JS Fundamentals QUIZ

# Introduction to JacaScript

1. **JavaScript може да се определи като:**

* Език от ниско ниво
* Компилиран език
* Език от високо ниво

1. **В каква среда може да бъде изпълнен JavaScript?** (изберете всички подходящи отговори)

* На сървър чрез Node.js
* Директно в браузъра
* Аналогови водомери
* Вградени системи

1. **Езика JavaScript спада към групата на C-производните езици.**

* Вярно
* Невярно

1. **За какво служи оператора let?**

* За промяна стойността на съществуваща променлива
* За деклариране на променлива
* За дефиниране на условна конструкция

1. **Коя е ключовата дума за деклариране на функция?**

* method
* proc
* void
* function

1. **За какво служи символа backtick (`) при декларирането на низ (string)?**

* За ескейпване на контролни символи
* За създаване на обикновен низ
* За създаване на интерполиран низ
* За отваряне на конзолата

1. **Метода toFixed() връща форматиран низ при изпълняние върху число.**

* Вярно
* Невярно

1. **Кой оператор за сравнение взима предвид типа на данните?**

* <=>
* ==
* ===
* !=

1. **Какъв ще е резултата при изпълнение на следния код:**

let a = 11;

let b = “11”;

if(a == b) {

console.log(“Yes”);

} else {

console.log(“No”);

}

* Ще отпечата "Yes" в конзолата
* Ще даде грешка при компилация
* Ще отпечата "No" в конзолата
* Ще се изпълни без да отпечата нищо

1. **В JavaScript има spaceship-оператор <=>**

* Вярно
* Невярно

1. **Кои ключови думи използваме за създаване на условна конструкция?**

* do-unless
* if-else-finally
* if-else
* if-then-else

1. **С кой оператор можем да заменим поредица от свързани if-else конструкции, които проверяват стойността на една и съща променлива?**

* ifs
* aggregate
* multiplex
* switch

1. **За какво служи логическия оператор !?**

* Обръща значението на Булева стойност (Logical NOT)
* Логически сбор на две Булеви стойности (Logical AND)
* Ексклузивен сбор между две двоични стойности (Binary XOR)

1. **Как наричаме следната конструкция:**

for (let i = 1; i <= 5; i++){

console.log(i);

}

* Блок-схема
* Условна конструкция
* Цикъл
* Функция

1. **Тялото на цикъл от тип do-while ще се изпълни поне веднъж, без значение дали условието за изпълнение е истинно.**

* Вярно
* Невярно

1. **Как наричаме нежелано или неочаквано повечедение на приложенията?**

* Регресия
* Feature
* Своеволие
* Бъг

1. **При достигане на breakpoint, дебъгера:**

* ще прескочи реда код, за който се отнася breakpoint-а
* ще спре временно изпълнението на приложението, за да можем да инспектираме текущия резултатт
* ще си вземе почивка
* ще прекрати изпълнението на приложението

# Data Types and Variables

1. **Данните в JavaScript нямат тип**

* Вярно
* Невярно

1. **Кой е типа данни за управление на целочислени стойности с произволен (неограничен) размер?**

* ArbitraryInt
* BigInt
* number
* int

1. **Типа на променливите в JavaScript може да бъде променен по време на изпълнение**

* Вярно
* Невярно

1. **Кога е подходящо използванто на var за деклариране на променлива?**

* Никога
* При по-големи приложения
* При по-малки приложения
* Винаги

1. **Какъв ще е резултата от изпълнението на следния код:**

**{**

**let x = 8;**

**}**

**console.log(x);**

* Грешка при изпълнение
* Ще се отпечата undefined в конзолата
* Ще се изпълни без да отпечатва нищо
* Ще се отпечата 8 в конзолата

1. **Кой оператор връща типа на дадена стойност?**

* typeof
* instanceof
* &&
* Is

1. **На коя позиция (индекс) се намира първия символ в типа данни string?**

* -1
* 0
* 1
* length – 1

1. **Стойностите от тип string са immutable (непроменими).**

* Вярно
* Невярно

1. **Какво ще отпечата на конзолата следния код:**

**let name = ‘Peter’;**

**let age = 23;**

**console.log(`$(name) = ${age}`);**

* ${name} = ${age}
* Peter = 23
* ${Peter} = ${23}
* Няма да отпечата нищо

1. **Колко числови типа с плаваща запетая има в JavaScript?**

* 1
* 2
* 3
* 4

1. **Операцията на кой ред ще резултира в стойност NaN (not-a-number)?**

**let a = 5;**

**let b = 3.14;**

**let c = ‘p’;**

**console.log(a + b);**

**console.log(a + c);**

**console.log(Number(c));**

* 3
* 4
* 5
* 6

1. **Коя от следните стойности НЕ Е falsy:**

* 0
* false
* празен масив
* undefined
* празен низ

1. **Какъв тип ще получим чрез оператора typeof за променлива без зададена стойност?**

* NaN
* null
* undefined
* 0

# Arrays

1. **Кой е индекса на първия елемент в масив?**

* -1
* 0
* 1
* 2

1. **Дължината на масивите в JavaScript може да се променя по време на изпълнение.**

* Вярно
* Невярно

1. **Посочете реда с правилна декларация на масив:**

* let myArr = [1, 2, 3]
* let myArr = number[]{1, 2, 3}
* let myArr = {1, 2, 3}

1. **Какво съдържа свойството на масива length?**

* Индекса на последния индекс в масива
* Броя елементи в масива
* Размера на масива в паметта на компютъра в битове
* Максималния брой елементи, които могат да се добавят в масива

1. **Достъпването на несъществуващ индекс хвърля грешка.**

* Вярно
* Невярно

1. **Как се нарича оператора за достъпване на конкретен елемент в масива []?**

* Индексиращ оператор
* Итератор
* Разпъващ оператор

1. **Кой метод служи за добавяне на нов елемент в края на масива?**

* push
* add
* pop
* pull

1. **Какво ще отпечата изпълнението на следния код:  
   let arr = [“a”, “b”,”c”];  
   arr[4] = “e”  
   console.log(arr[3]);**

* undefined
* e
* c
* null
* d

1. **Стойностите, зададени на невалиден индекс (различен от цяло неотрицателно число) в масив не променят неговата дължина.**

* Вярно
* Невярно

1. **Какъв е параметъра, който се подава на метода join()?**

* Масив, чиито елементи да бъдат съединени
* Разделител между елементите в генерирания низ
* Индекса на елемента, от който да започне съединяването

1. **Какъв е типа на стойността, върната от метода includes()?**

* Зависи от типа на елементите в масива
* number
* string
* boolean

1. **Изберете липсващия фрагмент, с който написаната програма да отпечата в конзолата последовтелно всички елементи на масива:  
   let arr = [10, 20, 30, 40];  
   for(let i = 0; \_\_\_\_\_\_\_; i++){  
    console.log(arr[i]);  
   }**

* i < arr.length - 1
* i <= arr.length
* i < arr.length
* i > arr.length

1. **С кой вид цикъл можем най-лесно да обходим елементите в масив?**

* do-while
* for-in
* for-each
* for-of

1. **При обхождане на масива с цикъл for-of нямаме пряк достъп до индексите на елементите.**

* Вярно
* Невярно

1. **Изберете липсващия фрагмент, с който написаната програма да сумира само четните стойности в масива:  
   let arr = [7, 1, 4, 3, 12];  
   let sum = 0;  
   for(let num of arr){  
    if(\_\_\_\_\_\_\_\_\_\_){  
    sum += num;  
    }  
   }  
   console.log(sum);**

* num % 2 != 0
* 2 % num == 0
* num % 2 == 0
* arr[i] % 2 == 0

# Functions

asd

# Arrays Advanced

asd

# Objects and Classes

asd

# Associative Arrays

asd

# Text Processing

asd