THE UNIVERSITY *of* ADELAIDE

Web and Database Computing

adelaide.edu.au

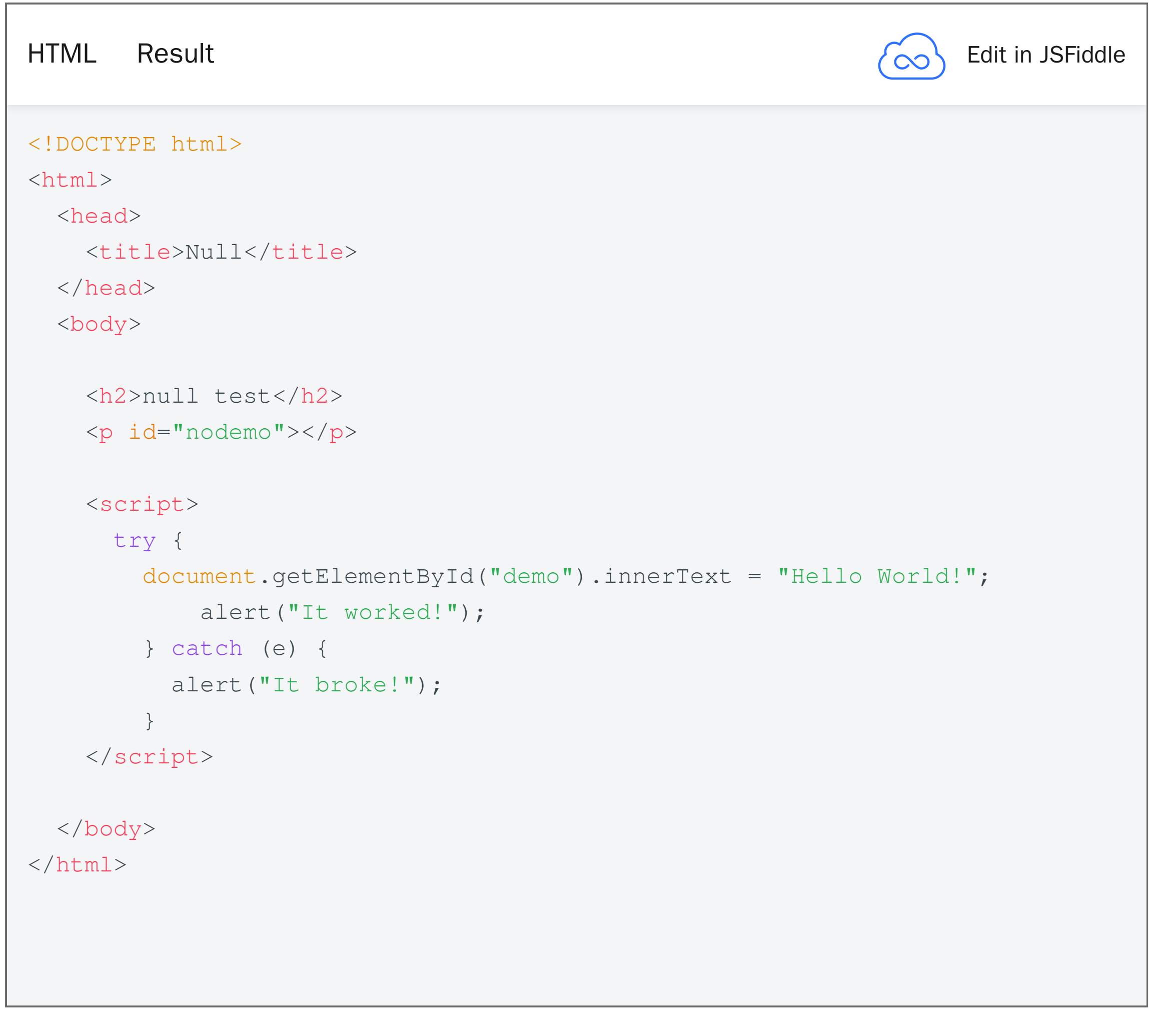
Introduction to JavaScript: JavaScript Tips

More things to watch out for

That might catch out new JavaScript Programmers

Null

- Usually happens when trying to select elements that don't exist
- Can be mitigated using if check or try-catch block



```
<!DOCTYPE html>
<html>
  <head>
    <title>Null</title>
  </head>
  <body>

    <h2>null test</h2>
    <p id="nodemo"></p>

    <script>
      try {
        document.getElementById("demo").innerText = "Hello World!";
            alert("It worked!");
        } catch (e) {
          alert("It broke!");
        }
    </script>

  </body>
</html>
```

Automatic type conversions

== vs ===

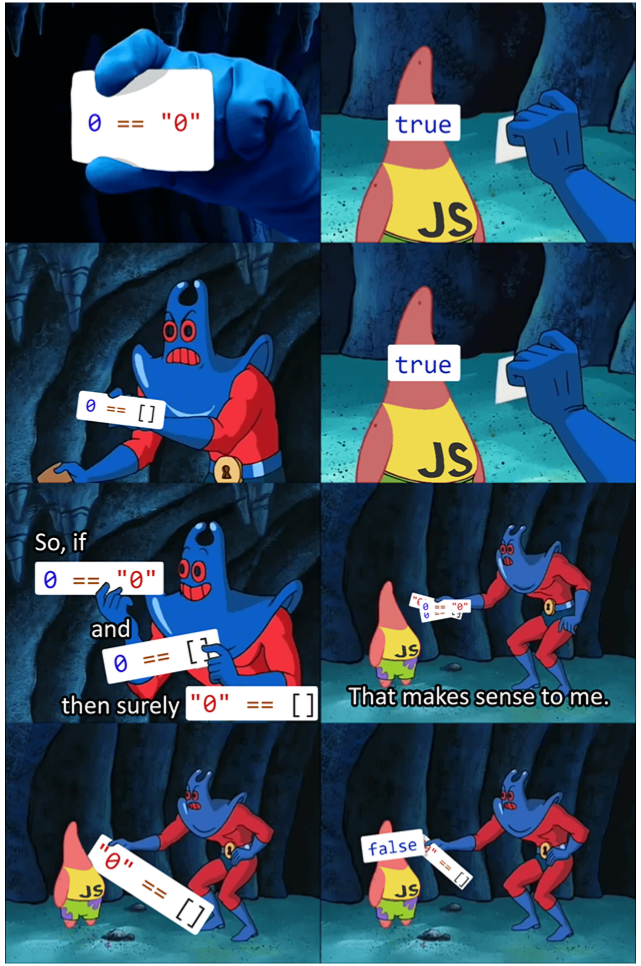
- JavaScript does automatic type conversions.
- Comparing two equivalent values will yield a `true` result
- Can result in unexpected behaviour
  - A common example is checking if something is `0`

```
if (x == 0) { ... }
```

  - If a `null` value is passed instead, it will still evaluate true
- Mitigate using `===` instead of `==`
  - This will compare value **and** type

```
if (x === 0) { ... }
```

- If in doubt, use `===`
- TypeScript is a superset of the JavaScript language that includes types



Credit: William Herrera

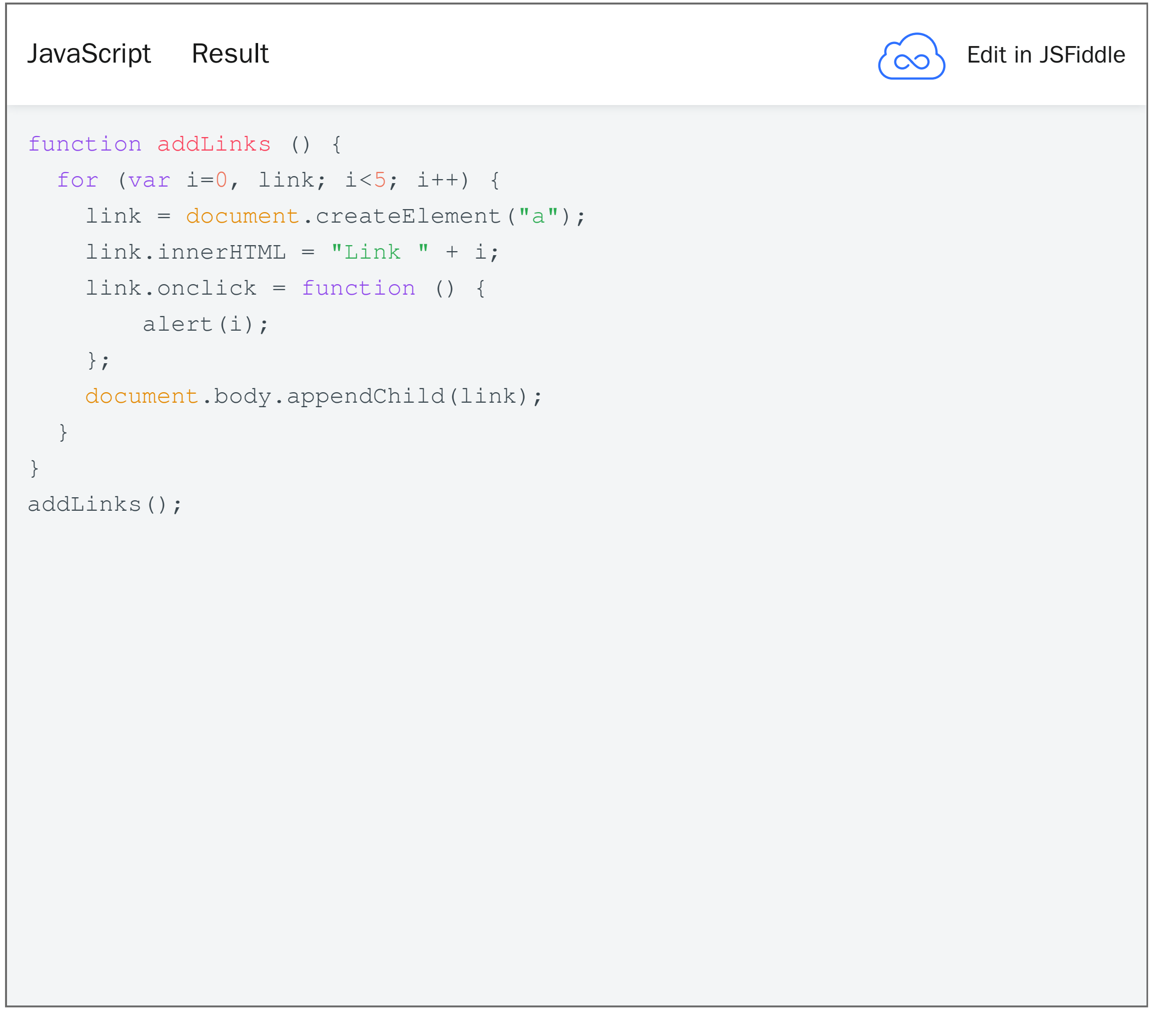
Variable scope & closures

What's happening?

- The onclick function is evaluated when the click occurs
- The last value of `i` is `5`

Remember variable scope is explicitly declared!

- If in doubt, use `let`.
- Can also mitigate using closures
    - See



```
function addLinks () {
  for (var i=0, link; i<5; i++) {
    link = document.createElement("a");
    link.innerHTML = "Link " + i;
    link.onclick = function () {
        alert(i);
    };
    document.body.appendChild(link);
  }
}
addLinks();
```

https://robertnyman.com/2008/10/09/explaining-javascript-scope-and-closures/

SUB CRUCE LUMEN

THE UNIVERSITY of ADELAIDE