Conditions and Loop

Assignment Questions

Assignment Questions

Q1. What are conditional statements? Explain conditional statements with syntax and examples.

Ans1. Conditional statements in JavaScript are used to execute different blocks of code based on specified conditions. There are several types of conditional statements in JavaScript, including if statements, if-else statements, and switch statements.

1. **if statement**: It executes a block of code if a specified condition is true.

Syntax:

```
if (condition) {

    // code block to execute if condition is true
```

```
}
```

Example:

```
let x = 10;

if (x > 5) {

    console.log("x is greater than 5");

}
```

2. **if-else statement**: It executes one block of code if the condition is true and another block if the condition is false.

Syntax:

```
if (condition) {

    // code block to execute if condition is true

} else {

    // code block to execute if condition is false
```

```
}
```

Example:

```
let x = 3;

if (x > 5) {

    console.log("x is greater than 5");

} else {

    console.log("x is not greater than 5");

}
```

3. **if-else if-else statement**: It allows you to check multiple conditions and execute different blocks of code based on the first condition that is true. The else block is optional.

Syntax:

```
if (condition1) {

    // code block to execute if condition1 is true

} else if (condition2) {

    // code block to execute if condition2 is true
```

```javascript
} else {

    // code block to execute if all conditions are false

}
```

Example:

```javascript
let x = 10;

if (x > 10) {

    console.log("x is greater than 10");

} else if (x === 10) {

    console.log("x is equal to 10");

} else {

    console.log("x is less than 10");

}
```

These conditional statements in JavaScript allow you to control the flow of your code based on different conditions.

Q2. Write a program that grades students based on their marks.

. If greater than 90 then A Grade

. if between 70 and 90 then a B Grade

. if between 50 and 70 then a C Grade

.Below 50 then an F Grade

Ans 2. Answer code via vscode

 Q3. What are loops, and what do we need them? Explain different types of loops with their syntax and examples.

Ans3.Loops are control flow structures in programming languages that allow you to execute a block of code repeatedly based on a condition. They are essential for automating repetitive tasks and iterating over collections of data.

JavaScript provides several types of loops:

1. **for loop**: It iterates a block of code a specified number of times.

   Syntax:

```
for (initialization; condition; iteration) {

    // code block to be executed

}
```

Example:

```
for (let i = 0; i < 5; i++) {

    console.log(i);

}
```

2. **while loop**: It iterates a block of code as long as the specified condition is true.

Syntax:

```
while (condition) {

    // code block to be executed

}
```

Example:

```
let i = 0;

while (i < 5) {
```

```
    console.log(i);

    i++;

  }
```

3. **do-while loop**: It is similar to a while loop but guarantees that the code block is executed at least once before checking the condition.

Syntax:

```
  do {

    // code block to be executed

  } while (condition);
```

Example:

```
    let i = 0;

  do {

    console.log(i);

    i++;

  } while (i < 5);
```

4. **for...in loop**: It iterates over the properties of an object.

Syntax:

```
for (variable in object) {

    // code block to be executed

}
```

Example:

```
const person = {

    name: 'John',

    age: 30,

    gender: 'male'

};


for (let key in person) {

    console.log(key + ': ' + person[key]);

}
```

5. **for...of loop**: It iterates over iterable objects like arrays, strings, maps, sets, etc.

Syntax:

```
for (variable of iterable) {

    // code block to be executed

}
```

Example:

```
const colors = ['red', 'green', 'blue'];

for (let color of colors) {

    console.log(color);

}
```

Each type of loop has its own use cases and syntax, allowing you to effectively handle different looping scenarios in JavaScript.

Q4. Generate numbers between any 2 given numbers.
Ex
```
const num1 = 10
const num2 = 25;
```
Otpt: 11, 12, 13, …., 25

Ans4.   code via vscode

Q5. Use the while loop to print numbers from 1 to 25 in ascending and descending order.

Ans5.   via vscode