1. invented by Brendan Eich in 1995
2. External JavaScript Advantages
3. It separates HTML and code
4. It makes HTML and JavaScript easier to read and maintain
5. Cached JavaScript files can speed up page loads
6. **arithmetic operators** ( + - \* / ) to **compute**
7. **assignment operator** ( = ) to **assign**

## JavaScript Identifiers / Names

A JavaScript name must begin with:

1. A letter (A-Z or a-z)
2. A dollar sign ($)
3. Or an underscore (\_)

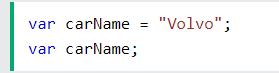
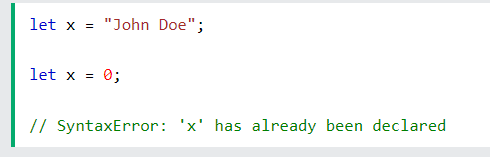
## Camel Case

1. **Upper Camel Case (Pascal Case):**
2. FirstName, LastName, MasterCard, InterCity.
3. Variables are containers for storing data
4. The var keyword is used in all JavaScript code from 1995 to 2015.
5. The let and const keywords were added to JavaScript in 2015.
6. If you want your code to run in older browser, you must use var.
7. constant values and cannot be changed

## JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

1. Creating a variable in JavaScript is called "declaring" a variable.
2. After the declaration, the variable has no value (technically it is undefined).
3. To **assign** a value to the variable, use the equal sign
4. If you re-declare a JavaScript variable declared with var, it will not lose its value. 
5. You cannot re-declare a variable declared with let or const.
6. If you put a number in quotes, the rest of the numbers will be treated as strings, and concatenated.
7. “5”+2+3=523
8. 5+2+”3”=73
9. JavaScript treats a dollar sign and underscore as a letter
10. The let keyword was introduced in [ES6 (2015)](https://www.w3schools.com/js/js_es6.asp).
11. Variables defined with let cannot be Redeclared in the same block. But in another block can be redeclared.
12. 
13. Graphical user interface, text, application, chat or text message

    Description automatically generated
14. Variables defined with let must be Declared before use.
15. Variables defined with let have Block Scope
16. With let, redeclaring a variable in the same block is NOT allowed:
17. The const keyword was introduced in [ES6 (2015)](https://www.w3schools.com/js/js_es6.asp).
18. Variables defined with const cannot be Redeclared.
19. Variables defined with const cannot be Reassigned.
20. Variables defined with const have Block Scope.
21. Background pattern

    Description automatically generated with low confidence
22. let and const.
23. These two keywords provide **Block Scope** in JavaScript.
24. Variables declared inside a { } block cannot be accessed from outside the block: Graphical user interface

    Description automatically generated with medium confidence
25. var keyword can NOT have block scope. Application

    Description automatically generated with medium confidence
26. Text

    Description automatically generated with low confidence
27. As a general rule, always declare a variable with const unless you know that the value will change
28. The keyword const is a little misleading.

Because of this you can NOT:

* Reassign a constant value
* Reassign a constant array
* Reassign a constant object

But you CAN:

* Change the elements of constant array
* Change the properties of constant object

1. 
2. Declaring a variable with const is similar to let when it comes to **Block Scope**.
3. Text, letter

   Description automatically generated

# **JavaScript Operators**

1. If you add a number and a string, the result will be a string. Because of type conversion.

|  |  |
| --- | --- |
| == | equal to |
| === | equal value and equal type |
| ? | ternary operator |

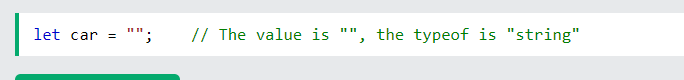
## JavaScript Arithmetic Operators

1. **Table

   Description automatically generated**
2. The numbers (in an arithmetic operation) are called **operands**.
3. the result of a **modulo operation** is the **remainder** of an arithmetic division.
4. x \*\* y produces the same result as Math.pow(x,y):
5. Operator precedence describes the order in which operations are performed in an arithmetic expression.

# **JavaScript Data Types**

1. When adding a number and a string, JavaScript will treat the number as a string.
2. JavaScript evaluates expressions from left to right. Different sequences can produce different results: Graphical user interface, text, application, chat or text message

   Description automatically generated
3. Object properties are written as name:value pairs, separated by commas.
4. he typeof operator returns the type of a variable or an expression: 
5. In JavaScript, a variable without a value, has the value undefined. The type is also undefined. 
6. empty string

# **JavaScript Functions**

1. Graphical user interface, text, application

   Description automatically generated
2. Function **parameters** are listed inside the parentheses () in the function definition.
3. Function **arguments** are the **values** received by the function when it is invoked.

## Why Functions?

1. You can reuse code: Define the code once, and use it many times.
2. You can use the same code many times with different arguments, to produce different results.

## () Operator Invokes the Function

1. Accessing a function without () operator will return the function object instead of the function result.

## Local Variables

1. Variables declared within a JavaScript function, become **LOCAL** to the function. Local variables can only be accessed from within the function. Local variables are created when a function starts, and deleted when the function is completed.

# **JavaScript Objects**

1. Graphical user interface, text

   Description automatically generated
2. Chart

   Description automatically generated with medium confidence
3. It is a common practice to declare objects with the const keyword.
4. The **name:values** pairs in JavaScript objects are called **properties**:
5. access object properties in two ways: Graphical user interface, text, application, chat or text message

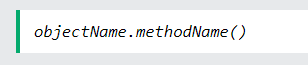
   Description automatically generated

## Object Methods

1. Methods are **actions** that can be performed on objects.
2. A method is a function stored as a property. Graphical user interface, text, application

   Description automatically generated

## Accessing Object Methods

1. 

## If you access a method **without** the () parentheses, it will return the **function definition**:

## Do Not Declare Strings, Numbers, and Booleans as Objects!

1. Text

   Description automatically generated
2. When a JavaScript variable is declared with the keyword "new", the variable is created as an object:

# **JavaScript Events**

1. events are **"things"** that happen to HTML elements.  JavaScript can **"react"** on these events. 

## Common HTML Events

1. Table

   Description automatically generated

# **JavaScript Strings**

1. Graphical user interface, application

   Description automatically generated

## Escape Character

1. use the **backslash escape character**. Graphical user interface, application, Word

   Description automatically generated
2. For best readability, programmers often like to avoid code lines longer than 80 characters.

## Strings as Objects

1. strings can also be defined as objects with the keyword new:

let y = new String("John");

1. Graphical user interface, application, website

   Description automatically generated
2. Graphical user interface, application, chat or text message, website

   Description automatically generated
3. Graphical user interface, text, application, chat or text message

   Description automatically generated

This both will always return false, as two objects are created every time

# **String Methods**

1. length property

## Extracting String Parts

1. Graphical user interface, application

   Description automatically generated

## slice()

1. Graphical user interface, text

   Description automatically generated with medium confidence
2. Graphical user interface, text

   Description automatically generated
3. If you omit the second parameter, the method will slice out the rest of the string:
4. A picture containing Word

   Description automatically generated
5. substring() is similar to slice().
6. The difference is that start and end values less than 0 are treated as 0 in substring().Graphical user interface, text

   Description automatically generated with medium confidence
7. substr() is similar to slice().
8. The difference is that the second parameter specifies the **length** of the extracted part.
9. If the first parameter is negative, the position counts from the end of the string. Graphical user interface, text, application

   Description automatically generated

Will return Kiwi

## Replacing String Content

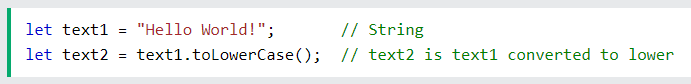
1. Graphical user interface, text, application

   Description automatically generated
2. The replace() method does not change the string it is called on.
3. The replace() method returns a new string.
4. The replace() method replaces **only the first** match
5. replace() method is case sensitive.
6. To replace case insensitive, use a **regular expression** with an /i flag (insensitive): 
7. To replace all matches, use a **regular expression** with a /g flag (global match): Graphical user interface

   Description automatically generated with medium confidence

## Converting to Upper and Lower Case

1. A picture containing graphical user interface

   Description automatically generated
2. 

## String concat()

1. VGraphical user interface, text, application

   Description automatically generated
2. All string methods return a new string. They don't modify the original string. Strings are immutable: Strings cannot be changed, only replaced.

## String trim()

1. Graphical user interface, application, Teams

   Description automatically generated
2. trimStart() ,  trimEnd()

## Extracting String Characters

1. There are 3 methods for extracting st
2. charAt(*position*)
3. charCodeAt(*position*)
4. Property access [ ]
5. Chart

   Description automatically generated
6. The charCodeAt() method returns the unicode of the character at a specified index in a string:
7. The method returns a UTF-16 code (an integer between 0 and 65535).
8. Graphical user interface, text, application, chat or text message

   Description automatically generated
9. 

## Converting a String to an Array

1. split() method:
2. A screenshot of a computer

   Description automatically generated with medium confidence