Javascript

History

- First Developed in 1995 by Netscape
- Originally named mocha
- Renamed to Javascript after obtaining license from Sun Microsystems
- Given to ECMA for standardization
- Javascript & ActionScript 3 adopted ECMAScript standard
- Current version ES6

“JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.”

Javascript Engines

- Google’s V8
- Apple’s JavascriptCore
- Microsoft’s Chakra

Using Javascript

- <script></script>
- <script src=”{filepath}.js”></script>

Dynamic Typing

JavaScript is a loosely typed or a dynamic language. That means you don't have to declare the type of a variable ahead of time. The type will get determined automatically while the program is being processed. That also means that you can have the same variable as different types.

Datatypes

- Boolean
- Null
- Undefined
- Number
- String
- Object
- Array
- Date

Arithmetic Operators

- +
- -
- *
- /
- %

Increment & Decrement Operators

- ++
- --

Assignment Operators

- =
- +=
- -=
- *=
- /=

Comparison Operators

- === Strict equality
- !== Strict-non-equality
- <
- >
- <=
- >=

String

```
var one = 'Hello, ';
var two = 'how are you?';
var joined = one + two;
```

Conditionals

- If..else
- If..else if..else
- Switch..case
- ? : - ternary

Loops

```
for (initializer; exit-condition; final-expression) {
  // code to run
}
while (exit-condition) {
  // code to run

  final-expression
}
```

Loops

```
do {
  // code to run


  final-expression
} while (exit-condition)
```

Functions

Functions allow you to store a piece of code that does a single task inside a defined block, and then call that code whenever you need it using a single short command.

```
function add(x, y) {
	return x+y;
}
```

Prototype

When it comes to inheritance, JavaScript only has one construct: objects. Each object has a private property which holds a link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. By definition, null has no prototype, and acts as the final link in this prototype chain.

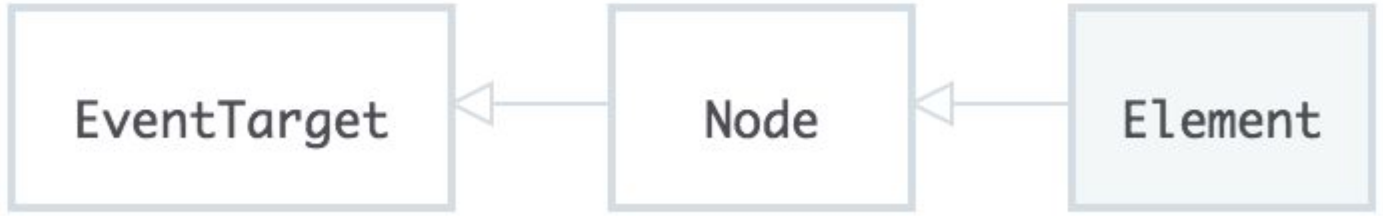
DOM

- Javascript uses DOM to access document and its elements
- The DOM was designed to be independent of any particular programming language, making the structural representation of the document available from a single, consistent API.

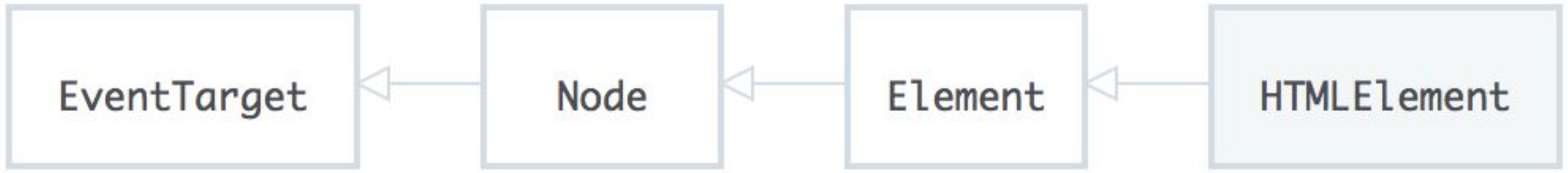
DOM Datatypes

- document
- element
- nodeList
- attribute
- HTMLCollection

Element



HTMLElement



Search/Find Node(s)

- getByElement*
- querySelector*

XMLHttpRequest

Use XMLHttpRequest (XHR) objects to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a Web page to update just part of a page without disrupting what the user is doing. XMLHttpRequest is used heavily in Ajax programming.

XMLHttpRequest

- progress
- load
- error
- abort

Fetch API (Self - Study)

The Fetch API provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as requests and responses. It also provides a global fetch() method that provides an easy, logical way to fetch resources asynchronously across the network.

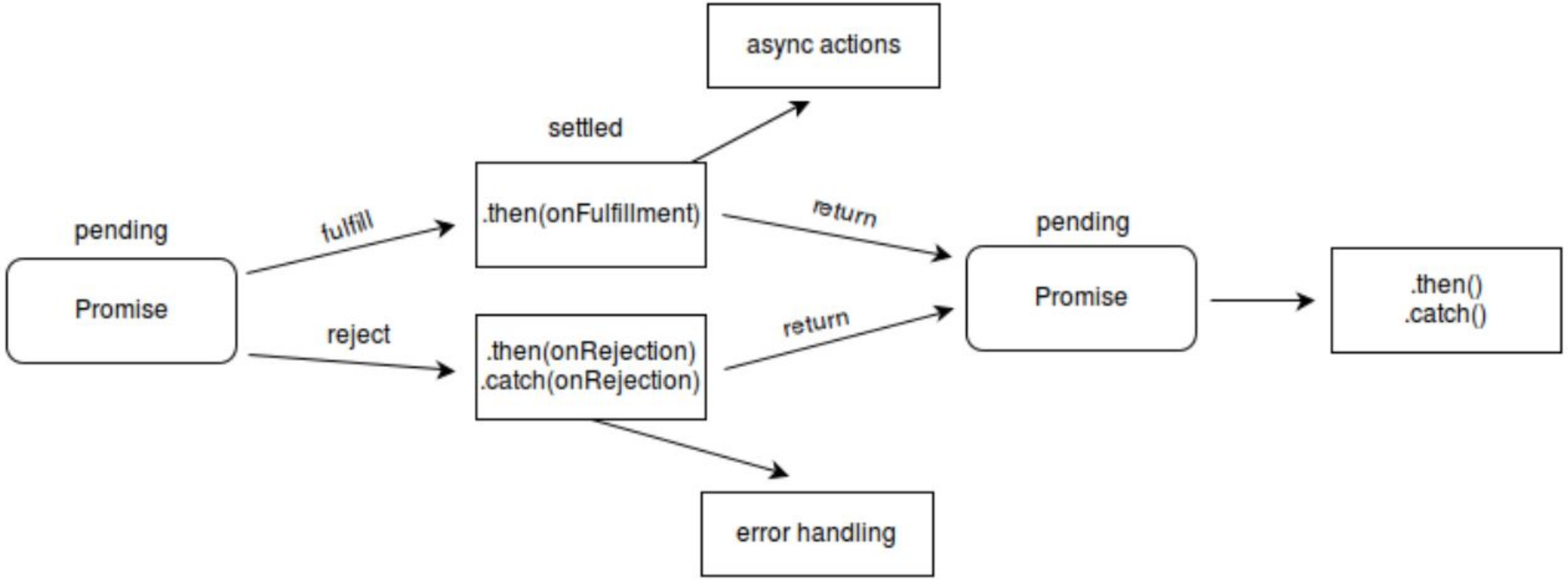
Promise

A Promise is a proxy for a value not necessarily known when the promise is created. It allows you to associate handlers with an asynchronous action's eventual success value or failure reason. This lets asynchronous methods return values like synchronous methods: instead of immediately returning the final value, the asynchronous method returns a promise to supply the value at some point in the future.

A Promise is in one of these states:

pending: initial state, neither fulfilled nor rejected.
fulfilled: meaning that the operation completed successfully.
rejected: meaning that the operation failed.

Promise


pending
Promise
fulfill
reject
settled
.then(onFulfillment)
.then(onRejection)
.catch(onRejection)
async actions
error handling
return
return
pending
Promise
.then()
.catch()

Events

Events are actions or occurrences that happen in the system you are programming, which the system tells you about so you can respond to them in some way if desired. For example if the user clicks a button on a webpage, you might want to respond to that action by displaying an information box.

Event Bubbling

Regex

Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects.

ES6 Features

Regular expressions are patterns used to match character combinations in strings. In JavaScript, regular expressions are also objects.