


# W03 Learning Activities

- Due Sep 18 at 11:59pm
- Points 15
- Questions 13
- Available until Oct 2 at 11:59pm
- Time Limit None
- Allowed Attempts 3

## Instructions

1. **Complete** the [learning activities](https://byui-cse.github.io/wdd131-ww-course/week03/).  (<https://byui-cse.github.io/wdd131-ww-course/week03/>)
2. **Complete** this quiz to check your understanding.
3. Conditions
  - The quiz is open note, application, and internet.
  - Your latest score will be recorded in the gradebook.

Take the Quiz Again

## Attempt History

	Attempt	Time	Score
<b>LATEST</b>	<a href="#">Attempt 1</a>	3 minutes	13 out of 15

 Correct answers are hidden.

Score for this attempt: 13 out of 15

Submitted Oct 2 at 12:59am

This attempt took 3 minutes.



Question 1

3 / 3 pts

I completed all the assigned learning activities for this week.

☒ True

☐ False

Incorrect

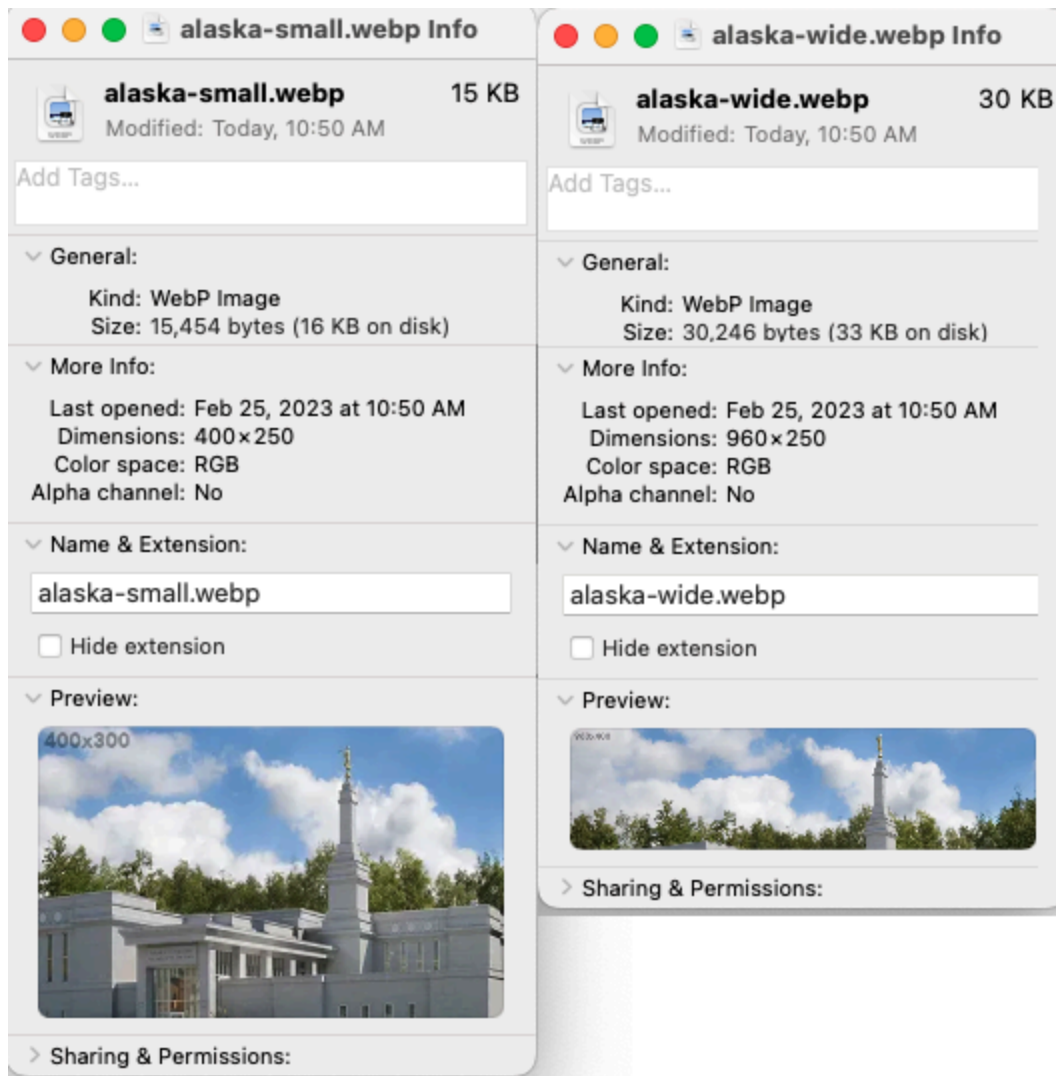


Question 2

0 / 1 pts

A member of your group has used these two images in their responsive image assignment. What feedback would you give them?

Check all that apply.



- ☒ The file size (KB) for these images are way too big.
- ☐ The files are saved using the image file types that are not supported on the web.
- ☐ Everything looks great, keep up the good work.
- ☒ Your images are too large, try reducing them using an image editing program.



### Question 3

1 / 1 pts

A student has this code for their responsive image. What feedback would you give them?

Check all the apply.

```
<picture>
  <source srcset="images/hero-small.webp" media="(max-width: 500px)">
  <source srcset="images/hero-medium.webp" media="(max-width: 1000px)">
  <img srcset="images/hero-large.webp" alt="Hero Photo" width="500" height="250">
</picture>
```

- ☐ You have the two source element lines backwards.

- ☐ Your source references are missing alt attributes.
- ☒ Good job, the code looks good.
- ☐ You forgot to add the units (px) to the width and height attribute values.
- ☐ You are not allowed to use max-width with the <source> element's media attribute.



#### Question 4

1 / 1 pts

One of your peers is confused as to why these JavaScript statements do not work as intended. They report that when a valid chapter is entered into the HTML input field with an id of "favchap" the message on the screen is wrong saying that the favorite chapter is something like [object HTMLInputElement]. And when they do not enter anything into the input field, the message is still outputting the "Thank you. your favorite chapter is [object HTMLInputElement]".

What advice would you give them?

Check all the apply.

```
const message = document.querySelector('#message');
const input = document.querySelector('#favchap');
if (input !== "") {
  message.innerHTML = `Thank you. Your favorite chapter is ${input}.`;
} else {
  message.innerHTML = `Please enter a chapter.`;
}
```

- ☒ Reference the value property of the #favchap input element, not just the element.
- ☐ The HTML input element must have an id of "input" to match the variable identifier used.
- ☐ Use single quotes and not double quotes in the if block condition.
- ☐ Template literal strings (``...``) cannot be used to produce output to the screen.



#### Question 5

1 / 1 pts

Another student has reviewed your favorite chapter application and commented that you must use **innerHTML** versus **textContent** when providing the list item's content.

```
li.innerHTML = item;
```

versus

```
li.textContent = item;
```

What would be your reply to them?  
Mark all that apply.

- ☐ Using innerText would be the best option in this case.
- ☒ Either property will work in this case given that the item variable is a simple string.
- ☒ That would be true if the item variable contained HTML tags.
- ☐ Using the textContent property is the only option when working with HTML list item content.



### Question 6

1 / 1 pts

JavaScript uses type conversion to coerce any value to a Boolean in contexts that require it, such as conditionals and loops. A **falsy** value is a value that is considered false when encountered in a Boolean context. Which of the following are JavaScript falsy values?

- ☒ 0
- ☐ "False"
- ☐ 999
- ☒ ", "", or ``
- ☒ NaN (not a number)
- ☒ null
- ☐ "0"
- ☒ undefined



### Question 7

1 / 1 pts

Which of the following methods can be used to add elements to the end of an **array** in JavaScript?

- ☐ unshift()
- ☐ spice()
- ☐ add()
- ☒ push()

Incorrect



### Question 8

0 / 1 pts

Given the following code snippet, which of the following is the output to the console?

```
let scores = [100, 72, 83, 94, 88, 87];
let accumulator = 0;
let count = 0;
scores.forEach(score => {
  if (score > 87) {
    accumulator += score;
    count++;
  }
});
if ( count > 0 ) {
```

```
    console.log(accumulator/count);
  } else {
    console.log("No scores reported.");
  }
}
```

- ☐ 94
- ☐ 87.5
- ☐ No scores reported.
- ☒ 92.25



### Question 9

1 / 1 pts

Which of the following code expressions produces the last element (index) of any **array**?

- ☒ array.length - 1
- ☐ array.lastIndex
- ☐ array.lastIndex()
- ☐ array.length



### Question 10

1 / 1 pts

Rewrite (refactor) this function declaration by using an arrow function instead of a function declaration.

```
function fullName(first, last) {
  return `${first} ${last}`;
}
```

- ☐ const generateFullName = (first, last) {first.concat(' ', last);}
- ☒ const fullName = (first, last) => `\${first} \${last}`;
- ☐ function = first + " " + last`
- ☐ const fullName = function (first, last) { `\${first} \${last}`; }



### Question 11

1 / 1 pts

In a JavaScript function declaration, which of the following is the keyword used to declare a function?

- ☒ function
- ☐ functionBlock
- ☐ def
- ☐ func



### Question 12

1 / 1 pts

*Finish this statement.*

The \_\_\_\_\_ pseudo-class is useful for declaring global CSS variables and represents the <html> element and is identical to the selector html, except that its specificity is higher.

- ☒ root
- ☐ global
- ☐ html
- ☐ specific



### Question 13

1 / 1 pts

Which of the following statements regarding pseudo-elements in CSS is correct?

- ☐ Pseudo-elements are used to style elements based on their position in the document tree.
- ☐ Pseudo-elements allow you to create virtual elements that do not exist in the HTML document.
- ☐ Pseudo-elements are used to select and style elements based on their attributes.
- ☒ Pseudo-elements are represented by double colons (::) and are used to style specific parts of an element.

**Quiz Score: 13 out of 15**