Assignment On Fundamentals of JavaScript

Variables

Q-1.

1. Declare two variables: `admin` and `name`.
2. Assign the value `"John"` to `name`.
3. Copy the value from `name` to `admin`.
4. Show the value of `admin` using `alert` (must output "John").

Q-2.

1. Create a variable with the name of our planet. How would you name such a variable?
2. Create a variable to store the name of a current visitor to a website. How would you name that variable?

Q-3.

Examine the following code:

```
const birthday = '18.04.1982';

const age = someCode(birthday);
```

Here we have a constant `birthday` for the date, and also the `age` constant.

The `age` is calculated from `birthday` using `someCode()`, which means a function call that we didn't explain yet (we will soon!), but the details don't matter here, the point is that `age` is calculated somehow based on the `birthday`.

Would it be right to use upper case for `birthday`? For `age`? Or even for both?

```
const BIRTHDAY = '18.04.1982'; // make birthday uppercase?

const AGE = someCode(BIRTHDAY); // make age uppercase?
```

Data Types

Q-1.

What is the output of the script?

```
let name = "Ilya";

alert( `hello ${1}` ); // ?

alert( `hello ${"name"}` ); // ?

alert( `hello ${name}` ); // ?
```

Interaction : (let, const, var)

Q-1.

Create a web-page that asks for a name and outputs it.

Type Conversion

Q-1.

What are results of these expressions?

```
"" + 1 + 0
"" - 1 + 0
true + false
6 / "3"
"2" * "3"
4 + 5 + "px"
"$" + 4 + 5
"4" - 2
"4px" - 2
"  -9  " + 5
"  -9  " - 5
null + 1
undefined + 1
" \t \n" - 2
```

Q-2.

Here's a code that asks the user for two numbers and shows their sum.

It works incorrectly. The output in the example below is `12` (for default prompt values).

Why? Fix it. The result should be `3`.

```
let a = prompt("First number?", 1);
let b = prompt("Second number?", 2);

alert(a + b); // 12
```

Basic Operators

Q-1.

What are the final values of all variables `a`, `b`, `c` and `d` after the code below?

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

let c = ++a; // ?
let d = b++; // ?
```

Q-2.

What are the values of `a` and `x` after the code below?

```
let a = 2;

let x = 1 + (a *= 2);
```

Q-3.

What will be the result for these expressions?

```
5 > 4
"apple" > "pineapple"
"2" > "12"
undefined == null
undefined === null
null == "\n0\n"
null === +"\n0\n"
```

Q-4.

What will the code below output?

```
alert( alert(1) || 2 || alert(3) );
```

What will this code show?

```
alert( alert(1) && alert(2) );
```

What will the result be?

```
alert( null || 2 && 3 || 4 );
```

What will the results of the expressions be inside `if(...)`?

```
if (-1 || 0) alert( 'first' );
if (-1 && 0) alert( 'second' );
if (null || -1 && 1) alert( 'third' );
```

Conditional Branching

Q-1

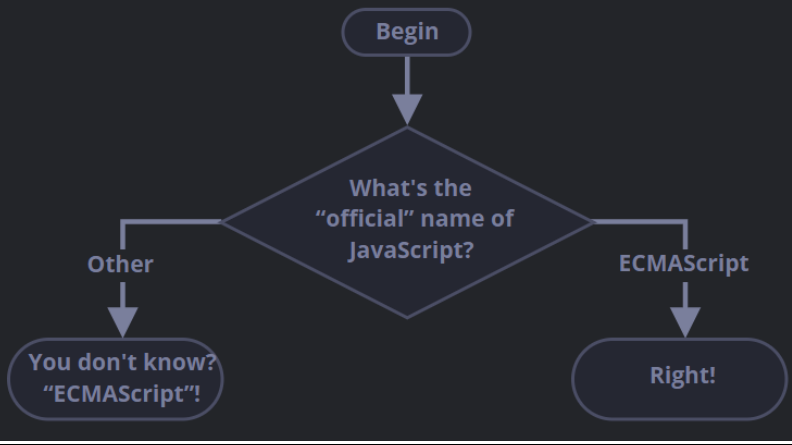
Will `alert` be shown?

```
if ("0") {
  alert( 'Hello' );
}
```

Q-2.

Using the `if..else` construct, write the code which asks: 'What is the "official" name of JavaScript?'

If the visitor enters "ECMAScript", then output "Right!", otherwise – output: "You don't know? ECMAScript!"



Q-3.

Rewrite this `if` using the conditional operator `'?'`:

```
let result;

if (a + b < 4) {
  result = 'Below';
} else {
  result = 'Over';
}
```

Q-4.

Write the code using `if..else` which would correspond to the following `switch`:

```
switch (browser) {
  case 'Edge':
    alert( "You've got the Edge!" );
    break;

  case 'Chrome':
  case 'Firefox':
  case 'Safari':
  case 'Opera':
    alert( 'Okay we support these browsers too' );
    break;

  default:
    alert( 'We hope that this page looks ok!' );
}
```

Q-5.

Using `if..else`, write the code which gets a number via `prompt` and then shows in `alert`:

- `1`, if the value is greater than zero,
- `-1`, if less than zero,
- `0`, if equals zero.

In this task we assume that the input is always a number.

Q-6.

Rewrite the code below using a single `switch` statement:

```
let a = +prompt('a?', '');

if (a == 0) {
  alert( 0 );
}
if (a == 1) {
  alert( 1 );
}

if (a == 2 || a == 3) {
  alert( '2,3' );
}
```

Looping Statements : (while, for, do-while)

Q-1.

An integer number greater than `1` is called a prime if it cannot be divided without a remainder by anything except `1` and itself.

In other words, `n > 1` is a prime if it can't be evenly divided by anything except `1` and `n`.

For example, `5` is a prime, because it cannot be divided without a remainder by `2`, `3` and `4`.

Write the code which outputs prime numbers in the interval from `2` to `n`.

For `n = 10` the result will be `2,3,5,7`.

P.S. The code should work for any `n`, not be hard-tuned for any fixed value.

Q-2.

Write a loop which prompts for a number greater than `100`. If the visitor enters another number – ask them to input again.

The loop must ask for a number until either the visitor enters a number greater than `100` or cancels the input/enters an empty line.

Here we can assume that the visitor only inputs numbers. There's no need to implement a special handling for a non-numeric input in this task.

Q-3.

Use the `for` loop to output even numbers from `2` to `10`.

Q-4.

Rewrite the code changing the `for` loop to `while` without altering its behavior (the output should stay same).

```
for (let i = 0; i < 3; i++) {
  alert( `number ${i}!` );
}
```

Functions

Q-1.

```
Write a javascript function which returns
"You are Eligible for Driving !" if the condition
is true otherwise it will return
"You are Not eligible for Driving !"
```

Q-2.

Write a function `min(a,b)` which returns the least of two numbers `a` and `b`.

For instance:

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

Q-3.

Write a function `pow(x,n)` that returns `x` in power `n`. Or, in other words, multiplies `x` by itself `n` times and returns the result.

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ...* 1 = 1
```

Create a web-page that prompts for `x` and `n`, and then shows the result of `pow(x,n)`.

Q-4

Replace Function Expressions with arrow functions in the code below:

```
function ask(question, yes, no) {
  if (confirm(question)) yes();
  else no();
}

ask(
  "Do you agree?",
  function() { alert("You agreed."); },
  function() { alert("You canceled the execution."); }
);
```
