5 JavaScript coding Exercises 2

FizzBuzz

Source code:

```
for (let i = 1; i <= 100; i++) {
  if (i % 3 === 0 && i % 5 === 0) {
    console.log('FizzBuzz');
  } else if (i % 3 === 0) {
    console.log('Fizz');
  } else if (i % 5 === 0) {
    console.log('Buzz');
  } else {
    console.log(i);
  }
}
```

Explanation:
The FizzBuzz exercise involves iterating from 1 to 100 and printing out "Fizz" for multiples of 3, "Buzz" for multiples of 5, and "FizzBuzz" for multiples of both 3 and 5. We can use the modulus operator to check if a number is a multiple of another number, and use if/else statements to determine what to print.

Palindrome checker

Source code:

```
function isPalindrome(str) {
  const len = str.length;
  for (let i = 0; i < len / 2; i++) {
    if (str[i] !== str[len - 1 - i]) {
      return false;
    }
  }
  return true;
}

console.log(isPalindrome('racecar')); // true
console.log(isPalindrome('hello')); // false
```

Explanation:

The palindrome checker exercise involves writing a function that checks if a given string is a palindrome. We can iterate over the first half of the string and compare each character to the corresponding character from the end of the string. If any characters don't match, we can return false. If we get through the entire loop without finding any mismatches, we can return true.

Hangman game

Source code:

```
const words = ['apple', 'banana', 'cherry',
'dragonfruit', 'elderberry'];

function getRandomWord() {
```

```
  const index = Math.floor(Math.random() *
words.length);
  return words[index];
}

function hideWord(word) {
  let hiddenWord = '';
  for (let i = 0; i < word.length; i++) {
    hiddenWord += '_';
  }
  return hiddenWord;
}

function playHangman() {
  const word = getRandomWord();
  let hiddenWord = hideWord(word);
  let remainingGuesses = 6;
  while (remainingGuesses > 0) {
    console.log(`Word: ${hiddenWord}`);
    console.log(`Guesses remaining:
${remainingGuesses}`);
    const guess = prompt('Guess a letter:');
    if (!word.includes(guess)) {
      remainingGuesses--;
    }
    for (let i = 0; i < word.length; i++) {
      if (word[i] === guess) {
        hiddenWord = hiddenWord.slice(0, i) + guess +
hiddenWord.slice(i + 1);
      }
    }
    if (hiddenWord === word) {
```

```
      console.log(`You win! The word was ${word}`);
      return;
    }
  }
  console.log(`You lose! The word was ${word}`);
}

playHangman();
```

Explanation:
The hangman game exercise involves creating a simple game where the computer chooses a random word and the player has to guess the letters of the word before running out of guesses. We can use an array of words to choose from, and prompt the user for their guesses. We can keep track of the remaining guesses and update the hidden word as the user guesses correct letters.

Shopping cart

Source code:

```
const products = [  { name: 'Apple', price: 1.99 },  { name: 'Banana', price: 0.99 },  { name: 'Cherry', price: 2.99 },  { name: 'Dragonfruit', price: 4.99 },  { name: 'Elderberry', price: 3.99 },];

function addToCart(cart, product) {
  cart.push(product);
}

function removeFromCart(cart, productName) {
```

```
  const index = cart.findIndex(item => item.name ===
productName);
  if (index >= 0) {
    cart.splice(index, 1);
  }
}

function calculateTotal(cart) {
  let total = 0;
  for (let i = 0; i < cart.length; i++) {
    total += cart[i].price;
  }
  return total.toFixed(2);
}

const shoppingCart = [];
addToCart(shoppingCart, products[0]);
addToCart(shoppingCart, products[2]);
addToCart(shoppingCart, products[4]);
console.log(shoppingCart);
removeFromCart(shoppingCart, 'Banana');
console.log(shoppingCart);
console.log(`Total: $${calculateTotal(shoppingCart)}`);
```

Explanation:

The shopping cart exercise involves creating a program that allows the user to add and remove items from their cart, calculate the total price, and apply discounts or coupons. We can use an array to represent the shopping cart, and objects to represent the products. We can define functions for adding and removing items from the cart, and for calculating the total price by iterating over the cart array and adding up the prices of the items. We can also

use the toFixed method to ensure that the total is displayed with two decimal places.

Typing speed test

Source code:

```
const paragraph = 'The quick brown fox jumps over the
lazy dog';
let startTime;
let endTime;

function startTest() {
  startTime = Date.now();
  document.getElementById('prompt').innerHTML =
paragraph;

document.getElementById('input').addEventListener('inpu
t', checkInput);
}

function checkInput() {
  const typedText =
document.getElementById('input').value;
  if (typedText === paragraph) {
    endTime = Date.now();
    const totalTime = (endTime - startTime) / 1000;
    const speed = Math.round(paragraph.length /
totalTime);
    document.getElementById('result').innerHTML = `You
typed at a speed of ${speed} characters per second!`;
```

```
document.getElementById('input').removeEventListener('i
nput', checkInput);
  }
}

document.getElementById('start-button').addEventListene
r('click', startTest);
```

Explanation:
The typing speed test exercise involves creating a program that measures the user's typing speed by having them type a given paragraph as quickly and accurately as possible. We can use HTML and CSS to create a simple user interface with a start button, a prompt to type, an input field to type into, and a result section to display the user's speed. We can use the Date.now() method to record the start and end times of the test, and the addEventListener method to listen for input events in the input field. We can then compare the typed text to the prompt text and calculate the speed by dividing the length of the prompt by the total time in seconds. Finally, we can update the result section with the user's speed and remove the event listener to stop listening for input events.