JavaScript

Programming

JavaScript

JavaScript was created in 1995 by Brendan Eich at Netscape

JavaScript was created in 10 days

Originally called Mocha, then LiveScript, and finally renamed to JavaScript for marketing reasons

JavaScript quickly became popular due to its ability to add interactivity and dynamic behavior to web pages

Characteristics of Javascript

- High-level Language
- Object-Oriented
- Dynamic Typing
- Interpreted Language
- Client-Side Language
- Server-Side Language
- Event-Driven
- Cross-Platform Compatibility
- Large Ecosystem

What is javascript used for

1. Creating dynamic web pages
2. Validating form inputs
3. Animating web page elements
4. Building web applications
5. Communicating with servers
6. Manipulating HTML and CSS
7. Creating web-based games
8. Building desktop and mobile applications: Electron and React Native to build cross-platform desktop and mobile applications.

JavaScript ecosystem

1. JavaScript language: The core programming language that enables developers to add interactivity and functionality to web pages.

2. Browser APIs: A set of built-in APIs provided by web browsers that allow developers to access and manipulate web page content, handle events, and make network requests.

3. JavaScript libraries: Collections of pre-written code that can be used to perform specific tasks or operations, such as manipulating the DOM, handling events, or making AJAX requests. Examples include jQuery, Lodash, and Moment.js.

4. JavaScript frameworks: Comprehensive sets of pre-written code that provide a structure for building applications, often including a set of rules, guidelines, and tools that dictate how an application should be structured and how different components should interact. Examples include React, Angular, and Vue.js.

5. Backend technologies: A variety of technologies and frameworks that enable developers to build server-side applications and APIs using JavaScript, such as Node.js, Express, and MongoDB.

6. Build tools: Tools that automate common tasks in the development process, such as transpiling code, bundling and minifying assets, and running tests. Examples include Webpack, Babel, and Jest.

Frontend web development:

- React: Facebook, Instagram
- Angular: Google, Microsoft, Forbes
- Vue.js: Alibaba, Xiaomi, 9GAG
- Ember.js: Square, Microsoft, Yahoo
- Backbone.js: Airbnb, Pinterest, Foursquare

Backend web development:

- Node.js: LinkedIn, Netflix, Uber
- Express.js: MySpace, Accenture, Mozilla
- Koa: Alibaba, IBM, Zillow
- Meteor: Mazda, Honeywell, IKEA
- Hapi: Walmart, Disney, Mozilla

Mobile app development

- React Native: Facebook, Instagram, Skype
- NativeScript: Progress, Sennheiser, Deloitte
- Ionic: Pacifica, Nationwide, Diesel
- PhoneGap/Cordova: Wikipedia, IBM, Salesforce

Desktop app development

- Electron: Slack, Discord, Visual Studio Code
- NW.js: Trello, Brackets, WebTorrent
- React Native Desktop: Microsoft, Atlassian, Discord

JavaScript Keywords

In JavaScript, keywords are reserved words that have a specific meaning and functionality within the language.

Examples:

break, case, catch, class, const, continue, debugger, default, delete, do, else, export, extends, false, finally, for, function, if, import, in, instanceof, let, new, null, return, super, switch, this, throw, true, try, typeof, var, void, while, with, yield

JavaScript Variables

A variable is a named storage location in the computer's memory that holds a value.

Variables pass through three different stages

- Variable Declaration
- Variable Initialization
- Variable Assignment

Variable Declaration

Declaring a variable means creating a named storage location in the computer's memory. In JavaScript, a variable can be declared using the var, let, or const keyword.

```
var myVar;

let myLet;

const myConst;
```

Variable Initialization

Initializing a variable means giving it an initial value. A variable can be initialized at the same time it is declared, or later in the program.

```
var myVar = 10;

let myLet = 'Hello';

const myConst = true;
```

Variable Assignment:

Assigning a value to a variable means updating the value stored in the variable. A variable can be assigned a new value any number of times in the program.

```
myVar = 20;

myLet = 'World';

// cannot reassign a value to a const variable
```

Note this

Variable Declaration, Initialization, and Assignment:

In JavaScript, variables can be declared, initialized, and assigned in one statement

```
var myVar = 10;

let myLet = 'Hello';

const myConst = true;
```

It is important to note that `var` variables can be redeclared and reassigned, while `let` and `const` variables cannot be redeclared but can be reassigned. Additionally, `const` variables must be initialized at the time of declaration and cannot be assigned a new value later in the program.

Rules and conventions of naming variables.

1. Variable names should start with a letter, underscore (_) or a dollar sign ($). They cannot start with a number.
2. Variable names can contain letters, numbers, underscores, and dollar signs.
3. Variable names are case sensitive. For example, myVar and myvar are two different variables.
4. Variable names should be descriptive and easy to understand. Avoid using short or single-letter names, except for loop counters.
5. Use camelCase notation for multi-word variable names. For example, myVariableName instead of myvariablename or my_variable_name.
6. Avoid using reserved keywords as variable names, such as var, let, const, if, else, for, while, etc.
7. Use uppercase letters for constants or variables that are not supposed to change, such as PI or MAX_SIZE.

More conventions

- Avoid using abbreviations or acronyms unless they are well-known or widely used.
- Use nouns or noun phrases for variable names, and avoid using verbs or verb phrases.
- Be consistent in your naming conventions throughout your code. If you use camelCase notation for one variable, use it for all variables.
- Use meaningful names that reflect the purpose or content of the variable. For example, firstName, lastName, age, price, etc.
- Avoid using single-letter names except for loop counters or other temporary variables. Use names that are descriptive and easy to understand, even if they are longer.
- Don't use names that are too similar to each other, as this can cause confusion and errors. For example, customer and customers are too similar and can easily be confused.

Assignment

What are the differences between

Var, Let and Const

Resources:
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables