

# Programming Foundations Course Assignment

## Read

Please read the following instructions carefully. If any part is unclear or you have questions, contact the designated teachers through Microsoft Teams direct message. **Discussing the course assignment in class channels is strictly prohibited.** Additionally, any discussions about the assignment outside of official teacher-student communications, including private messages or in-person conversations with classmates, are not allowed.

If you find the course assignment instructions confusing, asking your teachers for help is perfectly fine. Please understand that the teachers **can't give you the answers** because it's part of the assignment, which is like an exam, but they can **explain the instructions** if you need help understanding them.

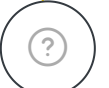
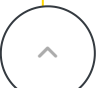
The course assignment will be **graded on a "pass / not pass" basis.**

## Source Code

Download the source code files in the zip folder at the bottom of this page.

The file is named "Programming Foundations - Course Assignment Source Code.zip".

Once these errors are corrected, you are required to complete a quiz. This is the only submission for this CA. The course assignment quiz has a total of **20 questions** that have to be completed within **45 minutes**.



## Submission

The submission for this course assignment is a quiz.

Complete the Course Assignment by following the instructions given below.

**Course Assignment quiz cannot be completed AFTER the deadline.**

**No quiz attempts past the deadline time will be gradable.**

**Failure to complete the quiz this will result in an automatic not passed grade.**

**Once the quiz has started the time count down will start. Even if you close your browser the time will continue to count down. Ensure you have enough time to take the quiz before the deadline.**

# Introduction

In this course assignment, you will need to prepare a project consolidating previous lessons' knowledge. The project is focused on HTML and JavaScript, and you will need to correct programming errors in 10 functionalities you have already learned about. All these functionalities are separate, you can do them in whichever order you prefer (Except for Function 8, which requires Function 7 as a prerequisite).

# Course assignment

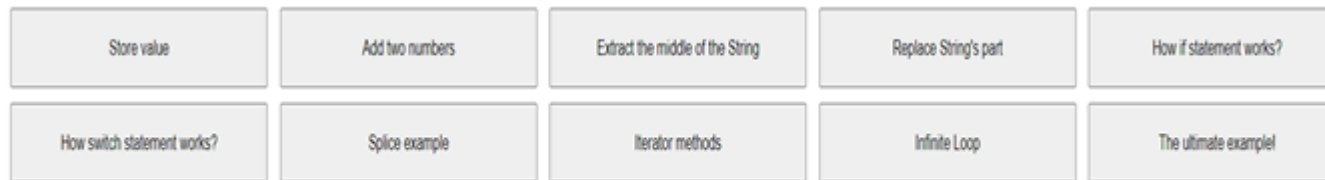
In this course assignment, you must prepare a website with functionalities you learned in the previous modules. You have been provided with one HTML and one JS file. You only need to edit the JS file to correct the source code errors for 10 functions, each of which is bound to one button in the HTML.

While correcting the functions, you might need some math operations. You can use the following methods:

- `Math.round()` – to round the number
- `Math.floor()` to get the integer part of the number

- `Math.ceil()` – to get the number rounded up
- `Math.random()` – to generate a random number from 0 to 1

The provided HTML source code creates a website that looks like this:



Below is a description of what is required from each function:

1. Function `storeValue()` which takes the data from a prompt and stores it within the global variable. If data is provided, show it in an alert. If the prompt is cancelled, show the current data in an alert instead. The prompt should appear on the button click and look this way:

This page says

Provide the new data if you want to change it. Otherwise click Cancel.

OK

Cancel

2. The function `addTwoNumbers()` should get two numbers from the user and show their sum in an alert. You can parse the string to the number with the `parseInt()` method, like in the example below:

```
1 | let a = parseInt("5");
```

You should check if both inputs are proper numbers. If they are not, instead show an alert that the data is not in the correct format.

3. Function `extractMiddleString()` should prompt the user for a string, of which the length should be at least 5. If the input is not correct, show an alert and finish. If the input string is correct, show in an alert the symmetric middle part of the provided string – rounded up. Make sure to use the correct string method. For example:

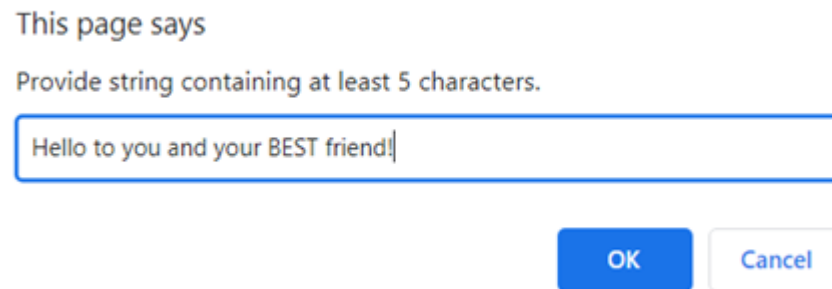
- For entry “string” – “trin” (because  $6 / 2 = 3$ , but we can’t achieve the symmetry with only 3 characters, so we must round up to use 4)
- For entry “sting” – “tin” (because  $5 / 2 = 2.5 \sim 3$  and because the entry has an odd length, we have achieved the symmetry)
- For entry “123456789” – “34567” (because  $9/2 = 4.5 \sim 5$ , the middle character of the string is 5, and both strings have the same middle character which is what we want to achieve)

4. Function `stringReplace()` should take a string from the user and replace all insensitive appearances of

- “a” (‘a’ and ‘A’) to ‘@’
- “e” (e or E) to ‘3’
- “o” (o or O) to ‘0’

You will need to use regular expressions to achieve that.

The resulting string should be shown to the user in an alert, such as the following result:



This page says

H3ll0 t0 y0u @nd y0ur B3ST fri3nd!

OK

5. Function `ifExample()` must prompt the user for their age. The function should check that this age is a number and is greater than 1. If this is true, confirm with the user: "Are you sure that your age is: *\_insert\_age\_*?".

If the user clicks OK, ask them again: Definitely? If the user clicks OK again, show their age in an alert.

Otherwise, show an alert to indicate that we still don't know their age.

This page says

How old are you?

23

OK

Cancel

This page says

Are you sure that your age is: 23?

OK

Cancel

This page says

I still don't know your age :(

OK

6. Ask the user to provide a number from 1 to 5. Check if the received data is correct. If it is, show the relevant story regarding what the number tells us about the user. Make use of a switch-case.

This page says

Please provide a number from 1 to 5

5

OK

Cancel

This page says

Your favourite number is 5. It means you probably are a perfect student!

OK

7. In the function `spliceExample()`, you must convert a string to an array, perform some operations on the array and save the result as a text.

Firstly, generate a string variable called 'splicedText' and store the long text found on the website, [Lipsum](#).

Manipulate this string to remove all commas and dots from the text.

Prompt the user for a positive integer (check if it is in the correct format). If it is, find the first word in our text which has the same length as the provided number (let's name it – X). From the string, remove the two words in the two positions after the position of X, and at the same place insert the copy of X. As a result, you should achieve a string that contains one word less and two of the same

words close to each other.

Remember to log the most important information in the console, it will help you to debug.

This page says

Precise the word's length:




```
m3.js:92
(100) ['Lorem', 'ipsum', 'dolor', 'sit', 'amet', 'consectetur', 'adipiscing', 'elit', 'Phasellu
s', 'ornare', 'mauris', 'eget', 'tortor', 'accumsan', 'posuere', 'Mauris', 'pharetra', 'pellente
sque', 'libero', 'ut', 'cursus', 'eros', 'consectetur', 'nec', 'Suspendisse', 'id', 'erat', 'vit
ae', 'tellus', 'cursus', 'rutrum', 'ut', 'sit', 'amet', 'nisi', 'Aliquam', 'cursus', 'ultrices',
'nisl', 'in', 'vestibulum', 'Nunc', 'lacinia', 'metus', 'a', 'venenatis', 'pretium', 'Nullam',
'vitae', 'tincidunt', 'ante', 'Duis', 'posuere', 'dolor', 'ac', 'accumsan', 'consequat', 'ex',
'mi', 'congue', 'sem', 'sit', 'amet', 'fringilla', 'tellus', 'velit', 'at', 'neque', 'Donec', 'l
uctus', 'mi', 'eu', 'ligula', 'volutpat', 'semper', 'Maecenas', 'vulputate', 'bibendum', 'veli
t', 'at', 'finibus', 'velit', 'consectetur', 'sed', 'Maecenas', 'commodo', 'feugiat', 'Lorem',
'vitae', 'eleifend', 'velit', 'iaculis', 'ut', 'Duis', 'ac', 'suscipit', 'nisl', 'Sed', 'vel',
'metus']

m3.js:92
(99) ['Lorem', 'Lorem', 'sit', 'amet', 'consectetur', 'adipiscing', 'elit', 'Phasellus', 'ornar
e', 'mauris', 'eget', 'tortor', 'accumsan', 'posuere', 'Mauris', 'pharetra', 'pellentesque', 'li
bero', 'ut', 'cursus', 'eros', 'consectetur', 'nec', 'Suspendisse', 'id', 'erat', 'vitae', 'tell
us', 'cursus', 'rutrum', 'ut', 'sit', 'amet', 'nisi', 'Aliquam', 'cursus', 'ultrices', 'nisl',
'in', 'vestibulum', 'Nunc', 'lacinia', 'metus', 'a', 'venenatis', 'pretium', 'Nullam', 'vitae',
'tincidunt', 'ante', 'Duis', 'posuere', 'dolor', 'ac', 'accumsan', 'consequat', 'ex', 'mi', 'con
gue', 'sem', 'sit', 'amet', 'fringilla', 'tellus', 'velit', 'at', 'neque', 'Donec', 'luctus', 'm
i', 'eu', 'ligula', 'volutpat', 'semper', 'Maecenas', 'vulputate', 'bibendum', 'velit', 'at', 'f
inibus', 'velit', 'consectetur', 'sed', 'Maecenas', 'commodo', 'feugiat', 'Lorem', 'vitae', 'ele
ifend', 'velit', 'iaculis', 'ut', 'Duis', 'ac', 'suscipit', 'nisl', 'Sed', 'vel', 'metus']
```

We can see that the first appearance is the word “Lorem”. We remove the two words “ipsum” and “dolor” that followed “Lorem”. Then we added a copy of the word “Lorem” word.

8. Function `iteratorMethods()` gets user-generated text and adds it to the end of every word in the string variable `'splicedText'` from task 7.

You must first prompt the user for some text.

Then convert the 'splicedText' from task 7 to an array. Using the map() method, add the user text to the end of every word in the string variable and convert it back to a string. The new string should be logged to the console.

This page says

Specify the text to add to each element of the current string




```

Lorem ipsum dolor sit amet consectetur adipiscing elit Phasellus ornare mauris eget m3.js:16
tortor accumsan posuere Mauris pharetra pellentesque libero ut cursus eros consectetur nec
Suspendisse id erat vitae tellus cursus rutrum ut sit amet nisi Aliquam cursus ultrices nisl in
vestibulum Nunc lacinia metus a venenatis pretium Nullam vitae tincidunt ante Duis posuere dolor
ac accumsan consequat ex mi congue sem sit amet fringilla tellus velit at neque Donec luctus mi
ligula volutpat semper Maecenas vulputate bibendum velit at finibus velit consectetur sed Maecen
commodo feugiat lorem vitae eleifend velit iaculis ut Duis ac suscipit nisl Sed vel metus

Lorem0o0 ipsum0o0 dolor0o0 sit0o0 amet0o0 consectetur0o0 adipiscing0o0 elit0o0 m3.js:16
Phasellus0o0 ornare0o0 mauris0o0 eget0o0 tortor0o0 accumsan0o0 posuere0o0 Mauris0o0 pharetra0o0
pellentesque0o0 libero0o0 ut0o0 cursus0o0 eros0o0 consectetur0o0 nec0o0 Suspendisse0o0 id0o0
erat0o0 vitae0o0 tellus0o0 cursus0o0 rutrum0o0 ut0o0 sit0o0 amet0o0 nisi0o0 Aliquam0o0 cursus0o0
ultrices0o0 nisl0o0 in0o0 vestibulum0o0 Nunc0o0 lacinia0o0 metus0o0 a0o0 venenatis0o0 pretium0o0
Nullam0o0 vitae0o0 tincidunt0o0 ante0o0 Duis0o0 posuere0o0 dolor0o0 ac0o0 accumsan0o0 consequat0
ex0o0 mi0o0 congue0o0 sem0o0 sit0o0 amet0o0 fringilla0o0 tellus0o0 velit0o0 at0o0 neque0o0
Donec0o0 luctus0o0 mi0o0 eu0o0 ligula0o0 volutpat0o0 semper0o0 Maecenas0o0 vulputate0o0
bibendum0o0 velit0o0 at0o0 finibus0o0 velit0o0 consectetur0o0 sed0o0 Maecenas0o0 commodo0o0
feugiat0o0 lorem0o0 vitae0o0 eleifend0o0 velit0o0 iaculis0o0 ut0o0 Duis0o0 ac0o0 suscipit0o0
nisl0o0 Sed0o0 vel0o0 metus0o0

```

9. In Function infiniteLoop(), create an Infinite loop which asks the user in a prompt to write "STOP". The loop should end when the correct input is given. If the user provides any other data, ask them again.

10. Function ultimateExample() asks the user to provide the number (n). Generate an image of size n times n and show it in the console. Based on the Math.random() method, generate the image consisting of these characters:

- " "
- "|"
- "-"
- "X"
- "."

After generating an image of n=10, something similar to this is shown in the console:



```
X| |.._|.  
X| . _X _.  
xxx| |xx|. |  
| |. _ | .X  
.|.xxx |. |  
. _..| |X|  
.X...|_XX  
.XX.. _..  
| X X X|X  
..| | X.X|
```



Programming Foundations - Course Assignment Source Code.zip

< AW03 - PRF - Course A...

Fortsett til...

PRF Course Assignment Qui... >

Follow Us

---

[Privacy Policy](#)