

Введение в JavaScript

Основы лучшего языка для фронтенда

Кирилл Талецкий

TeachMeSkills
27 июля 2023

Что будет

- Введение
 - Что такое JavaScript
 - Как он работает
- Переменные
- Встроенные функции браузера
- **Перерыв**
- Типы данных
- Преобразование типов
- Обсуждение ДЗ

Чего мы хотим добиться

- Знать основные принципы работы JS
- Знать особенности создания переменных
- Знать какими типами данных мы можем распоряжаться
- Знать как работает приведение типов
- Знать и уметь пользоваться встроенными функциями браузера

Введение

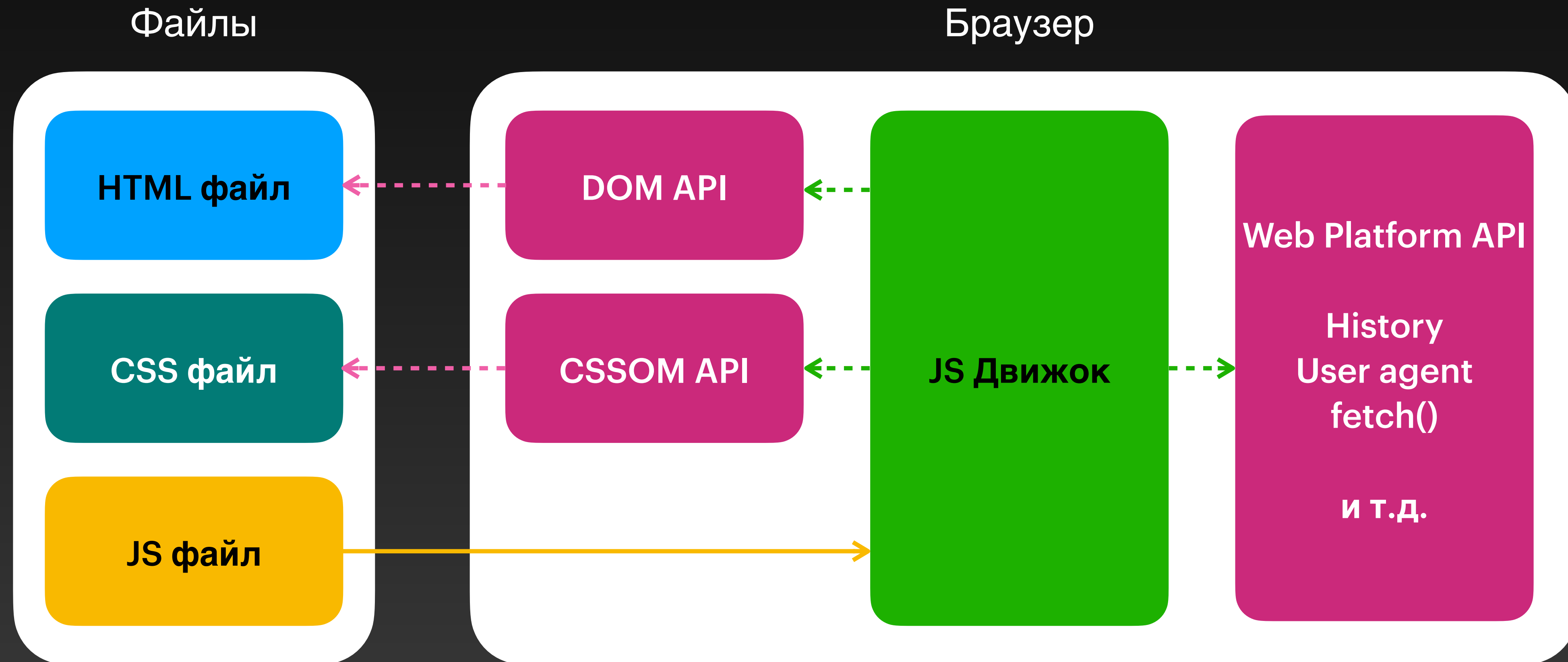
Что такое JS

- Язык программирования
- Начинаясь в 1995 как инструмент для оживления веб страниц
- Сейчас один из самых популярных языков программирования
- Ныне используется на разных платформах - от браузеров, до нативных приложений под iOS
- JS очень активно развивается
- Сейчас это доминирующий язык для браузеров

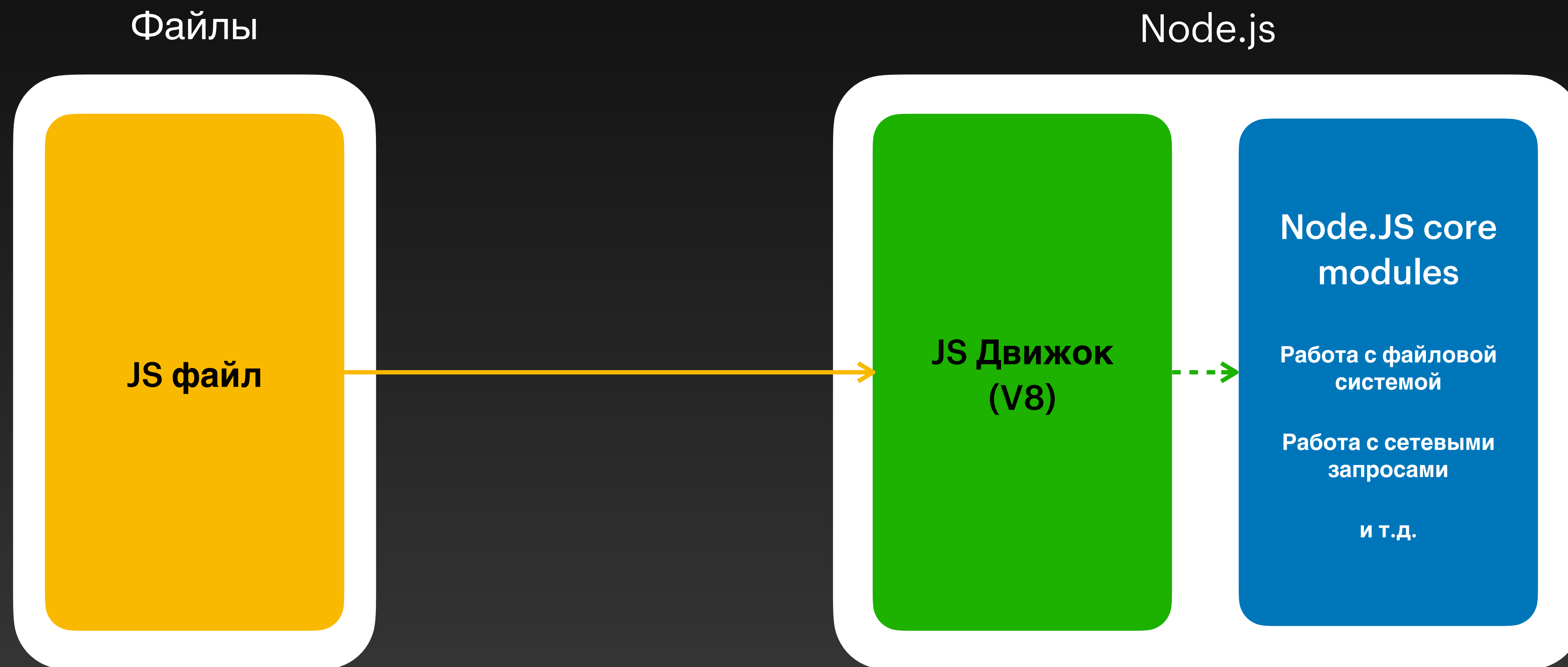
Факты про JS

- Безопасный язык - нет доступа к памяти и процессору
- Типизация
 - Динамическая
 - Слабая
- Интерпретируемый язык - не нужна предварительная компиляция, код исполняется “как есть”, построчно (но есть нюансы)

Как работает JS в вебе



Как работает JS отдельно (Node.js)



Где можно протестить JS код

Демо среда

- Dev Tools браузера -> Console

Стандарты JS

- Официальная спецификация ECMA-262
 - Каждый год выходит новый стандарт
- Функционал, который вот-вот войдёт в стандарт
 - Stage 3 proposals
- Mozilla Developer Network - можно сказать, более удобочитаемая версия официального стандарта

Переменные

Как создавать переменные в RHR?

Как создавать переменные в JS

- Переменные в JS объявляются через ключевое слово ``let``
- Присваивание значения происходит через оператор ``=``
- Возможно использование устаревшего ключевого слова ``var``
 - У него есть нюансы использования, о них мы поговорим после того как поработаем с функциями

Как создавать переменные в JS

const

- Неизменяемые значения в JS объявляются через ключевое слово ``const``
 - Позволяет создавать гарантировано неизменные значения
- Глобальные константы принято называть используя верхний регистр

Как создавать переменные в JS

Имена переменных

- Имя переменной должно содержать только буквы, цифры или символы `\$` и `_`
- Первый символ не должен быть цифрой

- Можно

```
const _keko$ = 0;  
const $ = 'lol';  
const юххууу = 42; // сработает, но можно не надо
```

- Нельзя

```
const lol+kek = 'lolkek'; // недопустимый символ  
const 123foo = 'bar'; // имя начинается с цифры  
const return = 'baz'; // `return` – зарезервированное слово
```

Встроенные функции

**Куда и как я могу вывести содержимое
переменной в РНР?**

Встроенные функции

- Вызов функции - имя функции и скобки (в скобках могут быть аргументы)
- Встроенные функции взаимодействия с пользователем
 - `alert(message)`
 - `prompt(message)`
 - `confirm(message)`
- Встроенная функция вывода в консоль — `console.log(value1, ..., valueN)`

Перерыв

Типы данных

Какие типы данных есть в RНР?

Типы данных в JS

- Число
- BigInt
- Строка
- Булев (логический тип)
- null
- undefined
- Объект
- Symbol

Числа

- Поддерживаются как целые, так и числа с плавающей точкой
- Существуют специальные числовые значения
 - Infinity / -Infinity
 - NaN
- Математические операции безопасны
 - Можно делить на ноль, добавлять в выражения любые строки и т.д.
 - Ваш скрипт от этого никогда не упадёт
 - В худшем случае, вы получите NaN как результат вычислений

BigInt

- JS плохо работает с числами вне диапазона

$$\pm 2^{53}-1 = \pm 9007199254740991$$

- Для работы с большими числами в JS введён тип BigInt

Строка

- Строка в JS — последовательность символов внутри кавычек
 - Кавычки могут быть одинарные, двойные и обратные
- Обратные кавычки позволяют вставлять переменные или выражения в строку

Булевый тип

- Только два значения - ложь (false) и истина (true)
- Булевым типом обычно является результат сравнений

Как в PHP создать пустую переменную?

Null и Undefined

- Null — специальное значение, которое означает “ничего”, “пусто”, “значение не известно”
- Undefined — означает, что “значение не было присвоено”, то есть пустую переменную
- Оба типа содержат только одно одноименное значение.

Как в RNP создать набор ключ-значение?

Объекты и символы

- Объект — коллекция, набор пар ключ-значение
- Все изученные ранее типы — примитивные
- Объекты — составной тип, они могут хранить в себе значения других типов
- Символ — специальный тип, который нужен для создания уникальных ключей в объектах.
- С объектами и символами мы детально познакомимся в следующих уроках

Оператор typeof

- Позволяет определить, к какому типу относится значение или переменная

```
typeof 42 // "number"

typeof 10n // "bigint"

typeof "foo" // "string"

typeof true // "boolean"

typeof undefined // "undefined"

typeof null // "object" – это отдельный кек

typeof {foo: 'bar'} // "object"

typeof Symbol("id") // "symbol"
```

Приведение типов

**Могу ли я сначала создать число а затем
привести его к строке в PHP?**

Приведение типов

Приведение к строке

- Чаще всего происходит неявно, при использовании функций вывода таких как ``alert(value)`` или ``console.log(value)``
- Мы можем сделать явное преобразование используя ``String(value)``

Приведение типов

Приведение к числу

- Численное преобразование происходит неявно в математических функциях и выражениях
- Мы можем сделать явное преобразование используя `Number(value)`

value	Number(value)
true / false	1 / 0
null	0
undefined	NaN
string	Пробельные символы (пробелы, знаки табуляции \t, знаки новой строки \n и т. п.) по краям обрезаются. Далее, если остаётся пустая строка, то получаем 0, иначе из непустой строки «считывается» число. При ошибке результат NaN

**В PHP, если я попытаюсь привести строку
"0" к булевому типу, какой будет
результат?**

Приведение типов

Приведение к логическому типу

- Происходит неявно в логических операциях (например, `if (value) ...`)
- Явное преобразование — `Boolean(value)`

value	Boolean(value)
0, null, undefined, NaN, ""	false
Всё остальное	true

К чему будет приведено `Boolean("0")` в JS?

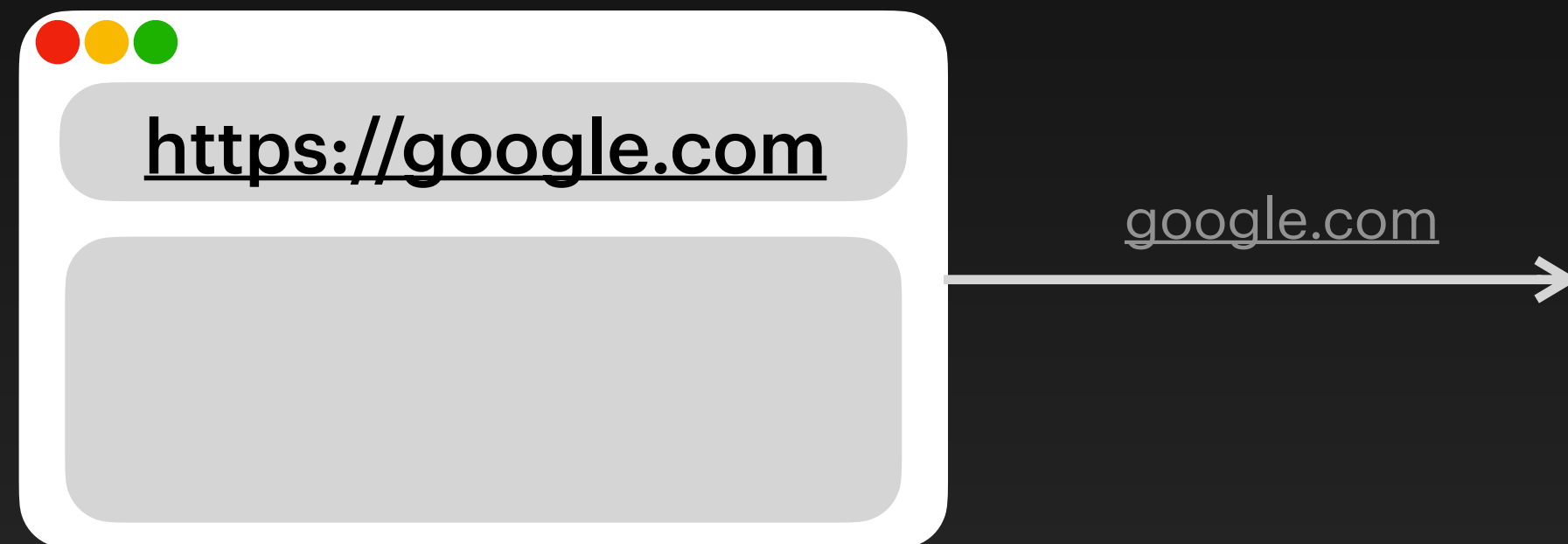
Дополнение

Нюансы исполнения JS кода

- Два прохода движка по файлу с кодом
 - Первый — сбор информации об объявлениях переменных, проверка синтаксиса
 - Второй — исполнение кода
- Как следствие - объявления переменных и функций в JS “всплывают” (hoisting)
 - В случае с `var`, к переменной можно обратиться до инициализации
 - Это устаревшее поведение, которое пофикшено в `let` и `const`
- Just in Time компиляция (JIT) - компиляция кода в процессе исполнения (в рантайме)
 - механизм оптимизации, при котором наиболее часто используемые части JS кода компилируются и хранятся в скомпилированном виде для более быстрого обращения к ним

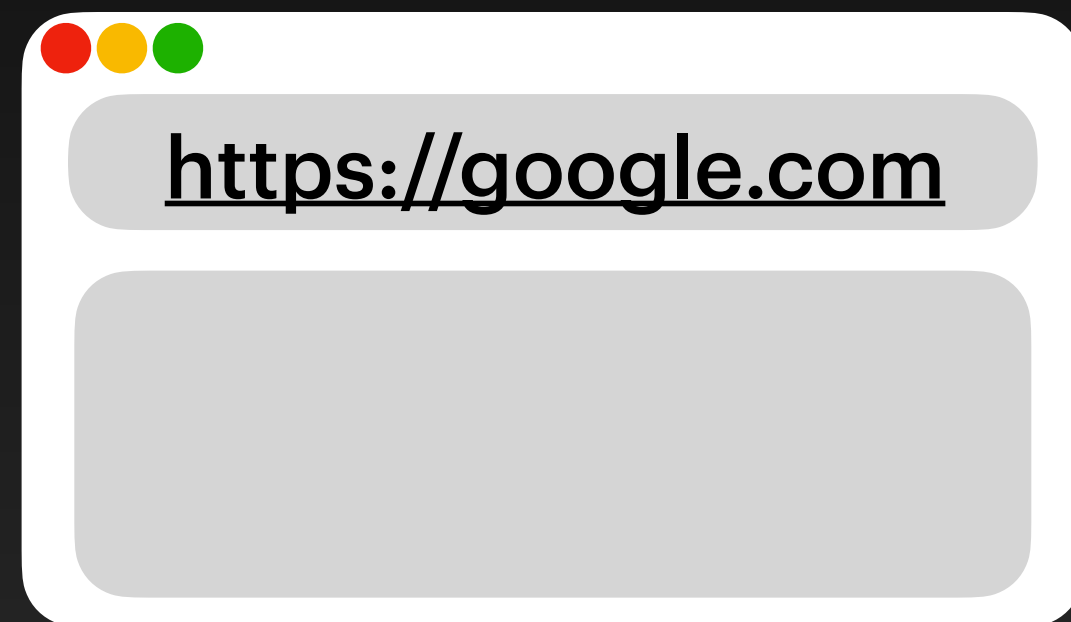
Как работает веб

Браузер



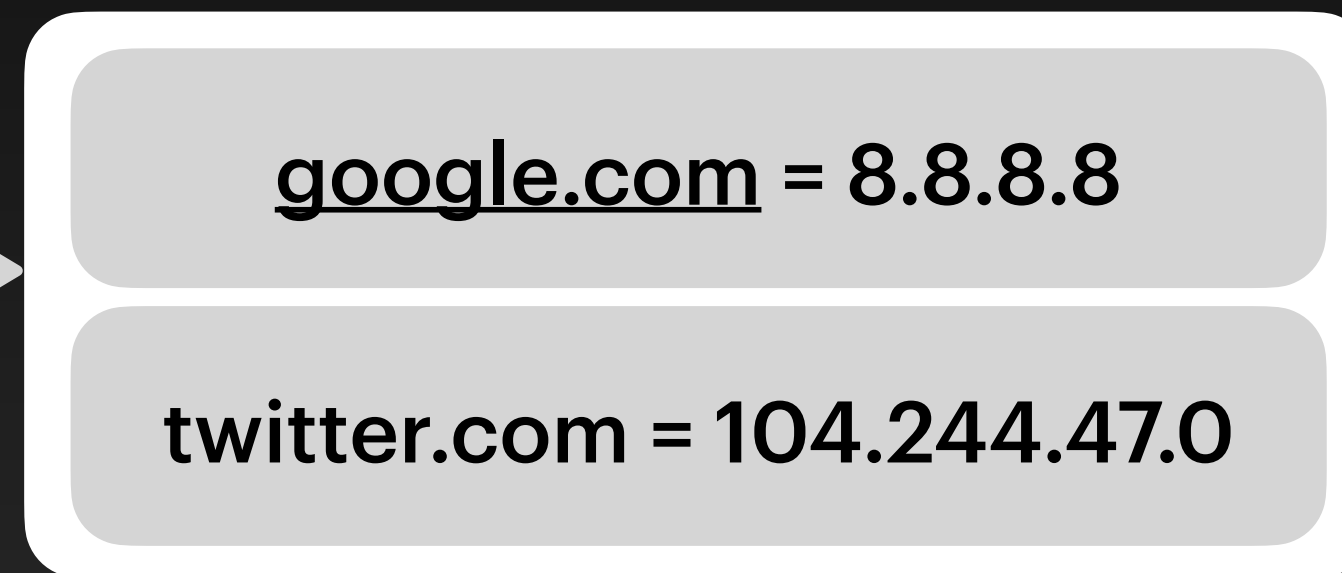
Как работает веб

Браузер



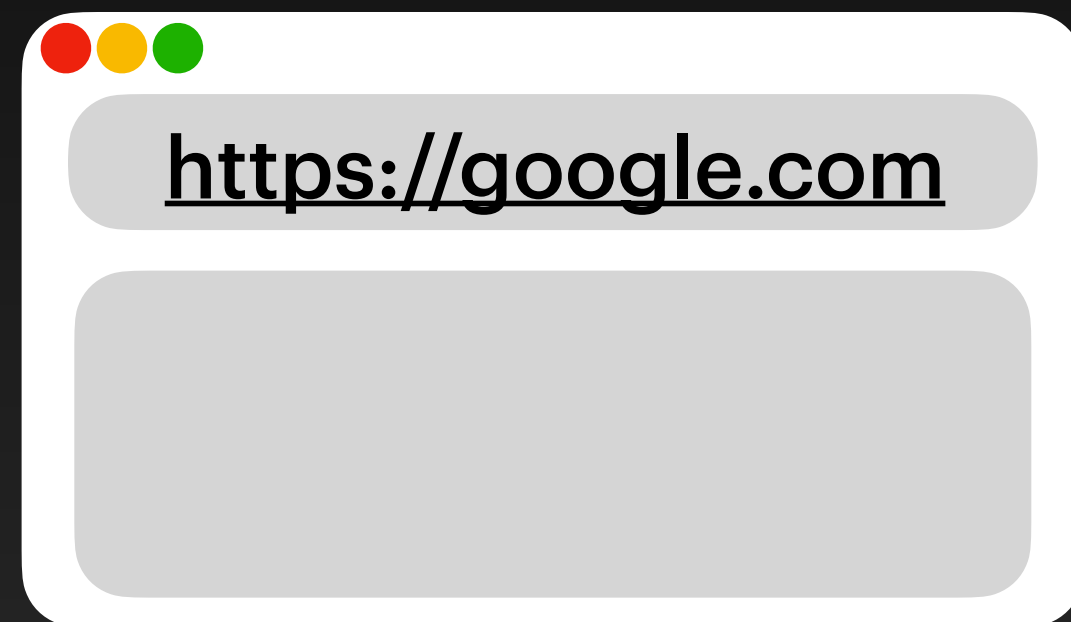
google.com

Domain Name Service
(DNS)

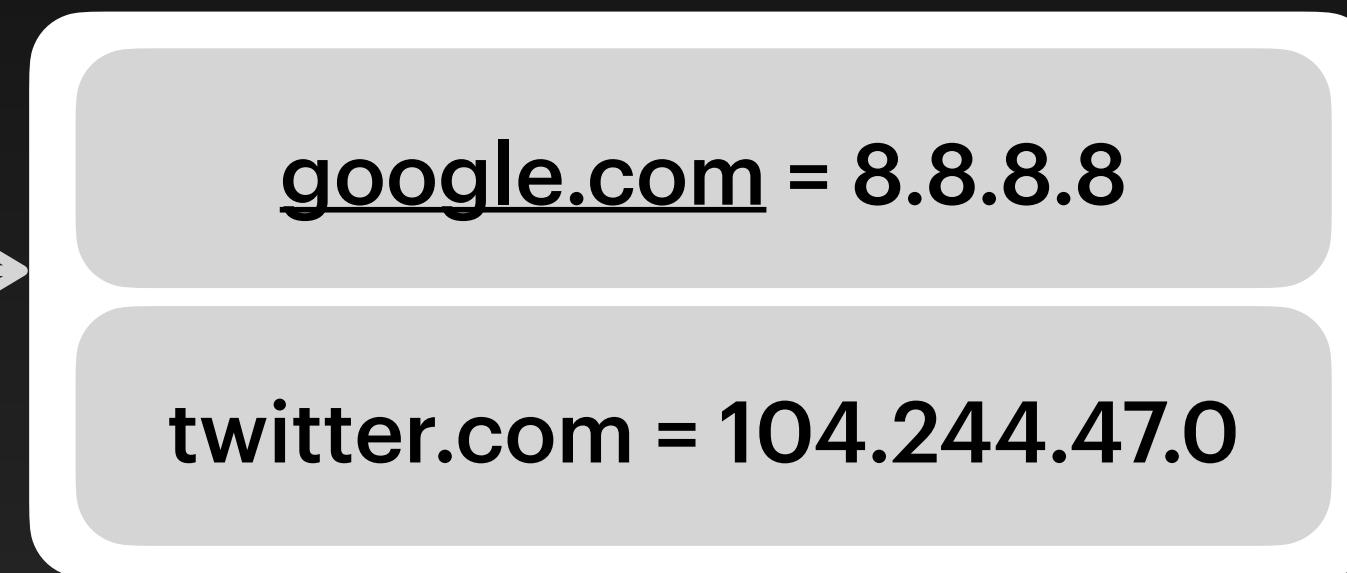


Как работает веб

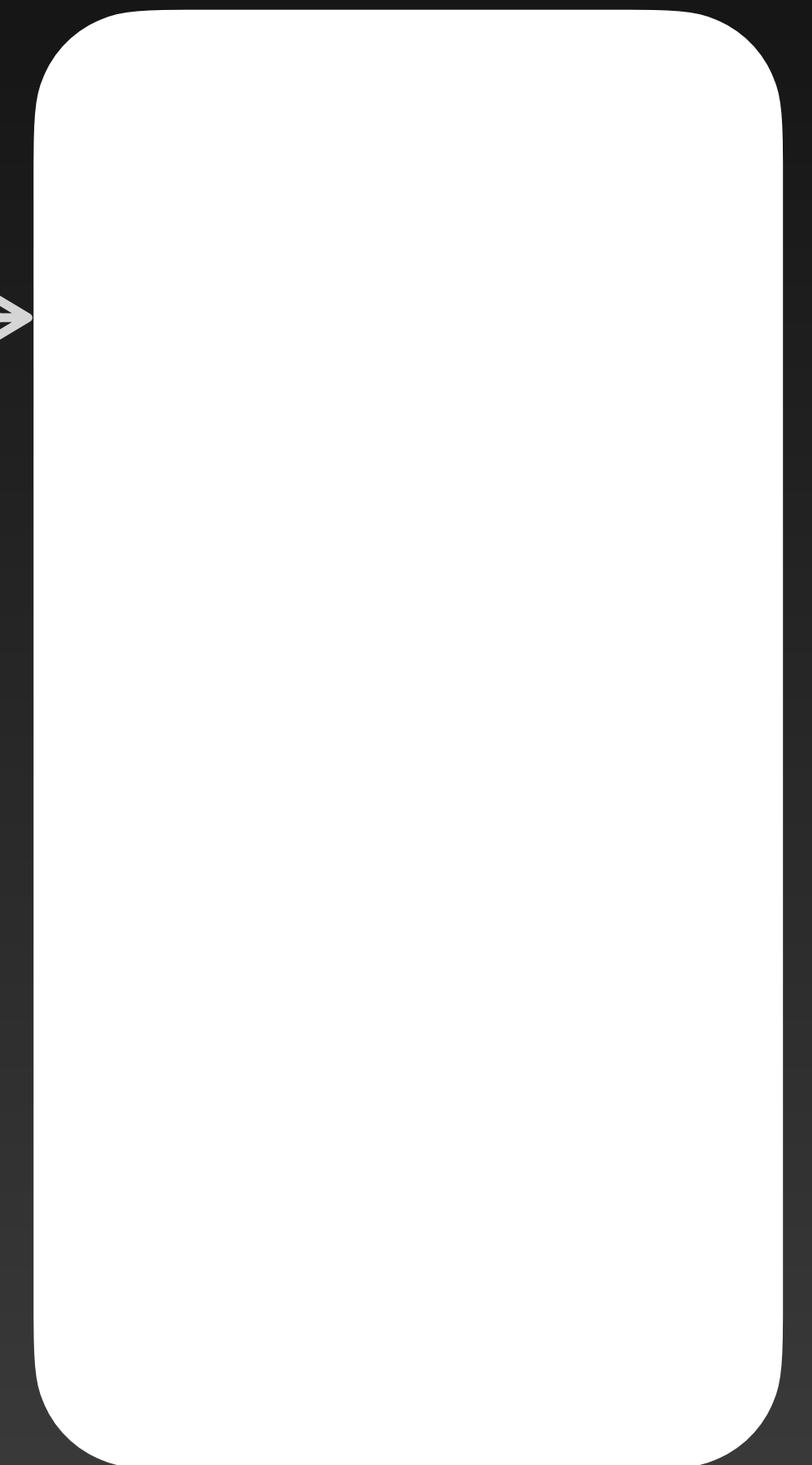
Браузер



Domain Name Service
(DNS)



Сервер гугла

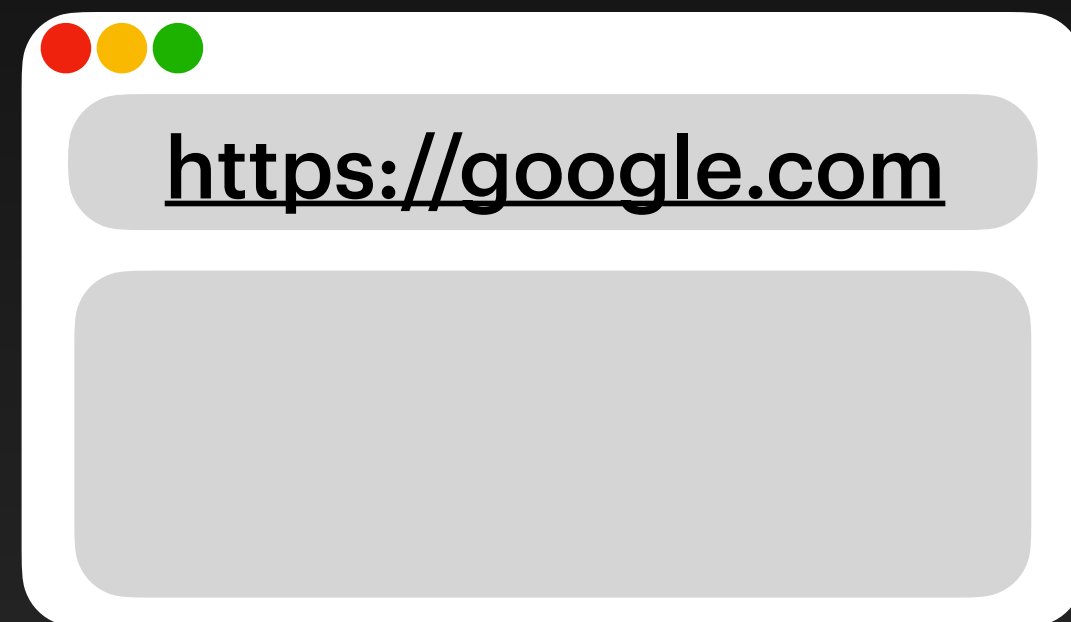


`google.com`

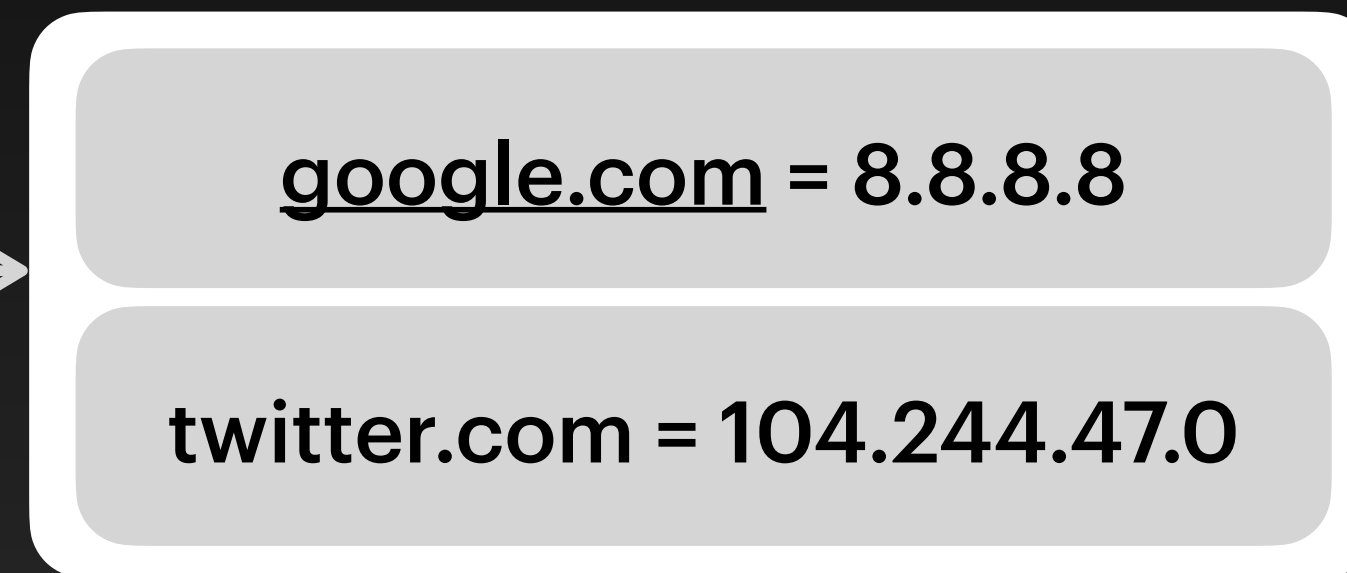
`8.8.8.8`

Как работает веб

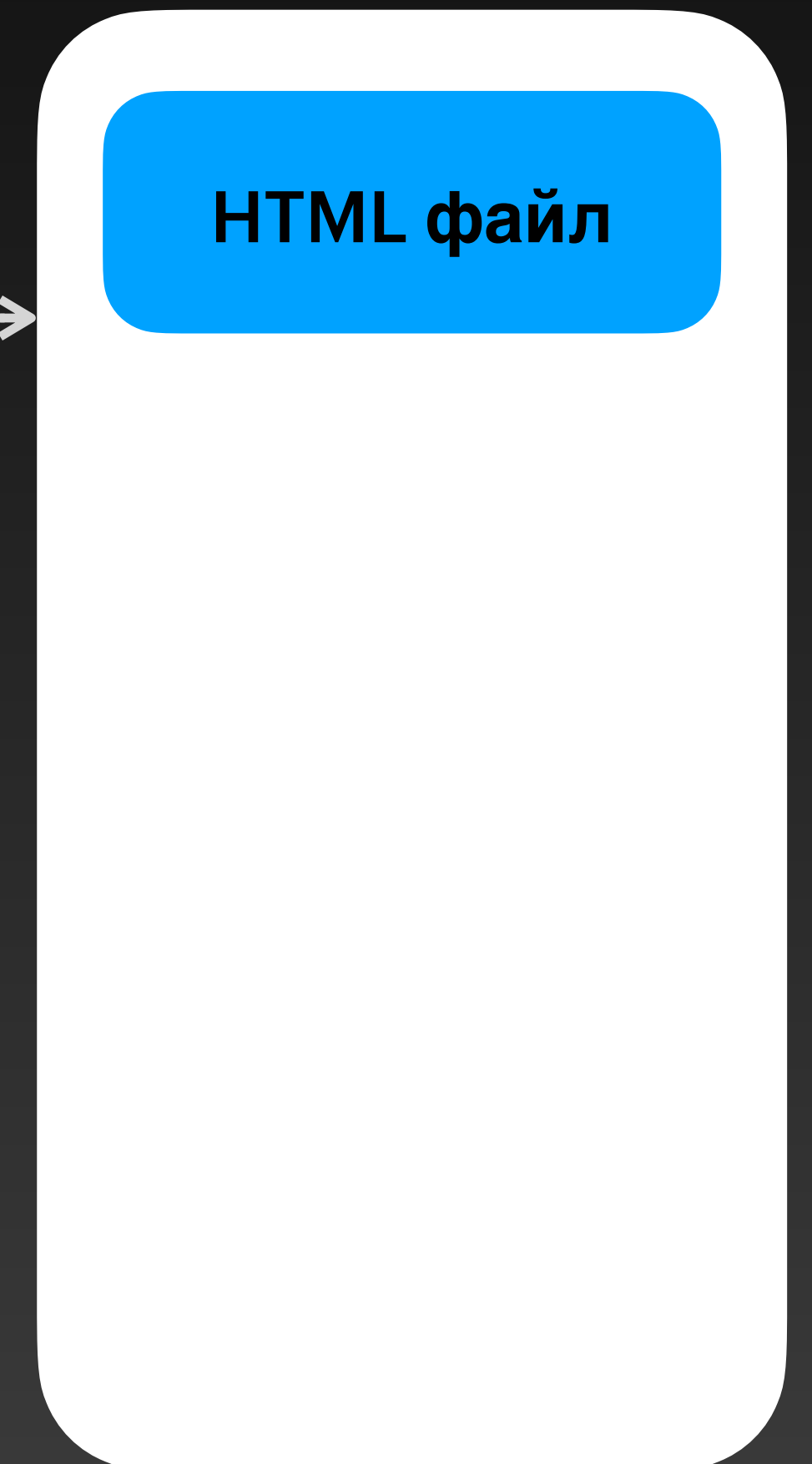
Браузер



Domain Name Service
(DNS)



Сервер гугла

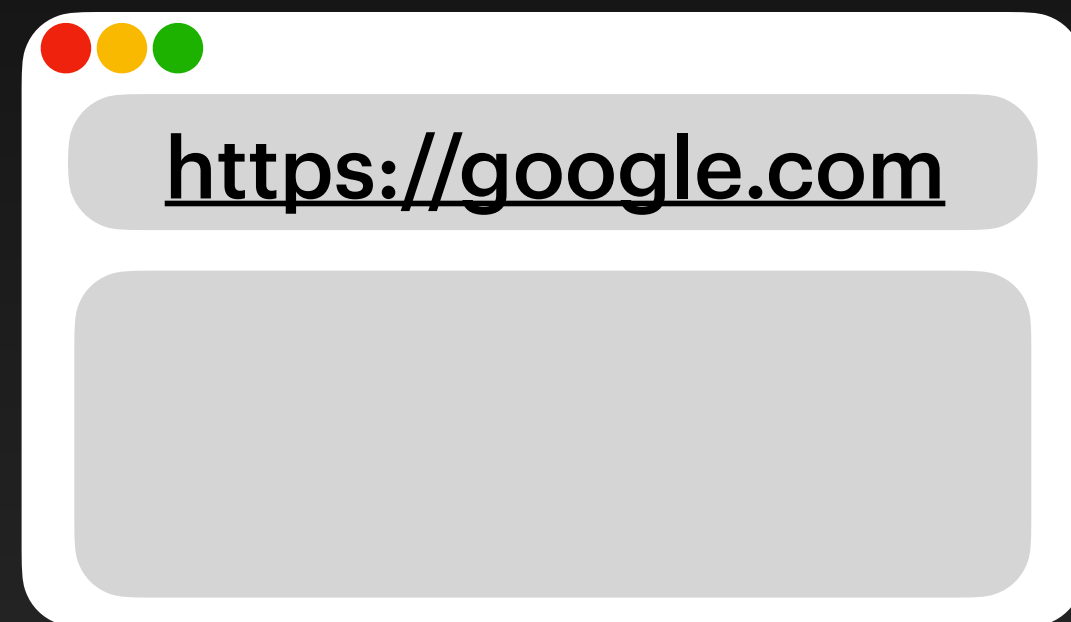


`google.com`

`8.8.8.8`

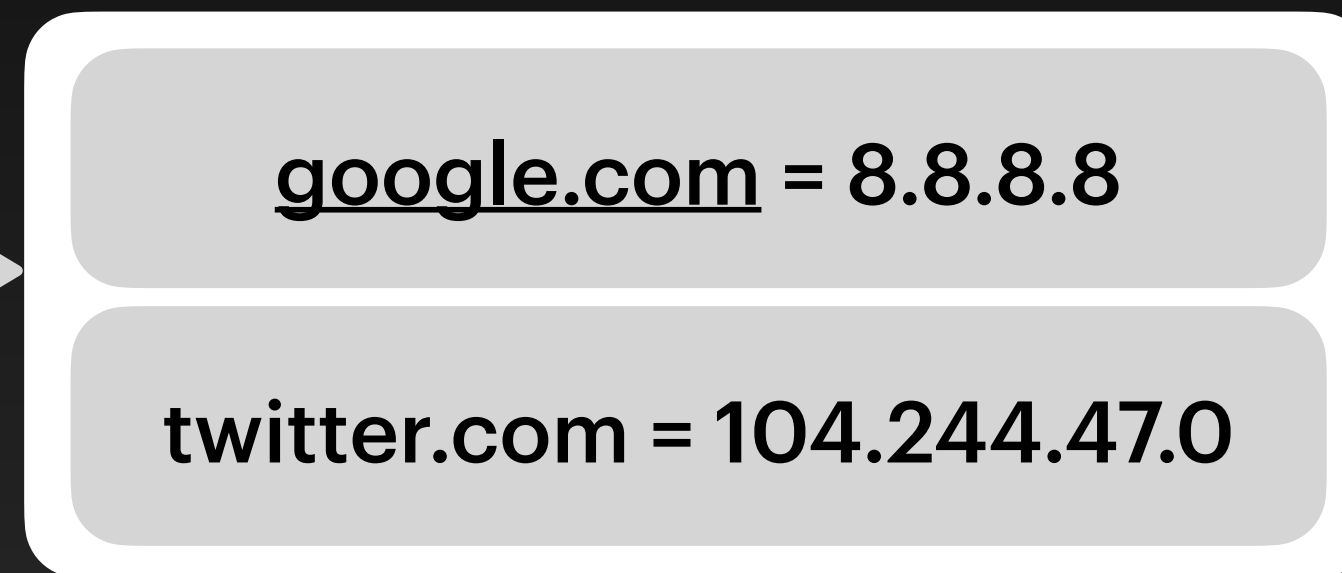
Как работает веб

Браузер



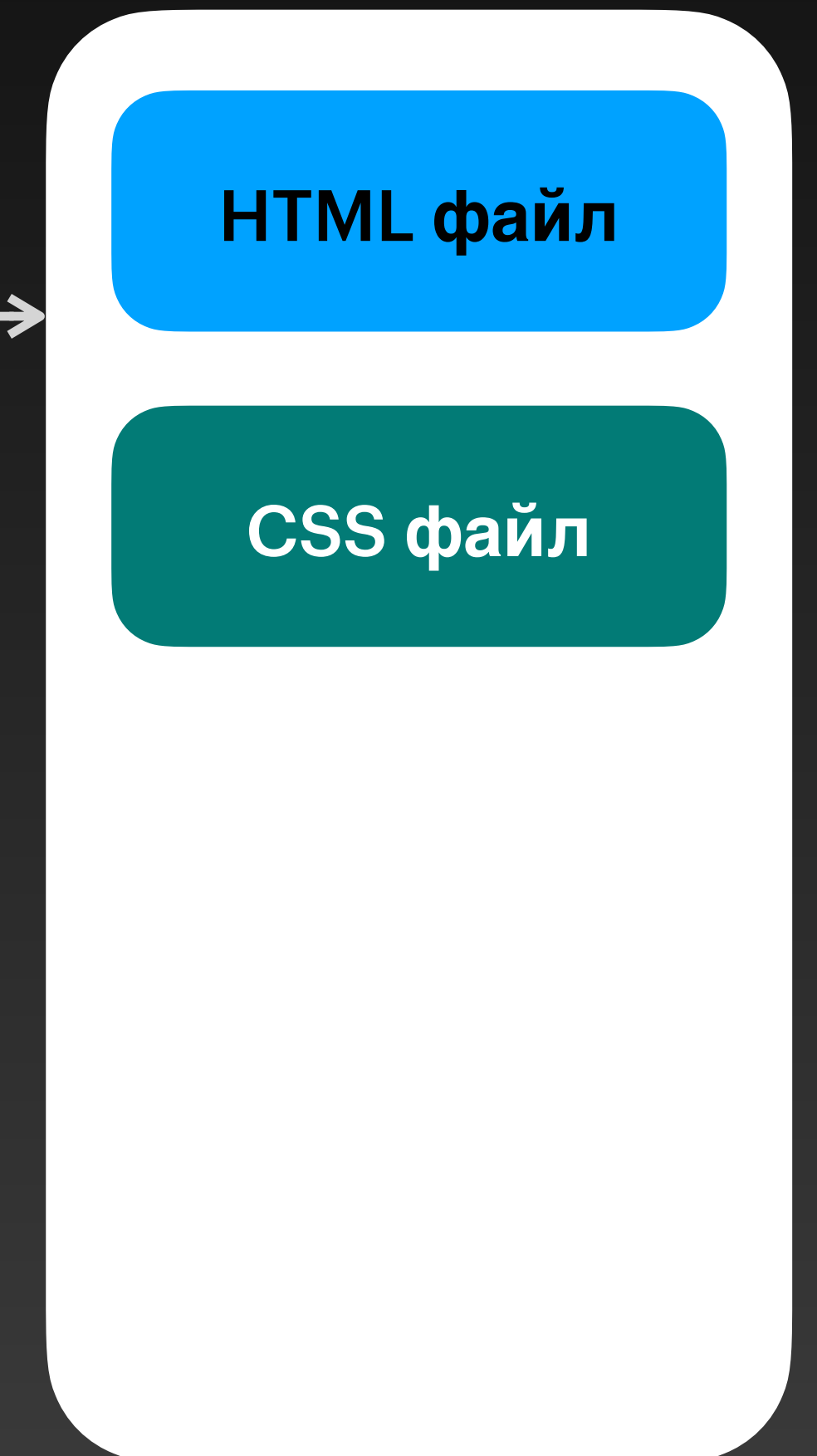
Domain Name Service
(DNS)

`google.com`



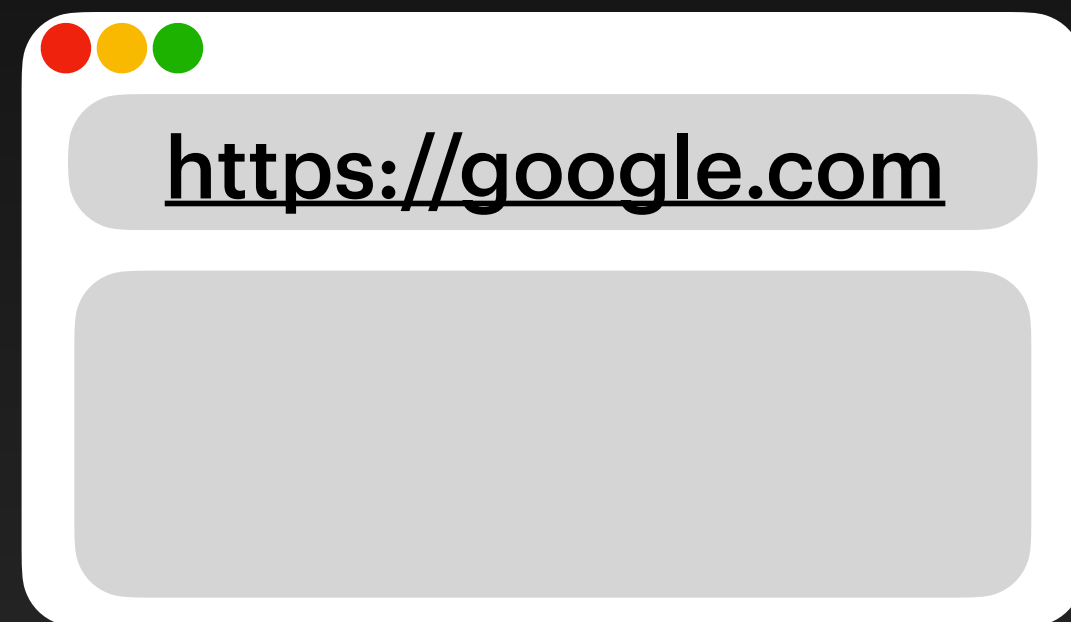
`8.8.8.8`

Сервер гугла



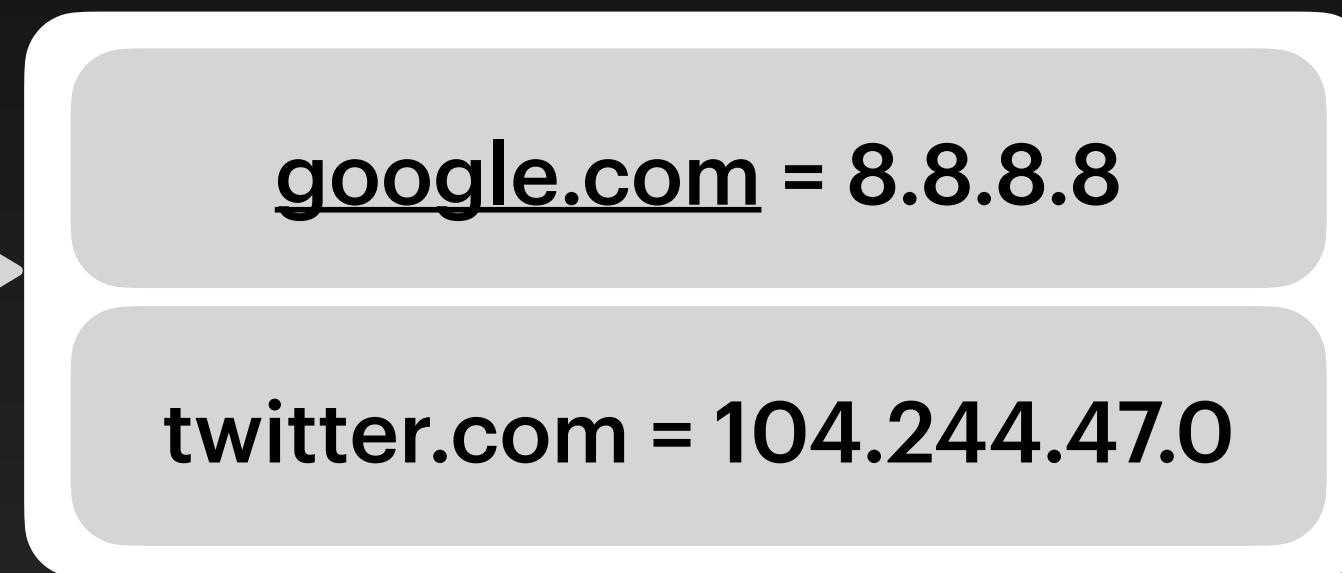
Как работает веб

Браузер



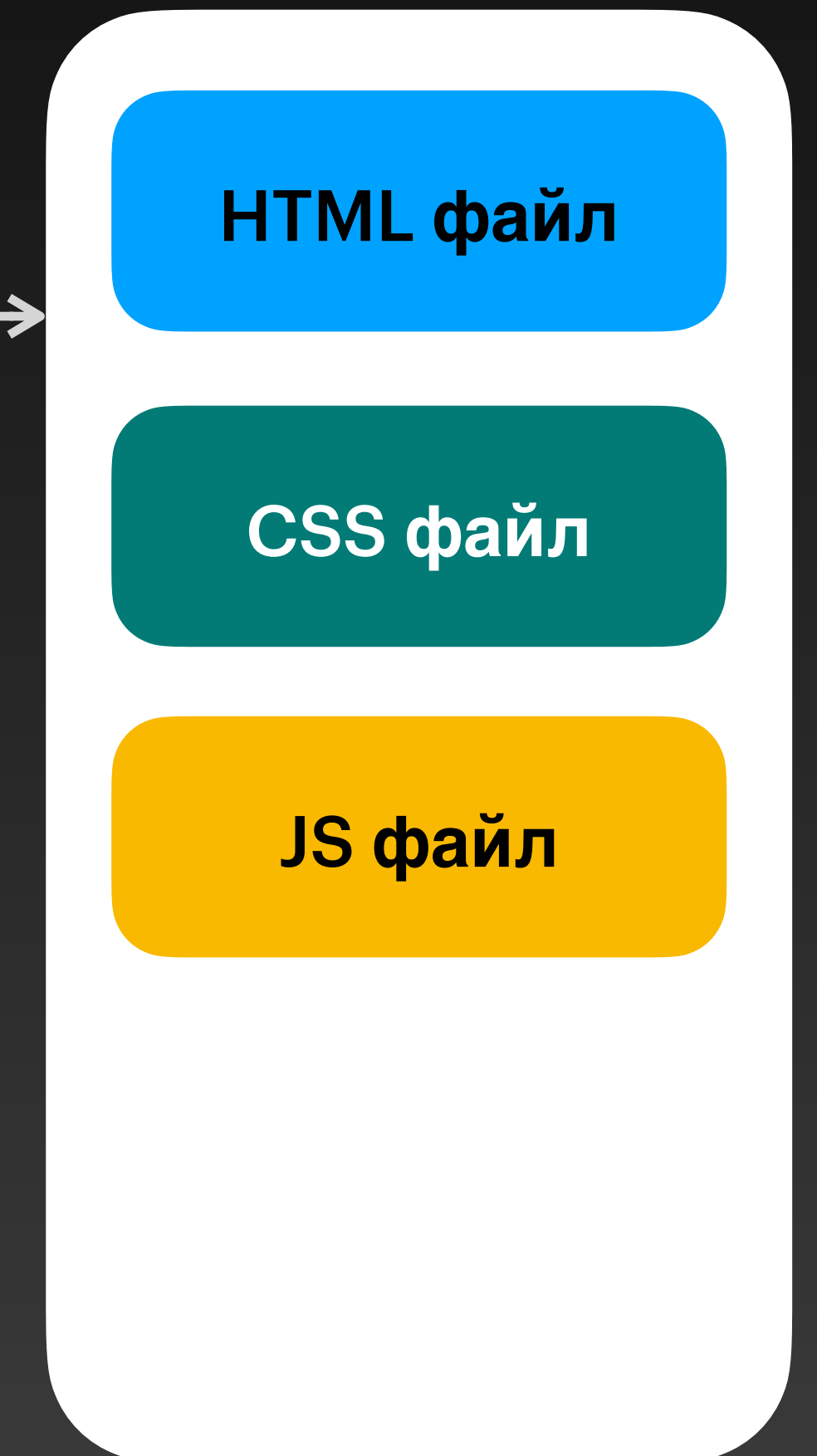
google.com

Domain Name Service
(DNS)



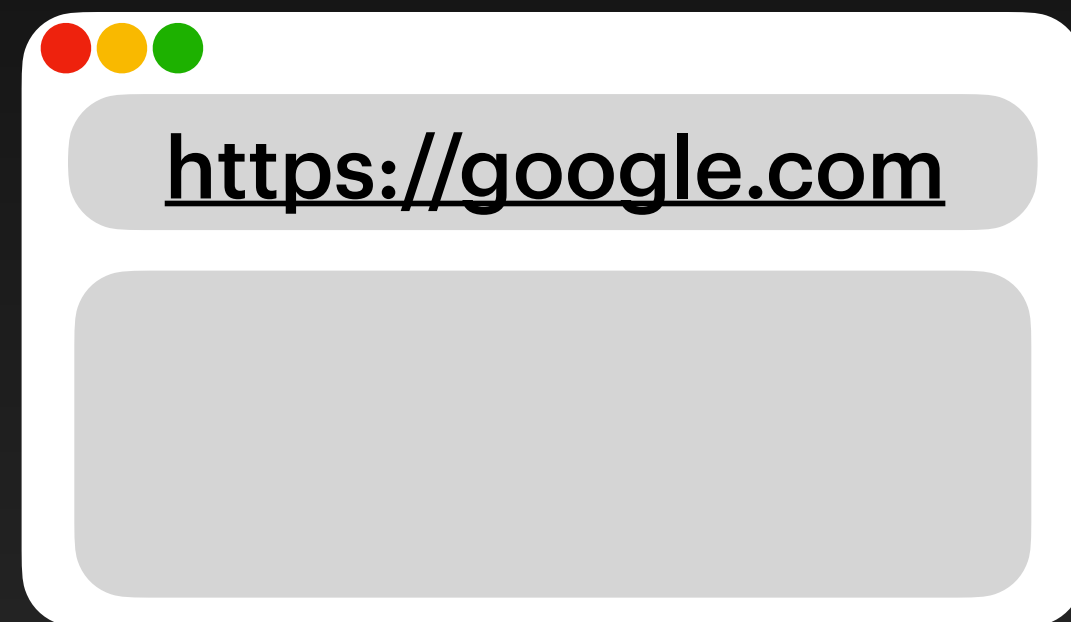
8.8.8.8

Сервер гугла



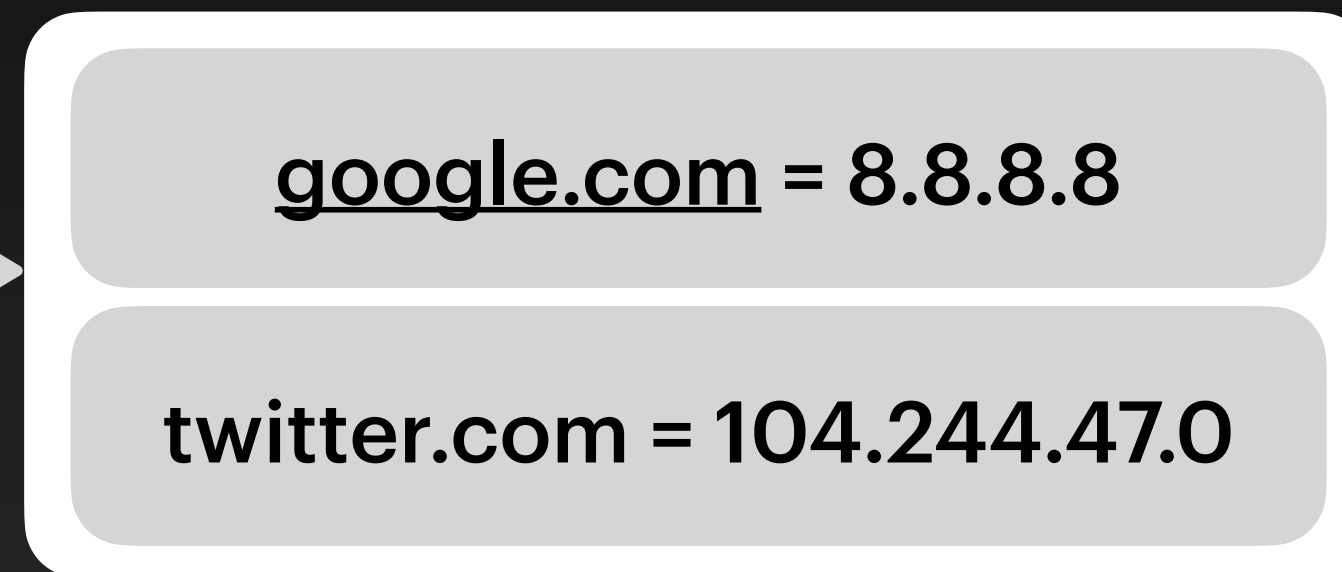
Как работает веб

Браузер



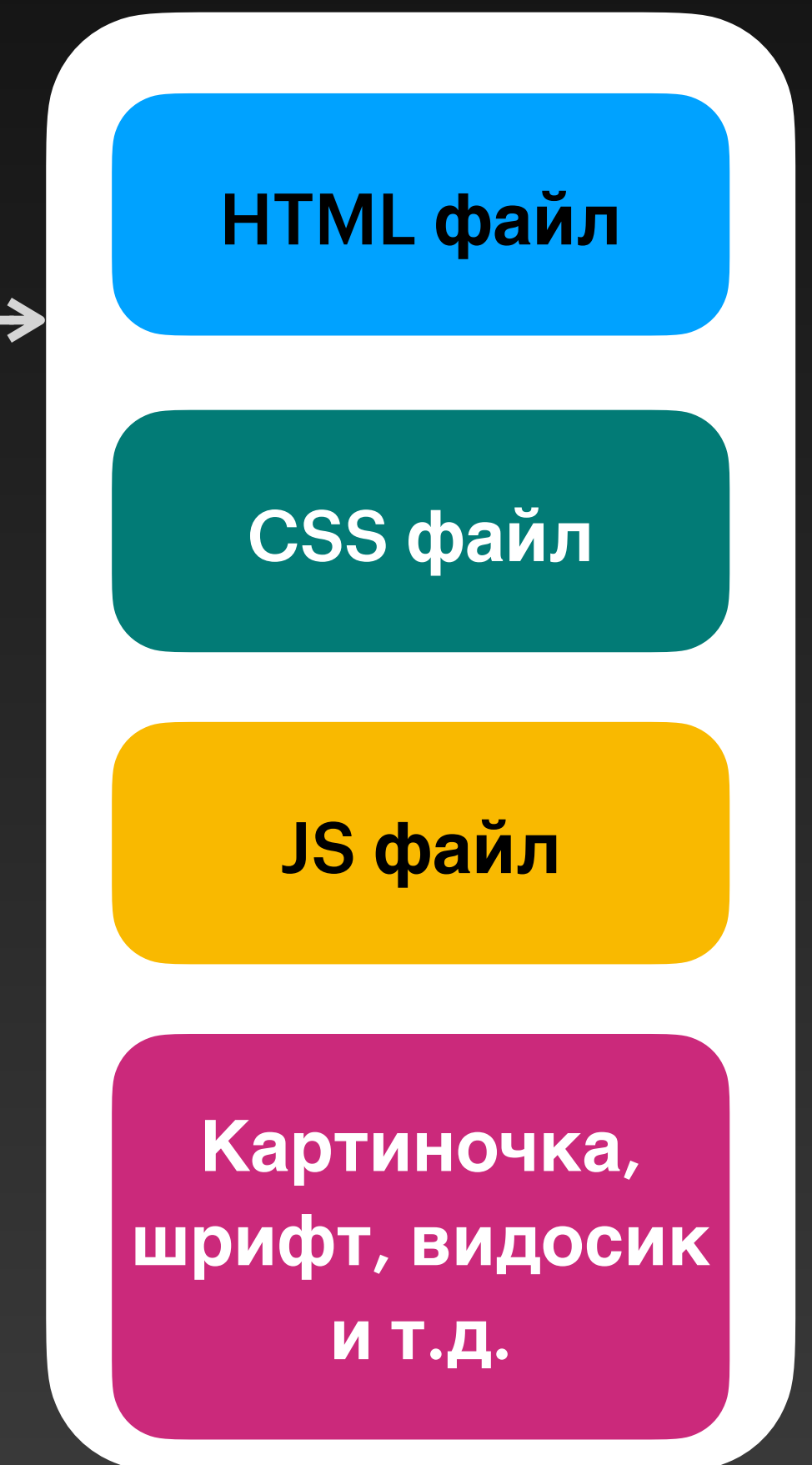
google.com

Domain Name Service
(DNS)

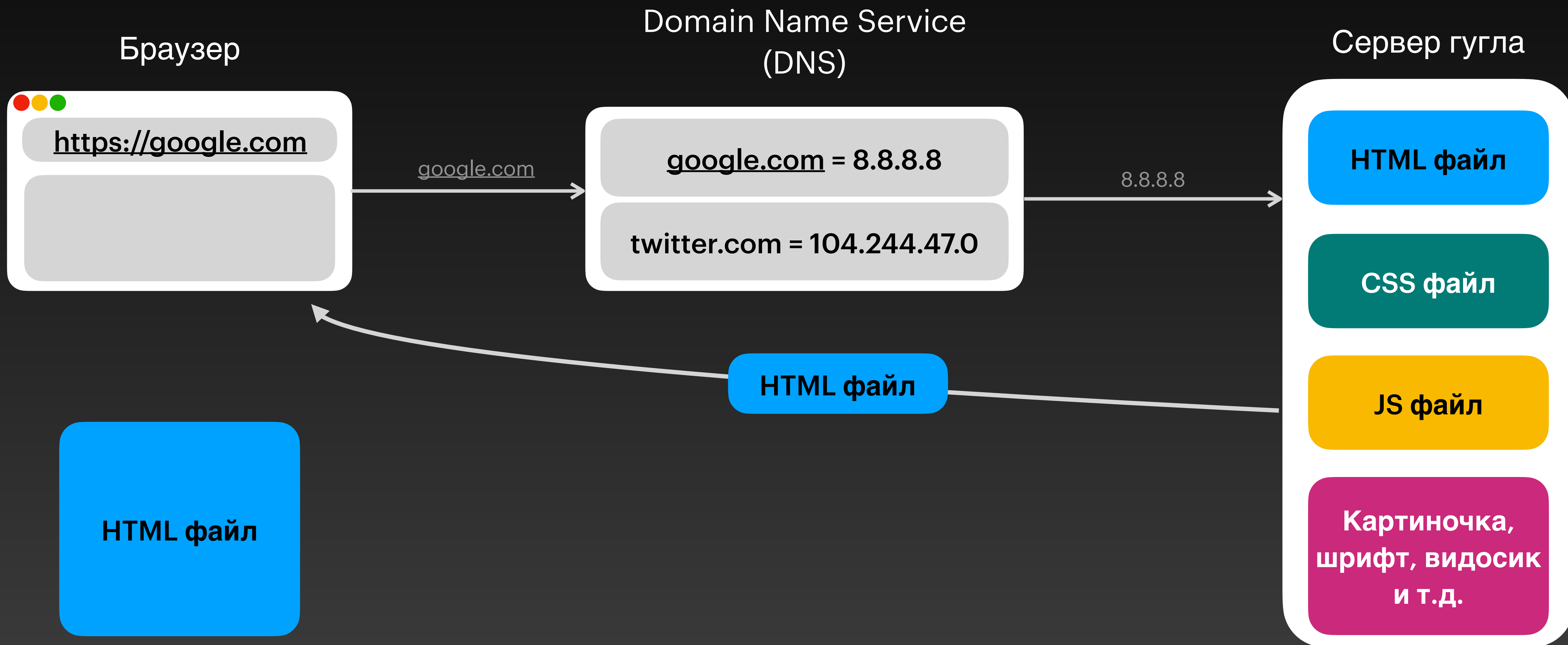


8.8.8.8

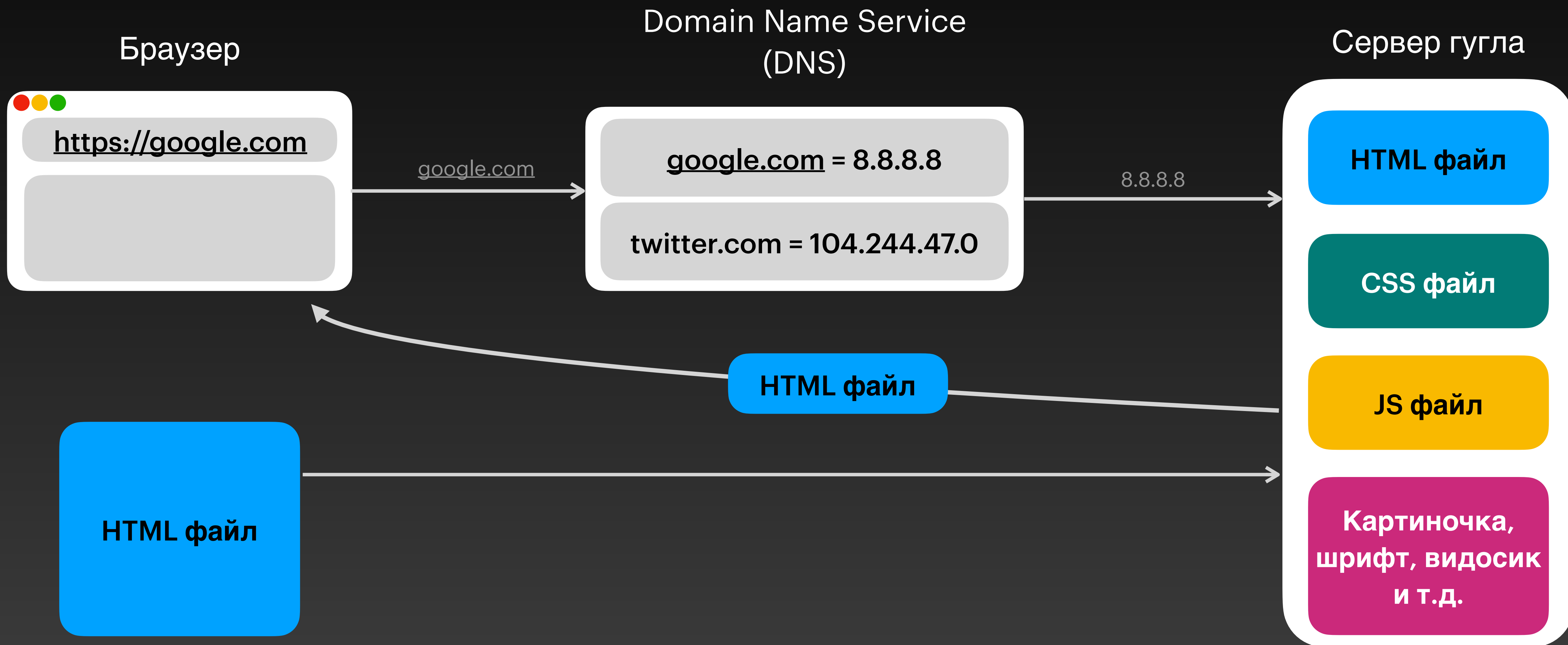
Сервер гугла



Как работает веб



Как работает веб



Как работает веб

