

Day 3

Loops in JS

Loops are used to execute
a piece of code again
& again

i) for loop

^{initialize} ^{stopping condition}
for (let i = 1; i <= 5; i++) { ^{operation}

console.log("Apna college");

}

dry run

i = 1; 1 <= 5; 1++

i = 2; 2 <= 5; 2++

i = 3; 3 <= 5 → T

i = 4; 4 <= 5 → T

i = 5; 5 <= 5 → T

{

} we block of code.

let sum = 0;

let n = 100;

for (let i = 1; i <= n; i++)

{
sum = sum + i;
console.log(sum);

In let me @ 'i' is only
access in block of scope.

⇒ loops in JS

infinite loop: A loop that never
ends

⇒ while loop ^{initialize}

```
while (condition)
{
```

```
} updatation
```

let i=1; ⇒ initialize

while (i <= 10) ⇒ condition

```
{
```

```
  console.log("Talha Khan");
```

```
  i++; ⇒ updatation
```

```
}
```


⇒ Do-While At least execute 1-time.

In Do-while the first time execute the code

```
let i = 20;  
do {  
    console.log("UCP");  
    i++;  
} while (i <= 10);
```

⇒ For-of loop ii) for-in
⇒ to help iterate for special data.

For of loop ⇒ only for strings
arrays

for (let val of strVal) {
 }
}

string ka string characters
ko access kar sakte hain
Tallha
↑↑↑↑↑


```
let str = "myNameIsTalha";
```

```
for (let i of str) // iterator => characters.
```

```
{  
  console.log(i);  
}
```

string length:-

```
let size = 0;
```

```
for (let i of str)  
{  
  console.log("i", i);  
  size++;  
}
```

```
console.log("string size=", size); //16
```


⇒ for-in loops use for Object & arrays.

Syntax

```
for (let key in objVar)
{
```

```
}
let student = {
```

```
  Name : "Talha",
```

```
  age : "21"
```

```
  cpga : 4.45,
```

```
  isPass : "True",
```

```
};
```

```
for (let key in student)
```

```
{
```

```
  console.log(key);
```

```
  console.log("key", key, "value",
```

```
}
```

```
  student[key]);
```


Practice :-

Game Number = 25;

for (let i = 0; i ~~user~~ != correct
value

i++;
if (user input == correct value)

{ console.log("User win")

}

Strings in JS

String is a sequence of characters used to represent text.

creation of string

```
let str = "Talha Khan";  
let str2 = 'Talha Khan';
```

string length

`str.length`

⇒ Property of string length

⇒ Function to calculate the string in JS.

```
str[0];
```

```
str[1];
```

```
str[2];
```

```
str[3];
```


Template literal is a
special type string.
Backticks

```
let specialString = `This is a template  
literal`;  
console.log(typeof specialString);
```

for Example

```
let obj = {  
  item: "Pen",  
  price: 10,  
};
```

```
let output = `The cost of ${obj.item}  
is ${obj.price} rupees`;
```

```
console.log(output);
```

```
console.log("The cost of", obj.item, "is",  
obj.price, "rupees")
```


String interpolation

to create string by the
place holder

`{ (expression) ;`

Template literal

A way to have embedded
expression in strings

`'this a template literal'`

Escape characters-

count as a single character \Rightarrow `(\n)` \Rightarrow next line
console.log ("Talha \n Khan");
tab space `\t` for tab space

→ String Method in Js

There are built-in functions to manipulate a string

Methods or Functions:-

```
let str = "My home";
```

```
str.toUpperCase(); new string();  
                  ↑                  New value();
```

This method never change in the original string it will change in the new string and new value.

for interview

⇒ strings are Immutable in JavaScript never change

* str.toLowerCase();

* str.trim() // removes whitespaces.
remove spaces starting and ending

```
let str = "x Apna college is x";
```

```
console.log(str.trim());
```


str.slice (start, end?) // return part of string
2 5

print(c, d, e) ^{not include}

a b, c, d, e, f

console.log(str.slice(2, 5));
0 1 2 3 4 5
(3);

★ str1.concat (str2) Join the two strings

let str1 = "talha"

let str2 = "khan"

let res = str1.concat(str2);

Talha Khan

⇒ str.replace(searchVal, newVal)
replace the value with search value

let str = "hello";

console.log(str.replace("lo", "p"))
replaceAll "h", "y"

⇒) `str.charAt (index)`
return character.

`console.log (str.charAt(4));`
B
S

@ full name ending with null
length.

`str[0] == "a"`