Meilenstein 2

# 2.2

- AI Assistent

Wir benutzen Github Copilot Chat. Man kann ihm den Code markieren und er antwortet auf die Frage. Dabei nutzt er die Variablennamen und kennt alle Eigenschaften des Codes. Man kann damit Zeit einsparen, da man meist eine sehr gute Hilfestellung bekommt. Manchmal muss man nur etwas selbst konfigurieren.

* Database

Hier kann man vieles sehen ohne zu coden. Man kann einfach die Struktur anklicken. Falls man doch mal was genauer sehen kann, wird eine Query erstellt, in der man dann mit SQL-Abfragen stellen kann. Dadurch, dass man alles direkt auf einen Blick hat und nicht alles per SQl abfragen muss, kann man viel Zeit sparen.

* Darstellung

Egal worüber man hovert, man bekommt immer sehr viele Informationen. Diese Informationen helfen, um dann Bsp.: mehr über die Parameter einer Datei zu erfahren, zu den Funktionen zu springen. Somit ist schneller klar, wie man diese dann genau nutzen muss.

* Fehlermeldung (Laravel)

Ist sehr gut und zeigt genau an, wo der Fehler ist. Leichtes Debugging.

# 2.3

## A

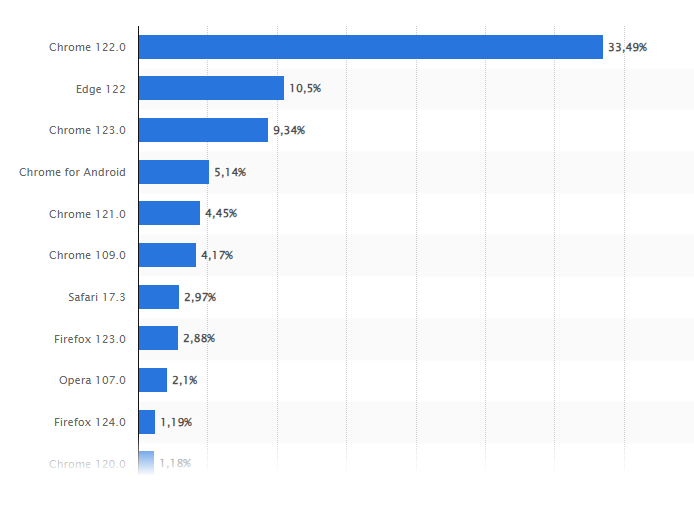
Die am häufigsten verwendeten Webbrowser auf Desktops sind in der Regel Google Chrome, gefolgt von Mozilla Firefox und Microsoft Edge. Auf mobilen Geräten dominieren oft mobile Versionen von Chrome, Safari und Firefox.

Google Chrome: 48,9 %

Mozilla Firefox: 22,3 %

Apple Safari: 9,9 %

Microsoft Edge: 5,8 %

Opera: 1,61 %

Stand vom Sep.2017

Man kann nicht direkt sagen, was genau das Beste ist.

## B

Ich mag die Funktion bei Edge: Arbeitsbereich. Zudem gibt es auf der rechten Seite Apps, die man bei Edge hinzufügen kann. So muss man den Browser nicht verlassen.

## C

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

CANIUSE.com

The caniuse browser scores are tallies of all features tracked on caniuse (excluding those marked as "unofficial"). The fully opaque part represents supported features, the semi-transparent part represents partial support.

Note that while caniuse tracks a wide variety of features, it only covers a subset of all web technologies so the scores are not 100% representative of any browser's capabilities.

## D

Ich werde weiterhin Edge nutzen, da es mein Standardbrowser ist. Ich vermute aber, da Chrome die höchste Prozentzahl hat, auch am besten geeignet ist. Zudem wissen wir nicht genau, was auf uns zu kommt, weshalb wir nur sehr grob recherchieren können.

1. Chrome
2. Firefox
3. Safari

Amier=> Edge

Kacper (vermutlich) = Opera

# 2.4

Variablen: Beide Sprachen unterstützen die Verwendung von Variablen zur Speicherung von Daten. Zum Beispiel:

javascript  
var x = 10; // JavaScript

C++:  
int x = 10; // C++

Bedingte Anweisungen: Sowohl JavaScript als auch andere Sprachen ermöglichen die Verwendung von bedingten Anweisungen wie if-else. Zum Beispiel:

javascript  
if (x > 5) {  
 // do something  
} else {  
 // do something else  
}

PHP  
if ($x > 5) {  
// do something  
} else {  
 // do something else  
}

Schleifen: Beide Sprachen bieten Schleifenkonstrukte wie for und while. Zum Beispiel:

javascript  
for (var i = 0; i < 5; i++) {  
 // do something  
}

C++:  
for (int i = 0; i < 5; i++) {  
 // do something  
}

Funktionen: Sowohl JavaScript als auch andere Sprachen unterstützen die Definition und den Aufruf von Funktionen. Zum Beispiel:

javascript  
function greet(name) {  
 return "Hello, " + name + "!";  
}

PHP:  
function greet($name) {  
 return "Hello, " . $name . "!";  
}

Klassen und Objekte: Moderne Versionen von JavaScript unterstützen die Definition von Klassen und die Erstellung von Objekten, ähnlich wie in anderen objektorientierten Sprachen wie C++. Zum Beispiel:

javascript

class Car {

constructor(make, model) {  
 this.make = make;  
 this.model = model;  
 }  
}

var myCar = new Car("Toyota", "Camry");

C++:  
class Car {  
public:  
 string make;  
 string model;  
};

Car myCar;  
myCar.make = "Toyota";  
myCar.model = "Camry";

Arithmetische Operatoren: Beide Sprachen unterstützen gängige arithmetische Operatoren wie Addition, Subtraktion, Multiplikation und Division. Zum Beispiel:

javascript  
var result = 10 + 5; // 15

PHP:  
$result = 10 + 5; // 15

Hier sind sechs Unterschiede zwischen JavaScript und einer anderen bekannten Programmiersprache:

Typisierung: JavaScript ist dynamisch typisiert, während Sprachen wie C/C++ statisch typisiert sind. Das bedeutet, dass in JavaScript Variablen keine expliziten Typen haben und sich während der Laufzeit ändern können, während in C/C++ Variablen einen festen Typ haben. Zum Beispiel

javascript  
var x = 10; // x ist eine Zahl  
x = "Hello"; // x ist jetzt ein String

C++:  
int x = 10; // x ist immer eine Ganzzahl  
// x = "Hello"; // Das würde zu einem Fehler führen

Kompilierung vs. Interpretation: C/C++-Code wird normalerweise vor der Ausführung in Maschinencode kompiliert, während JavaScript-Code zur Laufzeit von einem Interpreter interpretiert wird.

Syntax: Die Syntax von JavaScript unterscheidet sich von der von C/C++ oder PHP. Zum Beispiel die Verwendung von Semikolons am Ende von Anweisungen (optional in JavaScript, erforderlich in C/C++), Blockstrukturen usw.

Speicherbehandlung: In Sprachen wie C/C++ müssen Entwickler die Speicherallokation und -freigabe explizit verwalten (z. B. durch Verwendung von malloc() und free()), während JavaScript eine automatische Speicherbereinigung durch einen Garbage Collector bietet.

Asynchrone Programmierung: JavaScript unterstützt asynchrone Programmierung durch Funktionen wie Promises und async/await, während traditionelle Sprachen wie C/C++ asynchrone Operationen normalerweise durch Threads oder Callbacks implementieren.

Plattformabhängigkeit: JavaScript ist eine plattformunabhängige Sprache, die in verschiedenen Umgebungen wie Webbrowsern und Servern ausgeführt werden kann. C/C++-Code hingegen ist in der Regel plattformspezifisch und muss für verschiedene Plattformen kompiliert werden.