

JavaScript

Console- panel that displays important messages, like errors, for developers. Most code is invisible but we can use print or log to our console directly. Refers to object, collection of data and actions.

Keywords- built into JS language so computer will recognize and treat them separately.

Console.log()- inside parentheses get printed (logged)

Data Types- classifications we give to different kinds of data we use in programming. JS has 7.

1. **Numbers**- including decimals ex: 4, 1500, 24.92
2. **String**- any grouping of characters (letters, numbers, spaces, symbols) surrounded 'by' "quotes" ex: console.log('I love Animals');
3. **Boolean**- 2 values, True or False.
4. **Null**- represents the intentional absence of a value, represented by keyword null (without quotes)
5. **Undefined**- also represents the intentional absence of a value, keyword undefined (without quotes)
6. **Symbol**- Unique identifiers, more complex.
7. **Object**- collections of related data

Operators- character that performs a task in our code.

Arithmetic Operators- perform mathematical calculations on numbers.

1. Add: +
2. Subtract: -
3. Multiply: *
4. Divide: /
5. Remainder (modulo): %

Ex: console.log(4 * 2); (prints 8). Console.log(12 % 3); (prints 0)

String Concatenation- appending one string to another. Make sure to include spaces.

Ex: console.log('hi' + 'ya'); (prints hiya). Console.log('I love to ' + 'code.').

Console.log('One' + ' , ' + 'two' + ' , ' + 'three!'); ('One, two, three!')

Length- stores the number of characters in that string. Can retrieve info by appending the string ex: console.log('Hello'.length); (prints 5 because hello is 5 characters).

Methods- actions we can perform. Multiple string methods. We "call" or use these by appending an instance with:

- A period (the dot operator '.')
- The name of the method
- Opening and closing parentheses

Ex: 'example string'.methodName()

Console.log('hello'.toUpperCase()); (prints HELLO)

Console.log('hey'.startsWith('H')); (prints true)

Console.log(' Remove Whitespace '.trim()); (prints Remove Whitespace)

Built-in Objects- has built-in, eventually you will build your own built-ins.

Ex: 'Math' object. Using .random() method with built-in 'math' object:

Console.log(Math.random()); (prints a random number between 0 & 1).

To generate a random number between 0 and 50:

Math.random() * 50;

To ensure answer is a whole number, use Math.floor(). (rounds down to nearest whole number) ex: Math.floor(Math.random() * 50);

To see this printed: console.log(Math.floor(Math.random() * 50));

Great mathematical references:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number

Variables

Var- variable, keyword that creates or declares a new variable

Ex: var myName = 'Alice'

Console.log(myName); (output Alice)

Ex: var favoriteFood = 'avocado'

Console.log(favoriteFood) (output avocado)

Camel Casing- lower and every first after upper ex: camelCaseEverything

=- Assignment Operator. In above example, assigns Alice to myName

General rules for naming variables:

- Variable names cannot start with numbers
- Variable names are case sensitive. Do not create same name with differ cases.
- Variable names cannot be the same as keywords.

Here is a list of keywords:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#keywords

let- variable can be reassigned a different value.

Ex: let meal = 'Enchiladas';

Console.log(meal); (output enchiladas)

Meal = 'Burrito';

Console.log(meal); (output burrito)

If we don't assign a value, its undefined. Can reassign value of the variable.

Ex: let price;

Console.log(price); (output undefined)

Price = 350;

Console.log(price); (output 350)

let changeMe = true;

changeMe = false; (change the value)

const- constant value cannot be reassigned (if try, get TypeError). Must be assigned a value (if don't, get SyntaxError).

Ex: const myName 'Alice'

Console.log(myName); (Output Alice)

Mathematical Assignment Operators- use variables and math operators to calculate new values and assign them to a variable

let w = 4;

W = w + 1;

Console.log(w); (output 5)

OR

let w = 4;

W += 1;

Console.log(w); (output 5)

Let x = 20;

X -= 5;

Console.log(x) (output 15)

OR

let x = 20

x = x - 5

console.log(x); (output 15)

Let y = 50;

Y *= 2;

Console.log(y); (output 100)

OR

let y = 50;

y = y * 2

console.log(y); (output 100)

Let z = 8;

Z /= 2;

Console.log(z); (output 4)

OR

let z = 8

z = z / 2

console.log(z); (output 4)

Increment Operator (++)- increase the value of the variable by 1

Decrement Operator (--)- decrease the value of the variable by 1

Ex: let a = 10;

a++;

console.log(a); (output 11)

ex: let b = 20;

b--;

console.log(b); (output 19)

String Concatenation with Variables

'+' Operator- combine 2 string values even if they are being stored in variables.

Ex: let myPet = 'dog';

Console.log('I own a pet ' + myPet + '.'); (output: I own a pet dog.)

String Interpolation

```
Const myPet = 'dog';
```

```
Console.log(`I own a pet ${myPet}.`); (Output: I own a pet dog.)
```

- A template literal is wrapped by backticks `
- Inside template literal, a placeholder `${myPet}`. The value of `myPet` is inserted into the template literal
- When we interpolate ``I own a pet ${myPet}.``, (Output: I own a pet dog.)

```
Ex: let myName = 'Alice';
```

```
    Let myCity = 'Conifer'
```

```
    Console.log(`My name is ${myName}. My favorite city is ${myCity}.`)
```

Typeof operator- checks the type of data. Returns or passes back a string of the data type

```
Ex: const unknown1 = 'foo';
```

```
    Console.log(typeof unknown1); (output: string)
```

```
    Const unknown2 = 10;
```

```
    Console.log(typeof unknown2); (output: number)
```

```
    Const unknown 3 = true;
```

```
    Console.log(typeof unknown3); (output: Boolean)
```

```
    Let newVariable= 'Playing around with typeof';
```

```
    Console.log(type of variable); (output: string)
```

```
    newVariable= 1;
```

```
    console.log(typeof newVariable); (output: number)
```

Conditional Statements

Conditional Statements- checks specific condition(s) and performs task based on condition

If- ex: if (true) {

```
    Console.log('This message will print!');
```

```
} (output: This message will print!)
```

- 'if' followed by (), followed by code block or block statement, indicated by {}
- Inside (), condition is provided that evaluates to true or false
- If condition evaluates true, code {} runs/executes
- If condition evaluates false, no execution.

```
Let sale = (true)
```

```
If (sale) {
```

```
    Console.log('Time to buy!');
```

```
} (output: Time to buy!)
```

If...Else- if evaluates false run... Allow us to automate solutions to yes or no questions (binary decisions)

```
Ex: if (false) {  
    Console.log('The code in this block will not run.');
```

```
} else {  
    Console.log('But the code in this block will!');
```

```
}
```

Else statement:

- Uses else keyword following the code block of an if statement
- Has a code block that is wrapped by a set of {}
- Code inside else statement code block will execute when if statement's condition evaluates to false.

```
Ex: let sale = true  
    Sale = false;
```

```
If(sale) {  
    Console.log('Time to buy!');
```

```
} else {  
    Console.log('Time to wait for sale.')
```

```
} (output: Time to wait for a sale)
```

Comparison Operators- compare value on the left to value on the right

- Less than: <
- Greater than: >
- Less than or equal to: <=
- Greater than or equal to: >=
- Is equal to: ===
- Is not equal to: !==

```
Ex: 10 < 12 (evaluates to true)
```

```
Ex: 'apples' === 'oranges' (false)
```

```
Ex: let hungerLevel = 7
```

```
If(hungerLevel > 7) {  
    Console.log('Time to eat!')
```

```
} else {  
    Console.log('We can eat later!')
```

```
} (output: we can eat later)
```

Logical Operators- add more sophisticated logic to our conditionals. 3:

- The 'and' operator (&&)
- The 'or' operator (||)
- The 'not' operator, or 'bang' operator (!)

&&- checking that 2 things are true

```
Ex: if (stoplight === 'green' && pedestrians === 0) {  
    Console.log('Go!');
```

```
} else {  
    Console.log('Stop'); }
```

|| - either condition true

```
Ex: if (day === 'Saturday' || day === 'Sunday') {  
    Console.log('Enjoy the weekend!');  
} else {  
    Console.log('Do some work.');
```

! - reverses or negates value of Boolean.

```
Ex: let excited = true;  
    Console.log(!excited); (output false)  
    Let sleepy = false;  
    Console.log(!sleepy); (output true)
```

```
Ex: let mood = 'sleepy';  
    Let tirednessLevel = 6;  
  
    If (mood === 'sleepy' && tirednessLevel > 8) {  
        Console.log('time to sleep')  
    } else {  
        Console.log('not bed time yet')  
    }
```

Truthy and Falsy

How non-boolean data types (strings or numbers) are evaluated. Check if variable exists.

```
Ex: let myVariable = 'I Exist!';  
If (myVariable) {  
    Console.log(myVariable)  
} else {  
    Console.log('The variable does not exist.')} (output: if, because it has a truthy value)
```