КОНТРОЛЬНЫЕ ВОПРОСЫ

для подготовки к итоговой аттестации

«Fullstack-программирование»

профессиональная переподготовка (дополнительное профессиональное образование)

(наименование программы)

1. Что из себя представляет клиентский язык JavaScript. Основные особенности.

JavaScript - это интерпретируемый клиентский язык программирования, который используется для создания динамических и интерактивных веб-приложений. Он может бытьоен непосредственно в HTML-документы и выполняться веб-браузером на стороне клиента пользователя. Вот некоторые основные особенности языка JavaScript:

1. Динамическая типизация: JavaScript является слабо типизированным языком, что означает, что переменные могут хранить значения разных типов данных, и их типы могут изменяться в процессе выполнения программы.

Пример:

let age = 25; // число

age = "двадцать пять"; // строка

1. Объектно-ориентированное программирование (ООП): JavaScript поддерживает основные принципы ООП, такие как наследование, инкапсуляция и полиморфизм. Он позволяет создавать объекты и работать с ними, используя методы и свойства.

Пример:

**class** **Person** {

**constructor**(name) {

this.name = name;

}

**sayHello**() {

console.**log**(`Привет, меня зовут ${this.name}!`);

}

}

**const** person = **new** **Person**("Дмитрий");

person.**sayHello**(); // Выведет "Привет, меня зовут Дмитрий!"

1. Асинхронное программирование: JavaScript имеет возможность выполнять задачи асинхронно, что позволяет обрабатывать события и запросы без блокировки основного потока выполнения. Это делает его идеальным для создания интерактивных веб-приложений.

Пример:

setTimeout(() => {

console.log("Это отложенное действие!");

}, 2000); // Выведет "Это отложенное действие!" через 2 секунды

1. Широкая поддержка: JavaScript поддерживается всеми популярными веб-браузерами и может выполняться на различных операционных системах. Он также становится все более популярным в серверной разработке с помощью платформы Node.js.

Это лишь несколько основных особенностей языка JavaScript. Он предоставляет разработчикам широкие возможности для создания интерактивных и эффективных веб-приложений.

2. Дана строка. Сделайте заглавным первый символ этой строки не используя цикл. Найдите

два решения.

Способ 1: использование CSS и псевдоэлемента ::first-letter

<style>

.capitalize::first-letter {

text-transform: uppercase;

}

</style>

<p class="capitalize">hello world</p>

В этом способе мы применяем стиль к псевдоэлементу , который выбирает первый символ строки и применяет к нему превращение текста в заглавные буквы с помощью свойства .::first-lettertext-transform: uppercase;

Способ 2: использование JavaScript и метода toLocaleUpperCase()

<script>

**let** str = "hello world";

**let** result = str[0].**toLocaleUpperCase**() + str.**slice**(1);

console.**log**(result); // Выведет "Hello world"

</script>

3. Напишите фрагмент кода, в котором переменным а и b будут присваиваться числовые

значения пользователем, с помощью команды (prompt()). Далее необходимо проверить, что

число попадает в диапазон "однозначное положительное число". Затем следует присвоить

переменной с значение, составленное следующим образом: значение переменной а

является числом десятков, значение переменной b является числом единиц. Например,

если переменной а присваивается значение 4, а переменной b – значение 7, то переменная с

получает значение 47. Выведете результат на экран.

**let** a = parseInt(**prompt**("Введите число для переменной a"));

**let** b = parseInt(**prompt**("Введите число для переменной b"));

// Проверяем, что числа попадают в диапазон "однозначное положительное число"

**if** (a >= 0 && a < 10 && b >= 0 && b < 10) {

**let** c = a \* 10 + b;

console.**log**(c); // Выводим результат на экран

} **else** {

console.**log**("Неверный ввод чисел");

}

1. Дайте определения инструкции и функции в JavaScript.

В JavaScript инструкция и функция являются двумя основными концепциями, которые используются для организации и выполнения кода. Вот их определения:

Инструкция (statement):  
Инструкция - это отдельная команда или действие, которое может быть выполнено интерпретатором JavaScript. Инструкции являются строительными блоками программы и выполняют определенную операцию. Примеры инструкцийают присваивания значений переменным, условные выражения, циклы и вызовы функций. Программы состоят из набора инструкций, которые выполняются по порядку сверху вниз. Пример инструкции присваивания значения переменной:

let x = 5; // инструкция присваивания значения переменной

Функция:  
Функция - это блок кода, который выполняет определенную задачу и может быть вызван из других частей программы. Функции в JavaScript позволяют организовать код в логические блоки, которые могут быть многократно использованы. Функция принимает аргументы (входные значения), выполняет определенные операции и может возвращать результат. Определение функции включает имя функции, список параметров и тело функции, содержащее код. Пример определения и вызова функции:

**function** **multiply**(**a**, **b**) { // определение функции

**return** **a** \* b; // возврат результата

}

let result = multiply(3, 4); // вызов функции

console.log(result); // вывод результата на экран

В этом примере функция принимает два параметра и , выполняет операцию умножения и возвращает результат. При вызове функции с аргументами 3 и 4, переменная получает значение 12, которое выводится на экран с помощью . Функции позволяют создавать модульный и переиспользуемый код, что делает их важной концепцией в JavaScript. multiplyabresultconsole.log()

5. Кинетическая энергия тела подсчитывается как половина произведения массы тела на

квадрат скорости тела E=(mc2)/2. Напишите программу, которая принимает с клавиатуры

два числа, первое из которых – масса тела, а второе – его скорость. Программа должна

подсчитать и вывести на экран значение кинетической энергии тела.

// Считываем массу тела с клавиатуры

**var** mass = parseFloat(**prompt**("Введите массу тела:"));

// Считываем скорость тела с клавиатуры

**var** velocity = parseFloat(**prompt**("Введите скорость тела:"));

// Вычисляем кинетическую энергию

**var** kineticEnergy = (mass \* velocity\*\*2) / 2;

// Выводим значение кинетической энергии на экран

console.**log**("Кинетическая энергия тела: " + kineticEnergy);

6. В таблице users имеются поля Фамилия, Имя, Отчество, Возраст, Номер телефона.

Напишите SQL запрос который будет искать в таблице Иванова Игоря Петровича и

возвращать только номер телефона данной записи.

**SELECT** `Номер телефона`

**FROM** `users`

**WHERE** `Фамилия` = 'Иванова' **AND** `Имя` = 'Игорь' **AND** `Отчество` = 'Петрович';

7. Дайте определение массива и объекта в JavaScript.

Массив (Array):  
В JavaScript, массив - это упорядоченная коллекция элементов, которая может содержать различные типы данных, такие как числа, строки, объекты и другие массивы. Массивы в JavaScript индексируются с нуля, то есть первый элемент имеет индекс 0, второй - инд 1 и так далее. Пример объявления массива в JavaScript:

var fruits = ['яблоко', 'банан', 'апельсин'];

Объект (Object):  
В JavaScript, объект - это неупорядоченная коллекция пар "ключ-значение", где ключи являются строками или символами, а значения могут быть любого типа данных, включая числа, строки, массивы, другие объекты и функции. Объекты в JavaScript позволяют упорядочивать и группировать данные и представляют основу для работы с объектно-ориентированным программированием в JavaScript. Пример объявления объекта в JavaScript:

var person = {

name: 'John',

age: 30,

city: 'New York'

};

В этом примере объект "person" имеет три свойства: "name" со значением 'John', "age" со значением 30 и "city" со значением 'New York'. Доступ к свойствам объекта осуществляется через точку или через квадратные скобки. Например, чтобы получить значение свойства "name" объекта "person", можно использовать выражение "person.name" или "person['name']".

8. Напишите код, который принимает от пользователя фамилию, имя и отчество, а затем

выводит на экран фамилию и инициалы (без пробела между инициалами).

// Считываем фамилию, имя и отчество с помощью функции prompt()

**var** lastName = prompt("Введите фамилию:");

**var** firstName = prompt("Введите имя:");

**var** middleName = prompt("Введите отчество:");

// Получаем первую букву имени и первую букву отчества

**var** firstNameInitial = firstName.charAt(0);

**var** middleNameInitial = middleName.charAt(0);

// Формируем строку с фамилией и инициалами, без пробела между инициалами

**var** fullName = lastName + firstNameInitial + middleNameInitial;

// Выводим результат на экран

console.log("Фамилия и инициалы: " + fullName);

9. 1) Напишите функцию, которая принимает 2 параметра и возвращает сумму этих чисел.2)

Вызовите функцию передав на вход два любых числа.3) Выведите на экран результат

работы функции.

function sum(a, b) {

return a + b;

}

// Вызов функции и вывод результата на экран

var result = sum(10, 7);

console.log("Результат сложения: " + result);

sum принимает два параметра a и b, складывает их с помощью оператора + и возвращает результат. Затем мы вызываем функцию, передавая ей два любых числа - 10 и 7, и сохраняем результат в переменной result. Наконец, мы выводим значение переменной result на экран с помощью console.log(). Результатом будет число 17, так как 10 + 7 = 17.

10. Что такое регулярные выражения и даты в JavaScript.

Регулярные выражения (Regular Expressions) и работа с датами являются двумя важными аспектами JavaScript.

Регулярные выражения:  
В JavaScript, регулярные выражения представляют собой мощный инструмент для работы сом. Регулярные выражения позволяют искать, сопоставлять и изменять текст, основываясь на шаблонах. С помощью регулярных выражений можно проверять соответствие строки определенному формату, извлекать подстроки из текста, заменять части текста и многое другое. В JavaScript, для работы сегуляными выражениями используется встроенный объект RegExp или синтаксис литералов /pattern/.

Пример использования регулярного выражения для проверки, соответствует ли строка шаблону email:

var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

var email = "example@example.com";

var isValidEmail = emailPattern.test(email);

console.log(isValidEmail); // Выведет true

В этом примере, переменная emailPattern содержит регулярное выражение, которое проверяет, соответствует ли строка шаблону email. Метод test() объекта RegExp применяется к строке email для проверки, является ли она валидным email адресом. Результат проверки выводится на экран с помощью console.log().

Работа с датами:  
В JavaScript, работа с датами осуществляется с помощью встроенного объекта Date. Объект Date предоставляет методы и свойства для работы с датами и временем. Он позволяет создавать новые объекты Date, устанавливать и получать значения года, месяца, дня, часа, минут, секунд и миллисунд, а также выполнять операции с датами, такие как вычитание и сложение.

Пример использования объекта Date для получения текущей даты:

var currentDate = new Date();

console.log(currentDate); // Выведет текущую дату и время

В этом примере, мы создаем новый объект Date с помощью оператора new Date(), который автоматически устанавливает текущую дату и время Затем мы выводим значение этого объекта на экран с помощью console.log().

Объект Date также предоставляет множество методов для работы с датами, таких как getFullYear() для получения года, getMonth() для получения месяца (от 0 до 11), getDate() для получения дня и т.д.же можно выполнять операции сложения и вычитания дат с помощью методов setFullYear(), setMonth(), setDate() и других.

11. Напишите фрагмент кода (группу команд), который будет присваивать значения 5 и 9 двум

переменным, а затем третья переменная будет получать значение, равное сумме значений

первых двух переменных. Вывести на экран значение третьей переменной.

12. Назовем «весом числа» сумму его цифр. Напишите код, который принимает от

пользователя число и выводит на экран его «вес».

13. Что такое глобальный объект Window?

14. Что такое Fetch?

15. Напишите код, который принимает с клавиатуры целое положительное двузначное число и

выводит на экран его цифры, разделенные знаком «пробел».

16. Назовите основные характеристики объекта Document.

17. Дан массив [1, 2, 3]. Сделайте из него массив [3, 2, 1].

18. Дана кнопка. По нажатии на эту кнопку выведите широту окна браузера.

19. Что такое Event Loop?

20. Выведите с помощью цикла столбец чисел от 100 до 1.

21. Создайте функцию inArray, которая определяет, есть в массиве элемент с заданным

текстом или нет. Функция первым параметром должна принимать текст элемента, а

вторым - массив, в котором делается поиск. Функция должна возвращать true или false.

22. Назовите основные элементы HTML.

23. Дана строка 'aaa bbb ccc'. Вырежете из нее слово 'bbb' тремя разными способами (через

substr, substring, slice).

24. Получите url-адрес текущей страницы и проверьте, что строка заканчивается на .php.

25. Что такое объектная модель документа?

26. Дан массив с числами. Проверьте, есть ли в нем два одинаковых числа подряд. Если есть -

выведите 'да', а если нет - выведите 'нет'.

27. Составьте SQL запрос который получит все данные из таблицы tasks.

28. Как отменить стандартное действие события?

29. Сделайте функцию, которая отнимает от первого числа второе и делит на третье.

30. Даны переменные a = 10 и b = 3. Найдите остаток от деления a на b.

31. Дано число, например 31. Проверьте, что это число не делится ни на одно другое число

кроме себя самого и единицы. То есть в нашем случае нужно проверить, что число 31 не

делится на все числа от 2 до 30. Если число не делится - выведите 'false', а если делится -

выведите 'true'.

32. Даны картинки. Привяжите к каждой картинке событие, чтобы по клику на картинку alert

выводился ее src.

33. Дан массив ['Привет, ', 'мир', '!']. Необходимо записать в нулевой элемент этого массива

слово 'Пока, ' (то есть вместо слова 'Привет, ' будет 'Пока, ').

34. Создайте функцию getDigitsSum (digit - это цифра), которая параметром принимает целое

число и возвращает сумму его цифр.

35. Что из себя представляет серверный язык PHP?

36. Напишите фрагмент кода, который принимает от пользователя число и если число четное

то программа выводит на экран число, которое больше введенного с клавиатуры на 10, в

противном случае выводит на экран число в 10 раз больше введенного.

37. Дан массив с числами. Проверьте, есть ли в нем два одинаковых числа подряд. Если есть -

выведите 'да', а если нет - выведите 'нет'.

38. Какие циклы серверного языка PHP Вы знаете?

39. Дана строка вида 'var\_text\_hello'. Сделайте из него текст 'varTextHello'.

40. В таблице users имеются поля Фамилия, Имя, Отчество, Возраст, Номер телефона.

Напишите SQL-запрос который будет искать в таблице Иванова Игоря Петровича и

возвращать все поля данной записи.

41. Что из себя представляет сервер баз данных MySQL?

42. Дано число. Сложите его цифры. Если сумма получилась более 9-ти, опять сложите его

цифры. И так, пока сумма не станет однозначным числом (9 и менее).

43. Создайте функцию, которая параметрами принимает 2 числа. Если их сумма больше 10 -

пусть функция вернет true, а если нет - false.

44. Что такое сookie?

45. Получите url-адрес текущей страницы и проверьте, что строка начинается на http://.

46. Даны два массива: [1, 2, 3] и [4, 5, 6]. Объедините их вместе.

47. Назовите различия Microtask и Macrotask.

48. Дана строка 'js'. Сделайте из нее строку 'JS'.

49. С помощью цикла for сформируйте строку '-1-2-3-4-5-6-7-8-9-' и запишите ее в

переменную str.

50. Назовите разницу между префиксной формой и постфиксной формой инкремента

51. Дана кнопка. По нажатии на эту кнопку выведите высоту окна браузера.

52. Напишите функцию, которая принимает в качестве параметра число, и возвращает квадрат

этого числа.

53. Какие шаблоны проектирования сложных веб-проектов на РНР Вы знаете?

54. Даны переменные $a = 15 и $b = 6. Найдите остаток от деления a на b.

55. Составьте SQL запрос который получит все данные из таблицы tasks

56. Чем String отличается от Array?

57. Напишите программу, которая выводит на экран надпись: "Хочешь - решай задачи, не

хочешь - решай нехотя" 17 раз каждый раз с новой строки, строки должны быть

пронумерованы.

58. 1) Создайте массив чисел. 2) Добавь в массив 10 чисел с клавиатуры. 3) Вывести на экран

длину самой длинной последовательности повторяющихся чисел в списке. Пример для

списка <i>2, 4, 4, 4, 8, 8, 4, 12, 12, 14</i>:3.

Напишите фрагмент кода, в котором переменным а и b будут присваиваться числовые

значения пользователем, с помощью команды (prompt()). Далее необходимо проверить, что

число попадает в диапазон "однозначное положительное число". Затем следует присвоить

переменной с значение, составленное следующим образом: значение переменной а

является числом десятков, значение переменной b является числом единиц. Например,

если переменной а присваивается значение 4, а переменной b – значение 7, то переменная с

получает значение 47. Выведете результат на экран

Конечно! Вот фрагмент кода, который выполняет описанную вами задачу:

**let** a = parseInt(**prompt**("Введите число для переменной a"));

**let** b = parseInt(**prompt**("Введите число для переменной b"));

// Проверяем, что числа попадают в диапазон "однозначное положительное число"

**if** (a >= 0 && a < 10 && b >= 0 && b < 10) {

**let** c = a \* 10 + b;

console.**log**(c); // Выводим результат на экран

} **else** {

console.**log**("Неверный ввод чисел");

}

В этом коде мы используем функцию для преобразования строк, введенных пользователемез , в числа. Затем мы проверяем, что оба числа находятся в диапазоне "однозначное положительное число" с помощью условного оператора . Если числа проходят проверку, мы выполняем операцию умножения и сложения для получения значения переменной , а затем выводим результат на экран с помощью . Если введенные числа не соответствуют требованиям, выводится сообщение об ошибке.parseInt()prompt()ifcconsole.log()

После запуска этого кода, пользователь увидит два диалоговых окна , в которых нужно ввести числа для переменных и . Затем код выполняет проверку и выводит результат на экран.prompt()ab