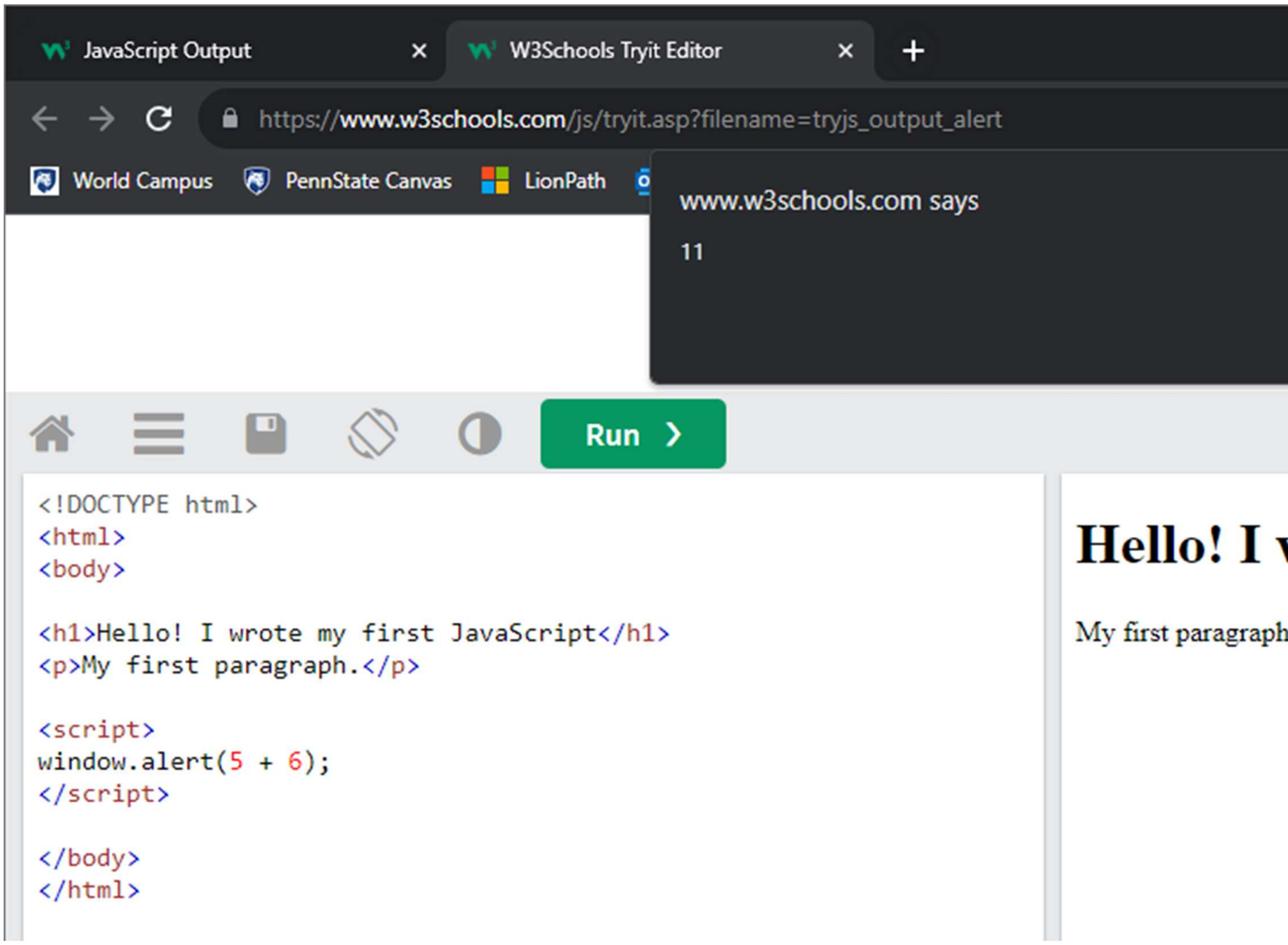
L09: JavaScript
IST 110

1. (Part 3) Screenshot of output when running the Using window.alert() code example



www.w3schools.com says

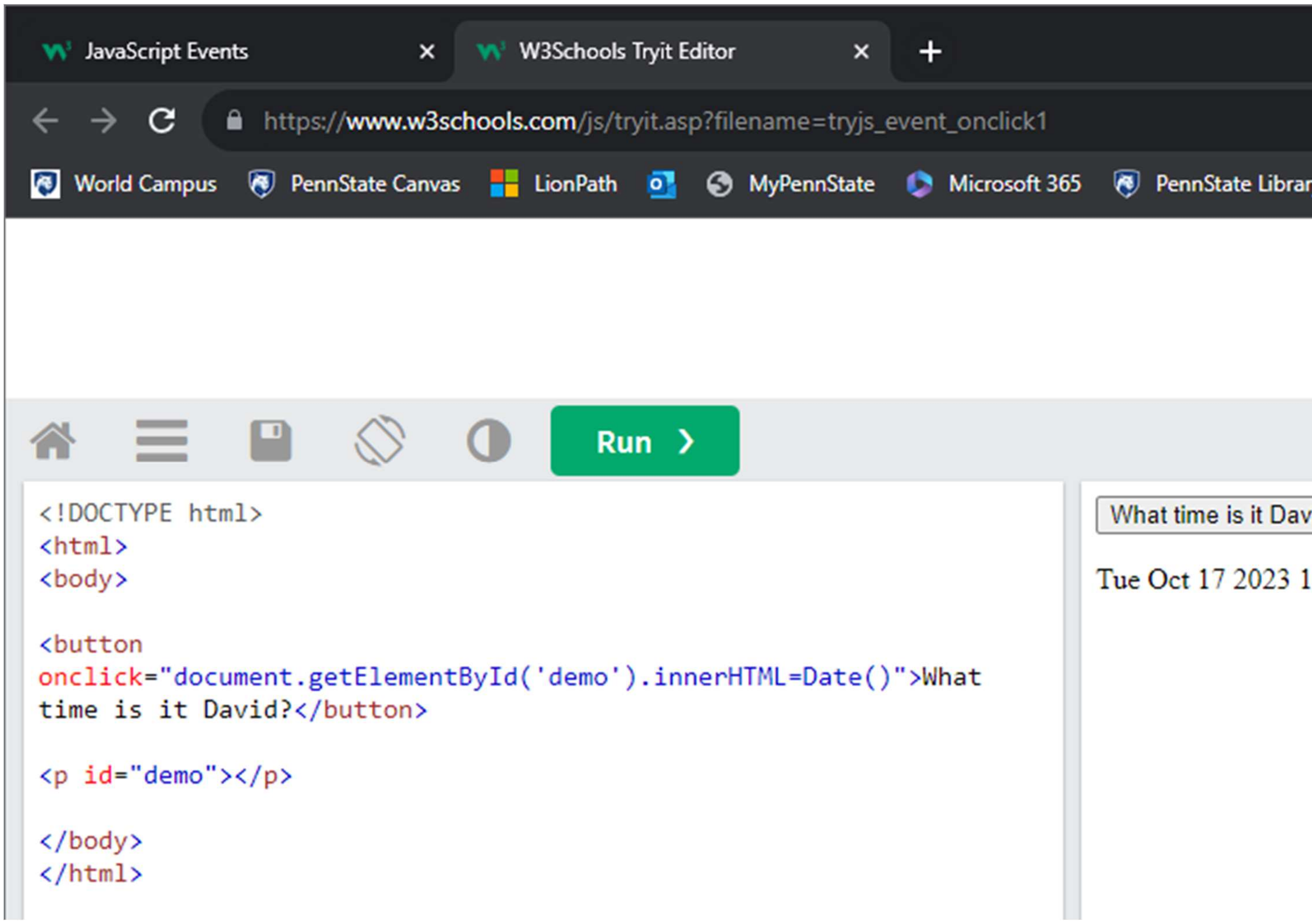
11

```
<!DOCTYPE html>
<html>
<body>

<h1>Hello! I wrote my first JavaScript</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

Hello! I v

My first paragraph

## 2. (Part 5) Screenshot of output when running the HTML Events code example



```
<!DOCTYPE html>
<html>
<body>

<button
onclick="document.getElementById('demo').innerHTML=Date()">What
time is it David?</button>

<p id="demo"></p>

</body>
</html>
```

```
What time is it Dav
Tue Oct 17 2023 1
```

## 3. (Part 5) How are the lines of code in the two examples different? Why is the code activated in the JS events example?

(How they differ)

In the JS events example, the JavaScript code is contained within a <script> tag, which is an element solely devoted to containing JavaScript code. Meanwhile the JS output example has its JavaScript code contained within the <button> element as an attribute with the value "onclick" which allows you to specify JavaScript code to be run when the user clicks on the element.

(Why the events example activates)

The code in the JS events example activates on page load since it is written in a way such that the alert() function is called without necessary conditions as soon as the <script> element is loaded. This means that unlike the code attached to the button in the JS output example, there is no action needed to trigger the function that generates the popup alert.

## 4. (Part 6) Which control structure is the best and why?

While in some instances there isn't technically a need to specify an else after an if condition, an if-else statement does provide the best control over conditional checks and is therefore the best control structure. The job of an if statement is to limit action based on met conditions. However, often you will find yourself in a situation where one data type/variable (especially dynamic ones such as user input) necessitates different action based upon its value. This is where the if-else structure enables us to sequentially check a single variable against multiple conditions while ensuring only the appropriate code. It also allows us to specify a fallback option should all other if checks fail, which helps in error handling.