JSF – Part 1

Exercises

The contained in this document exercises are related to the JavaScript Foundations – Part1 training course. The exercises are an integral part of the JSF lectures, designed to train IT Technical Consultants.

The following naming convention is followed: the first word (the prefix) points to the material presented in the relevant lecture, followed by a subsequent tasks' numbers.

Table of Contents

Document History

Document version:	**v.1.0**
Document date:	**25.04.2024**
	Initial version
Document version:	**v.1.1**
Document date:	**26.04.2024**
	Added: BASICS03-001 – BASICS03-020
Document version:	**v.1.2**
Document date:	**29.04.2024, 23:22h**
	Added: BASICS02-014, BASICS02-015, BASICS03-021, BASICS03-022 Added Document History Added Tasks/Exercises counters.
Document version:	**v.1.3**
Document date:	**08.05.2024, 21:51h**
	Added: DTYPES01-001 – DTYPES01-023 Added: DTYPES02-001 – DTYPES02-nnn

Tasks/Exercises Count

BASICS01	11 tasks
BASICS02	15 tasks
BASICS03	22 tasks
DTYPES01	23 tasks

Basics 01 – Names, Variables

BASICS01-001: Write ten correct identifiers, following the camelCase naming convention.

BASICS01-002: Imagine, you are solving a math problem. Declare ten variables, which you might need in your program.

BASICS01-003: Imagine, you are working for a cloud provider and are responsible for the servers. You must write a program to list and describe the servers. Declare ten variables, which you might need in your program.

BASICS01-004: Print on the console five alphabet characters, five numbers, five punctuation characters.

BASICS01-005: Declare five variables, assign some numbers, and print them on the console.

BASICS01-006: Declare ten variables, assign the numbers from 1 to 10 and print the even numbers on the console.

BASICS01-007: Declare ten variables, assign the numbers from 1 to 10 and print the first three odd numbers on the console.

BASICS01-008: Declare ten variables, assign the numbers from 100 to 109 and print the last two odd numbers on the console.

BASICS01-009: Declare five variables, assign the first five prime numbers, and print them on the console.

BASICS01-010: Declare ten variables. On the first five - assign the first five prime numbers. On the second five numbers, do the same, but multiply each value by 3. Print all of them on the console.

BASICS01-011: Declare ten variables. Assign them the first ten prime numbers. Print the numbers in reverse order.

Basics 02 – Character Set

BASICS02-001: Declare five variables. Assign them with the ASCII codes of the first five English capital alphabet characters. Print them on the console.

BASICS02-002: Declare five variables. Assign them with the ASCII codes of the last five English lowercase alphabet characters. Print them on the console.

BASICS02-003: Declare five variables. Assign them with the ASCII codes of randomly chosen punctuation characters. Print them on the console.

BASICS02-004: Declare five variables. Assign them with the UNICODE codes of randomly chosen emoji characters. Print them on the console – on different lines.

BASICS02-005: Declare five variables. Assign them with the UNICODE codes of randomly chosen emoji characters. Print them on the console – on one line, separated with four spaces.

BASICS02-006: Declare five variables. Assign them with the randomly chosen emoji characters. Print the UNICODE codes on the console – on different lines.

BASICS02-007: Declare five variables. Assign them with the randomly chosen emoji characters. Print the UNICODE codes on the console – on one line, separated with commas and space after each comma character.

BASICS02-008: Declare five variables. Assign them with the randomly chosen emoji characters. Print the UNICODE codes in hex format on the console – on different lines.

BASICS02-009: Declare five variables. Assign them with the randomly chosen emoji characters. Print the UNICODE codes in decimal format on the console – on different lines.

BASICS02-010: Declare five variables. Assign them with the randomly chosen emoji characters. For each of the variables - print the UNICODE code in binary, octal, decimal, and hex format on one line, separated with comas and space after it.

BASICS02-011: Declare two variables. Assign them with two English capital alphabet characters. Compare them with the “lower than” operator (<) and print on the console the result.

BASICS02-012: Declare two variables. Assign them with two English alphabet characters – one in capital and the other one in lowercase. Compare them with the “lower than” operator (<) and print on the console the result. Can you describe the result?

BASICS02-013: Declare two variables. Assign them with one English alphabet character and one number character. Compare them with the “greater than” operator (>) and print on the console the result. Can you describe the result?

BASICS02-014: Declare a variable. Assign one letter from the English alphabet. Print on the console the reverse – if the letter is capital, print it in lowercase; if the letter is in uppercase, print it in lowercase. Hint: Use the encoding table/codes/location in the table.

BASICS02-015: Declare a variable. Assign it with a letter from the English alphabet. Define a constant, named Cipher and assign it a value in the range of [3;13]. Print on the console the Cipher-th letter after the assigned letter to the variable. Imagine that the English

alphabet is linked in a circle (cycle) – after Z letter follows A letter, then B etc. Example: if you have Cipher=4, and the given letter is 'Y', then print on the console 'C' (the fourth after 'Y').

Basics 03 – Operations, Operators, Precedence

BASICS03-001: Declare two variables and assign them two integer numbers. Print on the console the result of their division.

BASICS03-002: Declare two variables and assign them two integer numbers. Print on the console the division remainder (modulus - остатък от целочислено деление).

BASICS03-003: Declare four variables. On two of them assign integer numbers. The third set with the division remainder. The fourth one set with the quotient (частното -> цялата част от делението). Print on the console the four variables with appropriate description.

BASICS03-004: Define a constant. Check and print on the console if the constant is positive, negative or zero. Hint: Use ternary operators. How many operators do you need?

BASICS03-005: Declare three variables and assign them with three randomly selected integer numbers. Print on the console those two of them, which have the biggest sum. Hint: Use the ternary operators.

BASICS03-006: Declare one variable, assign integer number. Check if the variable contains an even number. Print on the console appropriate message.

BASICS03-007: Declare a constant and assign one digit. Print on one line the constant, the power of two (N^2), the power of three (N^3) on the console.

BASICS03-008: Declare a variable. Assign one digit from the range of [1;9]. Print on the console the multiplication table with that variable.

BASICS03-009: Calculate and print on the console the perimeter of a triangle.

BASICS03-010: Calculate and print on the console the surface area (лицето) of a triangle. ($S=(a.h_a)/2$)

BASICS03-011: Calculate and print on the console the perimeter of a rectangle.

BASICS03-012: Calculate and print on the console the surface area of a rectangle.

BASICS03-013: Calculate and print on the console the perimeter (the length) of a circle. ($C=2.\pi.r$)

BASICS03-014: Calculate and print on the console the surface area of a circle.

BASICS03-015: Declare a variable. Assign one digit from the range of [1;9]. Print on the console the multiplication table with that variable, but in reverse order – first print the multiplication with 10, then with 9, etc. Also, print on each line the calculated result from the multiplication raised to the second power (на сменен 2).

BASICS03-016: A bus leaves from point A to point B with speed of 80 km/h. At the same time, a car leaves from point B to point A with speed of **x** km/h. The distance between point A and point B is **S** kilometers. After how many minutes, the bus, and the car will meet? Print the result on the console. (S = V.t)

BASICS03-017: Write a JavaScript program to convert degrees in radians. Print on the console an appropriate message.

BASICS03-018: Write a JavaScript program to convert km/h into km/min. Print on the console an appropriate message.

BASICS03-019: Write a JavaScript program to convert km/h into m/s. Print on the console an appropriate message.

BASICS03-020: Declare a variable. Assign an integer number. Print on the console the variable, the binary, octal and hexadecimal representation.

BASICS03-021: Are there any not correctly defined expressions in the following excerpt? If any – which one(s)? Why?

```
let a = 1;
let b = a;

let r1 = a+--b;
let r2 = a+++b;
let r3 = a---b;
let r4 = a-++b;
let r5 = a*++b;
let r6 = a*--b;
let r7 = a****++b*b;
let r8 = a****++b          //b;
```

BASICS03-022: You have the following excerpt of a JavaScript code:

```
let a = 1;
let b = 3;
let result = a****++b/b++++*a;
```

Try to calculate the value of the result variable, without executing the code.

DTypes 01 – Strings

DTYPES01-001: Define a string variable with your names (first and last) in two ways (hint: use different quotes).

DTYPES01-002: Declare two variables. Initialize one of them with your first name and the other one – with your last name. Define a variable called **fullName** with a template string. Use variable substitution for the two simple names and use a space character as a separator. Print fullName on the console.

DTYPES01-003: Declare two variables. Initialize one of them with your first name and the other one – with your family name. Define a variable called **fullRevName** with a template string. Use variable substitution for the two simple names, having the family name as a first name. Use comma character as a separator. Print fullRevName on the console.

DTYPES01-004: Define two multi-line string variables in three ways. (hint: use single quotes, double quotes, string literals).

DTYPES01-005: Define a string variable with following format:

My name's <name>. I'm <age> years old.

Previously define the name and the age values. They will not be changed during the further code.

DTYPES01-006: Define four string variables with your three names (first, middle and family). The fourth variable initialize with all names, separated with a tab character. Print on the console the length of each variable.

DTYPES01-007: Define four string variables with your three names (first, middle and family). The fourth variable initialize with all names, separated with a tab character. Print on the console the length of each string.

DTYPES01-008: Define four string variables with your three names (first, middle and family). The fourth variable initialize with the concatenation of the first characters from each name. Print the fourth variable on the console.

DTYPES01-009: Define four string variables with your three names (first, middle and family). The fourth variable initialize with the concatenation of the first characters (capitalize them) from each name. Print the fourth variable on the console.

DTYPES01-010: Define four string variables with your three names (first, middle and family). The fourth variable initialize with the concatenation of the first characters (capitalize them) from each name plus the concatenation in reverse order (again capital letters). Print the fourth variable on the console.

DTYPES01-011: Define three string variables with your three names (first, middle and family). Declare a fourth variable. Initialize it with the sum of the character codes from the first characters from each name. Print all variables with appropriate text on the console.

DTYPES01-012: Define three string variables with your three names (first, middle and family). Declare a fourth variable. Initialize it with the sum of the character codes from the last characters from each name. Print all variables with appropriate text on the console.

DTYPES01-013: Define three string variables with your three names (first, middle and family). Declare a fourth variable. Initialize it with the sum of the character codes from the first characters from each name, minus the sum of the character codes from the last characters from each name. Print the fourth variable on the console.

DTYPES01-014: Define a string variable with the following text (called pangram):

The quick brown fox jumps over the lazy dog

Use positive indexing to print on the console the character "b".
Use negative indexing to print on the console the character "s".

DTYPES01-015: Define three string variables with your three names (first, middle and family). Declare a fourth variable. Initialize it with the sum of the character codes from the first characters from each name, minus the sum of the character codes from the last characters from each name. Print the fourth variable on the console.

DTYPES01-016: Print the numbers from the following list: 1, 10, 38, 4, 824, 120, 999, 64 in one column, right justified. Pad all the digits with zeros so that the column is right-justified, and the width is exactly the width of the longest number written in the list. The length of a number is measured by the number of digits involved in its writing.

DTYPES01-017: Print the numbers from the following list: 1, 10, 38, 4, 824, 120, 999, 64 in one column, right justified. Pad all the digits with zeros so that the column is right-justified, and the width is exactly eight characters. Also, replace all eights with the digit nine, and all nines with the digit one.

DTYPES01-018: The following permanent text is given:

Primitive

Print on the console the following:
On the first line – print the first character.
On the second line – print first two characters.
On the third line – print first three characters and so on until the whole word is printed.

DTYPES01-019: The following permanent text is given:

Primitive

Print on the console the following:
On the first line – print the first character.
On the second line – print first two characters.

On the third line – print first three characters and so on until the whole word is printed.
Pad each line with space character(s), so the text is right justified.

DTYPES01-020: Print all even (четни) numbers from the string:
The number 28469 is not so big.
Hint 1: Check the remainder of integer division by two.
Hint 2: Use the ternary operator.

DTYPES01-021: Count the number of characters in the English word for each digit ([0;9]).
Print on the console, in different lines, and in comma separated format for each digit the following - the digit, the English word and the number of letters.
Example:
...
6, six, 3
7, seven, 5

DTYPES01-022: Check if the word “jump” exists in the following text (called pangram):
The quick brown fox jumps over the lazy dog
If it does not exist -> print “does not exist”.
If it is fount -> print “found at position: <position>.

DTYPES01-023: Check if the word “jump” exists in the following text (called pangram), ignoring the case sensitivity:
The quick brown fox jumps over the lazy dog
If it does not exist -> print “does not exist”.
If it is fount -> print “found at position: <position>.

DTypes 02 – Numbers

DTYPES02-001: