

WEBアプリケーション制作体験！



株式会社コンピュータ技研

WEBアプリケーション制作を通じて、WEBの仕組みや構成、仮想化技術などの知見を少しでも広げてみましょう！

このテキストでは以下の内容を軽く紹介します。

- ①WEBサイトが表示されるまでの仕組みとは？**
- ②WEBコンテンツの種類？**
- ③クライアントサイド・サーバサイド？**
- ④WEBコンテンツを公開に必要なサーバとは？**
- ⑤Amazon Web Service(AWS)とは？**
- ⑥仮想コンテナ(Docker)とは？**
- ⑦注目されている開発言語(Python)とは？**

私たちが何気なく見ているWEBサイトですが、ブラウザに表示されるまでに裏では様々な技術利用されています。

URLの先頭の文字列に「http://」と「https://」の2種類があります。これにより通信する方法や使われる技術が変わってきます。

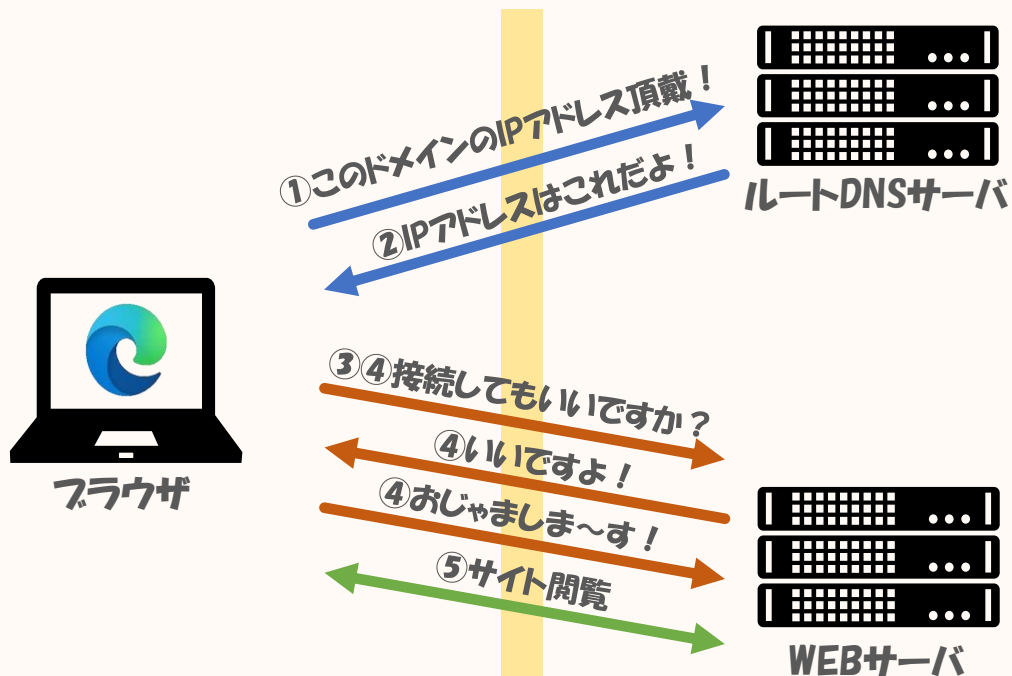
- **HTTP**(**H**yper**t**ext **T**ransfer **P**rotocol)

WebサーバとWebブラウザの間で、Web情報をやりとりするためのプロトコル(通信規則)です。

- **HTTPS**(**H**yper**t**ext **T**ransfer **P**rotocol **S**ecure)

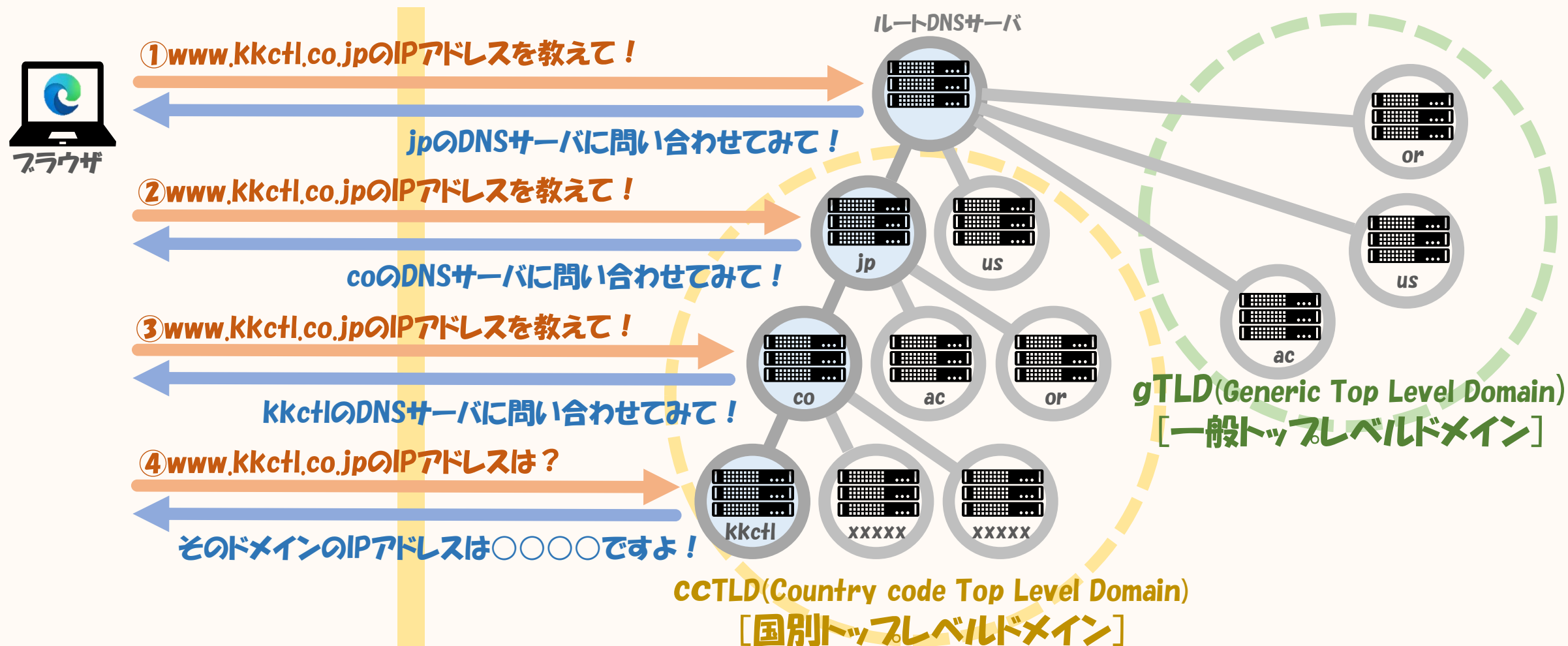
HTTPの通信を暗号化し、通信を安全に行えるようにしたプロトコルです。

HTTPでの接続よる、コンテンツ取得までの道のり



- ① ブラウザにURLを入力。
- ② URLのドメイン名に紐づけされたIPアドレスを **DNSサーバ** から取得。
- ③ IPアドレスを持つサーバにアクセス。
- ④ TCPによる**3ウェイハンドシェイク**認証後、HTTPによる通信開始。
- ⑤ アクセスされたサーバはWebページや画像データをブラウザに渡す。

DNS(Domain Name System)サーバは、ドメイン名に紐づく情報を管理していて、URLの問合せに対して、以下のような動きでIPアドレスを返します。



DNSサーバは、WEBサーバのアドレスだけでなく、メールサーバや他のDNS (ネームサーバ)のアドレスなども管理して名前の解決をします。

ルートDNSサーバとは、インターネットで利用されるDNSにおいて、ツリー構造の起点となっていて、問合せ時に一番最初に問合せされます。

DNSサーバが保持している主な情報は以下の通りです。

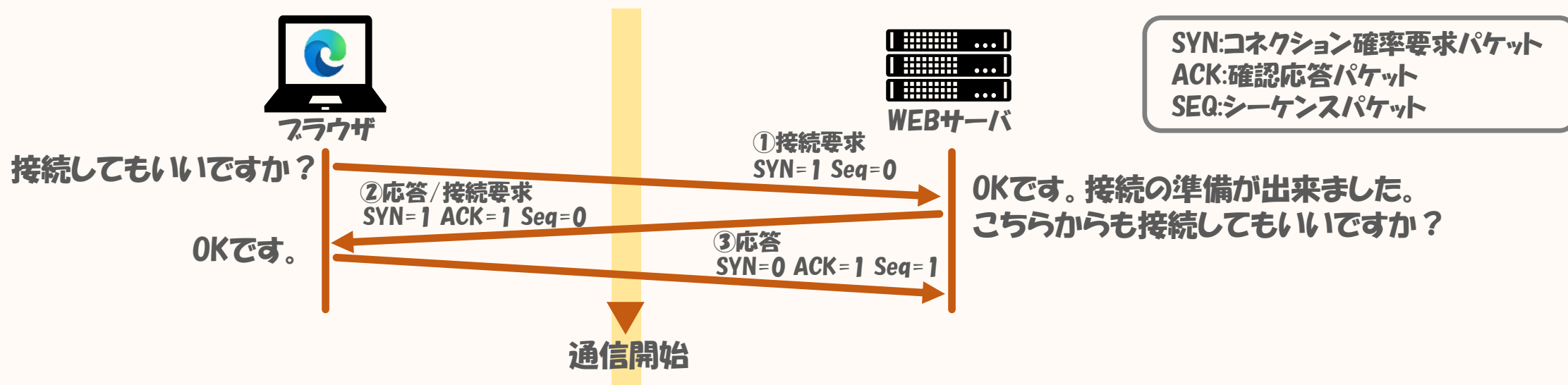
- | | |
|-----------|-----------------------------------|
| Aレコード | ドメイン名に対応するIPアドレス(IPv4)が記録されています。 |
| AAAAレコード | ドメイン名に対応するIPアドレス(IPv6)が記録されています。 |
| MXレコード | 対象ドメイン宛てのメール配信先(メールサーバ)が記録されています。 |
| NSレコード | ドメインのネームサーバのサーバ名が記録されています。 |
| CNAMEレコード | ドメインの別名が記録されています。「エイリアス」と呼ばれています。 |

3ウェイハンドシェイク認証とは？

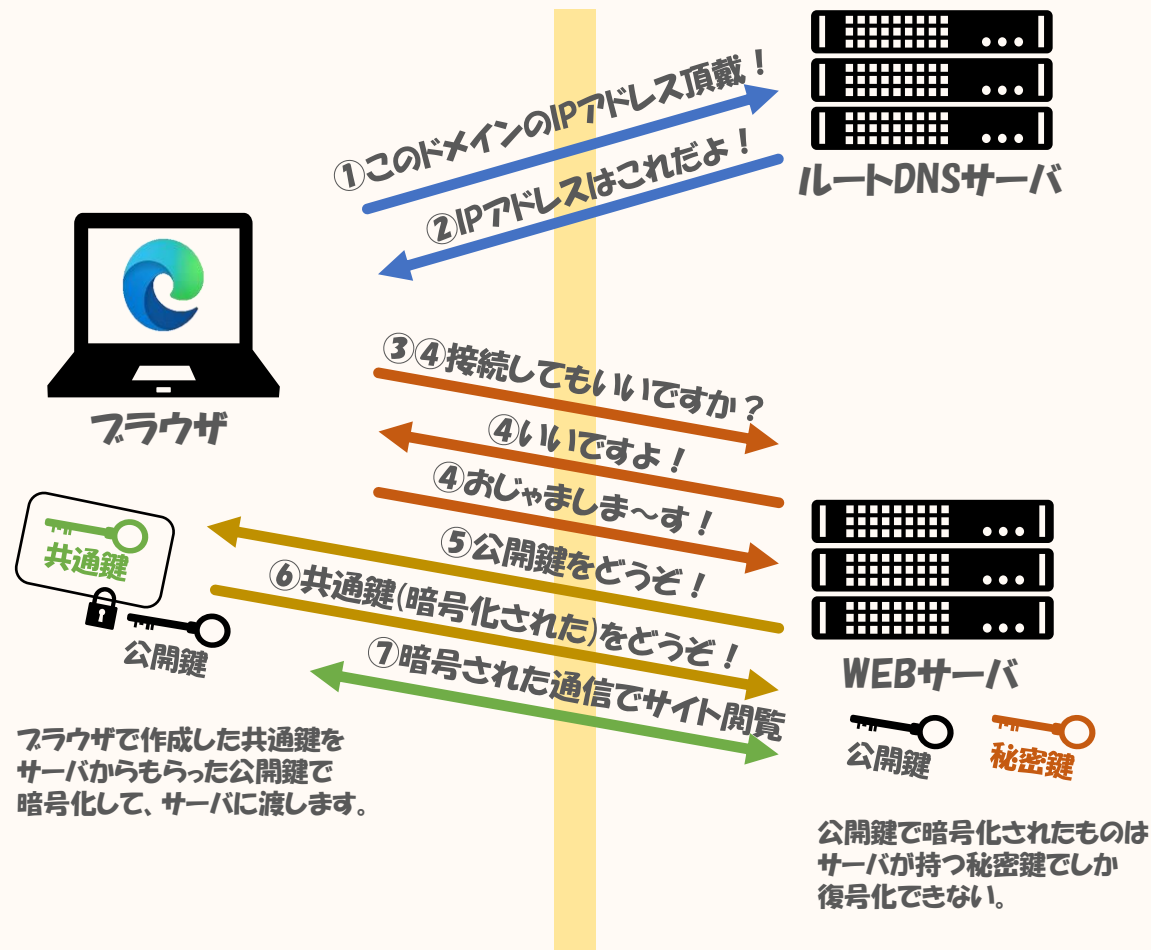
ネットワーク上でTCPによるコネクション(接続)を確立するための手順です。
なんのこっちゃってないますよね。簡単に説明します。

TCPとは、コネクション(接続)を確立するための標準的なプロトコルで、ポートを利用して、コンピュータ同士の接続の橋渡しをします。

コネクション(接続)を確立するために、以下の3回のやり取りします。



HTTPSでの接続による、コンテンツ取得までの道のり



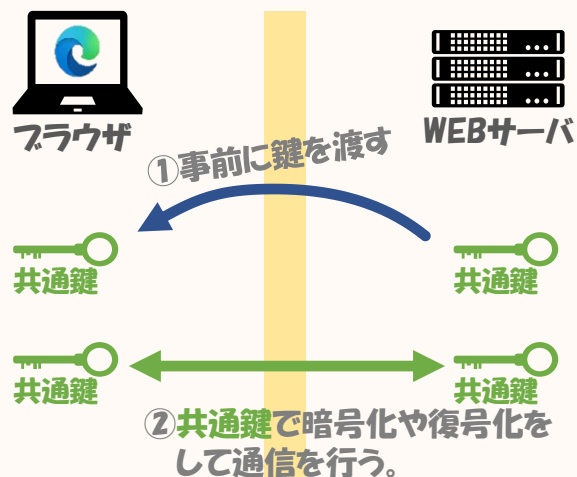
- ① ブラウザにURLを入力。
- ② URLのドメイン名に紐づけされたIPアドレスを**DNSサーバ**から取得。
- ③ IPアドレスを持つサーバにアクセス。
- ④ TCPによる**3ウェイハンドシェイク**認証後、HTTPSによる通信準備開始。
- ⑤ サーバは**デジタル証明(公開鍵)**をブラウザに渡す。
- ⑥ ブラウザは**デジタル証明書(公開鍵)**を使って**暗号化**した**共通鍵**をサーバに渡す。
- ⑦ その後は、**共通鍵**を使って暗号化された**通信プロトコル**で、サイトの表示やデータのやり取りをします。

※ 共通鍵を使って通信されている内容は暗号化されているため、悪意のある第三者から内容を見ることはできません。

通信を安全に行うために、「**共通鍵暗号方式**」または「**公開鍵暗号方式**」、**「ハイブリッド方式」**という暗号化技術が利用されています。

共通鍵暗号方式

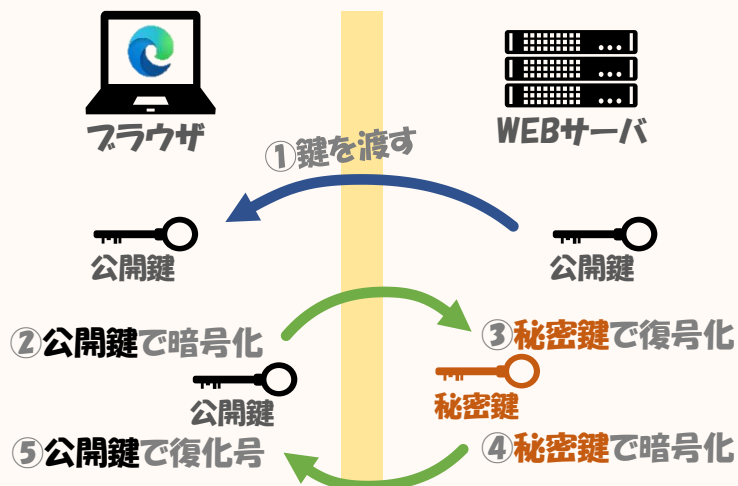
サーバとクライアントで共通の鍵を使って通信を暗号化と復号化をします。共通鍵を事前に引き渡す必要があり、その際に盗聴されるリスクがあります。



※ 共通鍵の引き渡し時に盗聴されるリスクがある。

公開鍵暗号方式

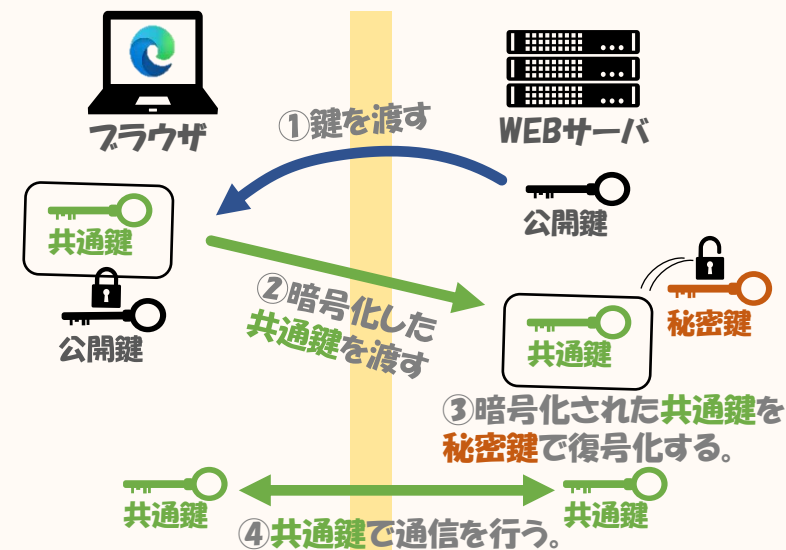
サーバは秘密鍵をクライアントは公開鍵を使って通信を暗号化します。公開鍵と秘密鍵は対になっていて、一方の鍵で暗号化したものは、もう一方の鍵でしか復号化できません。



※ 暗号化と復号化に時間がかかり過ぎるため遅い。

ハイブリッド方式

クライアントが公開鍵で共通鍵を暗号化してサーバに引き渡し、暗号化された共通鍵を秘密鍵で復号化します。その後の通信は共通鍵で暗号化されます。



Google ChromeはHTTPSに対応していないウェブサイトへ接続した場合は警告が表示されるようになりました。

またMicrosoft EdgeでもHTTPSのページからHTTPのページへ遷移する際に警告が表示されるようになっています。

サイト制作時に考慮が漏れていると、商品の注文や宿泊施設の予約時に、
「送信しようとしている情報は保護されません」
「サイトで使用されている接続は安全性が十分ではないため、情報が他人に読み取られる可能性があります。」
などと表示されてお客さんは購入を躊躇したり、取りやめる理由になります。

WEBコンテンツは大きく分けて2つの種類があります。

この2種類のコンテンツの組み合わせでサイトが構成されていることが多い。記事の変更が少ないページは**静的コンテンツ**で、入力フォームや商品の一覧などは**動的コンテンツ**で作成されています。

静的コンテンツ

サイトにアクセスがあったときに表示する、あらかじめ用意しておいたコンテンツのこと。誰が閲覧しても表示内容は同じになる。

動的コンテンツ

サイトにアクセスがあるたびに、プログラムによって画面を作成して表示するコンテンツのこと。ユーザや操作によって表示内容を変えられる。

静的コンテンツはあらかじめ用意されたコンテンツを画面に表示します。

表示速度も速く、セキュリティ対応も容易で、サーバダウンの発生率も低い。ユーザの操作でデータをサーバに送ることが出来ないため、入力が必要な処理には不向きなコンテンツ。

静的コンテンツのみで構成されたWEBサイトならWEBサーバさえあれば公開することが可能。

代表的な言語: HTML, CSS, JavaScript

関連用語: WEBサーバ、クライアントサイドプログラミング

動的コンテンツはサイトにアクセスされるたびに、プログラムが画面を作成して表示します。

表示内容をリアルタイムに変更ができるため、ユーザの操作により情報や、状況が変わっていくようなコンテンツに向いています。

ただ、都度サーバでコンテンツを作成するため、表示速度は静的コンテンツに比べて遅めで、コンテンツ作成用のプログラム実行環境として **APサーバ**や**DBサーバ**が必要となります。

代表的な言語: Java, PHP, R, Python, Perl

関連用語: **APサーバ**、**DBサーバ**、**サーバサイドプログラミング**

クライアントサイドや**サーバサイド**とは、実際にプログラムが処理される環境を指します。

クライアントサイド

こちら側で処理されるプログラムを**クライアントサイドプログラム**と呼びます。

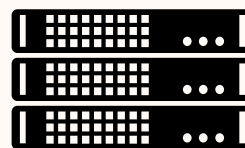


ブラウザ

Microsoft Edge
Google Chrome
Firefox
Safariなど

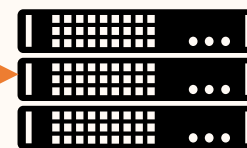
サーバサイド

こちら側で処理されるプログラムを**サーバサイドプログラム**と呼びます。



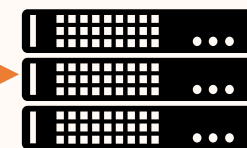
WEBサーバ

静的コンテンツを表示するための仕組みを持ったサーバソフトウェアまたはサーバ。



APサーバ

動的コンテンツを処理するための仕組みを持ったサーバソフトウェアまたはサーバ。



DBサーバ

大量のデータを高速に処理するためのソフトウェアまたはサーバ。

Webブラウザ上で動作するプログラムのこと。



主要なブラウザ

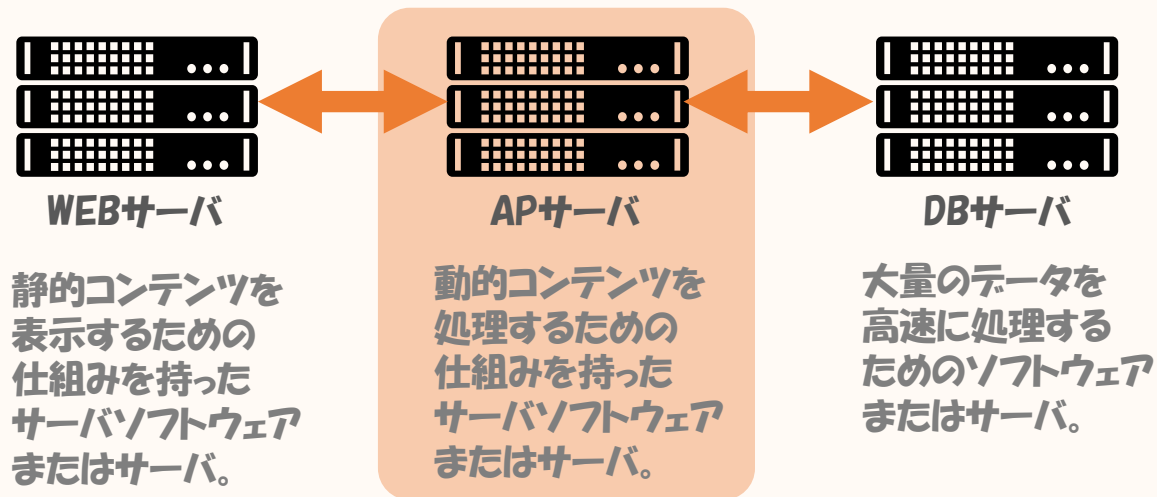
-  Microsoft Edge
-  Google Chrome
-  FireFOX
-  Safari

代表的な言語：HTML, CSS, JavaScript

メリット：HTMLやJavaScript、CSSの動作環境や開発環境は容易に準備できるため開発にかかる時間や費用が抑えられる。

デメリット：ブラウザの性質により動作が変わることがある。
またソースが閲覧できるため分析されやすい。

サーバ側(APサーバ)で動作するプログラムのこと。



APサーバ上のプログラムで作成したページをブラウザに送ります。
プログラムがサーバ上にあるため、ソースの解析をされることはない。

代表的な言語：Java,PHP,R,Python,Perl

メリット：サーバ側のソースが使えるため、大量のデータを扱ったり、複雑な計算なども可能。

デメリット：開発コストが高く、構築に時間がかかる。
プログラムの脆弱性対応など定期的なメンテが必要。

WEBサーバとは、利用者側のコンピュータに対しネットワークを通じて情報や機能を提供するサーバおよびソフトウェアのこと。
HTTPサーバやWWWサーバと呼ばれることもあります。
主には静的コンテンツを配布するための仕組みを持ったサーバソフトウェアです。

代表的なサーバ: NGINX, Apache, Cloudflare, LiteSpeed, Microsoft - IIS



CLOUDFLARE



LiteSpeed Web Server



Microsoft
IIS

APサーバとは、WEBアプリケーションを構成するサーバ側のプログラムの実行環境を提供するサーバおよびソフトウェアのこと。

WEBアプリケーションサーバやWAPサーバ

動的コンテンツを作成する仕組みを持ったサーバソフトウェアです。

※プログラムの開発環境は別途に必要ことが多い。

代表的なソフトウェア: Tomcat, GlassFish, WebSphere, gunicorn, uWSGI



Apache Tomcat



GlassFish



gunicorn uWSGI

DBサーバとは、システムが永続的に取り扱うデータを一元的に管理し、検索や保存、更新、削除、バックアップ、一貫性の確保などを行うサーバおよびソフトウェアのこと。

大量のデータを高速で安全に処理するためのサーバソフトウェアです。

代表的なソフトウェア: ORACLE, SQL Server, MySQL, PostgreSQL, MongoDB



WEBサーバを構築するには固定のグローバルIPアドレスが必要となります。理由としては前述した**DNS**に取得した**ドメイン名**を登録する必要があるからです。

ホームページを作成するだけであれば、**SaaSサービス**である**WIX**や**Word Press**などを利用する企業も多くなっています。

昔は**オンプレミス** [**自社運用**] で構築されることが普通でしたが、ハードウェアやソフトウェアの保守や更改にコストがかかるため、近年はクラウド上のサービスを利用しての構築が主流となっています。

WEBや開発環境の構築に**AWS**や**Docker**などがよく利用されています。

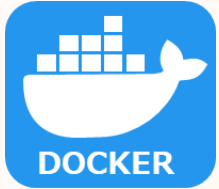
 Amazonが提供しているWebサービスの総称で、クラウドコンピューティングを利用して、ストレージやデータベース、サーバなど、90分類、700種類以上の様々なサービスの貸し出しをしています。

その中で今回のWEB制作研修に利用するサービスを紹介します。

AmazonEC2(**A**ma**z**on **E**lastic **C**ompute **C**loud)
必要なときにだけ使用できる仮想レンタルサーバ。

メリット: Webサイトの規模やアクセス量に応じて、サーバ容量や帯域などを
変更出来る

デメリット: 容量とアクセス量で料金の変動するため、見積もいが難しい。



Dockerとは軽量なコンテナ型仮想化サービスでOSとは独立したアプリケーション実行環境を構築することが出来ます。
構成ファイルやイメージファイルを利用することにより、再構築が短時間で可能。

メリット: 他の仮想サーバ技術より低コスト、低リソース。
Dockerイメージを利用することにより、構築難易度が低い。
環境の移行が簡単。

デメリット: 基本的にはLinuxOS上で動作するため、コンテナに構成出来るアプリケーションはLinux専用のものとなる。
通常的环境と異なるため、習得に時間がかかる。

DockerコンテナはホストOSと隔離されたアプリケーション領域を使用しているため、ホストOSの環境を汚すことなく新しい環境の構築・廃棄ができます。また、DockerイメージをDcokerHubからダウンロードすることにより、すでにある程度構築された状態から作業をすることができます。