

Knowledge checks (not graded and optional) - calc()

This section is optional material included for the curious. It will not appear on any graded question.

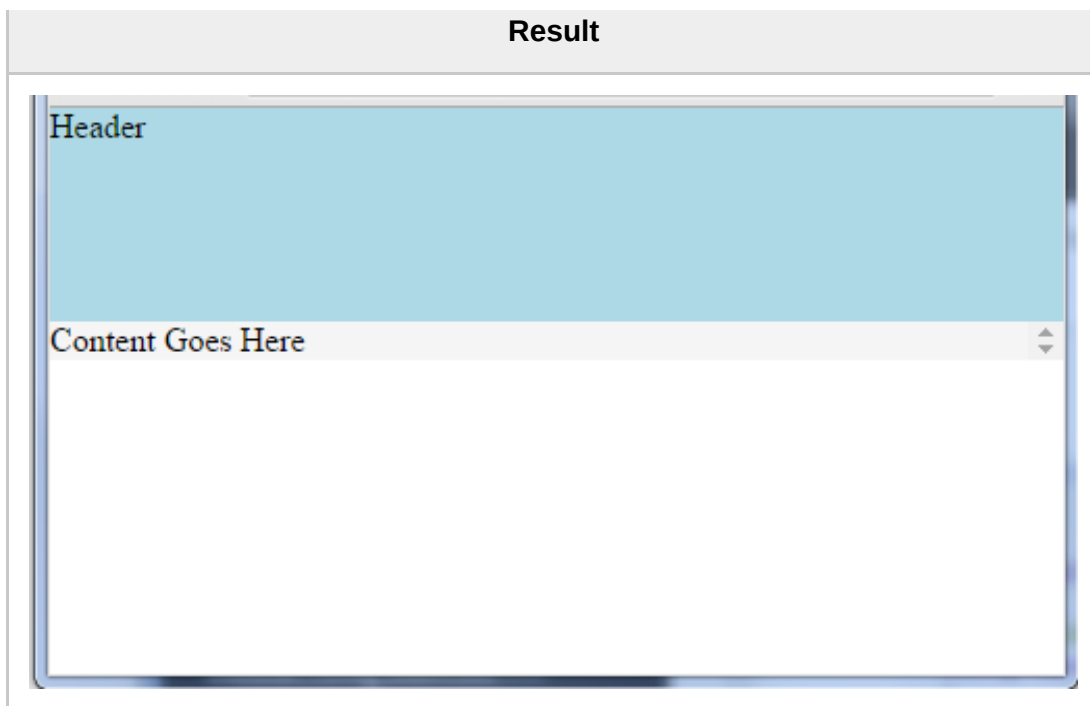
UNITS

Recall from Week 3 the different units that are available in CSS for specifying dimensions:

- px
- em
- rem
- %
- vh / vw

These are all convenient for sizing text and images and videos as might befit the need. But when pursuing page layout we often want to mix units. Examine this simple situation:

HTML	CSS
<pre><body> <header>Header</header> <main>Content Goes Here</main> </body></pre>	<pre>body{ margin: 0px; } header { height: 100px; background-color: lightblue; } main { background-color: whitesmoke; overflow-y: scroll; }</pre>



Because `<header>` and `<main>` are both block level elements, they extend full width. Great. And the `height` of the header has been set to `100px`. But the main does not yet have a height set, so it takes the height of its content. But for this design we want it to scroll its content, and we want it to extend to the bottom of the browser window. The scrolling is easily accomplished with the `overflow-y:scroll;` declaration.

So what should the height of the `<main>` section be to make it extend down?

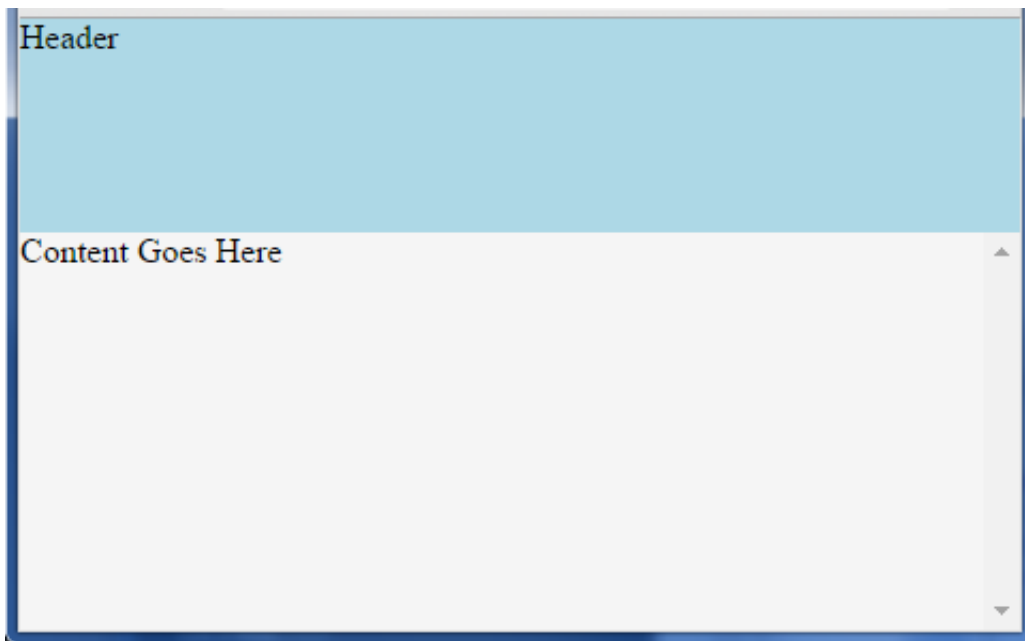
CALC()

The ideal height for the `main` section is that it be the height of the viewport minus the height of the header. You may remember that the `vh` unit is a percentage of the viewport. So `100vh` is one hundred percent of the viewport. And we can see that the height of the header in the CSS above is `100px`.

Wherever a CSS dimension unit is accepted we can also provide the `calc()` expression. This expression lets us mix units of different types. It looks like this:

```
main { height: calc(100vh - 100px); }
```

And if we try add that to the CSS above, we get the result we wanted:



You can download the [final result here](#).

So that's calc(), another weapon in your ever strengthening CSS arsenal.

NOTES

- The parentheses for calc() are required.
- The standard arithmetic operations are supported, addition, subtraction, multiplication and division: + - * /
- Overuse of calc() can make your page slow.
- Using calc() for sizing flexbox items (the items inside a flexbox container) may not always work as desired. In particular, calc() along the cross axis may not work in every situation.

EXPLORATION ACTIVITY

Go back to the original situation posed above. Can you think of any other solutions that will end up with the desired result that don't use calc() and do not use flexbox? We have already discussed quite a few layout concepts. Might some be leveraged to otherwise solve this problem?

9. calc() declarations

Which of the 'calc()' declarations are correct?

(select all that apply - 3 correct answers!)

☐ calc 100vh - 80px;

☐ calc(100vh - 80px);

☐ calc(80px + 120px);

☐ calc(75vw * 2);

CHECK

EXAMINE THE FOLLOWING HTML CODE FOR Q 10 THROUGH 13:

```
<body>
  <header>Header</header>
  <article>Article</article>
  <footer>Footer</footer>
</body>
```

Situation

For this design, you are asked to make the header and footer each have a height of 80px; The content of the article may vary in height. It could be a long series of paragraphs, or it could be just a sentence.

What is the CSS that will size the article height such that it will keep the footer at the bottom of the viewport if the article content is small? But, if the content is long, will let the article just size to the content (meaning the footer may be offscreen until the user scrolls the entire page down).

The next four questions are about the CSS needed to size the article as described.

10. which property for the height of the article

Examine the code and situation above. Think about the CSS required to acheive it.

```
article { property:value; }
```

What *property* should be set to control the overall height of the article as described?

☐ min-height

☐ max-height

☐ height

CHECK

11. Desired value

Examine the code and situation above. In the Question 10 you selected the property that should be used to control the height of the article. But what should its value be set to?

Which of these sentences describes most accurately the desired value?

☐ the maximum height of the article should be no more than twice the viewport height.

☐ at minimum, the height of the article should be the height of the viewport minus the combined heights of the header and footer.

☐ the minimum height of the article should be the height of the

viewport plus the height of the header and the footer

☐ the height of the article should be the same as the height of the viewport

CHECK

12. Height of the viewport

Which of these unit declarations represents the height of the viewport?

☐ 100vw

☐ 100vh

☐ 720px

☐ 100%

CHECK

13. What value should be set...

Examine the code and situation above. In question 10 you selected the property we need to control the height of the article as described. In question 11 you selected a description of its value. In question 12 you selected the declaration that represents the overall height of the viewport. Lastly, observe that the height of the footer and header are each 80 pixels. So, finally, what *value* should be set for the property?

```
article { property:value; }
```

What value should be set for the property?

CHECK

