## Task 1. FDS

Create function ***conc*** which should concatenate two parameters ***a*** and ***b*** and return concatanating string using Function Declaration Statement (FDS). Call this function before its declaration.
Test Data:
a = “1”, b = “1”, result = “11”
a = 1, b = 1, result = “11”

## Task 2. FDE

Create function ***comp*** which should compare two parameters a and b and return 1 if a equal b and -1 if a not equal b using Function Definition Expression (FDE). Call this function before its declaration.
Test Data:
a = “abc”, b = “abc”, result = 1
a = “abC”, b = “abc”, result = -1

## Task 3. AF

Create anonymous function which should log message “message in console” to the console and use it as a click handler for button.

## Task 4. NFE

Create function fibo to calculate fibonachi numbers using named function expression

## Task 5. IIFE

Make the function ***conc*** immediately-invoked function expression

## Task 6. Arguments Object

Create function ***parts*** which takes several parameters. Each parameter is a group of sentences. This function should extract the substring from the sign “:”(colon) to the sign “.”(period) of each parameter and return the array of this substrings
Use Function Definition Expression.
Test Data:
param1 = "This is the first sentence. This is a sentence with a list of items: cherries, oranges, apples, bananas."
param2 = "This is the second sentence. This is a sentence with a list of items: red, blue, yellow, black."
result = [“cherries, oranges, apples, bananas”, “red, blue, yellow, black”]

## Task 7a. Optional Arguments

Create function find(testString, test) which should return the position of test string in testString. If you omit the second parameter use test = testString. Use Function Definition Expression.
Test Data:
testString = “abc”, test =”b”, result = 1
testString = “abc”, result = 0
testString = ”abc”, test = “d”, result = -1
testString = “abc”, test=”a”, test2=”b”, result = 0

## Task 8. Function as an Object

Create the function str which takes one parameter and alert(“String is non empty”) if string is non empty and alert(“String is empty”) otherwise. Use following function to check this condition.
Create the function isNonEpmtyStr as a property of function str. This function takes one parameter and returns true if its parameter is NonEmptyStr.
Test Data:
str.isNonEmptyStr(), result = false
str.isNonEmptyStr(“”), result = false
str.isNonEmptyStr(“a”), result = true
str.isNonEmptyStr(1), result = false
str(), alert(“String is empty”)
str(“a”), alert(“String is non empty”)

## Task 9. Function as a Parameter

Create the function toConsole with one parameter, which displays the value of its parameter in console.log
Create the function toAlert with one parameter, which displays the value of its parameter using alert()
Create the function splitToWords with two parameters: msg and callback. This function splits msg into words and uses callback to display words in console or by alert. If the second parameter is omitted, function returns the array of words.
Test Data:
splitToWords("My very long text msg", toConsole);
result
My
very
long
text
msg
splitToWords("My very long text msg", toAlert);
result = alert(My), ….
console.log( splitToWords("My very long text msg") );
result = [“My”, “very”, “long”, “text”, “msg”]

## Task 10. Function as a Result

Create function copyright which returns another function with one parameter. Returned function adds sign © (“\u00A9”) at the beginning of its parameter. Declare copyright sign in outer function.

Test Data

console.log( copyright()(“EPAM”) ); result = © EPAM.

## Task 11. Function as a Method

Create object literal Employee with the following properties: name: “Ann”, work – function which display message "I am "+ this.name +". I am working..." in console.log.

Test Data

Employee.work() result in console "I am Ann. I am working..."