Control Flow

HyperionDev

Muhammad Zahir
Junejo

Lecture - Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- No question is daft or silly - **ask them!**
- Q&A session at the end of the lesson, should you wish to ask any follow-up questions.
- Should you have any questions after the lecture, please schedule a mentor session.
- For all non-academic questions, please submit a query: www.hyperiondev.com/support

Lecture Objectives

1. Controlling the flow of a program
2. Variable declaration
3. Setting up conditions and conditional statements
4. Loops

Intro to JavaScript

- Scripting language used for both frontend and backend development.
- ECMA created ECMAScript standards which set the groundwork for JavaScript.
- New versions of ECMAScript came out over time such as ES6.
- Modern browsers like Chrome, Safari and Firefox offer a built-in console that can be used for debugging.
- You can also run javascript code directly in the console as well.

Variables

- Variables are used to store the data which we can use for calculations later on.
- Like a box that holds information.
- We need to declare it first, meaning that we need to assign a storage space in memory and give it a name.
- Variable declaration:

  let exampleVariable = “Value of the variable”;

- You either use the keyword ‘let’ or ‘const’
- Then write the name of the variable after leaving a white space.

Variables

- Make sure,
  - The variable name contains only letters, number and underscores.
  - Follows a consistent convention like the camelCase.
  - The variable is not a reserved keyword like let, const or console
  - Is understandable.
- To assign a value to the variable, you use the **assignment operator** (=) and then the value. Finally end the line with a **semicolon** (;)
- The value assigned to a variable is called a data type.

Data Types

- Numeric → let gameScore = 100;
- String → let fullName=”Muhammad Zahir Junejo”;
- Boolean → let gameEnd = true;
- Array → let playerScores= [100, 200, 400];
- Object → let playerProfileData = {firstName: “Zahir”, lastName: “Junejo”};
- Arrays are used to store multiple values.
- An object is a data type that stores a collection of related data.
- JavaScript infers the data type of the variable from the value.

Mathematical Operators

```
let num1 = 152;
let num2 = 10;

console.log("num1 = " + num1); // num1 = 152
console.log("num2 = " + num2); // num2 = 10
console.log("num1 + num2 = " + (num1 + num2)); // num1 + num2 = 162
console.log("num1/num2=" + num1 / num2); // num1 / num2 = 15.2
console.log("num2 % num1 = " + (num2 % num1)); // num1 % num2 = 10
console.log("++num1 = " + ++num1); // ++num1 = 153
console.log("--num2 = " + --num2); // --num2 = 9
```

If Statements

- A way of comparing two variables and running an action on the basis of the outcome of the comparison.
- Syntax:

```
if ( condition )  {
        Indented statements;
}
```

- Example:

```
let num = 10;
if ( num > 5 ) {
        console.log( “Number is bigger than 5” );
}
```

Comparison Operators

- a > b → Greater than.
- a < b → Less than.
- a >= b → Greater than or equal to.
- a <= b → Less than or equal to.
- a == b → Equals operator.
- a === b → Check for equal value and datatype.
- a != b or a!==b → Does not equal.

Logical Operators

- AND

```
let runs = 150;
if ( runs >= 100 && runs <= 200 ) {
        console.log("Keep going, you are almost headed for a double century.");
}
```

- OR

```
let runs = 20;
if (runs > 40  || runs > 90) {
        console.log("Keep batting, you are almost hitting a century");
}
```

Else

- Syntax:

```
if (condition) {
        Statements to run when condition is true;
} else {
        Statements to run when condition is false;
}
```

- Examples:

```
if (num < 12) {
        console.log(“num is less than 12”);
} else {
        console.log(“num is greater than 12”);
}
```

While loop

- Loops help you repeat a task multiple times.
- While loops repeat an action until the controlling condition is false.
- Syntax:

```
while ( boolean expression ) {
        statements;
}
```

- Example:

```
while ( num <= 250 ) {
        num++;
        console.log(num);
}
```

For loop

- Similar to while loop but structured differently.
- Syntax:

```
for ( declare controlling variable; condition to end loop; increment control variable ) {
        Actions to perform over here;
}
```

- Example:

```
for ( let count = 0; count < 10; count++ ) {
        console.log(count);
}
```

Break statement

- Primarily used to break out of a loop.
- Example:

```
for ( let count = 0; count < 10; count++ ) {
        if ( count == 6 ) {
                break;
        }

        console.log(count);
}
```

Questions and Answers

References

- https://www.programiz.com/javascript/comparison-logical
- https://www.programiz.com/javascript/if-else
- https://www.programiz.com/javascript/for-loop
- https://www.programiz.com/javascript/while-loop
- https://www.programiz.com/javascript/break-statement
- https://www.programiz.com/javascript/continue-statement
- https://www.programiz.com/javascript/switch-statement

Thank You!