⇒Day 6 :-

Loops :-

A loop is a programming constructs that repeats a block of code until a specified condition is met. We use loop to avoid writing repetitive code, make our programs more efficient and scalable.

Syntax: for loop:-

```
    for( initialization; condition; inc)
        {
          }

for (let i=0 ; i<3; i++ ) {
      console.log(i);
  }
```

Syntax of while loop:-

While loop:-

```
while(condition){

  }
```

example:-

```
let count=0;
 while (count<3){
   console.log(count);
  count++;
   }
```

output:

```
0
1
2
```

Syntax of Do-while :-

```
do {
     }.
       while(condition);
```

example;

```
  let count=0;
   do {
      console.log(count);
       count++;
    }
   while (count<3);

    output: 0
            ↑
            2
```

while vs. do-while:-

The main difference is when condition is checked. A do-while loop always executes its code at least once before checking the condition, ~~fast~~, A while loop checks the condition first, so it might not run at all if condition is initially false.

⇒ Day 7 :-

⇒ Define function and syntax:-

function is a block of reusable code that perform a specific task or calculates a value. Functions are fundamental building block of language.

Syntax :-

```
function functionName(parameter1, Par2
                      ....)
{
    return value;
}
```

=> Function Declaration and Function expression:-

Function declaration is standalone statement. Its hoisted, meaning the javascript engine moves it to top of its scope before code executes

```
function sayHello(){
    console.log('Hello');
}
```

→) Function Expression :-

Function expression is part of another expression, often an assignment its not hoisted in same way.

```
const sayGoodbye = function () {
    consale.log ('Good bye!');
};
```

Arrow function Syntax :-

```
const add = (a, b) => a+b;
```

for single line.

for multiline:

```
const processData = (data)=> {
let result= data.toUppercase();
console.log(result);
  return result;
 };
```

~~Arro~~

regular function:-

~~Regu~~

```
     const user = {
       name: 'Alice',
    greet: function() {
       console.log('Hello, my
    name is ${this.name}');
 }
```

Arrow function:-

```
const user={
  name : 'Bob',
say Hi : () => {
    console.log('HI, my name
  is ${this.name}');
 }
};
```

Regular function.

```
function add(a,b) {}
```

Defining reusability logic blocks, object methods, or constructo

Arrow function:

```
const add (a,b)=> { }
```

short, concise callbacks for method like map(), filter(), or reduce(), where a simple function is needed.