Students notes

Student task-1:

```
var a = 2;
var b = 3;
var c = (a==b);

// check if a and b are equal or not
//if equal print "a and b are equal"
// "not equal"
```

Data Types:

- Variables can hold different types of data like:
    1. **Primitive Data Types:**
        - **Number:** Represents numeric values, e.g., 42, 3.14, -7.
        - **String:** Represents textual data, e.g., "Hello, World!", 'JavaScript', "123".
        - **Boolean:** Represents true or false values.
        - **Null:** Represents the intentional absence of any object value.
        - **Undefined:** Represents a variable that has been declared but has not been assigned a value.
        - **Symbol:** Introduced in ECMAScript 6 (ES6), symbols are unique and immutable data types, often used as object property keys.

- **BigInt:** Introduced in ES10, BigInts are used for working with arbitrarily large integers.

2. **Non Primitive Data Types:**
    - **Object:** Represents a collection of key-value pairs (properties and methods).
    - **Array:** Represents an ordered list of values, e.g., [1, 2, 3], ["apple", "banana", "cherry"].
    - **Function:** Represents reusable blocks of code that can be executed when called.

Truthy and Falsy values

MDN- truthy

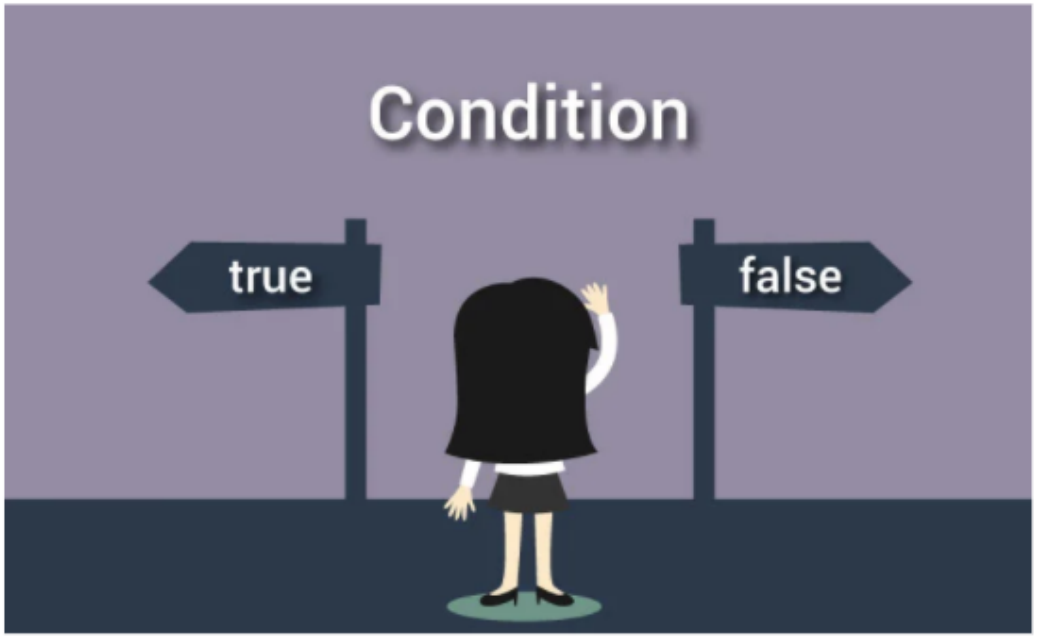
MDN- falsy

The values that can be assigned to a variable can be classified into **primitive** vs **non-primitive** categories. They can also be classified as **truthy** vs **falsy** values.

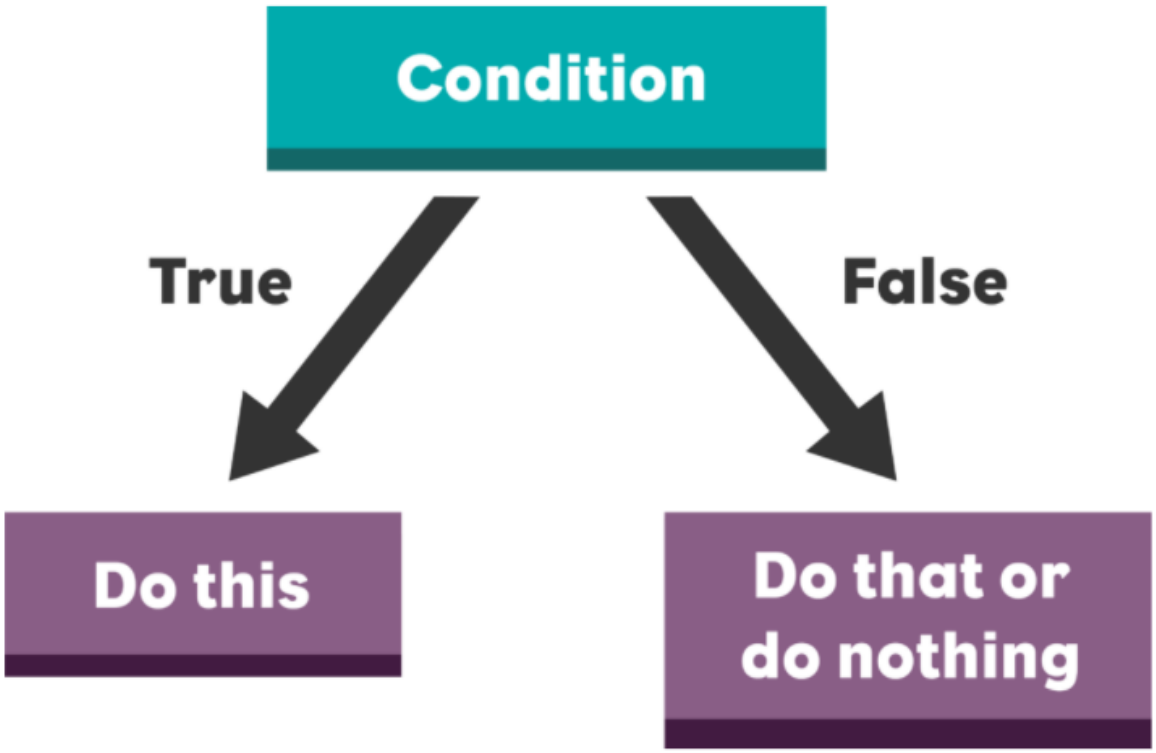
There are only six ***falsy*** values in JavaScript: `undefined` , `null` , `NaN` , `0` , `""` (empty string), and `false`. Anything that is not falsy it truthy.

Conditional Statements

conditional statements and relate it to the daily life



- Conditional statements are used to decide the flow of execution based on different conditions. If a condition is true, you can perform one action and if the condition is false, you can perform another action.
- Through Conditional Statements, we can control which code needs to run or which code will not run.



- Code runs based on certain conditions.
    - For Eg: let's understand with the analogy, the traffic light controls the flow of vehicles on the road. Depending upon the colour of light, the actions happened. If light is green , then it is a signal to move whereas if the light is red then it is a signal to stop.
- Based on the comparison , if the comparison is true then it will execute the one block of code otherwise another block of code.

Different Types of Conditional Statements

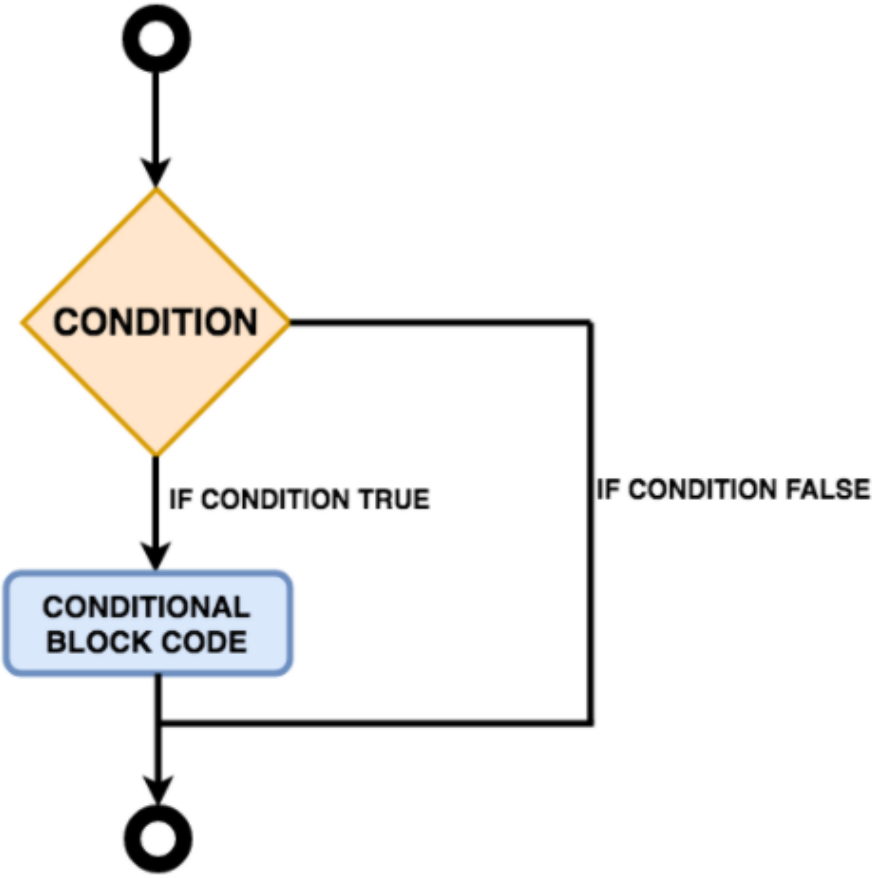
types of conditional Statements

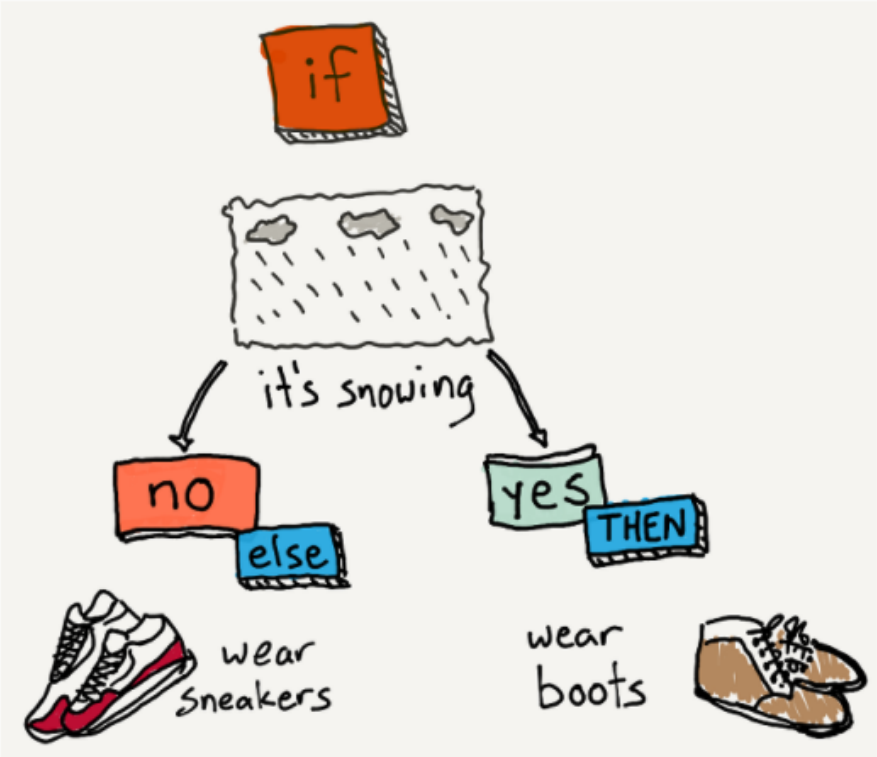
There are mainly three types of conditional statements in JavaScript.

1. If statement
2. If...Else statement
3. If...Else If...Else statement

if Statement

if statement with examples





- It is to specify a block of JavaScript code to be executed if a condition is true.

Syntax

condition

block of code to be executed if the condition is true

- **if()**
  - It takes a Boolean Value or the expression that will give boolean value.
- if() { }
  - { } known as code block.

Understand the if statement with 3 scenarios : with Boolean Value, with Expression and with Variables

a) If with Boolean Value

```
console.log("Code Start")
        if(true) {
          console.log("Inside Code")
        }
console.log("Code End")
```

b) If with Expression

- The decision is based on the value of Expression

For Example :

```
if(5>3){
       console.log("Inside Code");
  }
```

c) If with Variables

- The decision is based on the value of Expression

For Example :

```
var name1 = "rahul";
var name2 = "rahul";
var check = (name1==name2);

if(check){
   console.log("Both Names are same");
}
```

Student Task :

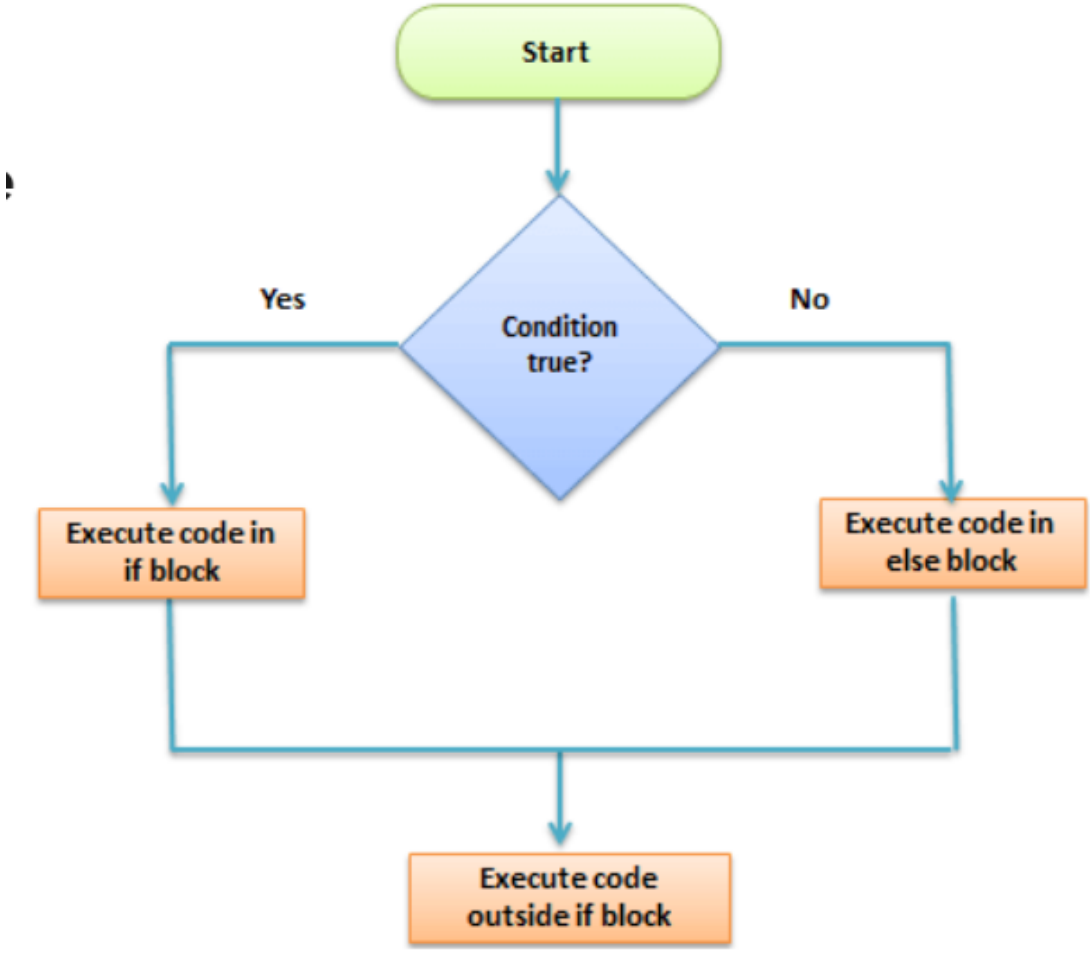
Check Whether two numbers are equal

```
var a = 2;
var b = 3;
var c = (a==b);

if(c)
{
    console.log("a and b are equal");
}
```

if/else Statement

if/else statements with real life example



- The `if...else` is a type of conditional statement that will execute a block of code when the condition in the `if` statement is `truthy`. If the condition is `falsy`, then the `else` the block will be executed.
- Here is a list of `falsy` values:
  - false
  - 0 (zero)
  - 0 (negative zero)
  - 0n (BigInt zero)
  - `""`, `''`, ``` `` ``` (empty string)
  - null
  - undefined
  - NaN (not a number)
- If the condition is true, then one block of code executes.
- Else another block of code executes.

Syntax

condition

block of code to be executed if the condition is true

block of code to be executed if the condition is false

Code 2 : Check which number is greater

```
var a = 3;
var b = 20;

if(a>b)
{
  console.log("a is greater");
}
else
{
```

```
  console.log("a is not greater");
}
```

Code 3 : Check Whether two names are equal or not

Student Task : Task For Students

```
var name1 = "suraj";
var name2 = "suraj";

if(name1==name2)
{
  console.log("Names are Equal");
}
else
{
  console.log("Names are not equal");
}
```

Hotel Bill Discount

Understanding if-else with Hotel Bill Example

Code 4 :

Student Task : Task For Students

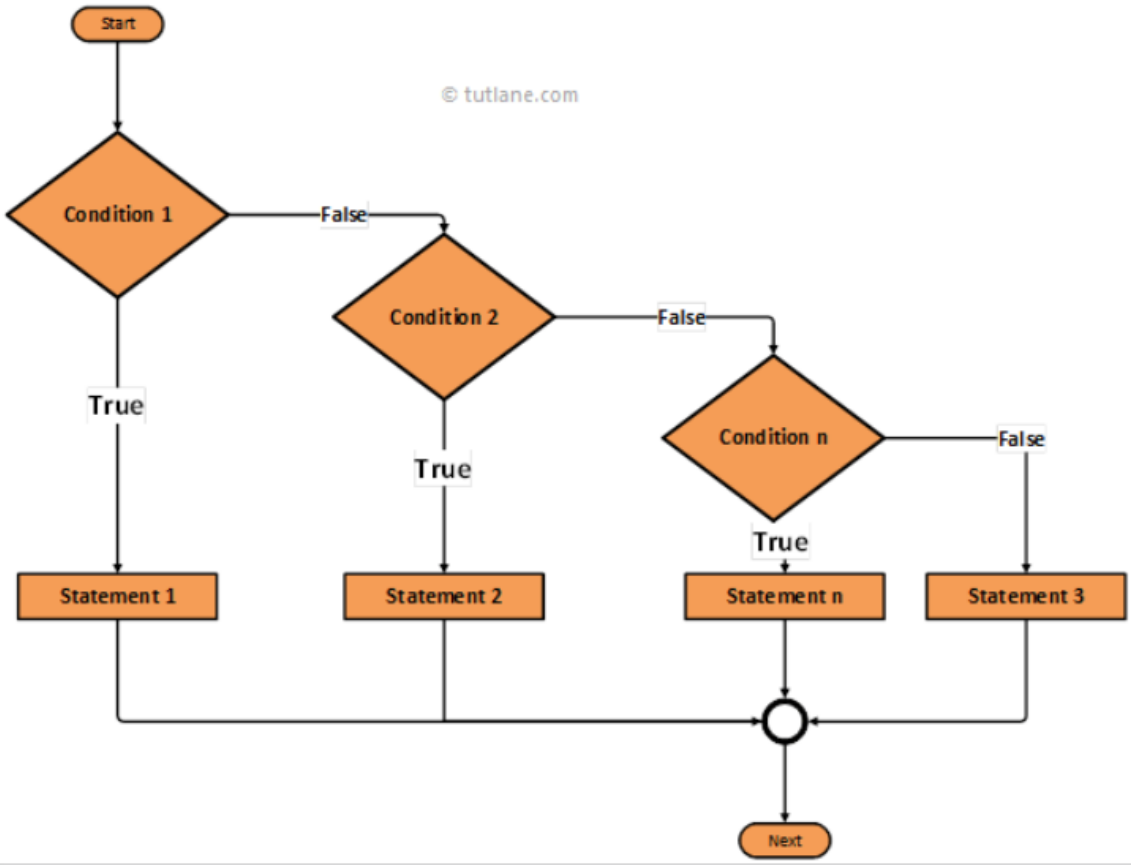
Given total_bill, discount_start_price if you satisfy the condition Print Discount Availaible Otherwise print No Discount

```
var total_bill = 699;
var discount_start_price = 500;

if(total_bill>=discount_start_price){
  console.log("Discount Availaible");
}
else{
  console.log("No discount");
}
```

Else-if Statement

Instructor Task (5 mins): Discuss Else-if Statement with Real-life examples



- There will be times where you want to test multiple conditions. That is where the `else if` the block comes in.
- When the `if` statement is `false`, the computer will move onto the `else if` statement. If that is also `false`, then it will move onto the `else` block.

Syntax

condition1

block of code to be executed if condition1 is true

condition2

block of code to be executed if the condition1 is false and condition2 is true

block of code to be executed if the condition1 is false and condition2 is false

Bill and Discount

Problem Statement: According to the total_bill, the discount will be applied.

Aa Total Bill	Discount Applied
Greater Than 500	10%
Greater Than 1000	20%
Others	No Discount

Code 5 : For a Restaurant, write the program for the following total_bill > 500 Then print 10% discount total_bill > 1000 Then print 20% discount Otherise No discount

```
var total_bill = 799;

if(total_bill > 1000)
{
  console.log("20 % discount");
}
else if(total_bill > 500)
{
  console.log("10 % discount");
}
else
{
  console.log("No discount");
}
```

If-Else-If vs if-if-if :

Instructor Task

Code 6 : If-Else-If

- ***My mother told me to get any one of the thing from the market**

1. **If Rice is availaible then print Buy rice**
2. **Else If wheat is availaible then print buy wheat**
3. **Else If apple is availaible then print buy apple****

```
var rice_availaible = false ;
var wheat_availaible = true;
var apple_availaible = true;

if(rice_availaible)
{
  console.log("Buy rice");
}
else if(wheat_availaible)
{
  console.log("Buy Wheat");
}
else if(apple_availaible)
{
  console.log("Buy apple");
}
else
{
  console.log("Nothing is availaible");
}
```

Code 7 : If - If - If

- ***My mother told me to get all of the thing if available from the market**

1. **If Rice is available then print Buy rice**

2. **If wheat is available then print buy wheat**
3. **If apple is available then print buy apple****

```
var rice_available = true ;
var wheat_available = true;
var apple_available = false;

if(rice_available)
{
  console.log("Buy rice");
}

if(wheat_available)
{
  console.log("Buy Wheat");
}

if(apple_available)
{
  console.log("Buy apple");
}
```

Logical Operators

What is Logical operators.

- A **logical operator** is a symbol or word used to connect two or more expressions.
- The logical operators are important in JavaScript because they allow you to compare variables and do something based on the result of that comparison.
- For example, if the result of the comparison is `true`, you perform a block of code; if it's `false`, you perform another block of code.

Whenever we need to connect two statements.

In the Last class we learn about conditional statements, that says that if one condition is true then do X otherwise do Y.

For Example: In traffic lights , If the lights are green then Move and if the lights are red then Stop.

But In real, there might be multiple condition on which some result depends.

Types of Logical Operators

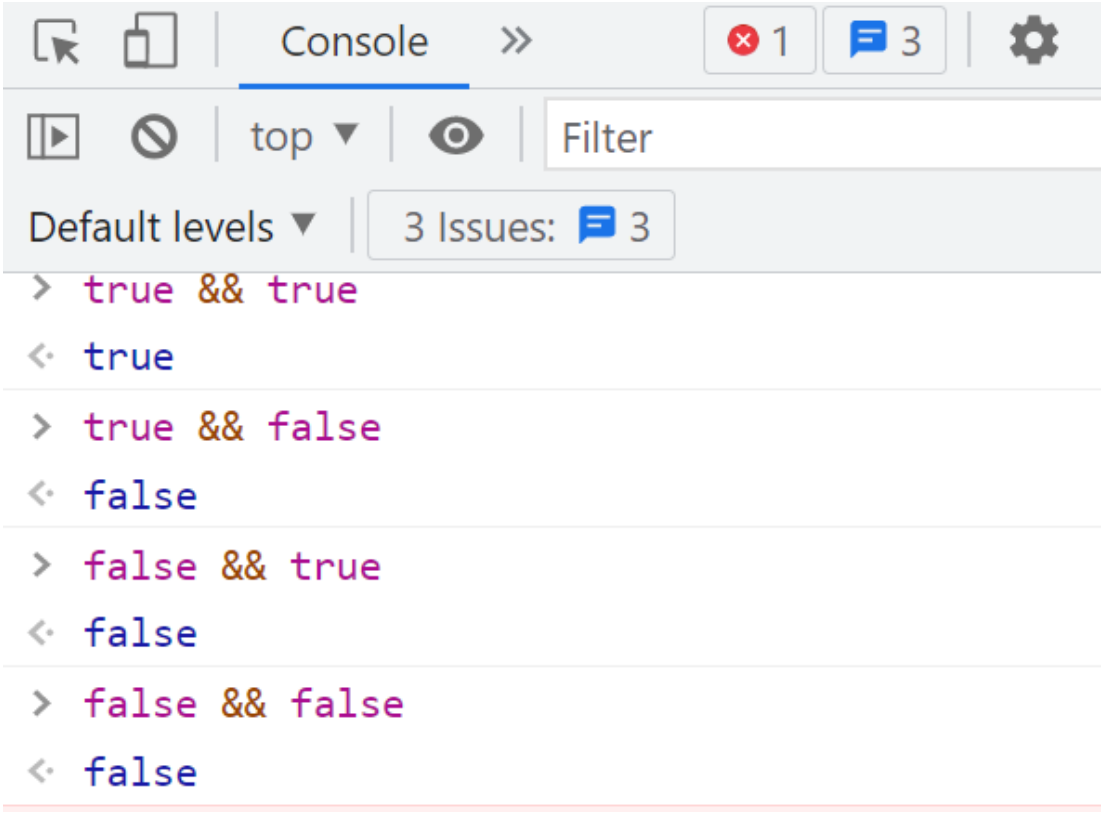
JavaScript provides three logical operators:

1. && (Logical AND)(Checks first falsey value and return it otherwise return last truthy value)
2. || (Logical OR) (Checks first truthy value and return it otherwise return falsey value)
3. ! (Logical NOT)

Logical AND Operator (&&)

Logical AND operator with some examples

- In a chain of `&&`s, each value will be evaluated from left to right. If any of these values happen to be `falsy`, that value will be returned and the chain will not continue.



- **For Example:** Suppose I need to submit some documents in chitkara, and the documents are pan card and License Id, then only I will get the admission.

here, you can observe that I will only get the admission only when I have the PAN Card and License Id (Both are important).

- **Another Example**: Suppose if one goes for a police selection, they check for height and weight right. We need both the parameters to get selected.

Similarly, we can have multiple condition on which , the result is dependent.

- Our Boolean operators takes the input values as boolean and produce the result in boolean.
- In programming, we use to denote the AND operator like in this way **&&**.
- **Input (Boolean Value) ——&&→ Output (Boolean Value)**

Code 1 : AND Operator

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

var c = a && b;
console.log(c);

a = true;
b = false;
console.log(a&&b);

a = false;
b = true;
console.log(a&&b);

a = false;
b = false;
console.log(a&&b);
```

Code 2 : AND with numbers

```
var a = 5>3;
var b = 6>3;

var c = a && b;
console.log(c);
```

Code 3 : if/else

```
// Ist Part : Without AND

if(5>3)
{
    if(6>3)
    {
        console.log("Both are true");
```

```
    }
}

// IInd Part : With AND

if(5>3 && 6>3)
{
    console.log("Both are true");
}
```

Code 4 : Combination of multiple statements

```
// Try out on Console

(5<4) && (3>1) && (2>1) && (4<1)
```

Student Task (5 mins): Solve the following problem

Code 5 : [Student Task] Check whether Rahul passed or not

```
// For English Subject, Check whether Rahul passed or not

var subject = "english";
var passing_marks = 70;

var rahul_marks = 75;
var rahul_subject = "english";

if((rahul_subject == subject) && (rahul_marks >= passing_m
arks))
{
    console.log("Rahul Passed");
}
else
{
```

```
    console.log("Rahul not passed");
}
```

Code 7 : Differentiate between ,(coma) and +

```
var a = 2;
var b = 3;
var c = "hello";

console.log(a,b,c);
console.log(a+b+c);

// Case 2 : Integers
var a = 2;
var b =. 3;
console.log(a+b);
console.log(a,b);

// Case 3 : Strings
var a = "Hello";
var b = "World";

console.log(a+b);

// Case 4 : Integer with Strings
var a = 2;
var b = "hello";

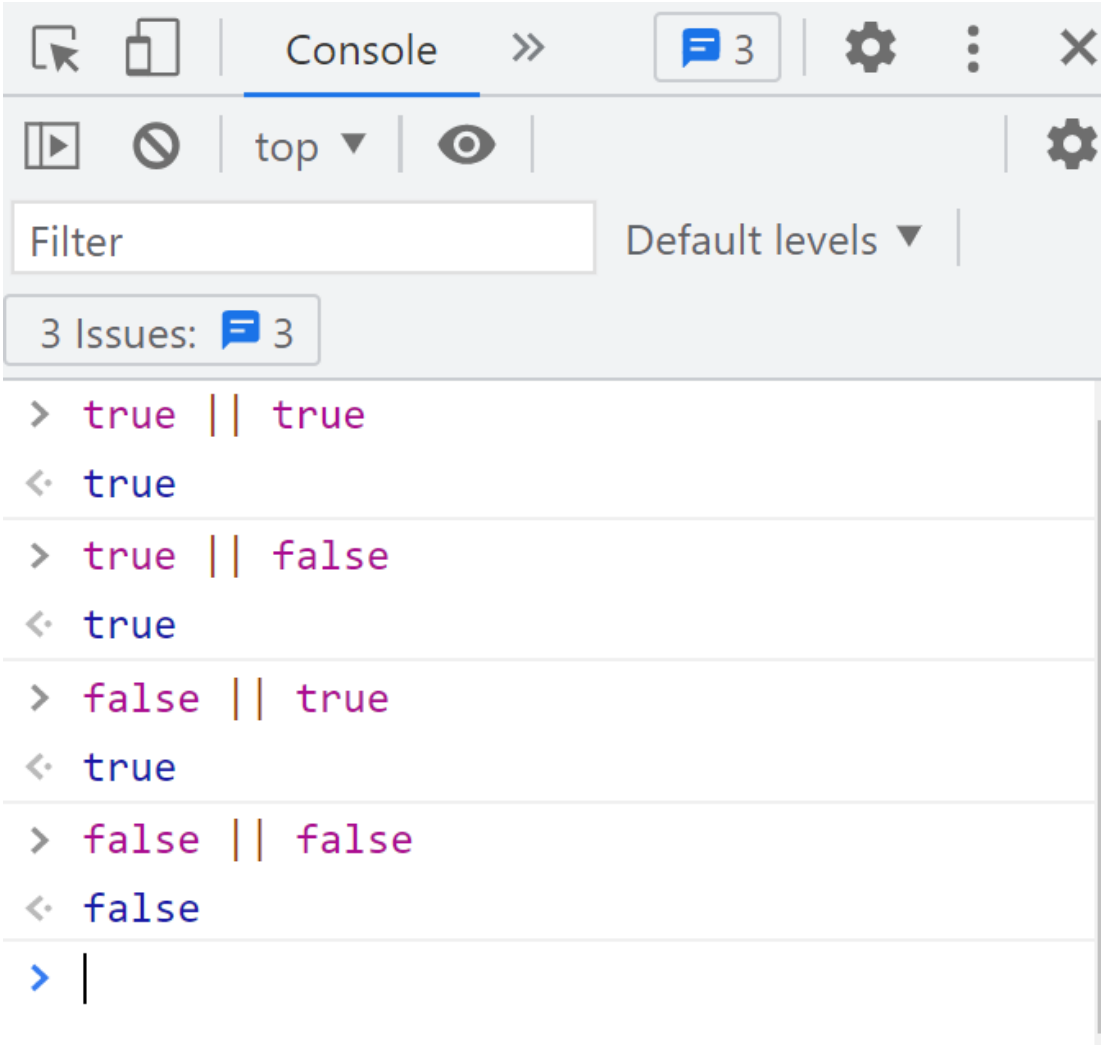
console.log(a,b);
console.log(a+b);
```

Logical OR Operator (||)

- The `||` operator behaves exactly like the `&&` does, only in reverse! While a chain of `&&`s will break if a `falsy` value is found, a chain of `||`s will break

when it finds a `truthy` value. And, just like with the `&&`s, if there are no `truthy` values and the end of the chain is reached, the last value in that chain is returned.

- If any of the statement is true , then the result will be true



For Example : DriveZy is a Renting bike service Startup, If you want to rent a bike then you need to submit any of the Identity Document

Aadhar Card or PAN Card or License or Voter id Card

Aa Title	☑ Aadhar Card	☑ PAN Card	☑ License	☑ Voter ID card	☑ Result
Untitled	☑	☑	☑	☑	☑

Aa Title	☑ Aadhar Card	☑ PAN Card	☑ License	☑ Voter ID card	☑ Result
Untitled	☑	☑	☑	☐	☑
Untitled	☑	☑	☐	☐	☑
Untitled	☑	☐	☑	☑	☑
Untitled	☐	☐	☐	☐	☐

and many more cases are possible

Observation :

1. If any of the case is true then the final result will be true.
2. If all the cases are false, then only the result will be false.

For Example : Masai ask for documents After Msat in the documentation phase, either submit the 12th Mark Sheet or Diploma

Aa Title	☑ 12th Marksheet	☑ Diploma	☑ Admission Result
Untitled	☑	☑	☑
Untitled	☑	☐	☑
Untitled	☐	☑	☑
Untitled	☐	☐	☐

- Show in console

Code 9 : OR Operator

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

var c = a || b;
console.log(c);
```

```
a = true;
b = false;
console.log(a||b);

a = false;
b = true;
console.log(a||b);

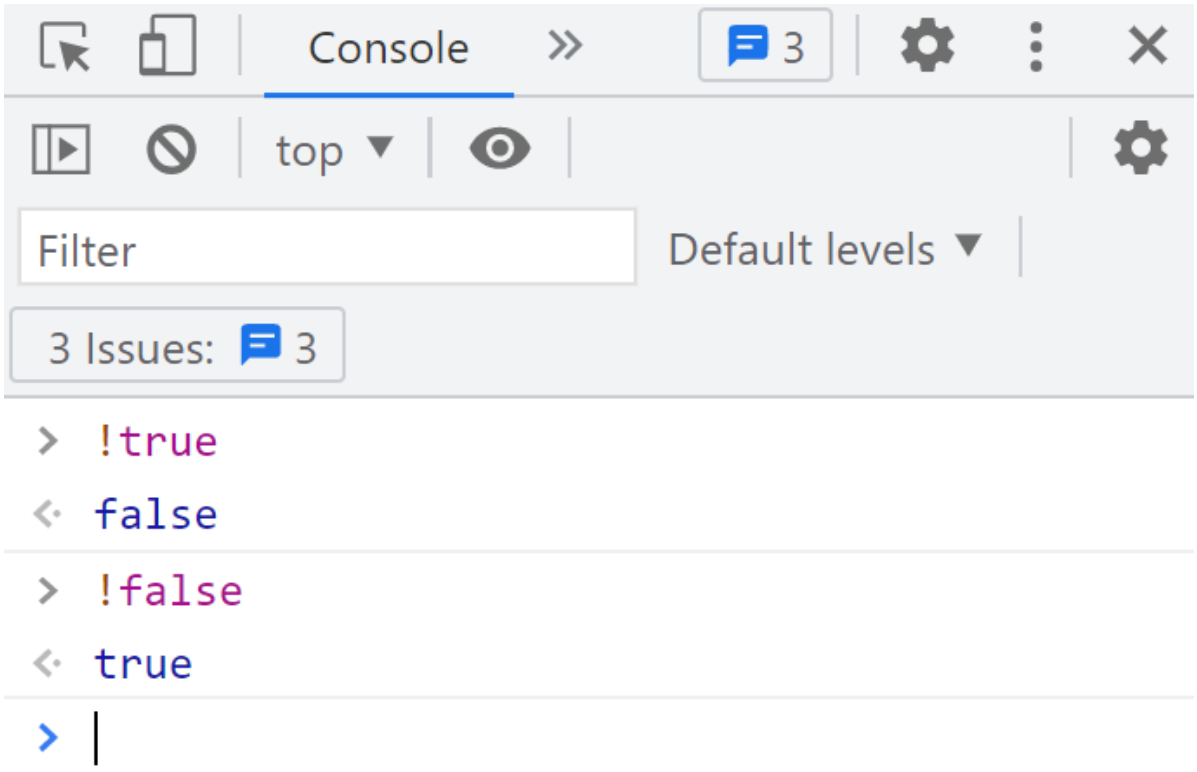
a = false;
b = false;
console.log(a||b);
```

Student Task (5 mins): What is the output of the following

Code 10 : [Student Task] OR Operator

1. true || false || true
2. false || true || false
3. false || false || true

Logical NOT Operator (!)



- The boolean NOT operator is represented with an exclamation sign !.
- The operator accepts a single argument and converts it into a boolean and then it inverses the output.
- Take an example of watching a movie on your laptop and you want to sit under a fan since it's very hottt. There are 2 switches for a fan, say. Initially both the switches are off. Now, you on SwitchA, the fan runs. Then , start watching the movie. Now, you the movie is finally finished or you want to go out with a friend. So, you shut down your laptop and Switch B is close to you, so you click the Switch B on but the fan stops running. So, the

action you're doing for Switch B is true but the fan stops running, i.e , the output is false.

- On applying to a boolean value, the *not* operator turns *true* to *false* and *false* to *true*.

For Example :

- For any website, there are multiple roles
  - Admin
  - User
- ```
  var admin_access = false;
  if(!admin_access){ console.log("Access Denied");
  }
  else{ console.log("Welcome");
  }
  ```

- **The precedence of NOT ! is the highest of all logical operators, so it always executes first, before && or | |.**

IW Assignment
  ```

- IW assignment means I WE Assignment , here we solve some question in the class . So, that student will get comfortable with the problems.
- In this IW Assignment, we will take some problems and work on the solution together with students

Problem 1: If the number is divisible by 3, print a "multiple of 3".

```
var number = 16;
var remainder = number % 3;

if(remainder == 0)
{
  console.log("Multiple of 3");
}
else
{
  console.log("Not multiple of 3");
}
```

Problem 2: If a person is allowed to drive in India print "Apply for a license" or "NA".

```
var yob = 1996;
var age = 2021-yob;

console.log(age);
if(age >= 18){
  console.log("Can Apply for license");
}
else{
  console.log("NA");
}
```

Switch Case

Whenever we have multiple options and we have a choice .

For Ex : ATM Machine , we have multiple options of Deposit, Withdraw, Change Pin , others

- Every option is connected to some code
  - Deposit ———–——→ Code 1 [To deposit the money]
  - Withdraw —————–→ Code 2
  - Change Pin ——–——→ Code 3
  - Default ————–——→ Code 4

In switch case, there are multiple cases, and with each case some code is connected.

Code 13 : Day Schedule

```
var option = 2;

switch(option)
```

```
{
    case 1 :
                    console.log("Day 1 : Scrum + Coding");

    case 2 :
                      console.log("Day 2 : Scrum + Coding +
Skillathon");

    case 3 :
                      console.log("Day 3 : Scrum + Skillath
on + Standups");

    default :
                             console.log("Holiday");

}
```

- On choosing the option in above code, it will output the code present corresponding to the given option and also print all the output of all the cases which present below the chosen option.
- To avoid this, we will use break

Code 14 : Day Schedule [with Break]

```
var option = 3;

switch(option)
{
    case 1 :
                    console.log("Day 1 : Scrum + Coding");
                      break;

    case 2 :
                      console.log("Day 2 : Scrum + Coding +
Skillathon");
                      break;
```

```
    case 3 :
                    console.log("Day 3 : Scrum + Skillath
on + Standups");
           break;

    default :
                    console.log("Holiday");
}
```

Code 16 :

```
 Problem 2 : Given any character, if it is a vowel print
"Vowel"

var char = "i";

if((char == "a") || (char=="e") || (char=="i") || (char=
="o") || (char=="u") )
{
  console.log("is a vowel");
}
else{
  console.log("not a vowel");
}
```

```
// Switch Case
var char = "i";

switch(char)
{
    case "a":
    case "e":
    case "i":
    case "o":
```

```
    case "u":console.log("vowel");
              break;
    default : console.log("Not a vowel");
}
```

```
var day = "Fri";


switch(day){
    case "Mon" :
        console.log("Monday");
        break;
        case "Tue" :
            console.log("Tuesday");
            break;
            case "Wed" :
                console.log("Wednesday");
                break;
                case "Thus" :
                    console.log("Thrusday");
                    break;
                    case "Fri" :
                        console.log("Friday");
                        break;
                        case "Sat" :
                            console.log("Saturday");
                            break;

            default :
            console.log("Sunday")
}
```