz</h1>

<div class="output"></div>

<button class="btn">Start Game</button>

<script src="app5.js"></script>

</body>

</html>

const url = 'quiz.json';

const questions = [];

const output = document.querySelector('.output');

const btn = document.querySelector('.btn');

let cur = 0;

const player = {

score: 0,

answers: []

}

const holder = [];

const totalOutput = document.querySelector('h1');

btn.addEventListener('click', (e) => {

if (cur >= questions.length) {

let html = `<hr><h1>Score = ${player.score}</h1>`;

player.answers.forEach((el) => {

let bg = el.correct ? 'green' : 'red';

html += `<div style="background:${bg}">Question : ${capitalizeText(el.question)}? <br>`;

html += `Response : ${el.response} (${el.correctAnswer})<br>`;

html += `Result : ${el.correct} </div><br>`;

})

output.innerHTML = html;

} else {

newQuestion();

}

btn.style.display = 'none';

})

window.addEventListener('DOMContentLoaded', () => {

//console.log('ready');

loadQuestions();

})

function capitalizeText(str) {

return str.charAt(0).toUpperCase() + str.slice(1);

}

function newQuestion() {

updateScore();

const el = questions[cur];

el.options.sort(() => {

return 0.5 - Math.random()

});

console.log(cur);

console.log(questions.length);

console.log(questions[cur]);

output.innerHTML = '';

const que1 = document.createElement('div');

que1.classList.add('que');

let strOutput = capitalizeText(el.question);

console.log(strOutput);

const ans1 = document.createElement('div');

que1.textContent = strOutput + '?';

holder.length = 0;

el.options.forEach((ans) => {

const div = document.createElement('div');

holder.push(div);

div.correctAnswer = el.correct;

div.textContent = ans.response;

div.classList.add('box');

div.classList.add('boxCursor');

div.correct = ans.correct;

div.addEventListener('click', selOption);

ans1.append(div);

})

output.append(que1);

output.append(ans1);

}

function selOption(e) {

//track the progress

console.log(e);

const tempObj = {

"question": questions[cur].question,

"response": e.target.textContent,

"correctAnswer": e.target.correctAnswer

}

endTurn();

if (e.target.correct) {

player.score++;

updateScore();

tempObj.correct = true;

e.target.style.backgroundColor = 'green';

} else {

e.target.style.backgroundColor = 'red';

tempObj.correct = false;

}

player.answers.push(tempObj);

e.target.style.color = 'white';

nextBtn();

console.log(player);

}

function updateScore() {

totalOutput.innerHTML = `${cur+1} out of ${questions.length} Score: ${player.score}`;

}

function endTurn() {

holder.forEach((el) => {

el.removeEventListener('click', selOption);

el.style.backgroundColor = '#ddd';

el.classList.remove('boxCursor');

})

}

function nextBtn() {

btn.style.display = 'block';

cur++;

if (cur >= questions.length) {

btn.textContent = 'See Score';

} else {

btn.textContent = 'Next Question';

}

}

function loadQuestions() {

fetch(url).then(rep => rep.json())

.then((data) => {

//console.log(data);

data.forEach(el => {

let temp = [];

el.incorrect.forEach((ans) => {

let tempObj = {

"response": ans,

"correct": false

}

temp.push(tempObj);

})

let tempObj = {

"response": el.correct,

"correct": true

}

temp.push(tempObj);

//console.log(temp);

let mainTemp = {

"question": el.question,

"options": temp,

"correct": el.correct

}

questions.push(mainTemp);

});

console.log(questions);

//document.write(JSON.stringify(questions));

})

}

[

{

"question" : "what color is the sky",

"correct" : "blue",

"incorrect" : [

"green","red","pink","purple"

]

},

{

"question" : "what color is the grass",

"correct" : "green",

"incorrect" : [

"blue","red","pink","purple"

]

},

{

"question" : "what color is the Sun",

"correct" : "yellow",

"incorrect" : [

"blue","green","pink","purple"

]

},

{

"question" : "what color is the Sun",

"correct" : "yellow",

"incorrect" : [

"blue","green","pink","purple"

]

}

]

# Example 3

const id = '1H2xq8J5uZKDABgu1JlvV-nVWKqZoV8\_UpB\_D\_snOrLo';

const url = 'https://spreadsheets.google.com/feeds/list/'+id+'/1/public/values?alt=json';

const questions = [];

const output = document.querySelector('.output');

const btn = document.querySelector('.btn');

let cur = 0;

const player = {

score: 0,

answers: []

}

const holder = [];

const totalOutput = document.querySelector('h1');

btn.addEventListener('click', (e) => {

if(btn.textContent == 'Download Report'){

console.log('download');

downloadReport();

}else{

if (cur >= questions.length) {

let html = `<hr><h1>Score = ${player.score}</h1>`;

player.answers.forEach((el) => {

let bg = el.correct ? 'green' : 'red';

html += `<div style="background:${bg}">Question : ${capitalizeText(el.question)}? <br>`;

html += `Response : ${el.response} (${el.correctAnswer})<br>`;

html += `Result : ${el.correct} </div><br>`;

})

btn.textContent = 'Download Report';

btn.style.backgroundColor = 'green';

btn.style.color = 'white';

btn.style.display = 'block';

output.innerHTML = html;

} else {

btn.style.display = 'none';

newQuestion();

}

}

})

window.addEventListener('DOMContentLoaded', () => {

//console.log('ready');

loadQuestions();

})

function downloadReport(){

let file;

let holder =`"QUESTION","Your Selection","Correct","Correct Answer"\n`;

let filename = 'Quiz.csv';

let prop = {type:"text/csv;charset=utf-8;"};

player.answers.forEach((el) => {

console.log(el);

holder += `"${el.question}","${el.response}","${el.correct}","${el.correctAnswer}"\n`;

})

console.log(holder);

file = new File([holder],filename,prop);

let link = document.createElement('a');

let url1 = window.URL.createObjectURL(file);

link.setAttribute('href',url1);

link.setAttribute('download',filename);

link.style.visibility = 'hidden';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

}

function capitalizeText(str) {

return str.charAt(0).toUpperCase() + str.slice(1);

}

function newQuestion() {

updateScore();

const el = questions[cur];

el.options.sort(() => {

return 0.5 - Math.random()

});

console.log(cur);

console.log(questions.length);

console.log(questions[cur]);

output.innerHTML = '';

const que1 = document.createElement('div');

que1.classList.add('que');

let strOutput = capitalizeText(el.question);

console.log(strOutput);

const ans1 = document.createElement('div');

que1.textContent = strOutput + '?';

holder.length = 0;

el.options.forEach((ans) => {

const div = document.createElement('div');

holder.push(div);

div.correctAnswer = el.correct;

div.textContent = ans.response;

div.classList.add('box');

div.classList.add('boxCursor');

div.correct = ans.correct;

div.addEventListener('click', selOption);

ans1.append(div);

})

output.append(que1);

output.append(ans1);

}

function selOption(e) {

//track the progress

console.log(e);

const tempObj = {

"question": questions[cur].question,

"response": e.target.textContent,

"correctAnswer": e.target.correctAnswer

}

endTurn();

if (e.target.correct) {

player.score++;

updateScore();

tempObj.correct = true;

e.target.style.backgroundColor = 'green';

} else {

e.target.style.backgroundColor = 'red';

tempObj.correct = false;

}

player.answers.push(tempObj);

e.target.style.color = 'white';

nextBtn();

console.log(player);

}

function updateScore() {

totalOutput.innerHTML = `${cur+1} out of ${questions.length} Score: ${player.score}`;

}

function endTurn() {

holder.forEach((el) => {

el.removeEventListener('click', selOption);

el.style.backgroundColor = '#ddd';

el.classList.remove('boxCursor');

})

}

function nextBtn() {

btn.style.display = 'block';

cur++;

if (cur >= questions.length) {

btn.textContent = 'See Score';

} else {

btn.textContent = 'Next Question';

}

}

function loadQuestions() {

fetch(url).then(rep => rep.json())

.then((data) => {

//console.log(data.feed.entry);

data.feed.entry.forEach(el => {

//console.log(el.content['$t']);

let mainTemp = {options:[]};

for(let key in el){

if(key.substring(0,3) == 'gsx'){

let header = key.slice(4);

let val = el[key]['$t'];

if(header=='question'){

mainTemp.question = val;

}else if(header=='correct'){

mainTemp.correct = val;

let temp = {

"response": val,

"correct": true

};

mainTemp.options.push(temp);

}else{

if(val.length>0){

let temp = {

"response": val,

"correct": false

};

mainTemp.options.push(temp);

}

}

//console.log(header);

//console.log(el[key]['$t']);

}

}

questions.push(mainTemp);

});

console.log(questions);

//document.write(JSON.stringify(questions));

})

}