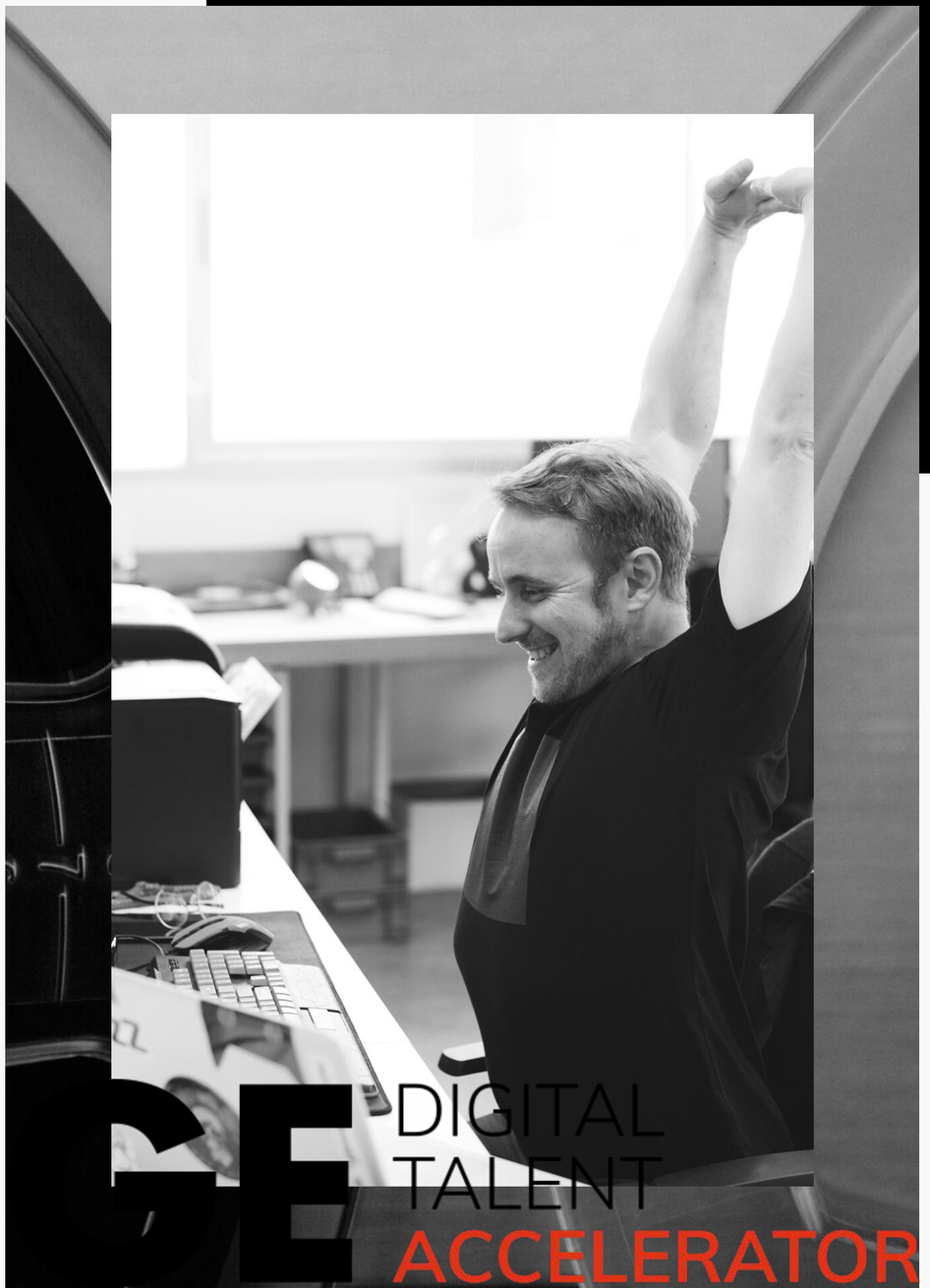
SPA

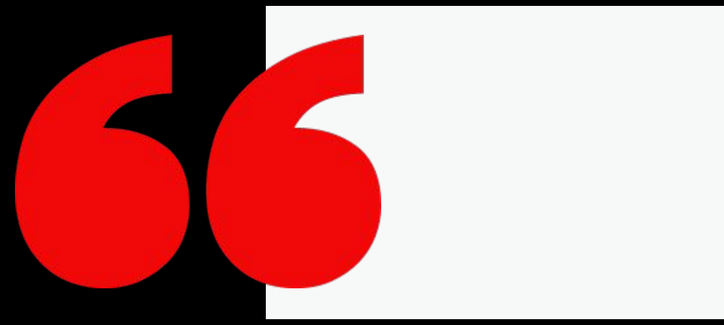
THE BRIDGE
DIGITAL TALENT ACCELERATOR

Índice

¿Que es una SPA?
Ejemplo simple SPA
Ejemplo Quiz



THE BRIDGE
DIGITAL TALENT ACCELERATOR

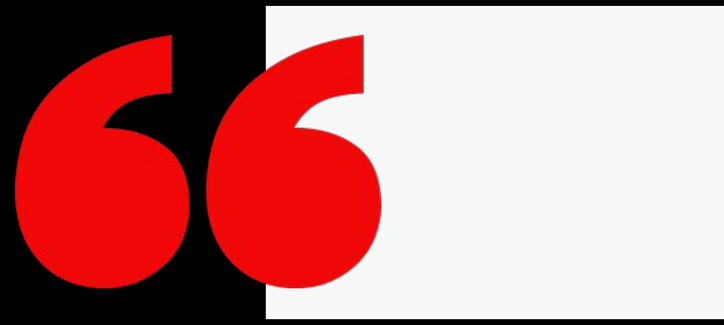


¿Qué son las Single-page application (SPA)?

SPA

Una **web SPA** (Single Page Application) es una **página web la cual está todo el contenido en una sola página**, es decir, **carga tan solo un archivo HTML y todo se produce dentro de este único archivo**.

Ejemplo SPA con JS



HTML

Para entenderlo mejor veamos un ejemplo:

```
<nav>
  <span id="homeNav">home</span>
  <span id="aboutNav">about</span>
</nav>
<div id="home">
  <h1>Home Page</h1>
</div>

<div id="about" class="hide">
  <h1>About page</h1>
</div>
```

le asignamos un id para poder traernoslo después al js

le asignamos la clase hide para que no se vea

CSS

Creamos la clase hide que esconderá las diferentes vistas de nuestra página:

```
.hide{
    display: none;
}
```

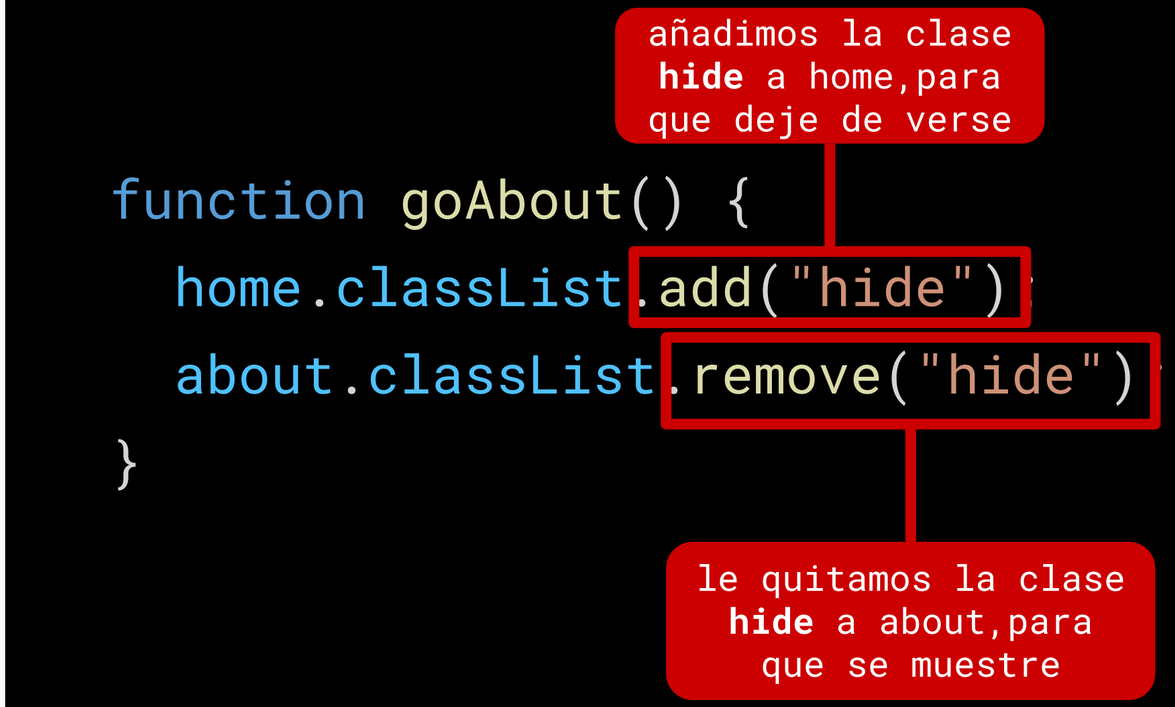
JS

Ahora nos traemos los diferentes id’s que habíamos definido previamente:

```
const home = document.getElementById("home");
const about = document.getElementById("about");
const homeNav = document.getElementById("homeNav");
const aboutNav = document.getElementById("aboutNav");
```

classList

La propiedad **classList** devuelve los nombres de clases CSS de un elemento.



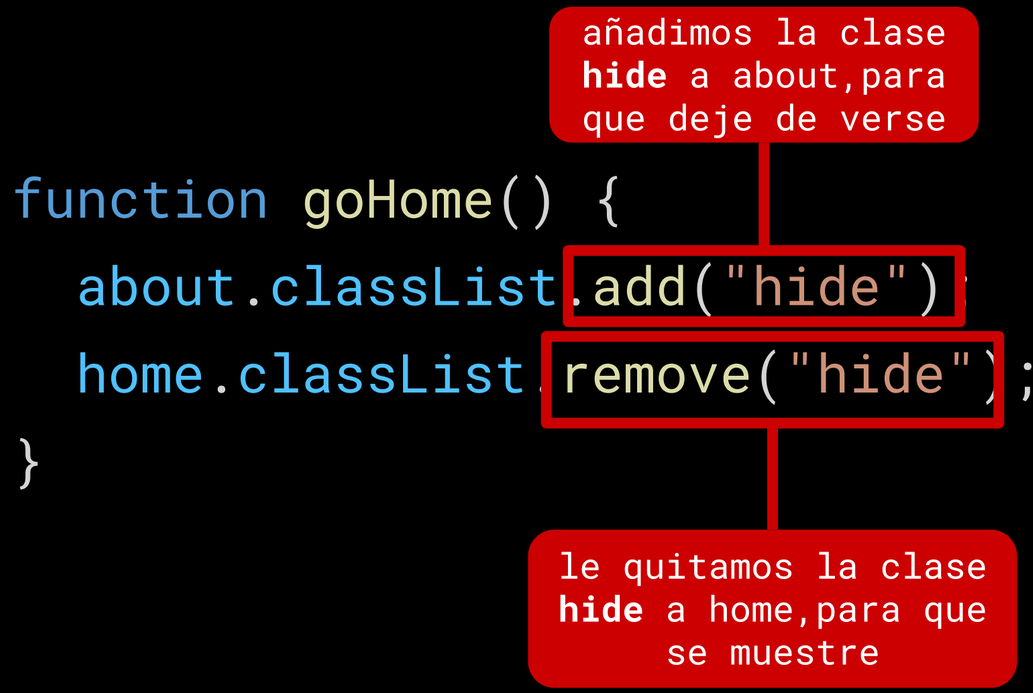
addEventListener

Ahora le decimos que cuando cliquen en la opción about del nav se ejecute la función goAbout:

```
aboutNav.addEventListener("click", goAbout);
```

goHome

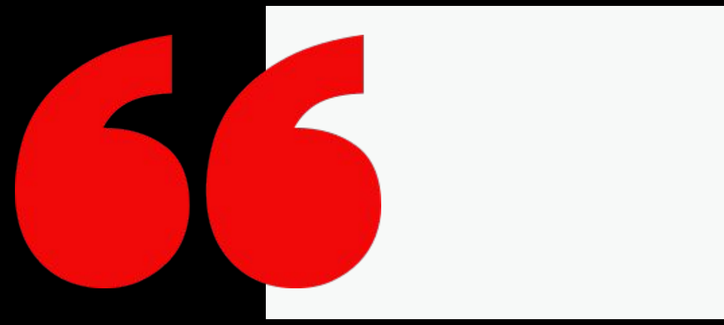
Para poder regresar a home, vamos a crear la siguiente función:



addEventListener

Ahora le decimos que cuando cliquen en la opción home del nav se ejecute la función goHome:

```
homeNav.addEventListener("click", goHome);
```



Vamos a complicarlo un poco mas…

Vista contacto

Creamos una vista contacto:

```
<nav>
    <span id="homeNav">home</span>
    <span id="aboutNav">about</span>
    <span id="contactNav">contact</span>
</nav>
<div id="home">
    <h1>Home Page</h1>
</div>

<div id="about" class="hide">
    <h1>About page</h1>
</div>

<div id="contact" class="hide">
    <h1>Contact page</h1>
</div>
```

añadimos la opción de ir a vista contacto

le asignamos la clase hide para que no se vea

JS

Ahora nos traemos los nuevos id’s de contacto:

```
const contact = document.getElementById("contact");
const contactNav = document.getElementById("contactNav");
```

hideView()

Ahora crearemos una función que se encargue de esconder las vistas:

```
function hideView() {
  home.classList.add("hide");
  about.classList.add("hide");
  contact.classList.add("hide");
}
```

goContact()

Para poder ir a contact, vamos a crear la siguiente función:

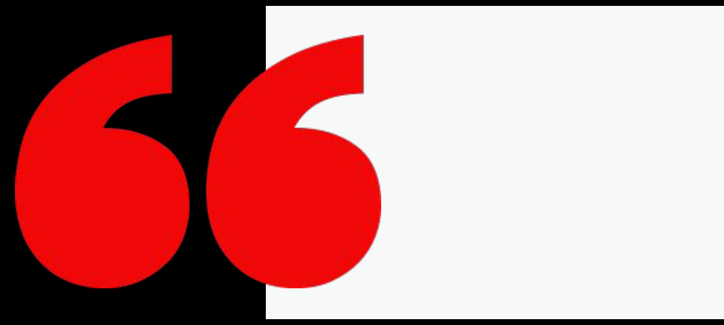
```
function goContact() {
  hideView();

  contact.classList.remove("hide");
}
```

escondemos las demás vistas

le quitamos la clase **hide** a contact,para que se muestre

Ejemplo Quiz

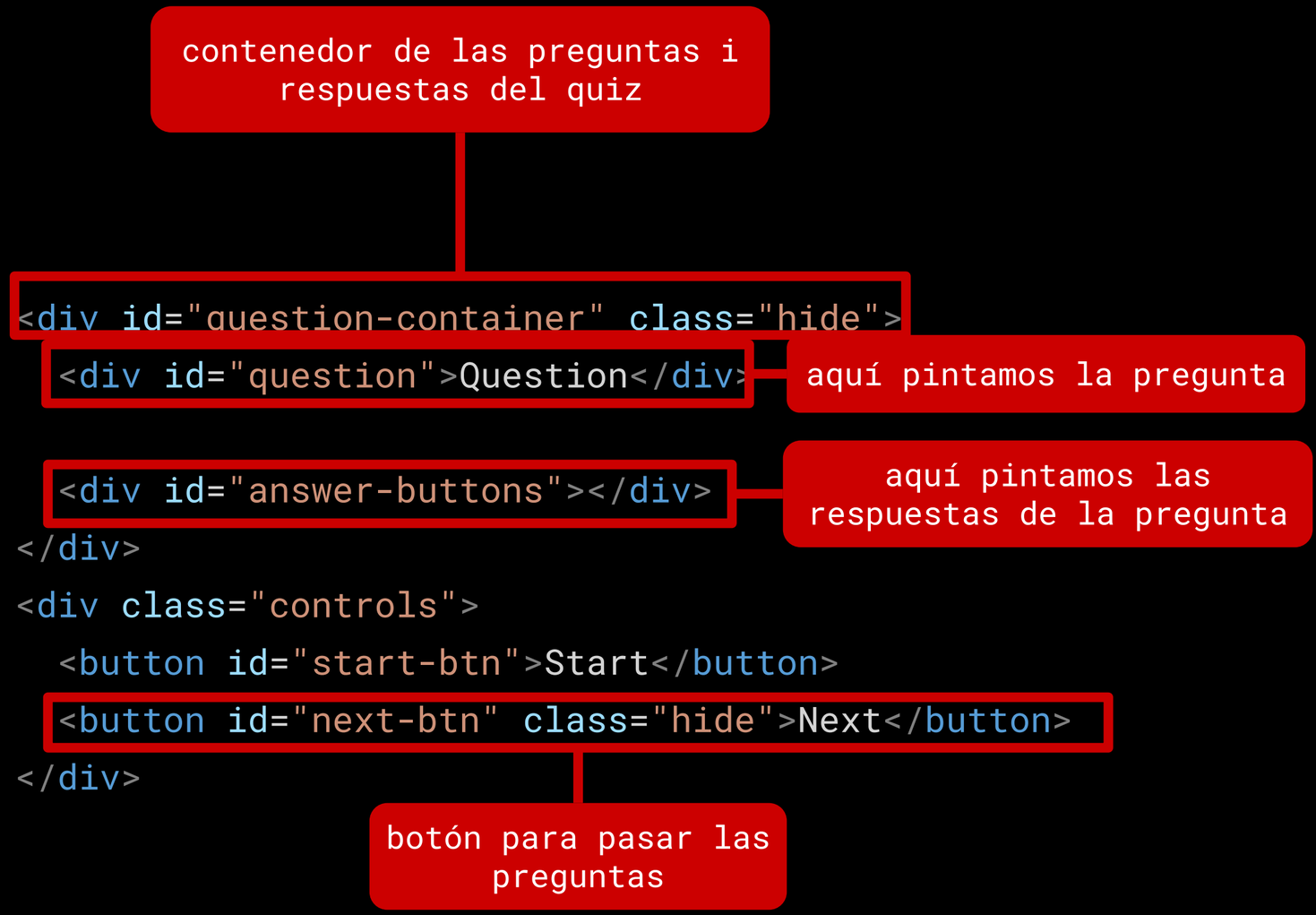


Quiz

En el siguiente ejemplo crearemos un Quiz SPA

HTML

Estructura HTML del quiz:



CSS

Creamos la clase hide:

```
.hide{
    display: none;
}
```

JS

Ahora nos traemos los diferentes id’s que habíamos definido previamente:

```
const startButton = document.getElementById("start-btn");
const nextButton = document.getElementById("next-btn");
const questionContainerElement = document.getElementById("question-container");
const questionElement = document.getElementById("question");
const answerButtonsElement = document.getElementById("answer-buttons");
```

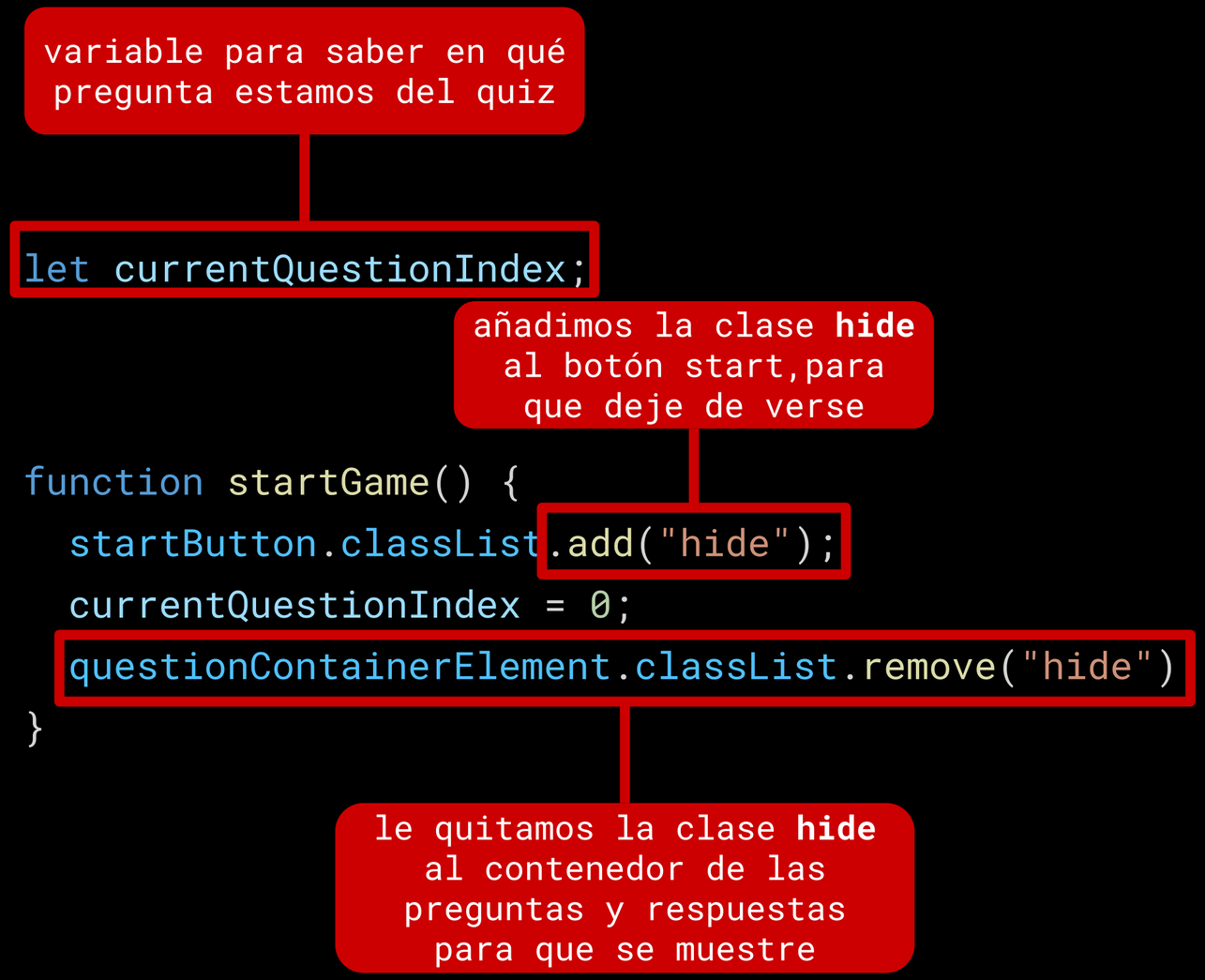
Questions

Creamos una constante questions que contendrá las preguntas del quiz:

```
const questions = [
  {
    question: "What is 2 + 2?",
    answers: [
      { text: "4", correct: true },
      { text: "22", correct: false },
    ],
  },
  {
    question: "Is web development fun?",
    answers: [
      { text: "Kinda", correct: false },
      { text: "YES!!!", correct: true },
      { text: "Um no", correct: false },
      { text: "IDK", correct: false },
    ],
  },
  {
    question: "What is 4 * 2?",
    answers: [
      { text: "6", correct: false },
      { text: "8", correct: true },
      { text: "Yes", correct: false },
    ],
  },
];
```

startGame

Creamos la función **startGame** para empezar el Quiz:



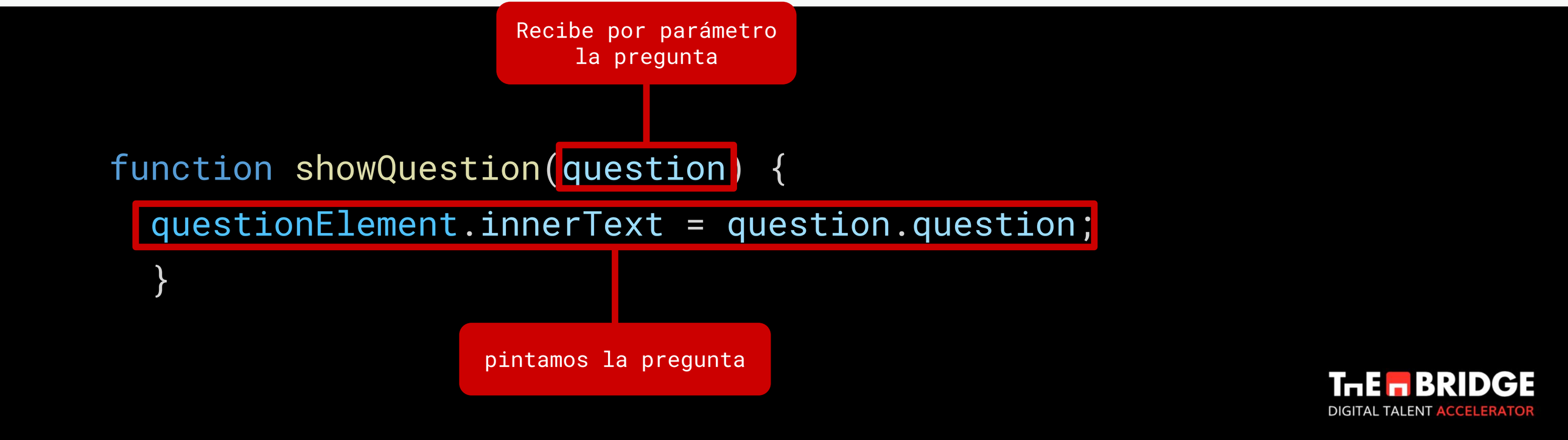
addEventListener

Ahora le decimos que cuando cliquen en el **botón start** se ejecute la función **startGame**:

```
startButton.addEventListener("click", startGame);
```

showQuestion

Creamos la función **showQuestion(),** esta función se encargará de mostrar en el HTML la pregunta en la que estemos, lo primero que haremos será mostrar el enunciado de la pregunta:



setNextQuestion()

Creamos la función **setNextQuestion()** que recibirá por parámetro la pregunta correspondiente con sus respuestas:

```
function setNextQuestion() {
  showQuestion(questions[currentQuestionIndex]);
}
```

startGame

Actualizamos la función **startGame**:

```
function startGame() {
  startButton.classList.add("hide");
  currentQuestionIndex = 0;
  questionContainerElement.classList.remove("hide");
  setNextQuestion();
}
```

le decimos que llame a la función **setNextQuestion** para que pinte la pregunta

showQuestion

Continuamos en la función y ahora vamos a hacer la lógica para mostrar las respuestas a la pregunta:

```
function showQuestion(question) {
. . .
question.answers.forEach((answer) => {
    const button = document.createElement("button");
    button.innerText = answer.text;

    if (answer.correct) {
      button.dataset.correct = true;
    }

    answerButtonsElement.appendChild(button);
  });
}
```

creamos el botón para las respuestas

pintamos las respuestas

si la respuesta es correcta al botón de esa respuesta le añadimos el atributo correct que será igual a true

le decimos que nos pinte los botones de las respuestas en el html

showQuestion

La función

showQuestion()

completa:

```
function showQuestion(question) {
  questionElement.innerText = question.question;
  question.answers.forEach((answer) => {
    const button = document.createElement("button");
    button.innerText = answer.text;

    if (answer.correct) {
      button.dataset.correct = true;
    }

    answerButtonsElement.appendChild(button);
  });
}
```

Recibe por parámetro la pregunta

pintamos la pregunta

creamos el botón para las respuestas

pintamos las respuestas

si la respuesta es correcta al botón de esa respuesta le añadimos el atributo correct que será igual a true

le decimos que nos pinte los botones de las respuestas en el html

CSS

Creamos las clases correct y wrong:

```
.correct {
  background-color: rgb(0, 255, 72);
}

.wrong {
  background-color: rgb(255, 0, 0);
}
```

setStatusClass()

Vamos a crear la función **setStatusClass()** que lo que hará es que si el elemento clicado es correcto le añadirá la clase de CSS **correct** y sino le añadirá la clase de CSS **wrong:**

```
function setStatusClass(element) {
  if (element.dataset.correct) {
    element.classList.add("correct");
  } else {
    element.classList.add("wrong");
  }
}
```

selectAnswer()

En esta función recorremos los botones de las respuestas y se lo pasamos por parámetro a la función **setStatusClass()** Además **creamos un if** que chequee que si la pregunta actual no es la última aparezca el **botón next** y sino que aparezca el **boton start**.

recorremos los botones de las respuestas

```
function selectAnswer() {
  Array.from…
    answerButtonsElement.childNodes.forEach((button) => {
      setStatusClass(button);
    });
    if (questions.length > currentQuestionIndex + 1) {
      nextButton.classList.remove("hide");
    } else {
      startButton.innerText = "Restart";
      startButton.classList.remove("hide");
    }
}
```

showQuestion

Actualizamos la función **showQuestion()**:

```
function showQuestion(question) {
  questionElement.innerText = question.question;
  question.answers.forEach((answer) => {
    const button = document.createElement("button");
    button.innerText = answer.text;
    if (answer.correct) {
      button.dataset.correct = true;
    }
    button.addEventListener("click", selectAnswer);
    answerButtonsElement.appendChild(button);
  });
}
```

le decimos que cuando clique el botón ejecute la función

nextButton addEventListener

Ahora le decimos que cuando clique en el botón next sume 1 a **currentQuestionIndex** y se ejecute la función **setNextQuestion** que acabamos de crear:

```
nextButton.addEventListener("click", () => {
  currentQuestionIndex++;
  setNextQuestion();
});
```

resetState()

Creamos la función **resetState()**, básicamente lo que hará es que borrará las respuestas escritas:

```
function resetState() {
  nextButton.classList.add("hide");
  answerButtonsElement.innerHTML=""
}
```

escondemos el botón next

setNextQuestion()

Actualizamos la función **setNextQuestion()** y le pasamos la función **resetState**:

```
function setNextQuestion() {
  resetState();
  showQuestion(questions[currentQuestionIndex]);
}
```