

ESMAD | TSIW | POO
Exercise Sheet **Resolution** nº2
Conditionals and Loops

Use Visual Studio Code to solve the following exercises:

1. Conditional Statements:

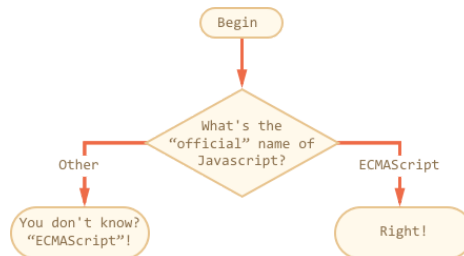
- a. Will the log be shown in the console?

```
if ("0") {  
  console.log( 'Hello' );  
}
```

Yes, it will be shown.

Any string, except an empty one (and "0" is not empty) becomes true in the logical context.

- b. Using an if..else block, write the code that asks: "What's the "official" name of Javascript?". If the visitor types "ECMAScript", send "Right!", Otherwise, the output: "You don ' t know? "ECMAScript"! "



```
let value = prompt('What is the "official" name of JavaScript?', '');  
  
if (value == 'ECMAScript') {  
  alert('Right!');  
} else {  
  alert("You don't know? ECMAScript!");  
}
```

- c. Using an if..else block, write the code that gets a number via prompt and then shows on alert:
- 1, if the value is greater than zero,
 - 1, if less than zero,
 - 0, if equal to zero.

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

```
let value = prompt('Write a number', 0);

if (value > 0) {
  alert( 1 );
} else if (value < 0) {
  alert( -1 );
} else {
  alert( 0 );
}
```

- d. Rewrite this if with the ternary operator '?':

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

```
result = (a + b < 4) ? 'Below' : 'Over';
```

- e. Rewrite the next if ... else using multiple ternary operators '?'. For easy reading, it is recommended to split the code into several lines.

```
let message;
```

```
if (login == 'Employee') {
  message = 'Hello';
} else if (login == 'Director') {
  message = 'Greetings';
} else if (login == "") {
  message = 'No login';
} else {
  message = "";
}
```

```
let message = (login == 'Employee') ? 'Hello' :
  (login == 'Director') ? 'Greetings' :
  (login == '') ? 'No login' :
  '';
```

2. Logical operators:

- a. What is the output for each line of code?

```
console.log( null || 2 || undefined );
console.log( alert(1) || 2 || alert(3) );
console.log( 1 && null && 2 );
console.log( alert(1) && alert(2) );
console.log( null || 2 && 3 || 4 );
```

```
console.log( null || 2 || undefined );
```

The answer is 2, this is the first true value.

```
console.log( alert(1) || 2 || alert(3) );
```

The answer: first 1 and then 2.

The call to alert does not return a value, or rather, it returns undefined.

The first OR || evaluates the left alert (1). This shows the first message with 1.

The alert returns undefined, so OR moves to the second operand looking for a true value. The second operand 2 is true, then the execution is interrupted, 2 is returned and then shown by the external alert. There will not be 3, because the assessment does not reach the alert (3).

```
console.log( 1 && null && 2 );
```

The answer: null, because it is the first false value in the list.

```
console.log( alert(1) && alert(2) );
```

The answer: 1, then undefined.

The alert call returns undefined (only shows a message, so there is no significant return). Because of this, && evaluates the left operand (shows 1), and immediately stops, because undefined is a false value. And && looks for a false value and returns.

```
console.log( null || 2 && 3 || 4 )
```

Answer: 3

The && precedence is greater than ||, so it is performed first.

The result of 2 && 3 = 3, then the expression becomes:

```
null || 3 || 4
```

Now the result is the first truthy value: 3.

- b. Write an "if" condition to verify that the age is between 14 and 90, inclusive. Inclusive means that age can reach extremities 14 or 90.

```
if (age >= 14 && age <= 90)
```

- c. Write an if condition to check that the age is NOT between 14 and 90, inclusive. Create two variants: the first using NOT !, the second without it!.

```
//1ª variante
```

```
if (!(age >= 14 && age <= 90))
```

```
//2ª variante
```

```
if (age < 14 || age > 90)
```

- d. Which of these alerts will be executed? What will be the results of the expressions inside if (...)?
- ```
if (-1 || 0) console.log('first');
if (-1 && 0) console.log('second');
if (null || -1 && 1) console.log('third');
```

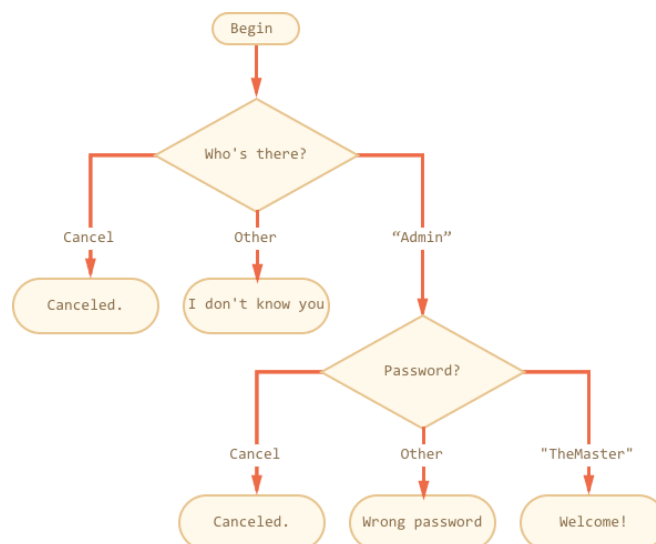
**The answer: the first and the third will be executed.**

```
// Executes
// The result of -1 || 0 = -1, truthy
if (-1 || 0) console.log(first);

// Does not executes
// -1 && 0 = 0, falsy
if (-1 && 0) console.log(second);

// Executes
// Operator && has a precedence greater than ||
// then -1 && 1 executes first, giving us the expression:
// null || -1 && 1 -> null || 1 -> 1
if (null || -1 && 1) console.log('third');
```

- e. Write the code that asks for a login with prompt.
- If the visitor types "Admin", ask for a password, if the entry is an empty line or Esc - display "Canceled". If it's another, show "I don't know you".
  - The password is verified as follows:
    - If it equals "TheMaster", then show "Welcome!",
    - Another string value: shows "Incorrect password",
    - For an empty string or canceled entry, show "Canceled".
  - The scheme:



Please use nested blocks. Observe the general readability of the code.

```

let userName = prompt("Who's there?", '');

if (userName == 'Admin') {

 let pass = prompt('Password?', '');

 if (pass == 'TheMaster') {
 alert('Welcome!');
 } else if (pass == '' || pass == null) {
 alert('Canceled.');
 } else {
 alert('Wrong password');
 }

} else if (userName == '' || userName == null) {
 alert('Canceled');
} else {
 alert("I don't know you");
}

```

### 3. Switch:

- a. Write the code using an if..else block which would correspond to the following switch:

```

switch (browser) {
 case 'Edge':
 console.log("You've got the Edge!");
 break;

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

 default:
 console.log('We hope that this page looks ok!');
}

```

```

if (browser == 'Edge') {
 alert("You've got the Edge!");
} else if (browser == 'Chrome'
|| browser == 'Firefox'
|| browser == 'Safari'
|| browser == 'Opera') {

```

```

 console.log('Okay we support these browsers too');
} else {
 console.log('We hope that this page looks ok!');
}

```

- b. Rewrite the code below using a single switch statement:  
**let a = +prompt('a?', "");**

```

if (a == 0) {
 console.log(0);
}
if (a == 1) {
 console.log(1);
}
if (a == 2 || a == 3) {
 console.log('2,3');
}

```

```

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

switch (a) {
 case 0:
 console.log(0);
 break;

 case 1:
 console.log(1);
 break;

 case 2:
 case 3:
 console.log('2,3');
 break;
}

```

#### 4. Loops:

- a. What is the last value logged by this code? Why?  
**let i = 3;**

```

while (i) {
 console.log(i--);
}

```

**Resposta: 1**

Each loop iteration decreases i by 1. The while (i) check stops the loop when i = 0.

Therefore, the steps of the loop form the following sequence:

```
let i = 3;
```

```
console.log(i--); // shows 3, decreases i to 2
```

```
console.log(i--); // shows 2, decreases i to 1
```

```
console.log(i--); // shows 1, decreases i to 0
```

```
// done, i changes to 0 and the condition is evaluated as false
```

- b. For each loop iteration, write down the value it generates and then compare it with the solution. Do both loops alert the same values or not?

- i. The prefix form ++i:

```
let i = 0;
while (++i < 5) console.log(i);
```

- ii. The postfix form ++i:

```
let i = 0;
while (i++ < 5) console.log(i);
```

**No!**

**First code the result is: 1 2 3 4**

**Second code the result is: 1 2 3 4 5**

- c. For each loop write down what values it will show. Then compare with the answer. Do both loops log the same values or not?

- i. The prefix form ++i:

```
for (let i = 0; i < 5; i++) console.log(i);
```

- ii. The postfix form ++i:

```
for (let i = 0; i < 5; ++i) console.log(i);
```

**The answer: from 0 to 4 in both cases.**

- d. Use the for loop to generate even numbers from 2 to 10.

```
for (let i = 2; i <= 10; i++) {
 if (i % 2 == 0) {
 console.log(i);
 }
}
```

```
}
}
```

- e. Rewrite the code by changing the for loop without changing its behavior (the output must remain the same).

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

```
let i = 0;
while (i < 3) {
 console.log(`number ${i}!`);
 i++;
}
```

- f. Write a loop that asks for a number greater than 100. If the visitor enters another number, ask them to enter it again. The loop must request a number until the visitor enters a number greater than 100 or cancels the entry/inserts an empty line. Here we can assume that the visitor only enters numbers. There is no need to implement a special treatment for a non-numeric entry in this task..

```
let num;

do {
 num = prompt("Enter a number greater than 100?", 0);
} while (num <= 100 && num);
```

- g. An integer 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 prime if it cannot be 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 that generates prime numbers in the range 2 to  $n$ .

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

P.S. The code must work for any  $n$ , it must not be set to any fixed value.

```
let n = 10;

nextPrime:
for (let i = 2; i <= n; i++) {
 for (let j = 2; j < i; j++) {
```



```
 if (i % j == 0) continue nextPrime;
 }

 console.log(i); // a prime
}
```