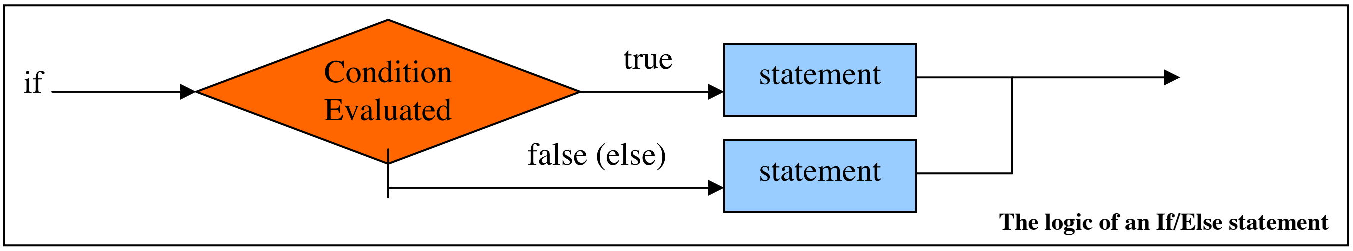
SELECTION: THE IF/ELSE STATEMENT

We often encounter the need to make decisions in our programs. Should the question be addition or subtraction? Should the program continue or stop? We often want to control what happens next based on the current circumstances or on some computed likelihood.

Control statements allow us to control the flow of our program's logic. Control statements make decisions based on the result of a condition. The condition produces a Boolean result, which determines which statements are executed next.

The basic control statement that allows us to make a decision is the IF/ELSE statement.

The logic of an if/else statement ensures that one set of statements **OR** the other is executed, but **NOT BOTH**.



The logic of an If/Else statement

In JS the if statement takes the form:

```
if (boolean condition) {
    statements to be executed;
}
```

Example:

```
if (age < 18) {
  document.write("Youth becomes you") ;
}
```

The condition that the if statement checks must be a Boolean statement. That is, it must either answer to TRUE or FALSE. You may use the following comparison operators to check a condition:

Operator	Example	Result
= =	a = = b	True if a is equal to b and false otherwise
= = =	a= = =b	True if has equal value and equal type a to b and false otherwise
!=	a!=b	True if a is NOT equal to b and false otherwise
<	a < b	True if a is less than b
<=	a <= b	True if a is less than or equal to b
>	a > b	True if a is greater than b
>=	a >= b	True if a is greater than or equal to b

Since the condition is a Boolean value, a Boolean variable may also be used to check the condition

Example:

```
var isChild = true;
if (isChild) {
  document.write("Youth becomes you") ;
}
```

Often you want to provide an alternate situation when the if statement is False. To do this you use an Else Statement

```
if (condition) {
    block of code to be executed if the condition is true
} else {
    block of code to be executed if the condition is false
}
```

Example:

```
var age = prompt(“Enter your age:”);
if (age < 18) {
    greeting = "Hello young one";
} else {
    greeting = "Ready to vote?";
}
```

The indentation of the If … Else statement is good programming style and has no effect on the execution of the statement. When reading the code, the indentation makes it easier to follow the logic of the statement. This is especially important when the statements become large and complex.

NESTED IF/ELSE STATEMENTS

An if/else statement inside an if/else statement is known as a nested if/else statement. In an if/else statement you have two choices (if not choice a then choice b). What happens if you had more choices? For instance, at a stop sign you have to choose to go left, right or straight. If there are more than 2 choices a nested if/else statements is often a good choice.

The following example shows a nested if/else statement. Notice that there are three choices. A nested if statement is far more efficient than writing 3 separate if statements. If the first condition evaluates to true the others are not even checked, each check takes time and although we can’t notice it in these small programs there will be a noticeable lag on millions of checks.   :

Efficient Nested Loop Example:

```
if (age < 10) {
    greeting = "Hello child";
} else if (age < 18) {
    greeting = "Good day youth, soon
you will vote";
} else {
    greeting = "Ready to vote?";
}
```

Inefficient Example:

```
if (age < 10) {
    greeting = "Hello child";
}
if (age < 18) {
    greeting = "Good day youth, soon
you will vote";
}
if (age >= 18) {
    greeting = "Ready to vote?";
}
```

Logical Operators : AND, OR, Not

Logical operators are used to determine the logic between variables or values.

Given that **x = 6** and **y = 3**, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x == 5 || y == 5) is false
!	not	!(x == y) is true

Extra Investigation:

If would like to see another kind of selection statement check out the SWITCH statement.
https://www.w3schools.com/js/js_switch.asp

Shortcut: JS also has a conditional operator that assigns a value based on a condition.

```
var voteable = (age < 18) ? "Too young":"Old enough";
document.write(“You are “ + voteable + “ to vote”);
```

Knowledge Review

1. What type of operator is used to compare two values?
2. Write the code that will tell a user “hello” if choice=1 and “goodbye” if choice=2.
3. When do you use a nested if/else loop?
4. Determine the possible values of a that will output “valid” in each case:

```
if ((a>0) && (a<10)) {
  document.write("valid");
}
```

```
if ((a>0) || (a<10)) {
  document.write("valid");
}
```

```
if (!((a==0) || (a<10)) {
  document.write("valid");
}
```

```
if ((a<0) && (a>10)) {
  document.write("valid");
}
```

```
if ((a<0) || (a>10)) {
  document.write("valid");
}
```

```
if (!((a>0) && (a<10)) {
  document.write("valid");
}
```

PRACTICE EXERCISES – SELECTION, MATH & RANDOM

Save the following Exercises in a Folder entitled “JS Selection”.

1. Write a program that will accept a score out of 100, and give the corresponding letter grade. The program should give a message of “Invalid entry” for an entry greater than 100 or less than 0. Level 4 80 +; Level 3 70 +; Level 2 60 +; Level 1 50+; R less than 50; any other values return INVALID grade. [Save as s1.html]
2. Practice with nested if statements: Write a program that simulates a Magic Eight Ball. It will ask the user for a question and then shake up a response. You should randomly choose between 8 possible answers – place the answer in the middle of an eight ball. [Save as s2.html]

3. Create a program that will take in a birth date and tell the user his/her zodiac sign with a picture. For example: if the user enters September 18 – Output – you are a VIRGO (Level 3: simplified use months only; Level 4: include the date) [Save as s3.html]
4. Practice with operators and if statements. Write a program to determine if the year entered by the user is a leap year. A year is a leap year if it **is** divisible by 4 but not by 100. If a year is divisible by 4 and by 100, it is **not** a leap year **unless** it is also divisible by 400. (1996 is a leap year, 1900 is not, but 2000 is!) [Save as s4.html]
5. Create a Higher-Lower Guessing Game. For the game you need to randomly generate a number between 1 and 10. Provide the user with a space to enter a number between 1 and 10. If the guess matches the random number award them a prize and congratulatory message. If the user guesses too high or too low let them know. Give the user 3 tries to get the correct number. For a level 4 - Keep track of and provide a scoring system with prizes. (for example – you could award 100 points if they guess it on the first try, 50 on the 2nd try, and 25 on the 3rd try. Come up with your own prizes. [Save as s5.html]
6. Create a Math Program that will:
   a. Randomly choose two integers between 1 and 15.
   b. Randomly choose to do either an addition question or a subtraction question.
   c. Prompt with a math question using the values generated to allow the user to enter the answer.
   d. Determine and indicate if the answer is correct.
   e. Output the question and the user’s response.
   f. Ask at least 3 questions and keep track of the number of correct responses.
   g. After they have answered 3 questions calculate & display the percentage

   Format your math game appropriately. [Save as s6.html]

	Level 1	Level 2	Level 3	Level 4
Overall Clarity and Organization of code [20]	Needs improvement	Approaching Expectations	Clear	Exceptional
Overall Style and Appearance of pages [10]	Basic	Some Effort	Looks Good	Exceptional
Implementation of required elements.	Shows limited understanding of Selection, Variables, and Math	Some Understanding	Most	Well done, Few Problems, Efficient
s1.html [5] S2.html [5]				
S3.html [5] S4.html [5]				
S5.html [25]				
S6.html [25]				