WESTCLIFF
UNIVERSITY
Educate. Inspire. Empower.

Week 3: JavaScript Foundations

Agenda

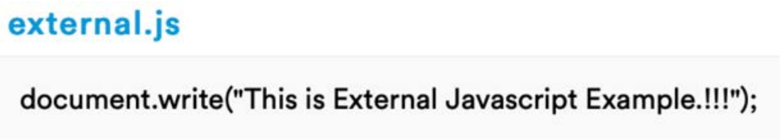
- The methods of writing JavaScript: Internal vs External
- What are variables & how to create and use them
- JavaScript data types
- What are functions & how to create and use them
- What are operators and their meanings
- What is Decision Making and the use of if-else if-else and switch statements
- Test codes using the browser console
- In-Class Demo
- Homework

The methods of writing JavaScript: Internal vs External

Writing JavaScript: Internal vs External

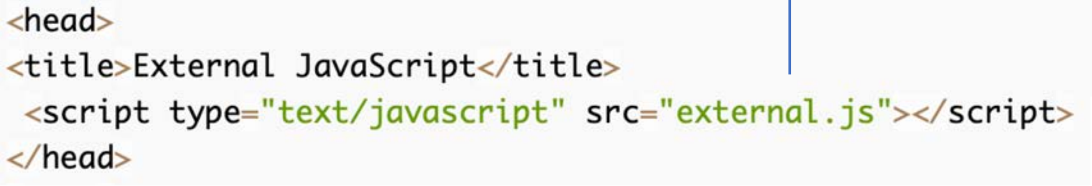
- You can write and access JavaScript codes in two ways:
  - Include JavaScript codes **within your HTML document** (internal)
  - Keep JavaScript codes in **a separate file** (external) and then link to that file from your HTML document.
- Example of both methods:

External Method:



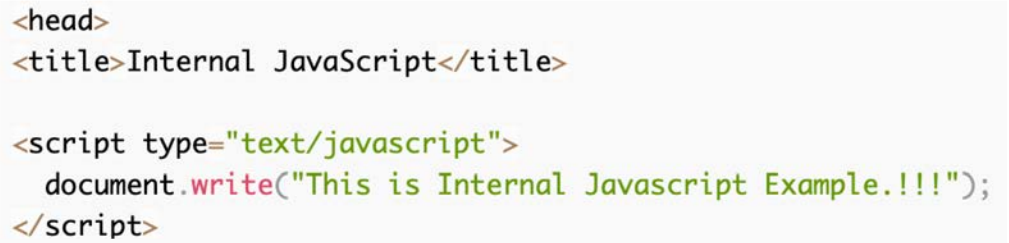
external.js

```
document.write("This is External Javascript Example.!!!");
```



```
<head>
<title>External JavaScript</title>
 <script type="text/javascript" src="external.js"></script>
</head>
```

Internal Method:



```
<head>
<title>Internal JavaScript</title>

<script type="text/javascript">
  document.write("This is Internal Javascript Example.!!!");
</script>
```

Writing JavaScript

- In the previous slide, notice that JavaScript codes are written separately from HTML codes.
  - If written internally, the codes must be wrapped within the <script> element.

    ie: <script>document.write("some message")</script>
  - If written externally, the <script> element is used to specify the source (src) of the external js file location.

    ie: <script src="external.js" type="text/javascript"></script>
  - The document.write() is a global method that allows you to "print" custom text message on to the window’s document (html file).
  - You will learn very soon other ways to "print" custom text messages to specific areas of the document.

Variables

Variables – What is

- Variable is piece of code in JavaScript that can store data so that the program can use it later. Think of variable as a storage container.
- The value or data stored in a variable is not always fixed. It can be changed.
- Some of the rules for creating/using variables:
  - Uses reserved keyword var to declare/create a new variable.
  - Must have a unique name.
  - When assigning a value/data to a variable, use the equal (=) operator when declaring a variable.
  - Can leave variable empty (no data assigned) when declaring.
  - Must be declared/exist before using in the program.

How to create Variables

- To create a variable is also known as declaring a variable.
- Use the keyword **var** which is a reserve word. Here’s how:

  var *variableName*

  - this creates an empty variable named *variableName*

  var *variableName* = data

  - this creates a variable named *variableName* with an assigned value of data

- Examples:

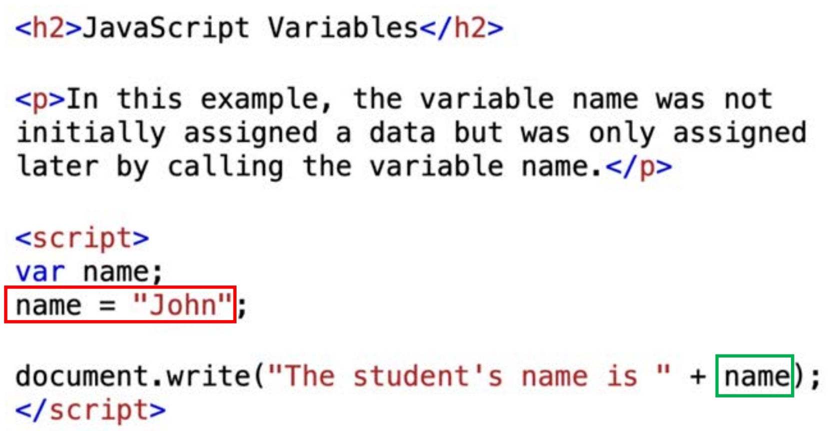
  var *name* = "John";

  var *age* = 25;

How to use Variables

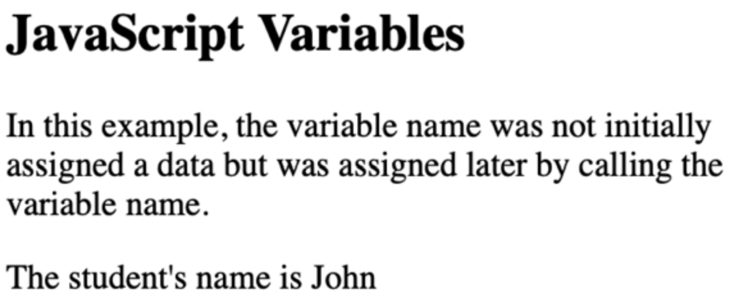
- After variables are declared, they can now be used within the JavaScript program.
- The following example shows how to use and assign data after it's declared:

HTML File:



```
<h2>JavaScript Variables</h2>

<p>In this example, the variable name was not
initially assigned a data but was only assigned
later by calling the variable name.</p>

<script>
var name;
name = "John";

document.write("The student's name is " + name);
</script>
```

View on the Browser:



JavaScript Variables

In this example, the variable name was not initially assigned a data but was assigned later by calling the variable name.

The student's name is John

JavaScript Data types

JavaScript Datatypes

- Datatypes in JavaScript describe the different types or kinds of data that the program can work with and store in variables.
- There are five basic, or primitive, types of data and they are:
  - Strings
  - Numbers
  - Booleans
  - Undefined
  - Null

JavaScript Datatypes

- Strings
  - A collection of alphanumeric characters, enclosed in quotation marks.
  - Example of alphanumerics: a person’s name, street number, email, etc.
  - Examples: “John” , “5023” , “johndoe@gmail.com”

- Numbers
  - Digits including negatives and decimals. They can be used for mathematical calculations.
  - Example of digits: a person’s age, temperature, total sales, etc
  - Examples: 34, -10, 188.95

JavaScript Datatypes

- Boolean
  - Boolean is like a on/off switch.
  - It has two values – true or false.
- Null
  - Basically null means empty or no value.
  - If a variable is assigned null ie: var *name* = null, that means the variable *name* contains a value but it’s empty.
- Undefined
  - Undefined happens when a program is requesting data from a variable that has not been assigned any value in it or the variable hasn’t been declared yet.

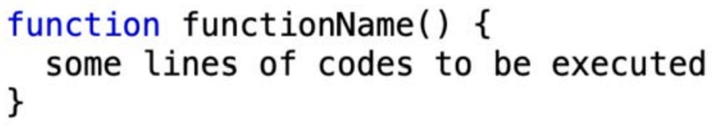
Functions

What are Functions

- A JavaScript function is a block of code designed to perform a particular task. The code block stays inactive ie. is not run or executed until it is called.
- It is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

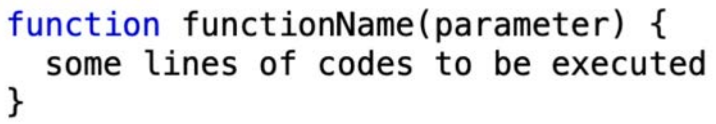
How Functions are created

- The basic syntax:



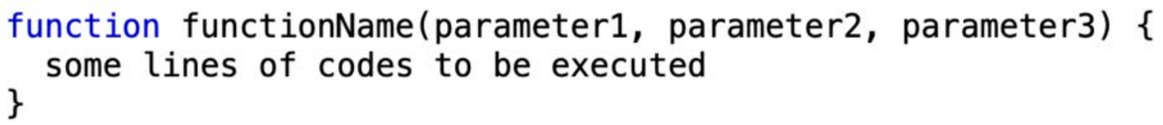
```
function functionName() {
  some lines of codes to be executed
}
```

- Functions can also accept a parameter:



```
function functionName(parameter) {
  some lines of codes to be executed
}
```

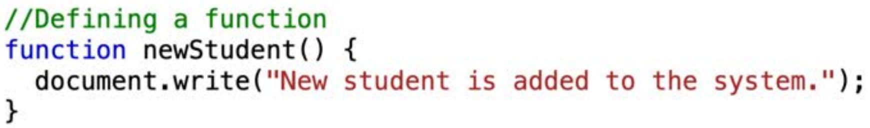
- Or multiple parameters:



```
function functionName(parameter1, parameter2, parameter3) {
  some lines of codes to be executed
}
```

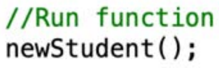
How Functions are created and used - Examples

1. Creating a simple function:



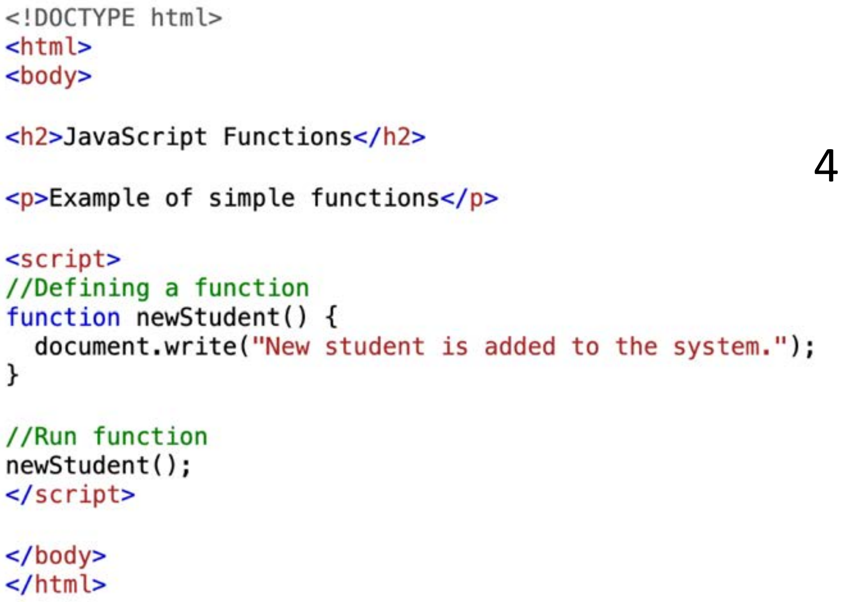
```
//Defining a function
function newStudent() {
  document.write("New student is added to the system.");
}
```

2. Calling the function:



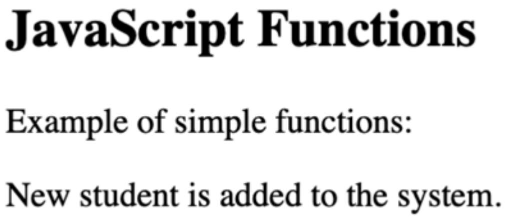
```
//Run function
newStudent();
```

3. Full Code:



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

<h2>JavaScript Functions</h2>

<p>Example of simple functions</p>

<script>
//Defining a function
function newStudent() {
  document.write("New student is added to the system.");
}

//Run function
newStudent();
</script>

</body>
</html>
```

4. View on Browser:



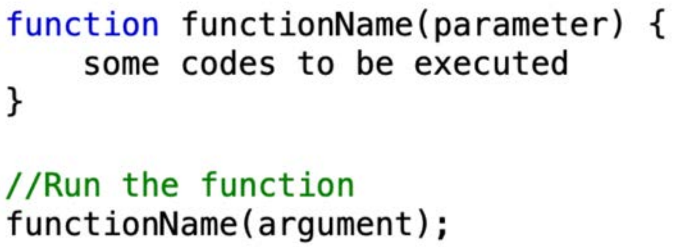
JavaScript Functions

Example of simple functions:

New student is added to the system.

How Functions are created and used - Examples

- The next example shows how a function takes a given parameter and pass it to its block of codes.
- For a function to accept a parameter, the parameter will likely come from calling of the function, where it is refer to as an argument. Here's how it looks like:



```
function functionName(parameter) {
    some codes to be executed
}

//Run the function
functionName(argument);
```

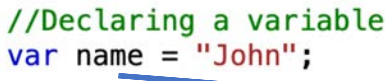
NOTES

Arguments can be one of the two:

1. It can be a static value – string, number, boolean, etc.
2. It can also be a variable that contain data/value that can change overtime.

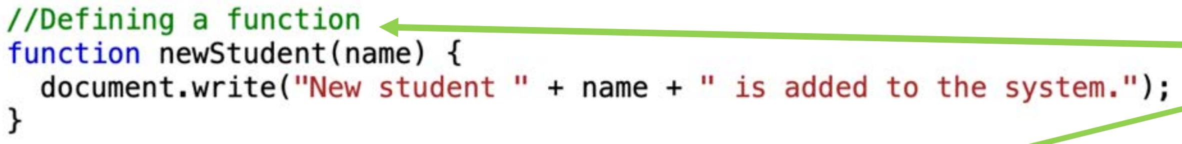
How Functions are created and used - Examples

1. Create variable that contains the data:



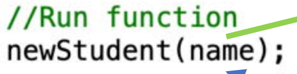
```
//Declaring a variable
var name = "John";
```

2. Creating a function that accepts a parameter:



```
//Defining a function
function newStudent(name) {
  document.write("New student " + name + " is added to the system.");
}
```

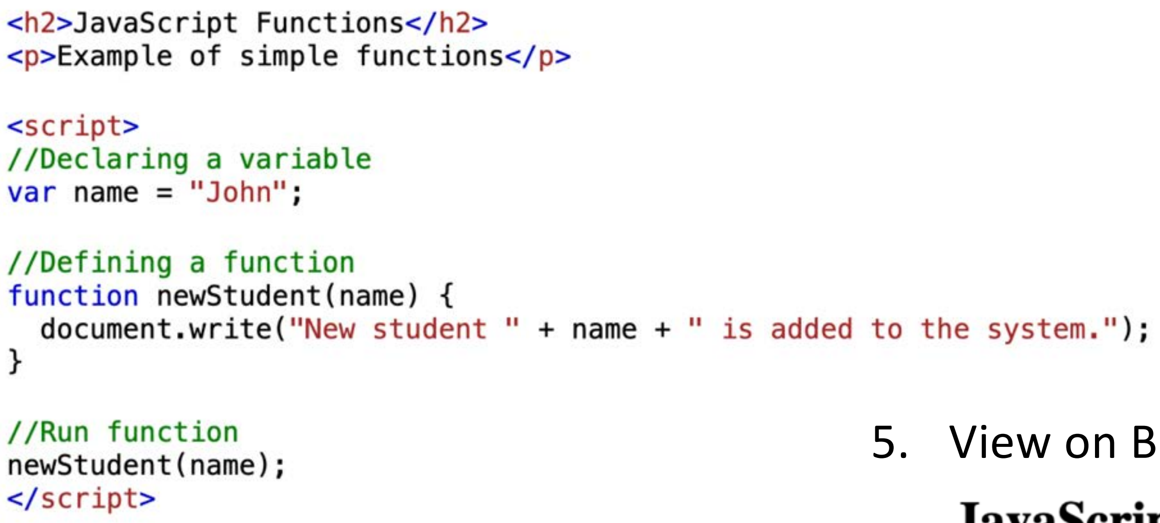
3. Run the function:



```
//Run function
newStudent(name);
```

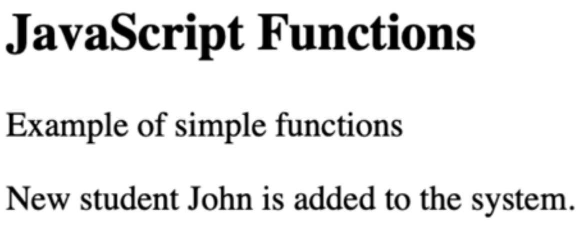
How Functions are created and used - Examples

4. Full Code:



```
<h2>JavaScript Functions</h2>
<p>Example of simple functions</p>

<script>
//Declaring a variable
var name = "John";

//Defining a function
function newStudent(name) {
  document.write("New student " + name + " is added to the system.");
}

//Run function
newStudent(name);
</script>
```

5. View on Browser:



JavaScript Functions

Example of simple functions

New student John is added to the system.

Operators

What are operators and their meanings

- Let us take a simple expression **4 + 5 is equal to 9**. In this mathematical expression, 4 and 5 are called **operands** and ‘+’ is called **operator**. JavaScript supports the following types of operators.
    1. Arithmetic Operators
    2. Comparison Operators
    3. Logical (or Relational) Operators
    4. Assignment Operators
    5. Conditional (or ternary) Operators

What are operators and their meanings

Arithmetic Operators

+	addition
-	subtraction
*	multiplication
/	division
++	increment by 1
--	decrement by 1
+=	increment by specified number
-=	decrement by specified number

Examples:

var a = 10;
var b = 5;

a + b = 15
a++ will give 11
b- - will give 4
a+=5 will give 15
b-=1 will give 4

What are operators and their meanings

Comparison Operators

==	equal
!=	not equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

* Comparison operators check by comparing 2 data with an outcome of true or false

Examples:

```
var  a = 10;
var b = 5;
var c = 5;
```

a == b	will give false
a != b	will give true
b > a	will give false
b <= c	will give true

What are operators and their meanings

Logical Operators

&&	logical AND
||	logical OR
!	Logical NOT

* Logical operators check by comparing 2 pieces of data or the result of it with an outcome of true or false

Examples:

```
var  a = true;
var b = false;

a && b will give false
a || b   will give true
!(b && a)   will give true
!(a || b)     will give false
```

What are operators and their meanings

Assignment Operators

=	assign
+=	add and assign
-=	subtract and assign
*=	multiply and assign
/=	divide and assign

Examples:

```
var  a = 10;
var b = 5;
var result;

result = (a = b)       will give 5
result = (a += b)      will give 15
result = (a -= b)      will give 5
result = (a *= b)      will give 50
result = (a /= b)      will give 2
```

What are operators and their meanings

Conditional or Ternary Operator

?: use for testing if a condition is true, output X if true and output Y if false

Syntax:

(condition) ? X : Y

Examples:

```
var  a = 10;
var b = 5;

(a > b) ? true : false;    will give true
(a < b) ? a : b;           will give 5
```

Decision Making

What is Decision Making

- As humans we make decisions on a daily basis like whether to skip breakfast, take the train instead of the bus, to walk the dogs, etc.
- In coding, we write a JavaScript program to make decisions as well. That decision will involve performing certain actions based on different decisions made. The decision making part here is refer to as condition.
- Here's an example using a real life condition:

  If skip breakfast, have a big lunch.

  Otherwise, eat a healthy breakfast.

Condition – skip breakfast (ie. to skip or not to skip)
Two possible actions or outcome here:
1. Have a big lunch (if skipping)
2. Eat a healthy breakfast (if not skipping)

How Decision Making is used

- In JavaScript, decision making is represented with conditional statements.
- There a 2 type of statements:
  1. If - Else statement
  2. Switch statement

How Decision Making is used

- First, let’s look at the basic syntax of a simple **if** statement:

  if (condition) { action }

- An **if – else** statement:

  if (condition) { action 1 } else { action 2 }

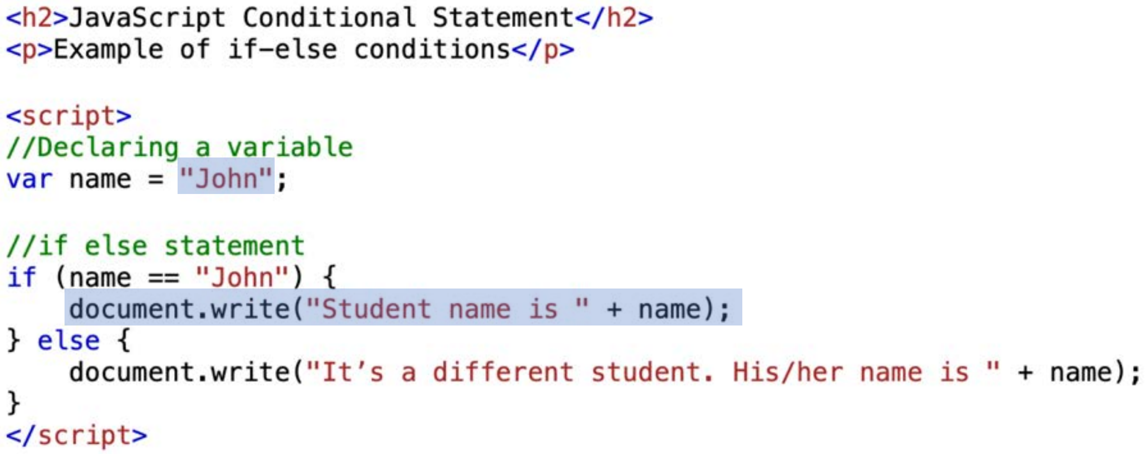
- An **if – else if – else** statement:

  if (condition 1) { action 1 } else if (condition 2) { action 2 } else { action 3 }

- You have the option to create multiple **else if** conditions if required

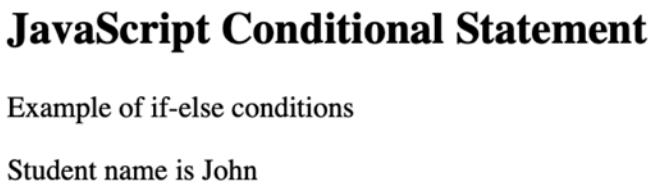
How Decision Making is used

- A simple example of how if-else is use in JavaScript:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else conditions</p>

<script>
//Declaring a variable
var name = "John";

//if else statement
if (name == "John") {
    document.write("Student name is " + name);
} else {
    document.write("It’s a different student. His/her name is " + name);
}
</script>
```

- What it looks like when view on the browser:



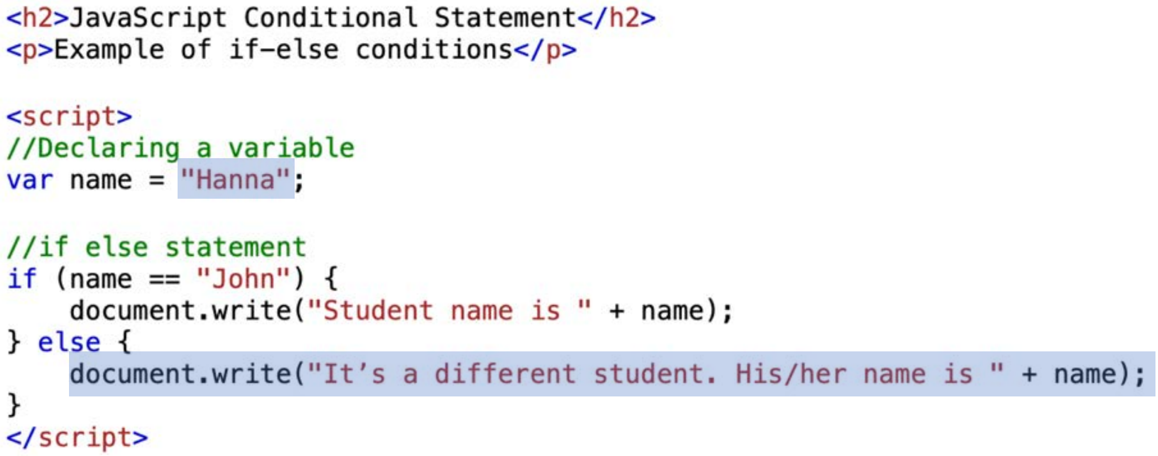
JavaScript Conditional Statement

Example of if-else conditions

Student name is John

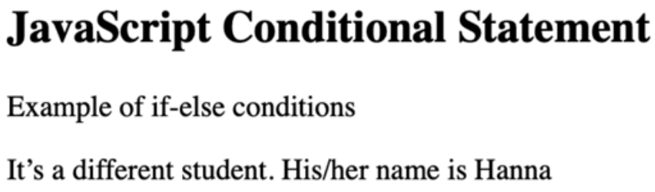
How Decision Making is used

- A simple example of how if-else is use in JavaScript:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else conditions</p>

<script>
//Declaring a variable
var name = "Hanna";

//if else statement
if (name == "John") {
    document.write("Student name is " + name);
} else {
    document.write("It’s a different student. His/her name is " + name);
}
</script>
```

- What it looks like when view on the browser:



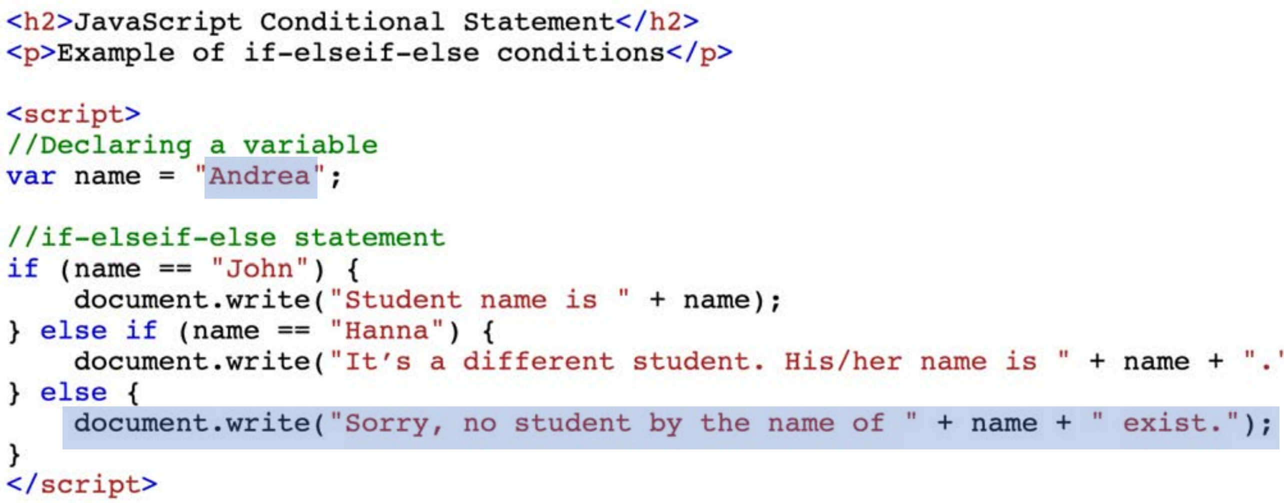
JavaScript Conditional Statement

Example of if-else conditions

It’s a different student. His/her name is Hanna

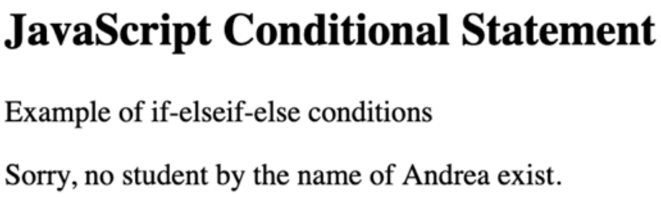
How Decision Making is used

- An example of how if-elseif-else is use in JavaScript:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-elseif-else conditions</p>

<script>
//Declaring a variable
var name = "Andrea";

//if-elseif-else statement
if (name == "John") {
    document.write("Student name is " + name);
} else if (name == "Hanna") {
    document.write("It’s a different student. His/her name is " + name + ".");
} else {
    document.write("Sorry, no student by the name of " + name + " exist.");
}
</script>
```

- What it looks like when view on the browser:



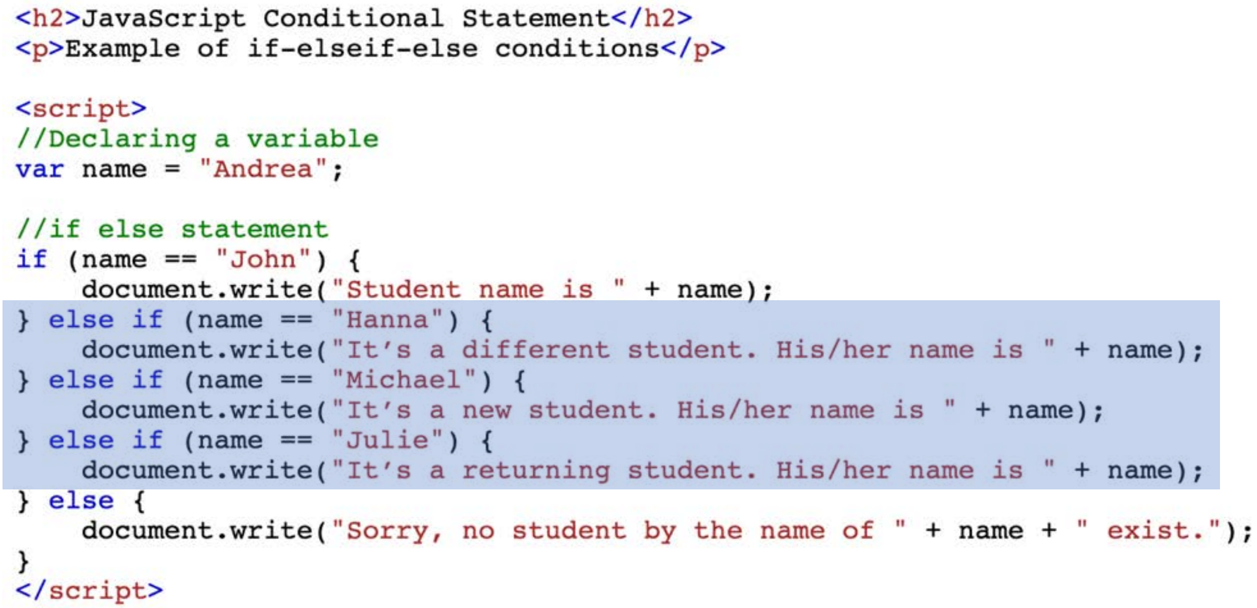
JavaScript Conditional Statement

Example of if-elseif-else conditions

Sorry, no student by the name of Andrea exist.

How Decision Making is used

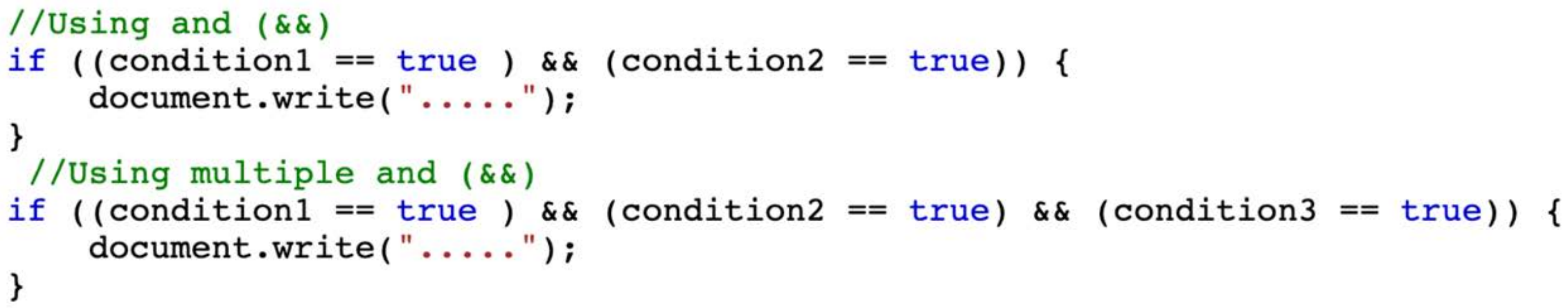
- How many elseif within a conditional statement can you have?
  - ➢ You can have multiple elseif:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-elseif-else conditions</p>

<script>
//Declaring a variable
var name = "Andrea";

//if else statement
if (name == "John") {
    document.write("Student name is " + name);
} else if (name == "Hanna") {
    document.write("It’s a different student. His/her name is " + name);
} else if (name == "Michael") {
    document.write("It’s a new student. His/her name is " + name);
} else if (name == "Julie") {
    document.write("It’s a returning student. His/her name is " + name);
} else {
    document.write("Sorry, no student by the name of " + name + " exist.");
}
</script>
```

How Decision Making is used

- So far we have been testing a single condition no matter how many elseif we used. You can actually test multiple conditions at the same time.
- There are two types of multiple conditions that you can test:
  - and (&&)
  - or (||)
- Here’s how it is used:

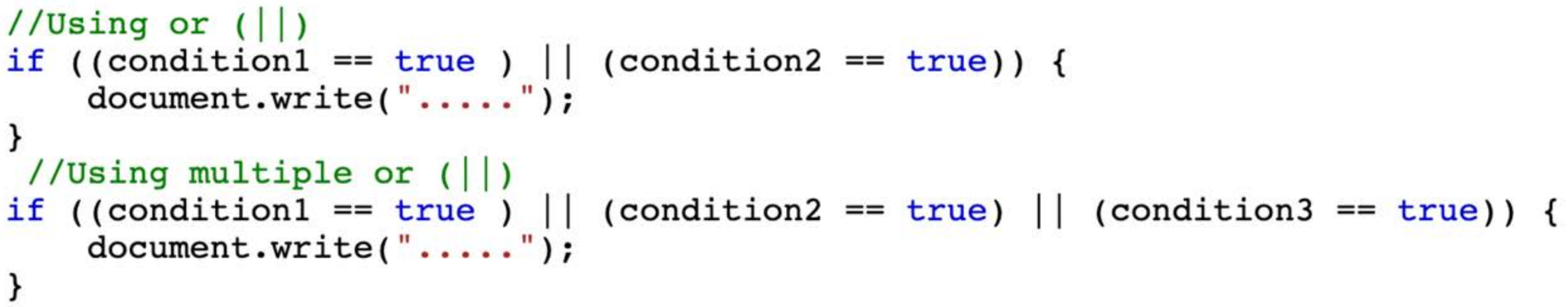


```
//Using and (&&)
if ((condition1 == true ) && (condition2 == true)) {
    document.write(".....");
}
 //Using multiple and (&&)
if ((condition1 == true ) && (condition2 == true) && (condition3 == true)) {
    document.write(".....");
}
```

Note: ALL the conditions must be evaluated to true to satisfy the test.

How Decision Making is used

- And this:

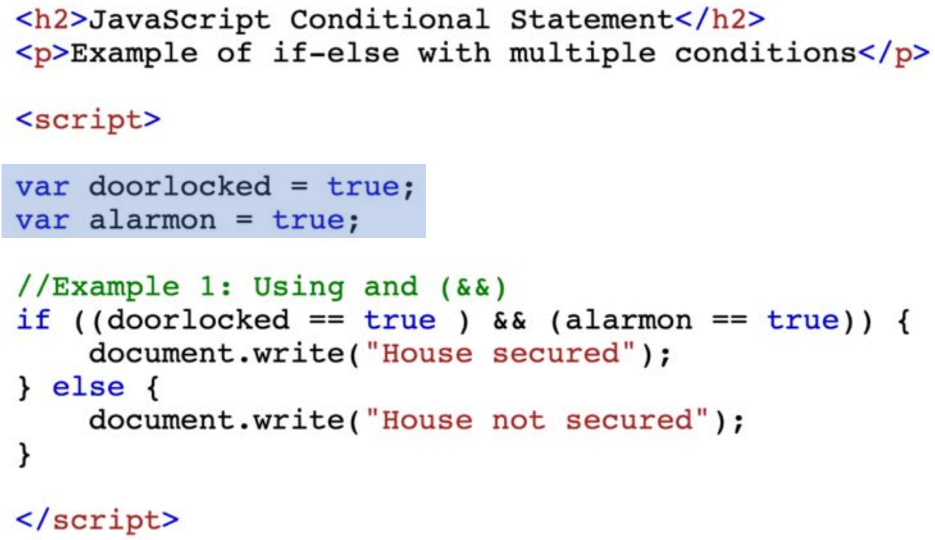


```
//Using or (||)
if ((condition1 == true ) || (condition2 == true)) {
    document.write(".....");
}
 //Using multiple or (||)
if ((condition1 == true ) || (condition2 == true) || (condition3 == true)) {
    document.write(".....");
}
```

Note: One or more of the conditions must be evaluated to true to satisfy the test.

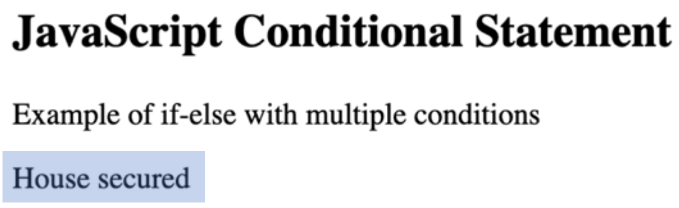
How Decision Making is used

- Let’s look at some practical examples:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else with multiple conditions</p>

<script>

var doorlocked = true;
var alarmon = true;

//Example 1: Using and (&&)
if ((doorlocked == true ) && (alarmon == true)) {
    document.write("House secured");
} else {
    document.write("House not secured");
}

</script>
```

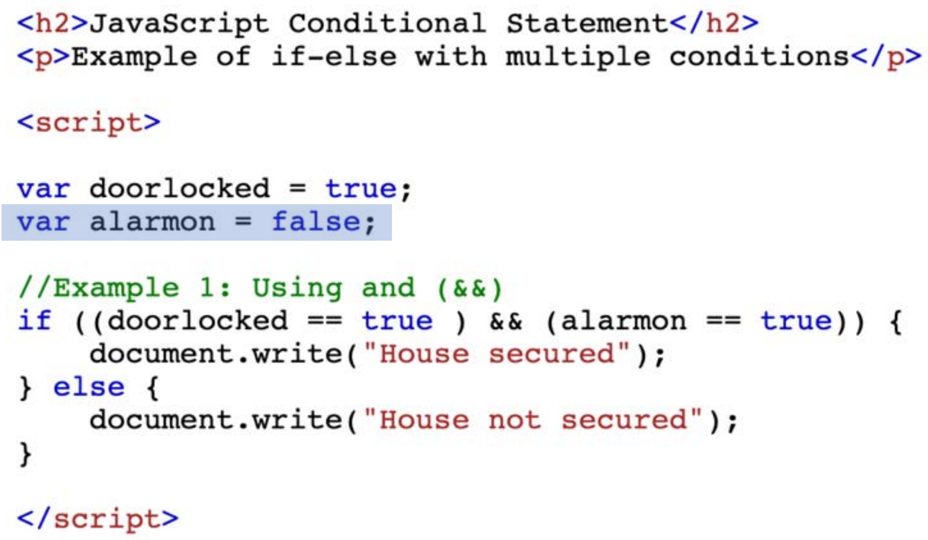
Output on the browser:



```
JavaScript Conditional Statement

Example of if-else with multiple conditions

House secured
```

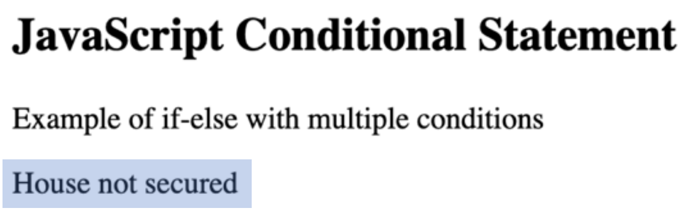
How Decision Making is used

- Let’s look at some practical examples:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else with multiple conditions</p>

<script>

var doorlocked = true;
var alarmon = false;

//Example 1: Using and (&&)
if ((doorlocked == true ) && (alarmon == true)) {
    document.write("House secured");
} else {
    document.write("House not secured");
}

</script>
```

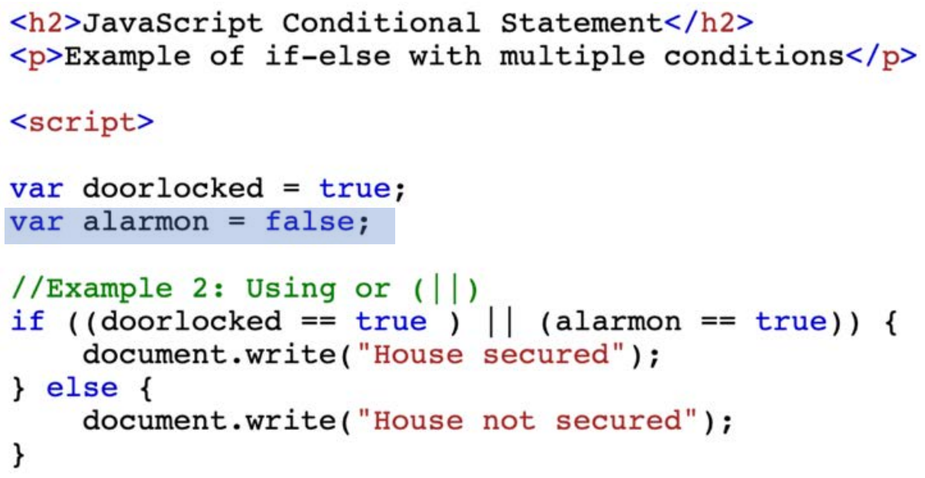
Output on the browser:



```
JavaScript Conditional Statement

Example of if-else with multiple conditions

House not secured
```

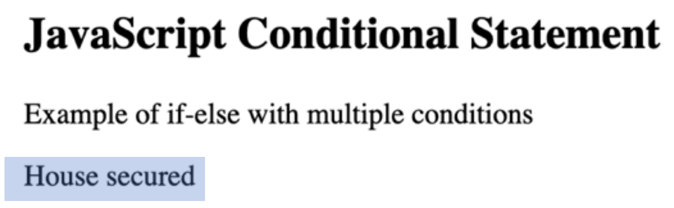
How Decision Making is used

- Let’s look at some practical examples:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else with multiple conditions</p>

<script>

var doorlocked = true;
var alarmon = false;

//Example 2: Using or (||)
if ((doorlocked == true ) || (alarmon == true)) {
    document.write("House secured");
} else {
    document.write("House not secured");
}
```

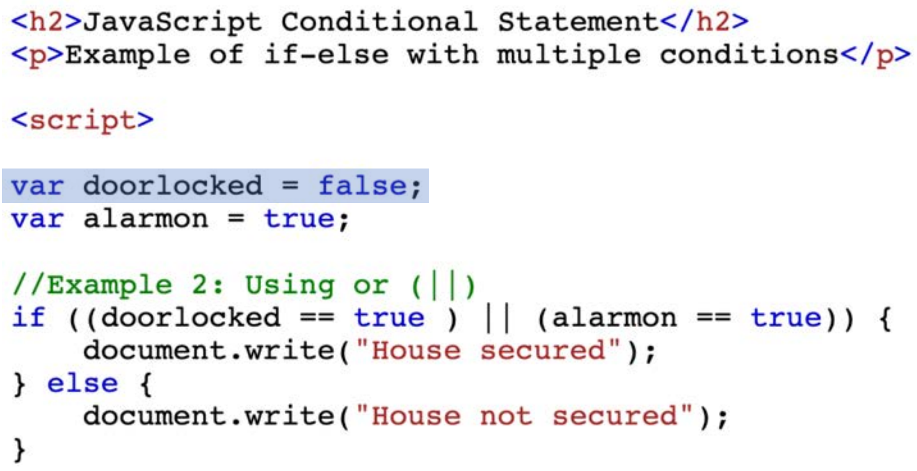
Output on the browser:



```
JavaScript Conditional Statement

Example of if-else with multiple conditions

House secured
```

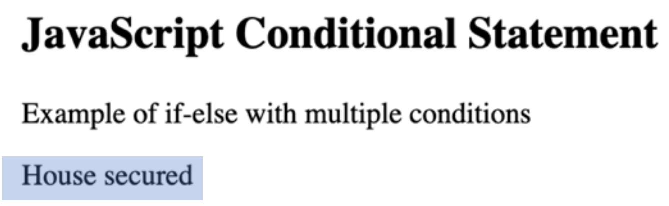
How Decision Making is used

- Let’s look at some practical examples:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of if-else with multiple conditions</p>

<script>

var doorlocked = false;
var alarmon = true;

//Example 2: Using or (||)
if ((doorlocked == true ) || (alarmon == true)) {
    document.write("House secured");
} else {
    document.write("House not secured");
}
```

Output on the browser:



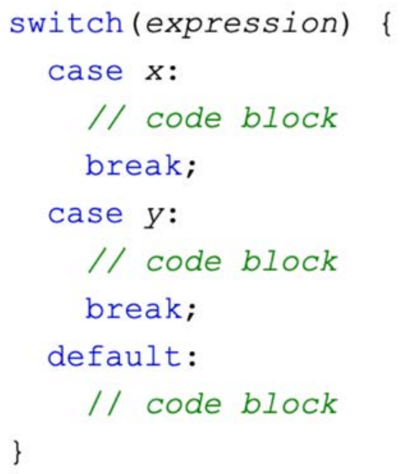
JavaScript Conditional Statement

Example of if-else with multiple conditions

House secured

How Decision Making is used

- Next we will discuss the other conditional statement: **switch**.
- Like if - else statement, switch could also evaluate multiple conditions.
- First, let’s look at the basic syntax of a simple **switch** statement:

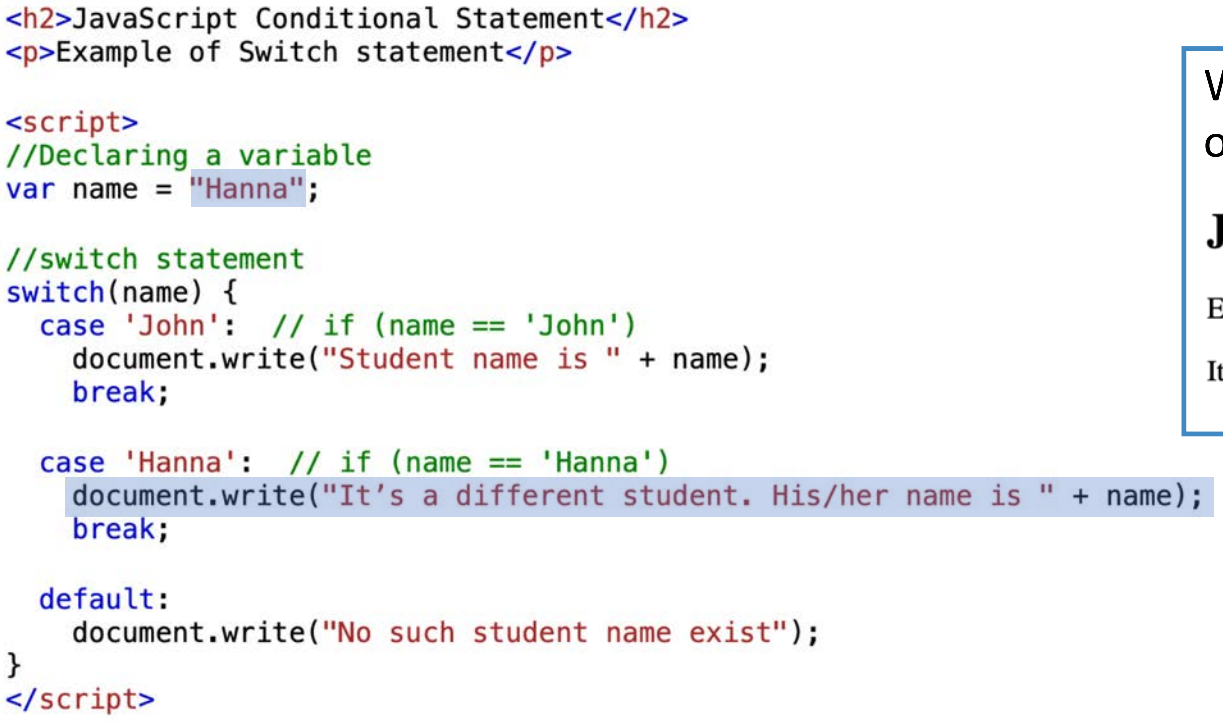


```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

expression - what are you evaluating
case x, *case y* – different conditions
default – same as *else* in if statement
break – exit if that condition is evaluate to true

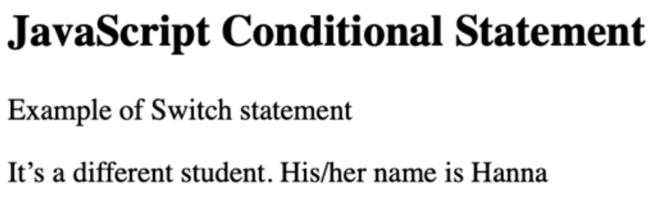
How Decision Making is used

- A simple example of how switch is use in JavaScript:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of Switch statement</p>

<script>
//Declaring a variable
var name = "Hanna";

//switch statement
switch(name) {
  case 'John':  // if (name == 'John')
    document.write("Student name is " + name);
    break;

  case 'Hanna':  // if (name == 'Hanna')
    document.write("It’s a different student. His/her name is " + name);
    break;

  default:
    document.write("No such student name exist");
}
</script>
```

What it looks like when view on the browser:



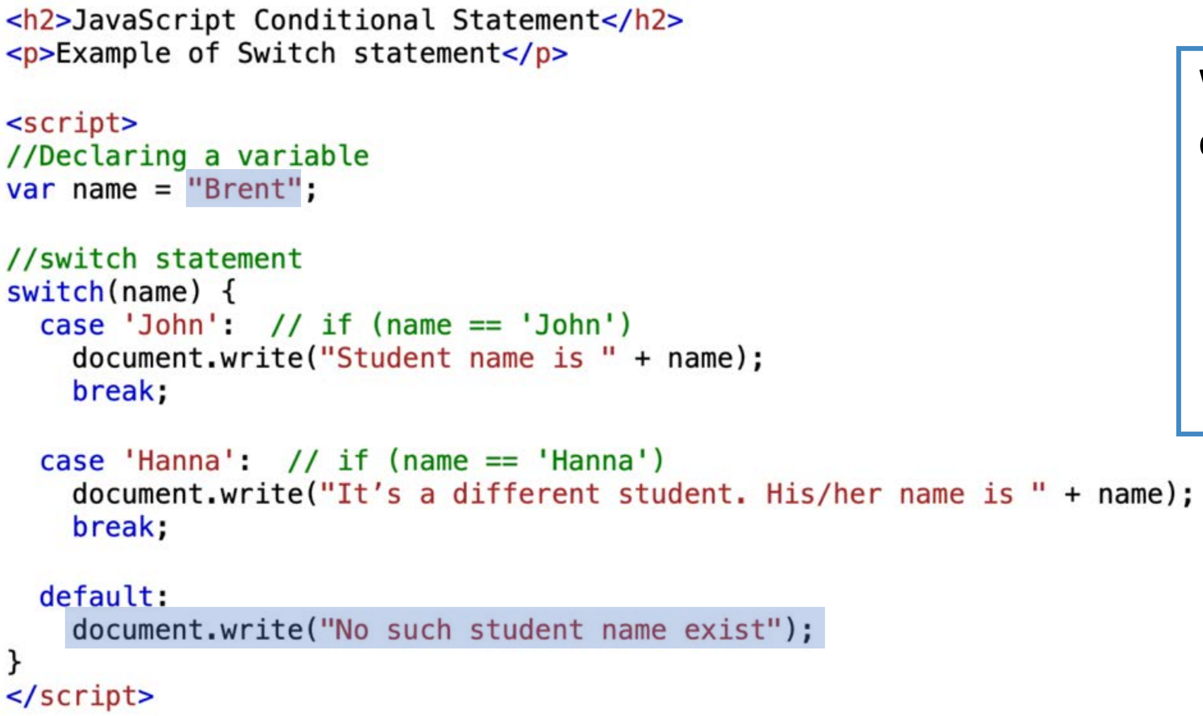
JavaScript Conditional Statement

Example of Switch statement

It’s a different student. His/her name is Hanna

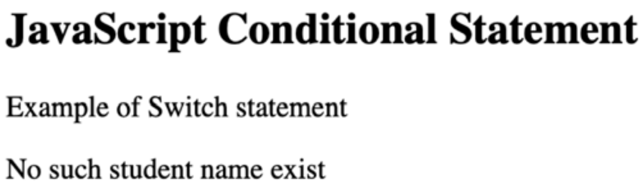
How Decision Making is used

- A simple example of how switch is use in JavaScript:



```
<h2>JavaScript Conditional Statement</h2>
<p>Example of Switch statement</p>

<script>
//Declaring a variable
var name = "Brent";

//switch statement
switch(name) {
  case 'John':  // if (name == 'John')
    document.write("Student name is " + name);
    break;

  case 'Hanna':  // if (name == 'Hanna')
    document.write("It's a different student. His/her name is " + name);
    break;

  default:
    document.write("No such student name exist");
}
</script>
```

What it looks like when view on the browser:



JavaScript Conditional Statement

Example of Switch statement

No such student name exist

Resources

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

https://www.w3schools.com/js/js_whereto.asp

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var

https://www.w3schools.com/js/js_functions.asp

https://www.w3schools.com/js/js_operators.asp

https://www.tutorialspoint.com/javascript/javascript_ifelse.htm

https://www.w3schools.com/js/js_switch.asp

Questions?