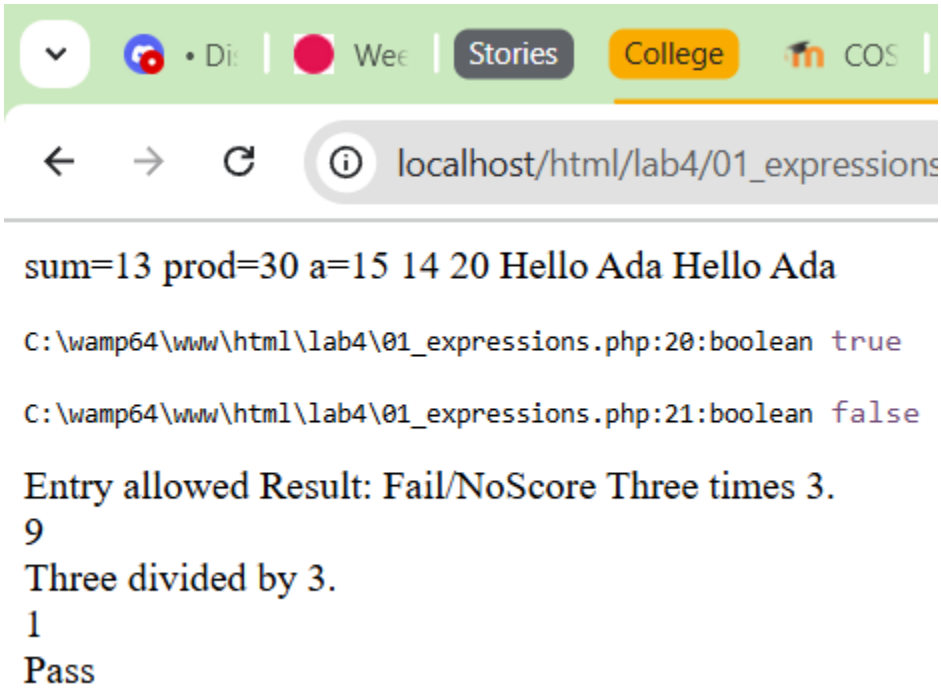
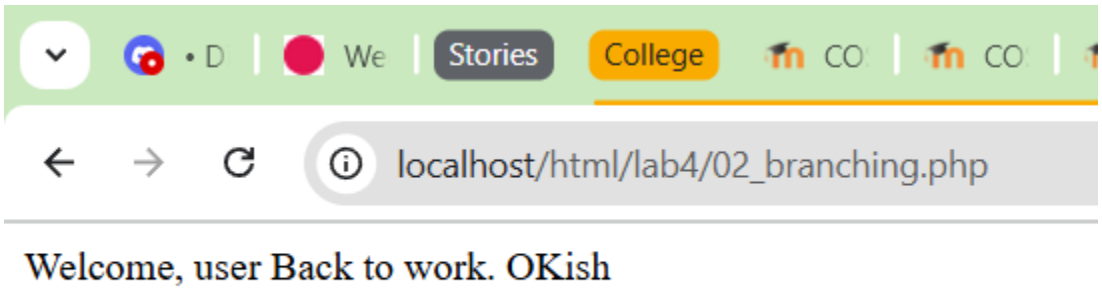
01_Expressions:



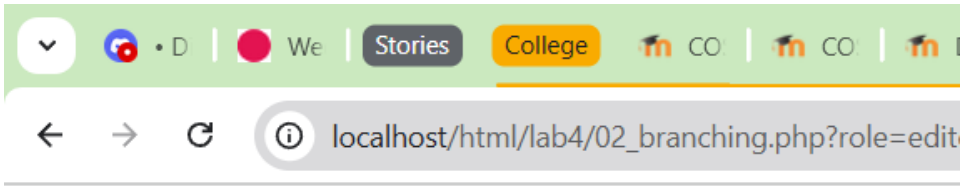
An interesting thing to note is that in the code, it calls for \n to create a new line after the first Hello Ada. This only creates a new line in a unix environment. In browsers, nl2br() to convert \n to
s is necessary.

I also determined that performing mathematical operations on strings automatically converts the result to a number.

02_Branching.php
Standard (Default)

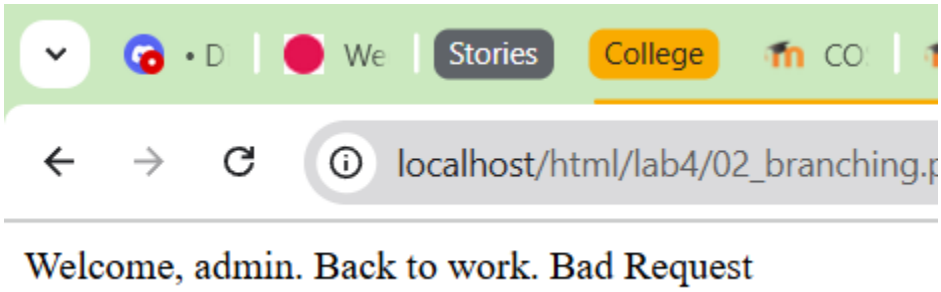


This is 02_branchine without any variables entered into the search bar, resulting in it displaying the results for the default values of role (guest), day (mon), code (200) and lang (en)
Entering ?role=editor&day=Sun&code=404

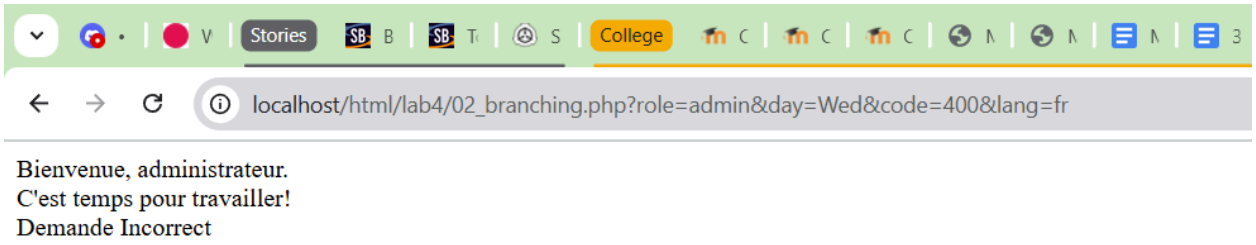


Here, we've entered some values into the search bar and thus prompted a different response. The language is still not entered and thus defaults to English.

Entering ?role=admin&day=Wed&code=400

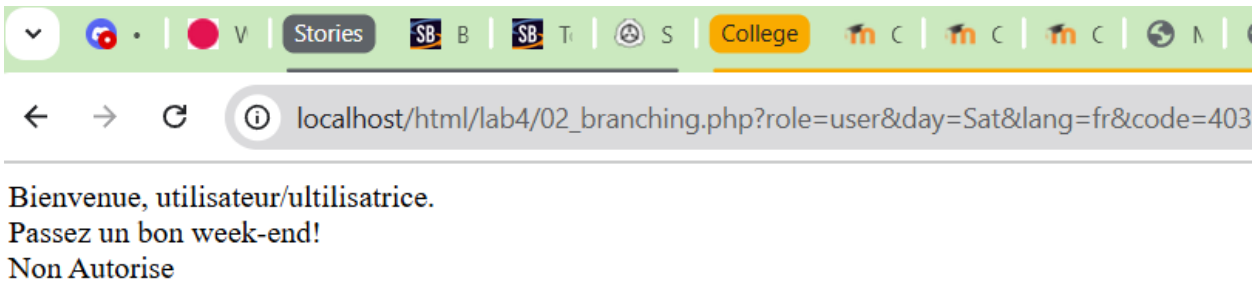


Same but in french



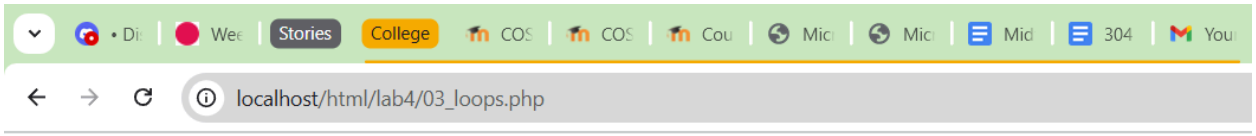
This shows an extreme example of the difference modifying one variable (that being lang) can make. Both of these requests input admin as the role, day as wednesday and code as 400. One is the default En(glish), the other Fr(ancais)

Entering ?role=user&day=Sat&lang=fr&code=403



Another example of output, this time inputting a role of user, saturday as the day (letting us see the week-end message), code as 403 and lang as fr(ancais). In this output, we can see that the order of variables in the search bar does not matter, they will all be interpreted in order by the file, meaning we have greater flexibility and leeway for error when entering these values.

03_loops



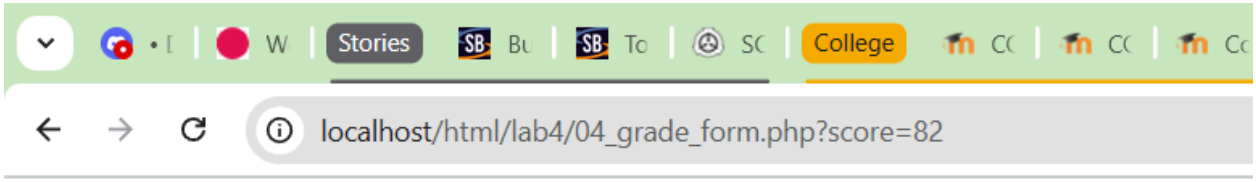
7 14 21 28 35 42 49 56 63 70 First n where sum>100 is 15, sum=105
do...while ran 1 time(s)
Subtotal (skip < 1): 15.7
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 14 FizzBuzz 16 17 Fizz 19 Buzz Fizz 22 23 Fizz Buzz 26 Fizz 28 29 FizzBuzz

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

This output is produced from a number of loops, with the initial being a for loop that adds 7 each turn, the second being a while loop that increases the value being added to sum until sum is greater than 100, a do…while loop with a condition that automatically evaluates as false but runs once (as a do… while loop would), a foreach loop that gets the price sum while skipping values under 1, and finally a for loop that replaces every value divisible by 3 or 5 as Fizz and/or Buzz while skipping 13. I have modified this result, as with the prior when in french, to actually execute newline functions as I believe the initial coder intended.

The table is for the stretch activity, depicting a 1-10 multiplication table. I couldn't get the borders to work, so I bolded the top and leftmost rows.

04_grade_form.php
score=82

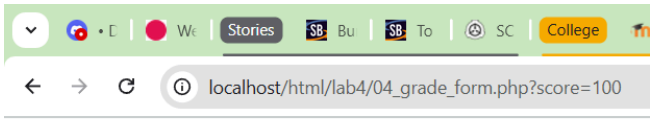


Grade Calculator

Score (0–100): 82 Compute

Your grade is **B**.

score = 100

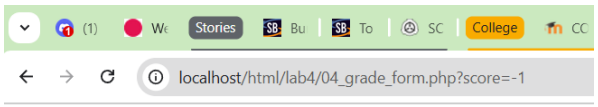


Grade Calculator

Score (0–100): 100 Compute

Your grade is **A**.

score = -1 and score = hello



Grade Calculator

Score (0–100): -1 Compute

Invalid score. Please enter 0–100.

After ‘Stretch’

The following is an example of 04_grade_form/index.php ‘s output after I modified it to display sub-grades such as A+ or B-, and a remark relating to each letter grade.

Grade Calculator

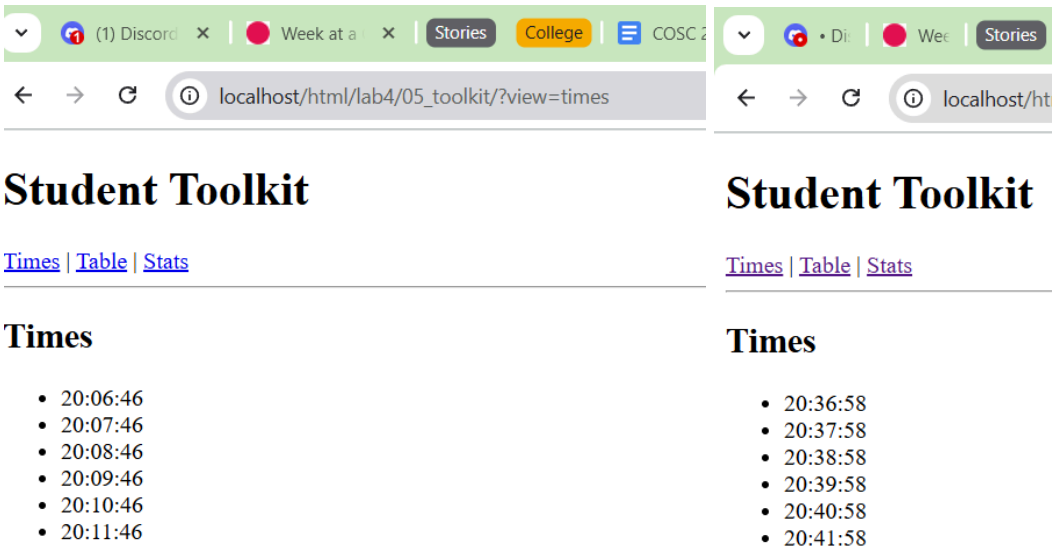
Score (0–100): Compute

Your grade is **F-**.

Perhaps consider enrolling in a different course..

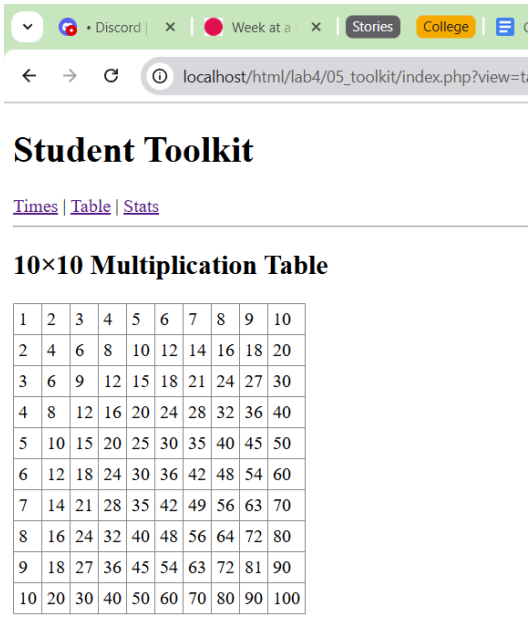
05_toolkit

Times



This page produces a list of 5 ‘Times’ in the Hour:Minute:Second Format upon being loaded. It derives the initial time from the time() function, and then display the list of 5 ‘times’ by incrementing the time it received by 60 seconds (or one minute). However, it is not deriving this value from my system time, as it was around noon local time when these screenshots were produced.
Note that even though it is not depicted, times is the default view if a view is not specified.

Table

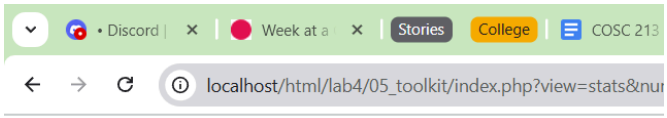


Self-Explanatory. This is a multiplication table of all whole numbers above zero and below eleven. It is produced using both a for loop that calculates the values in each box and an html table element.

Stats / Stats invalid input

As we can see, Stats refuses to process strings not in the specified format (a comma-separated list of integers)

Stats 1, 2, 3, 10, and -1



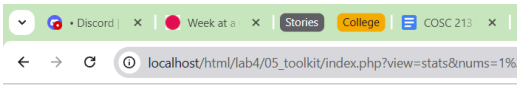
Student Toolkit

Times | Table | Stats

Stats

Numbers (comma-separated): 1 Analyze

- Count: 1
- Sum: 1
- Min: 1
- Max: 1
- Average: 1.00



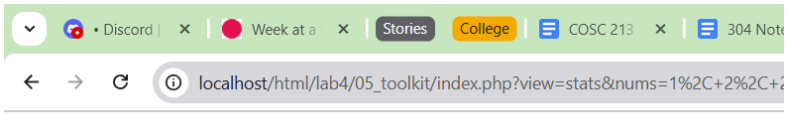
Student Toolkit

Times | Table | Stats

Stats

Numbers (comma-separated): 1, 2 Analyze

- Count: 2
- Sum: 3
- Min: 1
- Max: 2
- Average: 1.50



Student Toolkit

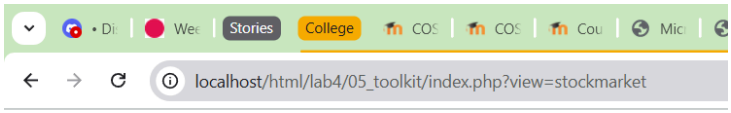
Times | Table | Stats

Stats

Numbers (comma-separated): 1, 2, 2, 10, -1 Analyze

- Count: 5
- Sum: 14
- Min: -1
- Max: 10
- Average: 2.80

Entering an invalid view:

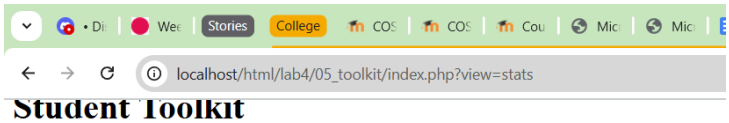


Student Toolkit

Times | Table | Stats

Unknown view.

How does this webpage display three different widgets? By using an if statement that only displays a block of html if a value (in this case, $view) is equal to its own string value, allowing us to make our webpages more concise by only displaying sections with the right value. Here is what happens if we remove this if statement:



Student Toolkit

Times | Table | Stats

Times

- 20:54:32
- 20:55:32
- 20:56:32
- 20:57:32
- 20:58:32
- 20:59:32

10×10 Multiplication Table

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Stats

Numbers (comma-separated): Analyze

Enter a valid comma-separated list of integers.