Javerscripts

Roojeja

PROGRMMING:-

programming refers to a technological process to for telling a computer which tasks to perform in order to solve problem. programming is a collabration blw human & computers in which human create to instructions to follow (code) in a language computer can understand and run.

Example:-

Programming language are javascript, C/C++, python, J.S, Node j.s Rect. etc.

Key Concepts:-

writing a series of step by step instructions that a computer can understand & execute solving problem & automating tasks through code.

Revision

Why we learn Programming?

Applicability in various domains such as web development, mobile app data Science etc.

PROGRAMMING:- A programming language is a system of notation for writing computer programs.

Programming languages are described in terms of their Syntax means form (error) usually define by formal language.

⇒ Computer can understand only binary language such 0 or 1

Binary Code Method:-

$5_{10} \rightarrow (101)_{10}$

2^7	2^3	2^2	2^1	2^0
128	8	4	2	1
		1	0	1

Rotary

Javascript?

=> javascript is the world's most popular language

=> javascript is the programming language of the web. and easy to learn

=> we insert the <script> in HTML body for javascripts.

Javascript has three kind of popup boxes the one we learn Alert and other are remaining "confirm" "prompt"

"ALERT"

An alert box is often used if you want to make sure information comes through the user

when an alert box pops up, the user will have to click "ok" to proceed.

Syntax :- (means the arrangement of words)

window.alert ("sometext"); ⇒ Go Global th.
we can also written only alert like this

alert("Hi"); Alert is a box that pop up to give the user a message.
alert is a keyword. In coding the quoted text "Hello world!"; is called text string or simply string

Q How to use in programe single quotation mark.

Royal 2018

2 variable for Strings

8:00 am Variables are containers for storing using var

8:30

9:00

In this first example x, y and z are undeclared variables.

9:30

They are automatically declared when first used

10:00

10:30

11:00

e.g:- var x = 5;
var y = 6;
var z = x + y;

11:30

Noon

12:30

Javascripts identifiers:-

1:00

=> All Javascripts variable must be identified with unique names.

1:30

2:00

=> These unique names are called identifiers

3:00

3:30

identifiers can be short names (like x and y) or more descriptive names (age, sum, total/volume).

4:00

4:30

5:00

5:30 pm The general rules for constructing

January : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

March : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

April : 18
M T W T F S S
30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

May : 18
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

June : 18
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

Roy

173/192 WEEK 25

Names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with small letter
- Names can also begin with $ and _ (but we will not use it in this)
- Names are case sensitive (y and Y are different variables).
- Reserved words (like javascript keywords) cannot be used as names. like window

the Assignment Operator

In javascript the equal sign (=) is an assignment operator not equal sign.

Roopesh

3 number

javascript variables can hold number like 100 and text values like "John Doe".

in programming, text values are called String.

javascript can handle many datatype but now just think about number & Strings

<u>String</u> are written inside double quotes

<u>numbers</u> are written without quotes.

If you put number inside quotes it would be consider or treated like text string.

e.g:-

```
Var x = 15
console.log(x + 5)
```

Review

Chp 5

"Math expressions"

Familiar operators:-

Javascript has many familiar operators from basics math, as well as a few additional operators specific to programming.

Here is a reference table of javascript arithmetic operators.

Operator	Syntax	Example	Def
addition	+	x+y	Sum of x & y
Subtraction	−	x−y	difference of x & y.
Multiplication	*	x*y	Product of x & y
Division	/	x/y	Quotient of x & y
Modulus	%	x%y	Remainder of x/y

Ro Vaja 2018

8:00 am — increment ++ x++ x plus one

8:30

9:00 — Decrement -- x-- x minus one

9:30

10:00

10:30 — Here are some example of add & sub

11:00 ⇒ addition { var x = 1 + "1";
Console.log (x);
output ⇒ 11 → This answer is string

11:30 — // Assign values to x & y

Noon — var x = 10;

12:30 — var y = 20;

1:00 — var z = x + y;

1:30 — Console.log ({z}) ⇒ output {z=30};

2:00

2:30 — **Sub:-**

3:00 — // Assign values to x & y

3:30 — var x = 10;
var y = 15;

4:00 — var z = x-y;

4:30 — Console.log (z)

5:00 — output 5

5:30 pm

The thing that is really hard, and really amazing, is giving up on being perfect and beginning the work of becoming yourself. – Anna Quindlen

+ - have same priority so if one come first like 2+3-4 so we right the left side first

* & / have also same priority as compared to + - so the ~~vat~~ priority of * & / have high.

=> + operator have two operator one it add the value and second change the value in string like

var x = 1 + "1"; 1 1

var x = 1 + "1" + 1; => 1 1 1

var x = 3 + 2 + "5"; 5 5

add apply operand concantenation apply

Subtraction :-

var = 2 - "2";

Subtraction automatically convert the string into value

Roopya

var x = "b" - 3;
⇓
output Nan => not a number.

Multiplication :-

var x = 2 * 5; output 10
console.log (x);

Multiplication can also convert the string into numbers.

e.g:-

var firstNumber = 2;
var secondNumber = 10;
var result = firstNumber * secondNumber;
~~var~~ console.log (result);

we can also * in string value or number.

e.g

var result = "4" * "2";
console.log(result);

Revision

```
var result = 5 + "V";
console.log (result);
```

output : Nan aye ga

Division /

returns quotient

```
var dividend = 17;
var divisior = 4;
var result = dividend / divisor;
console.log (result);
```

Modulus operator %

return remainder

e.g:-

```
var x = 15;
var y = 2;
var result = x % y;
console.log (result);
```

printing better messages

```
var firstnumber = 4;
var secondnumber = 5;
var sum = firstnumber + secondnumber;
console.log (`The sum of ${firstnumber} e
```

we used back tip

8:00 am $ {secondnumber} is $ {sum}!');

8:30

9:00 CHP: 6 unfamiliar

9:30 operators

10:00

=> increment ++

10:30 var salary = 150;

11:00 console.log(++salary);

11:30

Noon pre-increment

12:30 var salary = 159;

1:00 var incrementedSalary = ++salary;

1:30 Console.log({incremented salary}); // 160

2:00 console.log({salary})

2:30

3:00 => Post-increment

3:30 var salary = 159;

4:00 var incremented salary = salary++;

4:30 Console.log({incrementedsalary}); // 159

5:00 Console.log (salary); // 160

5:30 pm

January : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

March : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

April : 18
M T W T F S S
30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

May : 18
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

June : 18
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

Roger

decrement

var Charging = 65;

console.log (Charging = -);

CHP: 7 Math expressions

eliminating ambiguity in math expressions

<u>Bodmas</u> -> "Brackets, order, /, *, +, & - in js is called operator precedence (operator priority which is a bit change from bodmas) like + and -, and * , / have same precedence

<u>examples</u>:-

console.log (2 - 1 * 8 ~~14~~);

2 - 1 * 8

2 - 8

-6

console.log (2 * 4 * 4 + 2);

8 * 4 + 2

32 + 2

34

Roopesh

1 Sunday

console.log (5*5 - 3/3 + 7);

8:00 am // 25 - 3/3 + 7

8:30 // 25 - 1 + 7

9:00 // 24 + 7

9:30 // 31.

10:00

10:30

11:00

11:30

Noon

12:30

1:00

1:30

2:00

2:30

3:00

3:30

4:00

4:30

5:00

5:30 pm

Revise

Chp: 8

Concatenating Strings
(combining strings)

in chp: 1 we have learned to display a message in an alert, like this

=> alert ("Thanks for your input!");

or we can code ~~like~~ also this way

=> const message = "Thanks for your input!";
alert (message);

=> personalized message

```
const userName = "Ali";
alert("Thanks, " + userName + "!");
```

we can concatenate any combination of strings and variables, or all strings or all variables.

for example:-

```
const message = ~~"Thanks"~~; "thank, ";
const username = "Ali";
const punctuation = "!";
```


July : 18 | August : 18 | September : 18 | October : 18 | November : 18 | December : 18

Rovery

alert (message + username + punctuation);

Here it is, with three strings:
alert ("Thanks, " + "Ali" + "!");

CHP: 9 "Prompt"

we used prompt to take input from user like

```
const userage = prompt("please enter your age");
console.log("your are " + userage + " years old.");
alert("you are " + userage + " years old.");

const userName = prompt("Please enter your name");
alert("Hellooo " + userName);
Console.log("hello " + userName);

// prompts always return string values.
const pencil = +prompt("Price for pencil");
const noteBook = +prompt("price for notebook");
```

Roqeq

```
const totalBill = pencil + noteBook;
alert ("your total bill is" + totalBill);
```

CHP: 10 "If Statement"

The If Statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.
"If Statement" is a part of javascript "Conditional" Statement, which are used to perform different action based on different conditions

Simple example:-

driving license
The condition returns a boolean

```
let userage = +prompt("Please enter your age");
if (userage >= 18) {
```

Roger

```
Console.log("you are not eligible to get your
            license");
} else {
Console.log(" you are not eligible to get your license");

Brt card program

let Bartcardbalance = +prompt ("Enter your balance");
if (bartcardbalance >= 55) {
  alert("welcome to ZU Peshawar"); }
if (bartcardbalance < 55) {
  alert("Recharge your balance"); }
```

Chapter # 11

"Comparison Operators"

equal ===, ==

=== Three equal operator check that datatype and value both are same.

double == can check only the value doesn't matter what datatype is it

Example:- Double equall example

console.log (1 == 1); // true

console.log (1 == "1"); // true

double equal didn't think about what datatype is it

three equal operator ===

console.log (1 === 1); // true

console.log (1 === "1"); // false

not (!);

! work on boolen operator

!false is true and !true is false

not equal (!==, !=)

(!=)

```
Console.log (1 != 1); // false
```

Single not equal (!=) is not check the datatype.

```
Console.log ( 1 != 2); // true
```

(!==)

double not equal (!==) is check the datatype and value both

```
Console.log (1 !== 1); //false
Console.log (1 !== "1"); //true
```

Greater than >

```
Console.log (1 > 2); // false
Console.log (2 > 1); // true.
Console.log (1 > "1"); // false.
Console.log (0 > -1); // false.
```

less than <

```
Console.log ( 1 < 2); // true
Console.log (2 < 1); // false
```

Rayan

8:00 am console.log (1 < "1"); // false
console.log (1 < 0); // true.

8:30

9:00 less than or equal to <=

9:30 console.log (1 <= 1); // true

10:00 console.log (1 <= 0); // false.

10:30

Greater than or equal to >=

11:00 console.log (1 >= 1); // true

11:30 console.log (15 >= 15); // true

Noon console.log (2 >= 3); // false

12:30

1:00

1:30

2:00

2:30

3:00

3:30

4:00

4:30

5:00

5:30 pm

Rovers 2018

Chapter: 12

"If Else Statement"

Example:

```
// bored hai to barish, nahi to nahi
program.

// var bored = true;
if (bored){
    alert("It's gonna be raining soon in
        Peshawar!"); }
if(!bored) {
    alert("No chance of rain"); }

Using if-else.
if (bored) {
    alert ("Rain expected");
} else { alert("No chance of rain!");
}
```

Roopa

Real world example:-

negative value or variable

~~var~~ negative value allowed

```
var boolal = !false
console.log (boolal)
```

~~tru~~ even / odd value.

```
var num = +prompt ("enter a number")
if (num % 2 ===) {
    alert ("even number"); }
else { alert ("negative number");
```

Chapter # 13

Testing Sets of conditions

In javascript we have the following conditional statements.

- Use ~~if~~ to specify a block of code to be executed, if a specified condition is true.
- use else to specify a block of code to be executed, if the same condition is false.
- use else if to specify a ~~block of code~~ a new condition to test, if the first condition is false.

use switch to specify many alternative blocks of code to be executed.

ELSE IF Statement

```
if (condition 1){
  // block of code to be executed if condition is true.
```

```
else if (condition 2){
    // block of code to be executed if the
    condition 2 is ~~false~~ true
}
else {
    //block of code to be executed if the
    condition 1 is false and condition 2 is
    false
}
```

Example:- If time is less than 10:00, create a "Good morning" greeting if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening"

Soln

```
if (time < 10){
    greeting = "Good morning";
}
else if (time < 20){
    greeting = "Good day";
}
else {
    greeting = "Good evening";
}
```

Rovey 2018

CHAPTER : 14

IF STATEMENTS NESTED

```
// What is a nested if
// if (10 > 8) {
// if if (8 > 5) {
        console.log ("Yes 8 > 5");
    }
}
User Authentication
Const username = prompt ("Please enter your username");
let password = "";

if (username === "admin") {
  (password = ~~prompt~~ === "secret") {
  document.write ("login successful. Welcome, Admin!");
} else {
  document.write ("Incorrect password. Please try again.");
}
} else {
  document.write("User not found. Please check your username.");
}
```

Rover

8:00 am Ticket Pricing

8:30 Const age = + prompt("Please enter your age");

9:00 Const in Student = + prompt(Are you a student\no. No\n1. Yes'y);

9:30

```
if (age < 12) {
```

10:00

```
    document.write("child Tricket : $5");
```

10:30

```
} else {
```

11:00

```
    if (age >= 12 && age < 18 && isStudent) {
```

11:30

```
    document.write("student Tricket : $10");
```

Noon

```
    } else {
```

12:30

```
    document.write("Adult Ticket : $15");
```

1:00

```
    }
```

1:30

```
}
```

2:00

SARKARI NOKARI.

2:30

REQUIREMENTS

3:00

AGE: 18 - 25

3:30

GENDER: Female

4:00

```
var age = 19
```

4:30

```
var gender = 'female'
```

5:00

```
if (age >= 18 && age <= 25) {
```

5:30 pm

```
    if (gender === 'male') {
```

```
console.log('Eligible')
}
} else {
console.log('Not Eligible')
}
```

Revert

CHAPTER : 15

ARRAYS

Problem :

for multiple values we should have to create multiple variables / containers

```
var fruit0 = 'mango'
var fruit1 = 'banana'
var fruit2 = 'strawberry'
```

Solution:

to solve the above problem, we introduced the concept of arrays

so, array is a container that can store multiple values

create array

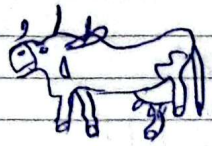
```
var fruits = [];
console.log(fruits);
```

Using emojis

var fruits = [" ", " ", " ", " ", " ", " "];

var animals = [" ", " ", " ", " ",];

Rogeyu2018

8:00 am console.log (Fruits);
8:30 console.log (animals);
9:00 document.write (animals);
9:30
10:00 putting elements in an array
var fruits = ["Banana", "Mango", "Apple", Orange"];
10:30
11:00 accessing array elements, whole array
11:30 console.log (fruits);
Noon
12:30 accessing specific array elements
1:00 concept of indices (singular: Index)
1:30 var hotelRoom = ["Ali", "Bilal", "choudary", Danyal"];
2:00 console.log (hotelRooms [4]);
2:30
3:00 var animals = [" ", " ", "  "];
console.log (animals [0]);
3:30
4:00 mix different types
4:30 heterogeneous data vs homogenous data
5:00 homogenous: same type
5:30 pm heterogenous: different types

January : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

March : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

April : 18
M T W T F S S
30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

May : 18
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

June : 18
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

8:00 am
```
var mixed Data = [1 , "hello" , true , 3.5 ,"BYE"];
```
8:30
```
console.log(mixedData);
```
9:00
```
console.log (mixed Data);
```
9:30
```
console.log (mixedData [1]);
```
10:00

arrays in js are dynamic.

10:30

11:00

11:30

Noon

12:30

1:00

1:30

2:00

2:30

3:00

3:30

4:00

4:30

5:00

5:30 pm

Teamwork is the secret that makes common people achieve uncommon results. – Ifeanyi Onuoha

CHAPTER : 16

ADDING AND REMOVING ELEMENTS:

var books = ["computer science", " mathematics"];
console.log (books);
add element to the array

single
books.push ("Physics");
console.log(books);

Multiple
book.push ("Physics", "chemistry");
console.log (books);

add element at a specific position /index, it will replace already existing element
var books = ["computer science", mathematics"];
books [1] = "Physics";
books [2] = "Physics";
console.log (books);

8:00 am adding elements to empty array

8:30 var books = [];

9:00

9:30 books [0] = "computer";

10:00 books [1] = " Urdu";

10:30 books [2] = "english";

11:00 books [10] = "maths";

11:30 console.log (books);

Noon

12:30 remove elements

1:00 var books = ["Urdu", "english", "Math"];

1:30 books.pop();

2:00 var poppodBook = books.pop();

2:30 console.log (books);

3:00 Console.log ({ poppedBook });

3:30

4:00

4:30

5:00

5:30 pm

Chapter : 17

Arrays, Removing, inserting, and extracting elements.

```
var numbers = ["1", "2", "3", "4"];
numbers.push("6");
let popped = numbers.popp();
numbers.push("5");
popped = numbers.pop();
console.log(number)
console.log(popped)
```

"Shift and unshift"

The unshift() method adds new elements to the begining of an array.

e.g

```
var fruits = ["Apple", "banana", "Orange"];
fruits.unshift("lemon", "pineapple");
```

Splice :- The method is used to insert, replace or remove elements in an array.

alphabets.splice (0, 0, " ")

where to add how many dtl

Adding elements

alphabets = ['a', 'b', 'c', 'd', 'e'];

↓ splice (1, 0, 'B');

⇒ Slice :- In slice method specific part of the array are copied through indexing we can put that part in another array.

example:-

var alpabets = ["A", 'B', "C", "D", "E"]

var sliced = alpabets.slice (0, 3);

var spliced = alpabets.spliced (0, 3);

[A, B, C]

[D, E]

Ropery

Boolean values:-

very often in programming we will ~~used~~ need a data type that can only have one of two values, like

Yes / No
on / off
True / false

The Boolean() function:-

You can use the Boolean() function to find out if an expression (or variable) is true.

Example:-

The boolean value of false is (you guessed it) false:-

```
let x = false;
Boolean(x);
```

The boolean value of NAN is false:-

```
let x = 10 / "Hello";
Boolean (x);
```

Flag:-

flag variable is used as a signal in programming to var the program know that a certain condition has met. It usually acts as a boolean variable indicating a condition to be either true or false.

Array length:-

Number of elements in an array

Number.length

loopus interrupter:-

To break a loop before it's comparison, meeting a condition

implementation:-

var decmescities = ["Peshawar", "Islamabad", "Karachi", "Lahore"]

July : 18
M T W T F S S
30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

August : 18
M T W T F S S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

September : 18
M T W T F S S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

October : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

November : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

December : 18
M T W T F S S
31 1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Rayaja

```
var cityToCheck = prompt("Plz enter your
    city name:");

Flag:-
    var cityMili = false;

for(var cityIndex = 0; cityIndex < cleanestCities.
length; cityIndex ++) {
    if (cityToCheck === cleanestCities[cityIndex]){
        cityMili = true;

Loops intrrrupts
    break;
alert("its Yes I fined the city");
}
    else {
        alert("I cannot fined city");  }
alert (cityIndex);
}
    if (cityMili){
    alert("It's one of the cleanest cities");
} else {
alert("It's not on the list");
```

Roger

"For Loops Nested"

Nested Loop -

```
var count = 0;

for (var j = 1; j <= 3; j++) {
    console.log (i: ${i}, j: ${j}');
    Count++;
} }
console.log ({ count});
```

example - Recycling characters

```
var characters = "ABC";
for (var i = 0; i < 3; i++) {
    for (var j = 0; j < characters.length; j++) {
        console.log (characters [j]);
} }
```

Changing case

Lowercase:-

hello world

```
toLowerCase();
var city = "PESHAWAR";
var smallcity = city.toLowerCase();
console.table ([city, smallcity]);
```

Rogers

8:00 am ToUpperCase()

8:30 ToUpperCase is used when all the

9:00 letters is small

9:30 var city = "karachi";

10:00

without using second variable

10:30 console.log (city.toUpperCase());

11:00 city = city.toUpperCase();

11:30

Noon using second variable

12:30 var capitalizedCity = city.toUpperCase()

1:00 console.log (city);

1:30 console.log (capitalizedCity);

2:00

2:30 Strings: Measuring length and

3:00 extracting parts

3:30

Finding length of String:-

4:00 console.log ("first".length);

4:30 var ChannelName = "CS with alam";

5:00 var channelNamelength = ChannelName.length

5:30 pm console.Log (ChannelNamelength);

Rovoya

Extract Part of a String

```
var number = [1, 2, 3, 4, 5];
var arraypart = numbers.slice(0, 2);
console.log(arraypart);
```

Slice with strings

```
var language = "javascript";
using another variable
var sliced = language.slice(0, 4));
console.log(language.slice(1, 5));
```

Finding a character at a location

```
// returns the character on the given address
var institute = "Sayfani mass it Training";
console.log(institute.charAt(4));
```

Email validation using Chart

```
var userEmail = prompt('Pls enter your email address');
var isValid = false;

for (var charIndex = 0; charIndex < useEmail.length;
charIndex ++) {
```

Review

```
if (userEmail.charAt (charIndex) === "@") {
    isValid = true;
    break;
}
}
if (isValid) {
    alert("Valid email address");
} else {
    alert ("Invalid email address");
}
```

Rounding Numbers

methods we're gonna talk about

=> round() -> it rounds a number to the nearest integer.

=> Ceil() -> it rounds a number UP to the nearest integer.

=> floor() -> it round a number DOWN to the nearest integer.

Rover

Generating random numbers

8:00 am

// Method we're discussing in this chapter

8:30

// random · Math.random ()

9:00

1. returns a random number between 0 (inclusive), & 1 (exclusive)

9:30

2. always returns a number less than 1.

10:00

3. always delivers a 16-places decimal that ranges from

10:30

0.0000000000000000 to 0.9999999999999999

11:00

11:30 Generating a random number

Noon

```
var randomNumber = Math.random ();
```

12:30

```
Console.log (randomNumber);
```

1:00

// Converting a random number to an integer

1:30

There is no such thing as JavaScript integers.

2:00

We are talking about numbers with no decimals here.

2:30

3:00 Getting just the integer part, removing the decimal part

3:30 (number after decimal point).

4:00

```
Var randomNumber = Math.random ();
```

4:30

```
Var integarNumber = Math.floor (random Number);
```

5:00

```
Console.log (integer number);
```

5:30 pm

If .. else & else-if

if-else:- To test 2 conditions if-else are used.
first part if the condition is "true" code will be executed
if not the 2nd will be.

Example:-

```
var badal = True;
if (badal = = = True) {
    console.log ("Train expected");
} else { console.log ("not"); }
```

It will always give true because we assign true to train.

⇒ Else if:- To check more than 2 conditions if the above statements fails then the last else part will be executed.

e.g

```
var marks :- ("Enter your marks");
if (mark >=90) {
    console.log ("A + Grade") }
else if ( mark < 90 && mark >= 80)
{ console.log (B Grade)
```

Review

```
else if (marks < 50 && marks > 40)
    { console.log ('C Grade') }
else { console.log ('fail');}
```

Testing sets of Conditions

To Test multiple condition we use "and" or "or" condition.

1- && (AND) operator:-

The logical operator are used to check multiple condition. it will be true only if all the statements are true.

2- || (OR) operator:- It is also used to check multiple conditions. if one statement is true or other is also it will give true result, it will give false only when both

Rover

are false

if Statement Nested

when one of if statement is used inside another if statement, it is said nested if - statement.

example:-

```
let name = ("Enter name");
let password = " ";
if (name === "admin"); {
    if (password === "secret") {
        console.log("Access it"); }
}
```

Report

while loop

8:00 am difference b/w for loop and while loops

8:30

For loop:- A for loop is an iteration method that is best used when you know the number of iteration ahead of time. its always followed by the initialization, expression and statements

9:00

9:30

10:00

10:30

11:00

11:30 While loop:- A while loop is an iteration method that is best used when you don't know the number of iteration ahead of time. The contents of the loop are executed as longas the expression evaluates to true.

Noon

12:30

1:00

1:30

2:00

2:30

3:00

3:30

```
var i = 0;
while (i < animals.length){
    if (animals[i] === "pig") {
        alert("Found it!");
        break;
    }
    i++;
}
```

4:00

4:30

5:00

5:30 pm

January : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

March : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

April : 18
M T W T F S S
30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

May : 18
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

June : 18
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

Rovers

Converting Strings to integer or decimal:-

Type of converting are of two type.

1- Coersion:- It is an implicit (automatic) type conversion

e.g:- "25" - 10; → convert to number automatically.

=> "25" + 10 → convert to string automatically

2- Casting:- It is an done with some methode / function

=> Converting to Number:- e.g:-

```
var number = prompt("Enter a num")
var Tonum = parseInt(number);
```

Converting to float:-

```
var tax = prompt("tax")
var toTax = parseFloat(tax);
```

Rovezon

Converting Strings to Number eg

Num to Strings:-

1- Number(); function is used to convert any string either int or float to a number:-

example:-

```
var mynumb = prompt("Enter num")
var convert = Number(mynumber);
```

we can covert number, either float or int;

2- .toString():- .toString is used to convert number to string;

example:-

```
var num = 1000;
var convert = number.toString();
```

Rovya

11

JavaScript for Loop:

The for statement creates a loop with 3 optional expressions:

```
for (expression 1; expression 2; expression3;) {
    // code block to be executed
}
```

Expression 1 is executed (one time) before the execution of the code block.

Expression: 2 defines the condition for execution the code block.

Expression 3 is executing (every time) after the code block has been executed.

Example:

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

For the example above, you can read Expression 1 sets a variable before the loop starts (let i=0).

Roqeya

Expression: 2 defines the condition for the loop to run (i must be less than 5).

Expression 3 increases a value (i++) each time the code block in the loop has been executed.

Expression 1:

Normally you will use expression 1 to "utilize" initialize the variable used in the loop (let i=0). This is not always the case. Javascript does not care. Expression 1 is optional. You will initiate many values in expression 1 (separated by comma).

Example:

```
for (let i=0, len = cars.length; text = "";
  i < len; i++) {
  text += cars[i] + "<br>";
}
```

You can omit expression 1 (like when your values are set before the loop starts):

Roveja

Expression 2.

Often expression 2 is used to evaluate the condition of the initial variable. This is not always the case. JavaScript doesnot care. Expression is also optional. If expression 2 return true, the loop will start over again. If it returns false, the loop will end.

Expression 3:

Often expression 3 increments the value of the initial variable.
This is not always the case. JavaScript doesnot care. Expression 3 is optional. Expression 3 can do anything like negative increment (i--), positive increment (i = i + 15), or anything else.
Expression 3 can also be omitted (like when you increment your values inside the loop):

Example:

```
let i = 0;
let len = cars.length;
let text = "";
```

Roy

```
for (; i < len ;) {
    text += cars [i] + "<br>";
  i ++ ;
}
```

While Loops:

The while loop and the do /while are explained in the next chapters.
The while loop loops through a block of cods as long as a specified condition is true.

Syntax:

```
While (condition) {
  // code block to be executed.
```

Example:

```
While (i < 10) {
    text += "The number is" + i;
  i ++ ;
}
```

The Do While Loop: The do while loop is a variant of the while loop. This loop will execute the code block once, before checking if

Roveja

the condition is true, then it will repeat the loop as long as the condition is true.

Syntax:

```
do {
  // code block to be executed
}
While (condition);
```

Example:

```
do {
    text += "The number is " + i;
  i++;
}
While (i < 10);
```

Roqeya

Chapter: 23

Strings; Finding Segments

```
// methods we're gonna discuss
// index of
// var book = "Think and Grow Rich";
// var indexofk = book.indexof("k");
// console.log(indexofk);

// Always returns the index of very first match
// var greetings = "hello";
// console.log(greetings.indexof("1"));

// Can pass string as well
// var book = "Think and Grow Rich";
// console.log(book.indexof("and Grow"));

// APKO -1 DETA HA AGAR MATCH NA MILE
// var greetings = "hello";
// console.log(greetings.indexof("r"));

// another example of -1 /not found
// var email = "test@test.com";
// console.log(email.indexof("r"));
```

Roy"

// console.log(email.indexof("@gmail.com"));

// We can use simple arrays as well
// var mohalla = ["ali", "ahmad", akram"];
// console.log(mohalla.indexof("aslam"));

// lastIndexof

// as the name suggests, it always returns the index of last match.
// var greetings = "hello";
// console.log(greetings.lastindexof("1"));

// var mohalla = ["ali", "aslam", "ajmal"];
// console.log(mohalla.lastIndexof("1"));

// Email variation without using index (must include @)
// var userEmail = prompt("please enter your email address");
// var isvalid = false;

// for (var charIndex = 0; charIndex < userEmail.length; charIndex++) {
// if (userEmail[charIndex] === "@") {
// isvalid = true;
// break;
// }

Roqeya

```
if (isvalid) {
    alert("valid email address");
} else {
    alert("Invalid email address");
}
```

Rovar

Controlling length of decimal

- toFixed() :- is a method that fixed decimal part of a number.
- it can argument.
- if we didn't pass an argument, then it remove all decimals & return only integer.
- It will round up the data automatically
- Return the data in the form of a String

example:-

```
var num = Math.random() * 0;
var fixing = num.toFixed(2);
var convert = num(fixing);
```

Getting the current date & time :-

Current date & time :-

Date() object is used to get the

The current date & time, its will return the date.

Example:-

```
var mytime = new Date();
```

→ new will create the new copy of that class or instance.

in millisec we use:-

```
var millisec = mytime.getTime();
```

Getting Year:-

Example:-

```
var year = mytime.getfullyear();
```

Date:-

example:-

```
var date = mytime.getDate();
```

Month example:-

```
var month = mytime.getMonth();
```

→ It will start from 0, January is at 0 index

Day:-

```
var Day = mytime.getDay();
```

Function :-

Function is a block of code that the same work again & again, whenever we call it
It save us from repititive coding.

Syntax :-

Function => its a keyword
( ) { body }
name of function

Noon → calling the function :-
we call the function in least if we don't call it, it will never executed.

example :-

function myName () { body }

calling
myName ();

=> function : Passing them Data :-
we can pass data in the form of parameter & argument

Revision

8:00 am Example:-

```
function fun (Greet) {
    alert (Greet); }
fun ("Hello");
```

8:30

9:00

9:30

10:00

10:30 Passing data back from them

11:00 Return:- is used to pass the data

11:30 back from the function.

Noon

12:30 Example:-

1:00

```
function numbers(num1, num2) {
    clg (num1 + num2);
        return [(num1 + num2]; }
fun (1, 2);
```

1:30

2:00

2:30

3:00

3:30

4:00

4:30

5:00

6:00 pm

Roraya

"Switch Statement"

8:00 am

Default

8:30

The key word default work as else as part of else-if.

9:00

9:30

if the case are not meet

10:00

then default will be executed.

10:30

11:00

11:30

Noon

example:-

12:30

```
var color = "black";
switch (color) {
case: 'blue';
break;
case: 'Green';
break;
default: cly ("Not found");
}
```

1:00

1:30

2:00

2:30

3:00

3:30

4:00

Placing scripts :-

4:30

In HTML code, JS code is inserted in b/w the opening & closing script tag.

5:00

5:30 pm

Roveya

8:00 am JS in Head:-

8:30 In HTML doc we used JS

9:00 code in the head section.

9:30 b/w opening & closing script

10:00 <script> " " </script>

10:30

11:00

11:30 Event and Links:-

Noon

=> Events are the user actions click

12:30 button, link, or mouse.

1:00

1:30 Inline Event Handling:-

2:00 Means combining JS with HTML

2:30 like:-

3:00 <a href="#" onclick="alert(Hi)">click

</a>

3:30

=>

4:00 calling function on link click:-

4:30 Inline event is not a good way,

5:00 the more efficient way is to write

5:30 pm

January : 18
M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

February : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

March : 18
M T W T F S S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

April : 18
M T W T F S S
30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

May : 18
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

June : 18
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

Royal Queensland Show (Brisbane metro area)

227-138 WEEK 33 the JS code is an function or the call that function in the hreftag.

Buttons :-

we can use events with buttons also, with the input yield button or with the button tag also.

Example :-

```
<input type = "button" value = click
onclick = "Alert (Hi)" >
```

List of Mouse events :-

1- onclick : A user click on element
2- onrightmenu : user right click on element
3- onclick : user double click on element.
4- onmousedown : A mouse button is pressed over an element.
5- onmouseenter : A mouse pointer moves into an element.
6- onmouse leave : Mouse pointer leave an element.

Rogeyi

Events : fields:

8:00 am
8:30
9:00
9:30
10:00
10:30
11:00
11:30
Noon
12:30
1:00
1:30
2:00
2:30
3:00
3:30
4:00
:30
:00
0 pm

onfocus:- perform a function when the user click in the field.

Example:- <input type="email" onfocus= fun( )>

onBlur:-
Perform an operation when the user clicks outside the feild or focus.

Example:-
<input type="email" onBlur=fun() >

Reading feild values:-

• value:-
•value is used to get or recad feild value;

Examples:-

```
function Address () {
var getEm = doc.getEleById ('email')
var getval = getEm.value
```

About the only time losing is more fun than winning is when you're fighting temptation. – Tom Wilson

Setting feild values.

The same methods (.value) is also used here.

Example:-

```
<form onsubmit = "func()"> </form>.

function fun() { let a = doc.getElmById("id")
    a.value = Rogeyer.gmail.com }
```

• InnerHTML :-

• innerHTML is used to get the internal HTML of any HTML tag

Example:-

```
var num = doc.getElmById("Hello").innerHTML
    = "Bye";
```

Manipulating Img & Text

In this chpter we will discussing 3 main thing,

=> Toggle img
=> Hide img
=> Show img

8:00 am Example (Toggle):-

8:30

```
<button id="Toggle" onclick="Toggleimage()">
Toggle </button>
```

9:00

9:30

10:00

```
<img src="1.jpg id="myImge">
var img = doc.get("myImage");
if (img.src.includes('1.jpg')){
    img.src = 2.jpg; }
else {
    img.src = 1.jpg;
}
```

10:30

11:00

11:30

Noon

12:30

1:00

1:30 Hide:-

```
<button id="~~show~~ hide" onclick="showImg();
function showImg(){ hidden
    var img = doc.get('img Image')
    img.style.display = none;
}
```

2:00

2:30

3:00

3:30

4:00

4:30

5:00 Show:

5:30 pm but show is also same to hide just change in place of hide write show.

"Swapping image and Setting classes"

- ClassName :-
  - className is used to get or set the values of a class attributes.

- classlist :-

1. add ()
2. Remove ()
3. Toggle ()
4. Contains ()
5. Replace ()

Example :-

HTML :-

```
<image src='1.jpg' onclick=swap(1.jpg, this)
class='active')
```

JS :-

```
function swap(newimage, click) {
    var mainImg = doc.get('main');
    mainImg = newImg;
    var images = doc.get('image');
for (var i=0; i< images.length; i++); {
    image[i].classlist.remove('active'); }
```

Revise

click. classlist. add('active'); }

Target All elements By Tag Name":-

In previous chpters we learned how to how target element by id or class name used in this chpter we learned by Tag Name. like

document.getElementsBy ~~Et~~ TagName {

Example:-

```
<p> lorem1 </p>
<p> lorem2 </p>
<button onclick = lorem()> click </button>
function lorem() {
    var paragraphs = doc.getEmBy TagName('p');
}
```

The DOM (document object model):

It is a programming interface for web document, represented the page so that the program can change the str, style content.

Components:-

- Document (enter the page)
- Element Nodes (div, ht, P, etc)
- Text Nodes (Text inside else nodes)
- Attributes (properties (id, class, src).

It make the web page interactive and functional through dom we connect the website.

Dom (Parent & children)

Parent:- Any element that contains another element

e.g

```
<div>
  <P>  </P>
</div>
```

Revision

→ div is the parent of the p. and p is the child of the

Finding children :-

1- childNodes :-

• childNodes will return a Nodelist of all child nodes including element node, text node, comment nodes

e.g :-

```
var div = doc.getElById ("parentDiv")
var nodes = div.childNodes;
div.childNode[1];
```

2- Children :-

children will return nodelist of only element nodes.

e.g :- div.children[2];

Junk Artifact ey Node type

1- Junk Artifacts:- All the nodes except element nodes are junk artifacts
Text and ~~document~~ comment Nodes are junk artifacts

Nodes Types:-

The node type property returns the node type number as

1- Element Node -> 1
2- Attribute Node -> 2
3- Text Node -> 3
4- Comment Node - 4

First child and lastchild

Through first child we target the first element of the div and lastchild we target the lastchild of the div

e.g:-

```
var a = doc.getElementById('main');
a.firstchild; -> returns
```

a. lastchild → return 'B'

3- nextSibling :-

It will return the next sibling node of the current node.

list2. nextSibling;

4- Previous Sibling :- It will return the previous sibling of the current node

list2. previousSibling;

Getting ur targets Name :-

⇒ • nodeName :-

• nodeName is used to target and get the name of the node.

e.g :-

```
<div id = "div1" >
    <p> 1 </p>
    <p> 2 </p>  </div>
var div = doc.getElBy Id ("div1")
var a = doc. firstchild;
a. Target . node Name;
```

• NodeName :-

a. target.nodevalue;

DOM: Attributes :-

In this chpter we will discuss Dom and attributes like id, class, src, etc.

e.g :-

```
<img src = "1.jpg"  id = "img" >
<input id = 'src'  placeholder = "enter src">
<input id = 'alt' >
<input id = "title" >
function change () {
    var img = doc.getElById ('img');
    var src = do.get ('src') .values;
    var alt = da.get ('alt').value;
    var title = doc.get ('title').value.;
if (new src) {
        img.Src = newsrc;  }
if (alt){ img.alt = alt;
    } };
```

Roqeya

Adding Nodes:-

1. Create Element :-

Create element is used to create an HTML element dynamically using JS.

e.g:-
```
var h = doc.~~get~~ createElement ("h1");
```

Here h1 is create, now we can set text or apply styling dynamically

```
h.textContent = "My first h1";
```

2- Create Text Node:-

We can simply create a text node, without a parent & later we join it with a parent.

e.g:-
```
var text = document.CreateTextNode();
```

3- • appendchild();

• appendchild() is used to

Roqeyer

add one node /element with another or with body.

e.g:-

lent . appendchild (h);

Inserting Node:

⇒ Inserting Nodes at specific position:-

To insert an element at specific position, insert before is used. it allow us to insert a node before an existing node.

e.g:-

var firstchild = PDiv . firstchild;

pDiv . insertBefore (newpara, firstchild);

⇒ Inserting after:-

var target = parentDiv . childNodes[1];

parentDiv . insertafter (newpar . target . nextSibling);

Objects :-

Rogeyer

In JS object is a collection of related data or functionality
It have key value pair.
property : variable data inside obj
method : function inside an obj.

E.g :-

1- Creating :-

```
var obj = {
    name : "Rogeya";
    r.No : 29
};
```

This is how we create an obj.

2 - Accessing Obj properties :-

we can access data / properties of an obj using dot or bracket notation

→ dot Notation

```
obj.Name;
```

Rogeya

→ Bracket notation:

obj ["node"];

Adding new property:-

obj. class = "10th B";

or

obj["class"] = "10th B";

Objects: Properties:-

- properties:- Data related to the obj like name, rollno, class, ID etc.

Creating object:-

```
Var person = {
    nam: "Rogey"
Property {roll/no = "29"} ⇒ value
};
```

Roqeya

if we want to keep the property empty, just set it undefined

```
obj.rollNo = undefined;
```

- Deleting obj prop:-

we can delete obj prop using delete keyword.

```
delete obj.Name;
```

Methods :-

- Methods:-

Methods are functions attached to an obj.

it behave like a normal func but operation in the context of the object.

- Adding Method:-

```
var car = {
    color : 'Red'
    model : "Sonata"
```

```
start : function () {
    clg("start");
}
```

Calling:-

```
car.start();
```