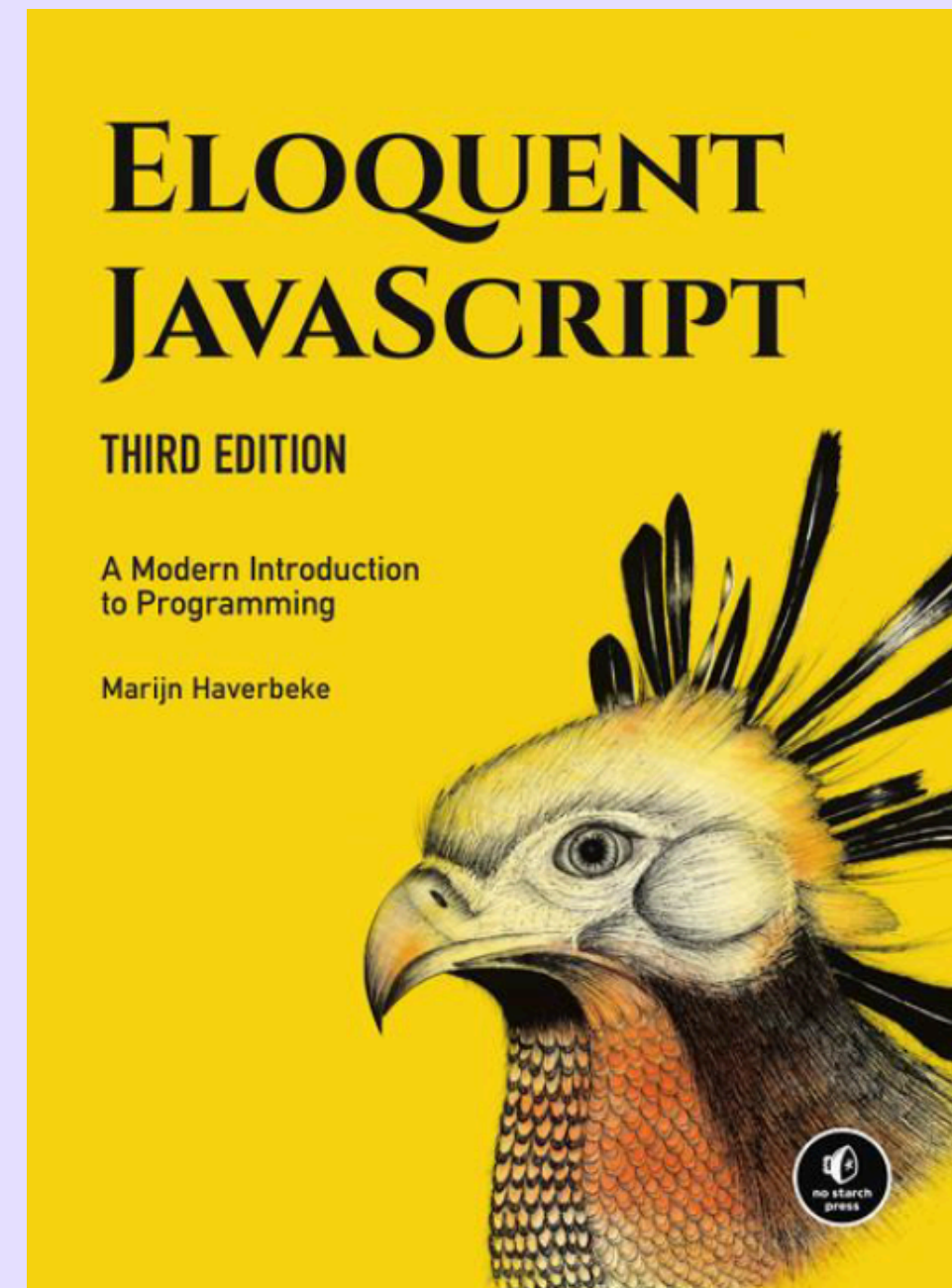


JavaScript *ohjelmointi*

MARGIT TENNOSAARIN MATERIAALISTA
LAURA JÄRVISEN MUOKKAAMA

Viikko	Aihe	Opittavat asiat
1	Mikä on JavaScript, Historia, Kehitysympäristön asetus, CLI	JavaScriptin historian ymmärtäminen, kehitysympäristön asettaminen ja siihen tutustuminen, peruskomennot komentorivillä (CLI), muuttujat, tietotyypit, perusoperaatiot
2	Muuttujat, Tietotyypit, Operaattorit, Perusfunktiot	Muuttuijen, tietotyyppien, operaattoreiden ja perusfunktioiden ymmärtäminen
3	Ehtolauseet, Silmukat, Näkyvyysalueet eli Scopet	Ehtolauseiden ja silmukoiden käyttö, Näkyvyysalueen eli scopen ymmärtäminen
4	DOM Perusteet, Elementtien valinta ja muokkaus, Funktiot	Perus DOM-muokkaus, elementtien valitseminen ja funktiot
5	Taulukot, Taulukkokofunktiot	Taulukoiden muokkaus ja taulukkokofunktioiden käyttö
6	JSON, Oliot	JSONin ja JavaScript olioiden ymmärtäminen
7	Virheen käsittely, JS kielioppi	Virheiden käsittely, JavaScriptin kirjoittamisen hyvät käytännöt
8	Asynkroninen JS, LocalStorage, Fetch ja API:t	Asynkronisen JavaScriptin ymmärtäminen, lupausten ja async/awaitin käyttö, LocalStoragen ja rajapintojen hyödyntäminen
9	Loppuprojekti ja esittely	Opittujen asioiden käyttäminen loppuprojektissa hyviä ohjelmointikäytäntöjä noudattaen, projektin esittely

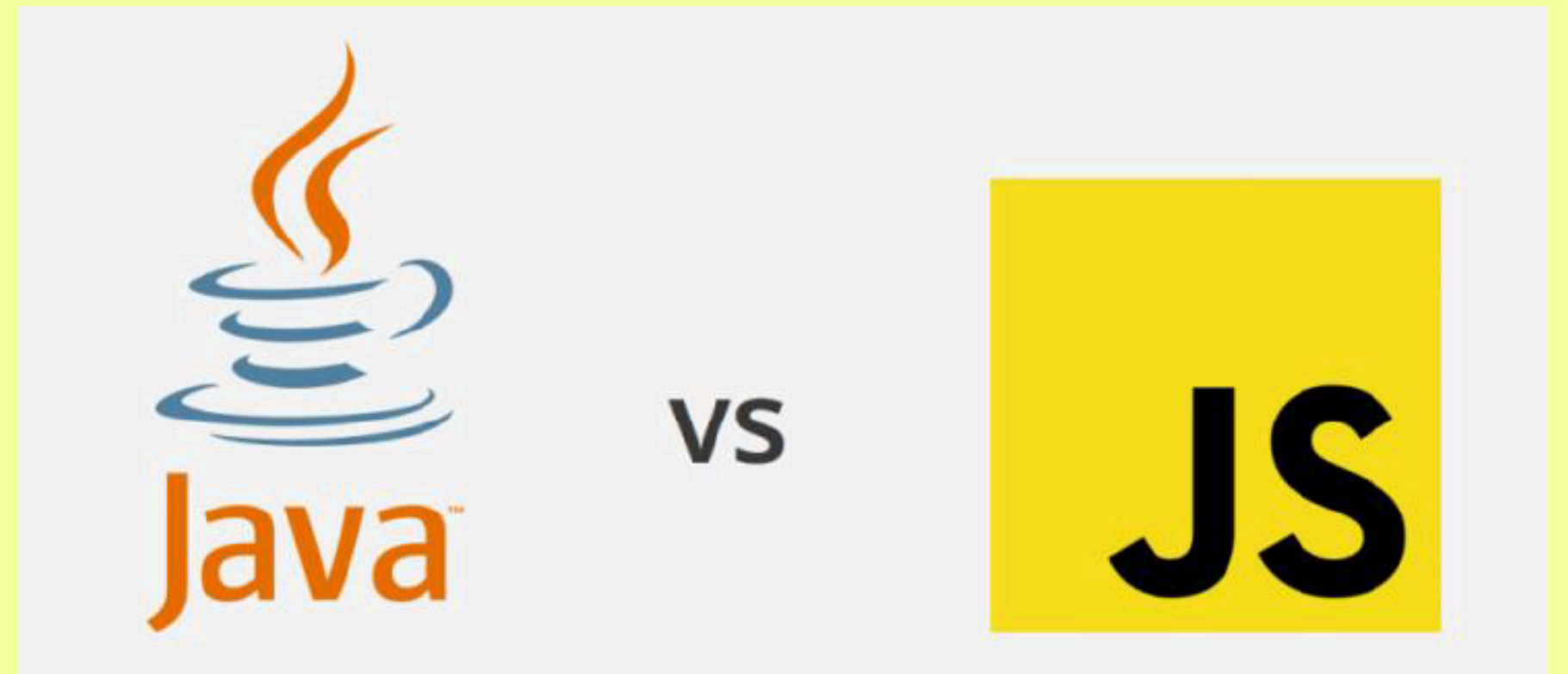
Kurssi pohjautuu kirjaan



<https://eloquentjavascript.net>

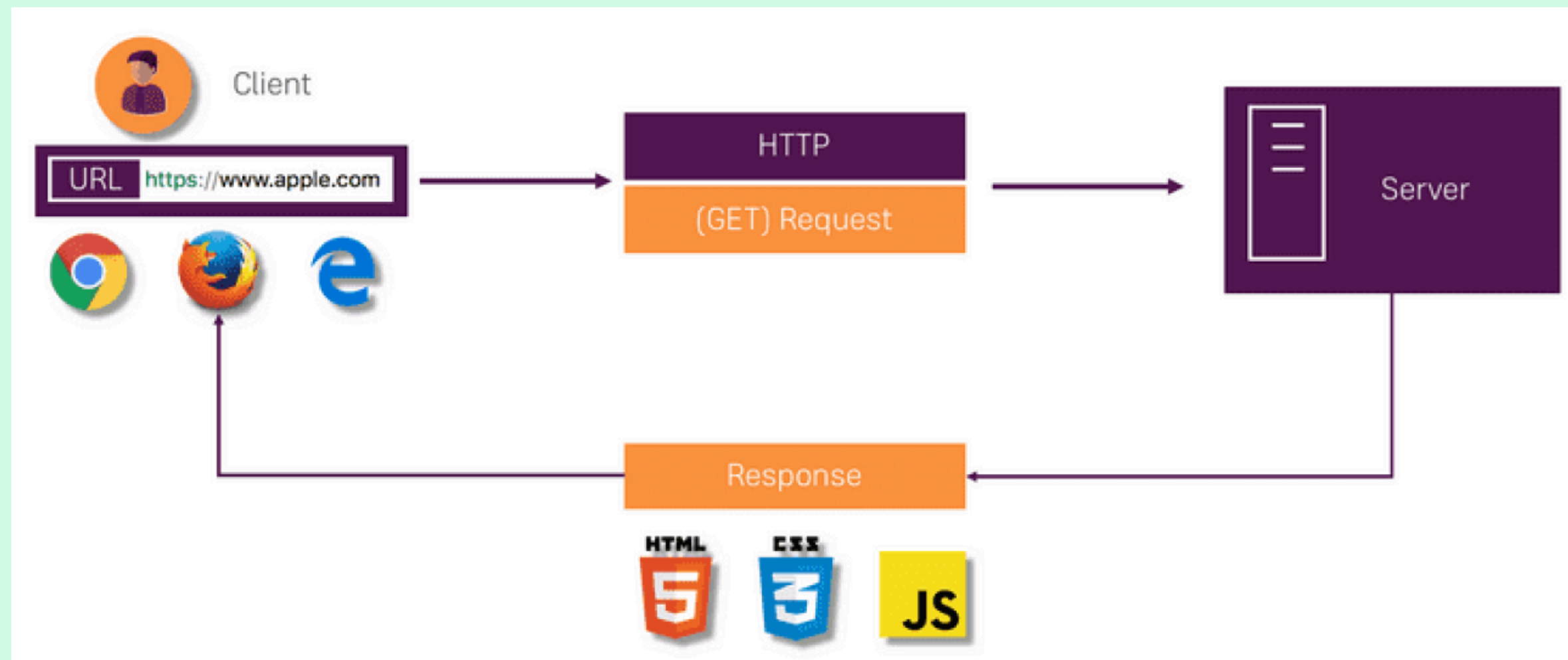
Onko *Java* ja
JavaScript sama asia?

Yhtä sama, kuin kahvi
ja kahvikuppi



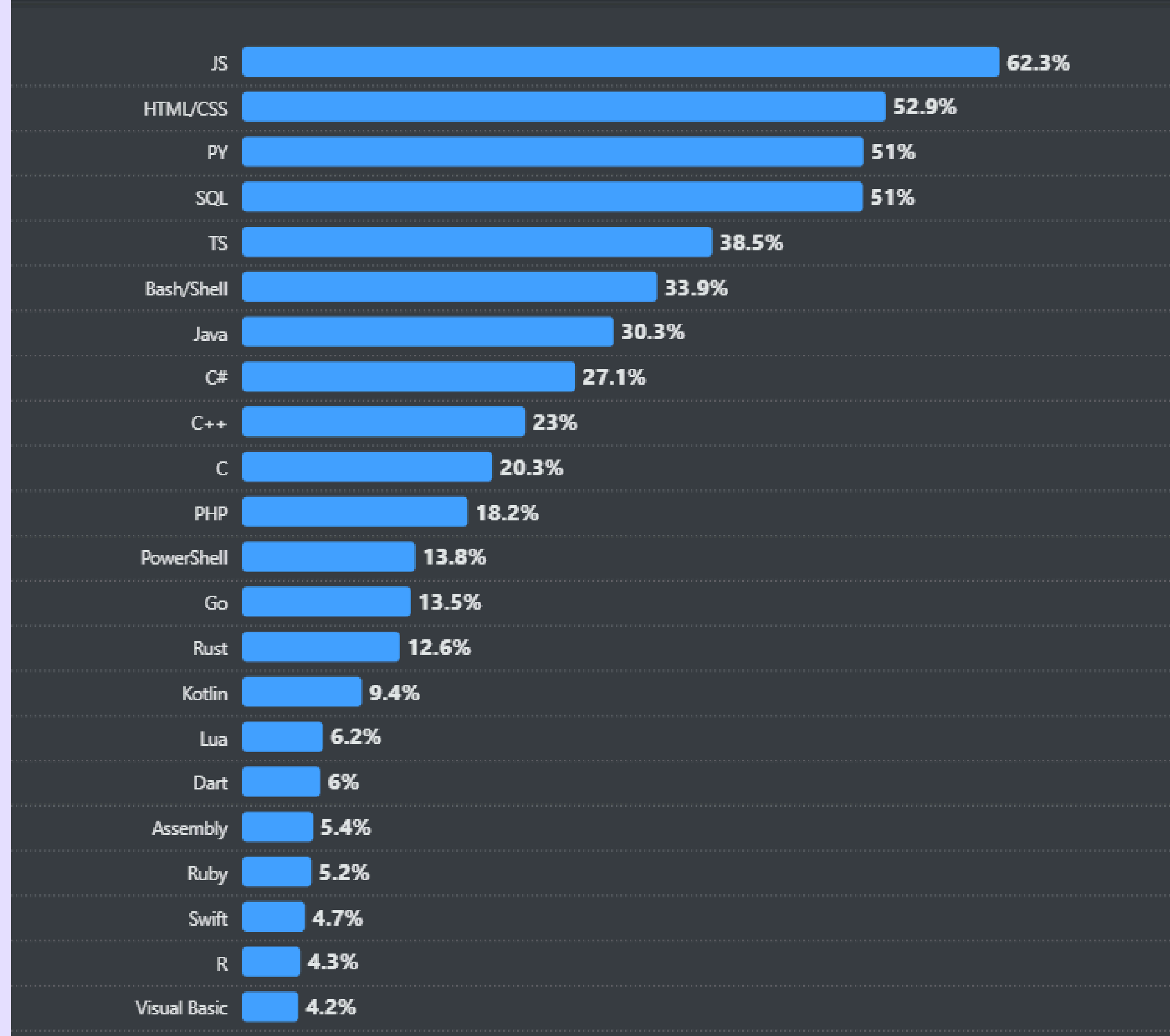
Ominaisuus	JavaScript	Java	C/C++	Python	PHP
Paradigma	Moniparadigma: tapahtumapohjainen, funktionaalinen	Oliopohjainen, luokkiin perustuva	Proseduraalinen, oliopohjainen (C++)	Moniparadigma: oliopohjainen, proseduraalinen	Palvelinpuolen skriptikieli
Tyypitys	Dynaaminen	Staattinen	Staattinen	Dynaaminen	Dynaaminen
Suoritusympäristö	Selain, palvelin (Node.js)	Java-virtuaalikone (JVM)	Järjestelmätasolla, itsenäiset sovellukset	Palvelinpuoli, itsenäiset sovellukset	Palvelinpuoli (verkkokehitys)
Syntaksi	Prototyyppipohjainen, vähemmän muodollinen	Tiukka, sanallinen	Matalan tason, sanallinen	Helppolukuinen, vähemmän muodollinen	Upotettu HTML:ään
Pääasiallinen käyttö	Verkkokehitys, reaaliaikaiset sovellukset	Yrityssovellukset, Android-sovellukset	Järjestelmä-/ohjelmist okehitys	Datatiede, verkkokehitys, automaatio	Verkkokehitys

Asiakas-palvelin -malli



Suosituimmat ohjelmointi-, skriptaus- ja merkinäkielet 2024

<https://survey.stackoverflow.co/2024/technology>



ECMAScript

JavaScriptin keksi Brendan Eich vuonna 1995. Se kehitettiin alun perin Netscape 2 -selainta varten. Kun Netscape siirsi JavaScriptin ECMA-järjestölle, Mozilla-säätiö jatkoi JavaScriptin kehitystä Firefox-selaimelle.

JavaScriptin standardia käytetään yleisesti asiakaspuolen (client-side) skriptaukseen.

Vuosi	Version nimi	Kuvaus
1997	ECMAScript 1	Ensimmäinen julkaisu, joka määritteli JavaScriptin ydintoiminnot.
1998	ECMAScript 2	Pieniä tarkennuksia ja virheidenkorjauksia ISO/IEC 16262 -standardin mukaisesti.
1999	ECMAScript 3	Säännölliset lausekkeet, parempi merkkijonokäsittely, try-catch-virheenkäsittely.
2009	ES5	Strict-tila, JSON-tuki, parannettu oliomäärittely.
2015	ES6 / ES2015	let/const, nuolifunktiot, luokat, moduulit, lupaukset (promises), mallipohjaiset merkkijonot.
2016	ES7 / ES2016	Potenssioperaattori (**), Array.prototype.includes().
2017	ES8 / ES2017	async/await, Object.entries(), Object.values(), merkkijonojen täydennys.
2018	ES9 / ES2018	rest/spread-ominaisuudet, asynkroninen iteraatio, Promise.finally().
2019	ES10 / ES2019	Array.flat(), Array.flatMap(), Object.fromEntries(), valinnainen catch-sidos.
2020	ES11 / ES2020	BigInt, dynaaminen import(), nulliyhdistämisoperaattori (??), valinnainen ketjutus (?).
2021	ES12 / ES2021	Loogiset sijoitusoperaattorit, replaceAll(), Promise.any().
2022	ES13 / ES2022	Ylätason await, uusia taulukkometodeja, parannetut yksityiset kentät luokissa.
2023	ES14 / ES2023	Array.prototype.toSorted(), Array.prototype.with(), Symbol.prototype.description.

Voit lukea lisää:

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>