JAVASCRIPT TOPIC 4

InfoTech 37

Jomar B. Colao

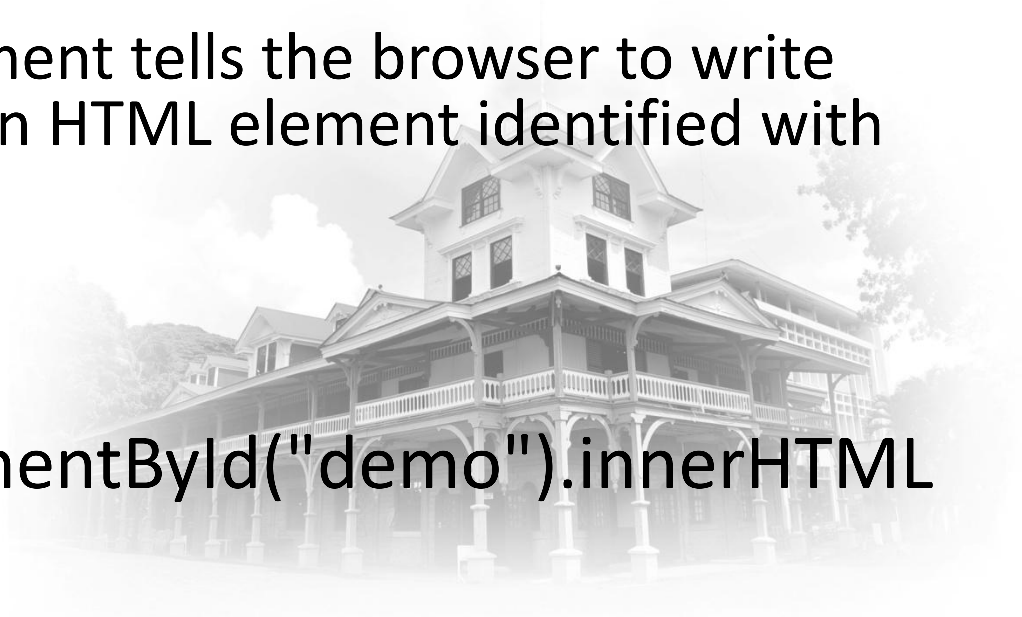
College Instructor

Intro

- In HTML, JavaScript statements are command lines executed by the web browser.

JavaScript Statements

- JavaScript Statements
  - In HTML, JavaScript statements are "commands" to the browser.
  - The purpose, of the statements, is to tell the browser what to do.
  - This JavaScript statement tells the browser to write "Hello Dolly" inside an HTML element identified with id="demo":

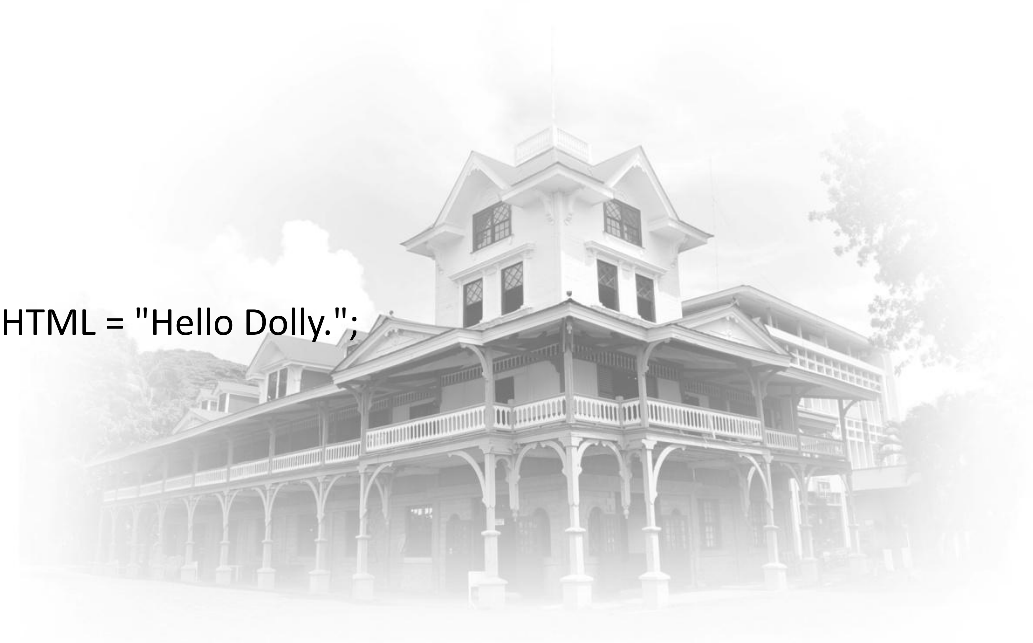


Example

```
document.getElementById("demo").innerHTML
= "Hello Dolly.";
```

Demo 1

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

        <h1>My Web Page</h1>

        <p id="demo">My first paragraph.</p>

        <script>
        document.getElementById("demo").innerHTML = "Hello Dolly.";
        </script>

        </body>
</html>
```



Semicolon

- Semicolon separates JavaScript statements.
- Normally you add a semicolon at the end of each executable statement.
- Using semicolons also makes it possible to write many statements on one line.

Writing:

```
a = 5;

b = 6;

c = a + b;
```

Is the same as writing:

```
a = 5; b = 6; c = a + b;
```

Demo 2

```
<!DOCTYPE html>
<html>
        <body>
        <h1>My Web Page</h1>
        <p id="demo1"></p>
        <p id="demo2"></p>

        <script>
        a = 1;
        b = 2;
        c = a + b;
        document.getElementById("demo1").innerHTML = c;
        x = 1; y = 2; z = x + y;
        document.getElementById("demo2").innerHTML = z;
        </script>

        </body>
</html>
```

Note: You might see examples without semicolons.
Ending statements with semicolon is optional in JavaScript.

JavaScript Code

- ❑ JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- ❑ Each statement is executed by the browser in the sequence they are written.

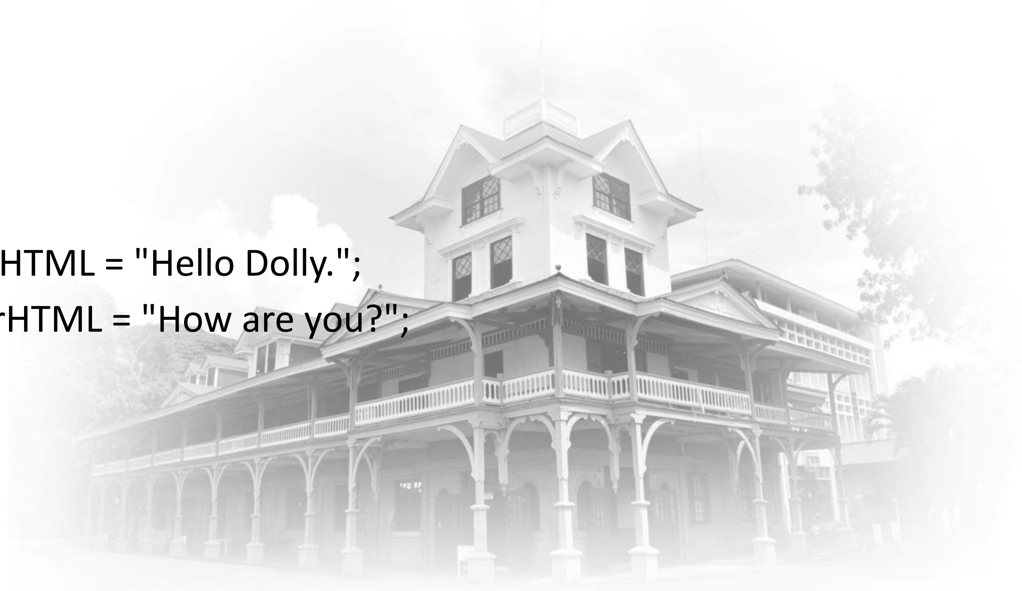
This example will manipulate two different HTML elements:

Example

```
document.getElementById("demo").innerHTML = "Hello Dolly.";
document.getElementById("myDiv").innerHTML = "How are you?";
```

Demo 3

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

        <h1>My Web Page</h1>
        <p id="demo">A Paragraph.</p>
        <div id="myDiv">An HTML div.</div>

        <script>
        document.getElementById("demo").innerHTML = "Hello Dolly.";
        document.getElementById("myDiv").innerHTML = "How are you?";
        </script>

        </body>
</html>
```



JavaScript Code Blocks

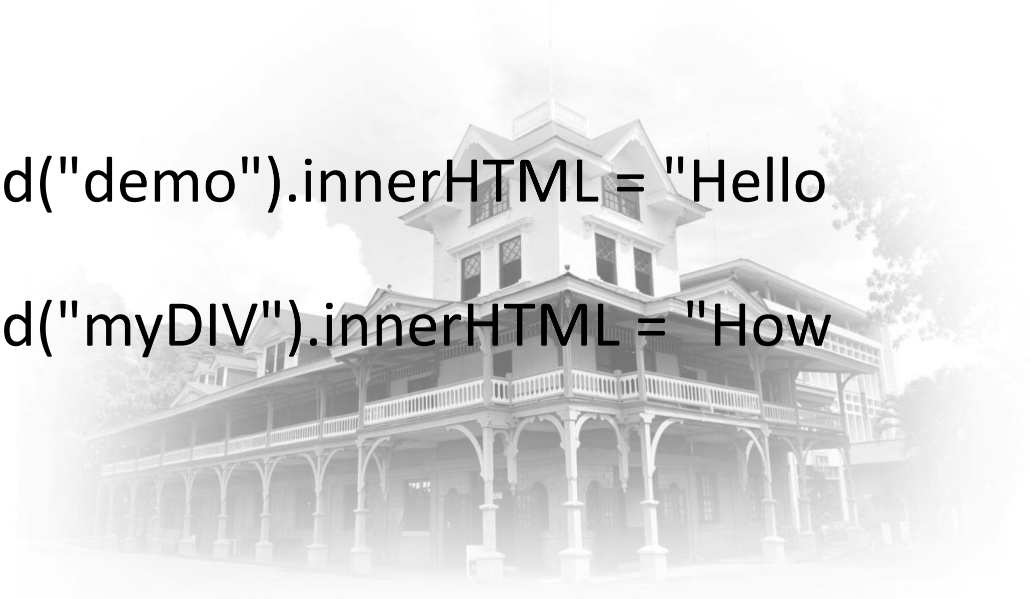
- JavaScript statements can be grouped together in blocks.
- Blocks start with a left curly bracket, and end with a right curly bracket.
- The purpose of a block is to make the sequence of statements execute together.
- A good example of statements grouped together in blocks, are in JavaScript functions.

JavaScript Code Blocks Example

This example will run a function that will manipulate two HTML elements:

Example

```
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello Dolly.";
    document.getElementById("myDIV").innerHTML = "How are you?";
}
```



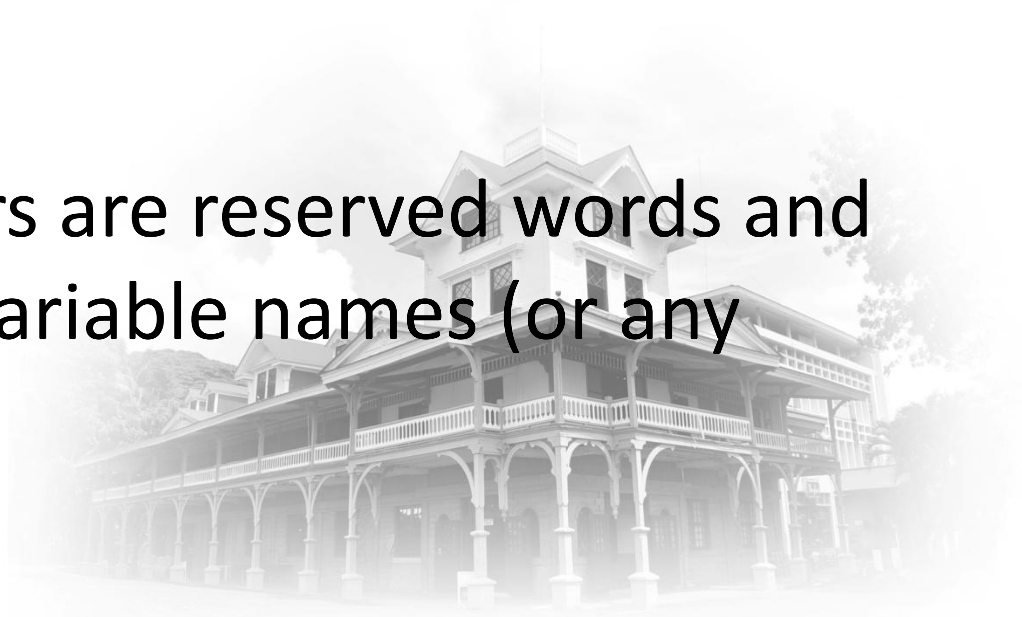
Demo 4

```
<!DOCTYPE html>
<html>
        <body>
        <h1>My Web Page</h1>
        <p id="myPar">I am a paragraph.</p>
        <div id="myDiv">I am a div.</div>
        <p>
        <button type="button" onclick="myFunction()">Try it</button>
        </p>

        <script>
        function myFunction() {
        document.getElementById("myPar").innerHTML = "Hello Dolly.";
        document.getElementById("myDiv").innerHTML = "How are you?";
        }
        </script>

        <p>When you click on "Try it", the two elements will change.</p>
        </body>
</html>
```

JavaScript Statement Identifiers

- ❑ JavaScript statements often start with a statement identifier to identify the JavaScript action to be performed.

- ❑ Statement identifiers are reserved words and cannot be used as variable names (or any other things).



Partial List Of Reserved Words

break Terminates a switch or a loop.

catch Marks the block of statements to be executed when an error occurs in a try block.

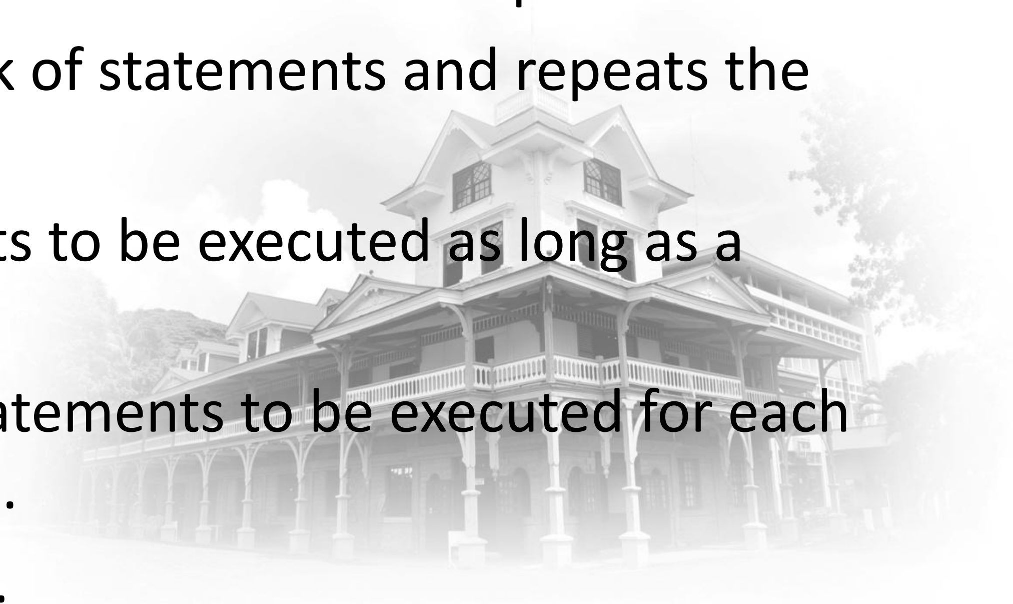
continue Jumps out of a loop and starts at the top.

do ... while Executes a block of statements and repeats the block while a condition is true.

for Marks a block of statements to be executed as long as a condition is true.

for ... in Marks a block of statements to be executed for each element of an object (or array).

function Declares a function.



Partial List Of Reserved Words

if ... else Marks a block of statements to be executed depending on a condition.

return Exits a function.

switch Marks a block of statements to be executed depending on different cases.

throw Throws (generates) an error.

try Implements error handling to a block of statements.

var Declares a variable.

while Marks a block of statements to be executed while a condition is true.

JavaScript White Space

JavaScript ignores extra spaces. You can add white space to your script to make it more readable.

The following lines are equivalent:

```
var person = "Hege";
var person="Hege";
```

JavaScript Line Length and Line Breaks

- For best readability, programmers often like to avoid lines longer than 80 characters.
- If a JavaScript statement does not fit on one line, the best place to break it, is after an operator or a comma.

Example

```
document.getElementById("demo").innerHTML =
    "Hello Dolly.";
```

Demo 5

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

        <h1>My Web Page</h1>

        <p>
        The best place to break a code line is after an operator or a comma.
        </p>

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

        <script>
        document.getElementById("demo").innerHTML =
        "Hello Dolly.";
        </script>

        </body>
</html>
```

JavaScript Line Break in a String

You can break up a code line within a text string with a backslash:

Example

```
document.getElementById("demo").innerHTML = "Hello \
    Dolly!";
```

Demo 6

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

        <h1>My Web Page</h1>

        <p>
        You can break a code line within a text string with a backslash.
        </p>

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

        <script>
        document.getElementById("demo").innerHTML = "Hello \
        Dolly.";
        </script>

        </body>
</html>
```

JavaScript Line Break in a String

you cannot break up a code line like this:

Example

```
document.getElementById("demo").innerHTML = \
    "Hello Dolly!";
```

Demo 7

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

        <h1>My Web Page</h1>

        <p id="demo">You cannot break a code line with a \ (backslash).</p>

        <script>
        document.getElementById("demo").innerHTML = \
        "Hello Dolly.";
        </script>

        </body>
</html>
```