1. Console

The console object provides access to the browser’s debugging console. There are various methods available on the console object such as console.log, console.error, and console.warn.

Ex:

console.log('Hello World')

console.error('This is an error')

console.warn('This is a warning')

2. Variables

Variable is a named location for storing a value. Variables can be set with var, let, or const.

var is globally scoped and can potentially cause problems, let allows direct value reassignment and can be initialized without a value, while const doesn’t allow value reassignment and needs to have a value.

Ex: var age = 30; let score; or let score = 80; const name = ‘John Doe’;

age = 20; score = 100;

3. Data types

Data type is a characteristic of a value affecting what kind of data it can store. The types of data that can be assigned to variables are

- Primitives

String is a sequence of characters marked with a single or double quotation mark.

Ex: const name = ‘John Doe’;

Number is a numeric data type. In other languages, there are different numeric types like integers, floats, doubles, and bignums.

Ex: const age = 25; or const rating = 4.5;

Boolean is a logical data type that can only have the values true or false.

Ex: const isWhite = false;

null represents the intentional absence of any object value. In JS, typeof null is object because null has 0 as a type tag in JS, resulting in the bogus typeof return value.

Ex: const x = null;

undefined is automatically assigned to variables that isn’t assigned a value

Ex: const y = undefined; or let z;

- Objects can contain many values of different types. Objects can have functions/properties as well.

Ex: var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

// To test the data type, use console.log(typeof var)

4. Converting strings to numbers

The parseInt() function converts a string of numbers into an integer number data type, while the parseFloat() function converts a string of decimal numbers into a float number data type. A string of words can’t be converted and will return NaN (Not-a-Number), and an integer number will be converted into an integer number data type even with the parseFloat() function.

Ex:

let number = '123.456'

console.log(parseInt(number))

Output: 123

let number = '123.456'

console.log(parseFloat(number))

Output: 123.456

5. String properties & methods

String properties doesn’t need to be written with a parenthesis unlike string methods.

Ex: const s = ‘Hello World’;

property --> console.log(s.length); //

method --> console.log(s.toUpperCase());

// It can also be combined with another dot. Ex: console.log(s.toUpperCase().substring(0,5));

6. Arrays

Arrays are special variables that can hold multiple values under a single name. Arrays can be created using array literals or using the JS keyword new. Ex:

var cars = ["Saab", "Volvo", "BMW"];

var cars = new Array("Saab", "Volvo", "BMW");

console.log(a.split()) is used to split a string into an array.

Ex:

var a = ['technology’];

console.log(a.split('')); // Output: “t”, “e”, “c”, “h”, “n”, “o”, “l”, “o”, “g”, “y”

var b = ['technology, computers, it, programming'];

console.log(b.split(', ')); // Output: “technology”, “computers”, “it”, “programming”

7. Concatenation & template string

Concatenation ex: console.log('My name is ' + name + ' and I am ' + age)

Template string ex: console.log(`My name is ${name} and I am ${age}`)

8. Arithmetic operation

+ Addition / Division

- Subtraction % Modulus (sisa bagi)

\* Multiplication ++ Increment (will plus 1)

\*\* Exponentiation -- Decrement (will minus 1)

Ex:

var x = 100 + 50;

var x = (100 + 50) \* a;  
var z = 5 \*\* 2; // Output: 25

9. Basic math object

console.log(Math.pow(base, exponent)) // Output: baseexponent

console.log(Math.sqrt())

console.log(Math.cbrt())

console.log(Math.round(2.8)) // rounds a number to the nearest integer. Output: 3

console.log(Math.floor(6.8)) // rounds a number down to the previous largest integer. Output:5

console.log(Math.ceil(9.2)) // rounds a number up to the next largest integer. Output: 10

console.log(Math.random()) // returns a floating-point, pseudo-random number in the range of 0-1

console.log(Math.max(3, 5, 7)) // returns the largest of the given numbers. Output: 7

console.log(Math.min(3, 5, 7)) // returns the smallest of the given numbers. Output: 3

10. Basic date object

Date objects are created with the new Date() constructor. There are 4 ways to create a new date object:

new Date() // creates a new date object with the current day and time  
new Date(*year, month, day, hours, …*) // creates a new date object with a specified date and time  
new Date(*milliseconds*) // creates a new date object with zero time plus milliseconds  
new Date(*date string*) // creates a new date object from a date string

Date methods allow you to get and set the year, month, day, hour, … using local time or UTC time. There are 2 types of date method:

- Get date methods can be used for getting information from a date object. A few get date methods are: getFullYear(), getMonth(), getDate(), getHours(), getTime(), Date.now().

- Set date methods lets you set date values (year, month, day, hour, …) for a date object. A few set date methods are:

setFullYear(), setMonth(), setDate(), setHours(), setTime().

11. Assignment operators

= assigns a value to a variable

+= adds a value to a variable

-= subtracts a value to a variable

\*= multiplies a variable

/= divides a variable

%= assigns a remainder (sisa bagi)

12. Comparison & logical operators

Comparison & logical operators are used to test for true or false. Comparison operators are used to determine equality or difference between variables or values, while logical operators are used to determine the logic between variables or values.

Comparison operators:

== equal to > greater than

=== equal value and equal type < less than

!= not equal >= greater than or equal to

!== not equal value or not equal type <= less than or equal to

Logical operators:

&& and

|| or

! not

13. Prompt & alert

Prompt is a function that displays a dialog with an optional message prompting the user to input some text, while alert is a function that displays an alert dialog with the optional specified content.

There are a few ways to use the prompt & the alert feature.

Ex:

let sign = prompt('Write something here')

sign = prompt('Write something here')

sign = window.prompt('Write something here’)

alert('This is an alert')

window.alert('This is an alert')

14. Conditionals

In JS, there 4 conditional statements:

- if is used to specify a block of code to be executed if a specified condition is true

- else is used to specify a block of code to be executed if the same condition is false

- else if is used to specify a new condition to test if the first condition is false

- switch is used to specify many alternative blocks of code to be executed

if (*condition1*) {  
  //*block of code to be executed if condition1 is true*} else if (*condition2*) {  
  //*block of code to be executed if the condition1 is false and condition2 is true*  
} else {  
  //*block of code to be executed if the condition1 is false and condition2 is false*}

switch(*expression*) {  
  case *x*:  
*// code block*    break;  
  case *y*:  
*// code block*    break;  
  default:  
    // *code block*  
}

15. Loops

Loops can execute a block of code a number of times as long as a specified condition is true. There are different kinds of loops:

- for loops through a block of code a number of times

Ex: for (i = 0; i < 5; i++) {  
  text += "The number is " + i + "<br>";  
}

- for/in loops through the properties of an object

Ex: var person = {fname:"John", lname:"Doe", age:25};  
var text = "";  
var x;  
for (x in person) {  
  text += person[x];  
}

- for/of loops through the values of an iterable object

Ex: var cars = ['BMW', 'Volvo', 'Mini']; var txt = 'JavaScript';   
var x; ;  
for (x of cars) { for (x of txt) {  
  document.write(x + "<br >"); document.write(x + "<br>");  
} }

- while loops through a block of code while a specified condition is true

Ex: while (i < 10) {  
  text += "The number is " + i;  
  i++;  
}

- do/while loops through a block of code while a specified condition is true but will execute it before confirming it

Ex: do {  
  text += "The number is " + i;  
  i++;  
}  
while (i < 10);

16. Function