1.10 Why whitespace matters

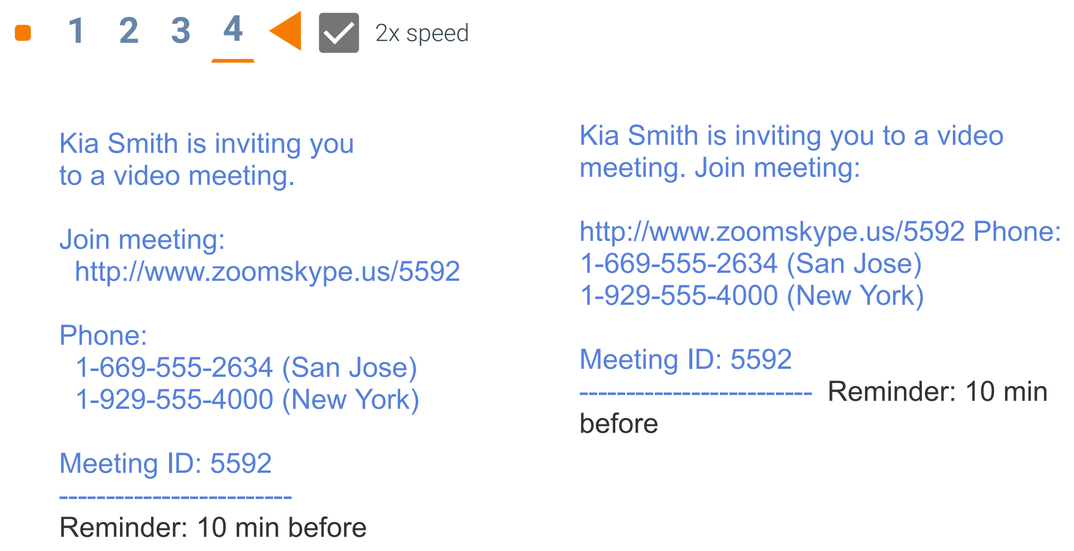
Whitespace and precise formatting

For program output, ***whitespace*** is any blank space or newline. Most coding activities strictly require a student program's output to exactly match the expected output, including whitespace. Students learning programming often complain:

> "My program is correct, but the system is complaining about output whitespace. "

However, correctness often includes output being formatted correctly.

PARTICIPATION ACTIVITY 1.10.1: Precisely formatting a meeting invite.



```
Kia Smith is inviting you
to a video meeting.

Join meeting:
  http://www.zoomskype.us/5592

Phone:
  1-669-555-2634 (San Jose)
  1-929-555-4000 (New York)

Meeting ID: 5592
-------------------------
Reminder: 10 min before
```

```
Kia Smith is inviting you to a video
meeting. Join meeting:

http://www.zoomskype.us/5592 Phone:
1-669-555-2634 (San Jose)
1-929-555-4000 (New York)

Meeting ID: 5592
-------------------------  Reminder: 10 min
before
```

The programmer also didn't end with a newline, causing subsequent text to appear at the end of a line, and even wrap to the next line. This output looks unprofessional.

Feedback?

PARTICIPATION ACTIVITY 1.10.2: Program correctness includes correctly-formatted output.

Consider the example above.

1) The programmer on the

left intentionally inserted a newline in the first sentence, namely "Kia Smith ... video meeting". Why?

○ Probably a mistake

● So the text appears less jagged

○ To provide some randomness to the output

Correct

The two resulting lines are closer to the same length, which may look more appealing to a reader. On the right, the text just wraps when reaching the end, which isn't as neat.

2) The programmer on the right did not end the first sentence with a newline. What effect did that omission have?

● "Join meeting" appears on the same line

○ No effect

Correct

On the left, "Join meeting" is clearly visible due to a preceding blank line. But on the right, "Join meeting" is easy to miss, appearing directly after the previous sentence.

✓

3) The programmer on the left neatly formatted the link, the "Phone:" text, and phone numbers. What did the programmer on the right do?

○ Also neatly formatted those items

● Output those items without neatly formatting

Correct

The output on the right is not neatly formatted. The link is harder to see, and "Phone" is in the wrong place due to a missing newline.

✓

4) On the right, why did the "Reminder..." text appear on the same line as the separator text "-------"?

○ Because programs behave erratically

● Because the

Correct

Programs should almost always end with a newline, so that any subsequent text appears on a separate line. Here, not only does the next text "Reminder..." appear on the same line (which looks bad), but that text also wraps, making the text even harder to read.

✓

programmer didn't end the output with a newline

5) Whitespace ____ important in program output.

● is

○ is not

✓

Correct

Learners often want to ignore whitespace, but most programs that have output must take care to create neat output, with spaces and newlines carefully placed (and usually ending with a newline).

Feedback?

Programming is all about precision

Programming is all about *precision*. Programs must be created precisely to run correctly. Ex:

- = and == have different meanings.
- Using i where j was meant can yield a hard-to-find bug.
- Declaring a variable as int when char was needed can cause confusing errors.
- Not considering that n could be 0 in sum/n can cause a program to fail entirely in rare but not insignificant cases.
- The difference between typing x/2 vs. x/2.0 can have huge impacts.
- Counting from i being 0 to i < 10 vs. i <= 10 can mean the difference between correct output and a program outputting garbage.

In programming, every little detail counts. Programmers must get in a mindset of paying extreme *attention to detail*.

Thus, another reason for caring about whitespace in program output is to help new programmers get into a "precision" mindset when programming. Paying careful attention to details like whitespace instructions, carefully examining feedback regarding whitespace differences, and then modifying a program to exactly match expected whitespace is an exercise in strengthening attention to detail. Such attention can lead programmers to make fewer mistakes when creating programs, thus spending less time debugging, and instead creating programs that work correctly.

PARTICIPATION ACTIVITY | 1.10.3: Thinking precisely, and attention to detail. ✓

Programmers benefit from having a mindset of thinking precisely and paying attention to details. The following questions emphasize attention to detail. See if you can get all of the questions correct on the first try.

1) How many times is the letter F

✓

(any case) in the following?
If Fred is from a part of France, then of course Fred's French is good.

8

Check Show answer

Correct

8

Each ocurrence of the letter F has been bolded: **If** **F**red is **f**rom a part o**f** **F**rance, then o**f** course **F**red's **F**rench is good.
Some people miss the f's in "of" because the f sounds like a v there.

2) How many differences are in these two lines?
Printing A linE is done using println
Printing A linE is done using print1n

1

Check Show answer

✓

Correct

1

Although the capitalization of A and line is odd, the instructions asked for differences, not errors. The first printIn has a capital I, and the second print1n has the number 1. In fact, this error is common among people learning certain programming languages: Not noticing that a name uses an uppercase I versus a number 1 versus a lowercase letter L (l).

3) How many typos are in the following?
Keep calmn and cary one.

3

Check Show answer

✓

Correct

3

The correct saying is "Keep calm and carry on". (You can web search the saying for the history). Thus, calmn should be calm, cary should be carry, and one should be on.

4) If I and E are adjacent, I should come before E, except after C (where E should come before I). How many violations are in the following?
BEIL CEIL ZIEL YIEIK TREIL

3

Check Show answer

✓

Correct

3

Any EI is a violation, unless coming after C.
BEIL: Violation
CEIL: I isn't before E, but OK because after C.
ZIEL: I comes before E, so OK.
YIEIK: The first IE is OK. But, that E is involved in EI, which is a violation.
TREIL: Violation.

5) A password must start with a letter, be at least 6 characters long, include a number, and include a special symbol. How many of the following passwords are valid?

✓

Correct

0

Not a single password meets all the rules.
hello is less than 6 characters.

hello goodbye Maker1
dog!three Oops_again 1augh#3

0

Check

Show answer

goodbye lacks number or symbol.
Maker1 lacks symbol.
dog!three lacks number.
Oops_again lacks number.
1augh#3 doesn't start with letter (the first character is the number 1, not letter l).

Feedback?

Programmer attention to details

The focus needed to answer the above correctly on the first try is the kind of focus needed to write correct programs. Due to this fact, some employers give "attention to detail" tests to people applying for programming positions. See for example this test, or this article discussing the issue. Or, just web search for "programmer attention to details" for more such tests and articles.