**Assignment 2     Total Grade: 70   (of possible 70 points)**

All questions are based on ECMASCRIPT 5.

Some questions have specific hints.  Make sure you press on the Hints link to see them.

**Question 1 of 7     Score: 7   (of possible 7 points)**

Consider the following function:

function mystery(name) {

    var result = '';

    for (var i = 0; i < name.length; i++) {

        result = name[i] + result;

    }

    return result;

}

What is mystery('Foothill')?

Correct{llihtooF}

 Answer Key: llihtooF|'llihtooF'|"llihtooF"

**Question 2 of 7     Score: 7   (of possible 7 points)**

Consider the following function:

function mystery(a) {

        var result = a + '0';

        return;

}

What is typeof mystery('5')?

function mystery(number) {

var result = number + '2';

return;

}

|  |  |  |
| --- | --- | --- |
|  | A. | "number" |
|  | B. | "string" |
|  | C. | "function" |
| Correct | D. | "undefined" |

 Answer Key: D

**Feedback**

If the return statement does not have an associated expression, it returns the undefined value.

**Question 3 of 7     Score: 15   (of possible 15 points)**

Consider the function longest that we defined in module 4.2:

function longest(first, second) {

    if (first.length >=  second.length) {

        return first;

    } else {

        return second;

    }

Use the || operator to **specify default values for first and second in the function.**  If one or both parameters are not specified, the empty string should be used as the default value.

Once you make your changes, you should be able to test the function as follows:

console.log(longest('Alice'));   // second is undefined - second defaults to the empty string

Alice

console.log(longest());    // both first and second are undefined - both default to the empty string

*(an empty string)*

console.log(longest('hi','hello'));

hello

console.log(longest('hi', 'me'));

hi

console.log(longest(''));

*(an empty string)*

**Answer**

 function longest(first, second) {  
    first = first || '';  
    second = second ||'';  
    if (first.length >=  second.length) {  
        return first;  
    } else {  
        return second;  
    }  
 }

[[https://myetudes.org/ambrosia_library/icons/expand.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list)

**Question 4 of 7     Score: 15   (of possible 15 points)**

Your task is to define a function dashes that takes a non-negative number as a parameter and returns a string with that many dashes.

For example dashes(3) should return the string '---'.

dashes(1) should return the string '-' .

Start with:

'use strict';

function dashes(number) {

    // enter your code here

    // don't forget to return a string of dashes

}

You may use the following to test your function:

console.log(dashes(3)) // this should print ---

console.log(dashes(1)) // this should print -

console.log(dashes(0)) // this should print *(an empty string)*

**Answer**

 'use strict';  
  
function dashes(number) {  
    var counter = 1;  
    var display ='';  
    while (counter <= number)  {  
       counter++;  
       display = display + "-";  
       }  
    return display;  
}

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list)

function dashes(number) {

    var result = '';

    for (var i = 1; i <= number; i++)  {

        result = result + '-';

    }

    return result;

}

**Question 5 of 7     Score: 11   (of possible 11 points)**

Your task is to define a function getMax2 that takes two numbers as arguments and returns the largest one.

Use the if...else control structure (not Math.max).

Start with:

function getMax2(first, second) {

    // enter your code here

    // don’t forget to **return**the maximum

}

You can test your function as follows:

console.log(getMax2(10, 3));  // this should print 10

console.log(getMax2(-10, -6));  // this should print -6

console.log(getMax2(3, 3));  // this should print 3

You do not have to provide default values for the parameters first and second in this question.  You may assume that they are defined and that they are numbers.

**Answer**

function getMax2(first, second) {  
    var max;  
    if (first >= second) {  
        max = first;  
    }  
     else {  
         max = second;  
     }  
    return max;  
}

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list)

function getMax2(first, second) {

    var result;

    if (first >= second) {

        result = first;

    } else {

        result = second;

    }

    return result;

}

**Question 6 of 7     Score: 15   (of possible 15 points)**

Your task is to define a function getMax3 that takes **three** numbers as arguments and returns the largest of them.

**Get the maximum of the first 2 numbers first using getMax2 from the previous question**.

maximum (a, b, c) = maximum(maximum (a, b), c)

Start with:

'use strict';

// copy your solution for getMax2 here

function getMax3(first, second, third) {

// enter your code here

// don’t forget to return the maximum

}

You can test your function as follows:

console.log(getMax3(1, 4, 8)); // this should print 8

console.log(getMax3(5, 4, 3)); // this should print 5

console.log(getMax3(1, 8, -2)); // this should print 8

console.log(getMax3(-5, -10,- 5)); // this should print -5

You do not have to provide default values for the parameters first, second and third in this question.  You may assume that they are defined and that they are numbers.

**Answer**

'use strict';

function getMax2(first, second) {  
    var max;  
    if (first>=second){  
        max = first;  
    }  
     else {  
         max = second;  
     }  
    return max;  
}  
  
function getMax3(first, second, third) {  
    var max2 = getMax2(first,second);  
    var finmax;  
    if (max2 >=third) {  
        finmax = max2;  
    }  
     else {  
         finmax = third;  
     }  
    return finmax;  
}

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list)

'use strict';

function getMax2(first, second) {

    var result;

    if (first >= second) {

        result = first;

    } else {

        result = second;

    }

    return result;

}

function getMax3(first, second, third) {

    var firstMax = getMax2(first, second);

    return (getMax2(firstMax, third));

}

**Comments**

Note that you can also use getMax2 a second time to get the maximum of max2 and third.

finmax = getMax2(max2, third);

**Question 7 of 7     Score: 0   (of possible 0 points)**

Optional Challenge Question:

How would you implement getMax2 and getMax3 to specify default values for first, second and third?  What default values would you use?

You can test your solution as follows:

console.log(getMax2(10, 3));  // this should print 10 - same as before

console.log(getMax2(2));  // this should print 2

console.log(getMax2(-5));  // this should print -5

**console.log(getMax2(-3, 0));  // this should print 0**

console.log(getMax3(1,-4)); // this should print 1

console.log(getMax3(5)); // this should print 5

console.log(getMax3(1, 8, -2)); // this should print 8 - same as before

**Answer**

*No text entered.*

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14621430/list)

There are two challenges here:

1.  How do we pick a default value that will not affect the outcome?  We need a value that will be**less or equal**to the other parameter(s) if any.  We can use **-Infinity or the value of the previous parameter**, if any.

2.  The second challenge is due to the fact that we are dealing with numbers and 0 is a valid input parameter.

When we write:

second = second || -Infinity;

If second has the value 0, second will be falsy and the new value of second will be set to -Infinity which is incorrect.

It will result in getMax(-3, 0) returning -3, which is wrong.

To fix that, we need to make sure that  the value of the parameter is not 0 before letting it default.

'use strict';

function getMax2(first, second) {

  if (first !== 0) {

    first = first || - Infinity;

  }

  if (second !== 0) {

    second = second || - Infinity;

  }

  if (first >= second) {

    return first;

  } else {

    return second;

  }

}

function getMax3(first, second, third) {

  var firstMax = getMax2(first, second);

  return (getMax2(firstMax, third));

}

Note that if we implement getMax3 by calling getMax2 twice, we only have to specify the default values in getMax2.

**Another way to implement the default in this case is simply to write:**

'use strict';

function getMax2(first, second) {

  if (first === undefined) {  // If the parameter is omitted

    first =  - Infinity;

  }

  if (second === undefined) {  // If the parameter is omitted

    second =  - Infinity;

  }

  if (first >= second) {

    return first;

  } else {

    return second;

  }

}

function getMax3(first, second, third) {

  var firstMax = getMax2(first, second);

  return (getMax2(firstMax, third));

}

 1 of 1